Bacula Enterprise Dedicated Backup Solutions

Bacula Systems Documentation

Contents

Contents

Bacula offers a wide range of backup solutions presented below:

1 Storage Backend

Important: Storage Backend solutions are used with the Storage Daemon.

1.1 Deduplication

Aligned Volumes

Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. One of those challenges is the growing amount of data to be backed up, together with limited time to run backup jobs (backup window). Bacula offers several ways to tackle these challenges, one of them being *Deduplication Optimized Volumes*, which write Bacula Volumes in a deduplication friendly format.

This document is intended to provide insight into the considerations and processes required to successfully implement this backup technique.

Deduplication

Deduplication is a complicated subject. Generally speaking, it permits detecting that data being backed (usually blocks) have already been stored and rather than making an additional backup copy of the same data, the deduplication software just keeps a pointer referencing the previously stored data (block). Detecting that a block has already been stored is done by computing a hash code (also know as signature or fingerprint) of each block, and comparing the hash code with those of blocks already stored.

The picture becomes much more complicated when one considers where the deduplication is done: either on the server and/or on the client machines. In addition, most deduplication is done on a block by block basis, which some deduplication systems permitting variable length blocks and/or blocks that start at arbitrary boundaries rather than on specific alignments (sliding blocks).

Bacula's first step in doing deduplication is to offer an alternate (additional) Volume format that is aligned on specific block boundaries. This permits an underlying file system that does deduplication to efficiently deduplicate this new Bacula Enterprise Deduplication Optimized Volume format (or often called "Aligned" Volume format. Another way of describing this is that we have filtered out all the metadata and record headers and put them in the Metadata Volume (same as existing Volume format) and put only file data that can be easily deduplicated into the Aligned Volume.

Since there are a number of deduplicating file systems available on Linux systems (ZFS, lessfs, ddumbfs, SDFS (OpenDedup), LiveDFS, ScaleDFS, NetApp (across NFS), Epitome (OpenBSD), Quantum (in their appliance), ..., this Bacula implementation allows the users to choose what deduplication engine they want to use.

Although a number of these dedupe engines use large blocks (128K), the block sizes on most can range from 4K to 1M. Bacula with Deduplication Optimized Volumes has implement several new Storage Device directives that permit tuning what Bacula does to match what is optimal for the underlying deduplication engine.

Advantages of Deduplication

There are two main advantages of deduplication:

- Deduplication can significantly reduce the disk space needed to store your data. In good cases, it may reduce disk space by half, and in the best cases, it may reduce disk space by a factor or 10 or 20.
- Storing data (backups) can be much faster since a whole block of data can be replaced by a pointer to an existing block that is already stored on disk.

Cautions About Using Deduplication

Here are a few of the things that you should be aware of before using deduplication techniques.

- Deduplication takes a lot of CPU and memory resources. To do efficient and fast deduplication, you will need lots of CPU power (for computing hash codes), and additional amounts of RAM memory (for fast lookups of hash codes). Bacula Systems provides a Deduplication Sizing Calculator that could help you calculating memory needs.
- The extra CPU power and memory needed can be expensive, but SSD can largely solve the memory requirements at a reasonable cost.
- Due high CPU and RAM requirements as well as the extra complexity of deduplication, performance tuning is more complicated.

- If your disk develops a bad block instead of damaging one file (that may be stored many times), it may damage all (hundreds or thousands) files that contain the same data. That is you have a single point of failure that can cause more damage than would happen on a non-deduplicated system. These problems can be reduced by using deduplicating systems that checksum everything such as ZFS, or by using hardware or RAID technology.
- Possible loss of data can be mitigated by using the duplicate copy feature of a number of the deduping filesystems (copies in ZFS). In fact some systems such as ZFS permit you to limit the number of references to a single block of data. Once that limit is reached, it will store the block another time. Can minimize any loss due to bad disk blocks.
- Deduplication collisions can cause data corruption. That is if the deduplicating system uses a weak hash code such as MD5, SHA1, Fletcher, … it is possible that several blocks with different data will hash to the same value. If the system does not do a byte for byte comparison, then the second block stored will be corrupted when it is read back, because it will get the first block that was stored rather than the second block that had the same hash code.
- The problem of hash code collisions can be mitigated by using a stronger (and usually more CPU intensive) hash code (SHA256 or SHA512) or by doing a byte by byte comparison with the block that is already stored. ZFS has options for enabling both of these techniques.

How Bacula Deduplication Optimized Volumes Work

- First, please be aware that you need the **aligned-sd.so** or the **bacula-sd-aligned-driver-x.y.z.so** Storage daemon plugin for Aligned Volumes to work. Please do not forget to define the **Plugin Directory** in the Storage daemon configuration file **bacula-sd.conf**.
- Aligned Volumes are enabled by specifying the Aligned keyword on a DeviceType directive in each Device resource in the bacula-sd.conf where you want to use aligned volumes:
- Aligned Volumes must have a unique Media Type that is different from Volumes that are used on non-Aligned devices.

```
DeviceType = Aligned
```

- When Aligned Volumes are enabled, the Device will create two files for each Volume. For example, if the Volume name is Test001. The files will be named Test001 and Test001.add. Test001 will contain all the metadata (filename, date created, permissions, …) in the same format as non-aligned Volumes, and Test001.add will contain all the aligned file data (i.e. it will be block aligned).
- If you are writing the Volume with a single Job, each file backed up will have its data stored contiguously (i.e. no block interleaving. We recommend this way to write Volumes.
- If you set a maximum volume size for an aligned volume device, Bacula will only allow one backup Job at a time. In any case, it is probably more efficient to only backup to aligned volumes with one Job at a time. Rather than rely on Bacula to allow only one backup Job at a time, we strongly recommend that you following the advice in the next item below.
- We strongly recommend that you only write a Volume with a single Job at a time. This will be most efficient use of the Volume. To ensure that only one Job accesses the Volume at a given time, use the following directive in each Aligned device:

```
Maximum Concurrent Jobs = 1
```

  Note, later versions of Bacula Enterprise (6.4.11) enforce this recommendation even if you do not specifically set the Maximum Concurrent Jobs.

New Storage daemon Device Directives

- Plugin Directory = <directory-name>

  This directive is required to enable the aligned volume feature. You must also have the Aligned Volume plugin loaded in the defined Plugin Directory before the Storage daemon is started.

- Device Type = Aligned

  This directive is required to make the Device write aligned volumes. Once this is turned on, each Bacula Volume will be split into two Volumes, one for the metadata, which does not deduplicate well, and one for the file data, which can be deduplicated.

- File Alignment = <bytes>

  Once the aligned volume format has been chosen, this is the key directive that will make the file data deduplication friendly. Each new file that is written into the .add file will be aligned to a multiple of this size. The value for this directive should be equivalent to the basic deduplication block size used by your Operating System. For NetApp, the deduplication block size is fixed at 4K. For ZFS the deduplication block size is set via the recordsize** variable, and is by default 128K.

- Padding Size = <bytes>

  When Bacula is ready to write a block, and if it is not full, the block will be padded to the next multiple of the Padding Size by filling with zeros. If you want all blocks to be the same size as the Maximum Block Size, then set the Padding Size to the same value as the Maximum Block Size. The default for this is 0 bytes. In general, you should set it to the minimum physical block size used by your Operating System. For NetApp, the minimum physical block size is 4K, and for ZFS it appears that the minimum block size is 512 bytes. Setting the Padding Size larger than the smallest physical block size will cause Bacula to use more disk space than is necessary. Setting it too small or to zero will, in general, do no harm.

- Aligned Device = <directory-name>

  This directive is similar to the Archive Device. It is optional, but if specified the Aligned Volume (xxx.add) will be placed in the Aligned Device directory specified, which is presumably different from the Archive Device where the Metadata Volume is placed. In the absence of the Aligned Device directive, both volumes will be placed in the Archive Device directory. Having two directives allows placing the Metadata volume in a directory (or partition) that is not deduplicated while the Aligned data volume can be deduplicated.

- Minimum Aligned Size = <bytes>

  This Directive is set to 4096 by default, and any file that is this size or smaller will be placed in the Metadata volume rather than the Aligned Volume. This allows you to put smaller files that are unlikely to deduplicate well into the Metadata volume rather than overload the Aligned volume with data that does not deduplicate well.

Below we supply a few examples. The directives show should be in the **bacula-sd.conf** file within a **Device** resource:

Listing 1: EMC Data Domain DD800

```
    Device Type = Aligned
    Media Type = Aligned
    Maximum Block Size = 64K
    Minimum Block Size = 0
    File Alignment = 4K
```

(continues on next page)

(continued from previous page)

```
    Padding Size = 4K
    Minimum Aligned Size = 4K
    Maximum Concurrent Jobs = 1
```

Listing 2: ZFS with default record size of 128K

```
    Device Type = Aligned
    Media Type = Aligned
    Maximum Block Size = 128K
    Minimum Block Size = 0
    File Alignment = 128K
    Padding Size = 512
    Minimum Aligned Size = 4096
    Maximum Concurrent Jobs = 1
```

Listing 3: NetApp

```
    Device Type = Aligned
    Media Type = Aligned
    Padding Size = 4K
    File Alignment = 4K
    Maximum Block Size = 64K
    Minimum Block Size = 0
    Minimum Aligned Size = 4K
    Maximum Concurrent Jobs = 1
```

Listing 4: ddumbfs

```
    Device Type = Aligned
    Media Type = Aligned
    Maximum Block Size = 128K
    Minimum Block Size = 0
    File Alignment = 128K
    Padding Size = 512
    Minimum Aligned Size = 4096
    Maximum Concurrent Jobs = 1
```

Listing 5: lessfs

```
    Device Type = Aligned
    Media Type = Aligned
    Maximum Block Size = 128K
    Minimum Block Size = 0
    File Alignment = 128K
    Padding Size = 512
    Minimum Aligned Size = 4096
    Maximum Concurrent Jobs = 1
```

Things to Know About Deduplication

- Generally system tools such as **ls** or **du** do not know about sparse files (files with holes such as Bacula's Aligned Volumes), nor do they know about deduplicating and compressing filesystems such as ZFS. As a result, the size numbers that such system tools produce can be misleading.
- In order to really know the disk space used by a given file or a directory, you generally will need to use tools such as **zpool** and **zfs** to report disk utilization, and file sizes.

Things to Know About Bacula

- You must take particular attention to define a unique Media Type for devices that are Aligned as well as for each Virtual Autochanger that uses a different Archive Device directory. If you use the same Media Type for an Aligned device type that you use for a normal disk Volume, you run the risk that you will have data corruption on disk Volumes that are used on Aligned and non-Aligned devices.
- Blocks are may be written in any size and the files can be aligned at any value. However, they must generally be a multiple of 1024 bytes.
- When the values that you have specified for **Padding Size** is smaller than **File Alignment**, Bacula will generate Volumes that are sparse (i.e. that will have holes or unused areas of the Volume). This is normal, and it permits aligning the beginning of a file at a suitable deduplication boundary without wasting space.
- If you are using the Virtual Disk Changer feature of Bacula, it should work fine.
- We strongly recommend not using the Bacula disk-changer script, because it was written only for testing. Instead of using disk-changer, we recommend using the Virtual Disk Changer feature of Bacula, for which there is a specific white paper.
- We strongly recommend that you update all File daemons that are used to write data into an Aligned Volume. It is not required, but old File daemons do not have the newer FD to SD protocol, so consequently the **Minimum Aligned Size** is not respected for any older File daemons. Other than the fact that all file data will be written to the Aligned volume regardless of the file size and the value set for **Minimum Aligned Size** old File daemons will work correctly.

Things to Know About ZFS

- First and most important, you **must** set the Bacula **File Alignment** value to the same sized as the ZFS **recordsize** (128K by default). If you do not, deduplication will not work correctly.
- Second, you **must** set the Bacula **Maximum Block Size** to the same size or greater than the **File Alignment** value. If you do not, deduplication on ZFS may not work.
- Our testing with a ZFS kernel module installed on a Ubuntu 12.04 system indicates that for our simple dataset (the Bacula Source + binaries), the following Bacula parameters are optimal:

  ```
  Device Type = Aligned
  Maximum Block Size = 128K
  Minimum Block Size = 0
  File Alignment = 128K
  Padding Size = 512
  Minimum Aligned Size = 4K
  ```

- Our testing (as noted above) indicates that the following ZFS parameters are optimal, where **tank** is the pool:

  ```
  sudo zfs set atime=off tank
  sudo zfs set compress=on tank
  sudo zfs set dedup=on tank
  sudo zfs set recordsize=128k tank (default)
  ```

- ZFS has a surprisingly large number of settable options. The key ones for doing deduplication are: dedup, compression, and recordsize.
- ZFS has a number of tools such as **zdb**, but unfortunately they do not seem to have complete documentation. This means that in some of the examples given below, we have had to guess at the meaning of certain values that are displayed by the various commands.
- As mentioned above, a number of the ordinary Unix tools such as **ls** and **du** either do not correctly report sizes for sparse files (ls) or do not report allocated disk space taking into account deduplication and compression (ls and du).
- To get an accurate view of how much disk and memory is actually being used with deduplication and/or compression turned on, one must resort to the output from several ZFS commands. The most important being **zpool list tank**. See below for more details.

Output from a Bacula Job report might look something like the following:

```
...
FD Files Written:       2,126
SD Files Written:       2,126
FD Bytes Written:       141,738,979 (141.7 MB)
SD Bytes Written:       142,047,709 (142.0 MB)
Rate:                   141869.5 KB/s
Last Volume Bytes:      367,264,779 (367.2 MB)
```

The interesting thing is that the size of the data backed up is 142.0 MB as indicated by the SD Bytes Written, but the Volume size used by this job is 367.2 MB. Note: this number represents the sum of the last address of the metadata Volume and the Aligned Data Volume.

Is the data more than doubling in size?

The answer is **No** the actual sized used with dedupe and compression enabled on ZFS is about 78.7 MB (1/2 the original size) as will be shown below. The largest disk address used by Bacula is indeed on the order of 300 MB, but much of the Volume has holes and no disk space is allocated to those holes.

When one sets the File Alignment to 128K as shown above, each of the 2,126 files that were written will be aligned to a 128K byte boundary, while the original data was aligned to a 4K byte boundary since it came from a Linux ext4 filesystem. Since the Padding Size is 512, Bacula will write short blocks zero filed to the next 512 byte boundary for every file that is less than 128K then it will seek to the next 128K boundary to begin writing the next file. This means that there will be holes in the file, and the apparent file size (Last Volume Bytes) will indicate the address of the last byte in the file rather than the real number of bytes allocated (used) on disk.

The simplest way to know how much space was really used is to look at the output from a **zpool list** command:

```
zpool list tank
NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH
tank  19.9G  78.7M  19.8G     0%  1.02x  ONLINE
```

and we see under the ALLOC column, the space used is 78.7 MB. This is approximately half of the original space of 142 MB that the original data used. The difference is due to compression in this case.

After doing a total of 10 identical Full backups, we get the following output:

```
zpool list tank
NAME   SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH
tank  19.9G  84.2M  19.8G     0%  10.00x  ONLINE
```

Notice that 10 times the data (i.e. 10 X 142 MB) increased the total ZFS disk space allocated to only 84.2 MB. This is due to the fact that it was perfectly deduplicated and only a small amount of Bacula metadata and ZFS pointers were written in addition to the original data. Bottom line, ZFS stores 1.42GB of data using 84.2MB of disk space! At the same time, the Bacula Aligned Volume appears to be using 3.4GB of data. See below for more details.

Another command that can be useful to understand what is going on is **zdb -DD tank**, shown in the next figure:

```
zdb -DD tank
DDT-sha256-duplicate: 2724 entries, size 306 on disk, 165 in core
DDT-sha256-unique: 45 entries, size 7884 on disk, 9830 in core

DDT histogram (aggregated over all DDTs):

bucket            allocated                     referenced
______  ___________________________   ________ ___________________
refcnt  blocks  LSIZE  PSIZE  DSIZE   blocks  LSIZE  PSIZE  DSIZE
------  ------  -----  -----  -----   ------  -----  -----  -----
     1      45  5.38M  2.23M  2.23M       45  5.38M  2.23M  2.23M
     8   2.61K   334M  75.4M  75.4M    26.1K  3.26G   754M   754M
    16      42  5.25M   466K   466K      860   108M  9.18M  9.18M
    32      10  1.25M   530K   530K      400    50M  20.7M  20.7M
 Total   2.70K   346M  78.6M  78.6M    27.4K  3.42G   786M   786M

dedup=10.00, compress=4.46, copies=1.00, dedup*compress/copies=44
```

The important numbers from above all all in the Total row. The first is the 78.6M under the allocated

DSIZE column (second column). It corresponds to the real disk space used by the data. The second number is 27.4K, the total number of referenced blocks (sixth column). This number is important when computing the memory requirements to keep all the the Dedupe table (DDT) in memory. Each DDT item takes approximately 170 to 300 bytes, so when the number of referenced blocks times 300 becomes a large percentage of your main RAM, all disk operations will slow down because the ZFS must page the lookup tables. The final number of importance is the 3.42G under the referenced LSIZE column (seventh column). This number is equivalent to the largest file address that Bacula reports in the Job report.

More on ZFS

Good sources of additional information on using and tuning ZFS can be found at:

```
http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide

http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide

http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide

http://www.brendangregg.com/Slides/zfsperftools2012.pdf
```

Creating a ddumbfs

You can use this command to create the ddumbfs filesystem:

```
mkddumbfs -s 200G -B 128k /d0/ddumbfs/

and used this in fstab

oparent=/d0/ddumbfs/        /tank        fuse.ddumbfs    noauto 0 0
```

Restrictions and Limitations

- Aligned Volumes do not permit running multiple concurrent Jobs to the same Volume. So you **must** add the following directive to all Aligned volume Device resources in the Storage daemon's configuration file:

  ```
  Maximum Concurrent Jobs = 1
  ```

  Despite the fact that a only a single Job can write to an Aligned Volume at a given time, you may run multiple simultaneous Aligned backup Jobs by having each Job write to a different Device. The easiest way to set this up is to use a Bacula Enterprise Virtual Autochanger (see the White Paper on this subject).

  Note that multiple Jobs may be written to the same Aligned volume, the only constraint is that they may not use the volume simultaneously. For example, if you have a nightly backup of a specific client machine, that client may write to the same Volume on different nights.

- You must take care to define unique Media Types to Aligned Volumes that is different from Media Types for non-Aligned Volumes.

- If you are using VTL software that is not part of a Bacula Enterprise release, it is unlikely it will work. If the VTL simulates a tape drive, it definitely will not work. Bacula's Aligned Volumes work only with Bacula disk Volumes. If you need a VTL, consider using Bacula's Virtual Autochanger feature (there is a Bacula Enterprise White Paper on this topic).

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Global Endpoint Deduplication

Bacula Enterprise offers two versions of the Global Endpoint Deduplication plugin:

Global Endpoint Deduplication

This is the version available to our customers for a few years.

Global Endpoint Deduplication 2

The Global Endpoint Deduplication version 2 offers the following advantages:

- Maintenance tools are eased and faster to process. There is only a need to run *vacuum* and *optimize* with GED 2
- All new data are grouped per job in the containers, which makes backup and restore jobs faster. The performance gain is particularly important on restore jobs. This is due to the data being aligned in the containers.
- The Containers structure brings more flexibility in regards to containers files size and hole creation, thus the storage space management is optimized. The new technology should create no fragmentation.
- GED 2 is available for production use from Bacula Enterprise version 16.0

Global Endpoint Deduplication

Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. One of those challenges is the growing amount of data to be backed up, together with limited time to run backup jobs (backup window). Bacula Enterprise offers several ways to tackle these challenges, one of them being *Global Endpoint Deduplication*, which minimizes network transfer and Bacula Volume size using deduplication technology.

This document is intended to provide insight into the considerations and processes required to successfully implement this backup technique.

Deduplication

Deduplication is a complex subject. Generally speaking, it detects that data being backed up (usually chunks) has already been stored and rather than making an additional backup copy of the same data, the deduplication software keeps a pointer referencing the previously stored data (chunk). Detecting that a chunk has already been stored is done by computing a hash code (also known as signature or fingerprint) of the chunk, and comparing the hash code with those of chunks already stored.

The picture becomes much more complicated when one considers where the deduplication is done. It can either be done on the server and/or on the client machines. In addition, most deduplication is done on a block by block basis, with some deduplication systems permitting variable length blocks and/or blocks that start at arbitrary boundaries (sliding blocks), rather than on specific alignments.

Advantages of Deduplication

- Deduplication can significantly reduce the disk space needed to store your data. In good cases, it may reduce disk space needed by half, and in the best cases, it may reduce disk space needed by a factor of 10 or 20.
- Deduplication can be combined with compression to further reduce the storage space needed. Compression depends on data type and deduplication depends on the data usage (on the need or the will of the user to keep multiple copies or versions of the same or similar data). Bacula takes advantage that both techniques work perfectly together and combines them in it's Dedupengine.
- Deduplication can significantly reduce the network bandwidth required because both ends can exchange references instead of the actual data itself. It works when the destination already has a copy of the original chunks.
- Handling references instead of the data can speed up most of the processing inside the Storage Daemon. For example, Bacula features like copy/migrate and Virtual Full can be up to 1,000 times faster. See the following article for more information on this subject.

Cautions About Using Deduplication

Here are a few of the things that you should be aware of before using deduplication techniques.

- To do efficient and fast deduplication, the Storage Daemon will need additional CPU power (to compute hash codes and do compression), as well as additional RAM (for fast hash code lookups). Bacula Systems can help you to calculate memory needs.
- For effective performance, the deduplication Index should be stored on SSDs as the index will have many random accesses and many updates.

- Due the extra complexity of deduplication, performance tuning is more complicated.
- We recommend Index and Containers are stored in xfs or ext4 file systems. But we are also compatible with zfs file system.
- Deduplication collisions can cause data corruption. This is more likely to happen if the deduplicating system uses a weak hash code such as MD5 or Fletcher. The problem of hash code collisions is mitigated in Bacula by using a strong hash code (SHA512/256).
- Deduplication is not implemented for tape devices. It works only with disk-based backups.
- The immutable flag is not compatible or does not apply to the dedup index or dedup containers.

Aligned Volumes

Bacula Systems' first step in deduplication technology was to take advantage of underlying deduplicating filesystems by offering an alternative (additional) Volume format that is aligned on specific chunk boundaries. This permits an underlying file system that does deduplication to efficiently deduplicate the data. This new Bacula Enterprise Deduplication Optimized Volume format is often called "Aligned" Volume format. Another way of describing this is that we have filtered out all the metadata and record headers and put them in the Metadata Volume (same as existing Volume format) and put only file data that can be easily deduplicated into the Aligned Volume.

Since there are a number of deduplicating file systems available on Linux or Unix systems (ZFS, lessfs, ddumbfs, SDFS (OpenDedup), LiveDFS, ScaleDFS, NetApp (via NFS), Epitome (OpenBSD), Quantum (in their appliance), etc. This Bacula Aligned Volume implementation allows users to choose the deduplication engine they want to use. More information about Deduplication Optimized Volume Format can be found in **Bacula Systems**' `DedupVolumes` whitepaper.

Global Endpoint Deduplication

Bacula Systems' first data source agnostic deduplication technology is the *Global Endpoint Deduplication* feature. With Global Endpoint Deduplication, Bacula will analyze data at the block level, then Bacula will store only new chunks in the deduplication engine, and use references in standard Bacula volumes to chunks stored in the deduplication engine. The deduplication can take place at the File Daemon side (saving network and storage resources), and/or at the Storage Daemon side (saving storage resources).

The remainder of this white paper will discuss only Global Endpoint Deduplication.

How Bacula Global Endpoint Deduplication Works

- First, please be aware that you need the **dedup-sd.so** or the **bacula-sd-dedup-driver-x.y.z.so** Storage Daemon plugin for Global Endpoint Deduplication to work. Please do not forget to define the `Plugin Directory` in the Storage Daemon configuration file `bacula-sd.conf`.
- Dedup devices are enabled by specifying the `dedup` keyword as a DeviceType directive in each disk Device resource in the bacula-sd.conf where you want to use deduplicated Volumes.

```
DeviceType = Dedup
```

- You must pay particular attention to define a unique Media Type for devices that are Dedup as well as for each Virtual Autochanger that uses a different Archive Device directory. If you use the same

Media Type for a Dedup device type as for a normal disk Volume, you run the risk that you will have data corruption on disk Volumes that are used on Dedup and non-Dedup devices.

- When Global Endpoint Deduplication is enabled, the Device will fill in disk volumes with chunk references instead of the chunks. Bacula encrypted data, and very small files will be stored in the Volumes as usual. The deduplicated chunks are stored in the "Containers" of the Dedupengine, and are shared by all other dedup-aware devices in the same Storage Daemon.
- We advise to set a limit on the number of Jobs or the usage duration when working with dedup Volumes. In case you prefer to use `Maximum Volume Bytes`, please consider that two Catalog fields are considered when computing the volume size. `VolBytes` represents the volume size on disk and `VolaBytes` considers the amount of non-dedup data stored in the volumes, i.e., the rehydrated data. If the directive `Maximum Volume Bytes` is used for a dedup Volume, Bacula will consider both VolBytes and VolaBytes values to check the limits.

Global Endpoint Deduplication During Backup Jobs

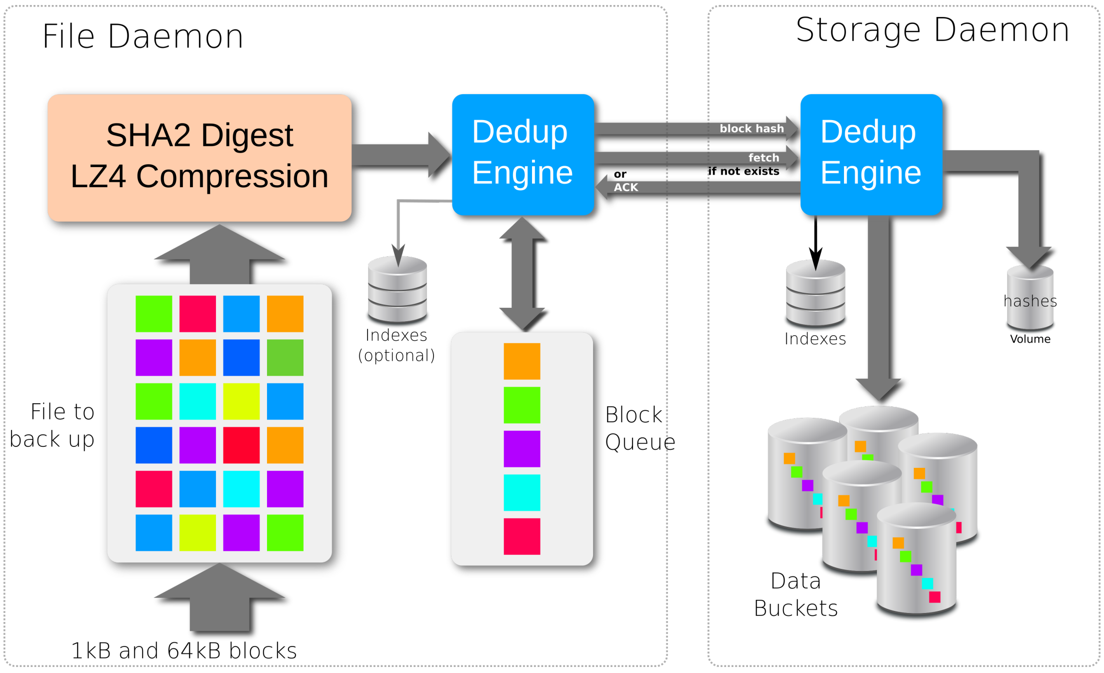


Fig. 1: Backup Scenario with bothsides deduplication

- When starting a Backup Job, the Storage Daemon will inform the File Daemon that the Device used for the Job can accept dedup data.
- If the Fileset uses the `dedup = bothsides` option, the File Daemon will compute a strong hash code for each chunk and send references to the Storage Daemon which will request the original chunk from the File Daemon if the Dedupengine is unable to resolve the reference.
- If the Fileset uses the `dedup = storage` option, the File Daemon will send data as usual to the Storage Daemon, and the Storage Daemon will compute hash codes and store chunks in the Dedupengine and the references in the disk volume.
- If the Fileset uses the `dedup = none` option, the File Daemon will send data as usual to the Storage Daemon, and the Storage Daemon will store the chunks in the Volume without performing any deduplication functions.
- If the File Daemon doesn't support Global Endpoint Deduplication, the deduplication will be done on the Storage side if the Device is configured with `DeviceType = dedup`.

Global Endpoint Deduplication During Restore Jobs

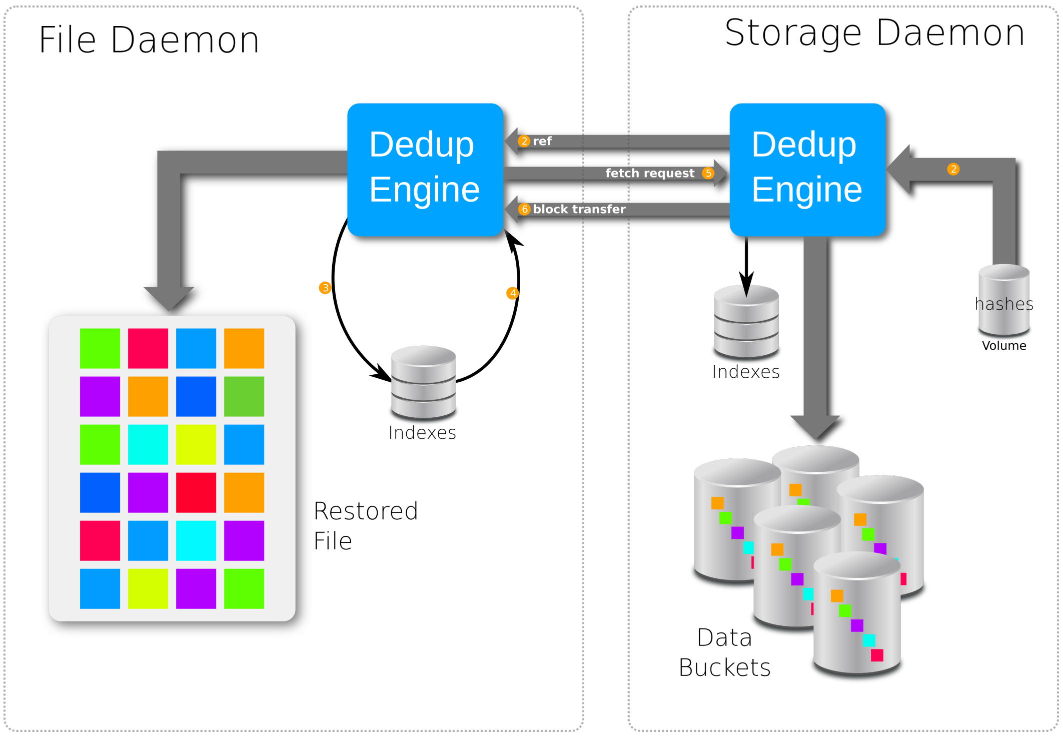


Fig. 2: Restore Scenario when usingthe directive ‘Enable Client Rehydration’

- If the directive `Enable Client Rehydration` is set to “yes” in the File Daemon configuration file, the Storage Daemon will send references to the File Daemon during a restore. If the directive is set to “no”, the Storage Daemon will rehydrate all the references and send the chunks to the File Daemon.
- When the File Daemon receives a reference, it will try to rehydrate the data using local data, see section *Client Side Rehydration* below.

Client Side Rehydration

Attention: Client Side Rehydration is deprecated

Client side rehydration is deprecated and should not be used with Bacula versions greater than 12.0.0. If you run into one of the specific cases described below for which this feature could be very useful, please contact the Support Team.

The File Daemon can try to do some rehydration on its own using local data. This feature can increase restore speeds for systems connected through a slow network and doesn’t consume any resources at backup time.

This feature is activated with a FileDaemon resource directive called **Enable Client Rehydration** in `bacula-fd.conf`.

We recommend against using this feature on a client connected through a fast network, because the extra disk accesses and computation can slow down the speed of the restore jobs.

To take advantage of this feature you must understand how it works. At restore time, the client receives the original location, the offset and the hash of every chunk to restore. It then looks to see if the original file still exists, opens it and checks if the chunk at the given offset matches the given hash. If it matches, the File Daemon uses it and does not download the chunk from the Storage Daemon.

It is obvious that to take advantage of this feature, you must:

- Restore the data to another location.
- Have some piece of the original data in the original location.

This feature can be very helpful to retrieve an old version of the current data.

Notice that this feature doesn't work for files that are not going into the Deduplication Engine like small files or when data encryption is used. This also doesn't work when the data is transformed by Bacula before reaching the Deduplication Engine. For example, when compression is used or when backing up Windows systems without the `portable = yes` option in the Fileset.

Unfortunately there is no evidence of the efficiency of the algorithm in the Job report yet. The only evidence is the `read chunk` counter shown by the `dedup usage` command that is not incremented for chunks found on the Client.

Storage Daemon Deduplication Related Directives

- Plugin Directory = `<directory-path>`

  This directive tells the Storage Daemon where to find plugins. The file **dedup-sd.so** or the **bacula-sd-dedup-driver-x.y.z.so** must be present in this directory before starting the Storage Daemon.

- Dedup Directory = `<directory-path>`

  Deduped chunks will be stored in the **Dedup Directory**. This directory is common for all **Dedup** devices configured on a Storage Daemon and should have a large amount of free space. We advise you to use LVM on Linux Systems to ensure that you can extend the space in this directory. The **Dedup Directory** directive is mandatory. We recommend that you do not change this directory afterward, because if you make a mistake, it would invalidate all of your backups. If you do change the `Dedup Directory` directive, the following files must be moved to the new directory:

  - `*.blk`

- Dedup Index Directory = `<directory-path>`

  Indexes will be stored in the **Dedup Index Directory**. Indexes will have a lot of random update accesses, and will benefit from fast drives such as SSD drives. By default, the **Dedup Index Directory** is set to the **Dedup Directory**.

  As with the **Dedup Directory**, we recommend against changing the `Dedup Index Directory` directive. If you do, the following files and directories must be moved to the new directory:

  - `*.idx`
  - `*.tch`
  - `recovery`
  - `recovery.new`

  The file `bee_dde.tch.new` is a temporary file used by the `optimize` part of the vacuum that remain when the process is interrupted. This file don't need to be moved.

- Maximum Container Size = `<size>`

No container will be allowed to grow to more than `<size>` bytes. When this size is reached, a new container file will be created. The default value is zero, meaning there is no limit. This limit is useful when you store your containers on a filesystem that limits the size of the file to a pretty low value.

The number of containers is limited to 511, so we recommend to keep this value unlimited or pretty high, at least 1TB. This value may be modified after the initialization of the DedupEngine. If a container is already bigger than the new limit, then no new data will be written to it, but its size will not be reduced. Other containers will comply with the new limit.

- Device Type = Dedup

  This directive is required to make the Device write Dedup volumes. Once turned on, Bacula will use references in Volumes and will store data chunks into specific container files.

  Once a Device has been defined with a certain Type (such as Dedup, Aligned, File or Tape), it cannot be changed to another Type. If you do so, the Bacula Storage Daemon will not be able to properly mount volumes that were created before the change.

```
# From bacula-sd.conf
Storage {
 Name = my-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Subsys Directory = /opt/bacula/working

 Plugin Directory = /opt/bacula/plugins
 Dedup Directory = /mnt/bacula/dedup/containers
 Dedup Index Directory = /mnt/SSD/dedup/index    # Recommended to be on fast␣
↪local SSD storage
 Maximum Container Size = 4TB # Maximum 511 containers can be created, please␣
↪adapt to your need
}

Device {
  Name = "DedupDisk"
  Archive Device = /mnt/bacula/dedup/volumes
  Media Type = DedupVolume
  Device Type = Dedup                   # Required
  LabelMedia = yes
  Random Access = Yes
  AutomaticMount = yes
  RemovableMedia = no
  AlwaysOpen = no
}
```

Deduplication Related Director Daemon Fileset Directive

Within the Director, the Global Endpoint Deduplication system is enabled with a Fileset Option directive called **Dedup**. Each Include section can have a different behavior depending on your needs.

```
# Use the default dedup option of 'storage' side deduplication
 Fileset {
   Name = FS_BASE
   Include {
     Options {
       Dedup = storage
     }
     File = /opt/bacula/etc
   }

   # Do not dedup my encrypted data
   Include {
     Options {
       Dedup = none
     }
     File = /encrypted
   }

   # Minimize the network transfer by using 'bothsides' dedup option
   Include {
     Options {
       Dedup = bothsides
     }
     File = /bigdirectory
   }
 }
```

The **Dedup** Fileset option can have the following values:

- **storage** - All the deduplication work is done on the Storage Daemon side if the device type is **dedup**. The File Daemon will send all data to the SD just as it normally would. (Default value)
- **none** - Disable dedpulication on both the File Daemon and Storage Daemon.
- **bothsides** - The deduplication work is done on the File Daemon and the Storage Daemon.

About Fileset Compression

The data stored by the Global Endpoint Deduplication Engine is automatically compressed using the LZ4 algorithm. Using the Fileset `Compression = LZO|GZIP` option might reduce the deduplication efficiency, and compressing the data twice consumes extra CPU cycles on the client side. Thus we advise that you do not use client-side GZIP or LZO compression when using a Dedup Device. To prevent such an inefficient configuration, we recommend setting the `Allow Compression` directive in a Director Storage resource to `No`:

```
# cat bacula-dir.conf
...
Storage {
   Name = Dedup
   Allow Compression = No    # Disable Fileset Compression
```

(continues on next page)

(continued from previous page)

```
                                # option automatically
   Address = baculasd.lan
   Password = xxx
   Media Type = DedupMedia
   ...
}
```

Deduplication Related File Daemon Directive

The `Enable Client Rehydration` FileDaemon directive is optional and allows Bacula to try to do rehydration using existing local data, see section *Client Side Rehydration*. The valid values are `Yes` or `No`. The default is `No`.

Starting with 8.2.0, the FileDaemon `Dedup Index Directory` in `bacula-fd.conf` directive is deprecated and replaced by `Enable Client Rehydration` directive.

```
# cat /opt/bacula/etc/bacula-fd.conf
 FileDaemon {
...
   Enable Client Rehydration = yes
 }
```

Things to Know About Bacula

- You must pay particular attention to define a unique Media Type for devices that are Dedup as well as for each Virtual Autochanger that uses a different Archive Device directory. If you use the same Media Type for a Dedup device type as for a normal disk Volume, you run the risk that you will have data corruption on disk Volumes that are used on Dedup and non-Dedup devices.
- Dedup devices are compatible with Bacula's Virtual Disk Changers
- We strongly recommend that you not use the Bacula `disk-changer` script, because it was written only for testing purposes. Instead of using `disk-changer`, we recommend using the Virtual Disk Changer feature of Bacula, for which there is a specific white paper.
- We strongly recommend that you update all File Daemons that are used to write data into Dedup Volumes. It is not required, but old File Daemons do not support the newer FD to SD protocol, and consequently the Global Endpoint Deduplication cannot not be done on the FD side.
- The immutable flag is compatible with dedup volumes, see more details in Volume Protection Enhancements and Volume Protection.

Deduplication Engine Vacuum

Over time, you will normally delete files from your system, and in doing so, it may happen that there will be chunks that are stored in dedup containers that are no longer referenced.

In order to reclaim these unused chunks in containers, the administrator needs to schedule a `vacuum` option of the `dedup` command. The `vacuum` option will analyze dedup volumes and mark any chunks that are not referenced as free, thus allowing the disk space to be reused. The vacuum command can run while other jobs are running.

```
* dedup
Dedup Engine choice:
     1: Vacuum data files
     2: Cancel running vacuum
     3: Display data files usage
Select action to perform on Dedup Engine (1-3): 1
The defined Storage resources are:
     1: File1
     2: Dedup
Select Storage resource (1-2): 2
Connecting to Storage daemon Dedup at localhost:9103 ...
3000 Found 1 volumes to scan for MediaType=DedupMedia
Ready to read from volume "Vol1" on dedup data device "Dedup-Dev1" (/mnt/
↪bacula/dedup/volumes).
End of Volume at file 0 on device "Dedup-Dev1" (/mnt/bacula/dedup/volumes),␣
↪Volume "Vol1"
Ready to read from volume "Vol2" on dedup data device "Dedup-Dev1" (/mnt/
↪bacula/dedup/volumes).
End of Volume at file 0 on device "Dedup-Dev1" (/mnt/bacula/dedup/volumes),␣
↪Volume "Vol2"
Ready to read from volume "Vol3" on dedup data device "Dedup-Dev1" (/mnt/
↪bacula/dedup/volumes).
End of Volume at file 0 on device "Dedup-Dev1" (/mnt/bacula/dedup/volumes),␣
↪Volume "Vol3"
End of all volumes.
Vacuum cleaning up index.
Vacuum done.
```

Deduplication Engine Status

Is it possible to query the Deduplication Engine to get some information and statistics. Note that the current interface is oriented toward developers and is subject to change. For example, the `Stats` counters can be reset to estimate the work done by the engine for one job or for one period of time. Here is an example output of the dedup `usage` command, followed by an explanation of each section in the output:

```
* dedup storage=Dedup usage
Dedupengine status:
 DDE: hash_count=1275 ref_count=1276 ref_size=78.09 MB
    ref_ratio=1.00 size_ratio=1.13 dde_errors=0
 Config: bnum=1179641 bmin=33554393 bmax=335544320 mlock_strategy=1
    mlocked=9MB mlock_max=0MB
 Containers: chunk_allocated=3469 chunk_used=1275
```

(continues on next page)

(continued from previous page)

```
   disk_space_allocated=101.2 MB disk_space_used=68.87 MB
   containers_errors=0
Vacuum: last_run="06-Nov-14 13:28" duration=1s ref_count=1276
   ref_size=78.09 MB vacuum_errors=0 orphan_addr=16
Stats: read_chunk=4285 query_hash=7591 new_hash=3469 calc_hash=3470
 [1] filesize=40.88KB/499.6KB usage=36/484/524288   7% ***...............
 [2] filesize=40.13KB/589.0KB usage=18/286/524288   6% **5...............
 [3] filesize=25.47KB/655.2KB usage=7/212/524288    3% *4................
...
 [64] filesize=4.096KB/4.096KB usage=0/0/524288      0% ..................
 [65] filesize=53.25MB/63.90MB usage=800/960/524288 83% ......3***********
```

DDE:

- `hash_count` Number of hashes in the Index.
- `ref_count` Number of references in all the Volumes.
- `ref_size` The total of all rehydrated references in all the volumes. This is the size that would be needed if deduplication was not in use.
- `ref_ratio` The ratio between `ref_count` and `hash_count`.
- `size_ratio` The ratio between `ref_size` and `disk_space_used`.
- `dde_error` The number of invalid data found in the Index.

Config:

- `bnum` The capacity of the hash table in the Index. This is the number of `buckets` in the Tokyo Cabinet hash database.
- `bmin` The minimum size of the hash table in the Index. Bacula will not go below this value when resizing the Index.
- `bmax` The maximum size of the hash table in the Index. Bacula will not go above this value when resizing the Index. Zero means no limit.
- `mlock_strategy` This is the strategy to apply to lock only the hash table or the hash table and Index into memory.
  - `0` Do not lock any memory.
  - `1` Use at most `mlock_max` bytes to lock only the hash table of the Index.
  - `2` Use at most `mlock_max` bytes to lock all the Index.
- `mlocked` The current number of bytes locked by the Index.
- `mlock_max` The maximum number of bytes that the Index can lock.

Containers:

- `chunk_allocated` The number of chunks allocated in all containers.
- `chunk_used` The number of chunks that are really in use.
- `disk_space_allocated` The space allocated for all containers.
- `disk_space_used` The space that is really used inside all containers.
- `containers_error` The number of errors related to the containers.

Vacuum:

- `last_run` The date of the last vacuum.
- `duration` The time the vacuum took to complete.
- `ref_count` Number of references handled by last vacuum.
- `ref_size` The total rehydrated size of all references handled by last vacuum.
- `vacuum_errors` Number of various errors reported during last vacuum. You can get more information in the trace file.
- `orphan_addr` Number of distinct addresses found in the volumes but not found in the Index during the last vacuum. These appear when the Storage Daemon crashes, because the DedupEngine is cleaned up but not the volumes.

Stats:

- `read_chunk` How many chunks have been read since the last reset.
- `query_hash` Number of chunk index queries since the last reset.
- `new_hash` How many new entries in the chunk index since the last reset.
- `calc_hash` How many hashes have been calculated since the last reset.

In the DDE section, both ratios give a different view of what is happening inside the dedup engine. While `ref_ratio` gives a true value, `ref_size` tell us how effective the dedup engine is, because we are more concerned about the space saved. The last one takes into account the LZ4 compression and also any possible disparity between small and big chunks. For example, if there are a lot of small chunks with a high dedup ratio, `ref_ratio` will be high, but the space saved will be small as it concerns only small blocks.

`ref_count` and `ref_size` are calculated during a vacuum and are used to reset the counter with the same names in section `DDE`. These two counters are then updated by future backups.

Example:

```
[7] 7k filesize=4.1GB/22.3GB usage=569910/3104523/3145728 \
                    18% 67003000000000000000000..........684**9
```

- `[7]` is the ID of the container. This is the number at the end of the container file which resides in the Dedup Directory defined in bacula-sd.conf. In this case, “bee_dde0007.blk”
- `7k` is the size limit for the chunks inside this container.
- `4.1GB/22.3GB` means that the container size (as shown with ’ls -l’) is 22.3GB, but only 4.1GB are used in this container. This means that 18.2GB (22.3GB - 4.1GB) can be written into this container without making it grow. Notice that ’ls -l’ doesn’t accurately represent the size of a container file when ’holepunching’ is used because some of this space can be unallocated (think ’sparse file’). “ls -s”, “stat” and “du” can display the size that is really used by the container. A command like this gives the size in bytes:

```
$ echo $((`stat -c "%b*%B" bee\_dde0007.blk`))
```

- `usage=569910/3104523/3145728`. The 2 first values are the same as 4.1GB and 22.3GB but are expressed in number of chunks. The third value is the the size of the `bit array` holding the map of the container. This array grows in increments of 64k = 524288 bits every time the current array gets full.
- `18%` is the usage of the container, here 18%=569910/3104523

- `67003000000000000000000000..........684**9` is the map of the container sliced in 40 parts. A "." means that the part is empty. "0" means that less 10% of the part is used, and "9" means that the part is used between 90% and 99%. Finally "*" means that the part is fully used.

Disaster Recovery

Catalog

The Catalog doesn't contain any reference to the deduplication engine. However, the dedup volumes' records in the Catalog are used during the vacuum process. For this reason, you must make sure to have the Catalog properly restored before starting a dedup vacuum process.

Volumes

If a dedup Volume is in the Catalog but not on disk, a dedup vacuum process will stop and report an error.

Index

The Index is essential for the deduplication engine. In the case of a disaster, contact Bacula Systems Support Team.

Free Space Map (FSM)

The deduplication engine creates a copy (during a commit) of the FSM after every important operation in the `recovery` sub-directory. When the deduplication engine is not shut down properly, the last copy is used as a reference by the recovery procedure to remove any operations that started after the time of the last commit and that could be incomplete. When the original and the copy of the FSM are lost, it is still possible to rebuild the FSM using references found in volumes. See section *Detect, Report and Repair Dedupengine Inconsistencies*.

Containers

Containers hold chunks of data. When a container (or part of a container) file is lost, the data is lost and it is not recoverable by Bacula. Use the deduplication engine recovery tools (*Detect, Report and Repair Dedupengine Inconsistencies*) to identify chunks of data that are lost and restore the deduplication engine consistency.

Dedupengine

The deduplication engine is the heart of Bacula's Global Endpoint Deduplication. It has to store, to index and to retrieve the chunks. All chunks are stored in the `Container Area` and registered in the `Index`. The `Index` is the bottleneck of the deduplication process because all operations need to access it randomly, and very quickly. Memory caching and storing the Index on SSD drives will help to maintain good performance.

The Deduplication Index maintains all the hashes of all chunks stored in the Dedup Containers. To get effective performance very fast low latency storage is critical. For large back up data it is best to have the Containers and Deduplication Index on the same hardware server with the Deduplication Index on solid-state drives (SSDs). The faster the disk performance, the faster and more efficient the deduplication process and the data protection will be. In production environments it is best to avoid configurations which introduce latency and delays in the storage infrastructure for the Deduplication Index. It is therefore best to avoid spinning disks, remote access configurations like NFS or CIFS and virtualized SDs. These can be acceptable for small containers (1-2TB) or to perform tests but will normally not provide acceptable performance in larger production environments.

Sizing the Index

The size of the index depends on the number of chunks that you want to store in the deduplication engine. An upper limit would be 1 chunk per file plus 1 chunk per 64K block of data.

$$\text{number\_of\_chunks} = \text{number\_of\_files} + \frac{\text{data\_amount}}{64\text{K}}$$

If all you have is the storage capacity of your Storage Daemon and want to maximize it, you must know the average compressed size of the chunks you expect to store in Containers. If you don't know the size, you may use 16K.

$$\text{number\_of\_chunks} = \frac{\text{storage\_capacity}}{16\text{K}}$$

When you know the number of chunks, you can calculate the size of your index.

$$\text{index\_size} = 1.3 * \text{number\_of\_chunk} * (8 + 70)$$

The index can be split into two parts: the table and the records.

$$\text{index\_size} = \text{table\_size} + \text{record\_size}$$

$$\text{table\_size} = 1.3 * \text{number\_of\_chunk} * 8$$

$$\text{record\_size} = 1.3 * \text{number\_of\_chunk} * 70$$

The table part is small and is accessed by all operations. The record part is bigger and is sometimes not used for read operations.

Table 1: Samples of Index size for chosen Storage sizes

Storage size	Index size	Table part	Record part
1 TB	6.3 GB	0.65 GB	5.7 GB
10 TB	63.3 GB	6.5 GB	56.9 GB

For good performance, you must try to lock the entire Index into memory, if this is not possible due to lack of memory resources, keeping at least the hash table in memory is highly recommended.

But these are not the only requirements. Bacula needs some extra space on disk and in memory to optimize and resize the Index. We recommend the following:

- Be sure to have 3 times the index_size on an SSD drive for the Index.
- Try to have index_size+table_size of RAM for the Index.
- At least be sure to have 2 times the "table_size" of RAM for the Index.

Setting up the Index size

The Index is based on a hash table that by design has a fixed size. A B-Tree structure is used to handle collisions in the hash table. The size of the table is important. If too small, the table will have to handle overflow that will slowdown the Index. If too big, the table will consume space and memory uselessly. The table can be resized online and Bacula takes advantage of the vacuum procedure to optimize the table size when needed. Creating the table at the right size from the start will ensure good performance from the beginning and avoid a reduction in performance. The user can define the minimum and maximum sizes of the table. At the end of the vacuum, if the amount of data to delete is large, or if the size of the table is unbalanced regarding the amount of remaining data, Bacula resizes the table to a size equal to 1.3 time the number of hashes remaining in the Index. This new size will be adjusted to match the minimum and maximum values chosen by the user.

$$bnum\_min < table\_size * 1.3 < bnum\_max$$

The default values for bnum_min is 33,554,432 and 0 for bnum_max, meaning that their is no limit. These numbers are the number of chunks that the Index can handle efficiently. A chunk can have a size between 1K to 64K. 16K is a good mean value. This means that the default index range is well suited for a storage space between 1TB and 10TB.

Keep in mind that the size of the index affects the amount of memory required to lock the index in memory.

Locking the index into memory

The operating system caches data that is used often in memory. Unfortunately the huge amount of data going in and out of the Storage Daemon usually wipes out the Index data from the system cache. The alternative is to force the system to map and lock some parts of the Index into memory.

The user has a choice between 3 strategies:

- 0 nothing is locked into memory
- 1 try to lock the table part of the Index into memory
- 2 try to lock the entire Index into memory

Bacula will not allocate more than the maximum value defined by the user (mlock_max) and will check the amount of memory available to not overload the system.

See how to change these variables in section *Commands to Tune the Index*.

Commands to Tune the Index

Bacula Enterprise 8.2 added 4 new parameters to tune the Index. These parameters are initialized with default values when the Dedupengine is created or when Bacula upgrades the Dedupengine from an older version. These parameters may be modified at any time. They will be saved inside the Dedupengine and will be used during the next vacuum.

The Dedupengine can be tuned by changing some internal variables. To have a good understanding of how the deduplication engine works, be sure to read sections *Sizing the Index* and *Commands to Tune the Index*.

- `bnum_min` The minimum capacity of the hash table in the Index. Bacula will not go below this value when resizing the Index.
- `bnum_max` The maximum capacity of the hash table in the Index. Bacula will not go above this value when resizing the Index. Zero means no limit.
- `mlock_strategy` This is the strategy to lock the Index into memory. You have the choice between 3 strategies:
  - `0` Do not lock any memory.
  - 1 Use at most `mlock_max` bytes to lock only the hash table of the Index into memory. (the default)
  - 2 Use at most `mlock_max` bytes to lock all the Index into memory.
- `mlock_max` The maximum amount of bytes that may be used to lock the Index into memory. Zero means no limit. (the default)

Each of these variables may be modified using the dedup command together with the name of the variable. The previous value is displayed for reference.

```
*dedup storage=Dedup bnum_min=33554393
3000 dedupsetvar bnum_min previous value was 33554393
*dedup storage=Dedup bnum_max=33554393
3000 dedupsetvar bnum_max previous value was 0
*dedup storage=Dedup mlock_strategy=1
3000 dedupsetvar mlock_strategy previous value was 1ff
*dedup storage=Dedup mlock_max=4096MB
3000 dedupsetvar mlock_max previous value was 0
```

You can review all of these values at once using the `dedup usage` command. At the top of the output you have the section `Config:`:

```
* dedup storage=Dedup usage
Dedupengine status:
...
 Config: bnum=1179641 bmin=33554393 bmax=33554393 mlock_strategy=1
    mlocked=9MB mlock_max=0MB
 ...
```

See the section *Deduplication Engine Status* for an explanation of the other variables.

These values will be used during the next vacuum if the Index needs to be `optimized`. You can force an `optimize` by adding the option `forceoptimize` to the the `dedup vacuum` command.

```
* dedup storage=Dedup vacuum forceoptimize
```

To force the Dedupengine to use a new `mlock` value without running a `dedup vacuum`, you may use the `dedup tune indexmemory` command.

```
* dedup storage=Dedup tune indexmemory
```

Punching holes in containers

Some Linux filesystems like XFS and EXT4 have the ability to `punch a hole` into files.

A portion of the file can be marked as unwanted and the associated storage released. Of course when a process writes into such a hole, the filesystem allocates space to this area.

Because the use of this technique can increase fragmentation of the filesystem and contribute to slower performance, it is recommended to avoid it when not needed, even though Bacula does its best to use it in a way that will not significantly impact performance.

The `hole punching` can happen in two places in the `DDE`:

1. detect and release large unused areas in containers,
2. prevent the allocation of chunks in these holes and prefer areas that are too small to be converted into holes.

Both of these processes are independent. As soon as you set up a `hole_size`, the `DDE` tries to allocate space outside of areas that are good candidates for hole creation, even if no holes have been created before.

Theory: creating holes

Because these holes can be reused by any container or file on the filesystem, this approach contributes to its fragmentation. That is why you must keep the size of these holes large enough to not reduce the performance of the filesystem. It as been shown that reading or writing random blocks of 4MB is done at a speed similar to sequential reads or writes. That is why we recommend setting the hole_size to 4MB. Smaller values can increase the work for the filesystem to manage all these small holes, reduce the performance, and make filesystem recovery processes (fsck) take longer. Using a higher value would reduce the probability to find such an unused amount of space inside the containers.

The DDE doesn't store the holes that it has created and doesn't use the information stored in the filesystem itself. The DDE creates the holes on top of the previous ones, and the filesystem ignores the requests for areas that are already holes.

The holes are aligned on the hole_size boundaries that we call extents. Remember that containers handle chunks of different sizes, and their sizes are not necessarily powers of 2, so they can span extents. Spanning chunks have weird consequences on the holes:

1. A single used chunk spanning two extents will prevent the conversion of these 2 extents into holes.
2. A hole that has free spanning chunks at one or both ends holds more space than the space that has been given back to the filesystem.

Theory: smart allocating in between holes

As previously stated, if an unused area is big enough, only the part that is aligned on the hole_size boundaries will be converted into a hole. This allows some space around these holes that is still allocated by the filesystem and can be used without "consuming" any new space. The DDE will chose to allocate new chunks in these spaces first, even if these areas have not been converted into holes yet because the DDE relies on existing free space and not on holes that have been created in the past.

When all space between holes has been allocated, the system goes back to the sequential allocation strategy and uses space in existing holes and finally allocates space at the end of the file.

Commands to create and manage holes

Add the holepunching option to the vacuum command to create the holes at the end of the vacuum procedure. The command in bconsole is:

```
* dedup vacuum holepunching storage=<DeviceName>
```

The first time you use the holepunching option, the DDE sets the hole size to 4194304 (4MB). The size is stored in the hole_size variable and can be modified or initialized before the first use. The option forceoptimize can be used together with the holepunching option without restriction. The time required to identify and create holes should not require more than 10s per TB.

You can change the hole_size to any value that is a power of 2 bigger than 1 MB. There is no upper limit, but values above 32MB are probably excessive. To change the hole_size, use the command:

```
* dedup storage=<DeviceName> hole_size=<Size_in_Byte>
```

For example you can chose a smaller value with the aim of releasing more space.

```
*dedup storage=Dedup hole_size=1048576
3000 dedupsetvar hole_size previous value was 4194304
```

This new value will be used the next time the vacuum is run with the holepunching option. However, this value will be immediately used by the allocation process to avoid using free space that could be released by the next holepunching procedure.

You can disable smart allocation by setting the value to zero. Notice that this value will set the default value to 4MB the next time you use the holepunching option in the vacuum command.

```
*dedup storage=Dedup hole_size=0
3000 dedupsetvar hole_size previous value was 4194304
```

You can review this value using the `dedup usage` command. At the top of the output you have the section `Hole::`

```
* dedup storage=Dedup usage
Dedupengine status:
...
 HolePunching: hole_size=1024 KB
 ...
```

Quiesce and Unquiesce

It is possible to quiesce the dedupengine to safely copy its data without shutting down the DDE. The commands `quiesce` and `unquiesce` allow to freeze and unfreeze the DDE.

```
* dedup storage=Dedup quiesce
3900 quiesce successful
* dedup storage=Dedup unquiesce
3900 unquiesce successful
```

Note: This functionality is available as of version 10.2.

When the `quiesce` command is run, all running backups and restores are suspended. If a `scrub` is running, it is paused. If a `vacuum` is running, the quiesce waits for the end of the vacuum before returning. When the DDE is frozen, you can backup or copy all the data related to the DDE. The data are in a crash-consistent state, this means that after a recovery, the data will be consistent. When the `unquiesce` command is run, all the backups and restores resume from the point where they had previously stopped. A `scrub` continues from where it was interrupted.

Detect, Report and Repair Dedupengine Inconsistencies

The `dedup vacuum` command provides three options: `checkindex`, `checkmiss` and `checkvolume` to detect, report and possibly repair inconsistencies in the DDE. `checkindex` can be used with the two others. When `checkmiss` and `checkvolume` are used together, `checkmiss` is ignored.

The `checkindex` and `checkvolume` options use a temporary file `chunkdb.tch` that stores the hash for every *suspicious* chunk to save multiple computations.

The three options will log information to the trace file.

checkindex option of the vacuum command

The option `checkindex` checks the consistency of the Index with itself and the coherence between the Index and the FSM. When multiple entries in the Index address the same chunk in one container (an address collision), the procedure reads the chunk, calculates the hash and deletes all invalid entries from the index. This procedure is executed before reading the volumes, and it iterates through the index twice: Once to detect collisions, and one more time to delete all invalid entries.

The `checkindex` option displays some statistics in the trace file:

```
cleanup_index_addr_duplicate unset2miss=0
cleanup index Phase 1 cnt=703783 badaddr=0 suspect=0 unset=0 2miss=0 miss=0
    (count=703784 err=0 2miss_err_cnt=0)
cleanup index Phase 2 cnt=703783 2miss=0 (count=703784 err=0 2miss_err=0)
```

- `cnt`: the number of data entries in the Index.
- `badaddr`: the number of entries in the Index with a fanciful address that don't match any container or any chunk inside a container.
- `suspect`: the number of colliding addresses that must be checked.
- `unset`: the number of addresses that were unexpectedly marked as free in the FSM and that have been temporary marked as used until the vacuum determines if the entry is needed or not.
- `2miss`: the number of new missing entries.
- `miss`: the number of entries that are missing, meaning that there is no matching chunk in the containers. This includes the newly created entries.
- `count`: the number of entries including the meta data (cnt + 1)
- `err`: a counter for uncommon errors.
- `2miss_err`: the number of errors when creating or converting an erroneous entry into a missing one.

A Scrub process always starts a `checkindex` as its final action.

checkmiss and checkvolume options of the vacuum command

The option `checkvolume` is deprecated since the availability of the Scrub process. In future releases the `checkvolume` option will be silently replaced by the option `checkmiss`.

These options search the Index for every reference found in the volumes. This can significantly increase the time of the vacuum if the Index doesn't fit into memory. Be sure to check that using the command `dedup storage=Dedup tune indexmemory`.

The option `checkmiss` simply creates dummy entries when a reference in not found in the Index. This entry indicates that the chunk is missing and could be resolved by future backups or by a Scrub. This option is less resource intensive than the `checkvolume` because it doesn't access the containers.

The option `checkvolume` checks the consistency of the Index with every reference found in the volumes. If the hash of the reference is not found in the Index or doesn't match the address, then the chunks at the given addresses are read, the hashes are calculated and the Index is fixed when appropriate. Incorrect entries are converted into `missing` to indicate that some chunks are missing. This option no longer uses the file `orphanaddr.bin`. This file is now deleted after a successful vacuum.

Every mismatch is logged in the `checkvolume` trace file with the coordinate of the file that holds the reference. Only one line is logged per file and per type of mismatch, others are counted in the statistics. Tools that can use this information to exclude the faulty file during a restore (for example) will come later.

The lines in the trace file look like this:

```
bacula-sd: dedupengine.c:4151 VacuumBadRef FixedIndex FI=1 SessId=1
  SessTime=1479138666 : ref(#55fd99e7 addr=0x0016000000000001 size=22254)
  idx_addr=0x0038000000000001
```

Every related line holds the keyword "VacuumBadRef" followed by one second keyword, see below for the details:

- `RefTooSmall`: The record in the volume that holds the reference is too small to hold a reference and is then unusable and not processed further.
- `BadAddr`: The address in the reference looks fanciful and is ignored. The record in the volume may be corrupted.
- `FixedIndex`: One reference has been verified and used to fix the Index. Maybe the Index had no entry for the hash of this reference or had a different address.
- `OrphanRef`: The hash related to this reference doesn't match the related chunk or the one given by the Index if any. This reference is an orphan. The file that holds this reference cannot be fully recovered.
- `RecoverableRef`: The hash related to this reference doesn't match the related chunk, but the Index has a different address that does match the chunk. Then the file can be restored using "dedup storage=XXXXX rehydra_check_hash=1" during the time of the restore. The address is written in file `orphanaddr.bin`

The other fields on the line depend on the type:

- `FI, SessId and SessTime` are the coordinates of the file as written in the Catalog.
- `fields inside ref()` are related to the reference.

At the end, Bacula displays some statistics in the trace file:

```
Vacuum: idxfix=0 2miss=0 orphan=0 recoverable=0
Vacuum: idxupd_err=0 chunk_read=0 chunk_read_err=0 chunkdb_err=0
```

- `idxfix`: The number of entries fixed in the Index. See `FixedIndex` above.
- `orphan`: The number of orphan chunks. See `OrphanRef` above.
- `recoverable`: The number of recoverable chunks. See `RecoverableRef` above.
- `idxfix_err`: The number of errors while trying to fix the entries.
- `chunk_read`: The number of chunks that have been read from disk to verify the hash.
- `chunk_read_err`: The number of errors while reading the chunks.
- `chunkdb_err`: The number of errors while updating the cache that stores the hashes of the block that have been read.

The last three counters are also updated by the "checkindex" option.

Self Healing

It is possible to enable an option to store all chunks of data to the Deduplication Engine even if the chunks are already stored.

```
dedup storage=Dedup self_healing=1
```

Container Scrubbing

The Scrub process reads every chunk in every container and compares them with the Index. If an inconsistency is found, the Index is corrected automatically.

Since the container files can be very large, the Scrub process can take days to read everything within them. Bacula jobs (backup, restore, verify, migration, copy, ...) can run while the Scrub process is running. A vacuum process automatically pauses the Scrub process for the duration of the vacuum.

It is recommended to run the Scrub on a regular basis. To minimize the impact of the Scrub process during your backup window, it is possible to control the speed or suspend and resume the process with a bconsole command. This may be done manually, or scripted as part of a cron job:

```
$ cat /etc/cron.d/bacula-scrub
BCONS=/opt/bacula/bin/bconsole
LOG=/opt/bacula/working/scrub.log

#M H  DOM M DOW USER   CMD
01 18  *  *  *  bacula echo "dedup scrub suspend storage=Dedup" | $BCONS >
↪$LOG
01  8  *  *  *  bacula echo "dedup scrub resume  storage=Dedup" | $BCONS >
↪$LOG

# a softer solution limiting then bandwidth
#01 18  *  *  *  bacula echo "dedup scrub_bwlimit=10mb/s storage=Dedup" |
↪$BCONS > $LOG
#01  8  *  *  *  bacula echo "dedup scrub_bwlimit=0 storage=Dedup" | $BCONS >
↪$LOG
```

To limit the speed of the Scrub process, you can set the `DedupScrubMaximumBandwidth` directive on the `Storage` resource in the `bacula-sd.conf` file. The default maximum bandwidth value is 50MB/s. This is the total amount that the scrub can use, all the `workers`, and this amount will not be available for backup jobs.

```
 Storage {
   Name
...
   DedupScrubMaximumBandwidth = 20MB/s
 }
```

This value may be adjusted manually with a bconsole command:

```
* dedup scrub_bwlimit=10mb/s
```

Scrubbing is more effective after a "`dedup vacuum checkmiss`". The `checkmiss` option forces the vacuum to create dummy entries in the Index for every orphan reference found in the volumes. The Scrub process will resolve these dummy entries when it finds a matching chunk. When the Scrub doesn't find any related entry in the Index, the chunk is marked as free. The `checkvolume` option of the vacuum command also creates dummy entries. See the differences in the vacuum section.

```
$ cat /opt/bacula/scripts/dedup-scrub
#!/bin/sh

SD=Dedup1
LOG=/opt/bacula/working/scrub.log
PATH=$PATH:/opt/bacula/bin

exec 1>> $LOG
exec 2>> $LOG

date
echo "dedup vacuum checkmiss storage=$SD" | bconsole
echo "dedup scrub run storage=$SD" | bconsole
date
```

Scrubbing can be done by multiple threads, each of them handling one container at a time and should be able to reach a throughput up to 400MB/s per CPU core (limited by the SHA512/256 calculation). The

Scrub saves it's state at regular intervals and can restart from where it has been interrupted. The Scrub process doesn't restart automatically after a restart or a reboot.

The Scrub starts handling the containers that are largest to efficiently balance the work between the threads.

At the end, the Scrub process does a `checkindex` to check the coherence between the Index and the FSM and to detect if an address is used twice or if an entry refers to an empty chunk.

The Scrub process can be controlled from `bconsole` via the `dedup` command:

```
*dedup
Dedup Engine choice:
     1: Vacuum data files
     2: Cancel running vacuum
     3: Display data files usage
     4: Scrub data files options
Select action to perform on Dedup Engine (1-4): 4
Dedup Engine Scrub Process choice:
     1: Run Scrub
     2: Stop Scrub
     3: Suspend Scrub
     4: Resume Scrub
     5: Status Scrub
Select Scrub action to perform on Dedup Engine (1-5):
```

It is possible to run every Scrub sub-command from the command line:

```
* dedup scrub run storage=Dedup
* dedup scrub run worker=3 storage=Dedup
* dedup scrub run reset storage=Dedup
```

The "`dedup scrub run`" command starts the Scrub process. If the Scrub has been interrupted by a crash or a restart of the daemon, the Scrub process will continue from its last saved point. Notice that the Scrub process doesn't continue automatically after a restart or a reboot. The "`worker`" option controls the number of threads, the default is one. The "`reset`" option forces the Scrub to ignore its last saved point and restart from the beginning.

Other Scrub commands available:

```
* dedup scrub wait storage=Dedup
* dedup scrub stop storage=Dedup
* dedup scrub suspend storage=Dedup
* dedup scrub resume storage=Dedup
```

- **wait**. Wait until the end of the Scrub process.
- **stop**. Stop any running threads of the Scrub process.
- **suspend**. Suspend all the threads of the Scrub process.
- **resume**. Resume all the threads of the Scrub process.

"`suspend`" and "`resume`" do not modify the options given to the "`run`" command. "`stop`" stops the threads and allows a restart of the Scrub process using the "`run`" command and a different number of threads for example.

Finally you can get the status of last Scrub that has been started.

```
* dedup scrub status storage=Dedup
Scrubber: last_run="10-Aug-2017 12:05:38" started=1 suspended=0 paused=0␣
↪quit=0
 pos=1225639597 pos=8% bw=44957018/50000000
* dedup scrub wait storage=Dedup
* dedup scrub status storage=Dedup
Scrubber: last_run="10-Aug-2017 12:05:38" started=0 suspended=0 paused=0␣
↪quit=1
 pos=14453933383 pos=100% bw=3269270/50000000
```

The "`status`" sub-command tells you if the Scrub process is running, if it has been suspended by the user or paused by the vacuum, the absolute position that is the total of all the bytes that have been read for all the containers, the relative position in percent and also the disk bandwidth in bytes/s compared with what has been allowed by the variable "`scrub_bwlimit`". The position is updated every 10 seconds.

Some useful information is logged to the trace file. The "`status`" sub-command, displays the status of the Scrub process for each container:

```
...
Scrub status [66] size=327677663 scrub_pos=-1 scrub_start=1502366453
Scrub status [67] size=327677780 scrub_pos=43670 scrub_start=1502367337
Scrub status [68] size=327679152 scrub_pos=0 scrub_start=0
...
Scrub status read_err=0 fix_err=0 feed=0
Scrub status fix miss=0 wrong=0 false_set=0 false_unset=0
```

Here, the Scrub process for container `[66]` is finished, container `[67]` is being processed and the Scrub process for container `[68]` is still pending.

The position is in bytes, and must be compared to the size of the container on the left - also in bytes. The "`scrub_start`" is the `epoch` when the Scrub started handling the container. The counters at the end of the output show the current general statistics:

- `read_err` is the number of chunks that were unreadable from the disk, or corrupted and finally ignored. As Holes are not yet skipped by the Scrub process, chunks in these areas will increment this counter.
- `fix_err` is the number of errors encountered when trying to fix an existing error
- `feed` is used internally and only relevant to our developers. It should always be 0.
- `miss` is the number of entries in the Index that were `missing` and have been resolved by the Scrub process.
- `wrong` is the number of entries in the Index that were pointing to the wrong chunk and that have been fixed by the Scrub process.
- `false_set` is the number of chunks that were incorrectly marked as used, but were not required by the Index and were then marked as free.
- `false_unset` is the number of chunks that were incorrectly marked as free, but required by the Index and were then marked as used.

When the Scrub process finishes a container, it logs the statistics for this container in the trace file:

```
ScrubContainer [106] end pos=-1 fatal=0 read_err=0 fix_err=0 feed=0
ScrubContainer [106] fix miss=0 wrong=0 false_set=0 false_unset=0
```

At the end, the Scrub process performs a cleanup identical to the cleanup done by the `vacuum` "`checkindex`" command and finally displays consolidated statistics for all of the containers.

```
cleanup_index_addr_duplicate unset2miss=1
cleanup index Phase 1 cnt=2362298 badaddr=0 suspect=0 unset=0 2miss=0 miss=0␣
↪(count=2362299 err=0 2miss_err=0)
cleanup index Phase 2 cnt=2362298 2miss=0 (count=2362299 err=0 2miss_err=0)
Scrubber index cleanup chunk_read=0 chunk_read_err=0 chunkdb_err=0
ScrubContainer END read_err=0 fix_err=0 feed=0 cleanup=OK
ScrubContainer FIX miss=0 wrong=0 false_set=0 false_unset=0
```

Hardware Requirements

CPU

Bacula's Global Endpoint Deduplication consumes CPU resources on both File Daemon and Storage Daemon. The table *below* shows operations done by both daemons depending on the deduplication mode.

Note that the Storage Daemon has to re-calculate hashes of the chunks sent by the File Daemon to ensure the validity of the data added to the Dedupengine.

Table 2: Operations done by each daemon

	Dedup=none	**Dedup=storage**	**Dedup=bothside**
Client	–	–	hash + compress
Storage	–	hash + compress + DDE	decompress + hash + DDE

On recent Intel processors, compression and hash calculations each require about 1GHz of CPU power per 100MB/sec (1Gbps). Decompression requires only 0.25GHz for the same throughput. The Dedupengine depends more on IOPs rather than on CPU power (about 0.1GHz per 100MB/sec). Both daemons must also handle network and disks (around 1GHz per 100MB/sec).

The rules of thumb might be to dedicate 3GHz per 100MB/s for the File Daemon or the Storage Daemon when doing deduplication.

Table 3: CPU requirements (Intel based)

	100MB/sec (Gbps)	**400MB/sec (4Gbps)**	**1000MB/s (10Gbps)**
Client or storage	3GHz	12GHz	30GHz

Add about 50% more GHz for latest generation of AMD processors.

Memory

The File Daemon requires additional RAM to do `bothsides` deduplication because it has to keep the chunks in memory until the Storage Daemon sends a command to send or to drop a chunk of data. The extra memory required is about 4MB per running job.

The Storage Daemon also requires about 4MB of memory for every running job. The Dedupengine also needs more of memory for efficient lookups in the index file, see section *Dedupengine*

Disks

On the File Daemon, the directive `Enable Client Rehydration = yes` can generate some extra `reads` during the restore process and increase the disk load and possibly slow down the job.

On the Storage Daemon, chunks are stored randomly in Containers, and the disk systems might have to do significantly more random I/O during backup and restore. Note that migration/copy and virtual full Jobs do not need to rehydrate data if the destination device supports deduplication. Chunks are stored in 65 or more container files in the `Dedup Directory`. All Volumes use references to those container files. This means that your system must be configured to manage disk space and extend disk space if necessary. We advise you to use LVM, ZFS, or BTRFS.

For effective performance, it is strongly recommended to store the deduplication engine Index on dedicated solid state storage, NVMe or SSDs. Please see section *Dedupengine*. It is not recommended to store deduplication engine containers on the same file systems the Catalog database resides on.

The index file used to associate SHA512/256 digests with data chunk addresses will be constantly accessed and updated for each chunk backed up. Using solid state storage for the index file will give better performance. The number of I/O operations per second that can be achieved will limit the global performance of the deduplication engine. For example, if your disk system can do 10,000 operations per second, it means that your Storage Daemon will be able to write between 5,000 and 10,000 blocks per second to your data disks. (310 MB/sec to 620 MB/sec with 64 KB block size, 5 MB/sec to 10 MB/sec with 1 KB block size). The index is shared between all concurrent Jobs.

To ensure thatfile systems required for containers, index, and volumes are mounted before the Storage Daemon starts, you can edit the `bacula-sd.service` unit file

```
# systemctl edit bacula-sd.service
```

This will create the file `/etc/systemd/system/bacula-sd.service.d/override.conf` to add `bacula-sd.service` customized settings. Please add the following line to the file and save it:

```
RequiresMountsFor=/bacula/dedup/index /bacula/dedup/containers /bacula/dedup/
↪volumes
```

Of course, paths may need to be adjusted per your actual configuration.

Installation

The recommended way to install deduplication plugin is using BIM, where the deduplication plugin installation can happen alongside the installation of Storage Daemon, at the point of choosing the plugin.

Linux

To install deduplication plugin for Linux, visit Linux: Install Storage Daemon and, in step 5, choose the dedup plugin. If you have already installed SD, run the installation again and choose the dedup plugin.

Important: While going through the installation steps again, your configuration file will not be overwritten.

Restrictions and Limitations

- You must take care to define unique Media Types for Dedup Volumes that are different from Media Types for non-Dedup Volumes.
- The "hole punching" feature is available on Linux systems with kernel 2.6.37 and later. The function was also backported by RHEL to their 2.6.32 kernel (on RHEL 6.7). The feature is not available on FreeBSD or Solaris OSes.
- Some files are not good candidates for deduplication. For example, a mail server using maildir format will have a lot of small files, and even if one email was sent to multiple users, SMTP headers will probably interfere with the deduplication process. Small files will tend to enlarge your chunk index file resulting in a poor dedup ratio. A good dedup ratio of 20 for a file of 1024 bytes will save only 19 KB of storage, so much less gain than with a file of 64 KB with a poor dedup ratio of 2.
- Dedup Volumes cannot just be copied for offsite storage. Dedup Volumes should stay where the deduplication engine is running. In order to do offsite copies, it is recommended to run a Copy Job using a second Dedup Storage Daemon for example, or to create rehydrated volumes with a Copy/Migration/Virtual Full job using a regular disk Device. The VirtualFull support was added in version 8.0.7. The Storage Daemon to Storage Daemon Copy/Migration process with deduplication protocol was added in version 8.2.0.
- A Virtual Full between two Storage Daemons is currently not supported.
- Data spooling cannot be used with deduplication. Since versions 8.2.12 and 8.4.10, data spooling is automatically disabled whenever a device resource is configured with **Device Type = Dedup**.
- All Bacula Enterprise File Daemons (including Linux, FreeBSD, Solaris, Windows, …) support the Global Endpoint Deduplication. The Community Bacula File Daemon supports only the Global Endpoint Deduplication in `dedup=storage` mode. The list of the platforms supported by Bacula Enterprise is available on www.baculasystems.com.
- We strongly recommend that you update all File Daemons that are used to write data into Dedup Volumes. It is not required, but old File Daemons do not support the newer FD to SD protocol, and consequently the Global Endpoint Deduplication will done only on the Storage daemon side.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Best Practices

RAID

If you plan to use RAID for performance and redundancy, please note that read and write performances are essential for the deduplication engine. The Index is highly accessed for reading and for writing during backup jobs run and during the maintenance tasks required by the deduplication plugin. Also, the forceoptimize process that rebuilds the Index strongly depends on the read and write performance of the disk infrastructure.

Some RAID solutions don't fit the deduplication engine read and write performance requirements. RAID 10 is recommended for both the dedup index and dedup containers as it provides both redundancy of performance better than RAID_1. Please note that many RAID solutions have better performance than RAID_5, thus RAID_5 should be avoided if possible.

ZFS

If you plan to use ZFS file system to store the dedup index, it is important to guarantee that you have enough memory and CPU resources in the Storage Daemon server to have both the deduplication plugin and ZFS in good condition.

The Global Endpoint Deduplication plugin does both deduplication and chunk compression. This means there is no need to enable deduplication or compression in the zfs pool that will be used to store containers. In fact, it is not recommended to have them enabled as it may cause slow performance and there will be no gain in space savings.

Aligned disk acess is a key factor when using ZFS. ZFS is able to detect the sector size of the drive, but disks can report the emulated value instead. As we recommend solid state storage for the dedup Index, performance can be improved if the `ashift` property is explicitly set to match the sector size of the underlying storage, which will often be 4096_bytes.

The disk block size is defined by the `ashift` value, but as ZFS stores data in records, there is another parameter that determines the individual dataset to be written in the ZFS pool, this is the `recordsize` ZFS pool parameter.

Thus another important ZFS pool setting to consider is the `recordsize` value. It defaults to 128K in recent ZFS versions. This value should be adjusted to match the typical I/O operations. For the ZFS pool used by the dedup Index, it was reported that setting the `recordsize` to 8K increased the vacuum performance. In addition, setting the ZFS kernel `zfs_vdev_aggregation_limit_non_rotating` parameter to the same value as `recordsize` highly improved performance.

Please note these values are recommended for most SSD disks, but they may vary depending on the SSD model. For example, 16K could fit some SSD models and give a better performance. We recommend to perform I/O benchmark using different settings before the deduplication engine is setup in production.

Regarding the use of ZFS to store dedup containers, as it is not possible to preview the typical dataset because some containers can be much more used than others, it is more probable that a `recordsize` value of 64K or even the 128K default value are sufficient to have a good performance. However, it is important to not allow container files to grow too much and limit the size of containers files to 3TB or 4TB.

Maximum Container Size

For better performance, it is strongly recommended to not allow container files to grow indefinitely even if the underlying file system supports very big files. This can be accomplished by setting the "Maximum Container Size" directive in the Storage resource in the Storage Daemon configuration file. It is recommended to set this directive to a value between 1 TB and 4 TB.

This setting is also recommended because container files cannot be shrunk. Once they grow, it is not possible to have these files reduced in size. The holepunching process will not reduce the container file sizes by shrinking them.

Vacuum and Scrub

It is strongly recommended to keep the deduplication engine healthy by regularly performing the maintenance tasks triggered by the vacuum and scrub processes.

Please make sure to regularly run, in all deduplication engines in your environment, the following tasks: daily pruning and truncation of volumes, a simple vacuum daily (it can be run weekly when not too many new chunks are added to the deduplication engine), a dedup vacuum checkindex checkmisss monthly and the scrub process preferably outside of the backup time windows.

These maintenance tasks can be scheduled in an Admin Job and they will help to clean both the deduplication engine index and containers, marking unused entries in the index and chunks in containers, thus allowing capacity reuse. They will also contribute to avoid invalid index entries to be used by backup jobs in case of any problems with the server hosting the deduplication engine.

Below are examples of two Admin Jobs that can be used to run a weekly and monthly vacuum.

```
Job {
  Name = "DedupSimpleVacuum_ADMTASK"
  Type = "Admin"
  Client = "bacula-fd"
  Fileset = "LinuxHome"
  Messages = "Standard"
  Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
    Console = "dedup vacuum storage=MyDedupStorage"
    FailJobOnError = no
    RunsOnClient = no
    RunsWhen = Before
  }
  Schedule = "Vaccum_daily_10AM"
  Storage = "MyDedupStorage"
}

Schedule {
  Name = "Vaccum_daily_10AM"
  Enabled = yes
  Run = at 10:00
}

Job {
  Name = "DedupDeepVacuum_ADMTASK"
```

(continues on next page)

(continued from previous page)

```
  Type = "Admin"
  Client = "bacula-fd"
  Fileset = "LinuxHome"
  Messages = "Standard"
  Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
    Console = "dedup vacuum checkindex checkmiss storage=MyDedupStorage"
    FailJobOnError = no
    RunsOnClient = no
    RunsWhen = Before
  }
  Schedule = "Vaccum_monthly_secondSunday_11AM"
  Storage = "MyDedupStorage"
}

Schedule {
  Name = "Vaccum_monthly_secondSunday_11AM"
  Enabled = yes
  Run = 2nd Sun at 11:00
}
```

In an Admin Job, the Fileset and the Pool configured are not used. Thus, you can setup any valid value available in your Bacula Enterprise environment configuration.

Holepunching

The holepunching process allows you to mark unused chunks in container files as free after a successful vacuum is run. This process will punch a hole in the file and it is required that the underlying file system support holes. We do recommend to use file systems that support hole punching, such as xfs and ext4.

It is important to note that the released amount of space will depend on the I/O block size of the underlying file system. This means that, if you have a file system configured with block size of 4 MB, only entire blocks of 4 MB will be released to the system to be used by either container files or other files in the file system.

Global Endpoint Deduplication 2

Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. One of those challenges is the growing amount of data to be backed up, together with limited time to run backup jobs (backup window). Bacula Enterprise offers several ways to tackle these challenges, one of them being *Global Endpoint Deduplication 2*, which minimizes the network transfer and Bacula Volume size using deduplication technology.

This document is intended to provide insight into the considerations and processes required to successfully implement this innovative backup technique.

Deduplication

Deduplication is a complex subject. Generally speaking, it detects that data being backed up (usually chunks) has already been stored and rather than making an additional backup copy of the same data, the deduplication software keeps a pointer referencing the previously stored data (chunk). Detecting that a chunk has already been stored is done by computing a hash code (also known as signature or fingerprint) of the chunk, and comparing the hash code with those of chunks already stored.

The picture becomes much more complicated when one considers where the deduplication is done. It can either be done on the server and/or on the client machines. In addition, most deduplication is done on a block by block basis, with some deduplication systems permitting variable length blocks and/or blocks that start at arbitrary boundaries (sliding blocks), rather than on specific alignments.

Advantages of Deduplication

- Deduplication can significantly reduce the disk space needed to store your data. In good cases, it may reduce disk space needed by half, and in the best cases, it may reduce disk space needed by a factor of 10 or 20.
- Deduplication is combined with compression to further reduce the storage space needed. Compression depends on data type and deduplication depends on the data usage (on the need or the will of the user to keep multiple copies or versions of the same or similar data). Bacula takes advantage that both techniques work perfectly together and combines them in it's original Dedup Engine.
- Deduplication can significantly reduce the network bandwidth required because both ends can exchange references instead of the actual data itself. It works when the destination already has a copy of the original chunks.
- Handling references instead of the data can speed up most of the processing inside the Storage Daemon. For example, Bacula features like copy/migrate and Virtual Full can be up to 1,000 times faster. See the following article for more information on this subject.

Cautions About Using Deduplication

Here are a few of the things that you should be aware of before using deduplication techniques.

- To do efficient and fast deduplication, the Storage Daemon will need additional CPU power (to compute hash codes and do compression), as well as additional RAM (for fast hash code lookups). Bacula Systems can help you to calculate memory needs.
- For effective performance, the deduplication Index should be stored on SSDs as the index will have many random accesses and many updates.

- It is highly recommended that you use the dedup index with storage that handles random read/writes quickly, such as NVMe or SSD. Ceph, NFS, and spinning disks are not recommended.
- Due the extra complexity of deduplication, performance tuning is more complicated.
- We recommend Index and Containers are stored in xfs or ext4 file systems. But we are also familiar with the zfs file system.
- Deduplication collisions can cause data corruption. This is more likely to happen if the deduplicating system uses a weak hash code such as MD5 or Fletcher. The problem of hash code collisions is mitigated in Bacula by using a strong hash code (SHA512/256).
- Deduplication is not implemented on tape devices. It works only with disk-based backups.
- Deduplication 2 is compatible with cloud based infrastructure taking into consideration the virtual hardware specification required.
- The immutable flag is not compatible or does not apply to the dedup index or dedup containers.

Global Endpoint Deduplication

Bacula Systems' latest step in deduplication technology is to offer the *Global Endpoint Deduplication* feature. With Global Endpoint Deduplication, Bacula will analyze data at the block level, then Bacula will store only new chunks in the deduplication engine, and use references in standard Bacula volumes to chunks stored in the deduplication engine. The deduplication can take place at the File Daemon side (saving network and storage resources), and/or at the Storage Daemon side (saving storage resources).

The remainder of this white paper will discuss only Global Endpoint Deduplication.

How Bacula Global Endpoint Deduplication Works

- First, be aware that you need the Dedup bacula-sd-dedup-driver-x.y.z.so Storage Daemon plugin for Global Endpoint Deduplication to work. This plugin holds both *Legacy* and *Dedup2* drivers.
- Dedup2 includes new directives that allows to have both Legacy and Dedup2 engine types in the same Storage Daemon. The goal of these new directives is to support multiple Dedup Engine in the same Storage Daemon. This will allow to handle smooth migration between two Dedup of different generations. To accomplish that, you must modify your current dedup engine to use the new directives and setting "Driver=Legacy".
- Dedup devices are enabled by specifying the `Dedup` keyword as a DeviceType directive in each disk Device resource in the bacula-sd.conf where you want to use deduplicated Volumes. If you have multiple Dedud Engine you must also specify the name that you have defined in the Dedup Engine.

```
Device {
  Name = DiskDedup_Dev0
  ...
  Device Type = Dedup
  Dedupengine = Dedupengine_name_with_dedup2
}
```

- You must pay particular attention to define a unique Media Type for devices that are Dedup as well as for each Virtual Autochanger that uses a different Archive Device directory. If you use the same Media Type for a Dedup device type as for a normal disk Volume, you run the risk that you will have data corruption on disk Volumes that are used on Dedup and non-Dedup devices.

- When Global Endpoint Deduplication is enabled, the Device will fill in disk volumes with chunk references instead of the chunks. Bacula encrypted data, and very small files will be stored in the Volumes as usual. The deduplicated chunks are stored in the "Containers" of the Dedupengine, and are shared by all other dedup-aware devices (related to the same Dedup LegacyDedupEngine) in the same Storage Daemon.
- We advise to set a limit on the number of Jobs or the usage duration when working with dedup Volumes. In case you prefer to use `Maximum Volume Bytes`, consider that two Catalog fields are considered when computing the volume size. `VolBytes` represents the volume size on disk and `VolaBytes` considers the amount of non-dedup data stored in the volumes, i.e., the rehydrated data. If the directive `Maximum Volume Bytes` is used for a dedup Volume, Bacula will consider both VolBytes and VolaBytes values to check the limits.

Global Endpoint Deduplication During Backup Jobs

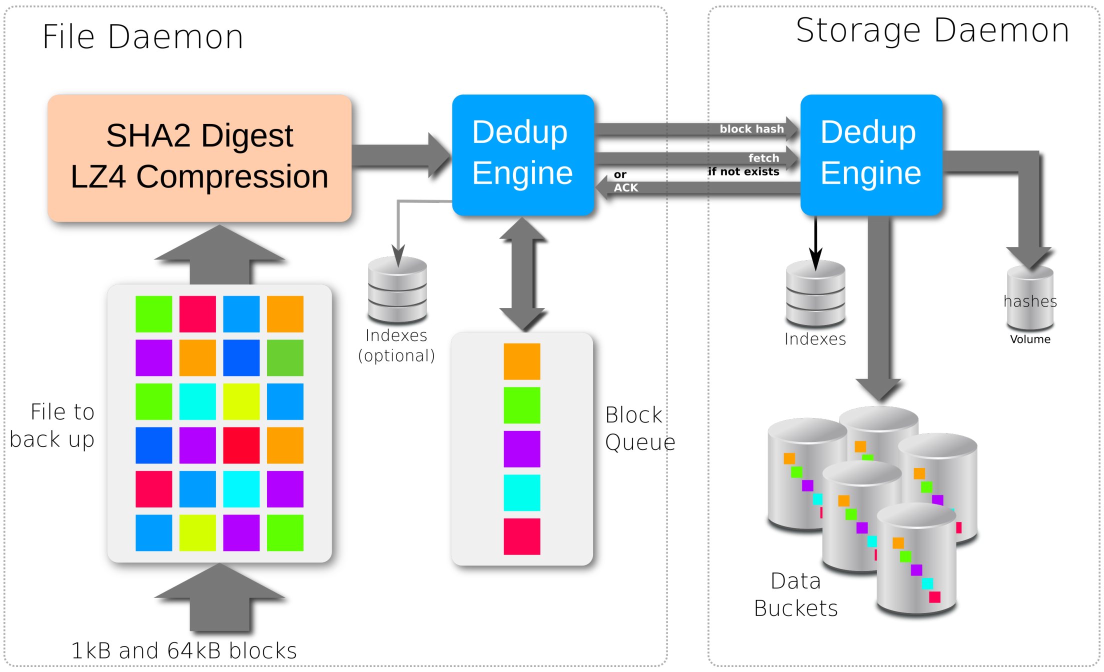


Fig. 3: Backup Scenario with bothsides deduplication

- When starting a Backup Job, the Storage Daemon will inform the File Daemon that the Device used for the Job can accept dedup data.
- If the Fileset uses the `dedup = bothsides` option, the File Daemon will compute a strong hash code for each chunk and send *references* including these hashes to the Storage Daemon which will request the original chunks from the File Daemon for any of them that the Dedupengine is unable to resolve.
- If the Fileset uses the `dedup = storage` option, the File Daemon will send data as usual to the Storage Daemon, and the Storage Daemon will compute hash codes and store chunks in the Dedupengine and the references in the disk volume.
- If the Fileset uses the `dedup = none` option, the File Daemon will send data as usual to the Storage Daemon, and the Storage Daemon will store the chunks in the Volume without performing any deduplication functions.

- If the File Daemon doesn't support Global Endpoint Deduplication, the deduplication will be done on the Storage side if the Device is configured with `DeviceType = dedup`.

New Storage Daemon Directives

Remember to specify the directive Plugin Directory.

- Plugin Directory = `<directory-path>`
  This directive tells the Storage Daemon where to find plugins. The file bacula-sd-dedup-driver-x.y.z.so must be present in this directory before starting the Storage Daemon.

New Dedupengine Resource

As said earlier, it is possible to have both the legacy and the new *Dedup2* engines in the same Storage Daemon. A new format has been introduced to allow it. The new resource *Dedupengine* will allow to have your current legacy dedup engine and a new *Dedup2* engine in the same Storage Daemon. All Dedup Devices that refer to the same *DedupEngine* shares the same storage space.

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup/containers
  Dedup Index Directory = /mnt/SSD/dedup/index
  Driver = Dedup2
  MaximumContainerSize = 2TB
}
```

- Name = `<name>`
  The name of the DedupEngine that will be used in the Device resource. The Dedup name is required.
- Dedup Directory = `<directory-path>`
  Deduped chunks will be stored in the Dedup Directory. This directory is shared by all Dedup devices configured on a Storage Daemon and should have a large amount of free space. We advise you to use LVM on Linux Systems to ensure that you can extend the space in this directory. The Dedup Dedup Directory directive is mandatory. We recommend that you do not change this directory afterward, because if you make a mistake, it would invalidate all of your backups. If you do change the `Dedup Directory` directive, the following files must be moved to the new directory:
  - `beed2-XXXXXXXX.ctn`
  - `beed2.esm0`
  - `beed2.esm1`

  The .esm0 and .esm1 files hold the state of every extent in each container.
- Dedup Index Directory = `<directory-path>`
  Indexes will be stored in the Dedup Index Directory. Indexes will have a lot of random update accesses, and will benefit from fast drives such as SSD drives. By default, the Dedup Index Directory is set to the Dedup Directory.

  As with the Dedup Directory, we recommend against changing the `Dedup Index Directory` directive. If you do, the following files and directories must be moved to the new directory:

  - `beeindex.tch`
  - `beeindex.tch.new`

  The file `beeindex.tch.new` is a temporary file created by the `optimize` part of the vacuum that remain when the process is interrupted. This file holds a new version of the index. At startup the DedupEngine rename this file into *beeindex.tch* and use this new file. If this *beeindex.tch.new* file exists you can just copy it as is and ignore *beeindex.tch*

- Maximum Container Size = <`size`>

  No container will be allowed to grow to more than <size> bytes. When this size is reached, a new container file will be created. The default value is 100GiB. This value is used at initialization time and cannot be modified later, any change would be ignored. This limit is useful when you store your containers on a filesystem that limits the size of the file to a pretty low value. For now, the number of containers is limited to 8192 but we recommend to keep the number below 1024. This directive should be used to limit the size of container files to any limit from the file system where containers are stored. Dedup2 allows up to 100 million container files.

- Maximum Container Open = <`count`>

  This is the maximum number of containers that can be simultaneously open. When *Maximum Container Size* has been setup to a small size, this directive force Bacula to close unused containers before to open a new one and avoid to have 1000th of simultaneously open files. The default value is 100 GB. The recommended value is 20 + the number of simultaneous backup + the number of simultaneous restore.

- Driver = <`driver`>

  You have the choice between the old Dedup **Legacy** or the new one **Dedup2**.

- MaximumIndexMemorySize = <`size`>

  This is the maximum memory the DedupEngine will allocate for the `Index`. If unspecified, Bacula uses the available RAM as a starting point, saving enough RAM for the system. When the Index is created the first time, the half of this value or the half of the RAM is used to size the hash table. The initial hash table should not use more than 8GiB, this is about 1 billion entries.

To modify your current legacy Dedup engine to the new configuration format:

- Stop the Storage Daemon
- Modify the current dedup engine settings by creating a DedupEngine resource with Dedup=Legacy and the same index and containers directories of your current dedup engine:

```
Dedupengine {
  Name = Dedupengine_legacy
  Dedup Directory = /mnt/bacula/dedup/containers
  Dedup Index Directory = /mnt/SSD/dedup/index
  Driver = Legacy
  MaximumContainerSize = 2TB
}
```

- Remove the index and containers directories from the Storage resource configuration
- Start the Storage Daemon

New Device Directives

```
Device {
  Name = FileDedup1
  ...
  Device Type = Dedup
  Dedupengine = Dedupengine_Name_1
}
```

- Device Type = Dedup
  This directive is required to make the Device write Dedup volumes. Once turned on, Bacula will use references in Volumes and will store data chunks into the specified DedupEngine.

  Once a Device has been defined with a certain Type (such as Dedup, Aligned, File or Tape), it cannot be changed to another Type. If you do so, the Bacula Storage Daemon will not be able to properly mount volumes that were created before the change.

- Dedupengine = <dedupengine-name>
  This directive defines the DedupEngine where this device will store its chunks.

```
# From bacula-sd.conf
Storage {
 Name = my-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Subsys Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
}

Dedupengine {
  Name = Dedupengine_Name_1
  Dedup Directory = /mnt/bacula/dedup/containers
  Dedup Index Directory = /mnt/SSD/dedup/index  # Recommended to be on fast␣
↪local SSD storage
  Driver = Dedup2
  Maximum Container Size = 2TB # Try to get a maximum of 1000 containers at␣
↪most
}

Device {
  Name = "DedupDisk"
  Archive Device = /mnt/bacula/dedup/volumes
  Media Type = DedupVolume
  Device Type = Dedup                   # Required
  dedupengine = Dedupengine_Name_1
  LabelMedia = yes
  Random Access = Yes
  AutomaticMount = yes
  RemovableMedia = no
  AlwaysOpen = no
}
```

Director Daemon Fileset Directive

Within the Director, the Global Endpoint Deduplication system is enabled with a Fileset Option directive called Dedup. Each Include section can have a different behavior depending on your needs.

```
# Use the default dedup option of 'storage' side deduplication
 Fileset {
   Name = FS_BASE
   Include {
     Options {
       Dedup = storage
     }
     File = /opt/bacula/etc
   }

   # Do not dedup my encrypted data
   Include {
     Options {
       Dedup = none
     }
     File = /encrypted
   }

   # Minimize the network transfer by using 'bothsides' dedup option
   Include {
     Options {
       Dedup = bothsides
     }
     File = /bigdirectory
   }

 }
```

The Dedup Fileset option can have the following values:

- **Dedup = storage** - All the deduplication work is done on the Storage Daemon side if the device type is Dedup. The File Daemon will send all data to the SD just as it normally would. (Default value)
- **Dedup = none** - Disable deduplication on both the File Daemon and Storage Daemon.
- **Dedup = bothsides** - The deduplication work is done on the File Daemon and the Storage Daemon. This reduce the network traffic between both Daemon as soon as the Storage Daemon already have the same data.

About Fileset Compression

The data stored by the Global Endpoint Deduplication Engine is automatically compressed using the LZ4 algorithm. Using the Fileset `Compression = LZO|GZIP` option might reduce the deduplication efficiency, and compressing the data twice consumes extra CPU cycles on the client side. Thus we advise that you do not use client-side GZIP or LZO compression when using a Dedup Device. To prevent such an inefficient configuration, we recommend setting the `Allow Compression` directive in a Director Storage resource "No":

```
# cat bacula-dir.conf
...
Storage {
   Name = Dedup
   Allow Compression = No      # Disable Fileset Compression
                               # option automatically
   Address = baculasd.lan
   Password = xxx
   Media Type = DedupMedia
   ...

}
```

Things to Know About Bacula

- You must pay particular attention to define a unique Media Type for devices that are Dedup as well as for each Virtual Autochanger that uses a different Archive Device directory. If you use the same Media Type for a Dedup device type as for a normal disk Volume, you run the risk that you will have data corruption on disk Volumes that are used on Dedup and non-Dedup devices.
- Dedup devices are compatible with Bacula's Virtual Disk Changers
- We strongly recommend that you not use the Bacula `disk-changer` script, because it was written only for testing purposes. Instead of using `disk-changer`, we recommend using the Virtual Disk Changer feature of Bacula, for which there is a specific white paper.
- We strongly recommend that you update all File Daemons that are used to write data into Dedup Volumes. It is not required, but old File Daemons do not support the newer FD to SD protocol, and consequently the Global Endpoint Deduplication cannot not be done on the FD side.
- The immutable flag is compatible with dedup volumes, see more details in Volume Protection Enhancements and Volume Protection.

Deduplication Engine Vacuum

Over time, you will normally delete files from your system, and in doing so, it may happen that there will be chunks that are stored in dedup containers that are no longer referenced.

In order to reclaim these unused chunks in containers, the administrator needs to schedule a `vacuum` option of the `dedup` command. The `vacuum` option will analyze dedup volumes and mark as free any chunks that are not referenced, thus allowing the disk space to be reused. The vacuum command can run while other jobs are running. During the index optimization backups are temporary suspended and resume just after the optimization.

```
* dedup
Dedup Engine choice:
     1: Vacuum data files
     2: Cancel running vacuum
     3: Display data files usage
Select action to perform on Dedup Engine (1-3): 1
The defined Storage resources are:
     1: File1
     2: Dedup
Select Storage resource (1-2): 2
Connecting to Storage daemon File at localhost:9103 ...
dedupvacuum start inventory
Found 1 volumes (4.547 MB) to scan for MediaType=File
Ready to read from volume "TestVolume001" on Dedup device "FileStorage" (/
↪home/bac/workspace2/bee/regress/tmp).
End of Volume "TestVolume001" at addr=4547374 on device "FileStorage" (/home/
↪bac/workspace2/bee/regress/tmp).
Found 0 volumes (0 B) to scan for MediaType=File1
dedupvacuum start index cleanup and container defrag
Vacuum OK.
```

Deduplication Engine Status

Is it possible to query the Deduplication Engine to get some information and statistics. Here is an example output of the dedup usage command, followed by an explanation of each section in the output:

```
Dedupengine status: name="dedup2-Dedupengine" now="11-Nov-2022 10:07:26"
Index: usage=0% hash_count=928859 index_capacity=1006632947
Storage: usage=5% used=62.80GB free=1.002TB unused=54.30GB
References: count=1881851 size=123.3GB count_ratio=2.03 size_ratio=1.96
Last Vacuum: start="04-Nov-2022 20:16:20" end="04-Nov-2022 20:18:48"
 total=0:02:28 inventory=0:00:06 index_cleanup=0:00:12 defrag=0:00:48
 volume=5 ref=479198 ref_size=31.40G
 suspect=0 orphan=0 2miss=0 miss=0 idx_err=0
```

Dedupengine: General information about the *Dedupengine*

- name This is the name of the Dedupengine
- now This is when the status has been generated in local time

Index: Information about the *Index*

- usage This is the usage of the index in %. A value above 100% means that the hash table is overloaded and that a vacuum should be run to optimize the index and increase the size of the hash table.
- hash_count This is the number of entries in the index. This is the number of chunk that are indexed.
- index_capacity This is the size of the hash table of the index

Storage: Information about the storage and the containers

- usage This is usage in % of space used by the *containers* of the Dedupengine. This is the ratio of the used space vs the total amount of space that ould be used.

- `used` This is the amount of space used inside the containers.
- `free` This is the amount of space that is available for new data. This include the space unused inside the container and the free space in the filesystem.
- `unused` This is amount of space that is unused inside the *containers*. This space will be used first by the Dedupengine before to grow the last *container* or create new ones.

References: Information about the references inside the *Volume* from the last *Vacuum*

- `count` This is the number of references inside all the volumes.
- `size` This is the total amount of data that the volumes refer to.
- `count_ratio` This is ratio between the number of reference and the number of chunks (*hash_count*)
- `size_ratio` This is ratio between the size without deduplication and the size with deduplication. The difference with the `count_ratio` give and idea of the efficiency of the LZ4 compression.

Last Vacuum: These information have been generated during the last *Vacuum*

- `start` This is the time of the start of the last *vacuum*.
- `end` This is when the last *vacuum* ended.
- `total` This is the total amount of time used by the vacuum in H:m:s
- `inventory` This is the time used read the volumes and make the *inventory* of the chunk that must be kept.
- `index_cleanup` This is the time to cleanup and optimize the index.
- `defrag` This is the time to reorganize the data inside the containers and release unused extends.
- `volume` This is the number of volume that have been read to create the inventory.
- `ref` This is the number of references found in the volumes.
- `ref_size` This is the total of all rehydrated references in all the volumes. This is the size that would be needed if deduplication was not in use.
- `suspect` This is the number of references that are missformed. This can come from an unexpected shutdown or a volume corruption. When the *checkmiss* option is used this also count the *orphan* references.
- `orphan` When the *checkmiss* option is used, this is the number of orphan references. Orphan references are reference that don't have a matching entry in the *index* and a matching chunk in the *containers*.
- `2miss` This is the number of new *missing* entries created in the index by the the last vacuum when the *checkmiss* option is used.
- `miss` This is the number of *missing* entries in the index. This is entries that are known to be needed but that don't have a matching chunk in the container. One missing entry can have multiple related *orphan* references.
- `idx_err` This is number of errors when creating *missing* entries in the index. If the this counter is greater than zero this can confirm that the old index was suffering of some problem. The value should goes to zero at the next use of the *checkmiss* option

Dedup command with the qcontainer option can be used to query used and unused space in the containter files.

To see more details on the containers usage, you can use the dedup command:

```
*dedup qcontainer storage=dedup2-autochanger
Connecting to Storage daemon bacula-sd at bacula-sd:9103...
...
dedup device=dedup2-Chgr drive=-1 cmd=qcontainer
container[00]  [0-476836]  00% free=476827 used=8 unavailable=1 other=0 file=/
↪mnt/dedup2/containers/beed2-00000000.ctn
container[00] 0..............................................
```

Disaster Recovery

Catalog

The Catalog doesn't contain any reference to the deduplication engine. However, the dedup volumes' records in the Catalog are used during the vacuum process. For this reason, you must make sure to have the Catalog properly restored before starting a dedup vacuum process.

Volumes

If a dedup Volume is in the Catalog but not on disk, a dedup vacuum process will stop and report an error.

Index

The Index is essential for the deduplication engine. In the case of a disaster, contact Bacula Systems Support Team.

Extent Space Map (ESM)

The DedupEngine use two ESM file. One is active the other is ready to receive a new version of the ESM. The active version can be guessed with a field that works like the Rock, Paper Scissor logic. There is no reconstruction procedure for ESM yet.

Containers

Containers are cut into extents that hold the chunks of data. When a container (or part of a container) file is lost, the data is lost and it is not recoverable by Bacula. There is no specific tool able to identify and update the index with the data that are missing.

Dedupengine

The deduplication engine is the heart of Bacula's Global Endpoint Deduplication. It has to store, to index and to retrieve the chunks. All chunks are stored in the `Container Area` and registered in the `Index`. The `Index` is the bottleneck of the deduplication process because all operations need to access it randomly, and very quickly. Memory caching and storing the Index on SSD drives will help to maintain good performance.

The Deduplication Index maintains all the hashes of all chunks stored in the Dedup Containers. To get effective performance very fast low latency storage is critical. For large back up data it is best to have the Containers and Deduplication Index on the same hardware server with the Deduplication Index on solid-state drives (SSDs). The faster the disk performance, the faster and more efficient the deduplication process and the data protection will be. In production environments it is best to avoid configurations which introduce latency and delays in the storage infrastructure for the Deduplication Index. It is therefore best to avoid spinning disks, remote access configurations like NFS or CIFS and virtualized SDs. These can be acceptable for a small amount of data (1-2TB) or to perform tests but will normally not provide acceptable performance in larger production environments.

Sizing the Index

The size of the index depends on the number of chunks that you want to store in the deduplication engine. An upper limit would be 1 chunk per file plus 1 chunk per 64K block of data.

$$\text{number\_of\_chunks} = \text{number\_of\_files} + \frac{\text{data\_amount}}{\text{64K}}$$

If all you have is the storage capacity of your Storage Daemon and want to maximize it, you must know the average compressed size of the chunks you expect to store in Containers. If you don't know the size, you may use 16K.

$$\text{number\_of\_chunks} = \frac{\text{storage\_capacity}}{\text{16K}}$$

When you know the number of chunks, you can calculate the size of your index.

$$\text{index\_size} = 1.3 * \text{number\_of\_chunk} * (8 + 70)$$

The index can be split into two parts: the hash table and the records.

$$\text{index\_size} = \text{table\_size} + \text{record\_size}$$

$$\text{table\_size} = 1.3 * \text{number\_of\_chunk} * 8$$

$$\text{record\_size} = 1.3 * \text{number\_of\_chunk} * 70$$

The hash table part is small and is accessed for all operations. The record part is bigger.

For good performance, Bacula try to lock the hash table into memory. The OS will keep in memory a part or the whole part of the records in memory depending the the amount of RAM and the size of the Index.

But these are not the only requirements. Bacula needs some extra space on disk and in memory to optimize and resize the Index. We recommend the following:

- Be sure to have 3 times the index_size on an SSD drive for the Index.
- Try to have index_size+table_size of RAM for the Index.
- At least be sure to have 3 times the "table_size" of RAM for the Index.

Punching holes in containers

New Dedup2 does not need and does not provide any kind of `hole punching`. The `vacuum` moves the `chunks` inside the containers to optimize the space. This also improve the backup speed and the restore speed. The vacuum `holepunching` option is then ignored.

Detect, Report and Repair Dedupengine Inconsistencies

The `dedup vacuum` command should be used to detect, report and possibly repair inconsistencies in the DDE.

When used with the `checkmiss` option, the vacuum process will additionally detect and report references used by backup jobs that are not present in the dedup index.

The `checkindex` option is not used in dedup2.

The vacuum process will log information to the trace file.

The rebuild of the dedup2 index is only necessary when requested by Bacula Systems support. Rebuilding the dedup2 index will establish a baseline verify directory, allowing for the identification of any inconsistencies between the data and the index that may not be detected through other methods. Once the baseline is set, it will facilitate recovery from a crash or unexpected shutdown of the dedup2 SD in the future.

checkmiss and checkvolume options of the vacuum command

In Dedup2 there is no more difference between options `checkvolume` and `checkmiss`.

These options search the Index for every reference found in the volumes. This can significantly increase the time of the vacuum if the Index doesn't fit into memory.

The option `checkmiss` simply creates dummy entries when a reference in not found in the Index. This entry indicates that the chunk is missing and could be resolved by future backups.

Every mismatch is logged in the trace file with the coordinate of the file that holds the reference. Only one line is logged per file and per type of mismatch, others are counted in the statistics.

The lines in the trace file look like this:

```
bacula-sd: dedupengine2.cc:872 VacuumBadRef OrphanRef FI=1729 SessId=1
  SessTime=1654854105 : ref(#1f41101f addr=0x0000000000000003 size=10769)
  idx_addr=0x0000000000000002
```

Every related line holds the keyword "VacuumBadRef" followed by one second keyword, see below for the details:

- `RefTooSmall`: The record in the volume that holds the reference is too small to hold a reference and is then unusable and not processed further.
- `RefWrongSize`: The record in the volume that holds the reference has an unexpected size and is then unusable and not processed further.
- `OrphanRef`: The hash related to this reference was not found in the index or is know by the Index to be missing. This reference is an orphan one. The file that holds this reference cannot be fully recovered.

The other fields on the line depend on the type:

- `FI, SessId and SessTime` are the coordinates of the file as written in the Catalog.
- `fields inside ref()` are related to the reference.
- `idx_addr` this is the address in the index. This should always be 0x2

At the end, Bacula displays some statistics in the trace file. these values can be see via the *dedup usage* command and are documented there.

```
Vacuum: volumes=1 ref_count=1975 ref_size=97382548 suspect_ref=1
Vacuum: 2miss=0 orphan=1 idxupd_err=0
```

Self-Healing

It is possible to enable the `self_healing` feature on the dedup Storage to store all chunks of data to the Deduplication Engine even if the chunks are already stored.

```
dedup storage=Dedup self_healing=1
```

This feature is helpful when a restore/migration/copy is failing due to a corrupted chunk. Then, enabling the `self_healing` feature, and re-running the same backup job using Full level can potentially recover the required corrupted chunk.

Hardware Requirements

CPU

Bacula's Global Endpoint Deduplication consumes CPU resources on both File Daemon and Storage Daemon. The table *1* shows operations done by both daemons depending on the deduplication mode.

Note that the Storage Daemon has to re-calculate hashes of the chunks sent by the File Daemon to ensure the validity of the data added to the Dedupengine.

Table 4: Operations done by each daemon

	Dedup=none	Dedup=storage	Dedup=bothside
Client	N/A	N/A	hash + compress
Storage	N/A	hash + compress + DDE	decompress + hash + DDE

On recent Intel processors, compression and hash calculations each require about 1GHz of CPU power per 100MB/sec (1Gbps). Decompression requires only 0.25GHz for the same throughput. The Dedupengine depends more on IOPs rather than on CPU power (about 0.1GHz per 100MB/sec). Both daemons must also handle network and disks (around 1GHz per 100MB/sec).

The rules of thumb might be to dedicate 3GHz per 100MB/s for the File Daemon or the Storage Daemon when doing deduplication.

Table 5: CPU requirements (Intel & AMD Xen based)

	100MB/sec (Gbps)	400MB/sec (4Gbps)	1000MB/s (10Gbps)
Client or storage	3GHz	4 cores at 3GHz	10 cores at 3GHz

Memory

The File Daemon requires additional RAM to do bothsides deduplication because it has to keep the chunks in memory until the Storage Daemon sends a command to send or to drop a chunk of data. The extra memory required is about 4MB per running job.

The Storage Daemon also requires about 4MB of memory for every running job. The Dedupengine also needs more of memory for efficient lookups in the index file, see *this section*.

Disks

On the Storage Daemon, chunks are stored together into *Extent* that are *randomly* stored into Containers free space. The disk systems might have to do a mix of sequential and random I/O during backup and restore. Note that migration/copy and virtual full Jobs do not need to rehydrate data if the destination device is connected to the same dedupengine.

Container files are stored in the `Dedup Directory`. All Volumes use references to those container files.

For effective performance, it is recommended to store the deduplication engine Index on dedicated SSDs, see section *#section:dedupengine*. It is not recommended to store deduplication engine containers with the Catalog.

The index file used to associate SHA512/256 digests with data chunk addresses will be constantly accessed and updated for each chunk backed up. Using SSD disks to store the index file will give better performance. The number of I/O operations per second that SSD devices can achieve will drive the global performance of the deduplication engine. For example, if your disk system can do 10,000 operations per second, it means that your Storage Daemon will be able to write between 5,000 and 10,000 blocks per second to your data disks. (310 MB/sec to 620 MB/sec with 64 KB block size, 5 MB/sec to 10 MB/sec with 1 KB block size). The index is shared between all concurrent Jobs.

To ensure that a file system required for container, disk, volumes is mounted before the Storage Daemon starts, you can edit the `bacula-sd.service` unit file

```
# systemctl edit bacula-sd.service
```

This will create the file `/etc/systemd/system/bacula-sd.service.d/override.conf` to add `bacula-sd.service` customized settings. Add the following line line to the file and save it:

```
RequiresMountsFor=/bacula/dedup/index /bacula/dedup/containers /bacula/dedup/
↪volumes
```

Installation

The recommended way to install deduplication plugin is using BIM, where the deduplication plugin installation can happen alongside the installation of Storage Daemon, at the point of choosing the plugin.

Linux

To install deduplication plugin for Linux, visit Linux: Install Storage Daemon and, in step 5, choose the dedup plugin. If you have already installed SD, run the installation again and choose the dedup plugin.

Important: While going through the installation steps again, your configuration file will not be overwritten.

Restrictions and Limitations

- You must take care to define unique Media Types for Dedup Volumes that are different from Media Types for non-Dedup Volumes.
- Some files are not good candidates for deduplication. For example, a mail server using maildir format will have a lot of small files, and even if one email was sent to multiple users, SMTP headers will probably interfere with the deduplication process. Small files will tend to enlarge your chunk index file resulting in a poor dedup ratio. A good dedup ratio of 20 for a file of 1024 bytes will save only 19 KB of storage, so much less gain than with a file of 64 KB with a poor dedup ratio of 2.
- Dedup Volumes cannot just be copied for offsite storage. Dedup Volumes should stay where the deduplication engine is running. In order to do offsite copies, it is recommended to run a Copy Job using a second Dedup Storage Daemon for example, or to create rehydrated volumes with a Copy/Migration/Virtual Full job using a regular disk Device. The VirtualFull support was added in version 8.0.7. The Storage Daemon to Storage Daemon Copy/Migration process with deduplication protocol was added in version 8.2.0.
- A Virtual Full between two Storage Daemons is currently not supported.
- Data spooling cannot be used with deduplication. Since versions 8.2.12 and 8.4.10, data spooling is automatically disabled whenever a device resource is configured with Type = Dedup.
- All Bacula Enterprise File Daemons (including Linux, FreeBSD, Solaris, Windows, ...) support the Global Endpoint Deduplication. The Community Bacula File Daemon supports only the Global Endpoint Deduplication in `dedup=storage` mode. The list of the platforms supported by Bacula Enterprise is available on www.baculasystems.com.
- We strongly recommend that you update all File Daemons that are used to write data into Dedup Volumes. It is not required, but old File Daemons do not support the newer FD to SD protocol, and consequently the Global Endpoint Deduplication will be done only on the Storage daemon side.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Best Practices

RAID

If you plan to use RAID for performance and redundancy, note that read and write performances are essential for the deduplication engine. The Index is highly accessed for reading and for writing during backup jobs run and during the maintenance tasks required by the deduplication plugin. Also, the optimize process that rebuilds the Index strongly depend on the read and write performance of the disk infrastructure.

Some RAID solutions don't fit the deduplication engine read and write performance requirements. RAID 1 and RAID 10 with a small strip size are recommended for the dedup index. RAID 5, 6, 10, 50, 60 can be used for the containers. The strip size should be compatible with the size of the *extent* that is 4MiB. The goal is for a *stripe* to never be shared by two *extent*. If the extents are well aligned and are a multiple in size with the size of the stripe, then writing a single 4MiB extent should not require to read and rewrite the stripes that are nearby. The *extent* are aligned on a 4MiB boundary inside the *containers*, up to you to configure your raid array, your partitions, your LVM layers and the filesystem parameters (like the sunit and swidth of XFS) to get the container aligned too.

ZFS

If you plan to use ZFS file system to store the dedup index, it is important to guarantee that you have enough memory and CPU resources in the Storage Daemon server to have both the deduplication plugin and ZFS in good condition.

The Global Endpoint Deduplication plugin does both deduplication and chunk compression. This means there is no need to have enabled deduplication or compression in the zfs pool that will be used to store containers. In fact, it is not recommended to have them enabled as it may cause slow performance and there will be no gain in space savings.

Aligned disk acess is a key factor when using ZFS. ZFS is able to detect the sector size of the drive, but disks can report the emulated value instead. As we recommend SSD disks for the dedup Index, performance can be improved if the `ashift` property is explicitly set to match the 4096 byte sector size of SSD disks.

The disk block size is defined by the `ashift` value, but as ZFS stores data in records, there is another parameter that determines the individual dataset to be written in the ZFS pool, this is the `recordsize` ZFS pool parameter.

Thus another important ZFS pool setting to consider is the `recordsize` value. It defaults to 128K in recent ZFS versions. This value should be adjusted to match the typical I/O operations. For the ZFS pool used by the dedup Index, it was reported that setting the `recordsize` to 8K increased the vacuum performance. In addition, setting the ZFS kernel `zfs_vdev_aggregation_limit_non_rotating` parameter to the same value as `recordsize` highly improved performance.

Note these values are recommended for most SSD disks, but they may vary depending on the SSD model. For example, 16K could fit some SSD models and give a better performance. We recommend to perform I/O benchmark using different settings before the deduplication engine is setup in production.

Regarding the use of ZFS to store dedup containers, as it is not possible to preview the typical dataset because some containers can be much more used than others, it is more probably that a `recordsize` value of 64K or even the 128K default value are sufficient to have a good performance. However, it is strongly important to not allow container files to grow too much and limit the size of containers files to 3TB or 4TB.

Maximum Container Size

For better performance, it is strongly recommended to not allow container files to grow indefinitely even if the underlying file system support very big files. This can be accomplished by setting the "Maximum Container Size" directive in the Storage resource in the Storage Daemon configuration file. It is recommended to set this directive to a value between 1 TB and 4 TB.

Vacuum

It is strongly recommended, to keep the deduplication engine healthy, to regularly perform the maintenance tasks triggered by the vacuum.

Make sure to run regularly, in all deduplication engines in your environment, the following tasks: daily prune and truncation of volumes, a simple vacuum daily (it can be run weekly when not too much new chunks are added to the deduplication engine), a dedup vacuum checkindex checkmiss monthly

These maintenance tasks can be scheduled in a job of type Admin and they will help to clean both the deduplication engine index and containers, marking unused entries in the index and chunks in containers, thus allowing the reuse of space. It will contribute to avoid invalid entries in the index to be used by backup jobs in the case of any problems with the server hosting the deduplication engine.

The scrub process is not implemented in Dedup2.

Below are examples of two Admin Jobs that can be used to run a weekly and monthly vacuum.

```
Job {
  Name = "DedupSimpleVacuum_ADMTASK"
  Type = "Admin"
  Client = "bacula-fd"
  Fileset = "LinuxHome"
  Messages = "Standard"
  Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
    Console = "dedup vacuum storage=MyDedup2Storage"
    FailJobOnError = no
    RunsOnClient = no
    RunsWhen = Before
  }
  Schedule = "Vaccum_daily_10AM"
  Storage = "MyDedup2Storage"
}

Schedule {
  Name = "Vaccum_daily_10AM"
  Enabled = yes
  Run = at 10:00
}

Job {
  Name = "DedupDeepVacuum_ADMTASK"
  Type = "Admin"
  Client = "bacula-fd"
  Fileset = "LinuxHome"
```

(continues on next page)

(continued from previous page)

```
  Messages = "Standard"
  Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
    Console = "dedup vacuum checkmiss storage=MyDedup2Storage"
    FailJobOnError = no
    RunsOnClient = no
    RunsWhen = Before
  }
  Schedule = "Vaccum_monthly_secondSunday_11AM"
  Storage = "MyDedup2Storage"
}

Schedule {
  Name = "Vaccum_monthly_secondSunday_11AM"
  Enabled = yes
  Run = 2nd Sun at 11:00
}
```

In an Admin Job, the Fileset and the Pool configured are not used. Thus, you can set up any valid value available in your Bacula Enterprise environment configuration.

Holepunching

Holepunching doesn't exist anymore in Dedup2. The last part of the *Vacuum* release the *extents* that are left empty but also merge the data that are in *extents* that are nearly empty or half used into new containers. This allows new backup to always write into fully recycled extents, reduce the fragmentation and increase the efficiency of the storage.

Dedup 2 Index Rebuild

The Bacula Systems Support Team may ask you to rebuild your deduplication engine version 2 index. This article provides instructions on how to perform this process.

Prerequisites

Before proceeding with the rebuild, ensure that the storage space where the index is stored has a maximum occupancy of 40%.

If the current storage location doesn't meet this requirement, choose an alternative destination folder that can adequately host the rebuilt index. The *-o* option of the `bdde-fsck`command permits to indicate the full path of the new index.

Following these guidelines will help ensure a smooth index rebuild process. The subsequent sections will detail the step-by-step procedure for rebuilding your dedup index.

Instructions

1. Stop the Storage Daemon: the proper use of the bdde-fsck and bdde-check tools require that the Storage Daemon is not running.

```
$ sudo systemctl stop bacula-sd
```

2. Rebuild the index: Use bdde-fsck.

In the example below, we use 10 threads (-T 10) to scan the container files in parallel and create smaller meta data files of the results.

These meta data files are stored in a subdirectory (verify) where the index file is located. Their names match the container files (like verify-00000000.ctn for beed2-00000000.ctn). These meta datafiles are much smaller (44 bytes/index record) than the container files themselves and their total size should be smaller than the size of the current index file.

The container scanning phase of the process is designed to be restartable. In the event of a failure before completion, it can be restarted without the need to rescan containers that have already been successfully scanned.

The following information is required to set up the bdde-fsck command - examples are included:

- the filename of index file: /mnt/index/beeindex.tch
- the directory where the container files are stored: /mnt/containers
- the output filename for the new index file: /mnt/index/repair.tch
- the output filename for the repair log: /mnt/index/repairlog.txt

Before running bdde-fsck, the user must set up the LD_LIBARY_PATH environment variable:

```
$ export LD_LIBRARY_PATH=/opt/bacula/lib
```

Then, run the tool in the background with /usr/bin/nohup to the repair log:

```
$ /usr/bin/nohup /opt/bacula/bin/bdde-fsck -i /mnt/index/beeindex.tch -D /mnt/
↪containers -s -R -o /mnt/index/repair.tch -g -T 10 -dt > /mnt/index/
↪repairlog.txt 2>&1 &
```

3. Monitoring progress (optional): The rebuild can take from several hours to many days depending on the amount of data in the containers and the performance of the container storage. The verify meta data sizes can be used to estimate progress.

For example:

First, determine the size in KiB of the verify directory:

```
$ du -sk /mnt/index/verify
38524
```

Second, use this result to compute the number of index entries that have been scanned from the containers:

```
$ expr 38524 \* 1024 / 44
896558
```

In this simple example there have been almost 900 thousand records scanned. This number can be estimated against the number of records in the current index to determine the progress. The current size of the index can be found with information in the SD tracelog or with `beed-check` on the current index, the RNUM (or record number):

```
$ /opt/bacula/bin/bdde-check -i /mnt/index/beeindex.tch
```

4. Proper completion: When the `bdde-fsck` job completes, there should be messages in the log file (`/mnt/index/repairlog.txt`):

```
: bdde-fsck.cc:595-0 syncing index
: bdde-fsck.cc:597-0 index closed
: bdde-fsck.cc:1103-0 Rebuild of the index ok
```

The user should submit the `repairlog.txt` along with the output from `bdde-check` for the new index file for review in a support ticket before moving on:

```
$ /opt/bacula/bin/bdde-check -i /mnt/index/repair.tch
```

Note: Depending on the errors/warning found on the `repairlog.txt` file, additional steps (to be determined with Bacula Systems Support Team) are required.

---

5. Backup of meta data: Once the rebuild process has been confirmed correct, a backup of the meta data files should be made to avoid rescanning the containers if some other error presents itself:

```
$ mkdir /mnt/index/verify.bkp
$ cp /mnt/indext/verify/* /mnt/index/verify.bkp
```

6. Copy the new index (`repair.tch`) to be the current dedup2 index (`beeindex.tch`), and start the Storage Daemon. At this point we are ready to start the SD with the new index file.

Note: Do not run any backup jobs or vacuum processes yet!

```
$ cp /mnt/index/repair.tch /mnt/index/beeindex.tch
$ sudo systemctl start bacula-sd
```

Send a tail -n 1000 of the SD tracelog to Bacula Systems Support Team for review before moving on.

```
$ tail -n 1000 /opt/bacula/working/bacula-sd.trace
```

7. Run a *vacuum checkmiss* to mark any unused space in the containers. Using bconsole:

   $ dedup vacuum checkmiss storage=<dedup2_storage_name>

Result

Dedup2 index has been rebuilt, and the Storage Daemon is ready to resume normal operations.

Moving from Dedup 1 to Dedup 2

Starting with Bacula Enterprise Edition 16.0, the new dedup2 engine brings a new engine design and a new configuration format. Find more details in the Global Endpoint Deduplication 2 article.

It is not possible to simply reuse the current dedup1 index, containers and volumes data in the new dedup2 engine. It means you cannot configure the dedup2 engine in your environment, and make it use the current dedup1 index, containers and volumes.

However, it is possible to migrate data from the current dedup1 engine (now called dedup legacy; the terms are used interchangeably) to the new dedup2 engine. Also, you may keep the current dedup1 engine for restore purposes, whilst using the new dedup2 engine to store new backups.

In this document, a few scenarios will be presented to help adopting the new dedup2 engine, assuming you are already using the deduplication plugin with the old dedup engine layout and configuration.

Important: Before implementing any strategy in production, read carefully the Important Considerations section before implementing any strategy in production.

Important Considerations

- Running two dedup engines in parallel may be quite intensive in regards to the I/O on the SSD disk, where the dedup index is stored, and also in regards to the I/O where containers are stored, especially if the containers are stored in NFS mounts.
- If the same NFS mount is used to store the containers of both the dedup1 and the dedup2 engines in a migration process, you may have a high load on the NFS mount.
- It is recommended to schedule migration jobs when backup jobs are not running.
- In scenarios where the dedup1 engine is kept for restore purposes, note that the storage occupancy in dedup1 will be the same until this storage is fully decommissioned. This is because the containers cannot be shrunk, even if data is pruned.
- In scenarios where the data is not migrated to dedup2, the new dedup2 engine will start from scratch without a high dedup ratio which also means that you can potentially have the very same data (chunks) in both dedup engines, and as a consequence, the storage occupancy will be high.

We present the scenarios that show how to move on from dedup1 to dedup2.

Note: Scenarios for migration of data from dedup1 to dedup2 are: Scenario 1, Scenario 2 and Scenario 3. Scenarios for keeping dedup1 for restore purposes only are: Scenario 4, Scenario 5 and Scenario 6.

Scenario 1. Dedup Legacy and Dedup2 Running on Different Hosts

Goal of the scenario: Migrating data from your current dedup1 engine to a new Dedup2 engine that use different Storage Daemon hosts.

Note: In this scenario, there are two Storage Daemons running.

It is required to have both dedup engines running at the same time to migrate the data.

The new dedup2 engine will be implemented in a different Storage Daemon host, so only its configuration, using the new DedupEngine resource, is needed. In the bacula-sd.conf file of the new Storage Daemon host, add a DedupEngine resource configured to use `Driver = Dedup2`:

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup2/containers
  Dedup Index Directory = /mnt/SSD/dedup2/index
  Driver = Dedup2
  Maximum Container Size = 2TB
}
```

And the dedup devices should point to the dedup2 engine, for example:

```
Autochanger {
  Name = FileDedup2-Chgr1
  Device = FileDedup2-Dev0, FileDedup2-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}

Device {
  Name = FileDedup2-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}

Device {
  Name = FileDedup2-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}
```

After you have the new Dedup2 engine correctly configured and running, you may use Migration Jobs to migrate jobs from the dedup1 storage to the new dedup2 storage. Once all jobs have been migrated, the current dedup1 engine can be decommissioned.

More details about Migration Jobs: Migration and Copy.

Scenario 2. Dedup Legacy and Dedup2 Running on the Same Storage Daemon

Goal of the scenario: Migrating data from your current dedup1 engine to a new dedup2 engine that use the same Storage Daemon host. The new dedup2 engine uses the same Storage Daemon port, SDPort 9103.

Note: In this scenario, there is only one Storage Daemon running.

It is required to have both dedup engines running at the same time to migrate the data.

If you decide to have both dedup engines in the same host and use the same Storage Daemon port, **it is required to modify the current dedup1 engine configuration to use the new DedupEngine resource**.

The new dedup2 configuration format uses the new DedupEngine resource, and it is not compatible with the old dedup1 configuration format. It means that it is not possible to add the dedup2 engine using the new Dedupengine Resource in the current Storage Daemon configuration if there is already a Dedup1 engine configured using the old configuration format.

For example, **it is not possible** to have both, the old and new formats, configured in the same Storage Daemon configuration file:

```
# From bacula-sd.conf
Storage {
 Name = Dedup1-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
 Dedup Directory = /mnt/bacula/dedup1/containers
 Dedup Index Directory = /mnt/SSD/dedup1/index
 Maximum Container Size = 2TB
}
```

and

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup2/containers
  Dedup Index Directory = /mnt/SSD/dedup2/index
  Driver = Dedup2
  Maximum Container Size = 2TB
}
```

However, **it is possible** to have one dedup legacy engine, and one dedup2 engine in the same Storage Daemon by modifying the current dedup1 configuration to the new format as documented in the New Dedupengine Resource section in the Global Endpoint Deduplication 2 article.

To modify your current legacy Dedup engine to the new configuration format:

1. Stop the Storage Daemon
2. Modify the current dedup1 engine settings by creating a DedupEngine resource with `Driver = Legacy`, and the same index and containers directories of your current dedup engine:

```
Dedupengine {
  Name = Dedupengine_legacy
```

(continues on next page)

(continued from previous page)

```
  Dedup Directory = /mnt/bacula/dedup1/containers
  Dedup Index Directory = /mnt/SSD/dedup1/index
  Driver = Legacy
  Maximum Container Size = 2TB
}
```

3. Add the `Dedupengine = Dedupengine_legacy` directive to all the dedup1 devices:

```
Autochanger {
  Name = FileDedup1-Chgr1
  Device = FileDedup1-Dev0, FileDedup1-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}

Device {
  Name = FileDedup1-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_legacy <--- @aga, should we highlight this?
}

Device {
  Name = FileDedup1-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_legacy <--- @aga, should we highlight this?
}
```

4. Remove or comment the index and containers directories from the Storage resource configuration:

```
# From bacula-sd.conf
Storage {
 Name = Dedup1-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
 # Dedup Directory = /mnt/bacula/dedup1/containers <--- you may remove this␣
↪line
 # Dedup Index Directory = /mnt/SSD/dedup1/index <--- you may remove this line
 Maximum Container Size = 2TB
}
```

5. Add the new dedup2 engine resource and autochanger/devices to the same bacula-sd.conf file:

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup2/containers
  Dedup Index Directory = /mnt/SSD/dedup2/index
  Driver = Dedup2
```

(continues on next page)

(continued from previous page)

```
  Maximum Container Size = 2TB
}

Autochanger {
  Name = FileDedup2-Chgr1
  Device = FileDedup2-Dev0, FileDedup2-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}

Device {
  Name = FileDedup2-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}

Device {
  Name = FileDedup2-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}
```

6. Start the Storage Daemon.

After you have the new Dedup2 engine correctly configured and running, you may use Migration Jobs to migrate jobs from the dedup1 storage to the new dedup2 storage. Once all jobs have been migrated, the current dedup1 engine can be decommissioned.

More details about Migration Jobs: Migration and Copy.

Scenario 3. Dedup Legacy and Dedup2 Running on the Same Host on Two Different Storage Daemons

Goal of the scenario: Migrating data from your current dedup1 engine to a new dedup2 engine that use the same Storage Daemon host. The new Dedup2 engine uses a different Storage Daemon port, for example, SDPort 9104.

Note: In this scenario, there are two Storage Daemons running: one using 9103 SDPort, the other using 9104 SDPort.

It is required to have both dedup engines running at the same time to migrate the data.

In this case, **it is not required** to modify the current dedup1 configuration to use the new format, unless you wish to use the new configuration format to have a standard configuration for all the dedup engines in your environment.

As the new dedup2 engine is to be configured in a new bacula-sd.conf file, there is no need to modify the current bacula-sd.conf file used by Dedup1.

Thus, you should keep the current Dedup1 configuration in the bacula-sd.conf file (if the SDPort = 9103 is not explicitly configured, it is used by default by the Storage Daemon).

Example of the configuration of your dedup1 engine, running on port 9103:

```
# From bacula-sd.conf
Storage {
 Name = Dedup1-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
 Dedup Directory = /mnt/bacula/dedup1/containers
 Dedup Index Directory = /mnt/SSD/dedup1/index    # Recommended to be on fast␣
↪local SSD storage
 Maximum Container Size = 2TB # Maximum 511 containers can be created, please␣
↪adapt to your need
}
```

Then, create a new **bacula-sd-dedup2.conf** file for the new dedup2 engine, using port 9104:

```
 Storage {
  Name = Dedup2-sd
  SDPort = 9104
  Working Directory = /opt/bacula/working
  Pid Directory = /opt/bacula/working
  Plugin Directory = /opt/bacula/plugins
  Maximum Concurrent Jobs = 20
 }

Director {
  Name = bacula-dir
  Password = "password-for-the-director-to-access-the-storage"
}

 Dedupengine {
   Name = Dedupengine_name_with_dedup2
   Dedup Directory = /mnt/bacula/dedup2/containers
   Dedup Index Directory = /mnt/SSD/dedup2/index
   Driver = Dedup2
   Maximum Container Size = 2TB
 }

 Autochanger {
   Name = FileDedup2-Chgr1
   Device = FileDedup2-Dev0, FileDedup2-Dev1
   Changer Command = /dev/null
   Changer Device = /dev/null
 }

 Device {
   Name = FileDedup2-dev0
   Drive Index = 0
   ...
   Device Type = Dedup
```

(continues on next page)

(continued from previous page)

```
   DedupEngine = Dedupengine_name_with_dedup2
 }

 Device {
   Name = FileDedup2-dev1
   Drive Index = 1
   ...
   Device Type = Dedup
   DedupEngine = Dedupengine_name_with_dedup2
 }

Messages {
  Name = Default
  Director = bacula-dir = all
  Append = "/opt/bacula/log/bacula-sd.log" = All, !Skipped
}
```

To start the new dedup2 Storage Daemon, you can use this command: `sudo -u bacula /opt/bacula/bin/bacula-sd -dt -c /opt/bacula/etc/bacula-sd-dedup2.conf`, but it is recommended to create a systemd service for the new Dedup2 Storage Daemon, for example, the bacula-sd-dedup2.service file:

```
# Description:
#    Used to start the bacula Dedup2 storage daemon service (bacula-sd-dedup2)
#    enable : systemctl enable bacula-sd-dedup2
#    start : systemctl start bacula-sd-dedup2
#
#

# from http://www.freedesktop.org/software/systemd/man/systemd.unit.html
[Unit]
Description=Bacula Enterprise Storage Daemon service
Requires=network.target
After=network.target
RequiresMountsFor=/opt/bacula/working /opt/bacula/etc /opt/bacula/bin
/mnt/bacula/dedup2/containers /mnt/SSD/dedup2/index /mnt/bacula/dedup2/volumes

# from http://www.freedesktop.org/software/systemd/man/systemd.service.html
[Service]
Type=forking
User=bacula
Group=disk
Environment="LD_LIBRARY_PATH=/opt/bacula/lib"
ExecStart=/opt/bacula/bin/bacula-sd -dt -c /opt/bacula/etc/bacula-sd-dedup2.
↪conf
StandardError=syslog
TimeoutStopSec=3min
LimitMEMLOCK=infinity
LimitNOFILE=10000
LimitNPROC=10000
OOMScoreAdjust=-1000

```

(continues on next page)

(continued from previous page)

```
[Install]
WantedBy=multi-user.target
```

Then, you can enable and start the new dedup2 engine Storage Daemon using systemd commands:

```
$ sudo systemctl enable bacula-sd-dedup2.service
$ sudo systemctl start bacula-sd-dedup2.service
```

After you have the new Dedup2 engine correctly configured and running, you may use Migration Jobs to migrate jobs from the dedup1 storage to the new dedup2 storage. Once all jobs have been migrated, the current dedup1 engine can be decommissioned.

More details about Migration Jobs: Migration and Copy.

Scenario 4. Setup Dedup2 on a New Host and Redirect All Backup Jobs There

Goal of the scenario: Keeping the dedup1 engine for restore purposes, and setting up a new dedup2 engine for new backups in a different Storage Daemon host.

Note: In this scenario, there are two Storage Daemons running.

In this scenario, you will need to setup the new dedup2 engine in the new Storage Daemon host as described in Scenario 1.

The new Dedup2 engine is to be implemented in a different Storage Daemon host, so only its configuration, using the new DedupEngine resource, is needed. In the bacula-sd.conf file of the new Storage Daemon host, add a DedupEngine resource configured to use **Driver = Dedup2**:

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup2/containers
  Dedup Index Directory = /mnt/SSD/dedup2/index
  Driver = Dedup2
  Maximum Container Size = 2TB
}
```

And the dedup devices should point to this dedup2 engine, for example:

```
Autochanger {
  Name = FileDedup2-Chgr1
  Device = FileDedup2-Dev0, FileDedup2-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}

Device {
  Name = FileDedup2-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
```

(continues on next page)

(continued from previous page)

```
}

Device {
  Name = FileDedup2-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}
```

Then, you should redirect all your backup jobs (this can be done gradually if you have a huge number of Jobs and/or Pools that need to be modified) to the new Dedup2 engine storage.

The dedup1 engine can be decommissioned when the data retention has expired, meaning the retention period for all the jobs previously stored in the dedup1 engine expired, and you will not need to restore any data stored in this dedup1 engine.

If you want to use this scenario to keep the dedup1 engine for restore purposes only, consider to have the dedup1 engine Storage Daemon shut down, and you may start it only when restore jobs are needed.

Scenario 5. Setup Dedup2 on the Same Storage Daemon Host and Using the Same SDPort and Redirect All Backup Jobs There

Goal of the scenario: Keeping the dedup1 engine for restore purposes, and setting up a new dedup2 engine for new backups in the same Storage Daemon host. The new Dedup2 engine uses the same Storage Daemon port, SDPort 9103.

Note: In this scenario, there is only one Storage Daemon running.

This scenario is similar to Scenario 2.

If you decide to have both dedup engines in the same host and using the same Storage Daemon port, **it is required to modify the current dedup1 engine configuration to use the new DedupEngine resource**.

Then, **you must modify** the current dedup1 configuration to use the new configuration format as documented in the New Dedupengine Resource section in the Global Endpoint Deduplication 2 article.

1. Stop the Storage Daemon
2. Modify the current dedup1 engine settings by creating a DedupEngine resource with `Driver=Legacy`, and the same index and containers directories of your current dedup engine:

```
Dedupengine {
  Name = Dedupengine_legacy
  Dedup Directory = /mnt/bacula/dedup1/containers
  Dedup Index Directory = /mnt/SSD/dedup1/index
  Driver = Legacy
  Maximum Container Size = 2TB
}
```

3. Add the `Dedupengine = Dedupengine_legacy` directive to all the dedup1 devices:

```
Autochanger {
  Name = FileDedup1-Chgr1
  Device = FileDedup1-Dev0, FileDedup1-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}

Device {
  Name = FileDedup1-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_legacy <--- @aga, should we highlight this?
}

Device {
  Name = FileDedup1-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_legacy <--- @aga, should we highlight this?
}
```

4. Remove or comment the index and containers directories from the Storage resource configuration:

```
# From bacula-sd.conf
Storage {
 Name = Dedup1-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
 # Dedup Directory = /mnt/bacula/dedup1/containers <--- you may remove this␣
↪line
 # Dedup Index Directory = /mnt/SSD/dedup1/index <--- you may remove this line
 Maximum Container Size = 2TB
}
```

5. Add the new dedup2 engine resource and autochanger/devices to the same bacula-sd.conf file:

```
Dedupengine {
  Name = Dedupengine_name_with_dedup2
  Dedup Directory = /mnt/bacula/dedup2/containers
  Dedup Index Directory = /mnt/SSD/dedup2/index
  Driver = Dedup2
  Maximum Container Size = 2TB
}

Autochanger {
  Name = FileDedup2-Chgr1
  Device = FileDedup2-Dev0, FileDedup2-Dev1
  Changer Command = /dev/null
  Changer Device = /dev/null
}
```

(continues on next page)

(continued from previous page)

```
Device {
  Name = FileDedup2-dev0
  Drive Index = 0
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}

Device {
  Name = FileDedup2-dev1
  Drive Index = 1
  ...
  Device Type = Dedup
  DedupEngine = Dedupengine_name_with_dedup2
}
```

6. Start the Storage Daemon.

Then, you should redirect all your backup jobs (this can be done gradually if you have a huge number of Jobs and/or Pools that need to be modified) to the new dedup2 engine storage.

The dedup1 engine can be decommissioned when the data retention has expired, meaning the retention period for all the jobs previously stored in the dedup1 engine expired, and you will not need to restore any data stored in this dedup1 engine.

If you want to use this scenario to keep the dedup1 engine for restore purposes only, consider to have the dedup1 engine Storage Daemon shut down, and you may start it only when restore jobs are needed.

Scenario 6. Setup Dedup2 on the Same Storage Daemon Host and Using Different SD-Ports and Redirect All Backup Job There

Goal of the scenario: Keeping the dedup1 engine for restore purposes, and setting up a new dedup2 engine for new backups that use the same Storage Daemon host. The new Dedup2 engine uses the a different Storage Daemon port, for example, SDPort 9104.

Note: In this scenario, there are two Storage Daemons running: one using 9103 SDPort, the other using 9104 SDPort.

This scenario is similar to Scenario 3.

In this scenario, **you must setup** the new Dedup2 engine in a new bacula-sd.conf file, for example, bacula-sd-dedup2.conf.

Also, **it is not required** to modify the current dedup1 configuration to use the new format, unless you wish to use the new configuration format to have a standard configuration for all the dedup engines in your environment.

As the new Dedup2 engine is to be configured in a new bacula-sd.conf file, there is no need to modify the current bacula-sd.conf file used by Dedup1.

Thus, you should keep the current Dedup1 configuration in the bacula-sd.conf file (if the SDPort = 9103 is not explicitly configured, it is used by default by the Storage Daemon).

Example of the configuration of your dedup1 engine, running on port 9103:

```
# From bacula-sd.conf
Storage {
 Name = Dedup1-sd
 Working Directory = /opt/bacula/working
 Pid Directory = /opt/bacula/working
 Plugin Directory = /opt/bacula/plugins
 Dedup Directory = /mnt/bacula/dedup1/containers
 Dedup Index Directory = /mnt/SSD/dedup1/index    # Recommended to be on fast␣
↪local SSD storage
 Maximum Container Size = 2TB # Maximum 511 containers can be created, please␣
↪adapt to your need
}
```

Then, create a new **bacula-sd-dedup2.conf** file for the new Dedup2 engine, using port 9104:

```
 Storage {
  Name = Dedup2-sd
  SDPort = 9104
  Working Directory = /opt/bacula/working
  Pid Directory = /opt/bacula/working
  Plugin Directory = /opt/bacula/plugins
  Maximum Concurrent Jobs = 20
 }

Director {
  Name = bacula-dir
  Password = "password-for-the-director-to-access-the-storage"
}

 Dedupengine {
   Name = Dedupengine_name_with_dedup2
   Dedup Directory = /mnt/bacula/dedup2/containers
   Dedup Index Directory = /mnt/SSD/dedup2/index
   Driver = Dedup2
   Maximum Container Size = 2TB
 }

 Autochanger {
   Name = FileDedup2-Chgr1
   Device = FileDedup2-Dev0, FileDedup2-Dev1
   Changer Command = /dev/null
   Changer Device = /dev/null
 }

 Device {
   Name = FileDedup2-dev0
   Drive Index = 0
   ...
   Device Type = Dedup
   DedupEngine = Dedupengine_name_with_dedup2
 }
```

(continues on next page)

(continued from previous page)

```
 Device {
   Name = FileDedup2-dev1
   Drive Index = 1
   ...
   Device Type = Dedup
   DedupEngine = Dedupengine_name_with_dedup2
 }

Messages {
  Name = Default
  Director = bacula-dir = all
  Append = "/opt/bacula/log/bacula-sd.log" = All, !Skipped
}
```

To start the new dedup2 Storage Daemon, you can use this command: `sudo -u bacula /opt/bacula/bin/bacula-sd -dt -c /opt/bacula/etc/bacula-sd-dedup2.conf`, but it is recommended to create a systemd service for the new Dedup2 Storage Daemon, for example, the bacula-sd-dedup2.service file:

```
# Description:
#    Used to start the bacula Dedup2 storage daemon service (bacula-sd-dedup2)
#    enable : systemctl enable bacula-sd-dedup2
#    start : systemctl start bacula-sd-dedup2
#
#

# from http://www.freedesktop.org/software/systemd/man/systemd.unit.html
[Unit]
Description=Bacula Enterprise Storage Daemon service
Requires=network.target
After=network.target
RequiresMountsFor=/opt/bacula/working /opt/bacula/etc /opt/bacula/bin
/mnt/bacula/dedup2/containers /mnt/SSD/dedup2/index /mnt/bacula/dedup2/volumes

# from http://www.freedesktop.org/software/systemd/man/systemd.service.html
[Service]
Type=forking
User=bacula
Group=disk
Environment="LD_LIBRARY_PATH=/opt/bacula/lib"
ExecStart=/opt/bacula/bin/bacula-sd -dt -c /opt/bacula/etc/bacula-sd-dedup2.
↪conf
StandardError=syslog
TimeoutStopSec=3min
LimitMEMLOCK=infinity
LimitNOFILE=10000
LimitNPROC=10000
OOMScoreAdjust=-1000

[Install]
WantedBy=multi-user.target
```

Then you can enable and start the new Dedup2 engine Storage Daemon using systemd commands:

```
$ sudo systemctl enable bacula-sd-dedup2.service
$ sudo systemctl start bacula-sd-dedup2.service
```

Then, you should redirect all your backup jobs (this can be done gradually if you have a huge number of Jobs and/or Pools that need to be modified) to the new Dedup2 engine storage.

The dedup1 engine can be decommissioned when the data retention has expired, meaning the retention period for all the jobs previously stored in the dedup1 engine expired, and you will not need to restore any data stored in this dedup1 engine.

If you want to use this scenario to keep the dedup1 engine for restore purposes only, consider to have the dedup1 engine Storage Daemon shut down, and you may start it only when restore jobs are needed.

Delta Plugin

Important: Delta Plugin is not under active development anymore. The solution has been enhanced, and moved to more recent plugins: ged and aligned-volumes.

Overview

This white paper presents various techniques and strategies to backup special files such as Outlook PST, Database datafile, VirtualBox/VmWare images using block level analysis with **Bacula Enterprise**.

Scope

This paper will present solutions for **Bacula Enterprise** 6.0 and later, which are not applicable to prior versions.

Presentation

The Delta plugin is designed to analyse file differences at the block level and generate binary patches for the backup. It should permit significant space reduction on files that have few modifications between two backups. This plugin is available on all platforms such as Linux and Windows 32/64bit.

Taking Advantage of Delta Backups

Special files such as :

- Outlook PST
- Unix Mbox
- Database
- VMware/VirtualBox/KVM images

will generally benefit most from this technology, other files like :

- Videos
- Photos
- Windows/Linux binaries
- Compressed files
- Encrypted files

probably will not have significant compression ratios using the Delta plugin. The main reason is that most of the time, a single byte modification in an editor will modify all blocks, so the binary patch may be larger than the original file.

Using Delta Plugin

Configuration

As with all Bacula plugins, you must to specify the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

The `delta` plugin uses Accurate mode information to handle incremental and differential backups, thus you must enable the Accurate option in your Job resource.

```
Job {
 Name = "Outlook_PST"
 Client = laptop1-fd
 Fileset = FS_PST
 Accurate = yes
 ...
}
```

To activate the delta plugin on a particular pattern of files such as all **.pst** files, use the following directives:

```
Fileset {
 Name = FS_PST
 Include {
```

(continues on next page)

(continued from previous page)

```
   Options {
     Signature = MD5
     Plugin = delta
     WildFile = "*.pst"
   }
   File = c:/Outlook
 }
}
```

Please note that in the above case, the `Plugin` is specified inside the `Options` block rather than at the same level as the `File` directive.

The Delta plugin also accepts the following parameters:

Table 6: Delta Plugin Options

Option	Comment	Example
min_size=x	Don't use the Delta plugin on files smaller than `x`	min_size=4k

```
Fileset {
 Name = FS_PST
 Include {
   Options {
     Signature = MD5
     Plugin = "delta: min_size=512k"
   }
   File = /home/bacula
 }
}
```

In this example, we split the Fileset to use the Delta plugin on PST files, and the standard compression for other files.

```
Fileset {
 Name = FS_PST
 Include {
   Options {
     Signature = MD5
     Plugin = delta
     WildFile = "*.pst"
   }
   File = c:/Outlook
 }
 Include {
    Options {
      Signature = MD5
      Compression = LZO
    }
    File = "C:/User"
  }
}
```

In this example, only the MYI/MYD files are selected.

```
Fileset {
  Name = FS_MYSQL
  Include {
    Options {
      Signature = MD5
      Plugin = "delta"
      Wildfile = "*.MYI"
      Wildfile = "*.MYD"
    }
    Options {
      Plugin = "delta"
      Exclude = yes
      Wildfile = "*"
    }
    File = /var/lib/mysql
  }
}
```

Requirements

To analyse differences between file versions, the `delta` plugin needs to store signatures in the working directory for each file backed up. Those signatures require some space on your client machine, but don't need to be backed up. They are stored in the `working/delta/` directory. At this time, Bacula will not cleanup the delta working directory, however, you may want to remove entries after some time. You can use the following command for this purpose. Note that you may need to adapt the `60` days parameter to your particular configuration to ensure that you do not remove signatures that are still needed:

```
% find /opt/bacula/working/delta/ -type f -mtime +60 -exec rm -f {} \;
```

The `delta` plugin also needs to store information specific to backup jobs. This information is kept in the `working/delta.history` file.

During restore, Bacula will restore the first backup of the file in a temporary location, then it will apply patches to the temporary file for each incremental/differential backup that was made, finally Bacula will move the temporary file into the right place. This means that you will need working storage to perform the restore. For example, to restore a 2G file, you will need at least 4GB of temporary space.

Development Direction

We are considering implementing the following items in a future version:

- Store signatures in a more efficient way
- Cleanup signatures automatically
- Exclude some file type automatically (jpg, mpg, avi, mkv)
- Apply Delta patches "inplace" rather than using a temporary file

Installation

The `delta` plugin is available as a Bacula Enterprise package for all supported platforms.

Note: More than with traditional Incremental backups, you need to **ensure** that all dependent jobs are available for restore. If even one of your Incremental jobs is missing at restore time, Bacula will not be able to restore your file correctly. In otherwords, if even a single delta is missing, the file cannot be reconstructed. As with normal backup, using the Differential level permits reducing the number of Jobs that are required for restore.

Current Limitations

- On Windows, file attributes such as ACLs are not handled by the Delta plugin.
- The Delta plugin is not compatible with Base job level backups.
- The Delta plugin is not compatible with the `sparse` Fileset option.
- The Delta plugin is not compatible with other plugins in the same Fileset

1.2 Cloud

About Cloud Backup

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. One of those challenges is the growing amount of data to be backed up, together with limited space to keep backups. Another major challenge is to provide adequate off-site backup. Bacula offers several ways to tackle these challenges, one of them being *Bacula Cloud Backup*, which writes Bacula Volumes to different types of cloud services.

This document is intended to provide insight into the considerations and processes required to successfully implement this backup technique.

A major problem of Cloud backup is that data transmission to and from the Cloud is very slow compared to traditional backup to disk or tape. The Bacula Cloud drivers provide a means to quickly finish the backups and then transfer the data from the local cache to the Cloud in the background. This is done by first splitting the data Volumes into small parts that are cached locally and then uploading those parts to the Cloud storage service in the background, either while the job continues to run or after the backup Job has terminated. Once the parts are written to the Cloud, they may either be left in the local cache for quick restores or they may be removed (truncate cache). Truncating cache volumes may also be configured to occur automatically in the background during a job, or after the job has completed. Truncation may also be disabled, or configured to be run manually.

Bacula Systems has implemented drivers for backup and restore to and from several cloud services, whether public or private. The architecture of the Bacula Enterprise cloud backup will provide the user with an array of features to keep the cloud costs to a minimum and the performance to a maximum.

In a continuous effort to increase end user choices, Bacula Systems has broadened its offer of cloud plugins over the last releases. You can now choose from the following plugins:

- S3/Amazon Cloud Plugin
- Azure Cloud Plugin
- Google Cloud Plugin
- Oracle Cloud Plugin
- Swift Object Storage Plugin

Each plugin can be purchased separately. A *Cloud File driver*, which is useful for testing the Cloud architecture without requiring a Cloud account, is also included and could possibly be useful for a disk media device that is very slow.

Cloud Volume Architecture

In the picture above you see two Bacula Cloud Volumes (Volume0001 and Volume0002) with their parts in the local cache. Below the cache, one can see that Volume0002 has been uploaded, or “synchronized” with the Cloud.

Note: Regular Bacula Disk Volumes are implemented as standard files that reside in the user defined **Archive Device** directory. Each regular Bacula Disk Volume is just one file. On the other hand, Bacula Cloud Volumes are directories that reside in the user defined **Archive Device** directory. Each Volume directory contains the Cloud Volume parts which are implemented as numbered files (part.1, part.2, . . .).

Cloud Plugin Installation

Cloud Installation

Cloud Plugin Functionality

The Cloud Plugin write the backup data in a local cache before the data is sent to the Cloud.

It is possible to keep the data in the local cache for some time, or to delete it as soon as the data is stored in the remote cloud.

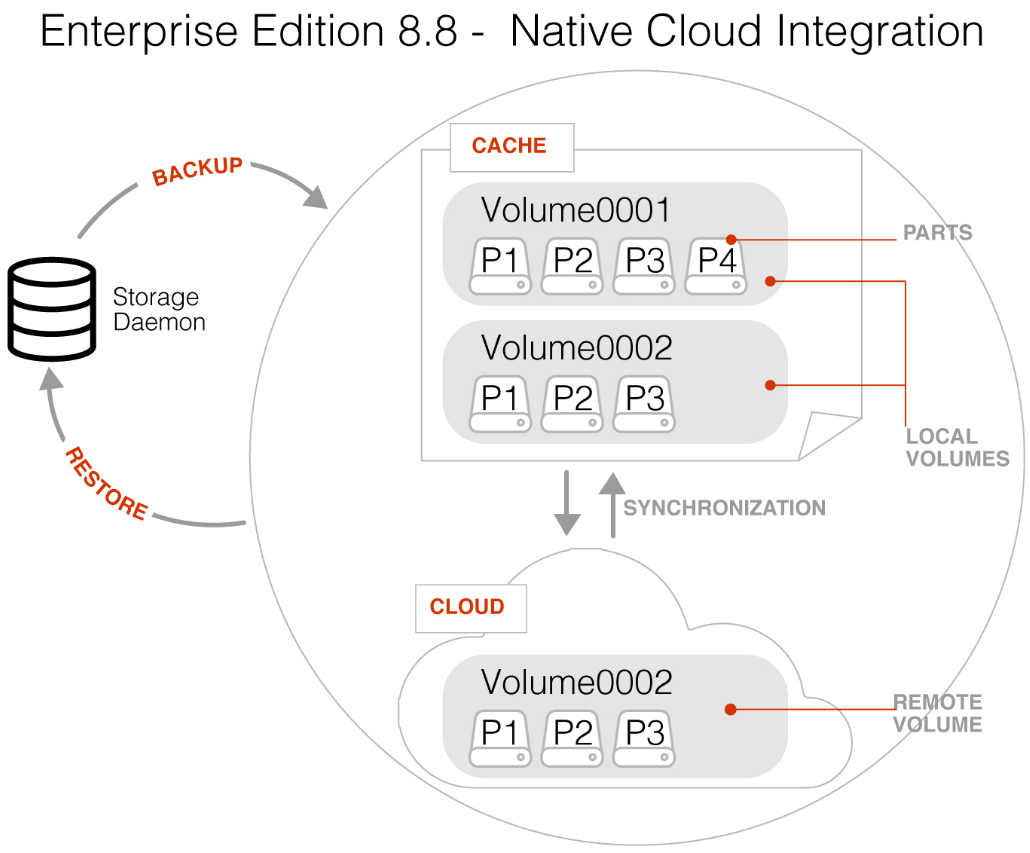


Fig. 4: Bacula Cloud Architecture

Cache and Pruning

The Cloud Cache is treated much like a normal Disk based backup, so that when configuring Cloud backups, the administrator should take care to set "Archive Device" in the Device resource to a directory where he/she would normally store data backed up to disk. Obviously, unless the local cache is configured to be pruned/truncated automatically, the Archive Device file system will continue to fill.

The retention of Cloud volumes in the local Cache is controlled per Volume with the new "CacheRetention" Volume attribute. The default value is 0, meaning that the pruning of Cloud Cache Volumes is disabled. The new "CacheRetention" attribute for Cloud Volumes is configured as a Directive in a Pool and is inherited by Cloud Volumes created in this Pool just as with other inherited attributes for regular disk based Pools and Volumes.

The "CacheRetention" value for a volume may be modified with the bconsole "update" command.

Truncate Cloud Volumes

When Cloud Volumes are truncated by using the `Truncate` bconsole command, all the part files of the volumes are deleted from the Cloud, except the `part.1` file that contains the Bacula Volume label. In the local cache, the `part.2` file is created, with zero bytes, and it is not uploaded to the Cloud. Next time the volume is used by a Bacula Job (backup, copy, etc.), the job will start writing to the Cloud volume `part.2` file.

Cloud Restore

During a restore, if the needed Cloud Volume parts are in the local cache, they will be immediately used, otherwise, they will be downloaded from the cloud as necessary. In such a case, Bacula is efficient and attempts to be as cost effective as possible by downloading only the actual parts of the Cloud Volumes required to perform the restore. The restore starts with Cloud Volume parts already in the local cache but will wait in turn for any part that needs to be downloaded. The downloads proceed in the background while the restore is running.

With most cloud providers uploads are free of charge. On the other hand, downloads of data from the cloud are usually billed. By using local cache and multiple small parts, you can configure Bacula to substantially reduce your Cloud download costs during restores.

The `MaximumFileSize` Device directive is still valid within the Storage Daemon and defines the granularity of a restore chunk. In order to limit volume parts to download during a restore (especially when restoring single files), it might be useful to set the `MaximumFileSize` to a value smaller than or equal to the `MaximumPartSize`.

Compatibility with Other Bacula Functionality

A Cloud Volume stores the same data as any other Bacula Volume type (disk, tape, etc.). Thus, all the existing Bacula funcionality, with the exception of deduplication, is available with the Bacula Cloud Plugin drivers.

Client encrypted data, volumeencryption, compressed data, other plugins data, etc., can be stored in Cloud Volumes.

At the current time, Bacula Global Endpoint Deduplication does not support writing to the cloud because the cloud would be too slow to support large hashed and indexed containers of deduplicated data.

Security and Data Immutability

All data that is sent to and received from the cloud by default uses the HTTPS protocol, so your data is encrypted while being transmitted and received. However, data that resides in the Cloud is not encrypted by default. If you wish extra security of your data while it resides in the cloud, you should consider using Bacula's data encryption features AFUDataEncryption.

For additional protection against backup data loss, or for regulatory compliance reasons, cloud stored parts can be set to be immutable, which means they can be downloaded from the cloud many times but uploaded to the cloud only once (Write Once Read Many: WORM).

Bacula Cloud Plugin supports the immutability features available from different cloud providers. Immutability needs to be configured externally in the destination storage entity (S3 Bucket, Azure Blob, Google Storage Bucket...) using the available native tools from each provider. Further information about these features:

- S3 Object Lock: https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html
- Azure Blob Immutable Storage: https://learn.microsoft.com/en-us/azure/storage/blobs/immutable-policy-configure-container-scope?tabs=azure-portal
- Google cloud Bucket Lock: https://cloud.google.com/storage/docs/bucket-lock
- Oracle cloud Retention Rules: https://docs.oracle.com/en-us/iaas/Content/Object/Tasks/usingretentionrules.htm

Once the destination storage has immutability capabilities enabled, Bacula will work transparently with it. The only requirement is to have greater Bacula retention for the implied volumes than the retention configured in the cloud.

Bucket Versioning

In the case you have bucket versioning enabled in the bucket used to store the Bacula Cloud volumes, you should setup a proper procedure to delete the versioned part files to avoid unnecessary costs.

Versioned part files are created e.g. when Bacula reuses a Cloud volume.

Cloud providers usually propose the setup of Lifecycle policies to delete periodically versioned objects from the bucket.

Cloud Backup Costs

Note: General cost considerations

As you will already know, storing data in the cloud will create additional costs. Please see below information for the different cloud providers.

Data transfer needs to be considered as well. While upload of data is typically free or very low cost, the download is typically not free, and you will be charged which means that restores from the cloud can become expensive. Also note, that when you write data to a volume with Bacula, some data will go the opposite direction for data verification and other important tasks.

You might be able to reduce your storage costs by enabling one of Bacula's available data compression techniques.

S3/Amazon

S3/Amazon for instance has a pricing model for each of its storage tiers, and in addition the costs will vary with the region you use:

- https://aws.amazon.com/s3/pricing/

Azure

With Azure redundancy might influence the price:

- https://azure.microsoft.com/en-us/pricing/details/storage/blobs/

Google

You can refer to the Google Cloud price calculator to estimate your storage costs.

- https://cloud.google.com/products/calculator/

Oracle

You can refer to the Oracle Cloud Cost Estimator to estimate your storage costs.

- https://cloud.oracle.com/en_US/cost-estimator

Remember that Bacula stores its parts to the Object Storage Oracle format. The bucket storage tier influences the price.

Swift

Use the Swift Storage price calculator to estimate your storage costs:

- https://www.swiftstack.com/pricing

Cloud Accounts

Cloud Management

Cloud Compatibility Considerations

S3/Amazon

The S3/Amazon driver is also compatible with any of the following cloud storage technologies:

- Ceph Object Storage, using S3 Gateway
- Swift3 Middleware for OpenStack Swift, using AWS Signature Version 4

Azure

Only Microsoft Azure Storage has been validated.

Commands, Resources, and Directives for Cloud Plugin

To support Cloud backups in Bacula Enterprise 8.8 and newer there are new bconsole cloud commands, *new Storage Daemon directives*, the new Pool directive mentioned above, and a new Cloud resource that is specified in the Storage Daemon configuration.

Cloud Resource

The **Cloud** resource has a number of directives that may be specified as shown in some examples later in this document.

Not all of the directives are mandatory for each plugin (see remarks in individual subsections).

Directives Used by Cloud Resource

The directives of the Cloud resource examples above are defined as follows:

- **Name** The name of the Cloud resource, for instance **MyCloud** in some of the examples above.
- **Description** The description is used for display purposes in Bweb as it is the case with all resources. Not shown in examples above.
- **Driver** This defines which driver to use. Valid options are (depending on the installed cloud driver): **Amazon**, **S3** (deprecated), **Azure**, **Google**, **Oracle**, **Swift** and **File** (**File** is used mostly for testing). **Amazon** is similar to **S3** but uses bacula-sd-cloud-aws-driver instead of bacula-sd-cloud-s3-driver. The specific differences for the Cloud directives that are different in the File driver are discussed in the next section.
- **Host Name** This directive specifies the hostname to be used in the URL. Each Cloud service provider has a different and unique hostname. The maximum size is 255 characters. This directive is not used with the Azure and Swift cloud drivers, but it needs to be specified with dummy data to satisfy the parser. For the Google driver this directive needs to be specified when gcloud has been installed in a custom location. For the Oracle driver this directive needs to be specified when the oci config file is installed in a custom location.
- **Bucket Name** This directive specifies the bucket name that you wish to use on the Cloud service. This name is normally a unique name that identifies where you want to place your Cloud Volume parts. With Amazon S3, the bucket must be created previously on the Cloud service. With Azure Storage it is generally referred to as Container and it can be created automatically by Bacula when it does not exist. With Google Cloud, the bucket must be created previously on the Cloud service. With Oracle Cloud, the bucket must be created previously in the OCI Console.

  The maximum bucket name size is 255 characters.
- **Access Key** The access key is your unique user identifier given to you by your cloud service provider.

  This directive needs to be specified for Bacula Storage daemon compatibility but is not relevant with the Swift driver.

Note: When dealing with the S3 Plugin (Amazon or S3 drivers), confirm the `AccessKeyId value` provided by your S3 Cloud provider is compatible with the Amazon API as it doesn't allow some special characters.

- **Secret Key** The secret key is the security key that was given to you by your cloud service provider. It is equivalent to a password.

  This directive needs to be specified for Bacula Storage deamon compatibility but is not relevant with the Swift driver.
- **Protocol** The protocol defines the communication protocol to use with the cloud service provider. The two protocols currently supported are: **HTTPS** and **HTTP**. The default is **HTTPS**.
- **Uri Style** This directive specifies the URI style to use to communicate with the cloud service provider. The two Uri Styles currently supported are: **VirtualHost** and **Path**. The default is **VirtualHost**.
- **Truncate Cache** This directive specifies if and when Bacula should automatically remove (truncate) the local cache Volume parts. Local cache Volume parts can only be removed if they have been successfully uploaded to the cloud. The currently implemented values are:
  - **No** Do not remove cache Volume parts. With this setting, you must manually delete the cache parts with a **bconsole Truncate Cache** command, or do so with an **Admin** Job that runs a **Truncate Cache** command. If not specified, this is the default.
  - **AfterUpload** Each cache Volume part will be removed just after it is uploaded. Note, if this option is specified, all restores will require a download from the Cloud.
  - **AtEndOfJob** At the end of the Job, every part that has been successfully uploaded to the Cloud will be removed (truncated). Note, if this option is specified, all restores will require a download from the Cloud.
- **Upload** This directive specifies when local cache Volume parts will be uploaded to the Cloud. The options are:
  - **Manual** Do not upload Volume cache parts automatically. With this option you must manually upload the Volume cache parts with a **bconsole Upload** command, or do so with an **Admin** Job that runs an **Upload** command. If not specified, this is the default.
  - **EachPart** With this option, each cache Volume part will be uploaded when it is complete i.e. when the next Volume part is created or at the end of the Job.
  - **AtEndOfJob** With this option all cache Volume parts that have not been previously uploaded will be uploaded at the end of the Job.
- **Maximum Concurrent Uploads** The default is 3, but by using this directive, you may set it to any value that fits your environment.
- **Maximum Concurrent Downloads** The default is 3, but by using this directive, you may set it to any value that fits your environment.
- **Maximum Upload Bandwidth** The default is unlimited, but by using this directive, you may limit the upload bandwidth used globally by all devices referencing this Cloud resource.
- **Maximum Download Bandwidth** The default is unlimited, but by using this directive, you may limit the download bandwidth used globally by all devices referencing this Cloud resource.
- **Region** The Cloud resource may be configured to use a specific endpoint within a region. This directive is required for AWS-V4 regions. ex: `Region="eu-central-1"`

Amazon Driver Directives

- **BlobEndpoint** (available with Bacula Enterprise 16.0.6 and later) this directive can be used to specify a custom URL to Amazon S3 Cloud blob. Example: ``BlobEndpoint="https://my.s3.endpoint"

Note: Starting with Bacula Enterprise 16.0.6, the **BlobEndpoint** directive needs to be set when Amazon driver is utilized with a non AWS cloud storage.

- **StorageClass** (available with Bacula Enterprise 14.0.5 and later) this directive can be used to specify the storage class for all parts transferred to the cloud, independently of the destination bucket class. Values can be **S3Standard**, **S3StandardIA**, **S3IntelligentTiering**, **S3OneZoneIA**, **S3GlacierInstantRetrieval**, **S3GlacierFlexibleRetrieval**, **S3GlacierDeepArchive**, **S3Rrs**. Please, refer to https://aws.amazon.com/s3/storage-classes/ for more details.
- **TransferPriority** (available with Bacula Enterprise 12.2.0 and later) S3 Glacier Flexible Retrieval and S3 Glacier Deep Archive support is provided as a separate option in Amazon Glacier Instant Retrieval and Amazon Glacier Deep Archive). When restoring directly a part from Glacier, this directive indicates the rehydration priority level. Values can be **High**, **Medium** or **Low**. Default is **Low**. Those values match respectively **Expedited**, **Standard** and **Bulk** transfers tiers within S3.

Note: Please note that according to this https://docs.aws.amazon.com/AmazonS3/latest/userguide/restoring-objects-retrieval-options.html, Deep Archive does not support **Expedited** retrieval, hence setting the **TransferPriority** to **High** will not work with retrievals from it.

- **TransferRetention** (available with Bacula Enterprise 12.2.0 and later) S3 Glacier Flexible Retrieval and S3 Glacier Deep Archive support is provided as a separate option in Amazon Glacier Instant Retrieval and Amazon Glacier Deep Archive). This directive indicates the time S3 should keep the rehydrated part online. The value should be at least 1 day. Default is 5 days.

Azure Directives

- **BlobEndpoint** this directive can be used to specify a custom URL for Azure Cloud blob

  https://docs.microsoft.com/en-us/azure/storage/blobs/storage-custom-domain-name.
- **EndpointSuffix** use this directive to specify a custom URL postfix for Azure. Example: `EnbpointSuffix="core.chinacloudapi.cn"`
- **StorageClass** (available with Bacula Enterprise 18.0.4 and later) this directive can be used to specify the storage class (also known as storage tier) for all parts transferred to the cloud, independently of the destination bucket class. Values can be **WASHot**, **WASCool**, **WASCold**, **WASArchive**. Please, refer to https://learn.microsoft.com/en-us/azure/storage/blobs/access-tiers-overview for more details.

Google Directives

- **StorageClass** (available with Bacula Enterprise 18.0.4 and later) this directive can be used to specify the storage class for all parts transferred to the cloud, independently of the destination bucket class. Values can be **GoogleStandard**, **GoogleNearline**, **GoogleColdline**, **GoogleArchive**. Please, refer to https://cloud.google.com/storage/docs/storage-classes for more details.

Oracle Directives

- **StorageClass** (available with Bacula Enterprise 18.0.4 and later) this directive can be used to specify the storage class for all parts transferred to the cloud, independently of the destination bucket class. Values can be **OracleStandard**, **OracleInfrequentAccess**, **OracleArchive**. Please, refer to https://docs.oracle.com/en-us/iaas/Content/Object/Concepts/understandingstoragetiers.htm for more details.

Cloud Resource Examples

S3/Amazon

Amazon Driver

Important: The S3 Driver is now deprecated. If you are still using it, move from the S3 driver to the Amazon driver with the same directives and parameters as they are most of them compatible. Both drivers come with the plugin Cloud S3 or the package **bacula-enterprise-cloud-storage-s3**.

Note: Starting with Bacula Enterprise 16.0.6, the **BlobEndpoint** directive needs to be set when Amazon driver is utilized with a non AWS cloud storage.

Default East USA Amazon Region (us-east-1):

```
Cloud {
  Name = MyCloud
  Driver = "Amazon"
  HostName = "s3.amazonaws.com"
  BucketName = "BaculaVolumes"
  AccessKey = "BZIXAIS39DP9YNER5DFZ"
  SecretKey = "beesheeg7iTe0Gaexee7aedie4aWohfuewohGaa0"
  Protocol = HTTPS
  UriStyle = VirtualHost
  Truncate Cache = AfterUpload
  Upload = EachPart
  Region = "us-east-1"
  MaximumUploadBandwidth = 5MB/s
}
```

Amazon Central Europe Region (eu-central-1):

```
Cloud {
  Name = MyCloud
  Driver = "Amazon"
  HostName = "s3-eu-central-1.amazonaws.com"
  BucketName = "BaculaVolumes"
  AccessKey = "BZIXAIS39DP9YNER5DFZ"
  SecretKey = "beesheeg7iTe0Gaexee7aedie4aWohfuewohGaa0"
  Protocol = HTTPS
  UriStyle = VirtualHost
  Truncate Cache = AfterUpload
  Upload = EachPart
  Region = "eu-central-1"
  MaximumUploadBandwidth = 4MB/s
}
```

For Amazon Simple Storage Service (Amazon S3) Regions, refer to http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region to get a complete list of regions and corresponding endpoints and use them respectively as Region and HostName directives.

Amazon Cloud Storage Plugin Authentication Additions

Use of Amazon `/opt/bacula/working/.aws/credentials` to specify the SecretKey and the AccessKey:

```
Cloud {
  Name = MyCloud
  Driver = "Amazon"
  HostName = "s3-eu-central-1.amazonaws.com"
  BucketName = "BaculaVolumes"
  AccessKey = ""      # Should be set to empty string
  SecretKey = ""      # Should be set to empty string
  Protocol = HTTPS
  UriStyle = VirtualHost
  Truncate Cache = AfterUpload
  Upload = EachPart
  Region = "eu-central-1"
  MaximumUploadBandwidth = 4MB/s
}
```

Note that the credentials file needs to be placed within the home directory of the user under which the Bacula Storage Daemon operates. By default, the user is **bacula** and the home directory is `/opt/bacula/working`. If that is not the case, the credentials file needs to be placed accordingly.

This configuration method is applicable when there is a need to specify the AWS access key ID and secret key outside the Bacula configuration (e.g., the values need to be periodically modified without the restart of the Storage Daemon) or AWS session tokens (temporary security credentials) need to be utilized. In such instances, the actual values for the AWS access key ID, secret key, and the current AWS session token can be set (and periodically reset, using an external program) in the configuration file `.aws/credentials`, and formatted as shown below.

```
[root@bacula-102 .aws]# cat /opt/bacula/working/.aws/credentials
[default]
```

(continues on next page)

(continued from previous page)

```
aws_access_key_id = <snip>
aws_secret_access_key = <snip>
aws_session_token = <snip>
```

Additionally, the same configuration file can be utilized to specify other options related to the AWS CLI, if required.

S3 Object Storages using CEPH Interface

For CEPH interface, use *UriStyle* = *Path* and set the *BlobEndpoint*:

```
Cloud {
  Name = CEPH_S3
  Driver = "Amazon"
  HostName = ceph.mydomain.lan
  BucketName = "CEPHBucket"
  AccessKey = "xxxXXXxxxx"
  SecretKey = "xxheeg7iTe0Gaexee7aedie4aWohfuewohxx0"
  Protocol = HTTPS
  Upload = EachPart
  UriStyle = Path            # Must be set for CEPH
  BlobEndpoint = "https://ceph.mydomain.lan"
}
```

Azure

```
Cloud {
  Name = MyCloud
  Driver = "Azure"
  HostName = "MyCloud"  #not used but needs to be specified
  BucketName = "baculaAzureContainerName"
  AccessKey = "baculaaccess"
  SecretKey = "/Csw1SECRETUmZkfQ=="
  Protocol = HTTPS
  UriStyle = Path
}
```

Google

Since the Google CLI installation already requests credentials, and buckets created with **gsutil** already specify the default localization, Bacula will detect them automatically.

Note: The only mandatory parameters are **Name**, **Driver** and the **BucketName**.

If you have configured **gcloud init** with the `HOME=/opt/bacula/etc/google` parameter, Bacula will search for Google credentials automatically in `/opt/bacula/etc/google`.

```
Cloud {
  Name = MyCloud
  Driver = "Google"
  BucketName = "MyBucket"
}
```

If your Google credentials are stored elsewhere, you can specify a custom **gcloud** configuration **path** through the **HostName** attribute:

```
Cloud {
  Name = MyCloud
  Driver = "Google"
  BucketName = "MyBucket"
  HostName = "/path/to/google-config/folder/"
}
```

Oracle

Since the OCI CLI installation requests credentials, and buckets created with **oci** already specify the default localization, Bacula will detect these automatically.

Note: The only mandatory parameters are **Name**, **Driver**, **BucketName** and **HostName**.

The HostName holds the oci config file location (see ocicliconfig).

If you configured oci into /opt/bacula/etc/oci/, you don't need to specify HostName:

```
Cloud {
  Name = MyCloud
  Driver = "Oracle" #mandatory
  BucketName = "MyBucket" #mandatory
}
```

You can specify a custom oci config **file** through the HostName attribute:

```
Cloud {
  Name = MyCloud
  Driver = "Oracle" #mandatory
  BucketName = "MyBucket" #mandatory
  HostName = "/path/to/oci/config"
}
```

Swift

Note: The only mandatory parameters are Name, Driver, BucketName, HostName, Protocol, AccessKey and SecretKey.

```
Cloud {
  Name = MyCloud
  Driver = "Swift"         # mandatory
  HostName = "MySwiftHost" # mandatory
  Protocol = "http"        # mandatory. Values: "http" or "https"
  BucketName = "MySwiftContainer" # mandatory
  AccessKey = "MySwiftUser" # mandatory
  SecretKey = "MySwiftPassword" # mandatory
  # optional directives
  Truncate Cache = AfterUpload
  Upload = EachPart
  MaximumUploadBandwidth = 5MB/s
}
```

File Driver

As already mentioned, it is possible to use the keyword **File** on the **Driver** directive of the Cloud resource. Instead of writing to the Cloud, Bacula will instead create a Cloud Volume but write it to disk. The rest of this section applies to the **Cloud** resource directives when the File driver is specified.

The File driver is mostly used for testing purposes.

However, if you have a particularly slow backup device you might want to stage your backup data into an SSD or disk using the local cache feature of the Cloud device, and have your Volumes transferred in the background to a slow File device.

The following Cloud directives are ignored: **Bucket Name**, **Access Key**, **Secret Key**, **Protocol**, **Uri Style**. The directives **Truncate Cache** and **Upload** work on the local cache in the same manner as they would for a Cloud.

Host Name, specifies the local destination directory for the Cloud Volume files, and this **Host Name** must be different from the **Archive Device** name, or there will be a conflict between the local cache (in the **Archive Device** directory) and the destination Cloud Volumes (in the **Host Name** directory).

Pool Resource

Within the **bacula-dir.conf** file, for each **Pool** resource there is an additional **CacheRetention** directive that may be specified.

For details, click here.

Storage Daemon Cloud Device Resource

Within the **bacula-sd.conf** file, for each **Device** resource there is an additional keyword **Cloud** that must be specified as the **Device Type** directive, and three new directives: **MaximumPartSize**, **MaximumVolumeParts** and **Cloud**.

For details, click here.

Storage Daemon Cloud Device Directives

- Device Type
- Cloud
- Maximum Part Size
- Maximum Volume Parts

Cloud Device Example

The following is an example of a Bacula Cloud Device resource:

```
Device {
  Name = CloudAmazon-dev0
  Device Type = Cloud
  Cloud = MyCloud
  Archive Device = /opt/bacula/backups
  Maximum Part Size = 10 MB
  Maximum File Size = 10 MB
  Media Type = CloudType
  LabelMedia = yes
  Random Access = Yes;
  AutomaticMount = yes
  RemovableMedia = no
  AlwaysOpen = no
}
```

Note: This is an example of a single device. Usually, you should use an Autochanger having many devices for an optimal performance with concurrent jobs.

As you can see from the above example, the **Cloud** directive in the **Device** resource contains the name (**MyCloud**) of the **Cloud** resource that is shown below. Note also the **Archive Device** is specified in the same manner as one would use for a File device, that is, it simply points to a directory.

However, since this is a Cloud Device, instead of the Storage Daemon writing one file per Bacula Volume in this directory, the Storage daemon will create one directory per Cloud Volume here, and in each of these Cloud Volume directories, the Volume parts will be written.

With the above Device resource example, the two cache Volumes shown in figure *Bacula Cloud Architecture* above would have the following layout on disk:

```
/opt/bacula/backups
   /opt/bacula/backups/Volume0001
      /opt/bacula/backups/Volume0001/part.1
      /opt/bacula/backups/Volume0001/part.2
      /opt/bacula/backups/Volume0001/part.3
      /opt/bacula/backups/Volume0001/part.4
   /opt/bacula/backups/Volume0002
      /opt/bacula/backups/Volume0002/part.1
      /opt/bacula/backups/Volume0002/part.2
      /opt/bacula/backups/Volume0002/part.3
```

Commands Used with Cloud Plugin

Using bconsole, the *cloud* command can be used to manage the cloud volumes and local cache.

See Bconsole Cloud Command.

Status Storage and Statistic Explained

We have information about the upload of cloud volume part files in either the job log or in the output of the bconsole "status storage" command when there is a cloud storage configured.

Job log example

```
20-Apr 16:31 bacula-sd JobId 27: Cloud Upload transfers:
20-Apr 16:31 bacula-sd JobId 27: cloudvolume-Vol-0001/part.1     state=done  ␣
↪  size=401 B duration=8s
20-Apr 16:31 bacula-sd JobId 27: cloudvolume-Vol-0001/part.2     state=done  ␣
↪  size=9.999 MB duration=17s
20-Apr 16:31 bacula-sd JobId 27: cloudvolume-Vol-0001/part.3     state=done  ␣
↪  size=9.999 MB duration=17s
20-Apr 16:31 bacula-sd JobId 27: cloudvolume-Vol-0001/part.4     state=done  ␣
↪  size=9.999 MB duration=14s
```

** Example of a bconsole status storage command output**

```
Cloud transfer status:
   Uploads    (1.642 MB/s) (ETA 0 s) Queued=0 0 B, Waiting=0 0 B, Processing=0␣
↪0 B, Done=53 524.7 MB, Failed=0 0 B
   Downloads (0 B/s) (ETA 0 s) Queued=0 0 B, Waiting=0 0 B, Processing=0 0 B,␣
↪Done=0 0 B, Failed=0 0 B
```

We also have information in BWeb.

Duration value needs to be considered as the amount of time the cloud upload operation took. It can be for a single part file or for multiple part files as multiple part files can be uploaded into the cloud. The main goal of the duration value is to have an idea if something takes too long and it is possible that another process is affecting the cloud volumes upload speed.

Timestamp in the job log of each part file upload doesn't match exactly with the value of the part file in the remote cloud as this value is the timestamp of the part file creation in the bucket and they can be different.

Uploads value (XXX KB/s) means that the upload rate is XXX KB/s. It is not a median of all the values uploaded, but a rate of the last uploaded part file or even part files if they have been uploaded concurrently. So, the Uploads value is not an average nor cumulative value.

On the other hand, the **Queued**, **Waiting**, and the **Processing** values are computed for all part files which been uploaded, even from other jobs.

Then, the **Done** and **Failed** values are cumulative values since the Storage Daemon startup. As soon as the Storage Daemon is restarted, these values are list and are reset to zero.

ETA = Data to transfer/global speed, and ETA for each transfer is the same, but applied only to the transfer. Different ETAs are computed at a given time with the given resources allocated to the cloud transfer and the global network capabilities. It's done based on the last part transfer duration vs size. So it's as close to an "instant" transfer rate that it can be. The values can change over the time.

Creating and Verifying your Cloud Account

Cloud Management

Limitations

- The support of RHEL 6 has been deprecated with Bacula Enterprise Cloud storage driver version 8.10.
- **MaximumVolumeBytes** and **MaximumPoolBytes** directives is not implemented in cloud volumes like in regular volumes. Other directives, like **MaximumVolumeParts**, **VolumeUseDuration** or **MaximumVolumeJobs** (with concurrent jobs =1) must be used to set a limit on these volumes. The **MaximumVolumeParts** can be used to limit the size of a cloud volume, used along with the **MaximumPartSize** directive. Setting **MaximumPartSize** will also help controlling the part size on the cloud.
- On restore, all cloud volume parts that are needed will be downloaded from the cloud into the local cache storage before being transferred to the FD. This means that if you have a Full backup of 20TB (for example) that needs to be restored, you will need to have at least 20TB of local cache storage available. Consider also, if there are backups running at the time of restore, or if there are local cloud volume parts that have not yet been truncated, there will be cloud storage space already in use so you will need to pay close attention before and during large restores from the cloud to be sure that the local cloud cache storage space is not filled to capacity.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Best Practices

- Set the **MaximumFileSize** to a value smaller than or equal to the **MaximumPartSize** as the **MaximumFileSize** defines the granularity of the restore chunk. Having the **MaximumFileSize** directive defined this way will allow Bacula to only download the required part file(s) to restore specific files and/or directories in the case of partial restores, thus reducing download costs.
- Have a reasonable value configured for **MaximumPartSize**. Smaller part sizes will reduce restore costs, but may require a small additional overhead to handle multiple parts. Also, some cloud providers limit the size of a single object to be uploaded. A part file is considered a single object and it will use a single upload operation. Thus you must set a value lower or equal than this limit

for the **MaximumPartSize**, otherwise the upload of part files will fail. Please confirm with your cloud provider the maximum object size upload, if any.

- Regularly run `cloud upload` and `cloud truncate` **bconsole** commands. They can be scheduled in an Admin Job. Even when the Cloud resource is configured to automatically upload volume part files and/or truncate them from the local cache, connection issues can prevent some part files from being successfully uploaded and a local cache truncation may not occur for part files which have not yet been uploaded. It is recommended to retry the volume part file upload and/or local cache truncation in the case of failures by manually running the `cloud upload` and `cloud truncate` **bconsole** commands. You can also use a Bacula Admin Job that will retry the part file upload automatically on a regular basis.

  If the restore process takes a long time, it is recommended to temporarily deactivate the `cloud upload` (if `TruncateCache = yes`) and/or the `cloud truncate` commands to prevent essential parts needed for the restore from being truncated from the local cache.

  Please find an example for such an Admin Job that can be used to periodically trigger the cloud upload and the cloud truncate commands:

```
Job {
  Name = CloudUpload-adminjob
  Type = Admin
  Client = bacula-fd  # any client can be used
  Schedule = DailyCloudUpload
  RunScript {
    RunsOnClient = No
    RunsWhen = Always
    Console = "cloud upload storage=<cloud_storage_name> allpools"
    Console = "cloud truncate storage=<cloud_storage_name> allpools"
  }
  Storage = <cloud_storage_name>
  Messages = Default
  Pool = Fake-pool
  Fileset = Fake-fileset
}
```

- After a restore, the downloaded part files are not truncated from the local cache even if **Truncate Cache** is configured in the Cloud resource (to truncate the local cache **AfterUpload** or **AtEndOfJob**). If you have your environment configured to truncate the local cache after the part is successfully uploaded or at the end of a job, we recommend to either manually truncate the local cache after the restore successfully finishes or to have it truncated by an Admin Job that periodically truncates the local cache (the Admin Job as recommended in the previous point).

- It is considered a best practice to set **MaximumConcurentJobs = 1** in all of the Cloud Device resources (**DeviceType = Cloud**) that are defined on the SD. This will guarantee that only one job is writing to the device at one time, so data belonging to different jobs will not be interleaved in the part files. When dealing with cloud backups, this helps to minimize the overall download costs by downloading only the required part files from specific backup jobs during a restore. Of course, you need to consider defining a larger number of Cloud Devices, appropriate to the maximum number of jobs you are expecting to run concurrently.

- Retention locked objects (terminology varies among providers) can be used, but to avoid failures when Volumes are recycled, it is strongly recommended to have Bacula's retention times set in a way that they expire only after the object is no longer in retention. This applies particularly to **VolumeRetention**, but **JobRetention** can indirectly affect Volume recycling too. If Bacula attempts to recycle a Volume that is still in locked state, the corresponding job will fail, and the offending Volume will be marked with status **Error**.

Cloud Installation

Installation of the Cloud S3/Amazon Plugin

Important: The S3 Driver is now deprecated. If you are still using it, move from the S3 driver to the Amazon driver with the same directives and parameters as they are most of them compatible. The S3 and the Amazon driver are part of the same package named *bacula-enterprise-cloud-storage-s3*.

Note: The difference between S3 and Amazon Drivers is that S3 uses libs3, while Amazon uses AWS CLI API. The newest solution is the Amazon driver based on Amazon native library, while the S3 driver is based on unofficial S3 libraries. Amazon Driver is recommended as it is continuously supported and maintained, unlike S3 Driver.

AWS CLI is a requirement for the Cloud S3/Amazon Plugin. Refer to the AWS Documentation: https://aws.amazon.com/

Both drivers S3 and Amazon drivers backup data to the cloud and are installed on the Storage Daemon. The Cloud S3 plugin backing up data from S3 cloud objects to the Storage Daemon can be found *here*.

On the Bacula Storage Daemon that you want to connect to your cloud storage, extend the repository file for your package manager to contain a section for the Cloud plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseCloudPlugin]
name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-s3/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Bookworm, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise deb
https://baculasystems.com/dl/@customer-id@/debs/bin/@version@/bookworm-64/
bookworm main deb
https://baculasystems.com/dl/@customer-id@/debs/cloud-s3/@version@/bookworm-
↪64/
bookworm cloud-s3
```

Then perform a `yum update` or `apt-get update`, and after that install the package *bacula-enterprise-cloud-storage-s3*.

Installation of the S3/Amazon Glacier plugin (Deprecated)

Important: This section is deprecated. It is recommended to use the “Amazon Driver” for the Glacier feature which does not need any additional package or plugin.

Note: Prerequisite: *bacula-enterprise-cloud-storage-glacier* requires *bacula-enterprise-cloud-storage-s3* to be installed first.

Additionally, if you also purchased the Glacier plugin (available with Bacula Enterprise 12.2.0 and later), extend the repository file for your package manager to contain a section for the Glacier plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseCloudPlugin] name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-s3/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseGlacierPlugin] name=Bacula Enterprise Glacier Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-glacier/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Bookworm, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise deb
https://baculasystems.com/dl/@customer-id@/debs/bin/@version@/bookworm-64/␣
↪bookworm main deb
https://baculasystems.com/dl/@customer-id@/debs/cloud-s3/@version@/bookworm-
↪64/ bookworm cloud-s3 deb
https://baculasystems.com/dl/@customer-id@/debs/cloud-glacier/@version@/
↪bookworm-64/ bookworm cloud-glacier
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-cloud-storage-s3* and, optionally, *bacula-enterprise-cloud-storage-glacier* can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

Installation of the Cloud Azure Plugin

Installation of the Azure CLI

Previous to the plugin installation, the corresponding distribution package of the CLI must be installed.

For RHEL, refer to:

- https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-yum?view=azure-cli-latest

For Debian or Ubuntu, refer to:

- https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latest

For other platforms, refer to:

- https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

Installation of the plugin

On the Bacula Storage Daemon that you want to connect to your cloud storage, extend the repository file for your package manager to contain a section for the Cloud plugin.

For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseCloudPlugin]
name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-azure/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Bookworm, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://baculasystems.com/dl/@customer-id@/debs/bin/@version@/bookworm-64/
↪ bookworm main
deb https://baculasystems.com/dl/@customer-id@/debs/cloud-azure/@version@/
↪bookworm-64/ bookworm cloud-azure
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-cloud-storage-azure* can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (rpm or dpkg) to do the plugin installation.

Installation of the Cloud Google Plugin

On the Bacula Storage Daemon that you want to connect to your cloud storage, extend the repository file for your package manager to contain a section for the Cloud plugin. For example in RHEL, /etc/yum.repos.d/bacula.repo:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseCloudPlugin]
name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-google/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Bookworm, /etc/apt/sources.list.d/bacula.list

```
#Bacula Enterprise
deb https://baculasystems.com/dl/@customer-id@/debs/bin/@version@/bookworm-64/
↪ bookworm main
deb https://baculasystems.com/dl/@customer-id@/debs/cloud-google/@version@/
↪bookworm-64/ bookworm cloud-google
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-cloud-storage-google* can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (rpm or dpkg) to do the plugin installation.

Installation of the Cloud Oracle Plugin

On the Bacula Storage Daemon that you want to connect to your cloud storage, extend the repository file for your package manager to contain a section for the Cloud plugin. For example in RHEL, /etc/yum.repos.d/bacula.repo:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

(continues on next page)

(continued from previous page)

```
[Bacula EnterpriseCloudPlugin]
name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-id@/rpms/cloud-oracle/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Bookworm, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://baculasystems.com/dl/@customer-id@/debs/bin/@version@/bookworm-64/
↪ bookworm main
deb https://baculasystems.com/dl/@customer-id@/debs/cloud-oracle/@version@/
↪bookworm-64/ bookworm cloud-oracle
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-cloud-storage-oracle* can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

Installation of the Cloud Swift Plugin

On the Bacula Storage Daemon that you want to connect to your cloud storage, extend the repository file for your package manager to contain a section for the Cloud plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel9-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseCloudPlugin]
name=Bacula Enterprise Cloud Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/cloud-swift/
↪@version@/rhel9-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Jessie, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪jessie-64/ jessie main
deb https://www.baculasystems.com/dl/@customer-string@/debs/cloud/@version@/
↪jessie-64/ jessie cloud
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-cloud-storage* can be installed with `yum install` or `apt-get install`. On Oracle Linux/AlmaLinux/Rocky Linux, the EPEL package needs to be installed to avoid a dependency problem with *libxml++*.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

Cloud Management

S3/Amazon Account Management

Important: The S3 Driver is now deprecated. If you are still using it, move from the S3 driver to the Amazon driver with the same directives and parameters as they are most of them compatible.

Create an Account

Amazon offers high-level S3 commands with the AWS Command Line Interface. The tool (aws) is mandatory for the Bacula Amazon driver to work. It can also be used to verify manually your account and list everything you have stored in you Amazon S3 buckets. The advantage of testing your setup with the **aws** command is that the same Amazon credentials are used in accessing S3 via **aws** as in the Bacula Cloud resource.

Go to the following link and create an S3/Amazon account.

- https://aws.amazon.com/s3

Then you might want to use the S3/Amazon tutorial. Note that some of the information in the tutorial is repeated below for your convenience.

- https://aws.amazon.com/s3/getting-started/

Install the AWS Command Line Interface

A guide which explains how to install the AWS Command Line Interface (CLI) can be found here:

- http://docs.aws.amazon.com/cli/latest/userguide/installing.html

We chose pip to install the AWS CLI. Amazon recommends to have at least Python version 2.7.

Make sure that Python is installed:

```
[root@localsd1 ~]# python --version
Python 3.10.6
[root@localsd1 ~]#
```

The pip tool was not installed in our case which can be checked with:

```
[root@localsd1 ~]# pip --help
-bash: pip: command not found
[root@localsd1 ~]#
```

Installation is easy (but produces warnings with Python versions < 2.7):

```
[root@localsd1 ~]# curl -O https://bootstrap.pypa.io/get-pip.py
[root@localsd1 ~]# python get-pip.py
```

Once you have pip, you can install the AWS CLI and check if the installation was successful:

```
[root@localsd1 ~]# pip install awscli
[root@localsd1 ~]# aws help
```

Define user and export credentials

This step is not mandatory if you plan to use aws only thru the Bacula Amazon cloud driver, since Bacula will automatically use the Bacula Cloud resource credentials, but it is recommended for using aws as a standalone command line interface.

To be able to access your S3/Amazon buckets, you will need to create an authorized user in the web interface: *Services* → *Security & Identity* → IAM. Attach the policy *AmazonS3FullAccess* to this user. Create access keys in the web portal (tab *Security Credentials* in IAM) and export them as a CSV file.

Use the configure command:

```
[root@localsd1 ~]# aws configure
```

to store the credentials locally (see also http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html)

Create a Bucket in S3/Amazon

Log into your AWS console (https://aws.amazon.com/console/), select the correct region, and choose *Services* → *Storage & Content Delivery* → S3. Create a bucket and give it a unique name. In our test scenario we created a bucket called *bsyssdsync* with one folder *volumes* inside it.

You can also create the bucket with the AWS Command Line Interface:

```
[root@localsd1 ~]# aws s3api --endpoint-url='https://s3.amazonaws.com' \
  create-bucket --bucket bsyssdsync
```

Copy and sync files

You can list your S3/Amazon buckets with:

```
[root@localsd1 ~]# aws s3 ls
2016-04-11 21:13:02 bsyssdsync
[root@localsd1 ~]#
```

To copy volumes from your local SD to the cloud, use:

```
[root@localsd1 ~]# cd /srv/bacula-storage
[root@localsd1 bacula-storage]# ls
Vol-0002  Vol-0005
[root@localsd1 bacula-storage]#
[root@localsd1 bacula-storage]# aws s3 cp Vol-0002 s3://bsyssdsync/volumes/
```

(continues on next page)

(continued from previous page)

```
upload: ./Vol-0002 to s3://bsyssdsync/volumes/Vol-0002
[root@localsd1 bacula-storage]#
```

This of course only makes sense when you have only one job per volume and your volumes are marked Full by Bacula after the job. You could trigger the upload to the cloud in a RunScript after the job and then delete the local copy. You would have to make sure though that in the restore case all volumes are available again in the local file system.

The **aws s3 cp** works in both directions:

```
[root@localsd1 ~]# aws s3 cp <source> <destination>
```

and behaves like the UNIX cp command (more details: http://aws.amazon.com/cli/). However, when you have more than one job per volume, and volumes with fixed maximum size configured in Bacula, you will want to sync the directory with Bacula volumes to S3. The AWS CLI has a command for that:

```
[root@localsd1 ~]# aws s3 sync /srv/bacula-storage s3://bsyssdsync/volumes
```

In this case you would identify Full volumes with a Bacula Catalog query and delete them after all backups have been run and the volumes have been synced. Again, you will need to make sure that they are available when you want to restore data.

When it comes to Bacula reusing volumes (after the configured retention times have passed), you would probably use a different configuration approach in a cloud scenario: Configure retention times to be indefinitely long (i.e. years), the volumes will be synced away into the cloud, deleted from disk, and then you will use Amazon mechanisms to tier the volumes to less expensive storage, from

General Purpose (Amazon S3 Standard)

↓

Infrequent Access (Amazon S3 Standard - Infrequent Access)

↓

Archive (Amazon Glacier)

and finally delete them. Learn more about Amazon Storage Classes: https://aws.amazon.com/s3/storage-classes/. Policies can be set for each bucket independently in the AWS web portal.

S3/Amazon Glacier Instant/Flexible Retrieval (Deprecated)

Important: The S3 Driver is now deprecated. If you are still using it, move from the S3 driver to the Amazon driver with the same directives and parameters as they are most of them compatible. The S3 and the Amazon driver are part of the same package named *bacula-enterprise-cloud-storage-s3*.

Another effective strategy is to use Bacula direct restoration from S3/Amazon Glacier Instant/Flexible Retrieval (available with Bacula Enterprise 12.2.0 and later) or S3/Amazon Glacier Deep Archive (available with Bacula Enterprise 14.0.0 and later). Configure the bucket lifecycle to automatically handle transition to S3 Glacier Instant/Flexible Retrieval or S3/Amazon Glacier Deep Archive after backup:

- https://docs.aws.amazon.com/en_pv/AmazonS3/latest/user-guide/create-lifecycle.html and use the **TransferPriority** and **TransferRetention** Cloud directives to configure rehydration before restoration. See *Amazon Driver Directives* for details. Bacula will monitor the rehydration process until its completion and proceed normally with parts download and restoration afterwards.

S3/Amazon Glacier Deep Archive (Deprecated)

Important: The S3 Driver is now deprecated. If you are still using it, move from the S3 driver to the Amazon driver with the same directives and parameters as they are most of them compatible. The S3 and the Amazon driver are part of the same package named *bacula-enterprise-cloud-storage-s3*.

Bacula direct restoration can also be used to restore from S3/Amazon Glacier Deep Archive (available with Bacula Enterprise 12.2.0 and later. S3/Amazon Glacier Instant Retrieval and S3/Amazon Glacier Deep Archive support is provided as a separate option). Configure the bucket lifecycle to automatically handle transition to S3/Amazon Glacier Deep Archive after backup:

- https://docs.aws.amazon.com/en_pv/AmazonS3/latest/user-guide/create-lifecycle.html

and use the same **TransferPriority** and **TransferRetention** Cloud directives as for Glacier to configure rehydration before restoration. see *Amazon Driver Directives* for details. Bacula will monitor the rehydration process until its completion and proceed normally with parts download and restoration afterwards.

Time coordination

It seems that S3/Amazon syncing is sensitive to clock differences. If you get a S3-RequestTimeTooSkewed error during **aws s3 sync** you should use Amazon NTP servers:

- http://www.emind.co/how-to/how-to-fix-amazon-s3-requesttimetooskewed

S3 Object Lock

If **Object Lock** is configured on your target bucket, do not use the **S3 driver** to backup to it, but rather the more recent **Amazon driver**, by changing the Driver keyword of the Cloud resource in your SD configuration from “S3” to “Amazon”.

Once the destination storage has immutability capabilities enabled, Bacula will work transparently with it. The only requirement is to have greater Bacula retention for the implied volumes than the retention configured in the cloud.

Note: To read more about security and data immutability, click *here*.

Azure Account Management

Create a Microsoft Azure Account

Please go to the following link and create an Microsoft Azure account. Trials are available.

- https://azure.microsoft.com

Then follow the following link to login.

- https://portal.azure.com

Configure your Azure Blob Storage

You have to do a few configuration steps to prepare your Azure portal for backup with Bacula Enterprise. The Bacula Systems Support Team can provide screenshots for the below steps upon request.

1. Login to your Azure portal and create a new *Storage account* that you either assign to an existing resource group (or you can also create a new one on the fly).
2. Give it a name; Must be lowercase letters and numbers only, and must be unique in all of Azure In the "Account kind" field select "Blob storage" For "Resource Group", click "Use existing", then choose the Resource Group from the drop-down Click on "Create" to save. This may take a minute to deploy.
3. Select the new storage account in your *Storage accounts overview* page and navigate to *Settings* and then *Access keys* to retrieve the **Secret Key** that you will need to specify in your Bacula Storage Daemon configuration. The *Storage account Name* will be the **Access Key** in your Cloud resource in `bacula-sd.conf`.
4. Now click on *Blob service* → *Containers* to create a new container with public access level *Blob (anonymous read access for blobs only)*. This will be the **BucketName** that we use in the Bacula Cloud resource.
5. Confirm that the Settings in the Azure portal which were just created/edited match the Bacula Cloud resource config file.
6. Stop the bacula-sd daemon: `systemctl stop bacula-sd`
7. Test the SD config syntax: `/opt/bacula/bin/bacula-sd -u bacula -t`
8. Start the bacula-sd daemon: `systemctl start bacula-sd`
9. Run a backup job to the Azure storage.

GCP Account Management

The Python application `gsutil` lets you access Cloud Storage from the command line. It is part of the Google Cloud SDK which installation is platform dependent.

Please, browse to this location:

- https://cloud.google.com/sdk/docs/

Choose your platform and follow the google-cloud-sdk installation steps:

> **Debian/Ubuntu**. Follow the installation steps up to `apt-get install google-cloud-sdk`. Optional steps are not required.

RedHat. Follow the installation steps up to `yum install google-cloud-sdk`. Optional steps are not required.

Initialize Cloud Platform access

In order to authorize the bacula user to access Google Cloud Platform through gsutil, run the `setup_google_cloud_cli` script located in `/opt/bacula/scripts` as root to guide you through this steps.

You will be asked for your account information and to select or create a *project*. These project and credentials will later be used by the Plugin.

Alternatively, you can launch the following command as root:

```
sudo -u bacula HOME=/opt/bacula/etc/google gcloud init
```

If you plan to use a service account to authenticate, use the following command using the JSON file that contains your service account key:

```
sudo -u bacula HOME=/opt/bacula/etc/google gcloud auth activate-service-
↪account --project=<project_id> --key-file=/path-to-json-file/service-
↪account-xxxxxxxxx.json
```

The Google credentials will be stored inside `/opt/bacula/etc/google` and will belong to the unix user "bacula". To use another location, it will be required to adapt the Cloud resource definition.

Create a Bucket in Google Cloud

There are at least 2 ways you can use to create a Bucket in your selected project. We recommend to use the web console since it's giving you usefull information on fields and pricing.

Use the web console

Log into the web console:

- https://console.cloud.google.com/

From the top right menu, select **Storage**, then click **Create Bucket**. Specify the *name*, *class* and *location* of the bucket with the help of context menus. Click **Create**.

Use the command line mb (make bucket) gsutil command

From a terminal, type:

```
gsutil mb [-c class] [-l location] gs://<some-bucket>
```

- *class*. Optional. The storage **class** you choose for this bucket (see https://cloud.google.com/storage/docs/storage-classes for details).
- *location*. Optional. The storage **region** location you choose for this bucket (see https://cloud.google.com/storage/docs/bucket-locations for details).
- *<some-bucket>* is your bucket **name**.

The storage *class* influences the storage price. Depending on your backup strategy, **nearline** or even **coldline** might be the most appropriated classes.

Configure objects to be retention locked

This configuration is optional

Retention locking of objects can provide additional security against premature loss of backup data.

See https://cloud.google.com/storage/docs/bucket-lock for details.

For an existing bucket, use

```
gsutil retention set <seconds>s gs://<some-bucket>/
```

- <*seconds*> is the number of seconds an object is going to be locked.
- <*some-bucket*> is the name of an existing bucket used to store backups in, probably one just created.

To verify retention lock time, use a command such as

```
gsutil retention get gs://<some-bucket>/
```

Oracle Cloud Account Management

Cloud Account

To configure your Bacula Storage daemon with Oracle Cloud Infrastructure, you need to create an account through the Oracle Cloud portal.

- https://cloud.oracle.com/

Note that the registration process requires a Default Data Region. Bacula Storage deamon stores to the Oracle Cloud Object Storage. Make sure you that you select a region that supports it. See the following link for details on regions

- https://cloud.oracle.com/data-regions/

Once you've entered the required information, you should be granted access to the Oracle Services Dashboard and receive a confirmation e-mail containing a link to connect the Oracle Services Dashboard.

Obtain Oracle Cloud Infrastructure Console URL

For the next operations, you'll need to access your Oracle Cloud Infrastructure Console. Follow the next steps to retrieve the OCI Console URL. (If you already know your OCI console URL, you can skip this section.)

- Follow the link provided in the registration e-mail to reach the **Services Dashboard**.
- In the top right corner of the Dashboard, select you **account Icon** and choose **My Admin Accounts** from the drop down menu.

- From the **Administrative Accounts** list Copy the URL associated to **Compute (OCI) Users**. That the URL is region-based. It should start like this: `https://console.<region>.oraclecloud.com/...`

Keep track of this URL, it's your main access point to the Oracle Storage.

Create a Bucket in Oracle Cloud

Before starting using Plugin, you need to create a Bucket in your Oracle Cloud Infrastructure Console. Bacula will create all its backup volumes in this Bucket.

1. Browse to your OCI Console
2. From the top left Menu, select **Object Storage**
3. Click **Create Bucket**
4. Fill the different creation fields, specifically:

- **Bucket Name**: Will be used as specified here within the Bacula Cloud Resource
- **Storage Tier**: The bucket Storage Tier influences the storage price and cannot be changed afterward: https://cloud.oracle.com/storage/archive-storage/faq

  Standard will provide instant access to the bucket objects, while with **Archive** you'll have to **manually restore your objects with the OCI client before Bacula can restore them**: https://docs.cloud.oracle.com/iaas/tools/oci-cli/latest/oci_cli_docs/cmdref/os/object/restore.html
- **Tags**: Tags can be attached to your bucket to organize your tenancy. Bacula will not consider them.

Retrieve the OCID keys

These keys are required to complete OCI Command Line Interface installation.

Tenancy's OCID

The tenancy OCID is in the Console in the Tenancy Details page: From the OCI Console, open the **User menu** in the top right corner and click The **tenancy OCID** is shown under Tenancy Information.

User's OCID

The user OCID is in the Console in the User Settings page: From the OCI Console, open the **User menu** in the top right corner and click The **user OCID** is shown under User Settings.

Install the OCI Command Line Interface tool

The OCI Command Line Interface (CLI) is a Python application that allows access to Oracle Cloud Storage from the command line. Its installation is platform dependent as described in:

- https://docs.cloud.oracle.com/iaas/Content/API/SDKDocs/cliinstall.htm

Since you should be running on a Unix-like platform, run the following bash command as root:

The installed binaries must be on the path so it's important to modify the default location. We recommand installing in **/usr/local/bin**

- *install location* should be changed to **/usr/local/lib/oracle-cli**
- *OCI executable location* should be changed to **/usr/local/bin**
- *OCI scripts location* should be changed to **/usr/local/bin/**
- later on, you may be asked to modify the path. Answer no.

Eventually, you should get prompted with *Installation successful*.

Setup the OCI CLI config

Once the OCI CLI has been installed, you can run the setup_oracle_cloud_cli script located in */opt/bacula/scripts* as root to guide you through this step.

You'll be asked to provide the following information:

- *config location*. Replace with **/opt/bacula/etc/oci/config**
- *user OCID*.
- *tenancy OCID*.
- *region:*
- *RSA key pair*. The script can generate it for you or you can provide your own: https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm.

  Make sure to locate them in **/opt/bacula/etc/oci**.

Alternativelly, you can manually run

```
sudo -u bacula /usr/local/bin/oci setup config
```

Upload public pem key to the OCI Console

Next the public key must be uploaded to the OCI Console. The public key is named *oci_api_key_public.pem* and is located in **/opt/bacula/etc/oci**. View the details for the user who will be calling the API with the key pair: Click **Identity**, **Users**, and then select the user from the list. Click **Add Public Key**. Paste the contents of the PEM public key in the dialog box and click Add.

Retrieve the Bucket namespace and the compartment id

From the OCI Console, open the navigation menu in the top left corner. Select Object Storage, then the Bucket used for Bacula backup. In the Bucket Information, note the **namespace** and copy the **compartment id**.

Compartment permissions

It may be necessary to set advanced policies such as below in order to allow access to the bucket:

```
allow group GROUPNAME to manage buckets in compartment COMPARTMENTNAME

allow group GROUPNAME to use buckets in compartment COMPARTMENTNAME

allow group GROUPNAME to manage objects in compartment COMPARTMENTNAME
```

Create the resource control file

Edit the config file (should be */opt/bacula/etc/oci/config*) and add the following section

```
[CLI]
# globally scoped default for all operations with a -compartment-id parameter
compartment-id = <compartment-id>
# globally scoped default for all operations with a -namespace parameter
namespace = <namespace>

[OCI_CLI_SETTINGS]
default_profile=CLI
```

Replace <*compartment-id*> and <*namespace*> with the values you retrieved in *Retrieve the Bucket namespace and the compartment id*. Save the config file.

Test OCI CLI

In a terminal, type on the same line:

```
sudo -u bacula /usr/local/bin/oci os object list -bn MyBucket  --cli-rc-file /
↪opt/bacula/etc/oci/config --config-file /opt/bacula/etc/oci/config
```

Where *MyBucket* is the name of the bucket created in *Create a Bucket in Oracle Cloud*.

You should get no error message and the following reply:

```
"prefixes": []
```

Swift Account Management

Install the python-swiftclient Command Line Interface tool

Openstack provides a Python application that lets you access their Cloud Storage from the command line. It is part of the Swift Stack which installation is platform dependent as described here: https://pypi.org/project/python-swiftclient/ .

Choose your platform and follow the installation steps accordingly.

1.3 Tape

SAN Tape Shared Storage Option

Overview

This white paper presents various techniques and strategies for tape backup on SAN networks using the tape Shared Storage option (plugin) with **Bacula Enterprise**.

Scope

The current version of the Shared Storage plugin supports SCSI persistent reservation using SPC-3 on tape drives in Bacula Enterprise and later. The older form of SPC-2 SCSI reserve is not supported by **Bacula Enterprise**.

SAN Backup

A SAN network of SCSI devices has several advantages over a LAN (Ethernet) network.

Faster
Generally a SAN is faster than a LAN. SANs are typically 2Gb to 10Gb while LANs are commonly 1Gb (there are, of course, 10GB LANs).

Connectivity
In a SAN, each SCSI device appears to each of the hosts connected to the SAN (unless you use zoning or VLAN techniques). This has the advantage that a tape library (autochanger) can appear to be a directly connected SCSI device to two or more servers at the same time.

Backups
Since the SCSI devices (tapes) appear to be directly connected, backups are faster and do not involve sending data over the LAN (LAN-free-backup).

The problem with backing up multiple servers at the same time to the same tape library (or autoloader) is that if both servers access the same tape drive same time, you will very likely get data corruption. This is where the **Bacula Enterprise** shared storage plugin comes into play. The plugin ensures that only one server at a time can connect to each device (tape drive) by using the SPC-3 SCSI reservation protocol.

Using Shared Storage

To use the shared storage plugin with **Bacula Enterprise**, you must have the shared storage plugin (**shstore**) installed and configured, you must have a tape library (autochanger) that supports the SPC-3 protocol, and you must install a package named **sg3_utils** on either a RedHat or Ubuntu system.

The next few sections describes how to install, configure, and use the shared storage option.

Packages

First you must have a working Bacula with multiple Storage daemons installed, and on each of the machines with the Storage daemons, you must install the **shstore** plugin package.

A package is available for RedHat Enterprise and Ubuntu LTS. Contact Bacula System to get them. A version for Solaris may be available on request, but is not currently implemented.

```
# rpm -ivh bacula-enterprise-shstore_6.0.0.rpm
  or
# dpkg -i bacula-enterprise-shstore_6.0.0.deb
```

These packages will ensure that your **Bacula Enterprise** version is compatible with the shstore plugin and will install **shstore-sd.so** and **storage-ctl** programs.

```
/opt/bacula# ls plugins/shstore* scripts/storage*
-rwxr-x--- 1 bacula tape 60463 Dec 15 12:27 plugins/shstore-sd.so
-rwxr-x--- 1 bacula tape  7147 Dec 15 11:44 scripts/storage-ctl
-rwxr-x--- 1 bacula tape   589 Dec 15 11:44 scripts/storage-ctl.conf
```

It is also possible to install packages through your package managing system (**apt**, or **yum**).

Example:

```
% grep bacula /etc/apt/sources.list
deb https://www.baculasystems.com/dl/<customer>/debs/main/6.0.0/squeeze-64␣
↪squeeze main
deb https://www.baculasystems.com/dl/<customer>/debs/shstore/6.0.0/squeeze-64␣
↪squeeze shstore
```

If you have troubles to setup your package manager, feel free to contact Bacula Systems support team.

Director Configuration Version 6.0.1

This section only applies to Bacula Enterprise version 6.0.1. For later versions of Bacula Enterprise, this section no longer applies, please see the next section.

The **Shared Storage** directive should be used when using the SAN Shared Storage plugin or when accessing from the Director Storage resources directly to Devices of an Autochanger.

```
Director {
  Name = bacula-dir
  Shared Storage = yes   # Not used in version 6.0.2
...
}
```

When sharing volumes between different Storage resources, you will need to execute the **reset-storageid** script before using the **update slots** command. This script can be scheduled once a day in a CRON or Admin job.

```
$ /opt/bacula/scripts/reset-storageid MediaType StorageName
$ bconsole
* update slots storage=StorageName drive=0
```

The above is not used in version 6.0.2, please see below.

Director Configuration Version 6.0.2

The **Share Storage** directive is removed in Bacula Enterprise 6.0.2, and the **reset-storageid** script and **update slots** command described above are no longer needed in Bacula Version 6.0.2. However, to make Bacula function properly, you must adapt your **bacula-dir.conf Storage** directives.

Each autochanger that you have defined in an **Autochanger** resource in the Storage daemon's **bacula-sd.conf** file, must have a corresponding **Autochanger** resource defined in the Director's **bacula-dir.conf** file. Normally you will already have a **Storage** resource that points to the Storage daemon's **Autochanger** resource. Thus you need only to change the name of the **Storage** resource to **Autochanger**. In addition no **Autochanger = yes** directive is needed in the Director's **Autochanger** resource.

In addition to the above change (**Storage** to **Autochanger**), you must modify any additional **Storage** resources that correspond to devices that are part of the **Autochanger** device. Instead of the previous **Autochanger = yes** directive they should be modified to be **Autochanger = xxx** where you replace the **xxx** with the name of the Autochanger.

For example, in the bacula-dir.conf file:

```
Autochanger {             # New resource
  Name = Changer-1
  Address = cibou.company.com
  SDPort = 9103
  Password = "xxxxxxxxxx"
  Device = LTO-Changer-1
  Media Type = LTO-9
  Maximum Concurrent Jobs = 50
}

Storage {
```

(continues on next page)

(continued from previous page)

```
  Name = Changer-1-Drive0
  Address = cibou.company.com
  SDPort = 9103
  Password = "xxxxxxxxxx"
  Device = LTO9_1_Drive0
  Media Type = LTO-9
  Autochanger = Changer-1  # New directive
  Maximum Concurrent Jobs = 5
}

Storage {
  Name = Changer-1-Drive1
  Address = cibou.company.com
  SDPort = 9103
  Password = "xxxxxxxxxx"
  Device = LTO9_1_Drive1
  Media Type = LTO-9
  Autochanger = Changer-1  # New directive
  Maximum Concurrent Jobs = 5
}

...
```

Note that Storage resources **Changer-1-Drive0** and **Changer-1-Drive1** are not required since they make up part of an autochanger. However, by referring to those Storage definitions in a job, you will be sure to use only the indicated drive. This is probably most used for reserving a drive for restores. See the Storage daemon example .conf below and the use of **AutoSelect = no**.

So, in summary, the changes are:

- Remove the **Shared Storage = yes** from the **Director** resource.
- Change **Storage** to **Autochanger** in the LTO9 resource.
- Remove the **Autochanger = yes** from the **Autochanger** LTO9 resource.
- Change the **Autochanger = yes** in each of the **Storage** device that belong to the **Autochanger** to point to the **Autochanger** resource with for the example above the directive **Autochanger = LTO9**.

Sharing an Autochanger

If you have a second Storage daemon that shares an Autochanger with the your first Storage daemon defined above, you define it much the same way as above, with of course different names, but in addition, you must tell the Director that this second Autochanger is shared with the first one. You do so, by adding a new directive: **Shared Storage = <autochanger-name>**.

For example, in the bacula-dir.conf file:

```
Autochanger {                # New resource
  Name = Changer-2
  Address = dir.company.com
  SDPort = 9103
  Password = "yyyyyyy"
```

(continues on next page)

(continued from previous page)

```
  Device = LTO-Changer-2
  Media Type = LTO-9
  Maximum Concurrent Jobs = 50
  Shared Storage = Changer-1  # New directive
}

Storage {
  Name = Changer-2-Drive0
  Address = dir.company.com
  SDPort = 9103
  Password = "yyyyyyy"
  Device = LTO9_2_0
  Media Type = LTO-9
  Autochanger = Changer-2     # New directive
  Maximum Concurrent Jobs = 5
}

Storage {
  Name = Changer-2-Drive1
  Address = dir.company.com
  SDPort = 9103
  Password = "yyyyyyy"
  Device = LTO9_2_1
  Media Type = LTO-9
  Autochanger = Changer-2     # New directive
  Maximum Concurrent Jobs = 5
}
...
```

Storage Configuration

The **Plugin Directory** option in the **Storage daemon** resource must point where the **shstore-sd.so** plugin is installed. Generally **/opt/bacula/plugins**

```
Storage {
   Name = bacula-sd
   Plugin Directory = /opt/bacula/plugins
...
}
```

New Device Directives

There are two new directives in the **bacula-sd.conf** Device resource:

Control Device = <device-name>
: The control device is the SCSI control device that corresponds to the **Archive Device**. See below for details.

Lock Command = <command>
: This is the command that must be executed to effect the SCSI persistent locking. Generally, it will be the following:

```
Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
```

Note, the edit codes above are percent small-el percent small-oh.

Your Storage daemons should have an appropriate HBA card that permits direct connections to a SAN. You can examine what SCSI devices are attached with the **lsscsi** command on the first Storage daemon machine **cibou** as follows:

```
/opt/bacula# lsscsi -g
[1:0:0:0] tape    HP       Ultrium 4-SCSI   H61W  /dev/st0  /dev/sg0
[1:0:0:1] tape    HP       Ultrium 4-SCSI   H61W  /dev/st1  /dev/sg1
[1:0:0:2] mediumx HP       MSL G3 Series    E.00  -         /dev/sg2
[1:0:1:0] mediumx NETAPP   VTL              0001  -         /dev/sg3
[1:0:1:1] tape    QUANTUM  DLT7000          022C  /dev/st2  /dev/sg4
[1:0:1:2] tape    QUANTUM  DLT7000          022C  /dev/st3  /dev/sg5
[2:0:0:0] disk    ATA      ST3250824AS      3.AD  /dev/sda  /dev/sg6
[3:0:0:0] cd/dvd  TSSTcorp DVD+-RW TS-H553A DE04  /dev/scd0  /dev/sg7
```

The above machine, **cibou**, is rather simple because it has only a single disk drive, with two autochangers, the first autochanger is an HP LTO-9 with two drives, and the second a NETAPP VTL with two DLT7000 drives.

Then on a different machine named **dir**, you may get something like the following:

```
/opt/bacula# lsscsi -g
[0:0:0:0] cd/dvd  HL-DT-ST DVD-ROM GDR8163B 0B26  /dev/sr0  /dev/sg0
[2:0:0:0] disk    ATA      Hitachi HDS72302 MN6O  /dev/sda  /dev/sg1
[3:0:0:0] disk    ATA      Hitachi HDS72302 MN6O  /dev/sdb  /dev/sg2
[3:0:1:0] disk    ATA      Hitachi HDS72302 MN6O  /dev/sdc  /dev/sg3
[4:0:0:0] tape    HP       Ultrium 4-SCSI   H61W  /dev/st0  /dev/sg4
[4:0:0:1] tape    HP       Ultrium 4-SCSI   H61W  /dev/st1  /dev/sg5
[4:0:0:2] mediumx HP       MSL G3 Series    E.00  /dev/sch0  /dev/sg6
[4:0:1:0] mediumx NETAPP   VTL              0001  /dev/sch1  /dev/sg7
[4:0:1:1] tape    QUANTUM  DLT7000          022C  /dev/st2  /dev/sg8
[4:0:1:2] tape    QUANTUM  DLT7000          022C  /dev/st3  /dev/sg9
```

The above machine, **dir**, has three disk drives, then the same two autochangers as on the **cibou** machine, the first an HP LTO-9 with two drives, and the second a NETAPP VTL with two DLT7000 drives. Note, the HP autochanger and the NETAPP VTL are the same devices, but they appear slightly differently on each machine, because the SCSI device names are quite different.

Configuring the Devices

Now, let's construct Bacula Device resources in **bacula-sd.conf** for each of these machines starting with **cibou**.

```
Device {
  Name = tape
  Media Type = tape
  AutomaticMount = yes;
  RemovableMedia = yes;
  Archive Device = /dev/nst0
```

(continues on next page)

(continued from previous page)

```
  AlwaysOpen = no;
  Control Device = /dev/sg0
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}
```

There are three things that are different from a standard tape **Device** resource:

- There is a new **Control Device** directive.
- There is a new **Lock Command** directive.
- We have set **AlwaysOpen** to **no** rather than **yes**.

If you examine the first **lsscsi** output above, you will see that the tape device **Ultrium 4-SCSI** has an archive address of **/dev/st0** while the SCSI control channel is **/dev/sg0**.

For the equivalent for machine **dir**, the second **lsscsi** example shown above, would be:

```
Device {
  Name = tape
  Media Type = tape
  AutomaticMount = yes;
  RemovableMedia = yes;
  Archive Device = /dev/nst0

  AlwaysOpen = no;
  Control Device = /dev/sg4
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}
```

Note that only the SCSI control channel address has changed.

You might ask: Why do we set **AlwaysOpen = no**. The answer is that when Bacula opens the device, it will be locked thus preventing any other machines (Storage daemons) from accessing it. If we have **AlwaysOpen = yes** the device will be locked as long as the Storage daemon is active. When **AlwaysOpen** is set to **no** the device will be closed when Bacula no longer uses the device and in closing the device, Bacula will also release the lock thus permitting other machines to access the device.

Complete Storage daemon configuration

A more complete Storage daemon configuration file **bacula-sd.conf** that corresponds to the example of the two autochanger configuration for two Storage daemons given above might be:

```
# Storage definition in bacula-sd.conf on machine cibou.company.com
Storage {
  Name = xxx
  ...
}

Autochanger {
  Name = LTO-Changer-1
  Changer Device = /dev/sg2
  Changer Command ="/opt/bacula/scripts/mtx-changer %c %o %S %a %d"
  Device = LTO9_1_Drive0, LTO9_1_Drive1
```

(continues on next page)

(continued from previous page)

```
}


Device {
  Name = LTO9_1_Drive0
  Media Type = LTO-9
  Archive Device = /dev/nst0
  AutomaticMount = yes
  Autochanger = yes
  Drive Index = 0
  RemovableMedia = yes
  Spool Directory =  /opt/bacula/working
  Device Type = Tape

  AlwaysOpen = no
  Control Device = /dev/sg0
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}

Device {
  Name = LTO9_1_Drive1
  Media Type = LTO-9
  Archive Device = /dev/nst1
  AutomaticMount = yes
  Autochanger = yes
  Drive Index = 0
  RemovableMedia = yes
  Spool Directory =  /opt/bacula/working
  Device Type = Tape
  AutoSelect = no    # reserved for restores

  AlwaysOpen = no
  Control Device = /dev/sg1
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}
```

and for the second Storage daemon that uses the same autochanger on a SAN network:

```
# Storage definition in bacula-sd.conf on machine dir.company.com
Storage {
  Name = yyy
  ...
}

Autochanger {
  Name = LTO-Changer-2
  Changer Device = /dev/sg6
  Changer Command ="/opt/bacula/scripts/mtx-changer %c %o %S %a %d"
  Device = LTO9_2_Drive0, LTO9_2_Drive1
}


```

(continues on next page)

(continued from previous page)

```
Device {
  Name = LTO9_2_Drive0
  Media Type = LTO-9
  Archive Device = /dev/nst0
  AutomaticMount = yes
  Autochanger = yes
  Drive Index = 0
  RemovableMedia = yes
  Spool Directory =  /opt/bacula/working
  Device Type = Tape

  AlwaysOpen = no
  Control Device = /dev/sg4
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}

Device {
  Name = LTO9_2_Drive1
  Media Type = LTO-9
  Archive Device = /dev/nst1
  AutomaticMount = yes
  Autochanger = yes
  Drive Index = 1
  RemovableMedia = yes
  Spool Directory =  /opt/bacula/working
  Device Type = Tape
  AutoSelect = no    # reserved for restores

  AlwaysOpen = no
  Control Device = /dev/sg5
  Lock Command = "/opt/bacula/scripts/storage-ctl %l %o"
}
```

Configuring the storage-ctl Script

Normally, the file **storage-ctl** found in the Bacula scripts directory should never be changed. On the other hand, it must be configured with the **storage-ctl.conf** file. The default file looks like:

```
#
# This file is sourced by the storage-ctl script every time it runs.
#   You can put your site customization here, and when you do an
#   upgrade, the process should not modify this file.  Thus you
#   preserve your storage-ctl configuration.
#

#
# The following key must be a unique 6 hex digit value on each SD
#
key=bac001

```

(continues on next page)

(continued from previous page)

```
# Set to 1 if you want debug info written to a log
debug_log=0

# Set to path to your sg_persist program
SG_PERSIST=/usr/bin/sg_persist

# Set to the maximum number of seconds to wait for a scsi lock
#  Default = 30*60 = 1800 or 30 minutes
max_lock_wait=1800
```

However, for operational reasons, you will certainly want to change they key (first configuration item) to be different (and unique) on each machine. This key is used by the reservation system, and it is far better to have unique keys on each Storage daemon so that in Bacula status command and in debug output it will be clear which machine holds what reservation.

The key is a 6 digit hexadecimal value (i.e. 0-9 and a-f), where the letters may be either uppercase or lowercase. If you enter a character that is not a valid hexadecimal character or more than 6 characters, the locking will fail, and thus your jobs will fail.

For the examples in this white paper that follow, we have set the **key** to **bac001** on the system named **cibou** and set it to **bac999** on the system named **dir**.

Now assume that machine **dir** has locked the device. If on the machine **cibou**, we issue the following command:

```
sg_persist -r /dev/sg0
```

we will get the following output:

```
HP        Ultrium 4-SCSI    H61W
Peripheral device type: tape
PR generation=0x468, Reservation follows:
  Key=0xbac999
  scope: LU_SCOPE,  type: Exclusive Access
```

This indicates that the device is Exclusively locked by the machine that used the key **bac999**, which we know is our machine named **dir**.

Status Command

When running a job on machine **cibou**, if we run a job that wants a tape device that is locked by another machine **dir**, we might get output that looks similar to:

```
Jobs waiting to reserve a drive:
   3612 JobId=2 waiting because device "Drive-0" (/dev/nst0)
   is reserved by: 0xbac999.
====

Device status:
Autochanger "tape" with devices:
   "Drive-0" (/dev/nst0)
Device "Drive-0" (/dev/nst0) is mounted with:
    Volume:      TestVolume001
```

(continues on next page)

(continued from previous page)

```
    Pool:        Default
    Media type:  tape
    Device is BLOCKED by another SD=0xbac999
    Total Bytes Read=64,512 Blocks Read=1 Bytes/block=64,512
    Positioned at File=0 Block=0
    shstore=1 registered=1 locked=0 blockedbySD=0xbac999
```

The important point here is that our job (JobId=2) is BLOCKED by another SD with the key **0xbac999** (i.e. **dir**). We know that shared storage is working and that our Storage daemon was able to register our key because of the last line of the output.

Once **dir** finishes with the drive and releases the lock, our job will continue and the status output will change to:

```
Jobs waiting to reserve a drive:
====

Device status:
Autochanger "tape" with devices:
   "Drive-0" (/dev/nst0)
Device "Drive-0" (/dev/nst0) is mounted with:
    Volume:      TestVolume001
    Pool:        Default
    Media type:  tape
    Total Bytes=48,900,096 Blocks=757 Bytes/block=64,597
    Positioned at File=1 Block=0
    shstore=1 registered=1 locked=1 blockedbySD=no
```

and now the job is no longer blocked and you can see that the last line of the output indicates that this storage daemon has the device locked (**locked=1**).

Problem Resolution

In general if any component of Bacula crashes, particularly the Storage daemon, any lock that it holds on any and all devices will be released. However, under unusual circumstances (a kill -9) or the OS crashes, the lock may remain.

In your Bacula binary directory (normally /opt/bacula/scripts) you can manually run the storage-ctl script with:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 query
```

If a lock is set on the device, you may forcibly remove it by using:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 clear
```

Please note that all no program should be actively accessing the device during a **clear** and that any lock that may exist will be forcibly be removed. The program that held the lock will not know the lock is removed and will blindly continue using the device without a lock, then if another program starts using device, it will acquire a new lock, but data corruption will likely occur because two programs will be writing to the same device at the same time (the first one with a broken lock, and the second with the new lock).

Prior to running the above **clear**, you might want to see who has what on the device, which can be done with the **regkeys** and **wholocked** commands:

In your Bacula binary directory (normally /opt/bacula/scripts) you can manually run the storage-ctl script with:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 regkeys
0xbac999
0xbac001

./storage-ctl /dev/sg0 wholocked
0xbac001
```

In this case, we see that **bac001** has the device locked, but both **bac001** and **bac999** have registration keys. Thus before trying to break the lock, it might be wise to find out who **bac999** is and why there is still a registration. Since after the **clear**, all locks and all registrations are lost.

Getting Information on the Locks

You can get a lot of information using the Bacula **status** command about which job is blocking a drive, as seen in examples in a previous section.

However, in some cases, it is easier to examine the overall picture by manually running the storage-ctl script which can normally be found in /opt/bacula/scripts.

Now if we examine from the stand point of the machine **dir** when there are no locks or registrations, using the query command, we get output as follows:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg4 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x493
  No full status descriptors
```

If we now manually register a key (defined in **storage-ctl.conf**, then do a query, we will get the following output:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg4 register
./storage-ctl /dev/sg4 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x494
    Key=0xbac999
      All target ports bit clear
      Relative port address: 0x1
      not reservation holder
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 00 00 1b 32 1f 43 65    !...2.Ce
```

Note that this time, it shows a key (bac999), the fact that there is no lock (no reservation holder), and the unique identification of the machine that registered the key (Transport Id of initiator).

Now let's actually lock the device:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg4 lock
./storage-ctl /dev/sg4 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x494
    Key=0xbac999
      All target ports bit clear
      Relative port address: 0x1
      << Reservation holder >>
      scope: LU_SCOPE,  type: Exclusive Access
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 00 00 1b 32 1f 43 65    !...2.Ce
```

This time the query shows us that we (bac999) is the reservation holder (we have it locked) for Exclusive Access.

If we now switch to another machine, **cibou**, and issue the same query (on a different device address, but for the same physical device), we get:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x494
    Key=0xbac999
      All target ports bit clear
      Relative port address: 0x1
      << Reservation holder >>
      scope: LU_SCOPE,  type: Exclusive Access
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 00 00 1b 32 1f 43 65    !...2.Ce
```

This is the same as what we see from machine **dir**, which means everything is consistent.

Now, let's register a key on the second machine, **cibou**:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 register
./storage-ctl /dev/sg0 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x495
    Key=0xbac999
      All target ports bit clear
      Relative port address: 0x1
      << Reservation holder >>
      scope: LU_SCOPE,  type: Exclusive Access
```

(continues on next page)

(continued from previous page)

```
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 00 00 1b 32 1f 43 65   !...2.Ce

    Key=0xbac001
      All target ports bit clear
      Relative port address: 0x1
      not reservation holder
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 01 00 1b 32 21 73 ee   !...2!s.
```

This time the query returns us information about two different "initiators" one, **dir**, which registered bac999 and holds the lock (Reservation holder), and the other **cibou**, which has a key (bac001) registered. Note that the two have different unique transport id names (the last line of each section that begins with 00).

If we try to obtain a lock on **cibou** we get the following error:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 trylock
persistent reserve out: scsi status: Reservation Conflict
PR out: command failed
```

Now if we remove the lock on **dir** (not shown), then acquire the lock on **cibou** and run another query, the output is as expected:

```
cd /opt/bacula/scripts
./storage-ctl /dev/sg0 lock
./storage-ctl /dev/sg0 query
  HP        Ultrium 4-SCSI    H61W
  Peripheral device type: tape
  PR generation=0x495
    Key=0xbac999
      All target ports bit clear
      Relative port address: 0x1
      not reservation holder
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 00 00 1b 32 1f 43 65    !...2.Ce

    Key=0xbac001
      All target ports bit clear
      Relative port address: 0x1
      << Reservation holder >>
      scope: LU_SCOPE,  type: Exclusive Access
      Transport Id of initiator:
        FCP-2 World Wide Name:
 00     21 01 00 1b 32 21 73 ee    !...2!s.
```

It shows the two machines (initiators), but this time **cibou** with key bac001 holds the Exclusive Access lock.

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

ACSLS Backup

Overview

Executive summary

This document is intended to provide insight into the considerations and processes required to integrate **Oracle ACSLS** with **Bacula Enterprise**.

Features Summary

- Provide all tape library management operations required by **Bacula Enterprise**.
- Support named user access to ACSLM if required.
- Support tape drive and volume locking in shared **ACSLS** environment.
- Support lock query and management.
- Static tape drive location mapping.
- Dynamic volume location mapping.

Theory of Operations

ACSLS Advantages

ACSLS offers many advantages when managing a tape library environment:

- Processes multiple requests in parallel and optimizes use of large library complexes.
- Avoids delays caused by pass-thru between robots.
- Automatically recovers and retries requests that fail.

- Allows multiple clients to share a library.
- Simplifies support of new libraries and library features.
- Provides a choice of interfaces.
- Presents logical libraries through a SCSI media changer interface over fibre channel.
- Changes library configurations while libraries remain online.
- Manages all libraries at a customer site from **ACSLS**.
- Provides a high availability option.

A major feature of **Oracle ACSLS** is the ability to manage any combination of tape libraries. This provides access to the latest ACS technology and to applications across libraries.

General ACSLS Operation

The **ACSLS** acts as a library management system. **ACSLS** manages the physical aspects of tape cartridge storage and retrieval through a system administrator interface and a programmatic interface. These real–time interfaces control and monitor tape libraries, including access control. **ACSLS** does not have access to the data path through which information is read from or written to tape cartridges. **ACSLS** operates only on the control path through which libraries are managed.

The **ACSLS** consists of the following:

- One or more tape libraries connected with pass-thru ports.
- Library(ies) that contains tape cartridges uniquely identified by an external label with a volume serial number (VOLSER) and a media domain and type.
- A set of tape drives (sometimes referred to as transports) in the library or libraries.
- A set of cartridge access ports (CAPs) attached to the library which allow cartridges to be physically entered or removed from the library.
- Robotics that physically move cartridges between library storage cells, cartridge drives, CAPs, and pass-thru ports.

Library requests originate from client applications running on a host system. These requests and messages then move across a network connecting client systems and the ACSLS. An example of a client application is Bacula Enterprise backup and restore software.

Bacula Enterprise Integration

Bacula Enterprise ACSLS allows management of ACS using ACSAPI for all required operations. The integration consist of three main components:

- `acsls-changer` - The application which interacts between Bacula Storage Daemon and makes requests to the tape library system and ACSLM system.
- `ssi` - The SSI component of **ACSLS** provided by **Bacula Enterprise** which could be used as a replacement for the standard SSI component distributed by **ACSLS** software vendor.
- `ssi_el` - The SSI Event Logger component required for saving event and trace logs generated by the SSI component provided by **Bacula Enterprise**.

ACSLS Changer

Bacula Enterprise's `acsls-changer` is a direct replacement for the `mtx-changer` script which manages tape libraries connected to ACSLM. You can use its services directly by manually inputing required commands and parameters or indirectly by Bacula Storage Daemon configuration and standard `bconsole` commands.

Installation

The first step is to install and configure the Bacula Storage Daemon on the backup machine which will manage tape libraries using ACSLM and **ACSLS** integration. Next, install the corresponding ACSLS Plugin package on this Storage Daemon.

Installation of the Plugin

On the Bacula Storage Daemon that you want to connect to your ACSLM, extend the repository file for your package manager to contain a section for the plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0

[BaculaACSLS]
name=Bacula Enterprise ACSLS Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/acsls/
↪@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

Then perform a yum update, and after that the package *bacula-enterprise-acsls* can be installed with `yum install`. If you prefer to manually install the packages, you can also download them from your download area, and use the low level package manager tool (`rpm`) to do the plugin installation.

Configuration

After installation of the **Bacula Enterprise ACSLS**, it should be configured to meet your backup environment requirements, including the required `acsls-changer` configuration and optional `SSI` component configuration.

Changer Configuration

The plugin is configured using a dedicated configuration file `acsls-changer.conf` located in the */opt/bacula/etc* directory, and a proper Bacula Storage Daemon resource configuration.

Changer Config File

The `acsls-changer.conf` file is a simple `key=value` configuration file with the following parameters:

- `ACSAPI_PACKET_VERSION=<nr>` specifies the ACSLM supported packet version (default 4). It should match the corresponding SSI parameter. This parameter is optional. If not set, the default will be used.
- `ACSAPI_SSI_SOCKET=<port>` specifies the socket for SSI interprocess communication (default 50004). This parameter is optional. If not set, the default will be used.
- `VOLDBT=<path/file>` specifies the `acsls-changer` working file which stores the volumes-to-slot mappings. The `path/file` value should be set to `/opt/bacula/working/acsls-changer.dbt`. This parameter is optional. If not set, a file named `acsls-changer.dbt` in the current working directory will be used.
- `USERID` specifies the default ACSAPI user. This parameter is optional. If not set, no user information will be sent to ACSLM.
- `LOCKING=<yes|no>` specifies if `acsls-changer` should use the resource (`Drive` and `Volume`) locking mechanism. This parameter is optional. If not set, no resource locking will be used. See: *Resource locking*.
- `DRIVE<N>=<acs:lsm:panel:drive>` specifies a static mapping between a Bacula parameter configured in a Bacula Storage Daemon resource configuration and ACSLS DRIVEID. Every tape drive defined in a `bacula-sd.conf` resource which is managed by **ACSLS** should be defined as an appropriate parameter in the config file. This parameter is required and should be defined at least once. See: *DRIVEID mapping*.
- `TIMEOUT=<secs>` specifies how long in seconds the `acsls-changer` should wait for a request to complete (default is 600 seconds). A value of zero means no timeout (infinite wait). The `TIMEOUT` value should not exceed the `Maximum Changer Wait` time configured in the Bacula Storage Daemon `bacula-sd.conf` file. See: *Request timeout*.

Here is an `acsls-changer.conf` file example:

```
ACSAPI_PACKET_VERSION = 4
ACSAPI_SSI_SOCKET = 50004
USERID = root
LOCKING = yes
VOLDBT = /opt/bacula/working/acsls-changer.dbt
DRIVE0 = 1:2:3:4
DRIVE1 = 1:2:3:5
```

Storage Daemon Configuration

To use an **ACSLS** storage in **Bacula** you should configure `Autochanger/Device` resources in the same way as for a standard tape library or tape autochanger managed by the `mtx` utility. The only exception is an updated `Changer Command` as in this example:

```
Changer Command = "/opt/bacula/bin/acsls-changer %o %d %S"
```

Where:

- **%o** is a changer command (load, unload, etc)
- **%d** is a index number from the Drive Index parameter of a Device resource
- **%S** is a slot number (started from 1)

Here is an example and resource configuration to use with **ACSLS**.

```
#
# An ACSLS system with two drives
#
Autochanger {
   Name = SL8500-ACSLS
   Device = IBMLTO7-1
   Device = IBMLTO7-2
   # %o=command, %d=drive-index, %S=slot(base1)
   Changer Command = "/opt/bacula/bin/acsls-changer %o %d %S"
   Changer Device = /dev/null
}
Device {
   Name = IBMLTO7-1
   DriveIndex = 0
   DeviceType = Tape
   MediaType = LTO7
   ArchiveDevice = /dev/tape/by-id/scsi-3500104f000899ece-nst
   AutomaticMount = no
   AlwaysOpen = no
   RemovableMedia = yes
   RandomAccess = no
   AutoChanger = yes
   OfflineOnUnmount = yes
}
Device {
   Name = IBMLTO7-2
   DriveIndex = 1
   DeviceType = Tape
   MediaType = LTO7
   ArchiveDevice = /dev/tape/by-id/scsi-3500104f000899ed1-nst
   AutomaticMount = no
   AlwaysOpen = no
   RemovableMedia = yes
   RandomAccess = no
   AutoChanger = yes
   OfflineOnUnmount = yes
}
```

Note: On some systems and tape libraries managed by **ACSLS** it might be required to ummount the tape drive by ejecting the tape after use (unmount). This can help **ACSLS** to allow the tape robot to grab the tape from the drive. Due to the above, the following parameter should be defined in the Bacula Storage Daemon configuration for every drive device:

```
OfflineOnUnmount = yes
```

This configuration setting is present in the example above.

SSI Component Configuration

You may use the SSI component provided by your **ACSLS** vendor or use the SSI component from the **Bacula Enterprise ACSLS** plugin. To use SSI component from the **Bacula Enterprise ACSLS** plugin you should prepare an SSI config file as described in the next section.

SSI Config file

The `acsls-ssi.conf` file is a simple `key=value` configuration with the following parameters:

- `CSI_HOSTNAME=<address>` defines the `CSI` component location address. Acceptable values are an IP address or a fully qualified domain name for the host. This parameter is required.
- `CSI_TCP_RPCSERVICE=<TRUE/FALSE>` is used to define whether the `CSI` operates as a TCP RPC Server (default TRUE). This variable must be set to TRUE for the firewall-secure CSC. The firewall-secure **ACSLS** application packets are all sent using the TCP network transport. This parameter is optional. If not set, the default will be used.
- `CSI_UDP_RPCSERVICE=<TRUE/FALSE>` is used to define whether the `CSI` operates as a UDP RPC server (default TRUE). This variable must be set to FALSE for the firewall-secure CSC. The firewall-secure **ACSLS** application packets are all sent using the TCP network transport. The CSI can operate as a TCP and a UDP server simultaneously. This parameter is optional. If not set, the default will be used.
- `CSI_CONNECT_AGETIME=<nr>` defines the value of the maximum age of pending requests in the `CSI`'s request queue. This variable is expressed as an integer representing a number of seconds. A value of 172800 indicates two days which is the default. Messages older than this value are removed from the queue and the `CSI` sends an entry to the Event Log when this happens. This parameter is optional. If not set, the default will be used.
- `CSI_RETRY_TIMEOUT=<nr>` defines the minimum amount of time, in seconds, that the `CSI` will wait between attempts to establish a network connection (default 4). This parameter is optional. If not set, the default will be used.
- `CSI_RETRY_TRIES=<nr>` defines the number of attempts the `CSI` will make to transmit a message (default 5). Pending messages are discarded if a connection cannot be established within the defined number of tries. The `CSI_RETRY_TIMEOUT` and `CSI_RETRY_TRIES` together determine the minimum total time the `CSI` will attempt to send a message. This parameter is optional. If not set, the default will be used.
- `ACSAPI_SSI_SOCKET=<port>` is the socket port number (default 50004) on which the SSI component listens to incoming requests from `acsls-changer`. This parameter should be the same as `ACSAPI_SSI_SOCKET` defined in the `acsls-changer.conf` file. This parameter is optional. If not set, the default will be used.

Here is an `acsls-ssi.conf` file example:

```
CSI_HOSTNAME=10.10.10.1
ACSAPI_SSI_SOCKET=50009
CSI_UDP_RPCSERVICE=FALSE
```

DRIVEID mapping

To configure drive mapping in the `acsls-changer.conf` file you should use tape drive serial number matching. To obtain the serial numbers from the **ACSLS** managed tape library drives, execute the following commands:

```
root@saxo:~# su - acsss
bash-4.1# cd bin
bash-4.1# ./cmd_proc
-------Oracle ACSLS (Automated Cartridge System Library Software) 8.4.0-3-----
↪-

ACSSA> display drive * -f serial_num
2018-03-12 05:07:17         Display Drive
Acs  Lsm  Panel  Drive  Serial_num
2    0    1      11     1013001383
2    0    1      15     1013001384
```

In the example above you can see two drives and their corresponding serial numbers. Then check the serial number of all drives attached to your Bacula Enterprise Storage server using the `tapeinfo` command.

To use this command you will need SCSI generic devices (eg: /dev/sgX) for all of your tape drives:

```
# lsscsi -g
(...)
[8:0:7:0]    tape    IBM      ULTRIUM-TD7      G9Q2  /dev/st0   /dev/sg6
[8:0:8:0]    tape    IBM      ULTRIUM-TD7      G9Q2  /dev/st2   /dev/sg8
```

Then, for each SCSI generic device (/dev/sgX) listed for your tape drives, get its serial number. For example:

```
# tapeinfo -f /dev/sg6
Product Type: Tape Drive
(...)
SerialNumber: '1013001383'
(...)

# tapeinfo -f /dev/sg8
Product Type: Tape Drive
(...)
SerialNumber: '1013001384'
(...)
```

Then match all available tape drive serial numbers in your storage server to the serial numbers from **ACSLS**. You should use `Acs`, `Lsm`, `Panel`, `Drive` to define a drive mapping in the acsls-changer.conf file as in the above example.

The acsls-changer.conf drive mapping for the above example would be configured as follows:

```
(...)
DRIVE0 = 2:0:1:11
DRIVE1 = 2:0:1:15
```

and the corresponding resources in bacula-sd.conf:

```
Device {
   Name = TapeDrive0
   Drive Index = 0
   Archive Device = /dev/nst0
(...)
}
Device {
   Name = TapeDrive1
   Drive Index = 1
   Archive Device = /dev/nst2
(...)
}
```

Operations

This chapter describes runtime operations for **ACSLS** management and testing with **Bacula Enterprise** integration.

After editing the acsls-changer.conf and acsls-ssi.conf files, the /opt/bacula/scripts/acsls-start.sh must be run to start the Bacula ssi component (replacement for the standard one from ACSLS vendor), and ssi_el (event logger) processes.

```
# /opt/bacula/scripts/acsls-start.sh
```

Testing ACSLS Integration

Before attempting to use the **ACSLS** integration with **Bacula**, it is preferable to manually test that the integration works. To do so, we suggest you perform the following steps:

listall command

This command should list all defined drives and available volumes. The command can take some time to complete depending on the number of available volumes in ACSLM. If an error is printed instead of a list please contact Bacula Systems support.

```
# /opt/bacula/bin/acsls-changer listall
D:0:E
D:1:F:2:F70170
S:1:F:F70168
```

(continues on next page)

(continued from previous page)

```
S:2:E
S:3:F:F71070
S:4:F:F71071
S:5:F:F71072
S:6:F:F71073
S:7:F:F71074
S:8:F:F71075
S:9:F:F71076
S:10:F:F71077
S:11:F:F71078
```

load command

This command should load the volume from the specified slot into the specified drive. If an error is printed instead of a 'done', the error message may provide enough information for easy troubleshooting. If not, please contact Bacula Systems support.

```
# /opt/bacula/bin/acsls-changer load 0 5
Loading Volume F71072 into drive 0 ... done
```

loaded command

This command allows you to check if a volume is loaded into a specified drive. The number printed is a volume slot loaded into the drive. If the value printed is zero, then no volume is loaded in the specified drive. As above, if an error is printed instead of a slot number, you can attempt to resolve the problem printed in the error message or ask for support.

```
# /opt/bacula/bin/acsls-changer loaded 0
5
```

unload command

This command allows you to unload the volume from the selected drive. If an error is printed instead of 'done' you can attempt to resolve the problem printed in the error message or ask for support.

```
# /opt/bacula/bin/acsls-changer unload 1
Unloading Volume F70170 from drive 1 ... done
```

Other commands

The main purpose for the `acsls-changer` command is to allow a Bacula Enterprise Storage Daemon to manage ACSLS tape libraries, but the command was also designed to allow management operations to be performed manually.

```
# /opt/bacula/bin/acsls-changer -?
Copyright (C) 2000-2017 Kern Sibbald.

Version: 9.0.6 (20 November 2017)

Usage: acsls-changer [options] <command> [<drive-index> <slot> <volumename>]
-d <nn>           set debug level to <nn>
-c <file>         acsls-changer config file
-b <file>         acsls-changer volumes database file
-?                print this message
```

The utility supports the following commands:

- `list` - prints the list of all available volumes and slots.
- `listall` - prints the list of all defined drives and all available volumes and slots. See *listall command*.
- `load <driveindex> <slot> [volume]` - loads a volume specified by a `slot` or `volume` into a selected tape drive specified by a `<driveindex>`. If you specify a `<volume>` parameter then the slot number will be ignored. You can use the `<volume>` parameter on manual operations only. This kind of operation is not supported by Bacula Storage Daemon. See: *load command*.
- `unload <driveindex>` - unloads a volume from a tape drive specified by a `<driveindex>`. See: *unload command*.
- `slots` - prints a number of all available slots including free slots.
- `loaded <driveindex>` - prints a volume slot number loaded into a drive. See: *loaded command*.
- `showdrvlock <driveindex>` - prints information about a tape drive status and lock. See: *showdrvlock command*.
- `showvollock <volume>` - prints information about a volume status and lock. See: *showvollock command*.
- `cleardrvlock <driveindex>` - clears a lock assigned to the specified drive. See: *cleardrvlock command*.
- `clearvollock <volume>` - clears a lock assigned to the specified volume name. See: *clearvollock command*.
- `showvollockall` - prints lock information about all available volumes. See: *showvollockall command*.

showdrvlock command

To print the lock information about a specific tape drive, use the following command example:

```
# /opt/bacula/bin/acsls-changer showdrvlock 0
ACS_QUERY_LOCK_DRV_RESPONSE: STATUS_SUCCESS (0)

DRV: [2:0:1:15]
    lock:14509 userid: regress
    duration:67 pending: 0
    status STATUS_DRIVE_IN_USE (29)
```

When no lock is assigned to the specified tape drive no lock information will be printed.

showvollock command

To print the lock information about a specific volume, use the following command example:

```
# /opt/bacula/bin/acsls-changer showvollock F71072
ACS_QUERY_LOCK_VOL_RESPONSE: STATUS_SUCCESS (0)

VOL: F71072
    lockid:14509 userid: regress
    duration:92 pending: 0
    status STATUS_VOLUME_IN_USE (99)
```

When no lock is assigned to the specified volume no lock information will be printed.

cleardrvlock command

To clear the lock assigned to a tape drive, use the following command example:

```
# /opt/bacula/bin/acsls-changer cleardrvlock 0
ACS_CLEAR_LOCK_DRV_RESPONSE: STATUS_SUCCESS (0)

DRV: [2:0:1:15]
    status STATUS_SUCCESS (0)
```

When no lock is assigned then `STATUS_DRIVE_AVAILABLE` will be returned.

clearvollock command

To clear the lock assigned to a volume, use the following command example:

```
# /opt/bacula/bin/acsls-changer clearvollock F71072
ACS_CLEAR_LOCK_VOL_RESPONSE: STATUS_SUCCESS (0)

VOL: F71072
    status STATUS_SUCCESS (0)
```

When no lock is assigned then `STATUS_VOLUME_AVAILABLE` will be returned.

showvollockall command

To print lock information for all available volumes use the following command example:

```
# /opt/bacula/bin/acsls-changer showvollockall
ACS_QUERY_LOCK_VOL_RESPONSE: STATUS_SUCCESS (0) Vols: 2

VOL: F71072
    lockid:27968 userid: regress
    duration:2553 pending: 0
    status STATUS_VOLUME_IN_USE (99)

VOL: F71073
    lockid:6903 userid: regress
    duration:2424 pending: 0
    status STATUS_VOLUME_IN_USE (99)
```

The number of volumes which have locks assigned is printed as **Vols:<nr>**.
When no volumes with assigned locks are found then the number of volumes will be zero.

Best practice

Label Volumes

To use tape volumes in Bacula Enterprise they must first be labeled. It is recommended to use the command `label barcodes` which will automatically assign **ACSLS** Volume names to the **Bacula** volumes.

Note: It is not recommended (but technically possible) to use a different **Bacula** volume name for an **ACSLS** Volume.

Access Control

To use the **ACSLS** Access Control feature you should enable it in your ACSLM software (verify *ACSLS User Guide* for this) and define a `USERID` parameter in `acsls-changer.conf` file. Without it your Access Control won't work. With **ACSLS** Access Control you can define fine-grained access rules to your resources including allowed commands and volumes (with *Volume Access Control*). The following commands are used by `acsls-changer` and cannot be disabled:

- `mount`
- `dismount`
- `dismount_force`
- `query`
- `lock` - when resource locking is enabled, see: *Resource locking*
- `query_lock` - when resource locking is enabled
- `clear_lock` - when resource locking is enabled

With *ACSLS Volume Access Control*, you can limit the volume pool available for **Bacula**. Consult your *ACSLS User Guide* for information on setting up volume pool limitations.

Resource locking

You can use the ACSLS resource locking mechanism to protect tape drive and volume resources when used by a Bacula Enterprise Storage Daemon. To use this mechanism you need to define a `USERID=<userid>` and `LOCKING=Yes` parameters in the `acsls-changer.conf` file (See: *Changer Config File*). If it is not strictly required to avoid resource locking in your environment, you can disable resource locking by simply removing the `LOCKING` parameter from the config file or set it to `No`.

Request timeout

Some **ACSLS** requests may take longer than expected due to busy tape libraries or other hardware issues. If you get the following error code:

```
# ./acsls-changer unload 1 2 F70170
Unloading Volume F70170 from drive 1 ... code 296 (296)
```

it means a timeout has occured and you should verify that this is the problem. It may be necessary to modify the `TIMEOUT` value in the `acsls-changer.conf` file (See: *Changer Config File*) and `Maximum Changer Wait` in the Bacula Storage Daemon `bacula-sd.conf` file.

You can get different timeout messages during mount ("*Timeout on volume available status!*") or unmount ("*Timeout on verify volume status!*") operations. These messages mean that Bacula is unable to confirm the expected volume status during these operations before a timeout. If you get these timeout messages you should verify a real volume status in ACSLS. If required, modify the `TIMEOUT` value in the `acsls-changer.conf` file (See: *Changer Config File*) and `Maximum Changer Wait` in the Bacula Storage Daemon `bacula-sd.conf` file.

Stalled lockid

During operations, if you get an error message about a stalled lockid, then you will need to manuallly clear the lock outside `acsls-changer` (i.e. using a `clear lock` of `cmd_proc` application).

```
Error: Stalled lockid found (nnn). Make sure that the system is not shared (..
↪.)
```

This makes the `acsls-changer` out of sync. In this case you should verify what caused a lockid to disappear from the **ACSLS** system and remedy this. One typical cause of this can be a shared `acsls-changer` setup where one installation creates resource locks and other does not. This kind of setup is unsupported.

When you've removed the root cause of this error it could be required to sync the `acsls-changer` with **ACSLS** system again otherwise you will get the same error again. To do this, you have to first stop the Bacula Storage Daemon then remove the first line from VOLDB file (`/opt/bacula/working/acsls-changer.dbt`) which should contain the following value:

```
LOCKID:nnn
```

If the first line of `/opt/bacula/working/acsls-changer.dbt` file does not contain a LOCKID word then you should not modify this file.

Trace file

You can enable debug trace file generation using the following `acsls-changer` command parameters: `-d<nn>`, `-dt` and `-T`, where `nn` is a required debug level, i.e.

```
Changer Command = "/opt/bacula/bin/acsls-changer -d100 -dt -T %o %d %S"
```

Then the debug trace file will be generated at: `/opt/bacula/working/acsls-changer.trace` The option `-T` enables trace file generation and option `-dt` add timestamps to every debug log.

Other

Supported Platforms

The Bacula Enterprise ACSLS Plugin is supported on the following platforms:

- RHEL version 6 and 7
- SLES version 11 and 12

Limitations

- **ACSLS** Media Management feature is not supported. You should use a standard **Bacula** `Media Type` directive to manage different media.
  This limitation may be removed in the future.
- **ACSLS** dynamic drive allocation feature is not supported. **Bacula** uses a static drive mapping defined in `acsls-changer.conf` config file.
  This limitation may be removed in the future.
- **ACSLS** Scratch Pools management feature is not supported. **Bacula** uses its own Scratch Pool management.
  This limitation may be removed in the future.
- **Bacula** `mtx-changer transfer` command is not supported as there is no such functionality in **ACSLS**.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

1.4 HPE StoreOnce Catalyst Plugin

The HPE StoreOnce Catalyst Plugin, also known as Bacula Catalyst File System (BCFS), is a filesystem client for HPE StoreOnce Catalyst Stores.

The HPE StoreOnce Catalyst Plugin enables the mounting of a StoreOnce Catalyst Store as a filesystem specifically for the storage of **Bacula's** backups. It offers the advantage of supporting Immutability, which is not available with StoreOnce NAS Share. It is important to note that the HPE StoreOnce Catalyst Plugin does not fully comply with POSIX standards; further details can be found in the section entitled *Limitations* below.

By default, access to the filesystem is restricted to the user who mounted it, and it is advised against using the `chown` command to alter this access.

To optimize functionality, it is advisable to utilize the `setuid=bacula` option, allowing BCFS to operate under the *bacula* user (not as root), while also ensuring that the *mountpoint* is owned by the *bacula* user.

Plugin Usage

The HPE StoreOnce Catalyst Store can be mounted as a Bacula Catalyst File System (BCFS) filesystem by using the following command:

```
bcfs <CONFIG-FILE> <mountpoint> [options]
```

The <CONFIG-FILE> file holds the *Catalyst's* configuration parameters, see below for the format and the parameters.

To unmount it, the following commands can be used depending on the operating system:

```
fusermount3 -u <mountpoint>    # Linux
umount <mountpoint>            # OS X, FreeBSD or recent Linux
```

bcfs Options

-o opt,[opt...]

mount options, see below for details. A variety of FUSE options can be given here as well. The `allow_root` is forced by the program. The use of the `direct_io` option is **discouraged** as Bacula does not meet any criteria to leverage this option. See *Mounting from fstab* for more options.

-h, --help	print help and exit.
-V, --version	print version information and exit.
-d, --debug	print debugging information.
-v, --verbose	print debugging information.
-f	do not daemonize, stay in foreground.
-s	Single threaded operation.

-o logfile=<PATH>

to specify the location of the BCFS log file. (default is *stderr*)

-o loglevel=<LEVEL>

to specify the level of BCFS log file. Valid values are: `debug`, `info`, `warning` and `error` (default is `error`)

BCFS Plugin Configuration

Configuration File

The configuration file, typically referred to as `bcfs.conf`, is structured as follows:

```
#
# Connection section
#
address = 10.0.99.242
store_name = test2
cmd_port = 9387
data_port = 9388
user = Admin
password = MySecretPassword
#
# Logging section
#
catalyst_log_file = /tmp/bcatalystfs.log
catalyst_log_size_mb = 50
# catalyst_log_level = debug|trace|info|quiet|error
catalyst_log_level = error
#
# Rule Section
#
rule_20 = fixed=4096:*.add
rule_50 = variable:*
```

Exercise caution, as this file contains a **password** and must be accessible by the BCFS process. If you use the `setuid=UID` option when initializing the filesystem, ensure that the user with the specified UID has the necessary permissions to access this file.

Note: This file is formatted as a `.ini` file. You may initiate a comment by placing a # (number sign) at the start of a line. *Quote* characters are treated like any other character; therefore, **do not enclose your strings in quotes**.

The = (Equal) sign serves as a delimiter between the *key* and the *value*. Any *spaces* at the beginning or end of the *key* or *value* will be disregarded, while spaces within the values are considered significant.

The configuration file options are:

address
  This is the *ip* address or FQDN of your *Catalyst Server* (*MANDATORY*).

store_name
  This is the name of your *Catalyst Store* (*MANDATORY*).

cmd_port
  This is the command session port number, default is 9387.

data_port
  This is the data session port number, default is 9388.

user
  This is the user name used to connect to the Store of the *Catalyst Server* (*MANDATORY*).

password
: This is the password of the user above (*MANDATORY*).

catalyst_log_file
: This is the name of your local *Catalyst's* log file. This log file is handled by the *Catalyst's* API (*MANDATORY*). This is different of the *fuse* log file, see *Logging* below for more.

catalyst_log_size_mb
: This is the maximum size of the *Catalyst's* log file in MB. Do not specify the *MB* unit! The beginning of the file is overwritten when the file reaches the limit, this is how the Catalyst log file works. Default is 50MB.

catalyst_log_level
: This is the log level of the *Catalyst's* log file. From the most to the less verbose you can use: *debug*, *trace*, *info*, *quiet*, *error*. Default is *error*.

rule_XXX
: Keys starting with `*rule_\*` are for rules that define the deduplication mode regarding the pattern of the file, see below.

Note: The BCFS daemon has its own log file that is controlled by the command line parameters.

Deduplication Rules

The *XXX* part in the *rule_XXX* has no other purpose than uniquely identifying the rule. It is not used to define any priority order. Each rule consists of two components, which are divided by a `:` (colon). The first part is the deduplication mode, and the second part is a well known shell *globbing* file name. See the unix glob(7) manpage for more.

The first part can be of the 2 forms:

variable
: meaning that the *Catalyst* will use variable block deduplication for the files matching the pattern.

fixed=SIZE
: meaning that the *Catalyst* will use fixed block deduplication to the files matching the pattern. `SIZE` is the size in bytes of the blocks and must be between `OSCMN_MINIMUM_FIXED_CHUNK_SIZE` and `MaximumFixedChunkSize`. These two values are displayed in the log file during the initialization of the BCFS client and can also be found in the `sys/version` special file. The minimum value is hardcoded in the API at 2048 (2KiB), and the maximum value is set at 8192 (8KiB) on our *Catalyst* test server. If the size exceeds these limits, the error message displays them. The recommended value is **4096**. **Use the fixed mode only for files that will be accessed on an aligned block boundary, otherwise file input/output errors will occur.**

Filenames corresponding to the second part will use the deduplication mode defined in the first part. Rules are applied in the order they are defined- - the first one that matches, applies. If no rules apply, the default is *variable*.

Deduplication Rules Examples

It is possible to use the Bacula Aligned Plugin, and store the Aligned volumes on the BCFS file system.

When using the Aligned Plugin and storing your Data and Metadata volumes on the BCFS mount point, it is advisable to deduplicate the Data part of the volume that has the *.add* extension and not deduplicate the metadata volume using these rules:

```
rule_20 = fixed=4096:*.add
rule_50 = variable:*
```

4096 is the value recommended by the HPE support.

When using a Bacula Storage without the Aligned Plugin, it is advisable to use the *Variable Block Deduplication* for all the volumes that need deduplication. By prefixing these volumes with the identifier "DedupVolume", the following rules can be applied:

```
rule_20 = variable:DedupVolume*
rule_50 = variable:*
```

Important: Remember that this is *globbing*, not *regex* patterns.

Logging

BCFS generates two distinct log files:

- The first log file is managed by the *Catalyst* library and is configured through parameters such as `catalyst_log_file`, `catalyst_log_level`, and `catalyst_log_size_mb` within the configuration file. This log operates as a circular buffer, resembling a ring structure, which renders the use of the `tail` command ineffective. Additionally, it contains a binary header of a few bytes.
- The second log file is of greater significance as it captures the BCFS logs. It is configured using the `-o logfile=PATH` and should be maintained at the `error` level with the option to increase the level for more information. The level can be set using `-o loglevel=<LEVEL>`, where the permissible levels include: `debug`, `info`, `warning`, and `error`

It is crucial to ensure that the BCFS process is able to write to these log files, particularly when using the `setuid=UID` option.

Mounting from fstab

The BCFS can be mounted automatically from `/etc/fstab`, using the following line:

```
/etc/bcfs.conf /backup fuse.bcfs setuid=bacula,nodev,noexec,noatime,
↪loglevel=error,logfile=/tmp/bcfs.log 0 0
```

The first parameter is the configuration file. Notice that the file system type is `bcfs` (for backwards compatibility, you may also use `fuse.bcfs`). The `bcfs` binary must be in the *system path*, usually it is installed into `/usr/bin`. The `setuid=bacula` changes the user ID of the process to `bacula`. If this adjustment is made, it is essential to ensure that the *bacula* user possesses the necessary read and write permissions for the mount point. Additionally, it is crucial to confirm that the `/sbin/nologin` shell is not set for user `bacula`:

```
# grep bacula /etc/passwd /etc/shadow
/etc/passwd:bacula:x:990:986:Bacula:/opt/bacula/working:/bin/sh
/etc/shadow:bacula:!!:19550::::::
```

It is possible to use the commands `chsh` to change the *shell* and `passwd` to lock the account:

```
# chsh -s /bin/sh bacula
# passwd -l bacula
```

The `nodev` and `noexec``options are recommended for enhancing security, while the ``noatime` option prevents unnecessary updates. For the `loglevel` and `logfile` options, see *Logging* above.

Bacula Storage Configuration

Using the *Aligned* driver with the *Fixed* mode and a 4KiB block size will yield optimal deduplication performance on the *Catalyst* system. If there is a discrepancy in alignment between your data source, such as when dealing with virtual machines that have misaligned partitions, or if your data is subject to changes over time due to the management of numerous uncompressed text documents, the *Variable* mode may deliver superior results. Regardless, the *Aligned* driver consistently outperforms the *File* driver in terms of deduplication efficiency.

Here is a typical device configuration for a Bacula *Aligned* device:

```
Device {
  Name = storeonce1
  DeviceType = Aligned
  MediaType = sobcfs
  ArchiveDevice = /opt/bacula/archive/catalyst001
  AlwaysOpen = no
  LabelMedia = yes
  AutomaticMount = yes
  RemovableMedia = no
  RandomAccess = yes

  MaximumConcurrentJobs = 1
  SetVolumeReadOnly = yes
  MinimumVolumeProtectionTime = 30 days
      #FileAlignment = 64k
}
```

The following directives are noteworthy:

Device Type = Aligned
Chose the *Aligned* driver for the best deduplication performance.

Media Type = sobcfs
Do not forget to define a specific *MediaType* for the volumes in the same store.

Set Volume Read Only = yes
This allows the device to activate the *immutability* on the volumes.

Minimum Volume Protection Time = 30 days
This value must match the value configured on the *Catalyst* system.

Maximum Concurrent Jobs = 1

If you are using *Aligned*, it is good to set the *MCJ* to 1, as the *Aligned* driver forces the *MCJ* to 1 anyway.

Do not modify the directives controlling the size of the *chunks* related to the *Aligned* driver; the default settings, particularly the *64K* for *FileAlignment*, are optimal.

Certainly, multiple devices or even an autochanger can use the same *Catalyst Store*.

Note: The `SetVolumeImmutable` and the `SetVolumeAppendOnly` directives are not supported with the Bacula Catalyst File System (BCFS). Instead, `SetVolumeReadOnly` must be used.

Note: It is useless to compress the data in the *Fileset* as the *Catalyst* already compress the chunks.

Immutability

The BCFS offers support for immutability, provided that this option is activated on the *Catalyst Store*.

First, verify in the *StoreOnce Management Console* that the option is enabled for your *Catalyst Store*. Check the *details* page of the *Store* to ensure that the value for *Server Controlled Data Immutability Retention* in the *Security* section is not set to *No limits*.

In the Bacula Device configuration, immutability is configured by setting `SetVolumeReadOnly = yes`, and the *Server Controlled Data Immutability Retention* value configured by using the Minimum Volume Protection Time directive.

When you enable the *Server Controlled Data Immutability* feature, you need to specify the duration of the Immutable Period, which defaults to 30 days. The Immutable Period begins once Bacula marks the item as Complete. See *Querying the Bacula Catalyst File System (BCFS) Tags*.

Note: The Immutable Period includes a one-hour grace period during which you can delete or append data to the end of a Catalyst item. After this grace period, the Immutable Period officially starts.

The *volumes* that are designated as read-only on the filesystem will have the *Complete* tag applied and will either be eligible for immutability or already be immutable.

When immutability is configured in the Bacula Device resource using the Bacula Catalyst File System (BCFS) to store the Bacula volumes, the volume will be marked as Read-Only if:

- during a backup job, the volume reaches its full capacity (for example, when `MaximumVolumeBytes` is reached), then it is

marked as Read-Only:

```
03-Jun 10:53 bacula-catalyst-sd JobId 7419: Marking Volume "Vol-0001" as read-
↪only. Retention set to 04-Jun-2025 10:53 (1 day).
```

- when the bconsole `update volumeprotect` command is run:

```
*update volumeprotect
Found 1 volumes with status Used/Full that must be protected
Connected to Storage "bacula-catalyst" at bacula-catalyst:9103 with TLS
3000 Marking volume "Vol-0001" as read-only.
```

Note: When Bacula considers the volume as Used (for example, when `MaximumVolumeJobs` is reached), the volume is not automatically set as read-only. It requires the `update volumeprotect` command to be run. See: *Best Practices*.

Note: It is recommended to disable `Autoprune` when using the immutability feature. Instead, use `prune volume expired yes` to prune volumes based on the `VolumeRetention` value.

About Deduplication

Two data segments of 4KiB will deduplicate due to their identical nature. Their similarity arises from sharing the same origin. It is highly unlikely for two data segments of a size significant enough to warrant consideration for deduplication to lack a common origin. Achieving a favorable deduplication ratio is not necessarily advantageous. The choice between duplicated and centralized data presents both advantages and disadvantages, and it is essential to determine which option best suits your needs.

The major sources of duplication are:

- Always running *Full* backups instead of *Incremental*.
- Running backups of Virtual Machines that share the same OS.
- Having multiple copies of identical data.

If you are aware of duplicated data and are experiencing a low deduplication ratio, it is advisable to take this matter seriously.

Additionally, the implementation of *compression* or *encryption* may disrupt the deduplication algorithm.

Querying the Bacula Catalyst File System (BCFS) Tags

The Bacula Catalyst File System (BCFS) adds extra attributes to the Bacula Volumes files.

This extra information can be retrieved from the volume file itself by using the `xattr` linux command.

The Bacula Catalyst File System (BCFS) tags will show a *true* value when present:

```
$ xattr -l /catalyst/Vol-0001
dedup: var
chunksize: 0
Complete: true
BaculaEnterprise: true
```

dedup
: can be *var* or *fixed* depending on the rules you have set up in the *Configuration File*.

chunksize
: when *dedup* is set to *fixed*, this is the size you have set up in the configuration file, else it is *0*.

Complete or Incomplete
: are *tags* handled by the *Catalyst*, *Complete* is synonymous to *read-only* while *Incomplete* is synonymous to *read/write*.

BaculaEnterprise
: is a *tag* that is added to every object that are created by the BCFS. A file without this *tag* has been created by another application.

If `xattr` is not available, you can try `getfattr`:

```
$ getfattr -d -m ".*" /catalyst/Vol-0001
getfattr: Removing leading '/' from absolute path names
# file: catalyst/Vol-0001
BaculaEnterprise="true"
Complete="true"
chunksize="0"
dedup="var"
```

Querying the sys/ Filesystem

The *sys/* sub-directory comprises a set of virtual files that serve as an interface to the BCFS internal.

command_arguments
: holds the arguments on the command line at the time BCFS was started:

```
bcfs -oallow_root -osubtype=bcfs,fsname=/etc/bcfs.conf -o rw,nodev,noexec,
↪noatime,nosuid /catalyst
```

configuration
: holds the configuration (the password line has been removed):

```
address = 10.0.99.242
user = Admin
cmd_port = 9387
data_port = 9388
store_name = test2
catalyst_log_file = /tmp/bcatalystfs.log
catalyst_log_size_mb = 50
# catalyst_log_level = debug|trace|info|quiet|error
catalyst_log_level = error
rule_10 = variable:*.add
#rule_20 = fixed=4096:*.quatre
rule_90 = variable:*
```

version
: holds the BCFS version and information from the *Catalyst's* library and server:

```
version: 1.0.1
fuse_version: 29
fuse_package_version: NA
read_bandwidth: high
write_bandwidth: low
OSCMN_MINIMUM_FIXED_CHUNK_SIZE: 2048
MaximumFixedChunkSize: 8192
ClientSoftwareVersion: RHEL_x64_2022-04-28T12.28.723_4.3.2-2217.20
ServerSerialNumber: 17TLGM2FH78D9WAM
ServerSoftwareVersion: 4.3.6-2323.20
```

(continues on next page)

(continued from previous page)

```
DiskCapacity: 882079956992
SupportIsvPerObjectImmutability: 1
MaximumDataSessions: 256
FreeDataSessions: 61
```

cmd_connections

holds information about recycled command connections:

```
0x55895bf8afb0 in_use=0 age=0s
0x55895bf8afc8 in_use=0 age=0s
0x55895bf8afe0 in_use=0 age=0s
0x55895bf8aff8 in_use=0 age=0s
0x55895bf8b010 in_use=0 age=14s
```

data_connections

one line per data connection to the *Catalyst*. Any open file has one line:

```
0x7ff0dc0225d0 off=6 age=14s name=foobar
```

store

holds the information about the store and the server that can change over time:

```
DiskCapacity: 882079956992
DiskSpaceFree: 870689128448
DedupeRatio: 3.9
UserDataSize: 25092777408
UserDataStored: 25092777408
DedupedDataSizeOnDisk: 6336564418
DedupeMetaSizeOnDisk: 501318090
DatabaseSizeOnDisk: 13312879
ScratchPadSizeOnDisk: 0
NumObjects: 51
SupportUserDataSizeQuotas: true
UserDataSizeLimit: 0 (unlimited)
SupportDedupedDataSizeOnDiskQuotas: true
DedupedDataSizeOnDiskLimit: 0 (unlimited)
last_update: 2024-11-20T13:36:20Z
last_update_epoch: 1732109780
```

The available `DiskSpaceFree` is shared among all the *Catalyst stores*. The `DedupeRatio` represents the ratio of `UserDataSize` and `DedupedDataSizeOnDisk`. The `UserDataStored` excludes the sparse regions from the total `UserDataSize`. The quota values, namely``UserDataSizeLimit`` and `DedupedDataSizeOnDiskLimit`, relate to the two previous values. The `last_update` indicates the timestamp when these value were retrieved.

Best Practices

In some cases, for example when the status of the Volume is changed by the Director via the update volume command, the Storage Daemon will not be able to change the permission on the Volume. Some Volumes may have the Full/Used status without the proper protection.

The command `update volumeprotect` is designed to determine the list of the volumes that are not protected and connect the Storage Daemon to update the permissions.

```
*update
Update choice:
     1: Volume parameters
     2: Pool from resource
     3: Slots from autochanger
     4: Long term statistics
     5: Snapshot parameters
     6: Volume protection attributes on Storage Daemon
Choose catalog item to update (1-6): 6
Found 1 volumes with status Used/Full that must be protected
Connected to Storage "bacula-catalyst" at bacula-catalyst:9103 with TLS
3000 Marking volume "Vol-0001" as read-only.
```

or

```
*update volumeprotect
Found 1 volumes with status Used/Full that must be protected
Connected to Storage "bacula-catalyst" at bacula-catalyst:9103 with TLS
3000 Marking volume "Vol-0002" as read-only.
```

This command can be scheduled in an Admin Job to run every day. For example:

```
Job {
  Name = update-protected-admin-job
  JobDefs = DefaultJob
  Type = Admin
  Runscript {
     Console = "update volumeprotect"
     RunsOnClient = no
     RunsWhen = Before
  }
}
```

Limitations

The HPE StoreOnce Catalyst Plugin has been implemented on top of the Catalyst API to address the storage requirements of the Bacula volumes and the Aligned Plugin.

In Bacula, the usage of the *file* based volumes mimics the use of the *tape* volumes, and the access patterns do not require the full range of features typically offered by contemporary filesystems.

Additionally, the Catalyst API has its limitations and may not readily fulfill all the demands of a traditional filesystem.

Nevertheless, it is anticipated that the Catalyst API will meet the storage needs for Bacula's volumes and its Aligned Plugin.

The following outlines the primary distinctions between *the HPE StoreOnce Catalyst Plugin* and a *POSIX* compliant filesystems.

No Sub-directory Tree

The Catalyst API exclusively supports *Stores* and *Items* and does not accommodate structures resembling a directory tree. However, this does not pose a limitation for Bacula, as each volume is required to have a unique name and is frequently stored within the same directory. Bacula handles all management tasks, eliminating the need for users to directly access the archive directory for their daily operations.

```
$ mkdir ~/mnt/dir
mkdir: cannot create directory ‘/home/bac/mnt/dir’: Function not implemented
```

File Cannot Be Renamed

The name of a Catalyst item is fixed and cannot be altered, which means the corresponding file is also unchangeable. However, the file can be duplicated under a new name, allowing for the original file to be removed.

```
$ mv ~/mnt/a4 ~/mnt/a99
mv: cannot move '/home/bac/mnt/a4' to '/home/bac/mnt/a99': Function not␣
↪implemented
```

Deletion of Open Files

When a file in use is deleted, FUSE attempts to rename it to a *hidden* file to delete it later. However, if the file cannot be renamed, as indicated in the section titled *File cannot be renamed* above, the operation will not succeed.

```
$ rm  ~/mnt/file
rm: cannot remove '/home/bac/mnt/file': Function not implemented
```

Content of File Cannot Be Modified

Modifying data within a file is possible, but **it will truncate the file at the point of modification, resulting in the loss of any subsequent data**. This is *CRUCIAL* to note, as attempting to run applications other than Bacula may lead to unforeseen consequences.

Characters Allowed in File Names

The name may consist of the following characters: `0-9 a-z A-Z _ . + [ ] ^ | ? * ( ) # :  ; = @ { }`. However, spaces are not permitted in the name.

Inconsistency between atime, mtime and ctime

The *Calatyst* API does not provide support for the `atime` value, `atime` is always a copy of the `mtime` value. When the file is not open (nobody reading or writing to it), these values come from the Catalyst server, using the Catalyst clock. However, when a file is open, these three values are managed at the fuse level in a consistent manner using the time of the FUSE host. They are initialized using the Catalyst API. If there is a discrepancy between the clocks of the client and the server, inconsistencies may arise.

2 Security

Important: Security solutions are used with the File Daemon.

2.1 Bacula Enterprise Antivirus Verify Job Plugin

Distinction between Antivirus Plugin and Malware

In an Antivirus Check, Bacula will transmit the files to the ClamAV Antivirus Socket, which will perform the scan and report to Bacula if any viruses are discovered. In the instance of Malware, Bacula will retrieve the Malware database signatures from https://abuse.ch/ and then do a file verification with those signatures. If a Backup job finds malware in the backup content, an error message is generated and the Job status is changed.

Overview

The Bacula Enterprise Antivirus Plugin provides integration between the ClamAV Antivirus daemon and Bacula verify jobs, allowing post-backup virus detection within Bacula Enterprise.

ClamAV Installation

ClamAV is an open source (GPLv2) anti-virus toolkit. It provides a flexible and scalable multi-threaded daemon. For more information on ClamAV architecture, best practices and options, please refer to https://docs.clamav.net/

Debian / Ubuntu

Installation:

Use the existing Debian packages:

```
sudo apt-get update
sudo apt-get install clamav clamav-daemon
```

TCP Configuration:

The ClamAV 'clamd' daemon is configured with the clamav.conf file (located in /etc/clamav/). By default, the ClamAV daemon listens on a Unix LocalSocket:

```
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket true
LocalSocketGroup clamav
LocalSocketMode 666
```

In order for Bacula to interact correctly with ClamAV, it is essential to reconfigure the ClamAV daemon so it allows TCP connections instead.

On Debian you can do so automatically by running:

```
sudo dpkg-reconfigure clamav-daemon
```

Answer "yes" to reconfigure automatically Make sure to change the socket type to "TCP". Choose the TCP port clamd will listen on (The default port 3310 will be assumed in the rest of the documentation). Continue the reconfiguration according to your needs (press enter to keep the default settings) At the end of the reconfiguration process, the ClamAV daemon restarts. You can verify that the clamav.conf file now contains:

```
TCPSocket 3310
```

Alternatively, you can reconfigure manually, by editing clamav.conf. Replace:

```
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket true
LocalSocketGroup clamav
LocalSocketMode 666
```

with:

```
TCPSocket 3310
```

and restart the ClamAV daemon:

```
sudo systemctl restart clamav-daemon
```

RHEL

Installation:

EPEL creates ClamAV packages. To enable the EPEL repository:

```
sudo dnf install -y epel-release
```

EPEL offers a selection of packages to install ClamAV:

```
clamd - The ClamAV Daemon
clamav - End-user tools for the ClamAV scanner
clamav-data - Virus signature data for the ClamAV scanner
clamav-devel - Header files and libraries for the ClamAV scanner
clamav-lib - Dynamic libraries for the ClamAV scanner
clamav-milter - Milter module for the ClamAV scanner
clamav-update - Auto-updater for the ClamAV scanner data-files
```

Bacula minimally requires clamav, clamd, and clamav-update to run:

```
sudo dnf install -y clamav clamd clamav-update
```

TCP Configuration:

The ClamAV daemon is configured with the clamav.conf file (located in /etc/clamav/).

Edit the configuration file:

```
sudo dnf install nano -y
sudo nano /etc/clamd.d/scan.conf
```

Make sure the Example line is commented out:

```
#Example
```

By default, the ClamAV daemon connects over Unix LocalSocket. In order for Bacula to interact correctly with ClamAV, it is essential to reconfigure the ClamAV daemon so it allows TCP connections instead. enable TCP connection instead by uncommenting the following line:

```
TCPSocket 3310
```

Optionally, you can also restrain the TCP binding (by default the ClamAV daemon binds to INADDR_ANY):

```
TCPAddr 127.0.0.1
```

Once that's done, you can run the virus definition database update:

```
sudo freshclam
```

Lastly, start the clamd service and run it on boot:

```
sudo systemctl enable clamd@scan
sudo systemctl start clamd@scan
```

Bacula Enterprise Antivirus Plugin Installation

Installation of the Bacula Enterprise Antivirus Plugin is most easily done by adding the repository file suitable for the existing subscription and the distributions package manager configuration. An example would be `/etc/apt/sources.list.d/bacula.list` for Debian based Linux distributions with the following content:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪bullseye-64/ stretch main
```

After that, a run of `apt-get update` is needed. Then, the plugin can be installed using `apt-get install bacula-enterprise-antivirus-plugin`

On RHEL, extend the repository file for your package manager to contain a section for the plugin - `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer@/rpms/bin/@version@/rhel7-
↪64/
enabled=1
protect=0
gpgcheck=0
```

Then perform a `yum update` and after that the package `bacula-enterprise-antivirus-plugin` can be installed with `yum install bacula-enterprise-antivirus-plugin`.

Manual installation of the packages can be done after downloading the right files from the Bacula Systems provided download area, and then using the low-level package manager (`rpm` or `dpkg`) to do the plugin installation.

Bacula Enterprise Verify Job Configuration

Parameters

The Bacula Enterprise Antivirus Plugin accepts two parameters:

- **hostname:** The binding address of the ClamAV daemon (specified in clamav.conf as TCPAddr). Can be any IP4 TCP address. Default is 'localhost'
- **port:** The ClamAV daemon port number (specified in clamav.conf as TCPSocket). Default port is 3310.

Plugin Options

Contrary to “classical” Bacula FD plugins, these parameters are configured in the Verify Job as PluginOptions rather than a “Plugin =” in a Fileset.

There are three possible ways to instruct a Bacula Verify Job to run the antivirus plugin:

# Add a PluginOptions directive to the Verify Job configuration (recommended):

```
Job {
  Name = Verify_and_AV_Scan
  Type = Verify
  Level = Data
  Client = localhost-fd
  Fileset = LinuxHome
  Storage = File
  Pool = Default
  Messages = Standard
  PluginOptions = "antivirus: hostname=127.0.0.1 port=3310"   # <---- Add␣
↪this line here
}
```

# Specify the `pluginoptions` as a parameter to the ‘run’ command in bconsole:

```
*run job=Verify_and_AV_Scan jobid=1 storage=File1 pluginoptions="antivirus:␣
↪hostname=localhost port=3310"
```

# Dynamically modify the verify job within bconsole

```
*run job=Verify_and_AV_Scan jobid=1 storage=File1

JobName:        Verify_and_AV_Scan
Level:          Data
Client:         localhost-fd
Fileset:        LinuxHome
Pool:           Default (From Job resource)
Storage:        File1 (From Command input)
Verify Job:     LinuxHome.2021-10-12_05.31.58_03
Verify List:
When:           2021-10-12 05:40:12
Priority:       10
OK to run? (yes/mod/no): m
Parameters to modify:
    1: Level
    2: Storage
    3: Job
    4: Fileset
    5: Client
    6: When
    7: Priority
    8: Pool
    9: Verify Job
    10: Plugin Options
Select parameter to modify (1-10): 10
```

(continues on next page)

(continued from previous page)

```
Please Plugin Options string: antivirus: hostname=127.0.0.1 port=3310
Run Verify Job
JobName:        Verify_and_AV_Scan
Level:          Data
Client:         localhost-fd
Fileset:        LinuxHome
Pool:           Default (From Job resource)
Storage:        File1 (From Command input)
Verify Job:     LinuxHome.2021-10-12_05.31.58_03
Verify List:
When:           2021-10-12 05:40:12
Priority:       10
Plugin Options: antivirus: hostname=127.0.0.1 port=3310
OK to run? (yes/mod/no): yes
```

Under normal conditions, the Verify Job will silently scan existing files from the specified backup and should terminate with a "Verify OK" Job status:

```
run job=Verify_and_AV_Scan jobid=1 storage=File1

JobName:        Verify_and_AV_Scan
Level:          Data
Client:         localhost-fd
Fileset:        LinuxHome
Pool:           Default (From Job resource)
Storage:        File1 (From Command input)
Verify Job:     LinuxHome.2021-10-12_06.04.11_03
Verify List:
When:           2021-10-12 06:04:17
Priority:       10
Plugin Options: antivirus: hostname=localhost port=3310
OK to run? (yes/mod/no): yes

12-Oct 06:04 localhost-dir JobId 2: Verifying against JobId=1 Job=LinuxHome.
↪2021-10-12_06.04.11_03
12-Oct 06:04 localhost-dir JobId 2: Start Verify JobId=2 Level=Data␣
↪Job=Verify_and_AV_Scan.2021-10-12_06.04.17_05
12-Oct 06:04 localhost-dir JobId 2: Connected to Storage "File1" at␣
↪localhost:8103 with TLS
12-Oct 06:04 localhost-dir JobId 2: Using Device "FileStorage1" to read.
12-Oct 06:04 localhost-dir JobId 2: Connected to Client "localhost-fd" at␣
↪localhost:8102 with TLS
12-Oct 06:04 localhost-fd JobId 2: Connected to Storage at localhost:8103␣
↪with TLS
12-Oct 06:04 localhost-sd JobId 2: Ready to read from volume "TestVolume001"␣
↪on File device "FileStorage1" (/mnt/archive).
12-Oct 06:04 localhost-fd JobId 2: Got plugin command = antivirus:␣
↪hostname=localhost port=3310
12-Oct 06:04 localhost-sd JobId 2: Forward spacing Volume "TestVolume001" to␣
↪addr=228
12-Oct 06:05 localhost-sd JobId 2: End of Volume "TestVolume001" at␣
↪addr=98844823 on device "FileStorage1" (/mnt/archive).
```

(continues on next page)

(continued from previous page)

```
12-Oct 06:05 localhost-sd JobId 2: Elapsed time=00:00:51, Transfer rate=1.935␣
↪M Bytes/second
12-Oct 06:05 localhost-dir JobId 2: Bacula localhost-dir 12.9.2 (11Oct21):
Build OS:               x86_64-pc-linux-gnu ubuntu 9.12
JobId:                  2
Job:                    Verify_and_AV_Scan.2021-10-12_06.04.17_05
Fileset:                LinuxHome
Verify Level:           Data
Client:                 localhost-fd
Verify JobId:           1
Verify Job:
Start time:             12-Oct-2021 06:04:19
End time:               12-Oct-2021 06:05:21
Elapsed time:           1 min 2 secs
Accurate:               no
Files Expected:         2,238
Files Examined:         2,238
Non-fatal FD errors:    0
SD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            **Verify OK**
```

When a virus is detected by ClamAV, an error is reported in the Bacula job log and the Job will continue to verify the rest of the files, but it will terminate with a “Verify OK – with warnings” Job status.

Within the job output, the antivirus plugin specifies the infected file(s) and the virus name(s) detected.

```
run job=Verify_and_AV_Scan jobid=3 storage=File1

JobName:        Verify_and_AV_Scan
Level:          Data
Client:         localhost-fd
Fileset:        LinuxHome
Pool:           Default (From Job resource)
Storage:        File1 (From Command input)
Verify Job:     LinuxHome.2021-10-12_06.05.22_07
Verify List:
When:           2021-10-12 06:05:27
Priority:       10
Plugin Options: antivirus: hostname=localhost port=3310
OK to run? (yes/mod/no): yes

12-Oct 06:05 localhost-dir JobId 4: Verifying against JobId=3 Job=LinuxHome.
↪2021-10-12_06.05.22_07
12-Oct 06:05 localhost-dir JobId 4: Start Verify JobId=4 Level=Data␣
↪Job=Verify_and_AV_Scan.2021-10-12_06.05.27_09
12-Oct 06:05 localhost-dir JobId 4: Connected to Storage "File1" at␣
↪localhost:8103 with TLS
12-Oct 06:05 localhost-dir JobId 4: Using Device "FileStorage1" to read.
12-Oct 06:05 localhost-dir JobId 4: Connected to Client "localhost-fd" at␣
↪localhost:8102 with TLS
```

(continues on next page)

(continued from previous page)

```
12-Oct 06:05 localhost-fd JobId 4: Connected to Storage at localhost:8103␣
↪with TLS
12-Oct 06:05 localhost-sd JobId 4: Ready to read from volume "TestVolume001"␣
↪on File device "FileStorage1" (/mnt/archive).
12-Oct 06:05 localhost-fd JobId 4: Got plugin command = antivirus:␣
↪hostname=localhost port=3310
12-Oct 06:05 localhost-sd JobId 4: Forward spacing Volume "TestVolume001" to␣
↪addr=98844823
12-Oct 06:05 localhost-sd JobId 4: End of Volume "TestVolume001" at␣
↪addr=98845442 on device "FileStorage1" (/mnt/archive).
12-Oct 06:05 localhost-sd JobId 4: Elapsed time=00:00:01, Transfer rate=197 ␣
↪Bytes/second
12-Oct 06:05 localhost-fd JobId 4: **Error: /home/nbizet/src/bacula-bee/
↪regress/tmp/eicar Virus detected stream: Eicar-Signature FOUND**
12-Oct 06:05 localhost-dir JobId 4: Bacula localhost-dir 12.9.2 (11Oct21):
Build OS:               x86_64-pc-linux-gnu ubuntu 9.12
JobId:                  4
Job:                    Verify_and_AV_Scan.2021-10-12_06.05.27_09
Fileset:                LinuxHome
Verify Level:           Data
Client:                 localhost-fd
Verify JobId:           3
Verify Job:
Start time:             12-Oct-2021 06:05:29
End time:               12-Oct-2021 06:05:29
Elapsed time:           1 sec
Accurate:               no
Files Expected:         1
Files Examined:         1
Non-fatal FD errors:    1
SD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            **Verify OK -- with warnings**
```

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

2.2 Security Plugin

- *Overview*
- *Security Hooks*
- *Installation*
- *Configuration*
- *Advanced*

- *Limitations*

Overview

Features Summary

The **Bacula Enterprise Security** plugin provides a framework that can be used to check for vulnerabilities using the Bacula File Daemon on your servers. The security checks are executed once a day during any Backup Job. Information about any vulnerabilities found is printed in the Job report and a potential error message can be logged in the Job log. A *Security Object* will be inserted in the catalog for further analysis.

Security Hooks

Security hooks are installed in `/opt/bacula/etc/bcheck_sys.d` and can be executed separately.

Basic

Linux

```
000-bacula-basic
```

The *basic* check will analyse the Bacula Director configuration to check the password policy. It also controls the different permission checks on various Bacula files under `/opt/bacula`.

Windows

```
001-WindowsUpdate.ps1
```

The *WindowsUpdate* check will analyse the Windows Security updates and report the uninstalled ones with relevant level of importance.

Installation

Packages

Packages of the **Security** plugin are available for supported platforms. Please contact Bacula Systems Support team to get them.

Download the **Security** plugin package to your server where a Bacula File Daemon is installed and then install using the package manager

Debian/Ubuntu

```
dpkg -i bacula-enterprise-security-plugin*.deb
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **Security** plugin.

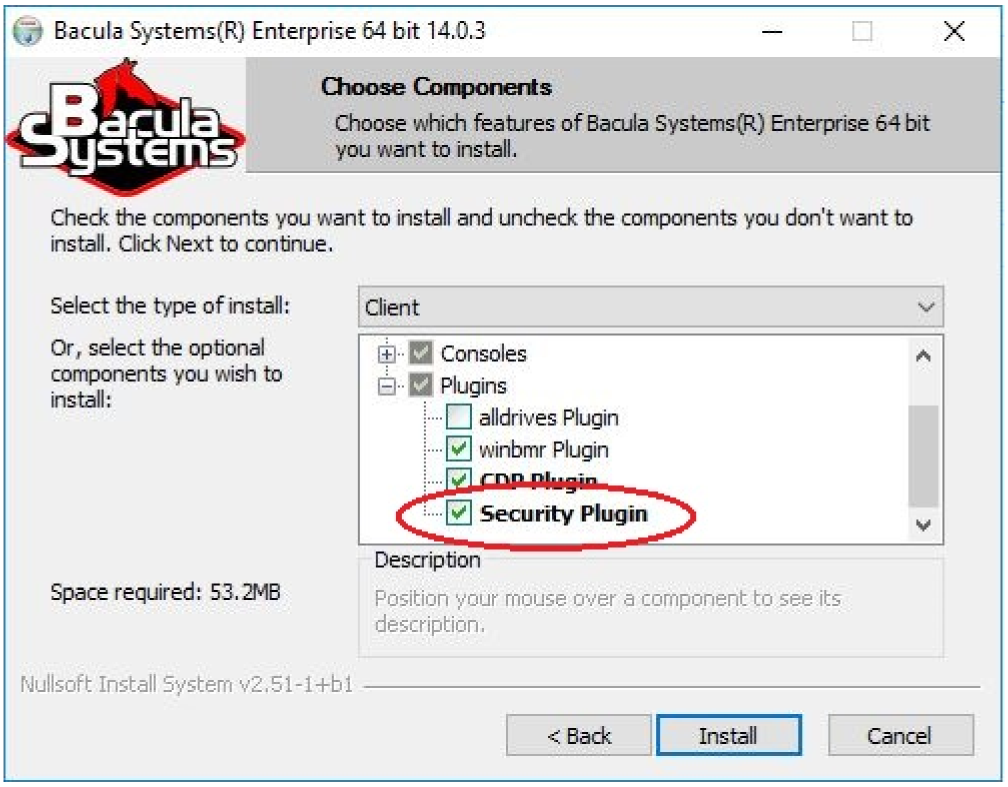
RHEL

```
rpm -ivh bacula-enterprise-security-plugin*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **Security** plugin.

Windows

The **Bacula Enterprise Security** plugin is selectable as a component of the **File Daemon** windows installer.



Bacula Systems(R) Enterprise 64 bit 14.0.3

Choose Components

Choose which features of Bacula Systems(R) Enterprise 64 bit you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Client

Or, select the optional components you wish to install:

- ✓ Consoles
- ✓ Plugins
  - alldrives Plugin
  - ✓ winbmr Plugin
  - ✓ CDP Plugin
  - ✓ **Security Plugin**

Space required: 53.2MB

Description

Position your mouse over a component to see its description.

Nullsoft Install System v2.51-1+b1

Fig. 5: The Security plugin in the File Daemon windows installer

Configuration

File Daemon Configuration

On the File Daemon host server, the **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` has to point to where the `security-fd.so` plugin is installed. The standard directory for Bacula plugins is `/opt/bacula/plugins`

```
FileDaemon {
   Name = bacula-fd
   Plugin Directory = /opt/bacula/plugins
   Plugin Options = "security: interval=2days"
   ...
}
```

The **Plugin Options** directive can be used to configure options of the **Security** plugin.

Table 7: Security plugin parameters

Option	Default	Description
interval	24h	The interval parameter specifies the time between two security checks.

Advanced

Forcing a New Check

It is possible to force a new check by deleting the file `/opt/bacula/working/security.ts`

Hook Protocol Definition

Security hooks can be written in any language. Some environment variables are passed to all hooks.

Table 8: Environnement variables

Option	Default	Description
BACULA_WORKINGDIR BACULA_SYSCONFDIR BACULA_BINDIR	/opt/bacula/working /opt/bacula/etc /opt/bacula/bin	Bacula Working directory Bacula Configuration directory Bacula Binary directory

The output provided by the hook is a JSON object with the following information:

```
   {
   "source": "chkrootkit",
   "version": "0.52",
   "error": 1,
   "events": [
       {
        "level": 'f',
        "message": "INFECTED: Possible Malicious Linux.Xor.DDoS installed"
       },
```

(continues on next page)

(continued from previous page)

```
        {
         "level": 'f',
         "message": "INFECTED: Possible Malicious Linux.XXX installed"
        }
    ]
},
```

Table 9: JSON fields

Option	Description
source | (String) Name of the hook version | (String) Version of the hook program error | (Int) different from zero to raise an error events | (Array) list of different events	

Each events have the following information

Table 10: JSON Events fields

Option	Description
level message	(char) Status of the test (f: fatal, T: ok, W: warning) (String) Error to be displayed. (contains simple characters)

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

2.3 BGuardian

The following article aims at presenting the reader with information about the **Bacula Enterprise BGuardian Plugin** (Bacula Guardian). The document briefly describes the target technologies of the plugin, defines the scope of its operations, and presents its main features.

Bacula Enterprise BGuardian Plugin is intended to become the ultimate tool designed to facilitate and automate some of the most important tasks around security analysis for a backup environment using Bacula Enterprise, providing a comprehensive overview of your system's security posture, highlighting potential issues, suspicious activities, and weak points.

With its advanced capabilities and comprehensive approach, BGuardian safeguards your environment by meticulously scrutinizing the whole Bacula Configuration, the evolution and behavior of the executed jobs, as well as the status of the different components of the target system.

Using statistical analysis and best-practices knowledge, it provides backup poisoning detection features, as well as secure configuration assessment. It does it by extracting valuable insights from the gathered information and presenting it in the form of easy to understand reports. On the other hand, it also generates persistent alerts that serve as a framework to control and act upon any found issue.

Even if BGuardian represents a very important help in terms of security for a given environment, it is crucial to remark that security must be considered as a whole in any organization. It starts on the very internal roots of any corporation when a security plan and recovery strategies are well-defined and followed. It continues with the application of best practices at all levels: Using secure communications, using less privileges principles (Zero-trust), using secure network architectures, strong passwords policies, monitoring and auditing processes, multifactor authentication, data encryption, data immutability and many other technical features. However, probably one of the most important parts, comes with the application of common sense and a good education in secure practices for all the people inside the organization, which means things like avoid phishing attacks, be careful with any untrusted or not updated software, not share private or internal information in sensible places, not reuse passwords and many more.

Together with BGuardian, **Bacula Enterprise** offers all the other needed features to make the backup environment an extremely secure place. In order to have more information about them, refer to the appropriate section associated to the different security features listed in: Security Features.

Through subchapters, more in-depth information can be found about the topics presented below.

Scope

Bacula Enterprise BGuardian Plugin currently supports any platform where Bacula Director can be deployed.

This plugin is available since **Bacula Enterprise 16.0.12**.

Features

General Features

The main feature this plugin offers is to act as an assistant in order to help the system administrator to have a more solid and secure environment. This can be divided into the following generic features:

- Backup poisoning detection: Mark jobs with unexpected values in the amount of data processed, which could be a result of ransomware activities
- Secure configuration assessment: Make suggestions of configuration modifications to help to comply with secure recommendations and best-practices
- Failure patterns detection: Detection of potential issues related to running services
- Friendly reports generation: Detailed logging of analysis activities while running
- Persistent alerts generation: Summarized information that generates Bacula Events when each alert is created or recovered, to be updated with each analysis execution

Services

Below is the list of services/checks that the plugin provides:

- Strong password checking
- Duplicated password checking
- Strong permissions in Bacula configuration
- Correct users for running processes
- Recent successful catalog backup
- Recent successful backup of Bacula configuration

- Recent usage of Restore jobs
- Recent usage of Verify jobs
- Recent usage of Copy/Migration jobs to a different storage tier
- Usage of malware protection in jobs
- Usage of restricted consoles
- Usage of antivirus jobs for every client
- Usage of Events in Message resources
- Usage of encryption in the environment
- Detect cloud devices without encryption
- Usage of volume protection in the environment
- Director status (errors, FIPS usage, debug flags)
- Usage of DirAddress setting to limit Director service to be listening on specific interfaces
- Reachability and status of Storage Daemons (errors, FIPS usage, debug flags)
- Control of having enough free space on Storage Daemon devices
- Control of having enough free space for Deduplication in Dedup enabled Storage Daemons
- Detection of any kind of errors in Global Endpoint Deduplication engine (general errors, container errors, vacuum errors…)
- Detection of orphan, suspect or missed references in Global Endpoint Deduplication engine
- Recent execution of the Global Endpoint Deduplication Vacuum process for Dedup enabled Storage Daemons
- Reachability and status of File Daemons (errors, FIPS usage, debug flags)
- Usage of security plugin in each Client
- Check running Bacula versions among the different daemons and report any unsupported differences
- Check recent executions of PostgreSQL vacuum procedures over key tables for Bacula
- Check recent executions of PostgreSQL analyze procedures over key tables for Bacula
- Check if PostgreSQL configuration values are under or over the recommended thresholds
- Backup poisoning detection through deviation analysis
- Detection of jobs under a threshold success ratio
- Detection of jobs without a recent successful execution
- Detection of objects (virtual machines, databases…) without a recent successful backup
- Detection of jobs failed consecutively a specified number of times
- Detection of successful Full backup jobs that did not backup any data
- Detection of jobs that were not copied to any 2-Tier storage layer
- Detection of jobs that were never verified
- Detection of jobs where a restore was never attempted
- Detection of BWeb users which do not have 2Factor authentication enabled

- Detection of Incremental or Differential jobs whose predecessor is no longer in the catalog
- Detection of jobs where “will not descend” is reported due to ‘onefs = yes’. This is a possible indication that data which might be expected to be backed up is not being backed up
- Report of jobs where viruses or malware is found
- Report of recent recorded Bacula events about security (for instance, failed bconsole connections)
- Report of jobs with Global Endpoint Deduplication enabled which show a low deduplication ratio

This list of features will be growing with future versions of this plugin.

Architecture

Bacula Enterprise BGuardian Plugin is a Bacula **Director plugin** which may be run manually, or automatically on a periodic schedule via a Bacula Admin Job.

This tool is packed as a configurable daemon built on top of the Java language. The daemon can automatically communicate with Bacula through the following channels:

- bconsole
- bdirjson tool
- Direct connection to the SQL catalog

Once the different services are performed, BGuardian writes user-friendly HTML reports into the local filesystem, JSON reports that can be processed by any other tool, as well as other internal files which provide a persistent alert framework with information that is kept through subsequent daemon executions and that is available also for other layers of Bacula.

Below is a simplified vision of the BGuardian architecture within a generic **Bacula Enterprise** deployment:

Installation

This article describes how to install Bacula Enterprise BGuardian Plugin.

Prerequisites

- The BGuardian Plugin need to be installed on the host where the Bacula Director is installed.
- The plugin works through a Java daemon, therefore Java needs to be installed onto the host through a JRE or JDK package (openjdk-11-jre for example).
- The Java environment needs to be in version 11 or above and the Java binary must be available in the system PATH.

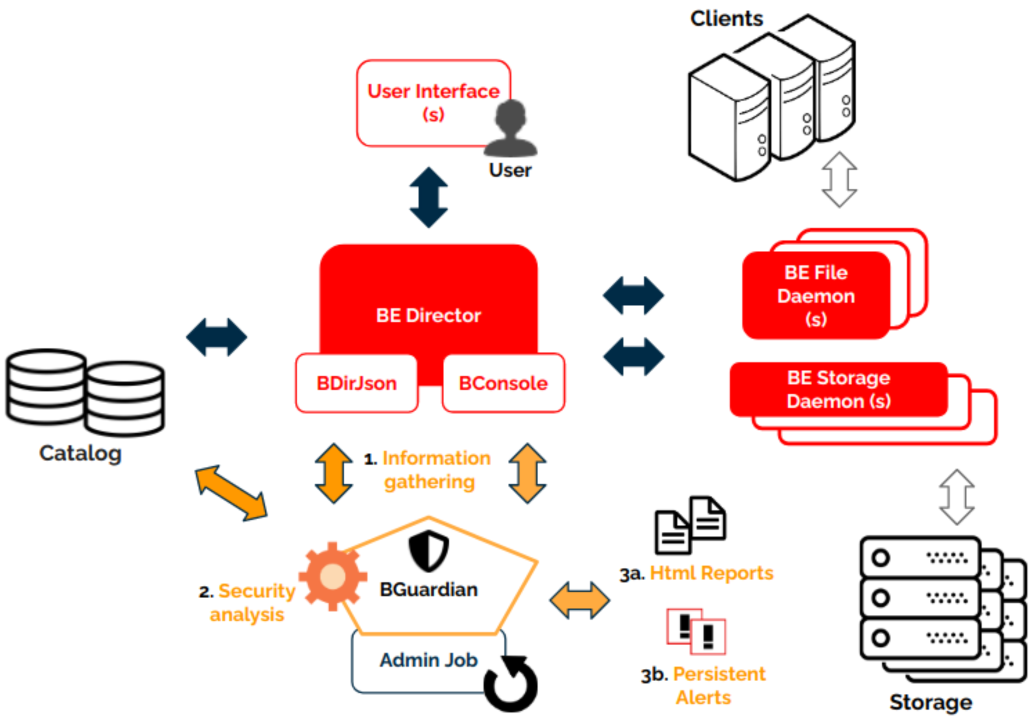


Fig. 6: BGuardian Plugin Architecture

Installation Methods

- *BGuardian Plugin - Installation with BIM* (recommended)
- *BGuardian Plugin - Installation with Package Managers*

BGuardian Plugin - Installation with BIM

The recommended way to install any Bacula component, including any Daemon and any Plugin is using the Bacula Installation Manager (BIM).

Steps

1. Install Director.
2. Select the bguardian key in the plugin selection step.

Result

BGuardian Plugin is installed. Click here for more details.

For more information, visit Linux: Bacula Enterprise Installation with BIM.

BGuardian Plugin - Installation with Package Managers

Another way to install this plugin is using the Package Manager of the used distribution.

Here, Debian Bullseye is taken as an example base Linux distribution. The process will be very similar in any version of Debian, or any other Debian-based distribution (e.g. such as Ubuntu). In case of using RPM based distributions, like RHEL, Oracle Linux or Suse Linux, the main difference is to switch to the proper package manager of that distribution, usually *yum*.

Steps

1. Add the repository file suitable for the existing customer subscription, and the Debian version utilized. An example would be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 6: **APT repositories**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bguardian/
↪@version@/bullseye-64/ bullseye bguardian
```

1. Run apt update:

Listing 7: **APT update**

```
apt update
```

3. Install the plugin using apt install:

Listing 8: **APT install**

```
apt install bacula-enterprise-bguardian-dir-plugin
```

Result

BGuardian Plugin is installed. Click here for more details.

Result

The package installs the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-bguardian-dir-plugin-x.x.x.jar). Note that the version of the jar archive is not aligned with the version of the package. However, that version can be shown in the json reports.
- The bguardian shell script file in /opt/bacula/bin which invokes the jar files. The bguardian script searches for the most recent bacula-bguardian-dir-plugin-x.x.x.jar file in order to launch it, even though usually there should only be one file.

Configuration

The following chapter presents information on how to configure BGuardian.

Base Function

Basics

BGuardian needs to be able to connect to bdirjson tools, bconsole and the catalog.

BConsole and bdirjson commands are expected to be inside the /opt/bacula/bin directory. If the installation was done using custom paths, it will be needed to create symlinks to this location.

BGuardian uses the same kind of connection to the catalog that Bacula instance is using in the same host. Therefore, it will use the same credentials and database name that is configured int he 'Catalog' resource of the Director configuration.

BGuardian may be executed manually from the command line by running the script. Normally, if the connection to the database is using 'peer' mode it should be run by the bacula user:

Listing 9: **Admin Job**

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian
```

Note that /bin/bash may not be required if the bacula user can find the bash binary regularly.

It is recommended to run it from a Bacula Admin Job and schedule it once a day:

Admin Job example:

Listing 10: **Admin Job**

```
Job {
   Name = "BGuardian"
   Type = Admin
   Schedule = Daily0100
   JobDefs = JobDefault

     Runscript {
      RunsWhen = Before
      RunsOnClient = no
      Command = "sudo -u bacula /bin/bash /opt/bacula/bin/bguardian"

   }

}
```

Daily Schedule example:

Listing 11: **Schedule**

```
Schedule {
   Name = "Daily0100"
   Run = Level=Full daily at 01:00
}
```

BGuardian will run an analysis of all the configured services and produce some reports by default, as well as the corresponding alerts, that are stored as individual files. However, it also has a mode where it is possible to interact with the existing alerts in the system in order to show them, remove them or mark them to be ignored.

In general there is no need to change any parameter of BGuardian as it will work out of the box. However, it is possible to tune it in order to adjust the tests which are run and the results based on the specific environment it is being run in.

The daemon can receive parameters in 3 different forms:

1. Daemon parameters

The format is:

```
--parameter_name parameter_value
```

Note the '–' before parameter name and the space in between to put the value.

Example:

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian --dev_min_executions 10 --
↪max_days_copy 30
```

2. Parameters in a file

It's needed to use the special parameter:

```
--config_file "/opt/bacula/etc/bguardian.conf"
```

Then, the file may contain the parameters using:

```
par1=val1
par2=val2
par3=val3
...
```

Example:

Listing 12: **Events Configuration**

```
reports_keep_number=10
disable_events=true
configuration_checks_exclude=restore,verify,copy
services_exclude=successratio,restorefrequency
success_factor=0.7
```

3. Special alert commands

There are some commands regarding the alerts that have some shortcuts which may be invoked directly as:

```
# List active alerts in json format
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list

# List ignore alerts in json format
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_ignore

# List active alerts in text format
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_text

# List ignore alerts in text format
```

(continues on next page)

(continued from previous page)

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_ignore_text

# Add ignore
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore "code"

# Remove ignore
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian remove_ignore "code"

# Remove alert
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian remove_alert "code"
```

BGuardian can send events regarding the alerts that it generates, which is recommended. To benefit from this feature, it is necessary to enable events in the proper message resource, adding the special ‘events’ keyword to the configuration:

Events configuration example:

Listing 13: **Events Configuration**

```
Messages {
     Name = Standard
     mailcommand = "/tmp/regress/bin/bsmtp -h localhost -f \"\(Bacula␣
↪regression\) %r\" -s \"Regression: %t %e of %c %l\" %r"
     operatorcommand = "/tmp/regress/bin/bsmtp -h localhost -f \"\(Bacula␣
↪regression\) %r\" -s \"Regression: Intervention needed for %j\" %r"
     console = all, !skipped, !terminate, !restored, events
     append = "/tmp/regress/working/log" = all, !skipped, events
     catalog = all, !skipped, events
}
```

Parameters

One of the most important reasons to modify the default parameters is to select the services to include or exclude during the execution of BGuardian. By default, all services are included.

Additionally, some services allow sub-checks and those may also be excluded. The parameters to control these two features are:

- service: For the main services
- configuration_checks_exclude: To exclude checks from the configuration security service

After selecting the services to apply, there are also parameters which can control the results of those services.

Described below are all services and all of their specific parameters, as well as what actions are recommended to take if results are detected for each of them.

Fileset Common Parameters

The following parameters are applicable to the general behavior of the plugin:

Option	Requir	Default	Values	Example	Description
mod	No	check	check, alert	alert	Run normal check mode or alert mode to query current alerts or add/remove alerts or ignores
service	No	configurationsecurity, infected, securityevents, deviation, successratio, failedinarow, empty, lastgood, nocopy, noverify, lowdedup, orphanchain, restorefrequency, differentfilesystem, nototp, objectlastgood	List (separated by ',') of elements from: configurationsecurity, infected securityevents, deviation, successratio, failedinarow, empty, lastgood, nocopy, noverify, differentfilesystem, lowdedup, orphanchain, restorefrequency, differentfilesystem, nototp, objectlastgood	configurationse curity, nocopy	Select the services that BGuardian will execute
service_	No		List (separated by ',') of elements from: configurationsecurity, infected, securityevents, deviation, successratio, failedinarow, empty, lastgood, nocopy, noverify, lowdedup, orphanchain, restorefrequency, differentfilesystem, nototp, objectlastgood	configurationse curity, nocopy	Select the services that BGuardian will exclude. Use this variable to exclude a list or the 'service' one to include a list, but do not use both
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/b	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset ings
log	No		An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be be stored in the working directory

Configuration Security Service

This service is activated if the service parameter contains the keyword: **configurationsecurity**.

Alert code is: **GC__[SUBSERVICE]**. Subservice codes are detailed below.

The purpose of this service is to report the result of different checks regarding security, status and best practices related to how Bacula is configured and running in the environment.

Option	Re-quire	De-fault	Values	Example	Description
max_days_l	No	10	Integer	5	Maximum number of days without a backup of the configuration or catalog before generating an alert
max_days_c	No	15	Integer	3	Maximum number of days without any copy job before generating an alert
max_days_v	No	30	Integer	30	Maximum number of days without any verify job before generating an alert
max_days_r	No	30	Integer	60	Maximum number of days without any restore job before generating an alert
max_days_v	No	7	Integer	2	Maximum number of days without running vacuum in dedup enabled storage daemons
dedup_free_	No	10737418	Long	536870912000	Limit on bytes of free space for dedup engines before raising the alert
permis-sions_base_	No	/opt/bacu	List of paths	/opt/bacula/etc	List of paths of bacula installation files where checking permissions
con-fig_backup_	No	Bacu-laDi-rector-Con-figs	Job Name	BaculaCon-fig	Name of the job of the Backup of the configuration of Bacula, to an-alyze if it's regularly run
cata-log_backup	No	(auto-detected)	Job Name	BaculaCata-log	Name of the job of the Backup of the catalog of Bacula, to analyze if it's regularly run
min_free_s	No	10	Integer	5	Minimum percentage of free space for a given Storage Daemon before generating an alert
con-figura-tion_checks	No		*List of checks separated by ',' (see next point)	restore, ver-ify, copy, malware, antivirus	List of subchecks to exclude from the execution of this Configuration security service

Configuration Security Service Subchecks

Configuration security is a special service of BGuardian that checks many things related with the configuration of the environment.

By default, it will check everything, but it is possible to exclude some checks using the **configuration_checks_exclude** parameter and adding a list of keywords there (separated by ‘,’).

Below we briefly describe the keyword and the function of each of the services and what is the recommended action if a related issue is reported. The code serves for the alerts functionality and to quickly identify each issue:

- **passwords**: Checks duplicated passwords and the strength of them inside Bacula Director configuration.
  - **Code**: GC__PASSWOR
  - **Action**: Make your passwords unique and use a strong keyword (+8 characters, include upper and lowercase, digits and symbols)
- **catalog_backup**: Checks configuration and recent execution (max_days_bacula_backup parameter) of the backup of the catalog of Bacula (configurable by config_backup_job_name).
  - **Code**: GC__CATALOG
  - **Action**: Immediately run a backup of the catalog and make sure your schedule is frequent enough (once a week at least)
- **config_backup**: Checks configuration and recent execution (max_days_bacula_backup parameter) of the backup of the configuration of Bacula (configurable by catalog_backup_job_name).
  - **Code**: GC__CONFIG_
  - **Action**: Immediately run a backup of the catalog and make sure your schedule is frequent enough (once a week at least)
- **restore**: Checks execution of some recent restore (max_days_restore parameter). It is a best practice to run some restore of the different kinds of data from time to time.
  - **Code**: GC__RESTOR
  - **Action**: Run some restore for each kind of data and each kind of storage from time to time.
- **verify**: Checks execution of some recent verify job (max_days_verify parameter). It is a best practice to verify the data of your jobs.
  - **Code**: GC__VERIFY
  - **Action**: Configure verify jobs for your data
- **copy**: Checks execution of some recent restore (max_days_copy parameter). It is a best practice to use a multi-tier strategy with your backups.
  - **Code**: GC__COPY
  - **Action**: Define a second storage tier and configure copy jobs to send your data there.
- **malware**: Checks jobs that could activate the Malware protection function, but have not enabled it.
  - **Code**: GC__MALWAR
  - **Action**: Enable Malware detection for any reported system that could be suitable to be infected because its location, usage pattern and data kind

- **antivirus**: Checks the existence of one antivirus job for each client.
  - **Code**: GC__ANTIVI
  - **Action**: Configure an antivirus job for your clients, specially for file servers.
- **consoles**: Checks the usage of restricted consoles.
  - **Code**: GC__CONSOL
  - **Action**: If you have different users accessing your Bacula environment, configure a restricted console for each of them, using only the minimal needed permissions
- **events**: Checks the activation of Events in Message resources for auditing purposes.
  - **Code**: GC__EVENTS
  - **Action**: Enable events messages in your configuration and store them at your convenience (it is recommended to store them in a file and also in the catalog)
- **dir_status**: Checks the status of the Director daemon to see if there is any reported error with the service.
  - **Code**: GC__DIR_ST
  - **Action**: Review the status of your Director service and start or restart it
- **dir_address**: Checks the usage of DirAddress setting to limit Director service to be listening on specific interfaces
  - **Code**: GC__DIR_AD
  - **Action**: Use the DirAddress setting, so you limit the service to be listening only on the required interface
- **sd_status**: Checks the status of the Storage Daemon(s) to see if there is any reported error with the service or if there is no connectivity with some of them.
  - **Code**: GC__SD_STA
  - **Action**: Review the status of the affected Storage Daemon and the connectivity to it from the Director host. Start or restart the service if needed
- **sd_free**: Checks free space in each Storage Daemon is above the threshold (defined by min_free_space_percent parameter).
  - **Code**: GC__SD_FREE
  - **Action**: Review the status of the affected Storage Daemon and start or restart it
- **dedup**: Enable getting dedup status for Storage Daemons in order to make dedup checks around Global Endpoint Deduplication. (Requires to not exclude sd_status)
- **ded_errors**: Check if GED is reporting some general error. (Requires to not exclude dedup)
  - **Code**: GC__DED_ER
  - **Action**: Review the message and the status of the affected Storage Daemon. Restart it, run vacuum procedures and re-check
- **ded_orphan**: Check if GED is reporting some orphan reference. (Requires to not exclude dedup)
  - **Code**: GC__DED_OR
  - **Action**: Run vacuum as soon as possible. If it is not solved, run scrub process.

- **ded_vacuum**: Check if GED vacuum procedure was executed recently enough. The number of days can be controlled with max_days_vacuum parameter. (Requires to not exclude dedup)
  - **Code**: GC__DED_VA
  - **Action**: Run vacuum as soon as possible.
- **ded_idx**: Check if GED is marking some error with the indexes. (Requires to not exclude dedup)
  - **Code**: GC__DED_ID
  - **Action**: Review the status of the filesystem holding the indexes.
- **ded_miss**: Check if GED is marking some missed reference. (Requires to not exclude dedup)
  - **Code**: GC__DED_MI
  - **Action**: Run vacuum as soon as possible. If it is not solved, run scrub process.
- **ded_free**: Check the free space available in GED engine (it relies on parameter dedup_free_space_min_bytes). (Requires to not exclude dedup)
  - **Code**: GC__DED_FR
  - **Action**: Provide more space to your GED containers filesystem. You can also purge data from your dedup storages and then run vacuum process to try to recover some space.
- **ded_suspect**: Check if GED is reporting some suspect reference. (Requires to not exclude dedup)
  - **Code**: GC__DED_SU
  - **Action**: Run vacuum as soon as possible. If it is not solved, run scrub process.
- **ded_derr**: Check if GED is reporting errors in the dedup engine. (Requires to not exclude dedup)
  - **Code**: GC__DED_DE
  - **Action**: Run vacuum as soon as possible. If it is not solved, run scrub process.
- **ded_cerr**: Check if GED is reporting errors in the containers. (Requires to not exclude dedup)
  - **Code**: GC__DED_CE
  - **Action**: Run vacuum as soon as possible. If it is not solved, run scrub process.
- **fd_status**: Checks the status of the client File Daemon(s) to see if there is any reported error with the service or if there is no connectivity with some of them.
  - **Code**: GC__FD_STA
  - **Action**: Review the status of the affected File Daemon and the connectivity to it from the Director host. Start or restart the service if needed
- **fips**: Checks if FIPS is enabled on daemons supporting it (for DIR requires to not exclude 'dir_status' ; for FDs requires to not exclude 'fd_status' ; for SDs requires to not exclude 'sd_status').
  - **Code**: GC__FIPS
  - **Action**: Consider enabling FIPS to the affected Daemon if your security posture needs to be very high
- **trace**: Checks if trace is enabled in any daemon with the risk of fulling a disk (for DIR requires to not exclude 'dir_status' ; for FDs requires to not exclude 'fd_status' ; for SDs requires to not exclude 'sd_status').
  - **Code**: GC__TRACE

  - **Action**: Disable debug and trace from the affected Daemon as soon as possible if you are not doing debug activities anymore

- **versions**: Checks if the FDs and/or SDs Bacula versions are aligned with the version of the Director (for FDs requires to not exclude 'fd_status' ; for SDs requires to not exclude 'sd_status').
  - **Code**: GC__VERSIO
  - **Action**: Install a supported Bacula version in the affected Daemon. Storage Daemon and Director must be on the same version, while File Daemons can run an older version than the Director.
- **security_plugin**: Checks if the security plugin is deployed in each FD (requires to not exclude 'fd_status').
  - **Code**: GC__SECURI
  - **Action**: Install the security plugin in the affected File Daemons
- **permissions**: Checks if permissions are strong enough for the given path and subpaths (permissions_base_paths parameter).
  - **Code**: GC__PERMIS
  - **Action**: Correct the permisisons on the affected paths. Usually you need to exclude the 'others' group from any bacula directory and to protect the bacula configuraton from undesiderable writes.
- **running_processes**: Checks if running processes are running with root user, which is generally not recommended for secure environments.
  - **Code**: GC__RUNNIG
  - **Action**: Configure your daemos with the correct user. Director and Storage Daemon do not need to be run with root.
- **encryption**: Check if encryption is used at all in the environment
  - **Code**: GC__ENCRYP
  - **Action**: Consider using encryption for any sensitive data or any untrusted storage.
- **volprotection**: Check if volume protection is used at all in the environment
  - **Code**: GC__VOLPRO
  - **Action**: Consider enabling volume protection for you disk backup over linux, as well as any backup sent to NAS from NetApp, DataDomain or HPE StoreOnce.
- **pg_vacuum**: Check if PostgreSQL vacuum process was executed recently enough on the key tables
  - **Code**: GC__PG_VAC
  - **Action**: Run Vacuum on the affected tables as soon as possible, during a low load window in your environment.
- **pg_analyze**: Check if PostgreSQL analyze process was executed recently enough on the key tables
  - **Code**: GC__PG_ANA
  - **Action**: Run Analyze on the affected tables as soon as possible, during a low load window in your environment.
- **pg_config**: Check if PostgreSQL configuration parameters are inside the recommended margins

- **Code**: GC__PG_CON
- **Action**: Correct the mentioned values to comply with the recommended configuration

Infected Service

This service is activated if the service parameter contains the keyword: **infected**.

Alert code is: **GIN**

The purpose of this service is to report jobs where some virus, ransomware or malware were detected, so a summary of them is easily available while a new alert for any new entry will also be generated.

Action: If you find any job containing some kind of malware or virus you should quickly isolate that system from your network and run healing activities on it. After, run a new backup and check that no more virus or malware are detected.

Security Events Service

This service is activated if the service parameter contains the keyword: **securityevents**.

Alert code is: **GSE**

This service will report any recent event registered in Bacula core with the security category.

Option	Required	Default	Values	Example	Description
securityevents_days_si	No	15	Integer	30	Defines the number of days to consider, from today, for the report of security events

Action: Review the nature of the event and act in consequence. If you find, for instance, many failed attempts to connect to the Director from BConsole, consider to change the location of your consoles or improve the security posture at networking level for them.

Deviation Service

This service is activated if the service parameter contains the keyword: **deviation**.

Alert code is: **GDV**

The purpose of this service is to analyze job executions statistically and find deviation from the expected values. Calculations are done over the size, number of files and duration of the jobs.

Depending on what kind of jobs and the nature of data of your environment, you may need to adjust the following parameters to maximize the utility of the deviation results. It is possible to adjust the different thresholds, as well as to decide if results should only be listed following regression deviation and deviation from average (default behavior) or exclude deviation from average if it is generating too much information and it is not pointing to issues in the environment (dev_include_by_avg parameter).

Parameters for deviation service are explained below:

Option	Require	Default	Values	Example	Description
dev_f	No	0.4	Float	0.25	Defines what jobs that will trigger the alert of deviation. It means what relation with the calculated standard deviation is considered significant enough. 1 means 200% of the standard deviation, 0.5 means 150% of the standard deviation. Example: If the standard deviation for size is 100Mb, with a value of 0.5: A job with a deviation from the average or regression of 160Mb will be selected, a job with a deviation from the average of 50Mb won't be selected, a job with a deviation from the average of 120Mb won't be selected
dev_s	No	0.5	Float	0.8	Defines the limit to consider a selected deviated job as severity Low
dev_s	No	0.75	Float	0.9	Defines the limit to consider a selected deviated job as severity Medium
dev_ı	No	0.5	Float	0.4	Minimum value of regression accuracy in order to use the regression analysis
dev_ı	No	0.8	Float	2	Minimum deviation from the average to consider a job as deviated for selection based on average
dev_ı	No	1200	Integer	3600	Minimum duration of a job in order to consider deviation by duration as something significant to select the job
dev_ı	No	5	Integer	20	Minimum number of executions of a given job in order to consider it for deviation analysis
dev_i	No	true	1, true, yes, Yes ; 0, false, no, No	false	Enable/disable selection of results based on the average deviaton
dev_ı	No	50	Integer	100	Minimum nuber of files in a job in order to consider deviation by number of files as something signficatn to select the job
dev_ı	No	1048	Integer	5242	Minimu size of a job in order to include it in deviation analysis

Action: When this service reports results, you should review every result and analyze what caused the given deviation. A deviation can be caused by many reasons, like sudden new information to backup, a sudden slowness problem, some controlled massive deletion… However, the same effect can be caused from ransomware activities or even a not controlled or desired user activity. If you find such event, you will need to solve it and consider adjusting or re-runnnig some backups. If there is an explanation, every issue can be marked to be ignored in further alerts through the ignoring mechanism.

Failed in a row Service

This service is activated if the service parameter contains the keyword: **failedinarow**.

Alert code is: **GFR**

The purpose of this service is to report jobs that have failed N or more times in a row, according to the parameter failedrow_times.

Option	Required	Default	Values	Example	Description
failedrow_	No	3	Integer	5	Defines thresshold of times a given job failed in order to select it to be included in the report

Action: Review as soon as possible the affected jobs and analyze the causes of the failure. You can run them manually to check the result and then adjust the configuration or the schedule according to the results of your analysis.

Restore Frequency Service

This service is activated if the service parameter contains the keyword: **restorefrequency**.

Alert code is: **GRF**

The purpose of this service is to report jobs whose restore frequency is below the factor established by the parameter: restore_factor.

Option	Required	Default	Values	Example	Description
restore_facto	No	0.15	Float	0.5	Defines thresshold of restoring frequency for a given job in terms of %

Action: Review your backup policies and include some periodic restores on it, in order to ensure your backups and restore strategies are correct and agile enough to be correctly prepared for the time when an urgent restore comes.

Success Ratio Service

This service is activated if the service parameter contains the keyword: **successratio**.

Alert code is: **GSR**

The purpose of this service is to report jobs whose successratio is below the factor established by the parameter: success_factor.

Option	Required	Default	Values	Example	Description
success_factor	No	0.8	Float	0.75	Defines thresshold of success ratio for the executions of a given job in terms of %
success_severity_medi	No	0.4	Float	0.5	Defines the limit to consider a selected deviated job as severity Medium
success_severity_low_l	No	0.6	Float	0.7	Defines the limit to consider a selected deviated job as severity Low

Action: Review the affected jobs and analyze the causes of the failures. If they are apparently random, consider to run a load analysis over your system in order to spread better the load over your network and hosts.

No Copy Service

This service is activated if the service parameter contains the keyword: **nocopy**.

Alert code is: **GNC**

The purpose of this service is to report jobs not included in any 2-tier policy, which means jobs that have never been copied or migrated.

Option	Required	Default	Values	Example	Description
nocopy_grace_per	No	10	Integer	10	Jobs that are more recent than the days defined by this parameter won't be included in the report

Action: Review your backup policies and include a 2-Tier storage where sending the reported jobs through the configuration and execution of Copy jobs.

No Verify Service

This service is activated if the service parameter contains the keyword: **noverify**.

Alert code is: **GNV**

The purpose of this service is to report jobs not included in any verification policy, which means jobs that have never been verified.

Option	Required	Default	Values	Example	Description
noverify_grace_perioc	No	20	Integer	50	Jobs that are more recent than the days defined by this parameter won't be included in the report

Action: Review your backup policies and include a verification phase where you run periodic verify jobs. Include the listed jobs in the report.

Empty Service

This service is activated if the service parameter contains the keyword: **empty**.

Alert code is: **GE**

The purpose of this service is to report Full jobs that were successful but have no contents (no files and no bytes stored).

Action: Review the affected jobs, including the joblog and the configuration. You may need to adjust the configuration or to run again the affected jobs.

Last Good Service

This service is activated if the service parameter contains the keyword: **lastgood**. Alert code is: **GLG**

The purpose of this service is to report jobs where its last successful execution is older than the number of days specified by the parameter: lastgood_max_since_days.

Option	Re-quire	De-fault	Val-ues	Ex-am-ple	Description
last-good_max_si	No	5	In-te-ger	10	Jobs that have no successful execution that is more recent than the days defined by this parameter will be included into the report

Action: Review as soon as possible the affected jobs and their latest executions. Run them manually if they have been missed, cancelled or failed. Troubleshoot any problem if they are not successful even with the manual execution.

Last Object Good Service

This service is activated if the service parameter contains the keyword: **objectlastgood**. Alert code is: **GOLG**

The purpose of this service is to report objects where its last successful associated backup job execution is older than the number of days specified by the parameter: lastgood_max_since_days.

Please, note that this object uses the same parameter as the *Last good service*.

Option	Re-quire	De-fault	Val-ues	Ex-am-ple	Description
last-good_max_s	No	5	In-te-ger	10	Objects that have no successful protection that is more recent than the days defined by this parameter will be included into the report

Action: Review as soon as possible the affected objects and their associated jobs, to review their latest executions. Run them manually if they have been missed, cancelled or failed. Troubleshoot any problem if they are not successful even with the manual execution.

Low Dedup

This service is activated if the service parameter contains the keyword: **lowdedup**.

Alert code is: **GLD**

The purpose of this service is to provide a report of jobs that are not having good enough deduplication and that are potentially miss-using resources for information that are not a good candidate to be deduplicated. This information is also useful if running out of space and need to delete or migrate the information of some jobs that are using a good amount of storage.

Option	Require	Default	Values	Example	Description
dedup_r	No	40	Float	60	Defines thresshold of dedup ratio. Jobs with a lower ratio will be reported. The threshold represents a percentage and jobs store it as 'compressratio' in the catalog
dedup_s	No	5242	Long	26214	Defines the limit in size to consider a selected low dedup job. Jobs with smaller size won't be considered

Action: Consider disabling deduplication for the affected jobs if the ratio is very poor for your needs. If running out of space, consider deleting (or moving to a different storage with a Migration) large jobs with a poor ratio that are old enough for your needs.

Orphan Chain

This service is activated if the service parameter contains the keyword: **orphanchain**.

Alert code is: **GOC**

Incremental and Differential jobs are part of a chain, and they are dependent on the information of a previous job. Sometimes, due to human mistakes or due to a bad retention policy, chains can be broken, and a dependent job is recycled before an after one. This service will detect this situation and report jobs that are 'orphan' in these terms.

It is important to note that depending on the backup nature, Incremental or Differential jobs alone can be still useful. For instance, for any job that contains files, the information is still fully recoverable, and they can also be based on a previous older Full or Incremental having the restore job still working. However, for some Virtual Machine or Database plugins, it is possible that one Incremental or Differential job without their predecessor will not contain recoverable information. In general, it is important to try to avoid having any orphan job and this service is intended to help in that direction.

Action: If you ever detect an orphan job, review your backup policies regarding retention times and adjust them, if necessary, to not be automatically producing any orphan job. Check also for the affected jobs that you have other valid copies of the information, if you don't, run new jobs as soon as possible.

Different filesystem

This service is activated if the service parameter contains the keyword: **differentfilesystem**.

Alert code is: **GDFS**

This service will detect and report jobs where the log message 'xxx is a different filesystem. Will not descend from yyyy' is produced. Paths reported there will be compared with the list of excluded 'knonw' paths configured by 'different_fs_exclude' parameter. If they do not match, jobs will be reported.

This situation happens when jobs are using filesets with OneFS option enabled. Depending on the environment, this behavior is absolutely desirable, but can hide some non desired exclussion. This service helps to avoid those kind of situations.

Option	Require	Default	Values	Example	Description
different_fs_	No	/proc, /sys, /tmp, /boot	List of paths separated by ','	/proc, /sys, /tmp, /boot, /myNonDesiredFS2, /mnt/myNonDesiredFS	List of known paths that are on different filesystems and there is no problem if they are reported as paths that won't be backed up

Action: Review the path that was excluded and consider modifying your backup configuration if it is necessary to include it, or add it to the list to be excluded on this service otherwise.

NoTOTP Service

This service is activated if the service parameter contains the keyword: **nototp**.

Alert code is: **GNT**

The purpose of this service is to report users that have not enabled TOTP 2-Tier authentication mechanism.

Action: Consider to enable the TOTP 2-Tier authentication mechanism for the reported users in order to improve your security posture regarding BWeb access.

BWeb

BGuardian is connected to BWeb in two forms.

The first of them is about the links that generates automatically to be able to open the joblogs of the reported job entries in the html reports. It also generates links to job configuration pages from any entry that is a job name. This feature can be disabled with the parameter: bweb_jobid_link

The second is the ability to open directly the html report from BWeb. To accomplish this goal, BGuardian can generate a symlink to the reports directory inside the proper BWeb directory, as it is shown below:

Listing 14: **Reports symlink**

```
# ls -l /opt/bweb/html/.reports
lrwxrwxrwx 1 root root 39 jul 28 11:12 /opt/bweb/html/.reports -> /tmp/
↪regress/working/bguardian/.reports
```

That symlink can be created manually, but if we run once BGuardian with the root user, the symlink will also be created. Note that if you run BGuardian from an AdminJob as we recommend, BGuardian will be executed by the 'bacula' user.

Once BGuardian detects the presence of BWeb and the symlink, the output of the script, available in the AdminJob joblog will show this kind of lines:

Listing 15: **HTML report in BWeb**

```
Open html report from: https://your.bweb.host.name:9180/.reports/Report__2023-
↪07-28_051730.html
```

As a result, you can copy that URL and directly see the report in your web browser.

Future versions of BGuardian and BWeb will expand this integration with more features, like listing the available reports and allowing to directly click on the links to see them.

Operations

The following article describes details regarding the different operations of **Bacula Enterprise BGuardian Plugin**, which essentially are to generate detailed reports and alerts with grouped information coming from the details of those reports.

Reports

Once BGuardian completes the different analysis that has been configured for it will produce a report of what has been found in different formats:

- It will directly output the detailed information to STDOUT in human-friendly text format. If it was invoked from a job, these output will be visible in the joblog.
- It will produce a computer-friendly report inside the configured reports_path, this is in json format.
- It will produce a friendly HTML report with the essential information inside the configured reports_path.

Text and json reports will contain the following data:

- A summary of the services that were run
- Produced errors, if any
- Version of BGuardian
- Date of the report
- Summary of the configuration used
- Summary of alerts generated
- List of issues found organized by service
- Totals for ignored results, passed checks and alerts
- Paths of the generated Reports

In general, the format of the issues is structured like this:

Listing 16: **Events Configuration**

```
Severity | Code | Entity+Details | Description.
```

Where:

- **Severity**: Reflects the relative relevance of the issue. It can be High, Medium or Low
- **Code**: Identifies the issue in a unique form. This code can be used to call the alerts function and ignore the issue in future executions
- **Entity** and **details** will represent the element affected by the issue (usually a job, a daemon name or a user).
- **Description**: Shows a message describing the situation

Below there is an output text example:

Listing 17: **BGuardian Example**

```
$ sudo -u bacula /bin/bash /opt/bacula/bin/bguardian
Cleaning old reports in /tmp/regress/working/bguardian/.reports...
Running service: configurationsecurity
Running service: deviation
Running service: successratio
Running service: failedinarow
Running service: empty
Running service: nocopy
Alert: GNC__guardianjob partially recovered:GuardianJob
Alert: GNC__guardianjob recovered
Running service: noverify
Alert: GNV__guardianjob partially recovered:GuardianJob
Alert: GNV__guardianjob recovered
Running service: restorefrequency
Running service: nototp
bweb_user not found
======================== BGUARDIAN Report ========================
Version: 1.0.0
Report Date: 2023-06-16 12:23:21
================ Config ================
ALERT_OPERATION : LIST
MODE : CHECK
REPORTS_KEEP_NUMBER : 100
SUCCESS_SEVERITY_LOW_LIMIT : 0.6
DEV_MIN_EXECUTIONS : 5
DEV_INCLUDE_BY_AVG : true
...
========================================
========== Active alerts ===========
GC__CATA | LOW | Service: configurationsecurity | Entity: catalog_backup
GC__CONF | LOW | Service: configurationsecurity | Entity: config_backup
GC__CONS | LOW | Service: configurationsecurity | Entity: consoles
GC__COPY | LOW | Service: configurationsecurity | Entity: copy
GC__EVEN | LOW | Service: configurationsecurity | Entity: events
GC__MALW | LOW | Service: configurationsecurity | Entity: malware
```

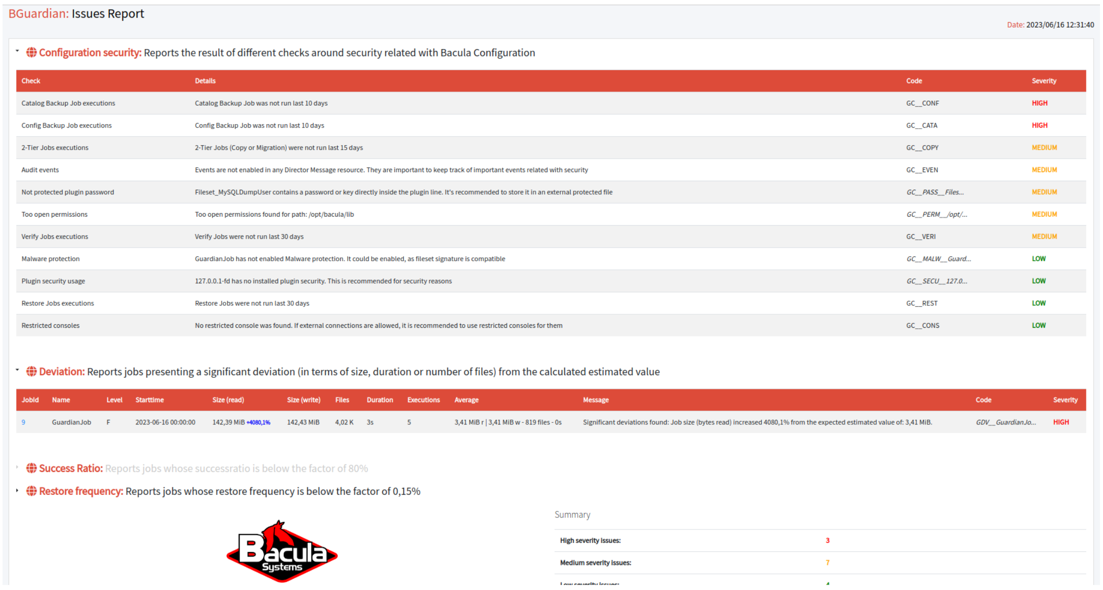
(continues on next page)

(continued from previous page)

```
GC__PASS | LOW | Service: configurationsecurity | Entity: passwords
GC__PERM | LOW | Service: configurationsecurity | Entity: permissions
GC__REST | LOW | Service: configurationsecurity | Entity: restore
GC__SECU | LOW | Service: configurationsecurity | Entity: security_plugin
GC__VERI | LOW | Service: configurationsecurity | Entity: verify
GRF__guardianjob | LOW | Service: restorefrequency | Entity: guardianjob
GSR__guardianjob | LOW | Service: successratio | Entity: guardianjob
=====================================
############### Service: Configuration security ###############
HIGH | GC__CONF | Catalog Backup Job executions : Catalog Backup Job was not␣
↪run last 10 days
HIGH | GC__CATA | Config Backup Job executions : Config Backup Job was not␣
↪run last 10 days
MEDIUM | GC__COPY | 2-Tier Jobs executions : 2-Tier Jobs (Copy or Migration)␣
↪were not run last 15 days
MEDIUM | GC__EVEN | Audit events : Events are not enabled in any Director␣
↪Message resource. They are important to keep track of important events␣
↪related with security
MEDIUM | GC__PASS__Fileset_MySQLDumpUser | Not protected plugin password :␣
↪Fileset_MySQLDumpUser contains a password or key directly inside the plugin␣
↪line. It's recommended to store it in an external protected file
MEDIUM | GC__PERM__/opt/bacula/lib | Too open permissions : Too open␣
↪permissions found for path: /opt/bacula/lib
MEDIUM | GC__VERI | Verify Jobs executions : Verify Jobs were not run last 30␣
↪days
LOW | GC__MALW__GuardianJob | Malware protection : GuardianJob has not␣
↪enabled Malware protection. It could be enabled, as fileset signature is␣
↪compatible
LOW | GC__SECU__127.0.0.1-fd | Plugin security usage : 127.0.0.1-fd has no␣
↪installed plugin security. This is recommended for security reasons
LOW | GC__REST | Restore Jobs executions : Restore Jobs were not run last 30␣
↪days
LOW | GC__CONS | Restricted consoles : No restricted console was found. If␣
↪external connections are allowed, it is recommended to use restricted␣
↪consoles for them
############### Service: Success Ratio ###############
LOW | GSR__GuardianJob | Job: GuardianJob | Executions: 11 | Ratio: 63,6%
############### Service: Restore frequency ###############
MEDIUM | GRF__GuardianJob | Job: GuardianJob | Executions: 7 | Restores: 0 |␣
↪Ratio: 0
========================================
Ignored results: 0
Passed checks: 11
Alerts: 13
==================================================================
Json report built in: /tmp/regress/working/bguardian/.reports/Report__2023-06-
↪16_122324.json
Html report built in: /tmp/regress/working/bguardian/.reports/Report__2023-06-
↪16_122324.html
```

The HTML report will reflect the issue information in a summarized way, with collapsible blocks with a more friendly format. It is also possible to see a summary of the issues grouped by severity.

Here we show an example HTML report:



BGuardian: Issues Report

Date: 2023/06/16 12:31:40

Configuration security: Reports the result of different checks around security related with Bacula Configuration

Check	Details	Code	Severity
Catalog Backup Job executions	Catalog Backup Job was not run last 10 days	GC__CONF	HIGH
Config Backup Job executions	Config Backup Job was not run last 10 days	GC__CATA	HIGH
2-Tier Jobs executions	2-Tier Jobs (Copy or Migration) were not run last 15 days	GC__COPY	MEDIUM
Audit events	Events are not enabled in any Director Message resource. They are important to keep track of important events related with security	GC__EVEN	MEDIUM
Not protected plugin password	Fileset_MySQLDumpUser contains a password or key directly inside the plugin line. It's recommended to store it in an external protected file	*GC__PASS__Files...*	MEDIUM
Too open permissions	Too open permissions found for path: /opt/bacula/lib	*GC__PERM__/opt/...*	MEDIUM
Verify Jobs executions	Verify Jobs were not run last 30 days	GC__VERI	MEDIUM
Malware protection	GuardianJob has not enabled Malware protection. It could be enabled, as fileset signature is compatible	*GC__MALW__Guard...*	LOW
Plugin security usage	127.0.0.1-fd has no installed plugin security. This is recommended for security reasons	*GC__SECU__127.0...*	LOW
Restore Jobs executions	Restore Jobs were not run last 30 days	GC__REST	LOW
Restricted consoles	No restricted console was found. If external connections are allowed, it is recommended to use restricted consoles for them	GC__CONS	LOW

Deviation: Reports jobs presenting a significant deviation (in terms of size, duration or number of files) from the calculated estimated value

JobId	Name	Level	Starttime	Size (read)	Size (write)	Files	Duration	Executions	Average	Message	Code	Severity
9	GuardianJob	F	2023-06-16 00:00:00	142,39 MiB +4080,1%	142,43 MiB	4,02 K	3s	5	3,41 MiB r | 3,41 MiB w - 819 files - 0s	Significant deviations found: Job size (bytes read) increased 4080,1% from the expected estimated value of: 3,41 MiB.	*GDV__GuardianJo...*	HIGH

Success Ratio: Reports jobs whose successratio is below the factor of 80%

Restore frequency: Reports jobs whose restore frequency is below the factor of 0,15%

Summary

High severity issues:	3
Medium severity issues:	7
Low severity issues:	4

Fig. 7: BGuardian HTML Report

For the deviation service, not that it will visually mark what value (from files, size or time) has been increased or decreased significantly enough to select and include the job into the report.

In future versions of Bacula, the Web User Interface will interpret this information and also make it directly accessible through the Web layer.

BGuardian will generate one json report and one html report for every execution by default. By default, will keep 100 reports of each kind before removing the oldest ones. This report rotation capability can be adjusted with the parameter reports_keep_number.

Alerts

On top of the reporting features described in the Reports section BGuardian also implements an Alert framework that will keep track of those issues through executions of the tool through the time. The purpose of this framework is to easy the management of the different issues and to provide the proper functions to see the status of an environment in a given point in time when BGuardian is run regularly.

Alerts will group the information of the issues by bigger entities. For example, the same job can present many records inside the deviation service, with different executions of it. In the issue report we will see individually each entry, while in the alerts framework we will only have a single alert for that jobname, grouping all the affected executions.

BGuardian allows to precisely select the services that are desired to be run. However, the backup administrator will find situations where the service is still interesting to be run, while there are some records that he/she acknowledged already and does not desire to see anymore in the reports or in the alert lists. Here is where the ignore feature comes in place. Using the code of a given issue (or part of it in some situations) it is possible to mark an issue or a group of them to be ignored in further executions.

In this section, the alerts structure, the different commands and soem examples of them are shown

Alerts structure

Alerts are store in the path defined by 'alerts_path' parameter. Each alert is represented by .json file, whose name is the code of the alert and the kind of it.

Listing 18: **Alerts files**

```
$ ls -l
total 56
-rw-rw-r-- 1 bac bac  343 jun 16 12:31 GC__CATA.on.json
-rw-rw-r-- 1 bac bac  342 jun 16 12:31 GC__CONF.on.json
-rw-rw-r-- 1 bac bac  396 jun 16 12:31 GC__CONS.on.json
-rw-rw-r-- 1 bac bac  324 jun 16 12:31 GC__COPY.on.json
-rw-rw-r-- 1 bac bac  394 jun 16 12:31 GC__EVEN.on.json
-rw-rw-r-- 1 bac bac  401 jun 16 12:31 GC__MALW.on.json
-rw-rw-r-- 1 bac bac  482 jun 16 12:31 GC__PASS.on.json
-rw-rw-r-- 1 bac bac  371 jun 16 12:31 GC__PERM.on.json
-rw-rw-r-- 1 bac bac  312 jun 16 12:31 GC__REST.on.json
-rw-rw-r-- 1 bac bac  406 jun 16 12:31 GC__SECU.on.json
-rw-rw-r-- 1 bac bac  310 jun 16 12:31 GC__VERI.on.json
-rw-rw-r-- 1 bac bac 1298 jun 16 12:31 GDV__guardianjob.on.json
-rw-rw-r-- 1 bac bac  219 jun 16 12:31 GRF__guardianjob.on.json
-rw-rw-r-- 1 bac bac  239 jun 16 12:31 GSR__guardianjob.on.json
```

The 'on.json' extension represents an active alert, while the 'off.json' extension represents an alert that will be ignored in future executions.

List alerts

The following command lists active alerts:

Listing 19: **Active alerts**

```
# Text mode
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_text

# Json mode
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list
```

Example output in text format:

Listing 20: **Active alerts**

```
GC__CONF | LOW | Service: configurationsecurity | Entity: config_backup |␣
↪Details: {"CONFIG_BACKUP":{"passed":false,"description":"Catalog Backup Job␣
↪executions","details":"Catalog Backup Job was not run last 10 days",
↪"subCheck":"CONFIG_BACKUP","severity":"HIGH"}}
GC__CONS | LOW | Service: configurationsecurity | Entity: consoles |␣
↪Details: {"CONSOLES":{"passed":false,"description":"Restricted consoles",
↪"details":"No restricted console was found. If external connections are␣
↪allowed, it is recommended to use restricted consoles for them","subCheck":
↪"CONSOLES","severity":"LOW"}}
```

The structure of an alert is similar to the structure of an issue. It is composed by:

Listing 21: **Alert structure**

```
Code | Severity | BGuardian service | Affected entity | Issue details
```

The following command lists ignore alerts:

Listing 22: **List Ignore alerts**

```
# Text mode
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_ignore_text

# Json mode
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian list_ignore
```

Ignore alerts

The following command adds 'ignores', which means that issues matching the ignore code will be ignored from active alerts and from further BGuardian executions.

Listing 23: **Ignore an alert**

```
# One single code
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore code1

# Several codes at once
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore code1, code2, code3
```

Using the codes shown in list alerts, we could ignore them with the following commands

Listing 24: **Ignore some alerts**

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore GC__CONF, GC__CONS
```

There are services that have entities, but also more information. For example the deviation service works with jobnames and with job ids. Example:

Listing 25: **Deviation issue**

```
 ############### Service: Deviation ###############
 HIGH | GDV__GuardianJob__9 | Job: 9 GuardianJob F 2023-06-16 00:00:00
     | Size (read): 142,39 MiB  +4080,1%  | Size (write): 142,43 MiB
     | Files: 4,02 K | Duration: 3s
     | Executions: 5 | Average: 3,41 MiB r | 3,41 MiB w - 819  files - 0s
     | Details: Significant deviations found: Job size (bytes read) increased␣
↪4080,1% from the expected estimated value of: 3,41 MiB.
```

It is possible to ignore here any execution of the GuardianJob. To do so:

Listing 26: **Ignore alert by entity**

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore GDV__GuardianJob
```

However, it is also possible to ignore only particular jobids:

Listing 27: **Ignore alert by id**

```
 sudo -u bacula /bin/bash /opt/bacula/bin/bguardian ignore GDV__GuardianJob__9
```

Doing so, new deviated results of the same job will be still considered in next executions.

Manually remove alerts

It is possible to remove active alerts using a very similar format:

Listing 28: **Remove active alert**

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian remove_alert GDV__
↪GuardianJob__9
```

To remove something from the ignore list:

Listing 29: **Remove ignore**

```
sudo -u bacula /bin/bash /opt/bacula/bin/bguardian remove_ignore GC__CATA
```

Alerts recovery

When a situation marked by a given issue is solved, BGuardian will automatically remove the associated alert.

Events

In order to notify the backup administrator when an alert is created or recovered, BGuardian uses the Events feature. This means it will send an Message of type Event with the information of what happened.

Example of BGuardian event about permissions recovery (source 'bguardian'):

Listing 30: **Events**

```
 *list events
 Automatically selected Catalog: MyCatalog
 Using Catalog "MyCatalog"
 +---------------------+---------------+---------------+----------------------
↪-+-----------------------------------------+
 | time                | daemon        | source        | type                 ␣
↪ | events                                  |
 +---------------------+---------------+---------------+----------------------
↪-+-----------------------------------------+
 | 2023-06-16 13:20:57 | 127.0.0.1-dir | *Director*    | daemon               ␣
```

(continues on next page)

(continued from previous page)

```
↪ | Director configuration reloaded          |
 | 2023-06-16 13:21:34 | 127.0.0.1-dir | *Console*     | connection          ␣
↪ | Connection from 127.0.0.1:8101           |
 | 2023-06-16 13:21:34 | 127.0.0.1-dir | **bguardian** |␣
↪configurationsecurity | BGuardian alert [GC__PERM] was recovered |
 | 2023-06-16 13:21:34 | 127.0.0.1-dir | *Console*     | connection          ␣
↪ | Disconnection from 127.0.0.1:8101        |
 | 2023-06-16 13:21:37 | 127.0.0.1-dir | *Console*     | connection          ␣
↪ | Connection from 127.0.0.1:8101           |
 +---------------------+---------------+---------------+---------------------
↪-+------------------------------------------+
```

Note that in order to have events feature working, it is needed to enable them in the Message resources as discussed in the Configuration section.

Best Practices

The following article presents best practices regarding usage and performance.

Usage

BGuardian performs a good number of different checks. Ideally, a backup environment would comply with all the rules of those checks and no issue or alert would be detected. However, in many cases, there are some external constrains around resources or about the nature of the data that can make it difficult to comply with some of them.

The main goal of this tool is to help the administrator to keep the system safe and to detect any non-desired behavior. To reach this goal, the tool should generate alerts or issues only when they are actually something that the administrator is going to change or review. This will be only possible if services producing results that will not be actually solved are deactivated, as well as marking those specific alerts that cannot either be solved as something to ignore.

The recommended usage cycle with this tool is:

1. Run it manually one time with default parameters.
2. From the reported information, select the services that are relevant for the given environment.
3. Configure BGuardian with the selection of relevant services.
4. Tune selected services, if needed, to reduce the number of results (adjust dates, factors…).
5. Configure an Admin Job to invoke BGuardian with all the resulting command line parameters needed as discovered in the previous steps.

Once the periodical Admin Job is in place, the user should be notified with the events of every alert creation. Then he should review the situation of a given alert and solve the problem if necessary, or mark the alert to be ignored otherwise.

Performance

The performance of this plugin is mainly dependent on:

- The performance/response time of the catalog, which is highly dependent on the size of it, the underlying filesystem and the configuration at the database level
- The number of daemons (Storage Daemons, File Daemons) available in the environment
- The configured services to be executed
- The load of the host at the moment of BGuardian execution

In summary, it is not possible to establish an exact reference about how much time the daemon execution will take to complete. It usually should take some minutes to complete. However, there are some services that will take significant time to complete if the catalog is very large as the implied queries are heavy. These services are 'deviation' and, specially, 'orphanchain'. If there are some services taking too long, it is possible to define different parameters for different executions of the service. For example, we can run a set of services daily or even hourly, while running different services once a week. This is easily set up by using different AdminJobs with command line parameters, testing different services, and then associating them with different schedules.

The general recommendation is to run the daemon once a day over a time window where the load of the backup environment is low. Following this principle, it should be possible to run smoothly this plugin using all the checks without inconvenience.

Limitations

The following article presents limitations of BGuardian Plugin.

While this tool is designed to help to monitor and detect important issues related with security and all kind of best-practices to keep an environment safe, it should not be used standalone as the only measure of protection.

As stated in the introductory article, many other mechanisms should be used to keep an environment safe, where doing things securely should be treated as a general culture put in place in each and every element of an organization, from software elements to human practices.

- BGuardian is only available for Bacula environments deployed over **PostgreSQL databases**.
- Catalog running on MySQL is not supported.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

In this article, there are suggested solutions to common situations that can cause trouble during the usage of the BGuardian plugin.

Log files and debug

The plugin supports a debug parameter and produces internal log files with details of what is happening. Increasing the debug parameter to a higher number will produce a more detailed output inside the plugin debug logs.

The location of the logs are controlled by the log parameter, by default they are placed in the working directory, inside 'bguardian' directory.

The internal plugin logging framework rotates log files automatically. Currently, each file can be 50Mb at maximum and the plugin will keep 25 files.

The ".err" file that can be found close to the log files, can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like 'logrotate'.

Out of Memory

If you ever face *OutOfMemory* errors from the Java daemon (you will find them in the bguardian-debug.err file), you have probably a large environment with a large catalog and/or many different daemons to connect to.

To overcome this situation you can increase JVM memory, you will need to create the following file:

Listing 31: **Memory parameters file**

```
/opt/bacula/etc/bguardian_backend.conf'
```

Then, add the following parameters to the file:

Listing 32: **Memory parameters**

```
BGUARDIAN_JVM_MIN=2G
BGUARDIAN_JVM_MAX=8G
```

Those values will define the MIN (BGUARDIAN_JVM_MIN) and MAX (BGUARDIAN_JVM_MAX) memory values assigned to the JVM Heap size. In this example we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough to handle every situation.

The '/opt/bacula/etc/bguardian_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

3 Virtualization

Important: Virtualization solutions are used with the File Daemon.

3.1 Hypervisors

OpenStack Cinder Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

The following article aims at presenting the reader with information about the Bacula Enterprise Openstack Cinder Plugin. Through subchapters, more in-depth information can be found about the following topics:

Scope

The **Bacula Enterprise Openstack-VM Plugin** currently supports the following platforms:

- 2024.1 Caraval
- 2023.2 Bobcat
- 2023.1 Antelope

Features

The main feature of Bacula Enterprise Openstack Cinder Plugin is to offer Full block level backup and restore of instance volume(s).

Architecture

Bacula Enterprise Openstack-VM Plugin is a Bacula File Daemon plugin built over Openstack Cinder-Backup service.

All information is obtained using a custom implementation of a Cinder-Backup driver feeding data from Openstack to Bacula or the other way around.

Below, there is a simplified vision of the architecture of this plugin within a generic **Bacula Enterprise** deployment:

Cinder Bacula Driver Backup

During volume backup operations, for every file to backup, the bacula driver will:

- Keep track of backup file name
- Snapshot volume
- Create a FIFO (named pipe) from Openstack to Bacula
- Send relevant command to Bacula to synchronize the named pipe
- Return opened FIFO for Cinder to write into.

Once all files are backed up, the process must be stopped by:

- Closing the named pipe

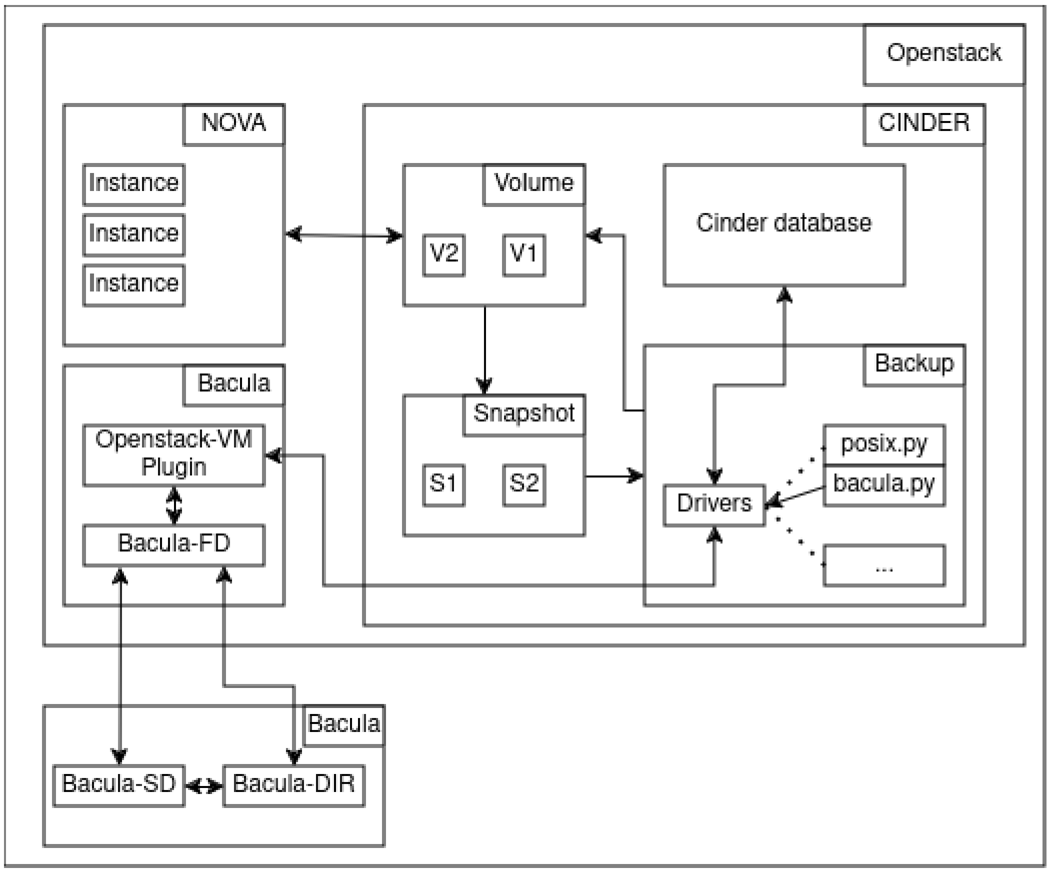


Fig. 8: Openstack-VM Plugin Architecture

- Gathering logs from Bacula
- Checking the list of backup files and job statuses
- Deleting the named pipe.

Cinder Bacula Driver Restore

During volume restore operation, the bacula driver will communicate with Bacula through two different channels.

The first instance will handle the restore procedure by performing the process analog to backup apart for the fact the Cinder will read into the named pipe.

The closing process is also analog to the backup process.

The second instance will provide Cinder with the list of restored files to compare with its own file list by:

- Opening the named pipe to Bacula.
- Gathering the backup job file list.
- Returning curated output to Cinder for control.

Encrypted Volume Support

Volumes encrypted with `LUKS` are supported by the Cinder driver API. However, the encryption keys usually managed by the Openstack Barbican service should be backed up separately following the Openstack backup procedure.

https://docs.openstack.org/operations-guide/ops-backup-recovery.html

Installation

This article describes how to install Bacula Enterprise Openstack Cinder Plugin.

The installation process consists of two parts.

Note: Bacula Enterprise Openstack Cinder Plugin must be installed on Openstack host machine.

First, the installation of the bacula-enterprise-openstack-vm plugin with the BIM tool.

Second, configure the plugin as described in OpenstackVMConfiguration.

Third. by running the install script located at `/opt/bacula/scripts/install-openstack-vm.sh` two times and adjusting the Bacula director configuration.

- First time with the *configure* option `root@user:~# /opt/bacula/scripts/install-openstack-vm.sh configure`
- Second time with the *install* option `root@user:~# /opt/bacula/scripts/install-openstack-vm.sh install`
- At this point a configuration sample located at `/opt/bacula/openstack/bacula-dir.conf.sample` is created. Inside this file, there is a configuration example that should be adjusted and added to the Director configuration, either by editing the Director `/opt/bacula/etc/bacula-dir.conf` configuration file, or using BWeb.

- The install script can be run the third time with the *test* option with `root@user:~# /opt/bacula/scripts/install-openstack-vm.sh test` to check if the installation is correct.

Note: If the Bacula director already has a *Client* resource, the Client in `bacula-dir.conf.sample` should be ignored as the Client resource should not be duplicated.

Note: The OpenStack account name should be the user running the cinder-backup service. Use any of the following commands to check the user running the cinder-backup service:

```
# systemctl status cinder.backup.service
#  ps aux | grep "cinder-backup"
```

Steps

Here is an example how the install script should be used.

1. Run:

```
root@user:~# /opt/bacula/scripts/install-openstack-vm.sh configure

Enter the unix Openstack account name [stack]:

Enter the Bacula Director Name [stackdev-dir]:

Enter the Bacula Director Address [stackdev]:

Enter the Bacula Director Port [9101]:

Enter the Bacula FileDaemon name [stackdev-fd]:

INFO: Creating configuration template for the Director
      /opt/bacula/openstack/bacula-dir.conf.sample will help you to setup
      a Job with the Bacula Enterprise Openstack Plugin.

      The template can be included in your Director configuration and
      you need to review all items marked as "might need to be adjusted"

root@user:~# /opt/bacula/scripts/install-openstack-vm.sh install

Enter the unix Openstack account name: [stack]


Enter path to cinder drivers folder or automatically search system for it

'/opt/stack/cinder/cinder/backup/drivers/bacula.py' -> '/opt/bacula/share/
↪bacula.py'
```

2. Once the director configuration is updated, run:

```
root@stackdev:/opt/bacula# scripts/install-openstack-vm.sh test

Enter the unix Openstack account name: [stack]

1000 OK: 10002 stackdev-dir Version: 18.0.2 (05 March 2024)
INFO: Connection to the Director OK
INFO: Connection from the Director to the Client OK
INFO: Plugin installed correctly
INFO: Job found on the Director
INFO: Fileset configured on the Director
INFO: RestoreJob found on the Director
INFO: Test job finished ok
```

bacula-dir.conf.sample file

This file contains the OpenStack Job, Fileset, Client and Console configuration required on the Director. Usually, there is no need to do any modification to the resources provided.

It will be located in the OpenStack node, in the `/opt/bacula/openstack` directory, after the installation process has finished.

The Job and the Client resources are configured with `MaximumConcurrentJobs = 10`. It means that you can have up to 10 instances of concurrent jobs running. If more concurrent jobs need to be run, you must increase this value to minimum the amount of instances being backed up at the same time.

Result

Openstack Cinder Plugin is installed.

Configuration

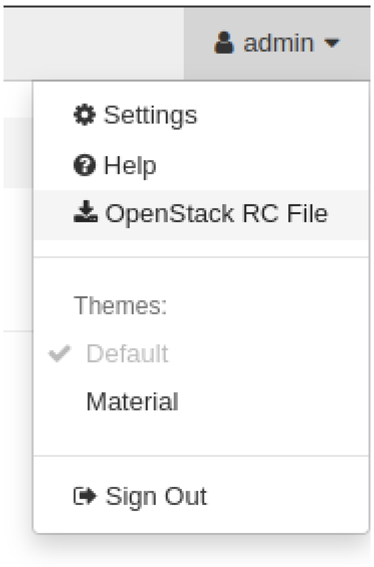
The following article presents the configuration of the plugin.

1. For each Tenant/Project in the OpenStack infrastructure that contains Instances to be backed up, it is required to download the `<tenant/project>-openrc.sh` file in the host where the OpenStack Cinder Plugin is installed. The `<tenant/project>-openrc.sh` file contains variables that enable the plugin to communicate with the cinder-backup service to perform the backup and restore of the instance(s) volume(s).

To backup the Instance(s) volume(s) in the Admin tentant, for example:

a) Download the `admin-openrc.sh` file.

   Downloading `admin_openrc.sh` script can be done through the Openstack dashboard. To do so, the user can click on the `OpenStack RC File` menu item located at the top right of the dashboard.



b) Inside the `admin_openrc.sh` file comment or replace both `echo "Please enter your OpenStack Password for project $OS_PROJECT_NAME as user $OS_USERNAME: "` and `read -sr OS_PASSWORD_INPUT` with `OS_PASSWORD_INPUT=<password>` like in the example below.

```
admin-openrc.sh - Bash
#!/usr/bin/env bash
# To use an OpenStack cloud you need to authenticate against the Identity
# service named keystone, which returns a **Token** and **Service Catalog**.
# The catalog contains the endpoints for all services the user/tenant has
# access to - such as Compute, Image Service, Identity, Object Storage, Block
# Storage, and Networking (code-named nova, glance, keystone, swift,
# cinder, and neutron).
#
# *NOTE*: Using the 3 *Identity API* does not necessarily mean any other
# OpenStack API is version 3. For example, your cloud provider may implement
# Image API v1.1, Block Storage API v2, and Compute API v2.0. OS_AUTH_URL is
# only for the Identity API served through keystone.
export OS_AUTH_URL=http://10.0.255.255/identity
# With the addition of Keystone we have standardized on the term **project**
# as the entity that owns the resources.
export OS_PROJECT_ID=abcdefghijklmnopqrstuvwxyz012345
export OS_PROJECT_NAME="admin"
export OS_USER_DOMAIN_NAME="Default"
if [ -z "$OS_USER_DOMAIN_NAME" ]; then unset OS_USER_DOMAIN_NAME; fi
export OS_PROJECT_DOMAIN_ID="default"
if [ -z "$OS_PROJECT_DOMAIN_ID" ]; then unset OS_PROJECT_DOMAIN_ID; fi
# unset v2.0 items in case set
unset OS_TENANT_ID
unset OS_TENANT_NAME
# In addition to the owning entity (tenant), OpenStack stores the entity
# performing the action as the **user**.
export OS_USERNAME="admin"
# With Keystone you pass the keystone password.
```

(continues on next page)

(continued from previous page)

```
# The two next lines are the one that need to be commented out or deleted
# echo "Please enter your OpenStack Password for project $OS_PROJECT_NAME as␣
↪user $OS_USERNAME: "
# read -sr OS_PASSWORD_INPUT

# Add this line with your Openstack password
OS_PASSWORD_INPUT=<password>

export OS_PASSWORD=$OS_PASSWORD_INPUT
# If your configuration has multiple regions, we set that information here.
# OS_REGION_NAME is optional and only valid in certain environments.
export OS_REGION_NAME="RegionOne"
# Don't leave a blank variable, unset it if it was empty
if [ -z "$OS_REGION_NAME" ]; then unset OS_REGION_NAME; fi
export OS_INTERFACE=public
export OS_IDENTITY_API_VERSION=3
```

c) Copy or symlink the file into `/opt/bacula/etc/admin-openrc.sh` or another directory in the OpenStack host where the Bacula OpenStack Cinder Plugin is installed. By default, the plugin uses the `DEFAULT=/opt/bacula/admin-openrc.sh` path. It is possible to store the `admin-openrc.sh` file in a different directory, and provide the relevant value for `admin-openrc.sh` plugin parameter.

2. The second important configuration step is to advise the cinder-backup service to use the Bacula Cinder driver.

   To do so, the Cinder configuration file located by default at `/etc/cinder/cinder.conf` needs to be modified.

   Inside the `[DEFAULT]` group, the line `backup_driver = cinder.backup.drivers.bacula.BaculaBackupDriver` need to be added.

```
[DEFAULT]
...
backup_driver = cinder.backup.drivers.bacula.BaculaBackupDriver
...
```

Configuring the Backint Parameter File

Backint can be configured with the `/opt/bacula/openstack/os-backint.conf` file.

Note: Usually the configuration file genrated by the install script should fit the current system. The following section is only for user who wish to manually configure `os-backint`.

The keywords presented here are accepted in the backint.conf file.

Parameter	Example	Description	Required	Default
client	`client=opens`	Bacula Client name.	Yes	NULL
restoreclient	`restoreclien`	Bacula Client name used to restore data.	No	NULL
job	`job=OPENSTAC`	Bacula Backup Job name.	Yes	NULL
bconsole	`bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/openstack/bconsole.conf"`	Bconsole command with all arguments.	Yes	`bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/openstack/bconsole.conf"`
RestoreJob	`restorejob=R`	Bacula Restore Job name. If multiple restore jobs are defined in your configuration and this option is not used, `backint` will automatically choose the first restore Job defined.	No	NULL
WaitJob-Completion	`waitjobcompl`	Indicates to wait for Job completion at the end of the `backint` session. The default is wait at the end of the `backint` session.	Yes	no
JobOpt	`jobopt="spoo`	Allows you to specify additional Job options.	No	NULL
CtrlFile	`ctrlfile=/opt/bacula/openstack/os-backint`	Specifies the base path of control files used to connect with the bacula-fd plugin. You must use the same location on the Plugin command line in the Fileset, and in the backint.conf configuration file.	No	`/opt/bacula/openstack/os-backint`
wait_retry	`wait_retry=3`	Specifies the number of times that `backint` will try to reach the Bacula Enterprise Openstack Cinder Plugin (10s between each try).	No	32
catalog	`catalog="MyC 2"`	Specifies a Bacula Catalog name if your director is using multiple catalogs.	No	NULL
trace	`trace=/tmp/log.txt`	Points to an optional trace file.	No	NULL
debug	`debug=50`	Debug level.	No	0

Operations

The following article describes details regarding backup, restore or query operations with Bacula Enterprise Openstack Cinder Plugin.

Bacula Enterprise Openstack Procedures

The Bacula Enterprise Openstack Cinder Plugin has its own set of procedures to interact with the Openstack environment.

The user should only interact with procedures that contains the keyword `execute` in their name with the exception of the `openstack-vm-query`.

- `openstack-vm-execute-backup` to instance's volume(s) backup.
- `openstack-vm-execute-restore` to instance's volume(s) restore.
- `openstack-vm-execute-interactive-delete` to delete backups, snapshots or volumes.
- `openstack-vm-query` to get more information about Openstack resources.

All these procedures have their own dedicated chapter in this Operation section.

Backup in Openstack Cinder

Execute the backup of the volume(s) for a specific instance, or a set of instances, by running the `/opt/bacula/bin/openstack-vm-execute-backup` procedure with relevant parameters.

Parameters

- `-b <instance-name>` - If this parameter is set, the procedure will try to backup all volume of an instance named <instance-name>.
- `-c <tenant/project-openrc.rc>` - Path to the `<tenant/project>-openrc.sh` file. Default value is `/opt/bacula/etc/admin-openrc.sh`.
- `-i` - If this parameter is set, the backup will be incremental.
- `-t <tools>` - Path to Openstack procedures. Default value is `/opt/bacula/bin/`.
- `-p <project-id>` - If set backup will target VM in this project only.
- `-v <instance-id>` - ID of the instance to backup.
- `-w <waiting-time>` - Waiting time between two poll operations. Default value is `5`.
- `--all-projects` - Backup all volume in all projects
- `-h` - Display help.

Note: Option `-v` has precedence over option `-b`.

Example

Backup of all the volumes attached to a specific instance, using the instance ID:

Note: To get either the ID or the name of a specific instance, the query procedure can be used with `/opt/bacula/bin/openstack-vm-query -l`. The ID can be found under the ID column.

First get the relevant instance id:

```
root@stackdev:/opt/bacula# bin/openstack-vm-query -l
+--------------------------------------+----------------+--------+------------
↪-----------------------------------------------------------------+-------
↪-------------------+----------+
| ID                                   | Name           | Status | Networks  ␣
↪                                                                 | Image␣
↪                   | Flavor   |
+--------------------------------------+----------------+--------+------------
↪-----------------------------------------------------------------+-------
↪-------------------+----------+
| instance_ID                          | instance_name  | ACTIVE | private=00.
↪0.0.0, 1111:1111:1111:0:1111:1111:1111:1111; shared=111.111.111.111 | N/A␣
↪(booted from volume) | m1.micro |
+--------------------------------------+-------+--------+--------------------
↪----------------------------------------------------------+----------------
↪----------+----------+
```

Then issue backup creation operation:

```
root@openstack-bck:~# /opt/bacula/bin/openstack-vm-execute-backup -v instance_
↪ID
Backup of VM=<instance_ID>  INCREMENTAL=False       ADMIN_OPENRC=/opt/bacula/
↪etc/admin-openrc.sh    SCRIPTS=/opt/bacula/bin/
I: Found 2 volumes to backup for <instance_ID>
I: Backing up <volume_ID>
D: Issue snapshot
I: Snapshot done
I: Backup status=creating
[...]
I: Backup status=creating
I: Backup status=creating
I: Backup status=available
I: Done proceeds to next
I: Issue delete command for snapshot=<snapshot_ID>
I: Backing up <volume_ID>
D: Issue snapshot
D: Snapshot creating ...
I: Snapshot done
I: Backup status=creating
[...]
I: Backup status=creating
I: Backup status=creating
I: Backup status=available
```

(continues on next page)

(continued from previous page)

```
I: Done proceeds to next
I: Issue delete command for snapshot=<snapshot_ID>
I: No more volumes to process END OF BACKUP
```

Backup of all the volumes attached to a specific instance, using the instance name:

```
root@host:/opt/bacula# /opt/bacula/bin/openstack-vm-execute-backup -b␣
↪instance_name
Backup of VM=<instance_ID>  INCREMENTAL=False       ADMIN_OPENRC=/opt/bacula/
↪etc/admin-openrc.sh    SCRIPTS=/opt/bacula/bin/
I: Found 2 volumes to backup for <instance_ID>
I: Backing up <volume_ID>
D: Issue snapshot
I: Snapshot done
I: Backup status=creating
[...]
I: Backup status=creating
I: Backup status=creating
I: Backup status=available
I: Done proceeds to next
I: Issue delete command for snapshot=<snapshot_ID>
I: Backing up <volume_ID>
D: Issue snapshot
D: Snapshot creating ...
I: Snapshot done
I: Backup status=creating
[...]
I: Backup status=creating
I: Backup status=creating
I: Backup status=available
I: Done proceeds to next
I: Issue delete command for snapshot=<snapshot_ID>
I: No more volumes to process END OF BACKUP
```

When using the -b or -v options, to backup an OpenStack instance, there will be one backup jobid in Bacula for each volume attached to the instance. Also, there will be one backup in the OpenStack server for each volume.

```
root@host:/opt/bacula# /opt/bacula/bin/openstack-vm-query -b
+---------------+---------------------+--------------------------------------
↪----+-----------+------+-------------+
| ID            | Name                | Description                         ␣
↪    | Status    | Size | Incremental |
+---------------+---------------------+--------------------------------------
↪----+-----------+------+-------------+
| <backup1_ID>> | <backup1_name>      | Backup done by Bacula Enterprise    ␣
↪    | available |   10 | False       |
|                                     | INSTANCE=<instance_name> DATE=
↪<datetime> |           |      |             |
|                                     |                                     ␣
↪    |           |      |             |
| <backup2_ID>> | <backup2_name>      | Backup done by Bacula Enterprise    ␣
```

(continues on next page)

(continued from previous page)

```
↪    | available |    5 | False       |
|                                    | INSTANCE=<instance_name> DATE=
↪<datetime> |           |      |             |
+---------------+---------------------+-------------------------------------
↪----+-----------+------+-------------+
```

Also, in the Catalog, two jobids are created:

```
|    xx | job.openstack-bck-fd.openstack-vm | 2024-01-01 12:00:00 | B    | I ␣
↪   |        6 | 12,345,678 | T          |
|    XX | job.openstack-bck-fd.openstack-vm | 2024-01-01 12:00:05 | B    | I ␣
↪   |        9 | 90,123,456 | T          |
```

Backup Job Example with a RunScript Block

As mentioned earlier, it is possible to define a backup job to trigger the openstack-vm-execute-backup program to execute the backup in the OpenStack server.

The RunScript block below triggers the openstack-vm-execute-backup program to backup all the volumes attached to the *MyInstance* instance in the OpenStack server, having the *openstack-bck-fd* bacula client installed:

```
Job {
  Name = OpenStack-test-job
  JobDefs = BackupsToDisk
  Fileset = None
  Client = openstack-bck-fd
  RunScript {
    Command = "/opt/bacula/bin/openstack-vm-execute-backup -b MyInstance"
    RunsOnClient = yes
    RunsWhen = Before
  }
}

Fileset {
  Name = None
  EnableVSS = no
}
```

The RunScript block below triggers the openstack-vm-execute-backup program to backup all the volumes attached to any instance whose name begins with *MyInstance* in the OpenStack server, having the *openstack-bck-fd* bacula client installed:

```
Job {
  Name = OpenStack-test-job
  JobDefs = BackupsToDisk
  Fileset = None
  Client = openstack-bck-fd
  RunScript {
    Command = "/opt/bacula/bin/openstack-vm-execute-backup -b MyInstance*"
    RunsOnClient = yes
```

(continues on next page)

(continued from previous page)

```
    RunsWhen = Before
  }
}

Fileset {
  Name = None
  EnableVSS = no
}
```

By default, the plugin will use the `/opt/bacula/etc/admin-openrc.sh` tenant OpenStack RC file. To trigger the backup of instance(s) volume(s) in a different tenant/project, it is required to use the `-c <tenant/project-openrc.rc>` option. For example, to backup all the instances whose names begin with `MyInstance` in the `demo` tenant/project:

```
Job {
  Name = OpenStack-test-job
  JobDefs = BackupsToDisk
  Fileset = None
  Client = openstack-bck-fd
  RunScript {
    Command = "/opt/bacula/bin/openstack-vm-execute-backup -c /opt/bacula/etc/
↪demo-openrc.sh -b MyInstance*"
    RunsOnClient = yes
    RunsWhen = Before
  }
}

Fileset {
  Name = None
  EnableVSS = no
}
```

Restore in Openstack Cinder

Restore volumes attached to a previous backup or a specific volume by running the `/opt/bacula/bin/openstack-vm-execute-restore` procedure with relevant parameters.

The `/opt/bacula/bin/openstack-vm-execute-restore` program available in the OpenStack server must be used for restores. This program can be added to a RunScript block of an Admin job, and this Admin job can be triggered from bconsole or BWeb.

The `/opt/bacula/bin/openstack-vm-execute-restore` program can be used with a few parameters.

Parameters

- `-b <backup_id>` - ID of a specific volume backup to restore.
- `-c <admin-openrc>` - Path to modified admin-openrc.sh Default value is `/opt/bacula/etc/admin-openrc.sh`.
- `-n <backup_name>` - If this parameter is set, the volume with this name will be restored.
- `-t <tools>` - Path to Openstack procedure. Default value is `/opt/bacula/bin/`.
- `-v <instance_id>` - ID of the instance to restore.
- `-p <project-id>` - If set restore will target VM in this project only.
- `-P <project-id>` - If specified the restore operation will restore all volumes related to this project ID.
- `-w <waitingTime>` - Waiting time in seconds between two completion check. Default value is `5`.
- `-h` - Display help.

Examples

Restore Volumes from an Instance Using `openstack-vm-execute-restore`

This example explain how to restore all the volumes, or a specific volume, from an existent instance, using the `/opt/bacula/bin/openstack-vm-execute-restore` procedure in the OpenStack node.

Note: To get the ID of a specific instance, the query procedure can be used with `/opt/bacula/bin/openstack-vm-query -l`. The ID can be found under the ID column.

In case the virtual machine was deleted beforehand Cinder backups created by the plugin will have the original virtual machine ID as a name. To access the list of backup the query procedure can be used with `/opt/bacula/bin/openstack-vm-query -b`.

Restore using the instance ID. Get the instance_ID using the `/opt/bacula/bin/openstack-vm-query -l` command:

```
root@stackdev:/opt/bacula# /opt/bacula/bin/openstack-vm-query -l -c /opt/
↪bacula/etc/demo-openrc.sh
+--------------------------------------+--------------+--------+------------
↪----------+--------------------------+---------+
| ID                                   | Name         | Status | Networks   ␣
↪          | Image                    | Flavor  |
+--------------------------------------+--------------+--------+------------
↪----------+--------------------------+---------+
| e20b1863-6a3f-4b09-90b6-66f77441f3ef | demoInstance1 | ACTIVE | shared=192.
↪168.233.32 | N/A (booted from volume) | m1.tiny |
+--------------------------------------+--------------+--------+------------
↪----------+--------------------------+---------+
```

Or list the backups available, and get the instance ID (e20b1863-6a3f-4b09-90b6-66f77441f3ef):

```
root@stackdev:/opt/bacula# /opt/bacula/bin/openstack-vm-query -b -c /opt/
↪bacula/etc/demo-openrc.sh
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
| ID                                   | Name                                 ␣
↪           | Description                                         | Status   ␣
↪ | Size | Incremental |
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
| 4cbd73d0-8c0e-4808-9900-240c280f1f12 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise                    |␣
↪available |    1 | False       |
|                                      |                                      ␣
↪           | INSTANCE=demoInstance1 DATE=Fri Oct 11 08:02:31 AM  |          ␣
↪ |      |             |
|                                      |                                      ␣
↪           | UTC 2024                                            |          ␣
↪ |      |             |
| a8967472-ed67-4cb5-8e45-c8781dffe970 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise                    |␣
↪available |    2 | False       |
|                                      |                                      ␣
↪           | INSTANCE=demoInstance1 DATE=Fri Oct 11 08:02:31 AM  |          ␣
↪ |      |             |
|                                      |                                      ␣
↪           | UTC 2024                                            |          ␣
↪ |      |             |
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
```

Then issue the restore command using the instance_ID value:

```
root@stackdev:~# /opt/bacula/bin/openstack-vm-execute-restore -v e20b1863-
↪6a3f-4b09-90b6-66f77441f3ef
Restore of INSTANCE_ID=e20b1863-6a3f-4b09-90b6-66f77441f3ef         SCRIPT_
↪PATH=/opt/bacula/bin/    ADMIN_OPENRC=/opt/bacula/etc/admin-openrc.sh
I: 1 backup to restore
I: Restoring 4cbd73d0-8c0e-4808-9900-240c280f1f12
I: Volume restoration in progress=restoring-backup
I: Restored volume found with ID=<restored_volume_ID>
I: Volume restoration in progress=restoring-backup

...

I: Volume restoration in progress=restoring-backup
I: Volume restoration in progress=available
I: Done moving on to next
I: No more backup to restore END
```

Once this procedure is done, the volumes will be in an *available* status, and you will need to either create a new instance and attach the restored volumes, or to attach the restored volumes to an existent instance using the openstack CLI or with the GUI from the dashboard.

It is also possible to restore a single volume from a backup id by using the -b option.

Get the backup_ID for the specific volume using the /opt/bacula/bin/openstack-vm-query -b command:

```
root@stackdev:/opt/bacula# /opt/bacula/bin/openstack-vm-query -b -c /opt/
↪bacula/etc/demo-openrc.sh
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
| ID                                   | Name                                ␣
↪           | Description                                         | Status  ␣
↪ | Size | Incremental |
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
| 4cbd73d0-8c0e-4808-9900-240c280f1f12 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise                    |␣
↪available |    1 | False       |
|                                      |                                     ␣
↪           | INSTANCE=demoInstance1 DATE=Fri Oct 11 08:02:31 AM  |         ␣
↪ |      |             |
|                                      |                                     ␣
↪           | UTC 2024                                            |         ␣
↪ |      |             |
| a8967472-ed67-4cb5-8e45-c8781dffe970 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise                    |␣
↪available |    2 | False       |
|                                      |                                     ␣
↪           | INSTANCE=demoInstance1 DATE=Fri Oct 11 08:02:31 AM  |         ␣
↪ |      |             |
|                                      |                                     ␣
↪           | UTC 2024                                            |         ␣
↪ |      |             |
+--------------------------------------+--------------------------------------
↪-----------+-----------------------------------------------------+----------
↪-+------+-------------+
```

Then issue the restore command using the backup ID value, for example:

```
root@stackdev:~# /opt/bacula/bin/openstack-vm-execute-restore -b 4cbd73d0-
↪8c0e-4808-9900-240c280f1f12
Restore of RESTORE_ID=4cbd73d0-8c0e-4808-9900-240c280f1f12  SCRIPT_PATH=/opt/
↪bacula/bin/    ADMIN_OPENRC=/opt/bacula/etc/demo-openrc.sh
I: 1 backup to restore
I: Restoring backup with ID=4cbd73d0-8c0e-4808-9900-240c280f1f12
I: Found restoration volume in progress ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
```

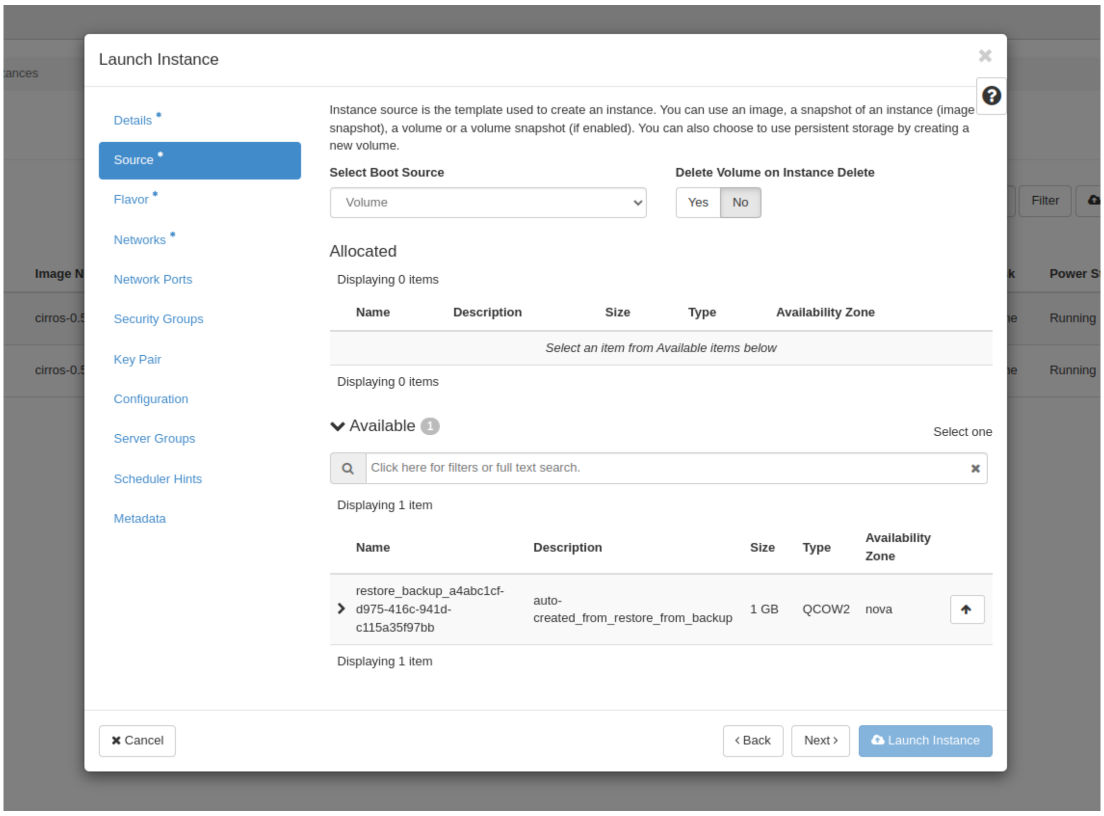
(continues on next page)

(continued from previous page)

```
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=restoring-backup
I: Volume restoration in progress with ID=efc209d6-11c4-4bf4-8917-
↪c4e6f07d50b1      STATUS=available
I: Done moving on to next
I: No more backup to restore FINISHED
```

Create instance from GUI with restored disk

- Under the Instances overview select the `Launch Instance` menu.
- In the source menu select `Volume` from `Select Boot Source`
- If the disk containing the operating system has to be restored. Under the `Available` section select the newly restored disk by hitting the up arrow sign on the right



Launch Instance

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Volume

Delete Volume on Instance Delete: Yes [✓ No]

Allocated

Displaying 0 items

Name	Description	Size	Type	Availability Zone
Select an item from Available items below				

Displaying 0 items

Available 1 — Select one

Click here for filters or full text search.

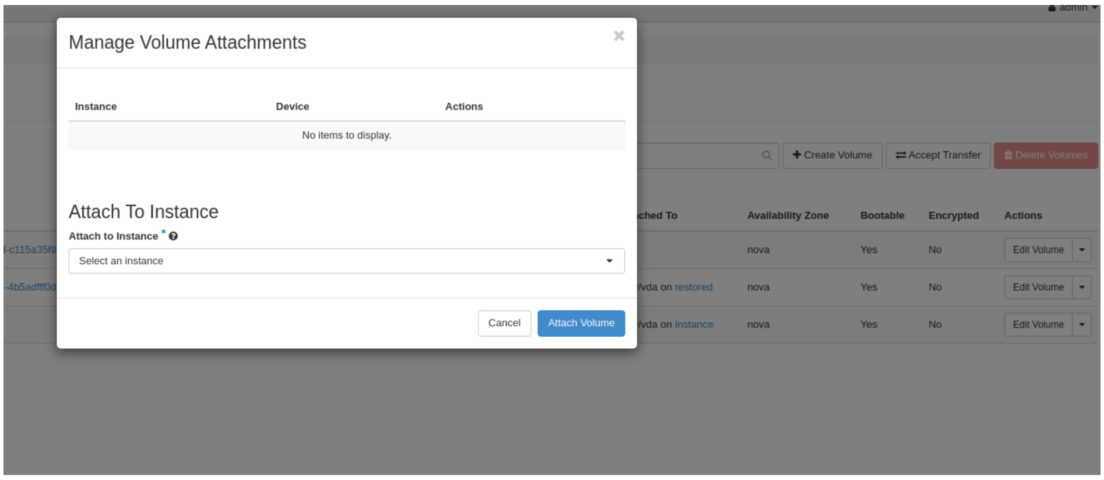
Displaying 1 item

Name	Description	Size	Type	Availability Zone
restore_backup_a4abc1cf-d975-416c-941d-c115a35f97bb	auto-created_from_restore_from_backup	1 GB	QCOW2	nova

Displaying 1 item

- Setup all other parameters, preferably with the same flavor as the backup instance.
- Launch instance
- Manually attach restored data disks to newly created instance via `Volumes` menu
- On the restored disk the `Edit volume` menu contains a `Manage Volume Attachments` section

- In the `Attach to instance` sub-menu select the relevant instance to attach the disk to


Manage Volume Attachments

Instance	Device	Actions
No items to display.		

Attach To Instance

Attach to Instance *

Select an instance

Use an Admin job to be triggered using bconsole or BWeb

The Admin job configuration to be defined in the Director:

```
# cat Job/cinder-01-restore-control-job.cfg
Job {
  Name = "cinder-01-restore-control-job"
  Type = "Admin"
  Client = "cinder-01-fd"
  Fileset = "Fake-fileset"
  Messages = "Default"
  Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
   Command = "ssh root@am-u22-openstack-bck \"/opt/bacula/scripts/restore_
↪instance.sh -n e20b1863-6a3f-4b09-90b6-66f77441f3ef_1728633751 -t demo -c -
↪f 1\""
   RunsOnClient = no
   RunsWhen = Before
  }
  Schedule = "Manual"
  Storage = "DiskAutochanger"
  WriteBootstrap = "/opt/bacula/bsr/%c_%n.bsr"
}
```

The `restore_instance.sh` parameters values:

`-t demo` is the name of the Tenant.

`-f 1` is the flavor of the new instance that will be created. These values you can get from:

```
# openstack flavor list
+----+-----------+-------+------+-----------+-------+-----------+
| ID | Name      |   RAM | Disk | Ephemeral | VCPUs | Is Public |
+----+-----------+-------+------+-----------+-------+-----------+
| 1  | m1.tiny   |   512 |    1 |         0 |     1 | True      |
```

(continues on next page)

(continued from previous page)

```
| 2  | m1.small  |  2048 |   20 |         0 |     1 | True      |
| 3  | m1.medium |  4096 |   40 |         0 |     2 | True      |
| 4  | m1.large  |  8192 |   80 |         0 |     4 | True      |
| 42 | m1.nano   |   192 |    1 |         0 |     1 | True      |
| 5  | m1.xlarge | 16384 |  160 |         0 |     8 | True      |
| 84 | m1.micro  |   256 |    1 |         0 |     1 | True      |
| c1 | cirros256 |   256 |    1 |         0 |     1 | True      |
| d1 | ds512M    |   512 |    5 |         0 |     1 | True      |
| d2 | ds1G      |  1024 |   10 |         0 |     1 | True      |
| d3 | ds2G      |  2048 |   10 |         0 |     2 | True      |
| d4 | ds4G      |  4096 |   20 |         0 |     4 | True      |
+----+-----------+-------+------+-----------+-------+-----------+
```

`-n e20b1863-6a3f-4b09-90b6-66f77441f3ef_1728633751` is the backup name for the instance demoInstance1 with instance ID `e20b1863-6a3f-4b09-90b6-66f77441f3ef`. This value can be collected by using the following cinder-01-list-backups-job admin job, or command line in the cinder node:

```
# cat conf.d/Director/am-u24-openstack-dir-tst-dir/Job/cinder-01-list-backups-
↪job.cfg
Job {
  Name = "cinder-01-list-backups-job"
  Type = "Admin"
  Client = "cinder-01-fd"
  Fileset = "Fake-fileset"
  Messages = "Default"
 Pool = "DiskBackup365d"
  Priority = 10
  Runscript {
   Command = "ssh root@am-u22-openstack-bck \"/opt/bacula/scripts/list_
↪backups.sh demo demoInstance1\""
   RunsOnClient = no
   RunsWhen = Before
  }
  Schedule = "Manual"
  Storage = "DiskAutochanger"
  WriteBootstrap = "/opt/bacula/bsr/%c_%n.bsr"
}
```

The `cinder-01-list-backups-job` job log to list all the backups for the `demoInstance1` instance:

```
2024-10-14 11:50:21 am-u24-openstack-dir-tst-dir JobId 451: shell command:␣
↪run BeforeJob "ssh root@am-u22-openstack-bck "/opt/bacula/scripts/list_
↪backups.sh demo demoInstance1""
2024-10-14 11:50:22 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob:␣
↪Listing backups for Instance demoInstance1...
2024-10-14 11:50:22 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: Use␣
↪the backup `Name` to restore all the instance volumes, or use the backup␣
↪`ID` to restore a single instance volume.
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: +------
↪--------------------------------+------------------------------------------
↪------+--------------------------------------------------------------------
↪-------------------------+-----------+------+-------------+
```

(continues on next page)

(continued from previous page)

```
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: | ID ␣
↪                                  | Name                                      ␣
↪      | Description                                                         ␣
↪                         | Status     | Size | Incremental |
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: +------
↪--------------------------------+------------------------------------------
↪------+-------------------------------------------------------------------
↪-------------------------+-----------+------+-------------+
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: |␣
↪4cbd73d0-8c0e-4808-9900-240c280f1f12 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise INSTANCE=demoInstance1␣
↪DATE=Fri Oct 11 08:02:31 AM UTC 2024 | available |    1 | False       |
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: |␣
↪a8967472-ed67-4cb5-8e45-c8781dffe970 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise INSTANCE=demoInstance1␣
↪DATE=Fri Oct 11 08:02:31 AM UTC 2024 | available |    2 | False       |
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: BeforeJob: +------
↪--------------------------------+------------------------------------------
↪------+-------------------------------------------------------------------
↪-------------------------+-----------+------+-------------+
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: Start Admin JobId␣
↪451, Job=cinder-01-list-backups-job.2024-10-14_11.50.18_29
2024-10-14 11:50:25 am-u24-openstack-dir-tst-dir JobId 451: Bacula 18.0.4␣
↪(06Sep24): 14-Oct-2024 11:50:25
  JobId:                  451
  Job:                    cinder-01-list-backups-job.2024-10-14_11.50.18_29
  Scheduled time:         14-Oct-2024 11:50:18
  Start time:             14-Oct-2024 11:50:25
  End time:               14-Oct-2024 11:50:25
  Termination:            Admin OK
```

And using the `openstack-vm-query` procedure in the Cinder node:

```
# /opt/bacula/bin/openstack-vm-query -c /opt/bacula/etc/demo-openrc.sh -b |␣
↪grep "^+\|^| ID\|demoInstance1"
+--------------------------------------+------------------------------------
↪-----------+-------------------------------------------------------------
↪------------------------------+-----------+------+-------------+
| ID                                   | Name                                  ␣
↪          | Description                                                      ␣
↪                              | Status     | Size | Incremental |
+--------------------------------------+------------------------------------
↪-----------+-------------------------------------------------------------
↪------------------------------+-----------+------+-------------+
| 4cbd73d0-8c0e-4808-9900-240c280f1f12 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise INSTANCE=demoInstance1␣
↪DATE=Fri Oct 11 08:02:31 AM UTC 2024 | available |    1 | False       |
| a8967472-ed67-4cb5-8e45-c8781dffe970 | e20b1863-6a3f-4b09-90b6-66f77441f3ef_
↪1728633751 | Backup done by Bacula Enterprise INSTANCE=demoInstance1␣
↪DATE=Fri Oct 11 08:02:31 AM UTC 2024 | available |    2 | False       |
+--------------------------------------+------------------------------------
↪-----------+-------------------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
↪-----------------------------+-----------+------+-------------+
```

The `cinder-01-restore-control-job` job log will report the new instance created:

```
2024-10-11 13:44:46 openstack-dir- JobId 285: shell command: run BeforeJob
↪"ssh root@am-u22-openstack-bck "/opt/bacula/scripts/restore_instance.sh -v␣
↪demoInstance1 -n e20b1863-6a3f-4b09-90b6-66f77441f3ef_1728633751 -t demo -c␣
↪-f c1""
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob:
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: +---------------------
↪----------------+------------------------------------------------------------
↪-----------------------------------------------------------------------------
↪-----------------------------------------------------------------------------
↪-------------+
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | Field               ␣
↪                 | Value                                                     ␣
↪                                                                             ␣
↪                                                                             ␣
↪              |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: +---------------------
↪----------------+------------------------------------------------------------
↪-----------------------------------------------------------------------------
↪-----------------------------------------------------------------------------
↪-------------+
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-DCF:diskConfig   ␣
↪                 | MANUAL                                                    ␣
↪                                                                             ␣
↪                                                                             ␣
↪              |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-
↪AZ:availability_zone         | nova                                            ␣
↪                                                                             ␣
↪                                                                             ␣
↪                                                                             ␣
↪                           |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:host                 | am-u22-openstack-bck                               ␣
↪                                                                             ␣
↪                                                                             ␣
↪                       |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:hostname             | demoinstance1-restored                             ␣
↪                                                                             ␣
↪                                                                             ␣
↪                       |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:hypervisor_hostname | am-u22-openstack-bck                                ␣
↪                                                                             ␣
↪                                                                             ␣
↪                       |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:instance_name        | instance-00000014                                  ␣
↪                                                                             ␣
```

(continues on next page)

(continued from previous page)

```
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:kernel_id           | None                                              ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:launch_index        | None                                              ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:ramdisk_id          | None                                              ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:reservation_id      | r-5drlfobi                                        ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:root_device_name    | /dev/vda                                          ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-SRV-
↪ATTR:user_data           | None                                              ␣
↪                                                                              ␣
↪                                                                              ␣
↪                      |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-STS:power_
↪state              | Running                                                 ␣
↪                                                                              ␣
↪                                                                              ␣
↪                |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-STS:task_
↪state               | None                                                   ␣
↪                                                                              ␣
↪                                                                              ␣
↪                 |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-EXT-STS:vm_state␣
↪                | active                                                     ␣
↪                                                                              ␣
↪                                                                              ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-SRV-USG:launched_
↪at              | 2024-10-11T13:47:56.000000                                 ␣
↪                                                                              ␣
↪                                                                              ␣
↪             |
```

(continues on next page)

(continued from previous page)

```
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | OS-SRV-
↪USG:terminated_at            | None                                        ␣
↪                                                                           ␣
↪                                                                           ␣
↪                           |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | accessIPv4        ␣
↪                 | None                                                    ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | accessIPv6        ␣
↪                 | None                                                    ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | addresses         ␣
↪                 | N/A                                                     ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | adminPass         ␣
↪                 | JiVRLeB59PCG                                            ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | config_drive      ␣
↪                 | None                                                    ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | created           ␣
↪                 | 2024-10-11T13:47:23Z                                    ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | description       ␣
↪                 | None                                                    ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | flavor            ␣
↪                 | description=, disk='1', ephemeral='0', extra_specs.hw_
↪rng:allowed='True', id='cirros256', is_disabled=, is_public='True',␣
↪location=, name='cirros256', original_name='cirros256', ram='256', rxtx_
↪factor=, swap='0', vcpus='1' |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | hostId            ␣
↪                 | 321b77de1457d2ad95d696392b037a456230edc2009395d7dd924b6c ␣
↪                                                                           ␣
↪                                                                           ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | host_status       ␣
↪                 | UP                                                      ␣
```

(continues on next page)

(continued from previous page)

```
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | id                   ␣
↪                 | 5854acee-28d4-4029-944a-ccef9fc88b5d                       ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | image                ␣
↪                 | N/A (booted from volume)                                   ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | key_name             ␣
↪                 | None                                                       ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | locked               ␣
↪                 | None                                                       ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | locked_reason        ␣
↪                 | None                                                       ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | name                 ␣
↪                 | demoInstance1_restored                                     ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | pinned_availability_
↪zone            | None                                                        ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | progress             ␣
↪                 | None                                                       ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | project_id           ␣
↪                 | 9c8aa25a903346e6af3877a6b4612a46                           ␣
↪                                                                              ␣
↪                                                                              ␣
↪            |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | properties           ␣
↪                 | None                                                       ␣
↪                                                                              ␣
↪                                                                              ␣
```

(continues on next page)

(continued from previous page)

```
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | security_groups    ␣
↪                | name='default'                                            ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | server_groups      ␣
↪                | None                                                      ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | status             ␣
↪                | ACTIVE                                                    ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | tags               ␣
↪                |                                                           ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | trusted_image_
↪certificates          | None                                                ␣
↪                                                                            ␣
↪                                                                            ␣
↪                   |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | updated            ␣
↪                | 2024-10-11T13:47:56Z                                      ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | user_id            ␣
↪                | 362d7bb6fcc64be3bcdc1c6fc99229d0                          ␣
↪                                                                            ␣
↪                                                                            ␣
↪             |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: | volumes_attached   ␣
↪                | delete_on_termination='False', id='efc209d6-11c4-4bf4-
↪8917-c4e6f07d50b1'                                                          ␣
↪                                                                            ␣
↪                 |
2024-10-11 13:48:00 openstack-dir JobId 285: BeforeJob: +---------------------
↪----------------+-----------------------------------------------------------
↪----------------------------------------------------------------------------
↪----------------------------------------------------------------------------
↪-------------+
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: +---------------------
↪--+------------------------------------+
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Field              ␣
↪  | Value                              |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: +---------------------
↪--+------------------------------------+
```

(continues on next page)

(continued from previous page)

```
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | ID                    ␣
↪   | 83989c2c-7bb0-4c8e-91c7-511a6e7836fb |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Server ID             ␣
↪   | 5854acee-28d4-4029-944a-ccef9fc88b5d |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Volume ID             ␣
↪   | 83989c2c-7bb0-4c8e-91c7-511a6e7836fb |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Device                ␣
↪   | /dev/vdb                             |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Tag                   ␣
↪   | None                                 |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: | Delete On␣
↪Termination | False                                 |
2024-10-11 13:48:08 openstack-dir JobId 285: BeforeJob: +---------------------
↪--+--------------------------------------+
2024-10-11 13:48:08 openstack-dir JobId 285: Start Admin JobId 285,␣
↪Job=cinder-01-restore-control-job.2024-10-11_13.44.44_09
2024-10-11 13:48:08 openstack-dir JobId 285: Bacula 18.0.4 (06Sep24): 11-Oct-
↪2024 13:48:08
  JobId:                  285
  Job:                    cinder-01-restore-control-job.2024-10-11_13.44.44_09
  Scheduled time:         11-Oct-2024 13:44:44
  Start time:             11-Oct-2024 13:48:08
  End time:               11-Oct-2024 13:48:08
  Termination:            Admin OK
```

Note: The scripts `trigger-cinder-backup.sh`, `list_backups`, and `restore_instances.sh` mentioned may not be readily available in the OpenStack Cinder Plugin. However, users have the flexibility to create and customize these scripts according to their specific needs and preferences.

Query

Display different information about backup, snapshot, instance and/or volumes by running the `/opt/bacula/bin/openstack-vm-query` with relevant parameters.

Parameters

- -b Lists backups
- `-c <admin_openrc>` Path to admin-openrc.sh
- `-f <format>` Format the output in one of the following format: json, table, value, yaml
- `-l` Lists instances
- -L Lists projects
- -p Check if `Cinder-backup` is running
- `-P <project-id>` To target query to a specific project
- -q Check if `Cinder` module is installed

- `-s` List snapshots
- `-v` Lists volumes
- `-V` Verbose output for `-p` and `-q` options
- `-h` Display help

Example

The query procedure is used to list different resources in a defined format.

`/opt/bacula/bin/openstack-vm-query` is the base command.

To list the instances available in the OpenStack server:

```
root@stackdev:/opt/bacula# bin/openstack-vm-query -l
+--------------------------------------+----------------+--------+------------
↪-----------------------------------------------------------------+-------
↪-------------------+-----------+
| ID                                   | Name           | Status | Networks ␣
↪                                                                 | Image␣
↪                   | Flavor   |
+--------------------------------------+----------------+--------+------------
↪-----------------------------------------------------------------+-------
↪-------------------+-----------+
| instance_ID                          | instance_name  | ACTIVE | private=00.
↪0.0.0, 1111:1111:1111:0:1111:1111:1111:1111; shared=111.111.111.111 | N/A␣
↪(booted from volume) | m1.micro |
+--------------------------------------+-------+--------+--------+------------
↪-----------------------------------------------------------------+-------
↪-------------------+-----------+
```

To list the backups performed:

```
root@host:/opt/bacula# /opt/bacula/bin/openstack-vm-query -b
+---------------+---------------------+--------------------------------------
↪----+-----------+------+-------------+
| ID            | Name                | Description                         ␣
↪    | Status    | Size | Incremental |
+---------------+---------------------+--------------------------------------
↪----+-----------+------+-------------+
| <backup1_ID>> | <backup1_name>      | Backup done by Bacula Enterprise    ␣
↪    | available |   10 | False       |
|                                     | INSTANCE=<instance_name> DATE=
↪<datetime> |           |      |             |
|                                     |                                     ␣
↪    |           |      |             |
| <backup2_ID>> | <backup2_name>      | Backup done by Bacula Enterprise    ␣
↪    | available |    5 | False       |
|                                     | INSTANCE=<instance_name> DATE=
↪<datetime> |           |      |             |
+---------------+---------------------+--------------------------------------
↪----+-----------+------+-------------+
```

Listing volumes and backups in a json format would result in:

```
root@openstack-bck:~# /opt/bacula/bin/openstack-vm-query -b -v -f json
[
  {
    "ID": "021b0a50-2729-4c2a-b8ae-252d76aecf42",
    "Name": "5b5bc409-ec90-4a12-b659-d2b4d04eb419_1711446820",
    "Description": "Backup done by Bacula Enterprise INSTANCE=testAna2␣
↪DATE=Tue Mar 26 09:53:40 UTC 2024",
    "Status": "available",
    "Size": 10,
    "Incremental": false
  },
  {
    "ID": "784b88d7-8262-4efe-9512-9ec483b8cb73",
    "Name": "5b5bc409-ec90-4a12-b659-d2b4d04eb419_1711446820",
    "Description": "Backup done by Bacula Enterprise INSTANCE=testAna2␣
↪DATE=Tue Mar 26 09:53:40 UTC 2024",
    "Status": "available",
    "Size": 5,
    "Incremental": false
  }
]
[
  {
    "ID": "6db88c97-1ddb-4f46-bcf4-06699f3f59f5",
    "Name": "testAna2-vol2",
    "Status": "available",
    "Size": 5,
    "Attached to": []
  },
  {
    "ID": "13a9d708-01c7-4320-9f04-534c8c380a64",
    "Name": "restore_backup_021b0a50-2729-4c2a-b8ae-252d76aecf42_at_1711448194
↪",
    "Status": "available",
    "Size": 10,
    "Attached to": []
  },
  {
    "ID": "c645558e-cdf3-4334-968b-3482bf9a6c18",
    "Name": "testAna2-vol2",
    "Status": "available",
    "Size": 5,
    "Attached to": []
  },
  {
    "ID": "39161ee6-ec34-4f1a-87d0-d08ce71fac80",
    "Name": "restore_backup_f90ad4ec-882f-4071-9ec0-367dd7cc73b5_at_1710855568
↪",
    "Status": "available",
    "Size": 10,
    "Attached to": []
  },
```

(continues on next page)

(continued from previous page)

```
{
  "ID": "5c257c96-c0be-4eb6-a751-462b56729e50",
  "Name": "testAna2-vol2",
  "Status": "in-use",
  "Size": 5,
  "Attached to": [
    {
      "id": "5c257c96-c0be-4eb6-a751-462b56729e50",
      "attachment_id": "00765039-5dc2-4bb3-bb95-fa1611b8ba81",
      "volume_id": "5c257c96-c0be-4eb6-a751-462b56729e50",
      "server_id": "5b5bc409-ec90-4a12-b659-d2b4d04eb419",
      "host_name": "openstack-bck",
      "device": "/dev/vdb",
      "attached_at": "2024-03-14T13:17:49.000000"
    }
  ]
},
{
  "ID": "b0412c13-2124-44ac-a11d-058c6146c104",
  "Name": "RestoreTestVolume",
  "Status": "in-use",
  "Size": 1,
  "Attached to": [
    {
      "id": "b0412c13-2124-44ac-a11d-058c6146c104",
      "attachment_id": "ac6c3444-180b-4535-abfc-467057dcad5d",
      "volume_id": "b0412c13-2124-44ac-a11d-058c6146c104",
      "server_id": "1952a2d3-6b0b-417d-a90d-cde7964074d1",
      "host_name": "openstack-bck",
      "device": "/dev/vda",
      "attached_at": "2024-03-08T13:52:40.000000"
    }
  ]
},
{
  "ID": "218b321c-7b7f-4f5b-af4b-9025f9ff4408",
  "Name": "",
  "Status": "in-use",
  "Size": 10,
  "Attached to": [
    {
      "id": "218b321c-7b7f-4f5b-af4b-9025f9ff4408",
      "attachment_id": "6347cad1-5d52-4698-b3f1-b0a746a2920d",
      "volume_id": "218b321c-7b7f-4f5b-af4b-9025f9ff4408",
      "server_id": "5b5bc409-ec90-4a12-b659-d2b4d04eb419",
      "host_name": "openstack-bck",
      "device": "/dev/vda",
      "attached_at": "2024-02-28T21:23:06.000000"
    }
  ]
},
{
```

(continues on next page)

(continued from previous page)

```
    "ID": "c231ae83-ef6f-488f-b479-5ab328ea3b52",
    "Name": "c231ae83-ef6f-488f-b479-5ab328ea3b52",
    "Status": "in-use",
    "Size": 1,
    "Attached to": [
      {
        "id": "c231ae83-ef6f-488f-b479-5ab328ea3b52",
        "attachment_id": "289c3223-3a23-4a48-975f-6d2a2600a7e9",
        "volume_id": "c231ae83-ef6f-488f-b479-5ab328ea3b52",
        "server_id": "74364187-d6e7-431d-a80a-6275f88a69ce",
        "host_name": "openstack-bck",
        "device": "/dev/vda",
        "attached_at": "2024-02-23T10:18:19.000000"
      }
    ]
  }
]
```

Interactive Delete

It is possible to delete backup(s), snapshot(s) or volume(s) by running the `/opt/bacula/bin/openstack-vm-execute-interactive-delete` procedure with relevant parameters

Note: The delete operation is sent through Openstack API, and it has the force flag activated by default.

Parameters

- `-b <backup-ID>`: For interactive backup deletion if no ID is specified the procedure will go through all backups asking for deletion.
- `-c <admin-openrc>` Path to modified admin-openrc.sh DEFAULT=/opt/bacula/admin-openrc.sh.
- `-s <snapshot-ID>` For interactive snapshot deletion if no ID is specified the procedure will go through all snapshots asking for deletion.
- `-t <tools>` Path to openstack-vm-scripts DEFAULT=/opt/bacula/bin/
- `-p <project-id>` - If set restore will target VM in this project only.
- `-v <volume-ID>` For interactive volume deletion if no ID is specified the procedure will go through all volumes asking for deletion.
- `-h` Display help

Example

The options of this procedure are analog to other operations, but used for backup/snapshot/volume deletion instead.

Interactive delete of backups would be `/opt/bacula/bin/openstack-vm-execute-interactive-delete -b`

With an output looking like this:

```
root@openstack-bck:~# /opt/bacula/bin/openstack-vm-execute-interactive-delete␣
↪-b
INTERACTIVE BACKUP DELETE START

Would you like to delete BACKUP
ID=<backup1_ID>
NAME=<backup1_name>
DESCRIPTION=Backup done by Bacula Enterprise INSTANCE=<instance_name>␣
↪DATE=Tue Mar 19 16:34:03 UTC 2024

Start deletion [y]es / [N]o ?y
Delete command sent


Would you like to delete BACKUP
ID=<backup2_ID>
NAME=<backup2_name>
DESCRIPTION=Backup done by Bacula Enterprise INSTANCE=<instance_name>␣
↪DATE=Tue Mar 19 16:34:03 UTC 2024

Start deletion [y]es / [N]o ?y
Delete command sent

...

INTERACTIVE BACKUP DELETE FINISHED
```

Backup and Restore Strategies

Installing Bacula Client on Each Guest

This strategy works by installing a Bacula Enterprise File Daemon on every virtual machine as if they were regular, physical clients. In order to optimize the I/O usage of Openstack, the user will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to spread backup jobs over the backup window. Since all VMs could use the same storage on the Openstack hypervisor, running all backup jobs at the same time could create a bottleneck on the disk/network subsystem since Bacula will walk through all filesystems to open/read/close/stat files.

Installing the Bacula Enterprise File Daemon on each virtual machine permits to manage virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files
- Checksum of individual files for Virus and Spyware detection

- Verify Jobs
- File/Directory exclusion (such as swap or temporary files)
- File level compression
- Accurate backups.

Cinder Driver Backup with Openstack Plugin

With the Cinder driver strategy, the Bacula Enterprise Openstack-VM will save all Openstack volume`s at the raw level, in the Openstack context.

Bacula's Openstack-VM plugin will read and save the content of Openstack instance using Cinder backup API.

Cinder allows the user to integrate various storage solutions into the Openstack cloud. It does this by providing a stable interface for hardware providers to write drivers that allow the usage of Cinder volumes backup capabilities.

Troubleshooting

This article presents recommended solutions for common issues that may arise while using the Openstack Cinder Plugin.

- `D: cannot unpack non-iterable VolumeBackupsRestore object`

At restore time the restore volume command might output the following message `D: cannot unpack non-iterable VolumeBackupsRestore object`. This issue shouldn't impact the restore process and it can be ignored.

- `W: Openstack returned too many values to unpack (expected 2)`

At restore time Openstack might output a warning `W: Openstack returned too many values to unpack (expected 2)`. Restore should go through regardless and not be impacted by the message. This issue also happens when using the OpenStack CLI, and a bug report has been reported to the Openstack team.

Limitations

The following article presents limitations of Openstack Cinder Plugin.

- After a restore-procedure only the volumes are restored. The specific restored instance must be manually

restored and by attaching the relevant volumes to a new instance.

- Currently, only full level instance(s) volume(s) backups are possible.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

OpenStack Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This document aims at presenting the reader with information about the **Bacula Enterprise OpenStack (Open Source Cloud Computing Infrastructure) Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features.

Features

Bacula Enterprise is a highly secure and reliable backup and recovery software that supports a wider range of databases and hypervisor types than nearly any other solution currently available. For instance, it integrates effortlessly with OpenStack, providing a particularly robust backup and recovery solution, even in highly demanding environments. OpenStack itself is a powerful and intricate environment, designed to facilitate the management of computer infrastructure. Bacula's OpenStack module aims to streamline and enhance the backup and restore procedure of OpenStack resources such as servers, volumes, images and flavors.

- Snapshot-based online backup of any volume, server, or image.
- Full, Incremental and Differential block level image backup.
- Backups are consistent at image, volume and host configuration level.
- Ability to restore complete virtual machine image and other volumes.
- Precise inclusion/exclusion mechanism to control the backup target.
- Automatic backup configuration via hypervisor VM scanning routines.
- Automatic snapshot cleanup processes.
- Compatibility with *Deduplication* techniques.
- Ability to send backup data to local volume, network volume, block storage, tape or cloud.

Best Practices

Backup of the .bmp, .sha Files and the /opt/bacula/etc/openstack.conf File

As the `.bmp` and `.sha` files generated in the `/opt/bacula/working/openstack` directory are very important for the Incremental backups, it is recommended to have a specific backup job to backup these files regularly, even if they are already included in the current OpenStack backups, so they can be easily restored in the case of a disaster with the Bacula proxy server.

Additionally, the `/opt/bacula/etc/openstack.conf` file, which is also important for the OpenStack backups, can be also included in the same backup job.

This backup job should be configured to run daily using Full level.

For example, below are the Job and Fileset configurations recommended to use for the backup of the `.bmp` and `.sha` files in the `/opt/bacula/working/openstack` directory, along with the `/opt/bacula/etc/openstack.conf` file:

```
Job {
    Name = "openstack_working_and_conf_files-job"
    Type = "Backup"
    Client = "bacula-proxy-vm-rhel9-fd"
    Fileset = "openstack_working_and_conf_files-fileset"
    JobDefs = "BackupsToDisk"
    Messages = "Default"
    Pool = "DiskBackup365d"
    Schedule = "DailyFull"
    Storage = "DiskAutochanger"
}

Fileset {
        Name = "openstack_working_and_conf_files-fileset"
        Include {
              Options {
          Compression = LZO
                      Signature = "Md5"
              }
            File = "/opt/bacula/working/openstack"
            File = "/opt/bacula/etc/openstack.conf"
    }
}
```

In the case of a partial or total disaster of the Bacula proxy server, it is possible to create a new Bacula proxy server, install the Bacula File Daemon and the OpenStack Plugin, and recover the `.bmp` and `.sha` files, as well as the `/opt/bacula/etc/openstack.conf` file from the latest successful Full backup.

Backup and Restore Strategies

This article presents information regarding backup and restore strategies of the OpenStack Plugin.

Installing Bacula Client on Each Guest

This strategy involves the installation of a Bacula Enterprise File Daemon on each server, treating them as if they were standard physical clients. In order to optimize the I/O usage of servers hosted on the Nova compute service, the user will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to distribute backup jobs over the backup window. Given that all servers may share the same storage on Cinder` block storage, running all backup jobs simultaneously could lead to a bottleneck on the volume/network subsystem, as Bacula will walk through all filesystems to open, read, close and stat files.

Installing a Bacula Enterprise File Daemon on each server allows for the management of virtual servers in the same manner as physical servers, while also enabling the use of all features offered by Bacula Enterprise, such as:

- Quick restores of individual files
- Checksums of individual files for Virus and Spyware detection
- Verify Jobs
- File/Directory exclusion (such as swap or temporary files)
- File level compression

- Accurate backups
- Additionally, various other plugins are available.

Instance Backup with OpenStack Plugin

With the instance backup strategy, the Bacula Enterprise OpenStack Plugin will save the instance volumes at the raw level within the OpenStack context.

The Bacula OpenStack Plugin will query the guest servers via the OpenStack API to read and save the content of server volumes by employing snapshots and the native Openstack Filesystem. During backups, the OpenStack Plugin will ensure the integrity of volume images and the configurations of guest servers, facilitating the restores with their original parameters.

All those operations are handled by an additional proxy server, which is described in the next section.

Proxy Server

To handle backup and restore operations in the OpenStack environment,it is essential to establish a proxy server. This particular server handles most of the operations (such as snapshot management, IO, etc.) during backup and restore. These operations will be discussed in more detail in the subsequent sections.

The proxy server must have the following characteristics:

- Linux-based operating system
- Bacula File Daemon installed (`bacula-enterprise-client package`) and running, configured as a Client Resource on the Bacula Director
- OpenStack Plugin (`bacula-enterprise-openstack-plugin package`) installed
- Network access to the OpenStack REST API.

Ingestion

The ingestion of a single virtual volume is a specific protocol which takes the following steps:

- The proxy server generates a snapshot of the current volume.
- The snapshot is then mounted as a new volume on the proxy server.
- The list of modified block regions, along with their corresponding hashes and raw volume images, is exported to a Bacula Storage Daemon.
- The volume is then detached from the proxy server.
- Finally, the volume and its associated snapshot are deleted.

Backups can be executed for a guest server regardless of its power state (running or halted). Backups can be performed on any volume available in the Cinder block storage.

The backup will create the following backup files for each guest server:

- A name file that associates the guest server name with its UUID: `//\@Openstack/nova/server/<serverUuid>_<serverName>.name`
- The configuration file of the guest server: `/\@Openstack/nova/server/<serverUuid>/<serverUuid>.conf`

- A list of data regions for each virtual volume: `/\@Openstack/nova/server/<serverUuid>/<volumeUuid>.bmp`
- A file to reconstruct data regions and their hashes for each virtual volume `/\@Openstack/nova/server/<serverUuid>/<volumeUuid>.bmpsha`
- A raw data file for each virtual volume: `/\@Openstack/nova/server/<serverUuid>/<volumeName>_<volumeUuid>.bvmdk`

The backup will also create the following backup files for each volume not attached to a server:

- A list of data regions for each virtual volume: `/\@Openstack/cinder/volume/<volumeUuid>.bmp`
- A file to reconstruct data regions and their hashes for each virtual volume: `/\@Openstack/cinder/volume/<volumeUuid>.bmpsha`
- A raw data file for each virtual disk: `/\@Openstack/cinder/volume/<volumeName>_<volumeUuid>.bvmdk`

Note: It is possible for a volume to have no name. In that case only their UUID will be displayed.

The backup will create the following files in the `/opt/bacula/working/openstack` directory:

```
# ls -l
total 32
-rw-r-----. 1 root bacula  4609 Apr 30 15:54 openstack-cirros-instance1-job.
↪2025-05-28_14.01.01_37_c4d13b99-5bdc-4b45-b673-bb6a81e443a1.bmp
-rw-r-----. 1 root bacula 21314 Apr 30 15:54 openstack-cirros-instance1-job.
↪2025-05-28_14.01.01_37_c4d13b99-5bdc-4b45-b673-bb6a81e443a1.sha
```

These files are important for Incremental backups. They contain the list of blocks and their checksum. In the next Incremental backup, if the checksum differs, the block is included in the backup.

Note: These files should not be deleted. If the list of blocks and their checksum are not available, the plugin needs to rebuild this list, resulting in a backup all blocks. Then, a subsequent Incremental level backup will backup all the blocks, as the plugin lacks the information about the changed blocks.

At restore time the user can identify the guest server by using the UUID to mark the corresponding files:

```
+-----------------------------------------------------------------------------
↪-----------------------------+
| filename                                                                    ␣
↪                              |
+-----------------------------------------------------------------------------
↪-----------------------------+
| /@openstack/nova/flavor/ds1G_d2                                             ␣
↪                              |
| /@openstack/nova/server/22d27bdf-1cdb-461e-8f61-a641e36f9ed7_almalinux9-
↪instance1.name                |                                              ␣
↪          |
| /@openstack/nova/server/22d27bdf-1cdb-461e-8f61-a641e36f9ed7/eafffb4a-634c-
↪4751-93e2-aa75778b8e5b.bvmdk  |
| /@openstack/nova/server/22d27bdf-1cdb-461e-8f61-a641e36f9ed7/eafffb4a-634c-
```

(continues on next page)

(continued from previous page)

```
↪4751-93e2-aa75778b8e5b.bmpsha |
| /@openstack/nova/server/22d27bdf-1cdb-461e-8f61-a641e36f9ed7/eafffb4a-634c-
↪4751-93e2-aa75778b8e5b.bmp    |
| /@openstack/nova/server/22d27bdf-1cdb-461e-8f61-a641e36f9ed7/22d27bdf-1cdb-
↪461e-8f61-a641e36f9ed7.conf   |
+-------------------------------------------------------------------------------
↪-----------------------------+
```

Installation

In order to use the OpenStack Plugin, a Bacula File Daemon has to be installed on a Nova server running in the OpenStack environment. This instance is referred to as a proxy server throughout this document.

The proxy server needs to have network access to the OpenStack API. The default network adapter should be enough.

Since all backup interactions are conducted over the network, the Bacula Enterprise File Daemon must have access to the required OpenStack endpoints.

This is an example of the OpenStack endpoint list:

```
# openstack endpoint list
+----------------------------------+-----------+--------------+---------------
↪-+---------+-----------+-----------------------------------------------+
| ID                               | Region    | Service Name | Service Type ␣
↪ | Enabled | Interface | URL                                           |
+----------------------------------+-----------+--------------+---------------
↪-+---------+-----------+-----------------------------------------------+
| 43dccd9ef03348d6ba2607a98eca3fcb | RegionOne | glance       | image        ␣
↪ | True    | public    | http://10.0.100.35/image                      |
| 5aff87259bd84784bcd062b11867cf3b | RegionOne | cinder       | block-storage␣
↪ | True    | public    | http://10.0.100.35/volume/v3/$(project_id)s   |
| 638573f7563546de91945ddbfbff0147 | RegionOne | keystone     | identity     ␣
↪ | True    | public    | http://10.0.100.35/identity                   |
| 7b117b1f5cd845d9aee6963455839115 | RegionOne | cinderv3     | volumev3     ␣
↪ | True    | public    | http://10.0.100.35/volume/v3/$(project_id)s   |
| 892289c4035e4220b8e4ed4333043196 | RegionOne | nova         | compute      ␣
↪ | True    | public    | http://10.0.100.35/compute/v2.1               |
| 8cf737698e5d49db84b8944877cdc44b | RegionOne | neutron      | network      ␣
↪ | True    | public    | http://10.0.100.35:9696/networking            |
| ab7f79fd8c3d4a7c8c46c96a5744dd98 | RegionOne | nova_legacy  | compute_
↪legacy | True    | public    | http://10.0.100.35/compute/v2/$(project_id)s␣
↪|
| e0ce34ec34834a8b9b6049dc697577d5 | RegionOne | placement    | placement    ␣
↪ | True    | public    | http://10.0.100.35/placement                  |
+----------------------------------+-----------+--------------+---------------
↪-+---------+-----------+-----------------------------------------------+
```

OpenStack Installation with BIM

In order to install the OpenStack Plugin with BIM, install the Bacula File Daemon with BIM and choose to install the OpenStack Plugin during the Bacula File Daemon installation.

Click here for more details on the plugin installation process with BIM.

See also:

See an alternative way of installing the OpenStack Plugin - Openstack Installation with Package Manager.

OpenStack Installation with Package Manager

Prerequisites

The `Plugin Directory` directive of the `File Daemon` resource in `/opt/bacula/etc/bacula-fd.conf` should point to the location where the `openstack-fd.so` plugin is installed. The default directory is: `/opt/bacula/plugins`

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Installation Steps

To install the Bacula File Daemon, refer to PackageManagerBERHELCentOS.

To install the OpenStack Plugin, it is necessary to configure the OpenStack Plugin repository and install the package using a package manager.

For example, the OpenStack Plugin repository for a Red Hat repository, named `/etc/yum.repos.d/Bacula-Enterprise-OpenStack-plugin.repo`, should have the following contents:

```
[Bacula-Enterprise-OpenStack-plugin]
name= Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@@customer@@/rpms/openstack/@@bee-
↪version@@/@@rhel@@-@@arch@@/
enabled=1
protect=0
gpgcheck=1
```

Then, use either `dnf install` or `yum install` to install the OpenStack Plugin:

```
# yum install bacula-enterprise-openstack-plugin
```

or

```
# dnf install bacula-enterprise-openstack-plugin
```

Configuration

The following article presents how to configure backup and restore using the OpenStack Plugin.

The plugin uses general parameters, which apply to backup, restore or query operations, as well as more specific parameters parameters for backup or restore only.

These parameters are configured within the `Plugin` directive defined in the "Include" section of a Fileset resource, which is used in a Job resource along with the Client resource to setup a backup job.

This is how an OpenStack Fileset is configured:

```
Fileset {
    Name = "Openstack_plugin-fileset"
    Include {
        Plugin = "openstack: <parameter1>=<parameter1_value> <parameter2>=
↪<parameter2_value>"
    }
}
```

The OpenStack Plugin parameters will be explained in the next sections.

Automatic Object Integration

Since Bacula version 16.0.7, a new solution has been introduced, so that each object can be backed up separately with different Jobs to maximize the throughput and the resiliency. It is highly recommended to use this new solution for that purpose - *Automatic Object Integration (Scan Plugin)*. See an example for OpenStack.

General Parameters

The following OpenStack Plugin parameters impact Backup and Restore Jobs as well as the Query bconsole commands.

Table 11: Openstack General Parameters

Parameter	Values	Default	Description
proxy_	<Str		UUID or name of the proxy server used to run the Bacula File Daemon with the Openstack Plugin. This parameter is required.
abort_(	[= <0 or 1>]	0	Specifies whether the plugin should abort on fatal errors during backup or restore. This parameter is optional.
OS_US	<Str		Specifies the username to access the Openstack API.
OS_PA	<Str		Specifies the password to access the Openstack API
OS_AU	<Str		Specifies the endpoint to access the Openstack API
OS_US	<Str		Specifies the domain that accesses the Openstack API
OS_PR	<Str		Specifies the project that accesses the Openstack API
openrc	<Str	/opt/bacula/etc/opensta conf	Specifies the path to an OpenStack RC file to automatically gather connection information.
verbose	[= <0 or 1>]	0	This parameter activate verbose output in Bacula joblog.
debug	[0 - 9]	0	Specifies the debug level. 0 is no debug, 9 is the highest level. Warnings and errors are always sent to the joblog and if any debug level is set, those messages are sent to the debug file as well. For the OpenStack Plugin, 1 displays debug level message, 2 displays trace level message. Any value higher than 2 displays additional information about external libraries that handle those values on their own. This parameter is optional.

Note: Parameters sent from the restore object have precedence over everything. Parameters sent from the CLI have precedence over parameters from the configuration file.

Plugin Parameters in the OpenStack RC File

Usually, to manage the OpenStack environment using the OpenStack command-line client, a file called OpenStack RC file can be used to set the required environment variables to access the OpenStack project API. More details about the contents and how the OpenStack RC file can be generated can be found in the OpenStack documentation: https://docs.openstack.org/newton/user-guide/common/cli-set-environment-variables-using-openstack-rc.html

This is an example of an OpenStack RC file for the demo OpenStack project:

```
unset OS_SERVICE_TOKEN
export OS_USERNAME=demo
```

(continues on next page)

(continued from previous page)

```
export OS_PASSWORD='thisIsAPassword'
export PS1='[\u@\h \W(keystone_demo)]\$ '
export OS_AUTH_URL=http://1.1.11.111:2222/v3

export OS_PROJECT_NAME=demo
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
```

This file can be used as a configuration file for the Bacula OpenStack Plugin to automatically fetch the values for the `OS_USERNAME`, `OS_PASSWORD`, `OS_AUTH_URL`, `OS_USER_DOMAIN_NAME`, and `OS_PROJECT_NAME` plugin options.

When using an OpenStack RC file with the required environment variables with the Bacula OpenStack Plugin, the `openrc=<pathToOpenStackRCfile>` parameter must be configured. By default, the plugin will check if there is the `/opt/bacula/etc/openstack.conf` as the OpenStack RC file if the `openrc` option is not configured.

Warning: The OpenStack RC file downloaded from the OpenStack dashboard requires user interaction for proper functionality. To ensure the plugin operates correctly, it is recommended to modify the downloaded OpenStack RC file by commenting out the lines `echo "Please enter your OpenStack Password for project $OS_PROJECT_NAME as user $OS_USERNAME: "`, `read -sr OS_PASSWORD_INPUT`, and `export OS_PASSWORD=$OS_PASSWORD_INPUT`, while adding `OS_PASSWORD_INPUT=<password_value>` with the appropriate password value.

Backup Parameters

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Important Notes

- The use of regular expressions in the parameters `include=` and `exclude=` must be a Java compatible regular expression.
- In order to be backed up, servers/images/volumes must match the `include=...` predicate and not match the `exclude=...`. However, any item that matches the `<server|image|volume>=...` options will be backed up regardless of the `include/exclude` specifications.
- By default all items match the `include` predicate and not the `exclude`. Therefore, if none of the parameters `<server|image|volume>=...`, `include=...` and `exclude=...` are provided, all available elements hosted in the OpenStack environment will be backed up. On the other hand, if the parameter `<server|image|volume>=...` is specified, all elements will no longer match the `include` predicate. This means that if only `<server|image|volume>=...` parameter is specified, no other items will be backed up.
- For any service, if any field from `<element>_include=...`, `<element>_exclude=...` or `<element>=...` is specified, the plugin will issue a backup for said service regardless of the value of `<service>_<element>_backup=...`.

Note: The Nova servers is the only service where the backup of all Nova servers is enabled by default.

See Fileset Examples for examples of `include/exclude/element` setups.

Restore Parameters

Table 12: OpenStack Restore Parameters

Parameter	Values	Default	Description
proxy_se	`<String`		UUID or name of the proxy server used to run the Bacula File Daemon with the OpenStack Plugin.
abort_on_	`[= <0 or 1>]`	`0`	Specifies whether the plugin should abort on fatal errors during backup or restore. This parameter is optional.
OS_USE	`<String`		Specifies the username to access the OpenStack API.
OS_PAS	`<String`		Specifies the password to access the OpenStack API
OS_AUT	`<String`		Specifies the endpoint to access the OpenStack API
OS_USE	`<String`		Specifies the domain that accesses the OpenStack API
OS_PRO	`<String`		Specifies the project that accesses the OpenStack API
openrc	`<String`	`/opt/bacula/etc/openstack conf`	Specifies the path to an OpenStack rc file to automatically gather connection information.
server_re	`<String`		Name to attribute to the restored server.
server_re	`<String`		Prefix to prepend to all restored server name.
server_re	`[= yes or no or auto>]`	`no`	If set to `no` the plugin will create a bogus network to attach to restore server. If `yes`, the plugin will try to find a network matching the server's original network. If `auto`, the plugin will let OpenStack system try to assign a network to restored server.
verbose	`[= <0 or 1>]`	`0`	This parameter activate verbose output in Bacula joblog.
debug	`[0 - 9]`	`0`	Specifies the debug level. `0` is no debug, `9` is the highest level. Warnings and errors are always sent to the joblog.

Important Notes

At restore time, if `server_restore_name` is not set, the plugin will use the original Nova server name.

If a Nova server with the same name already exists, the plugin will append the `-<index>` to the restored Nova server name, and it will not replace the existent Nova server.

Fileset Examples

In the example below, all the Nova servers and all the existent volumes (not in-use by Nova servers) will be backed up:

```
Fileset {
    Name = "Openstack_all_instances-fileset"
    Include {
        Plugin = "openstack: OS_AUTH_URL=http://11.11.11.111:2222/v3 OS_
↪USERNAME=admin OS_PASSWORD=123456 OS_USER_DOMAIN_NAME=Default OS_PROJECT_
↪NAME=admin proxy_server=12345678-aaaa-bbbb-cccc-012345678901"
    }
}
```

See also:

For more details about: `proxy_server` and the OS_* parameters, see: GeneralParametersOpenstack.

Note: In the next examples it is assumed that the user copied or symlinked the project OpenStack RC file as `/opt/bacula/etc/openstack.conf`.

In the example below, a single Nova server named “nfs-instance1” will be backed up.

```
Fileset {
    Name = "Openstack_nfs_instance1-fileset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server=nfs-instance1"
    }
}
```

See also:

For more details about `server`, see: BackupParametersOpenstack.

In the example below, using the `server_include` option, both the Nova servers named nfs-instance1 and nfs-instance2 will be backed up.

```
Fileset {
    Name = "Openstack_nfs_instance1_instance2-filset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server_include=nfs-instance[12]"
    }
}
```

See also:

For more details about `server_include`, see: BackupParametersOpenstack.

In the example below, using the `server_include` option, all the Nova servers whose names start with “nfs-” will be backed up.

```
Fileset {
    Name = "Openstack_all_nfs_instances-fileset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server_include=nfs-.*"
    }
}
```

In the example below, using both the `server_include` and the `server_exclude` options, all the Nova servers whose names begin with “nfs-” will be backed up, excpet the Nova server named nfs-instance1.

```
Fileset {
    Name = "Openstack_all_nfs_instances_not_nfs_instance1-fileset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server_include=nfs-.* server_exclude=nfs-instance1"
    }
}
```

See also:

For more details about `server_exclude`, see: BackupParametersOpenstack.

In the example below, using both the `server_include` and the `server_exclude` options, and the `server` option, all the Nova servers whose names begin with “nfs-” and the Nova server named “prod-main” will be backed up, except the Nova servers named nfs-instance1 and nfs-instance2.

```
Fileset {
    Name = "Openstack_all_nfs_instances_and_prod-main_and_not_nfs_instance1-
↪fileset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server_include=nfs-.* server_exclude=nfs-instance[12]␣
↪server=prod-main"
    }
}
```

In the example below, using the `server_include` and the `cinder_volume_backup` options, the Nova servers whose names begin with “nfs-” will be backed up as well as all volumes not attached to any Nova server. These volumes have its status as “available” in the OpenStack server.

```
Fileset {
    Name = "Openstack__all_nfs_instances_and_available_volumes-fileset"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 cinder_volume_backup=1 server_include=nfs-.* "
    }
}
```

See also:

For more details about `cinder_volume_backup`, see: BackupParametersOpenstack.

In the example below, using the `server_include` and the `volume` options, the Nova servers whose names begin with "nfs-" will be backed up as well a single cinder volume named secrets.

```
Fileset {
    Name = "Openstack_nfs_i"
    Include {
        # Using authentication from /opt/bacula/etc/openstack.conf
        Plugin = "openstack: proxy_server=12345678-aaaa-bbbb-cccc-
↪012345678901 server_include=nfs-.* volume=secrets"
    }
}
```

See also:

For more details about: `volume`, see: BackupParametersOpenstack.

Operations

The following article describes details regarding backup, ingestion, restore with Bacula Enterprise OpenStack Plugin.

Backup

Backup of Nova Servers

The backup of a single Nova server takes the following steps:

- Export the Nova server metadata configuration for future restore.
- For each volume attached to the Nova server, the plugin issues a snapshot of the volume.
- Backup all volumes attached to the Nova server by ingestion on the proxy server.
- Cleanup the snapshot and volume created from snapshot after it is stored in the Bacula Enterprise Storage.

This is an example of the Job and Fileset resources to backup an instance named `cirros-instance1`, using the OpenStack RC file `/opt/bacula/etc/openstack.conf` located in the proxy server:

```
Job {
  Name = "openstack-cirros-instance1-job"
  Description = "Backup cirros-instance1"
  Type = "Backup"
  Accurate = yes
  Client = "bacula-proxy-vm-rhel9-fd"
  Fileset = "openstack-cirros-instance1-fileset"
  JobDefs = "BackupsToDisk"
  Messages = "Default"
  Pool = "DiskBackup365d"
  Schedule = "Manual"
  Storage = "DiskAutochanger"
```

(continues on next page)

(continued from previous page)

```
}

Fileset {
  Name = "openstack-cirros-instance1-fileset"
  Include {
   Options {
    IgnoreCase = yes
    OneFs = no
    Signature = Md5
   }
   # Using authentication from /opt/bacula/etc/openstack.conf
   Plugin = "openstack: proxy_server=7172b28d-4d39-4e5a-9ece-d2c76768fd2b␣
↪server=cirros-instance1 verbose=1"
  }
}
```

And a successful joblog of the Nova servers backup job run, using `verbose=1` in the `Plugin` line in the Fileset:

```
bacula-dir JobId 166: Start Backup JobId 166, Job=openstack-cirros-instance1-
↪job.2025-05-05_16.25.00_51
bacula-dir JobId 166: Connected to Storage "DiskAutochanger" at 10.0.99.
↪131:9103 with TLS
bacula-dir JobId 166: Using Device "DiskAutochanger_Dev1" to write.
bacula-dir JobId 166: Connected to Client "bacula-proxy-vm-rhel9-fd" at 10.0.
↪100.35:9102 with TLS
bacula-proxy-vm-rhel9-fd JobId 166: Connected to Storage at 10.0.99.131:9103␣
↪with TLS
bacula-sd JobId 166: Volume "Vol-0001" previously written, moving to end of␣
↪data.
bacula-sd JobId 166: Ready to append to end of Volume "Vol-0001" size=2,149,
↪713,765
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Plugin log of this job␣
↪available in: "/opt/bacula/working/openstack-0.log"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Trying to get OSClient for␣
↪endpoint=http://10.0.100.35/identity/v3 user=admin password=**********␣
↪domain=Default project=demo
bacula-proxy-vm-rhel9-fd JobId 166: openstack: NOVA Server Backup START
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Start Server backup "cirros-
↪instance1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Server configuration backup␣
↪for "257a54e7-73ac-4c10-bd01-a37f61f9a05d"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Backing up empty name file␣
↪NAME="/nova/server/257a54e7-73ac-4c10-bd01-a37f61f9a05d_cirros-instance1.
↪name"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Server flavor backup
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Start backup for volume
↪"485b7cec-2c92-4d77-a1f8-ea981d6192cc"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Snapshot volume "485b7cec-2c92-
↪4d77-a1f8-ea981d6192cc"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Attaching volume "ff3c50df-
↪2e92-46d9-84a6-0adc0bb53ebb" created from snapshot to proxy
```

(continues on next page)

(continued from previous page)

```
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Proxy volume "ff3c50df-2e92-
↪46d9-84a6-0adc0bb53ebb" for volume "485b7cec-2c92-4d77-a1f8-ea981d6192cc"␣
↪attach SUCCESS
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Backup of volume "485b7cec-
↪2c92-4d77-a1f8-ea981d6192cc" SUCCESS      FD Bytes Written 1073747992
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Detach volume "ff3c50df-2e92-
↪46d9-84a6-0adc0bb53ebb" from proxy server
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Deleting detached volume
↪"ff3c50df-2e92-46d9-84a6-0adc0bb53ebb"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Issue volume delete command␣
↪for volume ID=ff3c50df-2e92-46d9-84a6-0adc0bb53ebb FORCE=true
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Deleting volume snapshot
↪"b3752123-792d-4c8c-a146-0b600fbf44fb"
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Issue snapshot delete command␣
↪for snapshot ID=b3752123-792d-4c8c-a146-0b600fbf44fb
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Backup done for volume
↪"485b7cec-2c92-4d77-a1f8-ea981d6192cc" total bytes sent 1073747992
bacula-proxy-vm-rhel9-fd JobId 166: openstack: Server backup done for "cirros-
↪instance1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)" 1 volumes found
bacula-proxy-vm-rhel9-fd JobId 166: openstack: NOVA Server backup DONE
bacula-sd JobId 166: Elapsed time=00:07:54, Transfer rate=2.265 M Bytes/second
bacula-sd JobId 166: Sending spooled attrs to the Director. Despooling 4,056␣
↪bytes ...
bacula-dir JobId 166: Bacula Enterprise bacula-dir 18.1.4 (02May25):
 Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Blue
 JobId:                  166
 Job:                    openstack-cirros-instance1-job.2025-05-05_16.25.00_51
 Backup Level:           Full
 Client:                 "bacula-proxy-vm-rhel9-fd" 18.1.4 (02May25) x86_64-
↪redhat-linux-gnu-bacula-enterprise,redhat,(Blue
 FileSet:                "openstack-cirros-instance1-fileset" 2025-05-05␣
↪15:07:44
 Pool:                   "DiskBackup365d" (From Command input)
 Catalog:                "BaculaCatalog" (From Client resource)
 Storage:                "DiskAutochanger" (From Command input)
 Scheduled time:         05-May-2025 16:25:00
 Start time:             05-May-2025 16:25:03
 End time:               05-May-2025 16:32:57
 Elapsed time:           7 mins 54 secs
 Priority:               10
 FD Files Written:       9
 SD Files Written:       9
 FD Bytes Written:       1,073,916,130 (1.073 GB)
 SD Bytes Written:       1,073,922,513 (1.073 GB)
 Rate:                   2265.6 KB/s
 Software Compression:   None
 Comm Line Compression:  96.4% 27.9:1
 Snapshot/VSS:           no
 Encryption:             no
 Accurate:               yes
 Volume name(s):         Vol-0001
```

(continues on next page)

(continued from previous page)

```
Volume Session Id:      13
Volume Session Time:    1746429478
Last Volume Bytes:      3,224,569,980 (3.224 GB)
Non-fatal FD errors:    0
SD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            Backup OK
```

Backup of Glance Images

The backup of the Glance images will generate a temporary file named `img-<number>.tmpimg` in the `/opt/bacula/working` directory of the Bacula proxy server:

```
# ls -l /opt/bacula/working/*.tmpimg
-rw-r-----. 1 root bacula 0 May 13 11:18 /opt/bacula/working/img-
↪9359619567641477501.tmpimg
```

Upon completion, this file is stored in the Bacula Storage used by the backup job and subsequently deleted from the `/opt/bacula/working` directory.

Note: It is recommended to have enough space in the `/opt/bacula/working` directory of the Bacula proxy server to store the `img-<number>.tmpimg` temporary file for Glance images backups. The space should be at least equivalent to the size of the largest Glance image in the OpenStack server. When running concurrent backups of Glance images, additional space will be necessary to hold the `img-<number>.tmpimg` temporary files generated by the concurrent jobs.

This is an example of the Job and Fileset resources to backup all the Glance images only, using the OpenStack RC `/opt/bacula/etc/openstack.conf` file located in the proxy server:

```
Job {
    Name = "openstack-images-job"
    Description = "Backup OpenStack images"
    Type = "Backup"
    Accurate = yes
    Client = "bacula-proxy-vm-rhel9-fd"
    Fileset = "openstack-images-fileset"
    JobDefs = "BackupsToDisk"
    Messages = "Default"
    Pool = "DiskBackup365d"
    Schedule = "Manual"
    Storage = "DiskAutochanger"
}

Fileset {
    Name = "openstack-images-fileset"
    Include {
      Options {
        Signature = Md5
      }
```

(continues on next page)

(continued from previous page)

```
      # Using authentication from /opt/bacula/etc/openstack.conf
      Plugin = "openstack: proxy_server=7172b28d-4d39-4e5a-9ece-d2c76768fd2b␣
↪glance_image_backup=1 nova_server_backup=0 verbose=1"
    }
}
```

Note: The above backup will not include Nova servers, but Glance images only, by using nova_server_backup=0.

And a successful joblog of the Glance images backup job run, using verbose=1 in the Plugin line in the Fileset:

```
bacula-dir JobId 215: Start Backup JobId 215, Job=openstack-images-job.2025-
↪05-13_11.14.43_46
bacula-dir JobId 215: Connected to Storage "DiskAutochanger" at 10.0.99.
↪131:9103 with TLS
bacula-dir JobId 215: Using Device "DiskAutochanger_Dev1" to write.
bacula-dir JobId 215: Connected to Client "bacula-proxy-vm-rhel9-fd" at 10.0.
↪100.35:9102 with TLS
bacula-proxy-vm-rhel9-fd JobId 215: Connected to Storage at 10.0.99.131:9103␣
↪with TLS
bacula-sd JobId 215: Volume "Vol-0021" previously written, moving to end of␣
↪data.
bacula-sd JobId 215: Ready to append to end of Volume "Vol-0021" size=1,171,
↪586,428
bacula-proxy-vm-rhel9-fd JobId 215: openstack: Plugin log of this job␣
↪available in: "/opt/bacula/working/openstack-0.log"
bacula-proxy-vm-rhel9-fd JobId 215: openstack: Trying to get OSClient for␣
↪endpoint=http://10.0.100.35/identity user=admin password=**********␣
↪domain=Default project=demo
bacula-proxy-vm-rhel9-fd JobId 215: openstack: Trying to get OSClient for␣
↪endpoint=http://10.0.100.35/identity/v3/ user=admin password=**********␣
↪domain=Default project=demo
bacula-proxy-vm-rhel9-fd JobId 215: openstack: GLANCE Image Backup START
bacula-proxy-vm-rhel9-fd JobId 215: openstack: Backup of volume "6ae3c741-
↪1ec3-4680-8e45-f5e6f67f7aaf" SUCCESS      FD Bytes Written 21430416
bacula-proxy-vm-rhel9-fd JobId 215: openstack: Backup of volume "d8dd9b83-
↪bc66-471e-ab03-27968d9edb4f" SUCCESS      FD Bytes Written 546002518
bacula-proxy-vm-rhel9-fd JobId 215: openstack: GLANCE Image Backup DONE
bacula-sd JobId 215: Elapsed time=00:26:37, Transfer rate=355.3 K Bytes/second
bacula-sd JobId 215: Sending spooled attrs to the Director. Despooling 3,621␣
↪bytes ...
bacula-dir JobId 215: Bacula Enterprise bacula-dir 18.1.4 (02May25):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Blue
  JobId:                  215
  Job:                    openstack-images-job.2025-05-13_11.14.43_46
  Backup Level:           Full
  Client:                 "bacula-proxy-vm-rhel9-fd" 18.1.4 (02May25) x86_64-
↪redhat-linux-gnu-bacula-enterprise,redhat,(Blue
```

(continues on next page)

(continued from previous page)

```
  FileSet:                "openstack-images-fileset" 2025-05-07 15:27:05
  Pool:                   "DiskBackup365d" (From Job resource)
  Catalog:                "BaculaCatalog" (From Client resource)
  Storage:                "DiskAutochanger" (From Job resource)
  Scheduled time:         13-May-2025 11:14:41
  Start time:             13-May-2025 11:14:47
  End time:               13-May-2025 11:41:27
  Elapsed time:           26 mins 40 secs
  Priority:               10
  FD Files Written:       8
  SD Files Written:       8
  FD Bytes Written:       567,521,260 (567.5 MB)
  SD Bytes Written:       567,527,548 (567.5 MB)
  Rate:                   354.7 KB/s
  Software Compression:   None
  Comm Line Compression:  0.8% 1.0:1
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               yes
  Volume name(s):         Vol-0021
  Volume Session Id:      25
  Volume Session Time:    1746632396
  Last Volume Bytes:      1,739,607,794 (1.739 GB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Restore

The OpenStack Plugin provides two targets for restore operations:

- Restore to the OpenStack hypervisor as a new or the original Nova server.
- Restore the Nova server to a local directory to the Bacula proxy server as `.bvmdk`, `.conf`, `.sha` and `.bmp` files.

Restore to OpenStack

To use this method, the `where=/` restore option should be used.

When using `where=/` in the Restore Job, the Nova server is restored in the OpenStack server as the original Nova server, provided it does not already exist. In cases where a Nova server with the same name exists in the OpenStack server, the restore process will create a new Nova server using the original Nova server name plus the `-<number>` suffix to avoid the current Nova server to be overwritten by the restore.

It is always possible to setup a `server_restore_name` during the restore process for the restored Nova server name.

See also:

For more details on `server_restore_*` parameters, see: ref:*RestoreParametersOpenstack*.

This is an example, using bconsole, how to restore a Nova server to the OpenStack server:

```
* restore where=/ client=bacula-proxy-vm-rhel9-fd

...

  You have selected the following JobId: 166

  Building directory tree for JobId(s) 166 ...
  6 files inserted into the tree.

  You are now entering file selection mode where you add (mark) and
  remove (unmark) files to be restored. No files are initially added, unless
  you used the "all" keyword on the command line.
  Enter "done" to leave this mode.

  cwd is: /
  $ mark *
  6 files marked.
  $ lsmark
  +@openstack/
    +nova/
      *flavor/
        *cirros256_c1
      *server/
        *257a54e7-73ac-4c10-bd01-a37f61f9a05d_cirros-instance1.name
        *cirros-instance1_257a54e7-73ac-4c10-bd01-a37f61f9a05d/
          *257a54e7-73ac-4c10-bd01-a37f61f9a05d.conf
          *485b7cec-2c92-4d77-a1f8-ea981d6192cc.bmp
          *485b7cec-2c92-4d77-a1f8-ea981d6192cc.bmpsha
          *485b7cec-2c92-4d77-a1f8-ea981d6192cc.bvmdk
  $ done
  Bootstrap records written to /opt/bacula/working/bacula-dir.restore.21.bsr

  The Job will require the following (*=>InChanger):
     Volume(s)                 Storage(s)                SD Device(s)
  ===========================================================================

      Vol-0001                  DiskAutochanger           DiskAutochanger

  Volumes marked with "*" are in the Autochanger.


  6 files selected to be restored.

  Run Restore job
  JobName:         Restore
  Bootstrap:       /opt/bacula/working/bacula-dir.restore.11.bsr
  Where:           /
  Replace:         Always
```

(continues on next page)

(continued from previous page)

```
  FileSet:          BaculaConfigs
  Backup Client:    bacula-proxy-vm-rhel9-fd
  Restore Client:   bacula-proxy-vm-rhel9-fd
  Storage:          DiskAutochanger
  When:             2025-05-09 15:06:29
  Catalog:          BaculaCatalog
  Priority:         10
  Plugin Options:   *None*
  OK to run? (Yes/mod/no): yes
  Job queued. JobId=207
```

and then set any other required parameter.

The OpenStack plugin restores data as a new Nova server.

- Obtain the configuration metadata file.
- Acquire the raw disk data file(s).
- Add a new virtual volume on the proxy server (hotadd).
- Write all data regions back onto the volume(s).
- Detach all restored volumes from the proxy server.
- Once all volumes have been restored, create a new Nova server from the metadata configuration file. This step includes block device mapping for the restored volumes.

Here is an example of a successful restore job of a Nova server, where the plugin identifies an existing Nova server with the same `cirros-instance1` name. Consequently, the new Nova server is restored as `cirros-instance1-1`:

```
bacula-dir JobId 207: Start Restore Job Restore.2025-05-09_15.06.35_33
bacula-dir JobId 207: Restoring files from JobId(s) 166
bacula-dir JobId 207: Connected to Storage "DiskAutochanger" at 10.0.99.
↪131:9103 with TLS
bacula-dir JobId 207: Using Device "DiskAutochanger_Dev0" to read.
bacula-dir JobId 207: Connected to Client "bacula-proxy-vm-rhel9-fd" at 10.0.
↪100.35:9102 with TLS
bacula-proxy-vm-rhel9-fd JobId 207: Connected to Storage at 10.0.99.131:9103␣
↪with TLS
bacula-sd JobId 207: Ready to read from volume "Vol-0001" on File device
↪"DiskAutochanger_Dev0" (/opt/bacula/archive).
bacula-sd JobId 207: Forward spacing Volume "Vol-0001" to addr=2149713765
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Plugin log of this job␣
↪available in: "/opt/bacula/working/openstack-0.log"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Trying to get OSClient for␣
↪endpoint=http://10.0.100.35/identity/v3 user=admin password=**********␣
↪domain=Default project=demo
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Starting new server restore␣
↪for "257a54e7-73ac-4c10-bd01-a37f61f9a05d"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Nova flavor restore START
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Nova flavor with ID=c1 has the␣
↪same characteristics as original is already on on the system  Skip flavor␣
↪restore
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Nova flavor restore END
```

(continues on next page)

(continued from previous page)

```
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring cinder volume
↪"485b7cec-2c92-4d77-a1f8-ea981d6192cc" from server "cirros-instance1␣
↪(257a54e7-73ac-4c10-bd01-a37f61f9a05d)"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Starting new unmatch volume␣
↪restore for "485b7cec-2c92-4d77-a1f8-ea981d6192cc"
bacula-sd JobId 207: Elapsed time=00:06:54, Transfer rate=2.594 M Bytes/second
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Creating volume from␣
↪configuration to restore
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Attach restore volume to proxy␣
↪server
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Volume creation and attach for␣
↪restore SUCCESS
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring volume "485b7cec-
↪2c92-4d77-a1f8-ea981d6192cc" as "a74434a4-aece-431e-aaa3-dbbc06b86313"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Cinder volume restore SUCCESS
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Finalizing server restore
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Starting pairing between␣
↪server restore and unmatch volumes
bacula-proxy-vm-rhel9-fd JobId 207: openstack: After unmatch volume pair all␣
↪there are still 0 volumes without a match
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Detach all restored volume␣
↪from proxy server
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restore volume detach SUCCESS
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring nova server "cirros-
↪instance1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)"      1 volumes to attach
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Start server restore for
↪"cirros-instance1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)" as "cirros-
↪instance1-1"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Orignal flavor lookup success
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring server network:␣
↪generic network creation
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring server network:␣
↪create associtaed subnet
bacula-proxy-vm-rhel9-fd JobId 207: openstack: A generic network "cirros-
↪instance1-1 (65190167-d2ad-4cfb-abab-48ddff1b297b)" and its associtated␣
↪subnet "cirros-instance1-1 (dace49cf-1b41-442e-b870-a10731b13aa2)" has been␣
↪created
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Network associated with␣
↪restored server "cirros-instance1-1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)"␣
↪is "cirros-instance1-1 (65190167-d2ad-4cfb-abab-48ddff1b297b)"
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring server network:␣
↪generic network creation
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Restoring server network:␣
↪create associtaed subnet
bacula-proxy-vm-rhel9-fd JobId 207: openstack: A generic network "cirros-
↪instance1-1 (e9b9f2a7-7dbd-484c-b8db-72e12c1f8908)" and its associtated␣
↪subnet "cirros-instance1-1 (4fddfed4-b7ab-4440-9daa-0cf2bc35bf6a)" has been␣
↪created
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Server build
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Server creation/boot
bacula-proxy-vm-rhel9-fd JobId 207: openstack: Server restore for "cirros-
↪instance1 (257a54e7-73ac-4c10-bd01-a37f61f9a05d)" success new server is
```

(continues on next page)

(continued from previous page)

```
↪"cirros-instance1-1" ("1f83ab5a-afa6-4214-9acb-aa157d0ad23b")
bacula-dir JobId 207: Bacula Enterprise bacula-dir 18.1.4 (02May25):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Blue
  JobId:                  207
  Job:                    Restore.2025-05-09_15.06.35_33
  Restore Client:         "bacula-proxy-vm-rhel9-fd" 18.1.4 (02May25) x86_64-
↪redhat-linux-gnu-bacula-enterprise,redhat,(Blue
  Where:                  /
  Replace:                Always
  Start time:             09-May-2025 15:06:37
  End time:               09-May-2025 15:15:32
  Elapsed time:           8 mins 55 secs
  Files Expected:         6
  Files Restored:         6
  Bytes Restored:         1,073,778,385 (1.073 GB)
  Rate:                   2007.1 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Restore to a Local Directory

It is possible to restore a Nova server to a local directory without transferring the data to the OpenStack server. To do so, the where restore option must point to a directory on the proxy server where the OpenStack Plugin is installed:

This is an example on how to restore a Nova server to a local directory using bconsole:

```
*restore jobid=166 where=/opt/bacula/archive/bacula-restores client=bacula-
↪proxy-vm-rhel9-fd
You have selected the following JobId: 166

Building directory tree for JobId(s) 166 ...
6 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ mark *
6 files marked.
$ done
Bootstrap records written to /opt/bacula/working/bacula-dir.restore.21.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================
```

(continues on next page)

(continued from previous page)

```
    Vol-0001                  DiskAutochanger           DiskAutochanger

Volumes marked with "*" are in the Autochanger.


6 files selected to be restored.

Run Restore job
JobName:         Restore
Bootstrap:       /opt/bacula/working/bacula-dir.restore.21.bsr
Where:           /opt/bacula/archive/bacula-restores
Replace:         Always
FileSet:         BaculaConfigs
Backup Client:   bacula-proxy-vm-rhel9-fd
Restore Client:  bacula-proxy-vm-rhel9-fd
Storage:         DiskAutochanger
When:            2025-05-09 16:08:37
Catalog:         BaculaCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): yes
Job queued. JobId=209
```

If the path specified for the `where` option does not exist, it will be created by the Bacula OpenStack Plugin.

This is a successful restore joblog:

```
bacula-dir JobId 209: Start Restore Job Restore.2025-05-09_16.08.44_33
bacula-dir JobId 209: Restoring files from JobId(s) 166
bacula-dir JobId 209: Connected to Storage "DiskAutochanger" at 10.0.99.
↪131:9103 with TLS
bacula-dir JobId 209: Using Device "DiskAutochanger_Dev0" to read.
bacula-dir JobId 209: Connected to Client "bacula-proxy-vm-rhel9-fd" at 10.0.
↪100.35:9102 with TLS
bacula-proxy-vm-rhel9-fd JobId 209: Connected to Storage at 10.0.99.131:9103␣
↪with TLS
bacula-sd JobId 209: Ready to read from volume "Vol-0001" on File device
↪"DiskAutochanger_Dev0" (/opt/bacula/archive).
bacula-sd JobId 209: Forward spacing Volume "Vol-0001" to addr=2149713765
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Plugin log of this job␣
↪available in: "/opt/bacula/working/openstack-0.log"
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Trying to get OSClient for␣
↪endpoint=http://10.0.100.35/identity/v3 user=admin password=**********␣
↪domain=Default project=demo
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Restoring Nova server␣
↪configuration "257a54e7-73ac-4c10-bd01-a37f61f9a05d-cirros-instance1" at "/
↪opt/baculaarchive//bacula-restores/nova/server/cirros-instance1_257a54e7-
↪73ac-4c10-bd01-a37f61f9a05d/257a54e7-73ac-4c10-bd01-a37f61f9a05d.conf"
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Starting new server restore␣
↪for "257a54e7-73ac-4c10-bd01-a37f61f9a05d"
```

(continues on next page)

(continued from previous page)

```
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Nova flavor restore START
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Restoring Nova Flavor "c1-
↪cirros256" locally at "/opt/bacula/archive/bacula-restores/nova/flavor/
↪cirros256_c1"
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Nova flavor restore END
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Restoring server volume␣
↪configuration "485b7cec-2c92-4d77-a1f8-ea981d6192cc-" at "/opt/bacula/
↪archive/bacula-restores/nova/server/cirros-instance1_257a54e7-73ac-4c10-
↪bd01-a37f61f9a05d/485b7cec-2c92-4d77-a1f8-ea981d6192cc.conf"
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Restoring cinder volume
↪"485b7cec-2c92-4d77-a1f8-ea981d6192cc" from server "cirros-instance1␣
↪(257a54e7-73ac-4c10-bd01-a37f61f9a05d)"
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Starting new unmatch volume␣
↪restore for "485b7cec-2c92-4d77-a1f8-ea981d6192cc"
bacula-sd JobId 209: Elapsed time=00:05:09, Transfer rate=3.475 M Bytes/second
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Restoring volume "485b7cec-
↪2c92-4d77-a1f8-ea981d6192cc" locally at /opt/bacula/archive/bacula-restores/
↪nova/server/cirros-instance1_257a54e7-73ac-4c10-bd01-a37f61f9a05d/485b7cec-
↪2c92-4d77-a1f8-ea981d6192cc.bvmdk
bacula-proxy-vm-rhel9-fd JobId 209: openstack: Cinder volume restore SUCCESS
bacula-dir JobId 209: Bacula Enterprise bacula-dir 18.1.4 (02May25):
Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat (Blue
JobId:                  209
Job:                    Restore.2025-05-09_16.08.44_33
Restore Client:         "bacula-proxy-vm-rhel9-fd" 18.1.4 (02May25) x86_64-
↪redhat-linux-gnu-bacula-enterprise,redhat,(Blue
Where:                  /opt/bacula/bacula-restores
Replace:                Always
Start time:             09-May-2025 16:08:47
End time:               09-May-2025 16:14:33
Elapsed time:           5 mins 46 secs
Files Expected:         6
Files Restored:         6
Bytes Restored:         1,073,778,385 (1.073 GB)
Rate:                   3103.4 KB/s
FD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            Restore OK
```

The restore job log will report that the restore was completed to a local directory: `locally at /opt/bacula/archive/bacula-restores`.

These are the files restored in the `/opt/bacula/archive/bacula-restores` directory of the `bacula-proxy-vm-rhel9` Bacula proxy instance for the `cirros-instance1` instance:

```
# ls -lR bacula-restores/
bacula-restores/:
total 0
drwxr-x--x. 4 root bacula 34 May  9 14:43 nova

bacula-restores/nova:
```

(continues on next page)

(continued from previous page)

```
total 0
drwxr-x--x. 2 root bacula 26 May  9 14:43 flavor
drwxr-x--x. 3 root bacula 67 May  9 14:43 server

bacula-restores/nova/flavor:
total 4
-rw-r-----. 1 root bacula 275 May  9 14:43 cirros256_c1

bacula-restores/nova/server:
total 4
drwxr-x--x. 2 root bacula 4096 May  9 14:47 cirros-instance1_257a54e7-73ac-
↪4c10-bd01-a37f61f9a05d

bacula-restores/nova/server/cirros-instance1_257a54e7-73ac-4c10-bd01-
↪a37f61f9a05d:
total 1048616
-rw-r-----. 1 root bacula       1221 May  9 14:43 257a54e7-73ac-4c10-bd01-
↪a37f61f9a05d.conf
-rw-r-----. 1 root bacula       4609 May  9 14:47 485b7cec-2c92-4d77-a1f8-
↪ea981d6192cc.bmp
-rw-r-----. 1 root bacula 1073741824 May  9 14:47 485b7cec-2c92-4d77-a1f8-
↪ea981d6192cc.bvmdk
-rw-r-----. 1 root bacula        620 May  9 14:43 485b7cec-2c92-4d77-a1f8-
↪ea981d6192cc.conf
-rw-r-----. 1 root bacula      21314 May  9 14:47 485b7cec-2c92-4d77-a1f8-
↪ea981d6192cc.sha
```

bconsole Query Commands

The Bacula Enterprise OpenStack Plugin supports the bconsole query command.

Bacula queries the OpenStack API and receives answers in the form of `key=value` pairs. The OpenStack Plugin supports two query parameters.

parameter=connection

When using `parameter=connection`, the query command sends a request across various OpenStack services to check whether the current plugin parameters allow the connection to each one of the service endpoints. The example below shows the execution of a correctly formatted `CONNECTION` query and response:

```
*.query plugin="openstack: OS_AUTH_URL=http://localhost:5000/v3 OS_
↪USERNAME=admin OS_PASSWORD=123456 OS_USER_DOMAIN_NAME=Default OS_PROJECT_
↪NAME=admin" client=bacula_proxy-fd parameter=connection
connection=ok
keystone=ok
cinder=ok
glance=ok
neutron=ok
nova=ok
```

When the query parameter is `connection`, each service will return a value that is either `ok` or `not ok`, indicating whether the connection was successful or not. An additional tuple `connection=` is reported at the end of the query result to indicate if all the other connections are successful.

parameter=server

When the query parameter is `server`, the query command sends a request to the OpenStack API to list all available Nova servers and their respective UUID.

This query displays two `key=value` tuples per server, where:

- the first `key` is the `server` keyword and the `value` is the name of the Nova server.
- the second `key` is the `uuid` keyword and the `value` is the Nova server UUID.

The example below shows the execution of a correctly formatted `server` query that finds six different guest servers.

This query also check if the `proxy_server` value is correctly set.

```
*.query plugin="openstack: proxy_server=bacula_proxy" client=bacula_proxy-fd␣
↪parameter=server
server=nfs-1
uuid=b12e3313-7c3e-4757-8a12-5a2c9884ad53
server=nfs-i
uuid=7629a85d-3bec-4734-8d35-586e8ba1253f
server=nfs-2
uuid=729ae7cb-0103-4ddd-8f23-5dfab6a00c0c
server=www-1
uuid=248dd682-f0e9-43c1-8c05-999115459e1a
server=www-2
uuid=727e0dbf-9d68-4f83-97b8-d3540133b4fc
server=bacula-proxy
uuid=3b8496f0-740c-41d1-8a08-8c1a50cae461
```

Note: The above query command is using authentication from `/opt/bacula/etc/openstack.conf`.

parameter=proxy_server

When using `proxy_server`, the query command sends a request to the OpenStack API to list all eligible candidates to the role of proxy server. This query displays two `key=value` tuples per candidate where:

- the first tuple `key` is `proxy_server` and the `value` is the name of the Nova server.
- the second tuple `key` is `uuid` and the `value` is the Nova server UUID.

The example below shows the execution of a correctly formatted `parameter=server` query that finds three different Nova servers candidates to the role of proxy server.

```
*.query plugin="openstack: endpoint=http://11.11.11.111:2222/v3 OS_
↪USERNAME=admin OS_PASSWORD=123456 OS_USER_DOMAIN_NAME=Default OS_PROJECT_
↪NAME=admin" client=bacula_proxy-fd parameter=server
proxy_server=nfs-instance1
```

(continues on next page)

(continued from previous page)

```
uuid=b12e3313-7c3e-4757-8a12-5a2c9884ad53
proxy_server=www-instance1
uuid=248dd682-f0e9-43c1-8c05-999115459e1a
proxy_server=bacula-proxy
uuid=3b8496f0-740c-41d1-8a08-8c1a50cae461
```

parameter=check

When using the `check` option, the query command will check if the base configuration is correct.

It will start by testing if all parameters connection are set.

If all parameters are set, it will then check if the connection to OpenStack services is valid.

Then, it will check if the proxy server is set and available.

If the answers to all those questions are positive, the query will output a single `check:ok` tuple.

The example below shows the execution of a correctly formatted `check` query.

```
*.query Plugin="openstack: proxy_server=bacula-proxy" client=bacula_proxy-fd␣
↪parameter=check
check=ok
```

Note: The above query command is using authentication from `/opt/bacula/etc/openstack.conf`.

Limitations

This article presents the current limitations of the OpenStack Plugin.

- For the plugin to function properly, the proxy server must be a Nova server in the OpenStack environment running a Linux distribution.
- Virtual Full jobs are not supported.
- The `restart` command has certain limitations with plugins, as it initiates the Job from the beginning instead of resuming it. Bacula determines whether a Job is to be restarted or continued, however, using the `restart` command will result in a new Job when using the OpenStack Plugin.

QEMU Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Features Overview

- Transaction based online backup of any running QEMU VM guests
- Full and Incremental virtual disk image-level backup
- VM guest configuration for easy recovery
- Ability to restore virtual disk images as QCOW2 files
- Ability to completely restore a Proxmox VM guest, including configuration
- Ability to completely restore a target disk images for Genuine QEMU
- Ability to restore VM virtual disk images to an alternate directory

Guest VM Backup Strategies

Installing a Bacula Client on Each Guest VM

Using this strategy, you do not use the Bacula Enterprise QEMU Plugin, but instead install a Bacula Enterprise File Daemon on every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on your QEMU system, you will use Bacula's Schedules, Priorities, and Job concurrency settings to spread backup jobs throughout the backup window. Since all guest VMs could possibly reside on the same storage on the Proxmox hypervisor, running all your backup jobs at the same time can create a bottleneck on the disk/network subsystem.

Installing a Bacula Enterprise File Daemon in each virtual machine allows you to manage your virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files
- Checksum of individual files for Virus and Spyware detection
- Verify Jobs
- File or Directory exclusion (such as swap or temporary files)
- File level compression
- Accurate backups
- Plugins

Image Backup with QEMU Plugin

With the image level backup strategy, the Bacula Enterprise QEMU Plugin will save VM disks as QCOW2 images for QEMU VMs for both Full and Incremental backups.

For this to work, a Bacula File Daemon is not needed in each guest VM. The Bacula QEMU Plugin will contact the QEMU hypervisor to read and save the contents of virtual machine disks using the QMP transaction backups feature and dump them using the QMP API.

Bacula does not need to walk through the Client filesystems to stast, open, read, and close files, so it consumes less resources on the QEMU infrastructure than a file level backup on each VM would. On the other hand, Bacula will also read and save useless data in the VMs such as swap files or temporary files.

Tip: The QEMU Plugin will save not only the disk images of the guest VMs, but also guest VM configurations which allows for very easy guest VM restores. This feature is available for Proxmox infrastructure only.

Backup and Restore Operations

Backup

The QEMU Plugin supports the QMP control protocol which is a low-level managing interface for every QEMU Hypervisor flavor. In most cases this interface is used by Hypervisor management tools to manage QEMU VM Guests.

The current Plugin version supports two QEMU Hypervisor types:

- Genuine QEMU - `mode=QEMU` (*the default*)
- Proxmox (PVE) - `mode=PVE`

The core backup and restore procedure is the same for every QEMU hypervisor type. The main difference is the way a VM is selected to be backed up and how the restore process is finished.

The core backup of a single guest VM consists of the following steps:

1. Query VM information and save it
2. For a Full backup level, during the backup transaction, create or clear Block dirty bitmap and dump VM disk image to the working area (check *Working Directory Location*)
3. For an Incremental backup level, during the backup transaction, dump the incremental VM disk image to the working area and clear the dirty bitmap
4. When a VM disk image dump is ready, the Bacula File Daemon will send the data to the Storage Daemon

Backups can be performed for guest VMs in a `running` state only. Any first Full level backup will create a backup chain used for subsequent Incremental backups. The current plugin version creates and maintains a single backup chain for each VM.

Warning: You should never mix different backup jobs for a single QEMU VM as it will break the backup chain and backups become unrecoverable!

The QEMU Plugin will log the start and end of each guest VM backup:

```
JobId 104: Start Backup JobId 104, Job=proxmox.2021-10-28_12.04.54_42
JobId 104: Recycled volume "vol01"
JobId 104: Using Device "FileChgr1-Dev2" to write.
JobId 104: qemu: Connected to Proxmox pve-manager/6.4-13/9f411e79 (running␣
↪kernel: 5.4.143-1-pve)
JobId 104: qemu: Start Backup vm: vm1 (100)
JobId 104: qemu: Finish backup of vm1/drive-scsi0
...
```

The backup will create a single (`.qcow2`) file for every VM disk device saved and a single VM configuration file (`config.json`) during a Full backup.

- `/@qemu/<vm-name>/<vmid>/<drive-name>.qcow2` for every VM disk image
- `/@qemu/<vm-name>/<vmid>/config.json` for VM configuration data

Tip: For *Genuine QEMU* mode <*vmid*> will generally always be set to zero and <*vm-name*> is the value set with the `-name <vm-name>` QEMU execution parameter.

Multiple files will be created during a backup if multiple guest VMs are backed up with one job (for Proxmox backup mode only). The distinct file names as shown above will help to locate the proper guest VM images for restore.

Restore

The QEMU Plugin allows the following targets for restore operations:

- Restore to a local directory as QCOW2 disk image files and single configuration file
- Restore to Genuine QEMU hypervisor to the original device file location
- Restore to Proxmox virtualization infrastructure as an original or new VMID

Restore To Local Directory

To use this mode, the **where=/some/path** parameter of a Bacula restore is set to a full path on the server where the QEMU Plugin is installed. If the path does not exist, it will be created by the Bacula QEMU Plugin. Bacula will automatically patch destination disk images with all Incremental backups. The end results will be a “ready to use” QCOW2 disk image file. For this target type you will always get disk images in QCOW2 format.

Restore to QEMU

To use this restore mode, the **where=/** parameter of a Bacula restore is used. The guest VM disk images will be restored and automatically patched with all Incremental backups in the *working directory* (check *Working Directory Location* for details). Then all patched and ready to use image files will be converted to the original destination location and image format when the **convdestination: yes** restore plugin parameter is set. This allow you to get ready to run QEMU guest VM images in their original location.

This restore mode is selected for all backups executed for *Genuine QEMU*.

Restore to Proxmox

To use this restore mode, the **where=/** parameter of a Bacula restore is used. The guest VM disk images will be restored and automatically patched with all Incremental backups in the *working directory* (check

Working Directory Location for details). Then all patched and ready to use image files will be transferred to Proxmox storage attached to the original image.

The QEMU Plugin will try to create the same :term::*VMID* as it was during the backup. If VM guest already exist then with this :term::*VMID*, the plugin will allocate a new VMID. It will never overwrite an existing VM guest.

All other guest VM configuration parameters will be restored as they were backed up, including network MAC addresses. For this reason, it is recommended to inspect and possibly update a guest VM's configuration before it is started. Otherwise, resource conflicts may arise.

The Storage target to be used for the restored guest VM disk(s) can be set using the plugin's **storage** restore option. If this option is not set, all guest VM disks will be restored to their original Storage.

To list available Storages, a listing mode is available, described in the chapter **listing**)

This restore mode is selected for all backups executed for *Proxmox*.

Attention: The QEMU Plugin cannot be used for automatic migration of QEMU guest VM from *Genuine QEMU* into a Proxmox virtualization infrastructure, but you can perform a *Proxmox* to *Genuine QEMU* migration using backup `mode=QEMU` for both.

Installation

The Bacula File Daemon and its QEMU Plugin need to be installed on the host of the Proxmox hypervisor which runs the guest VMs that are to be backed up. Proxmox uses a customized Debian distribution, so the Bacula Enterprise File Daemon for that platform has to be used.

Configuration

The **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `qemu-fd.so` plugin file is installed. The standard Bacula plugin directory is `/opt/bacula/plugins`

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
...
}
```

Installation of the Plugin

Installation of the Bacula Enterprise QEMU Plugin is most easily done by adding the repository file suitable for the existing subscription to the Linux package manager for your distribution of choice.

For RHEL distributions an example repository file would be `/etc/yum.repos.d/bacula-qemu.repo` with the following content:

```
[bacula-enterprise-qemu]
name=Bacula Enterprise QEMU Plugin for RHEL $releasever - $basearch
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/qemu/
```

(continues on next page)

(continued from previous page)

```
↪@version@/rhel8-64
enabled=1
gpgcheck=1
gpgkey=https://www.baculasystems.com/dl/@customer-string@/BaculaSystems-
↪Public-Signature.asc
```

For Ubuntu/Debian distributions an example repository file would be `/etc/apt/sources.list.d/bacula-qemu.list` with the following content:

```
# Bacula Enterprise QEMU Plugin
deb https://www.baculasystems.com/dl/@customer-string@/debs/qemu/@version@/
↪bullseye-64/ bullseye qemu
```

After that, a run of `apt-get update` for any Ubuntu/Debian distribution is needed. Then, the Plugin may be installed using `apt-get install bacula-enterprise-qemu-plugin` or `yum install bacula-enterprise-qemu-plugin` for RHEL.

Plugin Configuration

The plugin is configured using **Plugin Parameters** defined in a Fileset's **Include** section of the Bacula Enterprise Director configuration.

Generic Plugin Parameters

The following QEMU Plugin parameters effect any type of Job (Backup, Estimation, or Restore).

abort_on_error[=<0|1>]
specifies whether or not the plugin should abort execution (and fail the Bacula Job) if a fatal error occurs during a Backup, Estimation, or Restore operation. This parameter is optional. The default value is 0.

working=</path/to/dir>
specifies the directory used for any QEMU Plugin backup or restore operations. The default value is the *WorkingDir* Bacula FD configuration parameter (check *Working Directory Location*). This parameter is optional.

Genuine QEMU Estimation and Backup Plugin Parameters

These plugin parameters are relevant only for Backup and Estimation jobs:

qmpcontrol=</path/to/qmp/socket>
specifies the location of the QMP control socket used for backup operations. When this parameter is set it implies `mode=QEMU`. This parameter is required for *Genuine QEMU* and is automatically set up for the other virtualization platforms. For other virtualization platforms this parameter is prohibited.

mode=[QEMU|PVE]
specifies the default type of QEMU support. You should set it to `QEMU` for Genuine QEMU virtualization, and `PVE` for Proxmox virtualization. When not set, *QEMU* mode of operation is set which requires a proper *qmpcontrol=…* parameter to be configured.

timeout=<seconds>
specifies a time the QEMU Plugin will wait for an image dump to finish. When this timeout is reached, the job will be failed. If not set, a default value of 3600 seconds will be used. This parameter is optional.

bitmap-persistence=[0|1]
specifies if QEMU should persist a dirty block map used for Incremental backups in the device itself. The bitmap persistance is supported on the QCOW2 image files only. Setting this option on devices which do not support it will cause the backup job to fail. The bitmap persistance is set during the first Full backup job only. For Proxmox this parameter is set to '0'. This parameter is optional.

Important: Ephemeral dirty block bitmap (when `bitmap-persistence=0`) will be recreated on every VM restart forcing a Full VM backup.

---

vm=<name>
specifies a guest VM name to backup. All guest VMs with a <*name*> provided will be selected for backup. Multiple `vm=...` parameters are allowed. If a guest VM with <*name*> can not be found, then a single job error will be generated and the backup will proceed to the next VM unless `abort_on_error` is set which will cause the backup job to be failed. This parameter is optional and used when `mode=PVE` only.

vmid=<vmid>
specifies a guest VM VMID to backup. Multiple `vmid=...` parameters may be provided. If a guest VM with <*vmid*> can not be found, a job error will be generated and the backup will proceed to the next VM unless `abort_on_error` is set which will cause the backup job to be failed. This parameter is optional and used when `mode=PVE` only.

include=<name-regex>
specifies a list of a guest VM names to backup using regular expression syntax. All guest VMs with names matching the `name-regex` regular expression provided will be selected for backup. Multiple `include=...` parameters may be provided. The match performed is case insensitive.

If no guest VMs match the name-regex expression provided, the backup will proceed to the next parameters or finish successfully without backing up any VMs. The `abort_on_error` parameter will not fail the job when no guest VMs are found using name matching. This parameter is optional and used when `mode=PVE` only.

exclude=<name-regex>
specifies a list of a guest VMs names which will be excluded from backup using regular expression matching. All guest VMs with names matching the provided regular expression, and selected for backup using the `include=...` parameter will be excluded. The match is case insensitive. This parameter does not affect any guest VM selected to be backed up using `vm=...` or `vmid=...` parameters. Multiple `exclude=...` parameters may be provided. This parameter is optional and used when `mode=PVE` only.

If none of the paramaters `vm=...`, `vmid=...`, `include=...` and `exclude=...` are specified then all available guest VMs on the Proxmox hypervisor will be backed up. For *Genuine QEMU* a single VM pointed to by the `qmpcontrol=...` parameter will be backed up.

Plugin Restore Parameters

During restore, the QEMU Plugin will use the same parameters which were set for the backup job. These settings are saved in the catalog at the time the backup job is run. Some of them may be changed during the restore process if required.

convdestination: [yes|no]
specifies if a restored disk image should be converted to the original format and location as it was defined during backup. This parameter works in *Genuine QEMU* mode only. If not set or set to *no* (the default), then QEMU virtual disk images will be available in the *working* (see next parameter) location as QCOW2 format files. In this case you can manually move it to the required destination or convert it to the desired format.

This parameter is optional.

working: </path/to/dir>
specifies the directory used for QEMU Plugin restore operations as described at *Working Directory Location*. The default value is the same as for backup and can be set by the plugin configuration or the *WorkingDir* Bacula FD configuration parameter.

This parameter is optional.

storage: <storage>
specifies a Proxmox Storage where restored guest VMs will be restored to. If not set then a guest VM will be restored to the Proxmox Storage it was backed up from. If this parameter points to a nonexistent Storage, the original Storage of the guest VM will be used.

This parameter is optional.

Working Directory Location

To perform any of the backup or restore job operations, the QEMU Plugin requires some storage space to be available. The size space required depends on the size of the largest disk image file during backup and the sum of the VM disk images during restore. For the restore operations the space is required for proper incremental image patching.

This limitation is a direct consequence of the QEMU QMP drive backup limitation which supports disk images saved to a regular file only.

The exact location of the working area can be selected with the `working=...` plugin parameter. When this parameter is not set then a default value of *WorkingDirectory* Bacula parameter will be used.

This storage space is automatically cleaned after the operation unless you select a restore operation without disk image conversion to the destination (check **convdestination** restore parameter).

Fileset Examples

In the example below, a single QEMU VM qill be backed up.

```
Fileset {
  Name = FS_qemu
  Include {
    Plugin = "qemu: qmpcontrol=/images/vm1.qmp"
  }
}
```

Now the same QEMU VM but with custom working location.

```
Fileset {
  Name = FS_qemu_working
  Include {
    Plugin = "qemu: qmpcontrol=/images/vm1.qmp working=/tmp"
  }
}
```

In the example below, all Promox guest VMs will be backed up.

```
Fileset {
  Name = FS_ProxmoxAll
  Include {
    Plugin = "qemu: mode=PVE"
  }
}
```

In this example, a single guest VM with name of "VM1" will be backed up.

```
Fileset {
  Name = FS_Proxmox_VM1
  Include {
    Plugin = "qemu: mode=PVE vm=VM1"
  }
}
```

The same example as above, but using `vmid` instead:

```
Fileset {
  Name = FS_Proxmox_VM1
  Include {
    Plugin = "qemu: mode=PVE vmid=101"
  }
}
```

In the following example, all guest VMs which contain "Prod" in their names will be backed up.

```
Fileset {
  Name = FS_Proxmox_ProdAll
  Include {
    Plugin = "qemu: mode=PVE include=Prod"
  }
}
```

In this final example, all guest VMs except VMs whose name begins with "Test" will be backed up.

```
Fileset {
  Name = FS_Proxmox_AllbutTest
  Include {
    Plugin = "qemu: mode=PVE include=.* exclude=^Test"
  }
}
```

Restore

Restore with Genuine QEMU

To restore a VM or VMs using *Genuine QEMU* mode, you should execute the restore command and specify the **where** parameter as in this example:

```
* restore where=/
```

and then set any other required restore plugin parameters for the restore.

In the following restore session example, the **convdestination** plugin restore option is set to *"yes"*:

```
* restore where=/
...
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/qemu-test-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   qemu-test-fd
Restore Client:  qemu-test-fd
Storage:         File1
When:            2021-11-01 13:19:16
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : qemu: qmpcontrol=/images/vm1.qmp abort_on_error
Plugin Restore Options
Option               Current Value        Default Value
working:             *None*               (*None*)
convdestination:     *None*               (*None*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
   1: working (Plugin working directory)
   2: convdestination (Convert qcow2 restore to original destination)
```

(continues on next page)

(continued from previous page)

```
Select parameter to modify (1-2): 2
Please enter a value for convdestination: yes
Plugin Restore Options
Option                 Current Value        Default Value
working:               *None*               (*None*)
convdestination:       yes                  (*None*)
Use above plugin configuration? (yes/mod/no): yes
...
```

During a restore job you should get information about each restore process including Incremental patching of the restored disk devices and the destination convertion process.

```
JobId 121: Start Restore Job RestoreFiles.2021-11-01_13.05.11_03
JobId 121: Restoring files from JobId(s) 115,116,120
JobId 121: Using Device "FileChgr1-Dev1" to read.
JobId 121: qemu: VM to restore: VM1 (oid:0)
JobId 121: qemu: Start Restore vm: VM1 (rid:0) devices: 2
JobId 121: qemu: Restoring device: ide0-hd0
JobId 121: qemu: Restoring device: ide0-hd1
JobId 121: qemu: Patching incremental: ide0-hd0
JobId 121: qemu: Patching incremental: ide0-hd1
JobId 121: qemu: Patching incremental: ide0-hd0
JobId 121: qemu: Patching incremental: ide0-hd1
```

The new guest VM created during the restore will get a new VMID (if the original VMID is in use) but the name / hostname will stay the same as it was with the original VM.

Restore to a Proxmox Hypervisor

To restore a VM or VMs to a Proxmox hypervisor, you should execute the same restore command as above and the basic restore process is the same. The main difference is the final configuration and device importing procedure as shown below:

```
JobId 117: Start Restore Job RestoreFiles.2021-11-02_17.26.25_07
JobId 117: Restoring files from JobId(s) 104,105,106,115,116
JobId 117: Using Device "FileChgr1-Dev2" to read.
JobId 117: qemu: VM to restore: vm1 (oid:100)
JobId 117: qemu: Start Restore vm: vm1 (rid:104) devices: 1
JobId 117: qemu: Restoring device: drive-scsi0
JobId 117: qemu: Patching incremental: drive-scsi0
JobId 117: qemu: Patching incremental: drive-scsi0
JobId 117: qemu: Patching incremental: drive-scsi0
JobId 117: qemu: Patching incremental: drive-scsi0
JobId 117: Elapsed time=00:06:42, Transfer rate=20.28 M Bytes/second
JobId 117: qemu: Successfully imported device drive-scsi0 as local-lvm:vm-104-
↪disk-0
...
```

The new guest VM created during the restore will get a new VMID (if the original VMID is in use) but the name / hostname will stay the same as it was with the original VM.

Restore to Local Directory

It is possible to restore the guest VM disk image(s) to a local directory instead of restoring back to a hypervisor as a new VM. To do so, the **where** restore option should point to a local directory:

```
* restore where=/tmp/bacula/restores
```

Please check the following example for the test "VM local restore":

```
JobId 118: Start Restore Job RestoreFiles.2021-11-02_17.37.34_09
JobId 118: Restoring files from JobId(s) 104,105,106,115,116
JobId 118: Using Device "FileChgr1-Dev1" to read.
JobId 118: Ready to read from volume "vol01" on File device "FileChgr1-Dev1"␣
↪(/opt/bacula/archive).
JobId 118: qemu: VM local restore: vm1 (oid:100)
JobId 118: Forward spacing Volume "vol01" to addr=227
JobId 118: qemu: Restoring device: drive-scsi0
JobId 118: qemu: Patching incremental: drive-scsi0
JobId 118: qemu: Patching incremental: drive-scsi0
JobId 118: qemu: Patching incremental: drive-scsi0
JobId 118: qemu: Patching incremental: drive-scsi0
JobId 118: Elapsed time=00:06:57, Transfer rate=19.55 M Bytes/second
...
```

The restore job log will show that the restore was done to a local directory.

Other

Resource listing

The Bacula Enterprise QEMU Plugin supports the plugin listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available Proxmox resources such as:

- List of guest VM names
- List of guest VM VMIDs
- List of Proxmox Storages

This feature uses the special **.ls** "dot command" with a **plugin=<plugin>** parameter. The command requires the following parameters to be set:

client=<client>
A Bacula Client name with the QEMU Plugin installed.

plugin=<plugin>
A plugin name, which would be **qemu:** in this case, with optional plugin parameters as described in section **genericparameters**.

path=<path>
An object path to display.

The supported values for a **path=<path>** parameter are:

/
to display object types available to list.

vm
  to display a list of guest VM name-labels.

vmid
  to display a list of guest VM VMIDs and name-label pointers.

storage
  to show the list of available Storages.

To display available object types, follow the following command example:

```
*.ls client=proxmoxtest-fd plugin="qemu: mode=PVE" path=/
Connecting to Client proxmoxtest-fd at proxmoxtest:9102
drwxr-x---   1 root     root                      0 2021-11-01 12:55:55  vm
drwxr-x---   1 root     root                      0 2021-11-01 12:55:55  vmid
drwxr-x---   1 root     root                      0 2021-11-01 12:55:55 ␣
↪storage
2000 OK estimate files=3 bytes=0
```

To display the list of all available guest VMs, the following command example can be used:

```
*.ls client=proxmoxtest-fd plugin="qemu: mode=PVE" path=/vm
Connecting to Client proxmoxtest-fd at proxmoxtest:9102
-rw-r-----   1 root     root             8589934592 2021-11-01 12:56:18  vm1
-rw-r-----   1 root     root             8589934592 2021-11-01 12:56:18  vm3
2000 OK estimate files=2 bytes=17,179,869,184
```

To display the list of guest VM VMIDs, use the following command example:

```
*.ls client=proxmoxtest-fd plugin="qemu: mode=PVE" path=/vmid
Connecting to Client proxmoxtest-fd at proxmoxtest:9102
-rw-r-----   1 root     root             8589934592 2021-11-01 12:56:46  100 -
↪> vm1
-rw-r-----   1 root     root             8589934592 2021-11-01 12:56:46  102 -
↪> vm3
2000 OK estimate files=2 bytes=17,179,869,184
```

The VM and VMID lists display an estimated size of the guest VM based on *virtual* or *actual* disk image sizes.

To display available Proxmox Storages, the following command example can be used:

```
*.ls client=proxmoxtest-fd plugin="qemu: mode=PVE" path=/storage
Connecting to Client proxmoxtest-fd at proxmoxtest:9102
brw-r-----   1 root     root                      0 2021-11-01 12:57:56  data
brw-r-----   1 root     root                      0 2021-11-01 12:57:56 ␣
↪local-lvm
2000 OK estimate files=2 bytes=0
```

Resource Query

The Bacula Enterprise QEMU Plugin supports the plugin query feature of Bacula Enterprise 14.0 or newer. This mode allows a plugin to display the same information about available Proxmox resources which is defined at *Resource listing*. The QEMU Plugin returns response data in JSON format.

In this example is a query about available VMs:

```
*.query client=proxmoxtest-fd plugin="qemu: mode=PVE" parameter=vm
[{"name":"vm1","vmid":100},{"name":"vm3","vmid":102}]
*.query client=proxmoxtest-fd plugin="qemu: mode=PVE" parameter=vmid
[{"name":"vm3","vmid":102},{"name":"vm1","vmid":100}]
```

In this example is a query about available Proxmox storages:

```
*.query client=proxmoxtest-fd plugin="qemu: mode=PVE" parameter=storage
[{"name":"local"},{"name":"local-lvm"}]
```

Limitations

- Granular restore (*Single Item Restore*) is not available yet. A file level backup of the VM with the Bacula FD inside the Guest VM is needed to enable single file restores of a QEMU VM guest.
- It is not possible to run the very same backup job of the same guest VM concurrently (duplicate job).
- It is not possible to have different backup jobs that backup the same guest VM as it will break the Incremental dirty bitmap backup chain. In this case, a successful recovery won't be even possible.
- VM templates available on the Proxmox system can not be backed up. This is a Proxmox limitation.
- In listing and query mode with the **vm** or **vmid** parameters, the plugin will display running VMs only.
- Any VM backup or restore operation requires sufficient storage space in the *working directory* (check *Working Directory Location*). This is a general QEMU QMP backup procedure limitation.
- Differential backup level is not yet supported. Only Full and Incremental backup levels are supported. This limitation will be removed in the future.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Supported QEMU Hypervisor versions

The following QEMU Hypervisor infrastructure versions are tested and supported:

- Genuine QEMU 3.1 - 5.2
- Proxmox VE 6.4 - 8.2

Nutanix-Acropolis HyperVisor (Nutanix-AHV) Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This document aims at presenting the reader with information about the **Bacula Enterprise Nutanix-AHV Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features.

Features

Bacula Enterprise is an especially secure and reliable backup and recovery software that is compatible with more databases and hypervisor types than almost any other solution available today. One example is the way it seamlessly integrates with Nutanix AHV to offer an especially powerful backup and recovery solution - even for extremely demanding environments. Nutanix AHV itself is a powerful, Linux KVM based hypervisor, designed in order to make managing computer infrastructure easier. Bacula's Nutanix module is designed to simplify and make efficient the backup and restore procedure of Nutanix VM'S.

- Snapshot-based online backup of any guest VM.
- Full, Incremental and Differential block level image backup.
- Backups are consistent at image, disk and host configuration level.
- Ability to restore complete virtual machine image.
- Ability to recover and configure virtual machine network at restore time.
- Ability to restore the data also in raw format over a different filesystem.
- Precise inclusion/exclusion mechanism to control the backup target.
- Automatic backup configuration through hypervisor VMs scan routines.
- Automatic snapshot cleanup processes.
- Compatibility with *Deduplication* techniques.
- Send backup data to local disk, network disk, block storage, tape or cloud.

Backup and Restore Strategies

This article presents information regarding backup and restore strategies of the Nutanix-AHV Plugin.

Installing Bacula Client on Each Guest

This strategy works by installing a Bacula Enterprise File Daemon on every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on the Nutanix-AHV hypervisor, the user will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to spread backup jobs over the backup window. Since all VMs could use the same storage on the Nutanix-AHV hypervisor, running all backup jobs at the same time could create a bottleneck on the disk/network subsystem since Bacula will walk through all filesystems to open/read/close/stat files.

Installing a Bacula Enterprise File Daemon on each virtual machine permits to manage virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files

- Checksum of individual files for Virus and Spyware detection
- Verify Jobs
- File/Directory exclusion (such as swap or temporary files)
- File level compression
- Accurate backups.

Image Backup With Nutanix-AHV Plugin

With the image backup level strategy, the Bacula Enterprise Nutanix-AHV Plugin will save the Client disks at the raw level, in the Nutanix-AHV context.

Bacula's Nutanix-AHV Plugin will query the guest VM through the hypervisor API to read and save the content of virtual machines disks using snapshots and the native Nutanix Filesystem (NDFS). During backups, Nutanix-AHV Plugin will save the integrity of disks images and also guest VM configurations to allow guest VM restores with their original parameters.

All those operations are handled by an additional proxy VM which is described in the next section.

Proxy Virtual Machine

To handle backup and restore operations in the Nutanix-AHV environment a proxy VM needs to be set up. This specific virtual machine handles most of the operations (snapshot management, IO, ,. . .) during backup and restore. These operations will be discussed in more detail in the next sections.

The proxy VM must have the following characteristics:

- Linux based operating system
- Bacula File Daemon installed (bacula-enterprise-client package) and running, configured as a Client Resource on the Bacula Director
- Nutanix-AHV Plugin (bacula-enterprise-nutanix-ahv-plugin package) installed
- Network access to the Nutanix-AHV REST API.

While there is theoretically no restriction for the type of Linux OS and the number of cores, during development and testing an Ubuntu LTS VM with four cores was used as a proxy VM.

Ingestion

The ingestion of a single virtual disk is a specific protocol which takes the following steps:

- From guest VM snapshot find NDFS paths of all modified disks.
- Use said NDFS path to list changed regions on the disk either from its base state or from a previous snapshot.
- Use said NDFS path again to hotplug a copy of the disk onto the proxy VM.
- Export the list of changed regions and raw disk image data to a Bacula Storage Daemon
- Unplug relevant disk from proxy VM

Backups can be performed for a guest VM in any power state (running or halted). For proper execution of Incremental or Differential backups it is required by the Nutanix-AHV plugin to store previous snapshots in order to compute changed block information.

Snapshots created by the Nutanix-AHV plugin are identified by JobID and the UUID of the guest VM. The plugin keeps track of dependencies between jobs and their respective snapshots to automatically delete snapshots that are no more relevant.

The backup will create the following backup files for each guest VM:

- A single empty file to match a guest VM name to its UUID /@nutanix-ahv/<vmUuid>_<vmName>.name
- A list of data regions for each virtual disk: /@nutanix-ahv/<vmUuid>/<diskUuid>.bmp
- A list of zeroed regions for each virtual disk: /@nutanix-ahv/<vmUuid>/<diskUuid>.abmp
- A raw data file for each virtual disk: /@nutanix-ahv/<vmUuid>/<diskUuid>.bvmdk

---

Note: Configuration files are not showed in this view but are still handled by the plugin.

At restore time the user can identify the guest VM using the UUID to mark the corresponding files:

```
+--------------------------------------------------------------------------------
↪-------------------+
| filename                                                                      ␣
↪                   |
+--------------------------------------------------------------------------------
↪-------------------+
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7_VmName.name                ␣
↪                   |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪46a8c3412bec.bmp   |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪46a8c3412bec.abmp  |
| /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/eef9d485-dba3-4e3f-b828-
↪46a8c3412bec.bvmdk |
+--------------------------------------------------------------------------------
↪-------------------+
```

Network Restore

The Nutanix-AHV plugin restores network interfaces. The user can influence the process using the two restore plugin parameters `disconnect_network` and `network_address`.

When attaching a new network interface to a guest VM the API alters the creation based on three factors:

- Is the NIC connected to the network?
- Does the NIC requests an ip address?
- If the NIC requests an ip address does it requires a specific one?

By default the Nutanix-AHV plugin does the following:

- It uses the `disconnect_network` argument to know whether to disconnect the NIC or not. The default value is *false*

- If the original guest VM had an ip address, the restored guest VM will request one
- If specified with the `network_address` plugin parameter, it asks for the relevant static address

If this default execution fails, the Nutanix-AHV plugin will try again with a disconnected NIC.

Note: The Nutanix-AHV plugin allows the request of static IP addresses only on a managed network.

Installation

To use the Nutanix-AHV Plugin, a Bacula File Daemon has to be installed on a VM running in the Nutanix-AHV environment. This VM is referred to as a proxy VM throughout this document. The proxy VM needs to have network access to the Nutanix API. The default network adapter should suffice.

Since all backup interactions are network based, any Bacula Enterprise File Daemon with access to the necessary Nutanix endpoints can be used to run the plugin.

Nutanix-AHV Installation with BIM

In order to install the Nutanix-AHV Plugin with BIM, install the File Daemon with BIM and choose to install the Nutanix-AHV Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

Nutanix-AHV Installation with Package Manager

Prerequisites

The `Plugin Directory` directive of the `File Daemon` resource in */opt/bacula/etc/bacula-fd.conf* should point to the location where the `nutanix-ahv-fd.so` plugin is installed. The default directory is: */opt/bacula/plugins*

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Installation Steps

An example for a Debian based Linux package manager would be a configuration file `/etc/apt/sources.list.d/bacula.list` with the following content.

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@cust@/debs/bin/@version@/bionic-64/␣
↪bionic main
deb https://www.baculasystems.com/dl/@cust@/debs/nutanix/@version@/bionic-64/␣
↪bionic nutanix
```

Use `apt-get update` to update the package cache. After that the Plugin can be installed using `apt-get install bacula-enterprise-nutanix-ahv-plugin`

Configuration

The following article presents the configuration of the Nutanix-AHV Plugin.

The plugin is configured using `Plugin Parameters` defined in the "Include" section of a Fileset resource (Bacula Director configuration).

General Parameters

The following Nutanix-AHV Plugin parameters impact any type of Job (Backup, Restore, Query).

Table 13: Nutanix AHV General Parameters

Parameter	Values	Default	Description
abort	`[=<0 or 1>]`	`0`	Specifies whether the plugin should abort on fatal errors during backup or restore. This parameter is optional.
user	`<Str`		Specifies the username to access the Nutanix-AHV API.
password	`<Str`		Password for accessing the Nutanix-AHV API. Can be stored in a passfile as `host:user = password` tuple. An example of such a tuple would be `10.0.1.1:admin = password`. The passfile can be either at `/opt/bacula/snapmgr.conf` or at `/opt/bacula/etc/snapmgr.conf`. A `snapmgr.conf` file may have more than one tuple in it in the case that a Bacula FD is backing up more than one Nutanix environment. This parameter is optional.
host	`<Str`		Specifies the location of the Nutanix-AHV API.
port	`<Num`	`9440`	Port to access the Nutanix-AHV API. This parameter is optional.
prox	`<Str`		Name of the proxy VM used to run the Bacula File Daemon with the Nutanix-AHV Plugin.
working_	`<Str`	`/opt/bacul worki`	Specifies the working directory location. This parameter is optional.
debug	`[0 - 9]`	`0`	Specifies the debug level. `0` is no debug, `9` is the highest level. Warnings and errors are always sent to the joblog and if any debug level is set those messages are sent to the debug file as well. For the Nutanix-AHV Plugin `1` displays debug level message, `2` displays trace level message. Any value higher than `2` displays additional information about external libraries that handle those values on their own. This parameter is optional.

Backup Parameters

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Table 14: Nutanix AHV Backup Parameters

Parameter	Values	Default	Description
include	`<Java Regexp>`		Specifies a list of guest VM names to back up. Uses Java-compatible regular expressions.
exclude	`<Java Regexp>`		Specifies a list of guest VM names to exclude from the backup.
vm	`<guest VM name>`		Specifies the name of a single guest VM to back up, regardless of other include/exclude rules.
protection_do	`<protect domain name>`		When set, only virtual machines belonging to the specified protection domain are considered for backup. VMs in the domain are still filtered according to `include`, `exclude`, and `vm` parameters. This parameter is optional and is unspecified by default.
application_co	`<true|tr`	`try`	Specifies if the snapshot taken during the backup should be application consistent (`true`), crash consistent (`false`), or try application consistent first and fallback onto crash consistent if it fails (`try`). This parameter is optional.
dat_file	`<String>`	`nutan dat`	Specifies the name of the `.dat` file used to track dependencies between snapshots. This argument is optional.

The use of regular expressions in the parameters `include=` and `exclude=` must be a Java compatible regular expression.

In order to be backed up the guest VM must match the `include=...` predicate and not match the `exclude=...`. A guest VM that matches the `vm=...` will be backed up regardless of the include/exclude specifications.

By default all guest VMs match the include predicate and not the exclude. Therefore, if none of the parameters `vm=...`, `include=...` and `exclude=...` are provided, all available guest VMs hosted on the Nutanix-AHV hypervisor will be backed up.

On the other hand if the parameter `vm=...` is specified all guest VM will no longer match the include predicate. This means that if only `vm=...` parameter is specified no other guest VM will be backed up.

See Fileset Examples for examples of `include/exclude/vm` setups.

Restore Parameters

Table 15: Nutanix AHV Restore Parameters

Parameter	Values	Default	Description
disconnect_netw	[<0 or 1>]	0	Specifies whether restored network interfaces should be disconnected from their network.
network_addr	<Strin		Specified when a network interface should request a static IP address at restore. Takes a `name:ip_address` tuple. See the Fileset examples.
new_hostn	<Strin		Specified when a guest VM should be restored with a specific name.
where	<Strin		If this parameter is set, then the disks will be restored locally at the parameter value location. Each disk will be restored as `<where>/<uuid>.disk` file.

During restore the Nutanix-AHV Plugin uses the generic plugin parameters to access the Nutanix-AHV API. The parameters may be modified if necessary at restore time.

A restore job can only restore one guest VM at a time. To select the relevant VM the user should interact with Bacula *bconsole* and `mark` all files in the guest VM folder identified by its UUID.

```
cwd is: /
$ ls
$ @nutanix-ahv/
$ cd @nutanix-ahv
cwd is: /@nutanix-ahv/
$ ls
690d74f9-9ce4-45fb-9b23-149afebe18f7/
690d74f9-9ce4-45fb-9b23-149afebe18f7_VmName.name
$ cd 690d74f9-9ce4-45fb-9b23-149afebe18f7/
cwd is: /@nutanix-ahv/690d74f9-9ce4-45fb-9b23-149afebe18f7/
$ ls
eef9d485-dba3-4e3f-b828-46a8c3412bec.abmp
eef9d485-dba3-4e3f-b828-46a8c3412bec.bmp
eef9d485-dba3-4e3f-b828-46a8c3412bec.bvmdk
$ cd ..
cwd is: /@nutanix-ahv/
$ mark 690d74f9-9ce4-45fb-9b23-149afebe18f7*
4 files marked.
$ done
```

Warning: It is important to include the relevant `.name` file in the restoration tree to descriminate between files at restore time. Especially if more than one VM in the backup job.

Fileset Examples

In the example below, a single guest VM with a name of "VM1" will be backed up.

```
Fileset {
    Name= Nutanix_single
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM vm=VM1"
    }
}
```

In the example below, a single guest VM with a name of "VM1" inside a protection domain "pDomain" will be backed up.

```
Fileset {
    Name= Nutanix_single_domain
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM vm=VM1␣
↪protection_domain=pDomain"
    }
}
```

In the example below, all guest VMs which have `prod` in their name will be backed up.

```
Fileset {
    Name= Nutanix_prod
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM␣
↪include=(.*)prod(.*)"
    }
}
```

In the example below, all guest VMs which have `prod` in their name but do not start with `test` will be backed up.

```
Fileset {
    Name= Nutanix_no_test
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM␣
↪include=(.*)prod(.*) exclude=^test(.*)"
    }
}
```

In the example below, all `prod` VMs will be backed up. All `test` VMs will be ignored except for a VM named `exception.test`

```
Fileset {
    Name= Nutanix_prod_no_test_except
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM␣
↪include=(.*)prod(.*) exclude=(.*)test(.*) vm=exception.test"
    }
}
```

In the example below, at restore time the selected guest VM will have its network interfaces disconnected and the network interface linked to `net1` will request the `10.0.110.11` ip address.

```
Fileset {
    Name= Nutanix_disconnect
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM␣
↪disconnect_network network_address=net1:10.0.110.11"
    }
}
```

Note: There are different ways to use `disconnect_network` and `abort_on_error`: `<parameter>=1` is strictly equal to the presence of `<parameter>` inside the plugin line. In the same spirit the absence of such parameter in the plugin line is equivalent to `<parameter>=0`.

In the example below, at restore time the selected guest VM will have its network interface connected to `net1` request the ip `10.0.110.11` and the network interface `net2` request the address `127.0.0.11`. Note that the `disconnect_network network_address` must be present in the Fileset resource at the backup time to benefit from it at restore time.

```
Fileset {
    Name= Nutanix_net
    Include {
        Plugin="nutanix-ahv: user=root host=1.2.3.4 proxy_vm=proxyVM network_
↪address=net1:10.0.110.11,net2:127.0.0.11"
    }
}
```

Operations

The following article describes details regarding backup, ingestion, restore and network restore with Bacula Enterprise Nutanix-AHV Plugin.

Note: The Nutanix-AHV plugin allows the request of static IP addresses only on a managed network.

Backup

The backup of a single guest VM takes the following steps:

- Take a snapshot of the guest VM through a hidden Nutanix API endpoint.
- Export guest VM metadata configuration for future restore.
- Backup every disks on the guest VM by ingestion on proxy VM.
- Add new a snapshot to the tracker and look for snapshots to delete.

Restore

The Nutanix-AHV plugin restores data as a new guest VM.

The restore is analog to the backup process and goes through the following steps for each successive snapshot of the guest VM to be restored.

- Receive configuration metadata file.
- Receive two lists: one for the changed regions and another for zeroed regions.
- Receive the raw disk data file(s).
- Add a new virtual disk on the SCSI bus of the proxy VM (hotadd).
- Write all data regions back onto the disk(s).
- Once all disks have been restored create a new guest VM from the metadata configuration file
- Detach all restored disks from the proxy VM.
- Move all restored disks from proxy VM to the freshly restored guest VM.

bconsole Query Commands

The Bacula Enterprise Nutanix-AHV Plugin supports also the query parameter. Bacula queries the Nutanix-AHV API and receives answers in the form of `key=value`. The Nutanix-AHV Plugin supports three query parameters. When none of them is specified, the message: `Query not recognized possible value = {CONNECTION, MACHINES, NETWORK}` is displayed.

Connection

This query sends a REST API request to check whether the current plugin parameters allow connections to the Nutanix-AHV server. The example below shows the execution of a correctly formatted `CONNECTION` query and response.

When the query parameter is `CONNECTION`, the returned value will be either `OK` or `NOK`, indicating whether the connection was successful or not.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd␣
↪parameter=CONNECTION
CONNECTION=OK
```

VM

This query sends a REST API request that lists all available guest VMs and their respective UUID. This query displays two `key=value` tuples per virtual machine the first where `key` is `VM` and `value` is the name of the virtual machine, the second where `key` is `UUID` and `value` is the virtual machine's UUID.

The example below shows the execution of a correctly formatted `VM` query that finds three different guest VMs.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd␣
↪parameter=VM
VM=Virtual1
```

(continues on next page)

(continued from previous page)

```
UUID=87654321-1234-5678-9abc-defghijklmno
VM=Virtual2
UUID=12345678-abcd-dcba-1234-abcdefghijkl
VM=Virtual3
UUID=11111111-2222-aaaa-bbbb-12ab12ab12ab
```

Network

This query sends a REST API request that lists all available networks. This query displays one `key=value` tuple per network available where `key` is `NETWORK` and `value` is the network's name.

The example below show values output by a correctly formatted `NETWORK` query that finds two networks.

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd␣
↪parameter=NETWORK
NETWORK=Managed
NETWORK=Vlan-Nutanix
```

Protection domain

This query sends a REST API request that lists all available protection domains. This query displays protection domains names and the virtual machine included in each domain. This display starts with one `key=value` tuple for the protection domain name followed by zero or more `key=value` tuples one for each vritual machine inside the protection domain.

The example below shows the execution of a correctly formatted `PROTECTION_DOMAIN` query that finds

```
.query plugin="nutanix-ahv: user=admin host=1.2.3.4" client=client-fd␣
↪parameter=PROTECTION_DOMAIN
PROTECTION_DOMAIN=pDomain
VM=Virtual1
PROTECTION_DOMAIN=emptyPDom
PROTECTION_DOMAIN=pDomain01
VM=Virtual2
VM=Virtual3
```

Interactive Snapshot Deletion

The Bacula Enterprise Nutanix-AHV plugin backend comes with an additional feature to manually keep track of snapshots, which is a simple command line tool located in `/opt/bacula/bin`. To launch the CLI the user just needs to execute the backend script wih an additional `delete` parameter. The script asks the user for their username and the Nutanix-AHV host. The password will be read from the `snapmgr.conf` file.

The example below shows a regular execution of the interactive snapshot deletion feature.

```
root@backupvm:~# /opt/bacula/bin/nutanix_ahv_backend delete
Enter credential    USER=
admin
```

(continues on next page)

(continued from previous page)

```
Enter credential    HOST=
10.0.100.90
Do you want to treat virtual machine
    VM_NAME=TestTest
    VM_UUID=e95339ab-a15a-472e-977a-961e7f5440b1
[y]es / [n]o
y
Start interactive delete for UUID=e95339ab-a15a-472e-977a-961e7f5440b1
Snapshots detected= 11
Do you want to delete snapshot n°1
    UUID=0929948b-04ee-4df7-b7d2-1b451be9ad06
    SNAPSHOT_NAME=NutanixAhv.2022-02-15_11.07.41_09
[y]es / [n]o
```

Troubleshooting

This article presents troubleshooting for the Nutanix-AHV Plugin.

If the proxy VM boots to an initramfs shell after a failed backup/restore it is probably due to the fact that some of the disks that were mounted on the proxy VM are still attached.

To solve this issue the user has to manually remove the supernumerary disks manually through the PRISM console.

Another way to remove the disk is to issue REST API commands. A simple bash script like the following example will do the trick:

Warning: This script is pretty aggressive and will remove all disks from index 1 to 10 without asking anything. It should be used carefully and adapted if the initial proxy VM configuration has more than one disk on the SCSI bus.

```
for i in {1..10}
do

curl -k -X POST --header 'Content-Type: application/json' --header 'Accept:␣
↪application/json' -d '{
  "uuid": "$proxyVmUuid",
  "vm_disks": [
    {
      "disk_address": {
        "device_bus": "SCSI",
        "device_index": '$i'
      },
      "is_cdrom": false
    }
  ]
}' 'https://$HOST:$PORT/PrismGateway/services/rest/v2.0/vms/$proxyVmUUID/
↪disks/detach' -u '$USER:$PASSWORD'

done
```

The script will try to unmount all disks from index 1 up to 10. Index 0 is skipped since the proxy VM first disk is the one containing the VM data.

Limitations

This article presents limitations of the Nutanix-AHV Plugin.

- The proxy VM must be a VM in the Nutanix-AHV environment running a Linux distribution for the plugin to work.
- Disks located on the IDE bus cannot be backed up. Disk tray on the other hand will be backed up.
- Content mounted on a virtual disk tray will not be restored.
- Only one VM may be restored at a time. If a backup job contains more than one VM, then each VM will need to be restored separately.
- Virtual full jobs are not supported.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

CitrixHypervisor (XenServer) Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Features Summary

- Snapshot-based online backup of any guest VM
- VSS-based guest snapshots for quiescing VSS-based applications
- Full, Incremental and Differential block level image backup
- Ability to restore complete virtual machine image
- Ability to restore VM archive (.xva) to an alternate directory
- Support for CitrixHypervisor and XCP-ng

Note: This plugin was introduced with Bacula Enterprise version 10.0.

Incremental and Differential backup support was introduced with version 12.4.1.

Single item restore for XenServer VMs was introduced with Bacula Enterprise version 12.8.8.

Guest VM Backup Strategies

Installing Bacula Client on each Guest

With this first strategy, you do not use the Bacula Enterprise XenServer Plugin, but instead install a Bacula Enterprise File Daemon on every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on your XenServer hypervisor, you will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to spread your backup jobs over your backup window. Since all VMs could use the same storage on the XenServer hypervisor, running all your backup jobs at the same time could create a bottleneck on the disk/network subsystem.

Installing a Bacula Enterprise File Daemon on each virtual machine permits you to manage your virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files.
- Checksum of individual files for Virus and Spyware detection.
- Verify Jobs.
- File/Directory exclusion (such as swap or temporary files).
- File level compression.
- Accurate backups.
- etc.

Image Backup With XenServer Plugin

With the image backup level strategy, the Bacula Enterprise XenServer Plugin will save the Client disks at the raw level, in the XenServer context.

For this to work, you don't need a Bacula File Daemon on each guest VM. Bacula's XenServer plugin will contact your XenServer hypervisor to read and save the contents of your virtual machine disks using snapshots and XAPI. In this case Plugin can perform full range of block level image backup including Incremental and Differential ones.

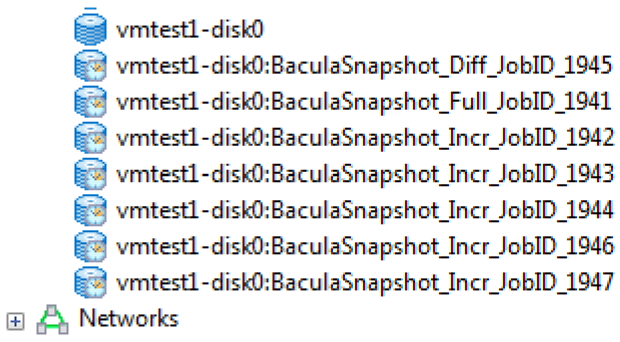
Bacula doesn't need to walk through the Client filesystem to open/read/close/stat files, so it consumes less resources on your XenServer infrastructure than a file level backup on each guest machine would. On the other hand, Bacula will also read and save useless data such as swap files or Internet temporary files. The XenServer Plugin will save not only the disk images of the:term:*guest VM*, but also the guest VM configurations which allows for very easy:term:*guest VM* restores.

Backup and Restore Operations

Backup

The backup operation of a single guest VM takes the following steps:

- Find and delete any old backup snapshots list in Full level backup.
- Create a new guest VM Bacula snapshot and prepare it for backup.
- Export VM Guest metadata configuration for future restore.
- For any Incremental or Differential backups compute block changed list for every virtual disk in snapshot.
- Export all raw images data or data based on changed block list if required.
- Execute the XenServer `vm-export` command and save the data to a Bacula storage daemon.
- Delete a backup snapshot data and maintaining a snapshot metadata only.



vmtest1-disk0
vmtest1-disk0:BaculaSnapshot_Diff_JobID_1945
vmtest1-disk0:BaculaSnapshot_Full_JobID_1941
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1942
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1943
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1944
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1946
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1947
Networks

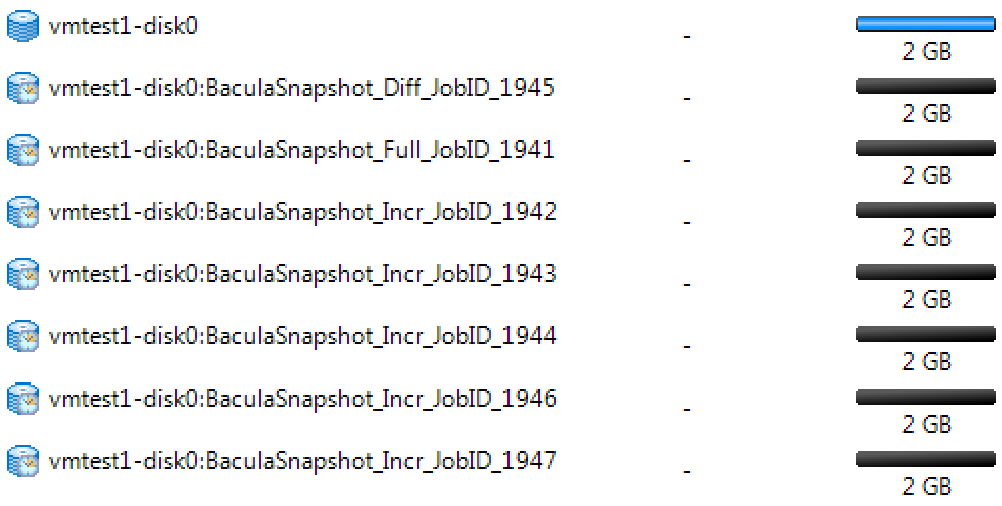
Fig. 9: Backup snapshot set.

Backups can be performed for a guest VM in any power state (running or halted). For proper execution of Incremental or Differential backups it is required by XenServer to maintain snapshot metadata which stores changed block information used to compute changed block list during backup. You can find it and display on VDI objects list. Every single backup for single VM will create single metadata VDI snapshot for every VDI attached to guest VM.

Metadata snapshots take minimal space and cannot be used directly for restore. They save a changed block bitmaps and no real data blocks. Every CBT-enabled disk has an additional CBT-metadata disk which is named as *<vdi_uuid>.cbtlog*, on the same SR.

- Size of a CBT-metadata disk on LVM based SRs is 4MB
- Size of a CBT-metadata disk on file based SRs is proportional to the size of the VDI, and can grow up to a size of 4MB (for a 2TB VDI)

Blocks of 64 kB within the VDI are tracked and changes to these blocks recorded in the log layer.



vmtest1-disk0	-	2 GB
vmtest1-disk0:BaculaSnapshot_Diff_JobID_1945	-	2 GB
vmtest1-disk0:BaculaSnapshot_Full_JobID_1941	-	2 GB
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1942	-	2 GB
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1943	-	2 GB
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1944	-	2 GB
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1946	-	2 GB
vmtest1-disk0:BaculaSnapshot_Incr_JobID_1947	-	2 GB

Fig. 10: Metadata snapshots free space.

Any guest VM snapshot with a name-label which matches the following template: *BaculaSnapshot_<UUID>_JobID_<NR>* or VM Guest VDI snapshot *<VDI-name-label>:BaculaSnapshot__JobID_<NR>* will be treated as an old backup snapshot for this VM Guest and automatically deleted during backup (*VDI snapshots during Full backups*). You should avoid creating a such snapshots manually.

Any other guest VM snapshots will be unaffected. The XenServer Plugin will inform you about every guest VM backup start and finish including information about old stalled backup snapshots and backup snapshot activities. For example:

```
JobId 1936: xenapi: Start Backup vm: vmtest1 (8024379c-c753-872a-5c25-
↪6c815ee617b4)
JobId 1936: xenctx: Cleaning old snapshots ...
JobId 1936: xenctx: Snapshot created: 5af34331-c6f2-e11b-c2c5-f1482c779eda
JobId 1936: xenapi: Finish backup of vmtest1-disk0:8066b4e4-9b42-4ed7-b908-
↪3494c9bd9094
...
```

The backup will create a following backup files during backup:

- a single file for VM configuration metadata saved in the form of: */@xen/<name-label>/<vmuuid>.conf*.
- a single file for VM disks configuration saved in the form of: */@xen/<name-label>/<vmuuid>.vmdisks*.
- a single file for every VM Guest VDI saved in the form of: */@xen/<name-label>/<vmuuid>/<vdi-name:vdi-uuid>.vdi*.

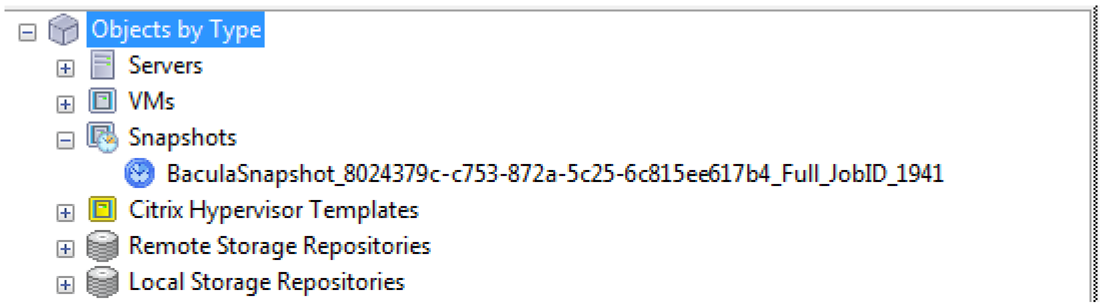

Objects by Type
Servers
VMs
Snapshots
BaculaSnapshot_8024379c-c753-872a-5c25-6c815ee617b4_Full_JobID_1941
Citrix Hypervisor Templates
Remote Storage Repositories
Local Storage Repositories

Fig. 11: VM snapshot during backup.

Multiple files will be created during backup if multiple VM Guest found to backup. You can use this information to locate the proper VM Guest archive during restore.

```
+-----------------------------------------------------------------------------
↪----------------------+
| filename                                                                    ␣
↪                      |
+-----------------------------------------------------------------------------
↪----------------------+
| /@xen/vmtest1/8024379c-c753-872a-5c25-6c815ee617b4.conf                     ␣
↪                      |
| /@xen/vmtest1/8024379c-c753-872a-5c25-6c815ee617b4.vmdisks                  ␣
↪                      |
| /@xen/vmtest1/8024379c-c753-872a-5c25-6c815ee617b4/disk0:8066b4e4-9b42-4ed7-
↪b908-3494c9bd9094.vdi |
+-----------------------------------------------------------------------------
↪----------------------+
```

XenServer Backup preparation

Before you start configuring your Backup Jobs you need to configure your XenServer to allows proper network operations with Bacula Enterprise. This part of the prerequisites includes:

1. enable network access to the XenServer API from the backup server - HTTP/HTTPS - ports `tcp:80` and `tcp:443`
2. enable network access to the XenServer NBD service from backup server - NBD/NBD-SSL - port `tcp:10809`
3. enable the XAPI NBD server on required network

To enable network access to the XenServer API you should enable and verify the firewall rules:

```
# iptables -L|grep http
ACCEPT     tcp  --  anywhere             anywhere             ctstate NEW tcp␣
↪dpt:http
ACCEPT     tcp  --  anywhere             anywhere             ctstate NEW tcp␣
↪dpt:https
```

To enable the XAPI NBD Server you should first check available virtual networks:

```
# xe network-list
uuid ( RO)                  : 774e87dd-a096-827a-5a30-6ef9123cfd7b
          name-label ( RW): Pool-wide network associated with eth0
    name-description ( RW):
              bridge ( RO): xenbr0

uuid ( RO)                  : 9fa3ecb2-6493-2849-e30b-eb772d1dbc1c
          name-label ( RW): Host internal management network
    name-description ( RW): Network on which guests will be assigned a␣
↪private link-local IP
              bridge ( RO): xenapi
```

Then you should enable NBD by setting up a *nbd-purpose* on selected networks. You can enable the NBD service in FORCEDTLS or NOTLS mode. You cannot have a mix of normal NBD (FORCEDTLS) and insecure NBD (NOTLS) networks. To switch the purpose of all networks, you must first disable normal NBD connections on all networks before enabling either normal or insecure NBD connections on any network.

Note: The XenServer vendor recommend to use TLS with NBD connections. When NBD connections with TLS are enabled, any NBD clients that attempt to connect to the XenServer must use TLSv1.2.

The Bacula Enterprise XenServer Plugin will work in either of the available NBD modes. To enable NBD connections with or without TLS, use the purpose parameter of the network. Set this parameter to include the value nbd for FORCEDTLS mode and the value `insecure_nbd` for NOTLS mode. Ensure that you wait for the setting to propagate before attempting to use this network for NBD connections. The time it takes for the setting to propagate depends on your network and is at least 10 seconds.

Below you can find an example of how to enable the NBD service in FORCEDTLS mode for selected *<network-uuid>*.

```
# xe network-param-add param-name=purpose param-key=nbd uuid=<network-uuid>
```

Some examples below.

```
 # Remove Insecure
    [19:14 po-xcp-ngserver home]# xe network-param-remove param-name=purpose␣
↪param-key=insecure_nbd uuid=b17e189c-5b13-feb6-c5cf-e509153ce3f0
    [19:15 po-xcp-ngserver home]# xe network-param-remove param-name=purpose␣
↪param-key=insecure_nbd uuid=53265e56-8ca0-db10-2c78-e80c89e2e4dd

# Enable Secure
    [19:15 po-xcp-ngserver home]# xe network-param-add param-name=purpose␣
↪param-key=nbd uuid=e80f1afe-ce32-af6e-a8d2-3047350d44ef
    [19:15 po-xcp-ngserver home]# xe network-param-add param-name=purpose␣
↪param-key=nbd uuid=b17e189c-5b13-feb6-c5cf-e509153ce3f0
    [19:15 po-xcp-ngserver home]# xe network-param-add param-name=purpose␣
↪param-key=nbd uuid=53265e56-8ca0-db10-2c78-e80c89e2e4dd

 # Restart Xapi
    systemctl restart xapi
```

For NOTLS mode you should replace `param-key=nbd` with `param-key=insecure_nbd`. You can check the network configuration as follows:

```
# xe network-param-list uuid=774e87dd-a096-827a-5a30-6ef9123cfd7b
uuid ( RO)                    : 774e87dd-a096-827a-5a30-6ef9123cfd7b
              name-label ( RW): Pool-wide network associated with eth0
        name-description ( RW):
               VIF-uuids (SRO): e61b768a-6460-af03-abf5-a2361b5b2f36;
               PIF-uuids (SRO): cbeab33d-b1c9-6fcb-bf03-d434c8623628
                     MTU ( RW): 1500
                  bridge ( RO): xenbr0
                 managed ( RO): true
            other-config (MRW):
                   blobs ( RO):
                    tags (SRW):
    default-locking-mode ( RW): unlocked
                 purpose (SRW): insecure_nbd
```

If you want to use the NBD service in FORCEDTLS mode you should setup the Bacula Enterprise plugin using the secure nbd configuration parameters to use a TLS certificate. Check the `certfile`, `keyfile`, `cacertfile` and `tlshostname` parameters for more information.

To use secure, TLS-encrypted and authenticated transport, note that you need to use a certificate and related key trusted by the Xen or XCP-ng server. If you are running a XCP-ng environment or a recent XenServer version, this will be mandatory, as any access to the APIs are protected by default with TLS and a self-signed certificate. More information:

- XenServer: https://docs.xenserver.com/en-us/xencenter/current-release/hosts-certificates.html
- XCP-ng: https://docs.xcp-ng.org/guides/TLS-certificates-xcpng/

If you plan to override your server certificates and use it with this plugin, below you will find an example of how to generate and install a self-signed one, but note that in most cases, these steps would involve a properly managed Certificate Authority and specific procedures:

```
# Generate a Certificate with its key
backupteam@example.org:~/certs# openssl req -new -newkey rsa:4096 -x509 -
↪sha256 -days 365 -nodes -out MyCertificate.crt -keyout MyKey.key
Generating a RSA private key
...............................................................................
↪..................++++
..............................................++++
writing new private key to 'MyKey.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
```

(continues on next page)

(continued from previous page)

```
Email Address []:

# Copy Certificates to XCP-NG Server
scp MyKey.key  root@My.Sample.Xen.Server:/tmp
scp MyCertificate.crt  root@My.Sample.Xen.Server:/tmp

# Generate a Pem Key on XCP-NG server
openssl req -new -x509 -key /etc/xensource/xapi-ssl.pem -subj '/CN=XCP-ng␣
↪hypervisor/' -out xcp-ng.csr

# Install Certificates on XCP-NG Server
xe host-server-certificate-install certificate=/tmp/MyCertificate.crt private-
↪key=/tmp/MyKey.key certificate-chain=/home/xcp-ng.csr

# Now use that certificate and key with the plugin parameters 'certfile' and
↪'keyfile'
```

Restore

The XenServer Plugin provides three main targets for restore operations:

- Restore to XenServer system as new VM Guest
- Restore to local directory as a number of archive files
- Restore individual files

Restore to XenServer

To use this restore method you have to set a “where=/” Bacula restore parameter. The guest VM archive will be sent to the XenServer hypervisor and always restored as a new guest VM. You can change the Storage Repository where your VM Guest will be restored.

To list available Storage Repositories you can use a listing mode, see **listing**. If you set an improper (eg: non-existent) Storage Repository for restore, then the restore process will fail. The Restore process requires a block level patching of saved disks which has to be performed on local filesystem (default directory: `/$WorkingDirectory/xenapi/$JobID/`).

Note: This is a XenServer limitation as XAPI does not support block level incremental restore but full image restore only.

Restore To Local Directory

To use this restore method you have to set a `where=/some/path` Bacula restore parameter. The `path` has to be a directory location on the server where the XenServer plugin is installed. If the path doesn't exist, it will be created by the XenServer Plugin.

Installation

You have to install and configure the Bacula File Daemon on your Backup machine which has access to XenServer API and XAPI-NBD services.

Note: It is not recommended to install it on Hypervisor dom0 as some XenServer services will not work properly:

> *"NDB connections do not work when an NBD client is in dom0 on the same host as the NBD server."*
>
> (source: https://support.citrix.com/article/CTX230619/how-to-troubleshoot-changed-block-tracking-in-xenserver)

As all backup interactions are network based, any Bacula Enterprise File Daemon with access to the necessary XenServer endpoints can be used to run the plugin.

Important: Since Bacula Enterprise 16.0.5, the XenServer Plugin is available for Debian 11 (bullseye).

Configuration

The `Plugin Directory` directive of the `File Daemon` resource in */opt/bacula/etc/bacula-fd.conf* should point where the `xenserver-fd.so` plugin is installed. The standard Bacula plugin directory is: */opt/bacula/plugins*

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
...
}
```

Installation of the Plugin

An example for a Debian based Linux distributions would be a file

`/etc/apt/sources.list.d/bacula.list` with the following content:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@cust@/debs/bin/@ver@/bullseye-64/␣
↪bullseye main
deb https://www.baculasystems.com/dl/@cust@/debs/xenserver/@ver@/bullseye-64/␣
↪bullseye xenserver
```

After that, a run of `apt-get update` is needed. Then, the Plugin can be installed using `apt-get install bacula-enterprise-xenserver-plugin`

Plugin Configuration

The plugin is configured using `Plugin Parameters` defined in a Fileset's "Include" section of the Bacula Enterprise Director's configuration.

Generic Plugin Parameters

The following XenServer plugin parameters effect any type of Job (Backup, Estimation, or Restore).

url=<address>
: specifies the XenServer API url address used for operations. This parameter is optional. If omitted the parameter will be assembled from `server=...` and `port=...` parameters defined below. You have to define the one of *url* or *server* parameters to connect to the XenServer API.

server=<address>
: specifies the XenServer API address used for operations. This parameter is optional if `url=...` parameter above is defined. This is the address used in *xe* command as *-s <address>* parameter during restore in legacy mode.

port=<number>
: specifies the XenServer API port used for operations. The value of the parameter have to be in range 1..65536. Invalid value will abort the job. This parameter is optional. If omitted the default 80/http access will be used. This is the value used in *xe* command as *-p <number>* parameter during restore in legacy mode.

user=<string>
: specifies the user name used to access the XenServer API. This parameter is required. This is the value used in *xe* command as *-u <string>* parameter during restore in legacy mode.

password=<string>
: specifies the password used to access the XenServer API. This parameter is optional if `passfile=...` is provided else it is required. This is the value used in *xe* command as *-pw <string>* parameter during restore in legacy mode. It is advised to use the `passfile=` option for more security.

passfile=<string>
: specifies a file local to the File Daemon that contains the password for the *user name*. This parameter is optional if `password=...` is provided else it is required. This is the value used in *xe* command as *-pwf <string>* parameter during restore in legacy mode.

abort_on_error[=<0 or 1>]
: specifies whether or not the plugin should abort it's execution if a fatal error happens during Backup, Estimation or Restore. This parameter is optional. The default value is 0.

ignore_ssl[=<0 or 1>]
: specifies whether or not the plugin should ignore SSL certificate checking when connecting to XenServer API. This parameter is optional. The default value is 0. With `NBD-SSL` configuration this parameter is not needed.

certfile=<string>
: Enable `NBD-SSL` communication during backup and use the specified file as the client certificate for TLS authentication to the server. This parameter is optional.

cacertfile=<string>
Enable `NBD-SSL` communication during backup and use the specified file as the CA certificate for TLS authentication to the server. This parameter is optional.

keyfile=<string>
Enable `NBD-SSL` communication during backup and use the specified file as the private key for the client cerificate. This parameter is optional.

tlshostname=<string>
Enable `NBD-SSL` communication during backup and use the specified hostname for the TLS context. If not specified, the hostname used to connect to the server will be used. This parameter is optional even if NBD-SSL connection is desired and normally it is not needed, but if you get any 'tls hostname not matching' kind of error, set a hostname here that is accepted in your TLS certificate CN values, as the behavior of the underlying NBD tools can vary among different environments.

Important: `certfile`, `cacertfile`, `keyfile` and `tlshostname` are available since Bacula Enterprise 18.0.4.

Estimation and Backup Plugin Parameters

vm=<name-label>
specifies a guest VM name to backup. All guest VMs with a name-label provided will be selected for backup. Multiple `vm=...` parameters may be provided. If a guest VM with <`name-label`> can not be found, then a single job error will be generated and the backup will proceed to the next VM unless `abort_on_error` is set which will cause the backup job to be aborted. This parameter is optional.

uuid=<uuid>
specifies a guest VM UUID to backup. Multiple `uuid=...` parameters may be provided. If a guest VM with <`uuid`> can not be found, then a single job error will be generated and the backup will proceed to the next VM unless `abort_on_error` is set which will cause the backup job to be aborted. This parameter is optional.

include=<name-label-regex>
specifies list of a guest VM names to backup using regular expression syntax. All guest VMs which match the name-label-regex provided will be selected for backup. Multiple `include=...` parameters may be provided. If no guest VMs match the <`name-label-regex`> provided, the backup will proceed to the next VM parameters or finish successfully without backing up any VMs. The `abort_on_error` parameter will not abort the job when no guest VMs are found using a <name-label-regex>. This parameter is optional.

exclude=<name-label-regex>
specifies list of a guest VMs names which will be excluded from backup using regular expression syntax. All guest VMs which match the name-label-regex provided and were selected for backup using `include=...` parameters will be excluded. This parameter does not affect any guest VMs selected to backup with `vm=...` or `uuid=...` parameters. Multiple `exclude=...` parameters may be provided. This parameter is optional.

quiesce[=<0 or 1>]
specifies if the guest VM snapshot should be created using a quiesce method or not. The quiesce method is supported by XenServer for Windows OS with Guest-Tools installed only. This is a limitation of the XenServer itself. If the guest VM snapshot with quiesce cannot be created, the

whole backup job will be aborted. In this case you should repeat a backup without the quiesce parameter.

If none of the parameters `vm=...`, `uuid=...`, `include` and `exclude` are specified, all available guest VMs hosted on the XenServer hypervisor will be backed up.

Plugin Restore Parameters

During restore, the XenServer Plugin will use the same parameters which were set for the backup job and saved. Some of them may be changed during the restore process if required. You can change all the parameters described in chapter *Generic Plugin Parameters* during restore.

- `storage_res: <storage>` specifies a XenServer Storage Repository where restored guest VMs will be saved. If not set, then a guest VM will be saved to a XenServer Storage Repository configured as the default. This parameter is optional.
- `preserve: <yes or no>` (**this option is deprecated and works for legacy (*.xva) restores only**) specifies if a restore job should preserve as much guest VM configuration parameters as possible. The default is to create a new VM on restore. A restore job with this preserve option set to 'yes' could fail if the restore might create duplicate objects on the XenServer hypervisor. This parameter is optional.

Fileset Examples

In the example below, all guest VMs will be backed up.

```
Fileset {
  Name = FS_XenAll
  Include {
    Plugin = "xenserver: url=http://10.10.10.10/ user=root password=root"
  }
}
```

In the example below, a single guest VM with a name-label of "VM1" will be backed up.

```
Fileset {
  Name = FS_Xen_VM1
  Include {
    Plugin = "xenserver: url=http://10.10.10.10/ user=root password=root␣
↪vm=VM1"
  }
}
```

The same example as above, but using `uuid` instead:

```
Fileset {
  Name = FS_Xen_VM1
  Include {
    Plugin = "xenserver: url=(...) uuid=fe1ccf3b-1865-3942-c928-d98138397ff1"
  }
}
```

where: `url=(...)` is a short for: *url=http://10.10.10.10/ user=root password=root* above and below.

In the example below, all guest VMs which have 'Prod' in the name will be backed up.

```
Fileset {
  Name = FS_Xen_ProdAll
  Include {
    Plugin = "xenserver: url=(...) include=Prod"
  }
}
```

In the example below, all guest VMs except VMs whose name-label begins with "Test" will be backed up.

```
Fileset {
  Name = FS_Xen_AllbutTest
  Include {
    Plugin = "xenserver: url=(...) include=.* exclude=^Test"
 }
}
```

In the example below, we connect through `NBD-SSL` mode to backup a XenExampleVM host:

```
Fileset {
   Name = "Xen-Example-Secure-VM"
   EnableVss = no
   IgnoreFilesetChanges = no
   Include {
      Options {
         Signature = Sha256
      }
      Plugin = "xenserver: server=\"my-xcp-server.example.org\" user=\"root\"␣
↪password=\"SamplePass\" abort_on_error include=XenExampleVM certfile=\"/
↪root/certs/MyCertificate.crt\""
   }
}
```

Restore

Restore to a XenServer Hypervisor

To restore a VM or VMs to a XenServer hypervisor, you should execute the restore command and specify the "where" parameter as in this example:

```
* restore where=/
     ...
```

Then set any other required restore plugin parameters for your restore.

In the following restore session example, the "Preserve vm config on restore" plugin restore option is set to "yes":

```
* restore where=/
...
```

(continues on next page)

(continued from previous page)

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/srv-xen-01-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   srv-xen-01-fd
Restore Client:  srv-xen-01-fd
Storage:         File1
When:            2018-01-05 12:47:16
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : xenserver: uuid=fe1ccf3b-1865-3942-c928-d98138397ff1
Plugin Restore Options
server:              *None*               (*None*)
port:                *None*               (*None*)
user:                *None*               (*None*)
password:            *None*               (*None*)
passfile:            *None*               (*None*)
storage_res:         *None*               (*Default location*)
preserve:            *None*               (*No*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: server (Restore server name)
     2: port (Restore server port number)
     3: user (Restore user name)
     4: password (Restore user password)
     5: passfile (Restore user password file)
     6: storage_res (Storage Resource location for restore)
     7: preserve (Preserve vm config on restore)
Select parameter to modify (1-7): 7
Please enter a value for preserve: yes
Plugin Restore Options
server:              *None*               (*None*)
port:                *None*               (*None*)
```

(continues on next page)

(continued from previous page)

```
user:                    *None*                  (*None*)
password:                *None*                  (*None*)
passfile:                *None*                  (*None*)
storage_res:             *None*                  (*Default location*)
preserve:                yes                     (*No*)
Use above plugin configuration? (yes/mod/no):
```

The restore job log will inform you about what guest VM is restored and what new guest VM was created.

```
JobId 131: Start Restore Job RestoreFiles.2017-12-28_14.42.25_15
JobId 131: Using Device "FileChgr1-Dev2" to read.
JobId 131: xenserver: VM restore: vm1/10908c8a-f932-6f91-9cac-3034e3acf45b
JobId 131: Forward spacing Volume "Vol-0002" to addr=1758441248
JobId 131: Elapsed time=00:04:51, Transfer rate=3.158 M Bytes/second
JobId 131: xenserver: VM UUID created: 45c49e07-ff20-ab55-e622-05ff2fbb0c1f
```

The new VM Guest created during restore will get the same name-label as the original VM. All VDIs connected to the restored VM will be marked with *-restored* suffix.

Restore to Local Directory

```
* restore where=/tmp/bacula/restores
```

Please check the following example for the test “VM local restore”:

```
JobId 112: Start Restore Job RestoreFiles.2017-12-28_11.30.19_34
JobId 112: Using Device "FileChgr1-Dev2" to read.
JobId 112: xenserver: VM local restore
JobId 112: Forward spacing Volume "Vol-0001" to addr=5190619786
JobId 112: Elapsed time=00:00:30, Transfer rate=30.64 M Bytes/second
```

The restore job log will inform you that a restore will go to a local directory.

Other

Resource listing

The Bacula Enterprise XenServer Plugin supports the new Plugin Listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available XenServer resources such as:

- List of guest VM name-labels
- List of guest VM UUIDs
- List of XenServer Storage Repositories

The new feature uses the special `.ls` command with a new `plugin=<plugin>` parameter. The command requires the following parameters to be set:

- `client=<client>` A Bacula Client name where the XenServer Plugin is installed.

- plugin=<plugin> A XenServer Plugin name - *xenserver:* with optional plugin parameters described at Sec. *Generic Plugin Parameters*
- path=<path> An object path to display

The supported values for the path=<path> parameter are:

- / - Display object types available to list
- vm - Display a list of guest VM name-labels
- uuid - Display a list of guest VM UUIDs and name-label pointers
- storage_res - Display a list of Storage Repositories

To display available object types, run the following command example:

```
*.ls client=srv-xen-01-fd plugin="xenserver: url=http://10.10.10.10/␣
↪user=root \
    password=root" path=/
Connecting to Client srv-xen-01-fd at 127.0.0.1:9102
drwxr-x---   1 root     root                 0 2018-01-02 09:36:32  vm
drwxr-x---   1 root     root                 0 2018-01-02 09:36:32  storage_res
drwxr-x---   1 root     root                 0 2018-01-02 09:36:32  uuid
2000 OK estimate files=3 bytes=0
```

To display the list of all available guest VMs, run the following command example:

```
*.ls client=srv-xen-01-fd plugin="xenserver: url=http://10.10.10.10/␣
↪user=root \
    password=root" path=VM
Connecting to Client srv-xen-01-fd at 127.0.0.1:9102
-rw-r-----   1 root     root      8589934592 2017-12-29 17:12:48  Another-
↪Copy of vm1
-rw-r-----   1 root     root     13958643712 2017-12-29 17:12:48  vm2
-rw-r-----   1 root     root      8589934592 2017-12-29 17:12:48  vm1
-rw-r-----   1 root     root     10737418240 2017-12-29 17:12:48  RHEL
-rw-r-----   1 root     root      8589934592 2017-12-29 17:12:48  Copy of vm1␣
↪a label with spaces
-rw-r-----   1 root     root     10737418240 2017-12-29 17:12:48  Copy of RHEL
-rw-r-----   1 root     root     19327352832 2017-12-29 17:12:48  vm1-orig
2000 OK estimate files=7 bytes=80,530,636,800
```

To display a list of all available guest VM UUIDs, run the following command example:

```
*.ls client=srv-xen-01-fd plugin="xenserver: url=http://10.10.10.10/␣
↪user=root \
    password=root" path=uuid
Connecting to Client srv-xen-01-fd at 127.0.0.1:9102
...
   8589934592 2018-01-02 09:39:06  4f5c9e10-a3c4-fc29-c967-4981f22d3f86 ->␣
↪Another-Copy of vm1
  13958643712 2018-01-02 09:39:06  50705972-0a88-5aa7-6721-f70b866ed0b6 -> vm2
   8589934592 2018-01-02 09:39:06  10908c8a-f932-6f91-9cac-3034e3acf45b -> vm1
  10737418240 2018-01-02 09:39:06  fe1ccf3b-1865-3942-c928-d98138397ff1 ->␣
↪RHEL
   8589934592 2018-01-02 09:39:06  c8efc2ca-ca1a-ebdf-5409-5dd8c158e3eb ->␣
```

(continues on next page)

(continued from previous page)

```
↪Copy of vm1 a label with spaces
  10737418240 2018-01-02 09:39:06  6e84929a-1c52-4c79-c67c-8455f76d3e7c ->␣
↪Copy of RHEL
  19327352832 2018-01-02 09:39:07  03fad8c9-d88b-ea7e-98da-2f3bcd20d0c4 ->␣
↪vm1-orig
2000 OK estimate files=7 bytes=80,530,636,800
```

The VM and UUID lists display an estimated size of the guest VM. To display a XenServer Storage Repositories, run the following command example:

```
*.ls client=srv-xen-01-fd plugin="xenserver: url=http://10.10.10.10/␣
↪user=root \
    password=root" path=storage_res
Connecting to Client srv-xen-01-fd at 127.0.0.1:9102
brw-r-----    1 root     root    586081632256 2018-01-02 09:39:22  ISO
brw-r-----    1 root     root    586081419264 2018-01-02 09:39:22  Local␣
↪storage
brw-r-----    1 root     root               0 2018-01-02 09:39:22  Removable␣
↪storage
brw-r-----    1 root     root      1073741312 2018-01-02 09:39:22  DVD drives
brw-r-----    1 root     root    586081632256 2018-01-02 09:39:22  Exported␣
↪Storage
brw-r-----    1 root     root              -1 2018-01-02 09:39:22  XenServer␣
↪Tools
2000 OK estimate files=6 bytes=0
```

Single Item Restore

(see separate article about CitrixHypervisor (XenServer) Single Item Restore that you can download on the top of the page of that article)

CitrixHypervisor Single Item Restore

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This article presents how to use the CitrixHypervisor (XenServer) Single File Restore feature with **Bacula Enterprise** and the CitrixHypervisor Plugin.

Features Summary

The **Bacula Enterprise** CitrixHypervisor (XenServer) Single File Restore provides the following main features:

- Console interface
- Support for Full/Differential/Incremental jobs
- Support for Linux filesystems (ext3, ext4, btrfs, lvm, xfs)
- Support for Windows filesystems (FAT, NTFS)

CitrixHypervisor (XenServer) SIR is available starting with Bacula Enterprise 12.8

This document will present solutions for **Bacula Enterprise** 12.8 and later, which are not applicable to prior versions. The CitrixHypervisor (XenServer) Single File Restore has been tested and is supported on RHEL 7.x, RHEL 8.x, Ubuntu Focal and Debian Stretch. SELinux is currently not supported.

Installation

Packages for the CitrixHypervisor (XenServer) Single File Restore plugin are available for supported platforms. Please contact Bacula Systems to get them.

Download the plugin package to your **Storage Daemon** server and then install using the package manager like so:

```
rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the CitrixHypervisor (XenServer) Single File Restore plugin and will install dependencies. On RHEL, it will be needed to install `perl-JSON` package from **rpmforge** and the `libguestfs-winsupport` package.

Note: On RHEL 7/8.x, it is necessary to install a custom version of the libguestfs packages from our repository to support NTFS devices. Those should not be updated with a newer version from official repositories. The YUM package manager has plugins to prevent package updates, try **yum-plugin-versionlock** or **yum-plugin-priorities**.

Additionally, the `ntfs-3g` package from the EPEL repository is needed for NTFS support. To install the EPEL respository, please follow the official instructions on the EPEL website to install the "epel-release" package here:

https://fedoraproject.org/wiki/EPEL

Note: On RHEL 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/DAG/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0

# cat /etc/yum.repos.d/baculasystems.repo
[single_file_restore_hyperv]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/<xxx>/rhel6-64
enabled = 1
```

(continues on next page)

(continued from previous page)

```
protect = 0
gpgcheck = 0


Note: This last repository is required on RHEL7:

[Bacula-Enterprise-DAG-Guestfish]
name = Bacula Enterprise - DAG for Guestfish
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64/guestfish/
enabled = 1
protect = 0
gpgcheck = 0
```

```
# yum install bacula-enterprise-single-item-restore perl-JSON
```

If BWeb Management Suite is used:

```
# service bweb restart
```

Notes about the “bacula” Account on RHEL

All commands in this document use the “bacula” unix account to run.

On RHEL, the Unix “bacula” account is locked by default. It means that it’s not possible by default to execute a command such as “su - bacula”.

It is possible to unlock the “bacula” account, or to use “sudo -u bacula” to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

Can be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using

```
root@storage# sudo -u bacula /bin/bash
```

Or unlock the “bacula” unix account and use “su -” with a command such as:

```
root@storage# chsh -s /bin/bash bacula
root@storage# su - bacula
bacula@storage# whoami
bacula
```

Fuse FileSystem

If a restore session is not properly cleaned up, some directories might still be mounted with the Bacula Fuse FileSystem.

```
baculafs on /opt/bacula/working/cat-ro type fuse.baculafs (ro,user=bacula)
backend0 on /opt/bacula/working/test-vm-0 type fuse.backend0 (ro,user=bacula)
/dev/fuse on /opt/bacula/working/test-vm type fuse (rw,nosuid,nodev,
↪user=bacula)
```

It is possible to unmount directories with the `fusermount -u` command.

```
bacula@storage# fusermount -z -u /opt/bacula/working/26
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm-0
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm
```

Samba SMB Shares

The **Bacula Enterprise** CitrixHypervisor (XenServer) Single File Restore plugin can automatically set up Samba SMB shares from the console program or the BWeb Management Suite.

To enable Samba SMB network shares, installing and configuring the “samba” package is mandatory. To configure the `/etc/samba/smb.conf` file correctly, you need to run `install-single-item-restore.sh` script.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
include = /etc/samba/conf.d/all
```

At this point, it is possible to modify `/etc/samba/smb.conf` to add your own configuration directives.

Network share descriptions will be stored in the directory `/etc/samba/conf.d`. It is possible to create and customize the template used by Bacula to generate configuration files.

```
root@storage# cat /etc/samba/conf.d/custom.tpl
[__share__]
   path = __path__
   follow symlinks = yes
   wide links = yes
   writable = yes
```

Configuration

On the **Storage Daemon** host server, the `bconsole` program should be configured properly to let the "bacula" user connect to the Director with `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 12.8.0
Enter a period to cancel a command.
* version
mydir-dir Version: 12.8.0 x86_64-redhat-linux-gnu
* quit
```

The package contains a script to test the connection with the Director and to test if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage#  /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you may not see any files or files may have strange permissions, ownerships and sizes and will inaccessible even to the root user.

Restore Scenario With Text Console Interface

The CitrixHypervisor (XenServer) Single File Restore plugin provides a simple console program that provides access to files inside VMs.

```
bacula@storage# /opt/bacula/bin/mount-vm
Automatically Selected Catalog: MyCatalog

Client list:
1: 127.0.0.1-fd
2: win2008-fd
3: rhel7-fd
Select a Client: 1
Selected Client: 127.0.0.1-fd

Job list:
1: XENSERVER.2021-02-15_19.12.51_34
2: XENSERVER.2021-02-16_12.12.29_39
3: XENSERVER.2021-02-16_12.37.54_03
```

(continues on next page)

(continued from previous page)

```
4: XENSERVER.2021-02-16_14.23.47_03
5: XENSERVER.2021-02-16_15.45.32_03
6: XENSERVER.2021-02-16_17.00.47_52
Select a Job: 6
Selected XENSERVER.2021-02-16_17.00.47_52

Virtual Machine:
1: squeeze2
2: win2008
3: rhel7
4: sir-test-vm
Select a Virtual Machine: 4
Selected sir-test-vm

Actions list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Cleanup
Select a Actions: 1
Selected Mount guest filesystem locally

I: Files are available under /opt/bacula/working/mount-vm-6434/disks/sir-test-
↪vm
I: Press enter to finish and cleanup the session
```

In this step, the virtual machine filesystem is mounted locally (in the example above, files are available under `/opt/bacula/working/mount-vm-6434/disks/sir-test-vm`. It is possible to browse directories and copy files (with `cp`, `scp`, `ftp`) as with a standard filesystem from another terminal session with the Unix "root" and "bacula" accounts. If you need to use another Unix account to operate on files, use the "-o allow_other" option when starting the `mount-vm` script.

```
bacula@storage# ls /opt/bacula/working/mount-vm-6434/disks/sir-test-vm
bin   dev  home        lib         media  opt   root  selinux  sys  usr ␣
↪vmlinuz
boot  etc  initrd.img  lost+found  mnt    proc  sbin  srv      tmp  var
```

To clean up the session, just press "Enter" in the terminal session where the `mount-vm` script was started.

It is possible to limit the Job list with the following command line options:

- `-s=<days>` Limit the job list to the last *days*
- `-l=<number>` Limit the job list to the last *number* entries
- `-f=<filter>` Specify an advanced filter based on the Job name, the Fileset name or the JobId

```
# Limit the job output to the last 100 jobs
bacula@storage# /opt/bacula/bin/mount-vm -l 100

# Limit the job output to the last 30 days
bacula@storage# /opt/bacula/bin/mount-vm -s 30

# Limit the job output to jobs that start with ``MyXenServer''
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyXenServer*'
```

(continues on next page)

(continued from previous page)

```
# BAD USAGE for the filter option, it will search for a job named␣
↪``MyXenServer''
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyXenServer'

# Limit the job output to jobs that start with ``MyXenServer''
# and that use the Fileset Test1
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyXenServer*␣
↪fileset=Test1'

# Limit the job to the jobid XX
bacula@storage# /opt/bacula/bin/mount-vm -f jobid=XX
```

In some cases, the device detection doesn't work properly. It is possible to use the `-m` option to mount recognized disks in a simple way. The option is automatically set when only one disk is selected during the restore.

```
bacula@storage# /opt/bacula/bin/mount-vm -m
```

Notes

Cache Directory

To speed up future CitrixHypervisor (XenServer) Single File restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
bacula@storage# ls /opt/bacula/working/mount-cache
sir-test-vm-0.bmp  sir-test-vm-2.bmp    MyCatalog-2.idx  MyCatalog-5.idx ␣
↪MyCatalog-8.idx
sir-test-vm-1.bmp  sir-test-vm.profile  MyCatalog-4.idx  MyCatalog-6.idx ␣
↪MyCatalog-9.idx
```

It is possible to remove files in the cache after some time; they will be re-generated if needed.

Support

The `install-single-item-restore.sh` script can collect traces automatically when a `mount-vm` session is running.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh support
```

Limitations

- The CitrixHypervisor (XenServer) Single File Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backend due to internal MySQL limitations with indexes on TEXT colums. For CitrixHypervisor (XenServer) Single Item Restore there should not be too much impact on performances (the backup structure is usually quite small) but we advise using the PostgreSQL backend for the best experience.
- The CitrixHypervisor (XenServer) Single File Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was created with a large number of concurrent jobs.
- The Storage Daemon where the CitrixHypervisor (XenServer) Single File Restore is installed should be have a CPU with the VT-x/EPT extensions. If these extensions are not available, the performance will be degraded. (From 20s to 10mins in our lab).
- RHEL 7/8 does not support mounting NTFS disks with the libguestfs provided with their system. To mount Microsoft NTFS disks on RHEL 7/8, it is required to install a patched version of the libguestfs packages. Please see notes in the "Installation" section of this document for more information.
- The CitrixHypervisor (XenServer) Single File Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.

Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a "multi-VM" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called "abort_on_error" in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 configuration (one VM backed up per job) means that the "abort_on_error" option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.

- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up "serially", one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

Limitations

- Snapshot with quiesce backups are only supported for Windows OSes with the XenServer Guest-Tools installed. This is a XenServer limitation and not a limitation of the Bacula Enterprise XenServer Plugin.
- You cannot run two concurrent backups of the single VM Guest if the later one is a Full backup.
- You have to provide enough free space at `/$workingDirectory/xenapi/` which allows to raw disk images to restore and perform a block level incremental/differential patching. This is a XenServer limitation and not the Bacula Enterprise XenServer Plugin as API requires full virtual disk image uploading only - no partial block level patching. This limitation could be removed in the future as soon as XenServer API will provide a sufficient functionality.
- The XenServer Plugin requires an NBD connection, which is not compatible with RHEL systems, consequently impacting RHEL derivative platforms such as Alma or Rocky Linux.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a "multi-VM" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.

- To address this issue, there is a plugin option called "abort_on_error" in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 configuration (one VM backed up per job) means that the "abort_on_error" option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.
- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up "serially", one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

Hyper-V Backup

Note: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Bacula Enterprise offers three ways to backup your Hyper-V virtual machines:

Important: The HyperV WinAPI plugin is the newest and most advanced plugin available for backing up Hyper-V environments. It offers the most comprehensive set of features, ensuring seamless integration and efficient backups. Due to its robust functionality and extensive capabilities, this plugin is now the recommended solution for all Hyper-V backup needs.

- *Hyper-V Win API*

The HyperV WinAPI plugin allows for full, differential, and incremental backups of Hyper-V virtual machines, supporting migration of VMs in cluster environments without requiring local snapshots. Single Item Restore is supported.

Warning: The Hyper-V VSS and Hyper-V WMI solutions are older methods for backing up Hyper-V environments. These options are not recommended for use.

- *Hyper-V VSS Plugin*

VSS services enable **Bacula Enterprise** to do snapshot backup of your Hyper-V virtual machines.

Single Item Restore is supported, but differential and incremental level backup cannot be done using this method.

- *Hyper-V WMI Plugin*

The WMI provider allows you to perform differential and incremental backup for Microsoft Hyper-V. Single Item Restore is not possible using this method.

Hyper-V WinAPI Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Important: The HyperV WinAPI plugin is the newest and most advanced plugin available for backing up Hyper-V environments. It offers the most comprehensive set of features, ensuring seamless integration and efficient backups. Due to its robust functionality and extensive capabilities, this plugin is now the recommended solution for all Hyper-V backup needs.

Features summary

- Backups are performed on virtual machines based on VMs, with a configuration that relies on Failover Cluster. After migrating to the local node, remote VMs can be backed up. The migration process to or from the local node is automated.
- The hosted VMs are backed up using Full, Differential, and Incremental backup methods, without the need for exporting local snapshots. This approach helps to avoid excessive disk space usage for backups.
- When restoring, the complete virtual machine images are directly placed in their original location, eliminating the need for additional disk space.

Important notes

- The **Hyper-V WinAPI Plugin** exclusively supports local VMs. Therefore, remote VMs are automatically **migrated** to the local node, backed up, and then migrated back to their original node. For more information, refer to the Backup section.
- To perform backups using the **Hyper-V WinAPI Plugin**, the **accurate** option must be enabled.
- It is highly recommended to enable **compression** when using the **Hyper-V WinAPI Plugin** for backups.
- Attempting multiple backups of the same VM in parallel is not advisable. Each backup will attempt to recover ongoing actions from the previous one, which can result in errors.
- Backups created with the **Hyper-V WinAPI Plugin** are not compatible with Virtual Full jobs. It is important not to combine these two backup strategies as you will not be able to properly restore jobs from Virtual Full backups.
- Similarly, backups created with the **Hyper-V WinAPI Plugin** are not compatible with backups created using the **Hyper-V WMI plugin** or **Bacula Hyper-V Plugin**..

- The **Hyper-V WinAPI Plugin** requires Hyper-V Virtual Machines version 6.2 or higher to correctly handle Differential and Incremental backups.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Supported platforms

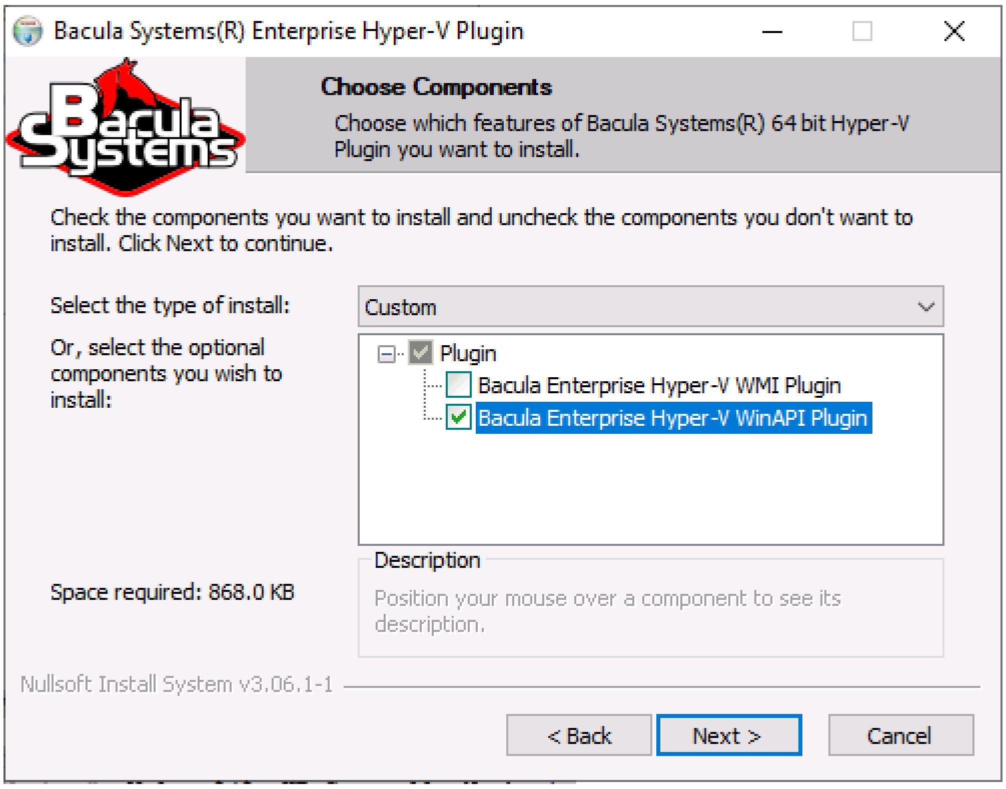
This documentation presents solutions for **Bacula Enterprise** 18.0 and higher, and is not applicable to prior versions of Bacula.

This Plugin supports Windows 8, Windows Server 2012 and later (Hyper-V Powershell module is required).

Installation

The Bacula File Daemon and the **Hyper-V WinAPI Plugin** need to be installed on the Hyper-V host server. The **Hyper-V WinAPI Plugin** Windows installer is the same as the **Hyper-V WMI Plugin** installer.

You can choose to install one or the other from the Plugin tree.


Bacula Systems(R) Enterprise Hyper-V Plugin

Choose Components

Choose which features of Bacula Systems(R) 64 bit Hyper-V Plugin you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Custom

Or, select the optional components you wish to install:

- [✓] Plugin
  - [ ] Bacula Enterprise Hyper-V WMI Plugin
  - [✓] Bacula Enterprise Hyper-V WinAPI Plugin

Space required: 868.0 KB

Description

Position your mouse over a component to see its description.

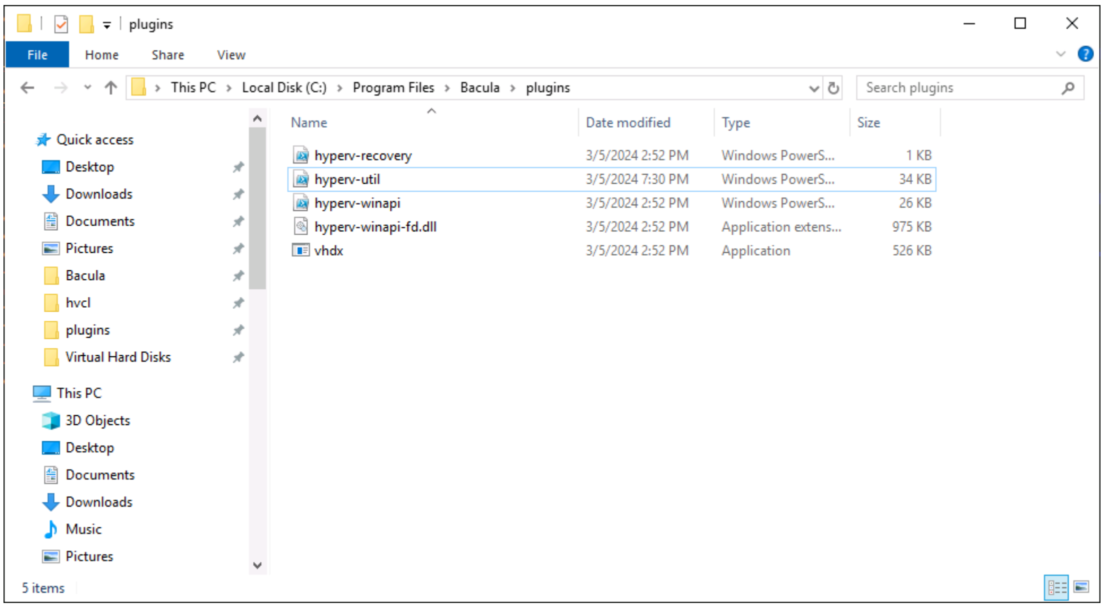
Nullsoft Install System v3.06.1-1

< Back | Next > | Cancel

It will deploy required components within the Bacula File Daemon plugins directory.

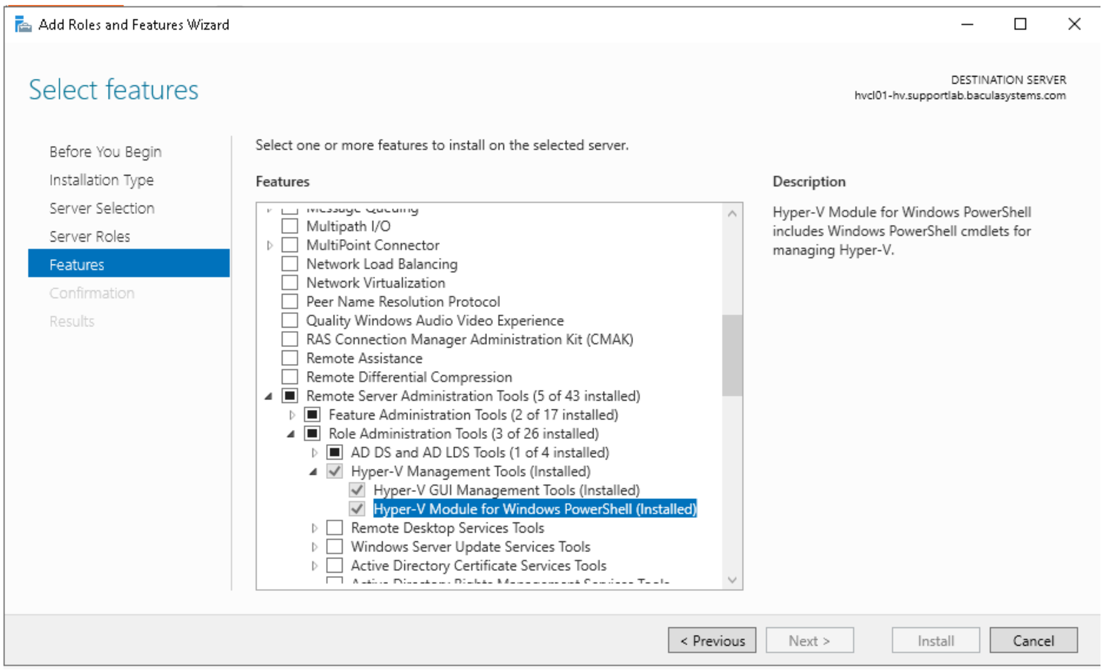
To configure the Bacula File Daemon, refer to the general Bacula installation documentation.

On the server side, the *Hyper-V PowerShell Module* needs to be enabled. On Windows Server or Hyper-V server 2012, 2016 and 2019, use Server Manager to install it. It should be located under Remote Server



Name	Date modified	Type	Size
hyperv-recovery	3/5/2024 2:52 PM	Windows PowerS...	1 KB
hyperv-util	3/5/2024 7:30 PM	Windows PowerS...	34 KB
hyperv-winapi	3/5/2024 2:52 PM	Windows PowerS...	26 KB
hyperv-winapi-fd.dll	3/5/2024 2:52 PM	Application extens...	975 KB
vhdx	3/5/2024 2:52 PM	Application	526 KB

Administration Tools -> Role Administration Tools -> Hyper-V Management Tools and check Hyper-V Module for Windows PowerShell.



Select one or more features to install on the selected server.

Features

```
Message Queuing
Multipath I/O
MultiPoint Connector
Network Load Balancing
Network Virtualization
Peer Name Resolution Protocol
Quality Windows Audio Video Experience
RAS Connection Manager Administration Kit (CMAK)
Remote Assistance
Remote Differential Compression
Remote Server Administration Tools (5 of 43 installed)
  Feature Administration Tools (2 of 17 installed)
  Role Administration Tools (3 of 26 installed)
    AD DS and AD LDS Tools (1 of 4 installed)
    [✓] Hyper-V Management Tools (Installed)
      [✓] Hyper-V GUI Management Tools (Installed)
      [✓] Hyper-V Module for Windows PowerShell (Installed)
    Remote Desktop Services Tools
    Windows Server Update Services Tools
    Active Directory Certificate Services Tools
    Active Directory Rights Management Services Tools
```

Description

Hyper-V Module for Windows PowerShell includes Windows PowerShell cmdlets for managing Hyper-V.

Verify the correct installation of the FD and the **Hyper-V Plugin** by running status client from bconsole or from BWeb.

```
*status client=w2019-hv01-fd
Connecting to Client w2019-hv01-fd at 172.22.22.50:9102
w2019-hv01-fd Version: 16.8.0 (06 April 2021) VSS Linux Cross-compile Win64
Daemon started 19-Jun-21 16:31. Jobs: run=23 running=2.
Microsoft Windows 2012 Standard Edition (build 9200), 64-bit
```

(continues on next page)

(continued from previous page)

```
Priv 0x73f
Memory: WorkingSetSize: 34,168,832 QuotaPagedPoolUsage: 183,768␣
↪QuotaNonPagedPoolUsage: 17,368 PagefileUsage:
43,687,936
APIs=OPT,ATP,LPV,CFA,CFW,
WUL,WMKD,GFAA,GFAW,GFAEA,GFAEW,SFAA,SFAW,BR,BW,SPSP,
WC2MB,MB2WC,FFFA,FFFW,FNFA,FNFW,SCDA,SCDW,
GCDA,GCDW,GVPNW,GVNFVMPW,LZO,EFS
Heap: heap=34,168,832 smbytes=39,489,074 max_bytes=69,259,353 bufs=395 max_
↪bufs=396
Sizes: boffset_t=8 size_t=8 debug=10 trace=1 mode=0,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL
APIs: !GPFS
Plugin: alldrives-fd.dll(1.2) hyperv-winapi-fd.dll(0.1) winbmr-fd.dll(3.1.0)
```

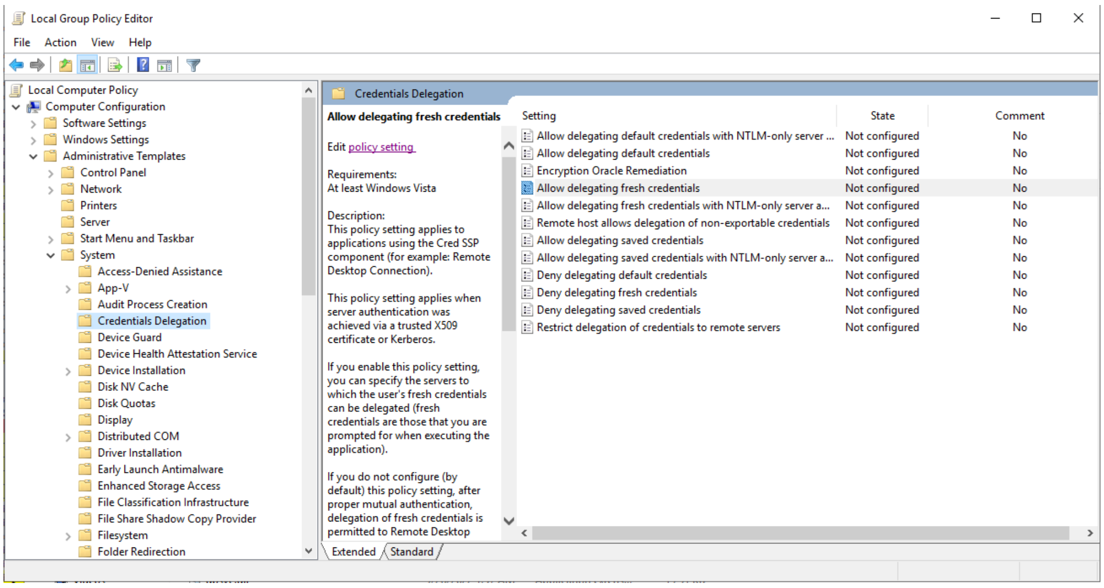
Verify that **hyperv-winapi-fd.dll** is in the "Plugin" line (last line in the above example output).

Important Considerations regarding Credentials Settings

Important: In order to access and backup the Hyper-V server, the delegation of the User credentials must be enabled and the Bacula File Daemon must be logged as an authorized user within the Hyper-V server.

Enable Delegation of User Credentials on Hyper-V Server

- Run gpedit.msc (normally in C:\Windows\System32) on the Hyper-V server and look at the following policy: Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials.



Allow delegating fresh credentials

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

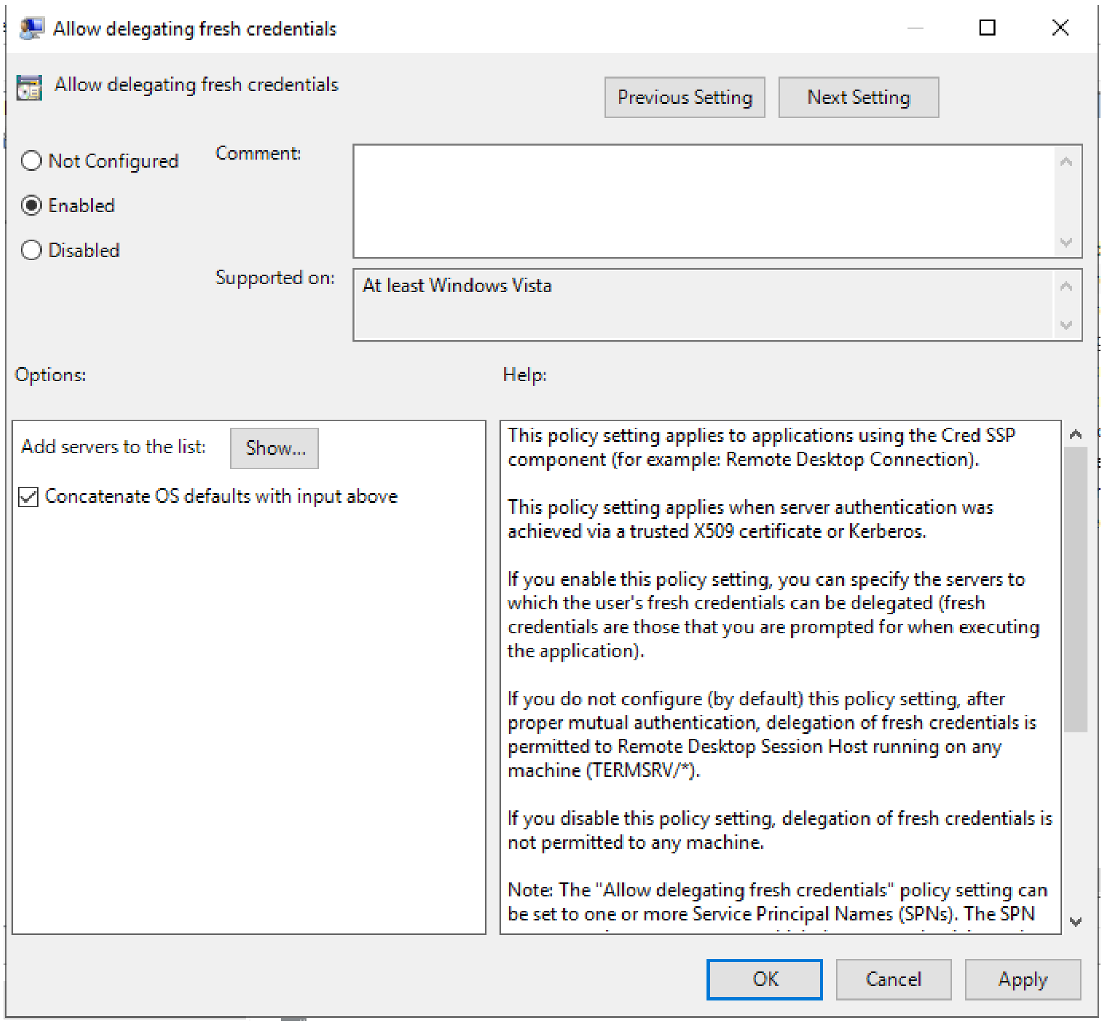
This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop

Setting	State	Comment
Allow delegating default credentials with NTLM-only server ...	Not configured	No
Allow delegating default credentials	Not configured	No
Encryption Oracle Remediation	Not configured	No
Allow delegating fresh credentials	Not configured	No
Allow delegating fresh credentials with NTLM-only server a...	Not configured	No
Remote host allows delegation of non-exportable credentials	Not configured	No
Allow delegating saved credentials	Not configured	No
Allow delegating saved credentials with NTLM-only server a...	Not configured	No
Deny delegating default credentials	Not configured	No
Deny delegating fresh credentials	Not configured	No
Deny delegating saved credentials	Not configured	No
Restrict delegation of credentials to remote servers	Not configured	No

- Verify that it is enabled and configured with the WSMAN SPN appropriate for the target computer.


Allow delegating fresh credentials

Allow delegating fresh credentials

( ) Not Configured
(●) Enabled
( ) Disabled

Comment:

Supported on: At least Windows Vista

Options:

Add servers to the list: Show...

[✓] Concatenate OS defaults with input above

Help:

This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop Session Host running on any machine (TERMSRV/*).

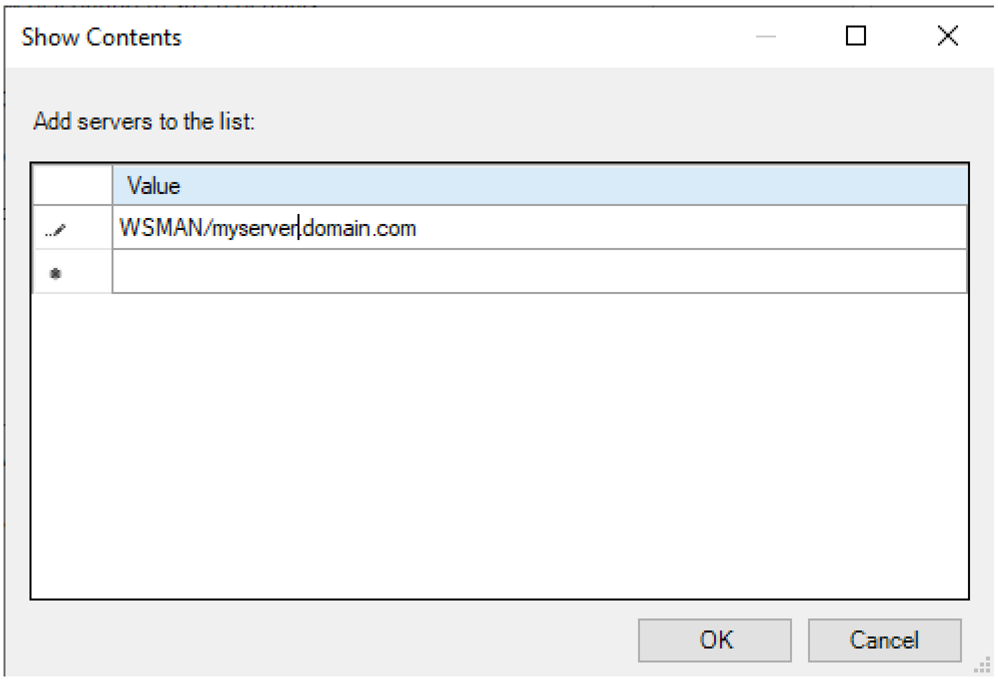
If you disable this policy setting, delegation of fresh credentials is not permitted to any machine.

Note: The "Allow delegating fresh credentials" policy setting can be set to one or more Service Principal Names (SPNs). The SPN

For example, for a target computer name "myserver.domain.com", the SPN can be one of the following: WSMAN//myserver.domain.com or WSMAN//*.domain.com. Introduce it in the "Add servers to the list" "Show" dialog box.

- Finally run a powershell console on the Hyper-V server (normally in C:\Windows\Systeme32\WindowsPowerShellv1.0powershell.exe) and enter the following commands:

```
Enable-WSManCredSSP -Role Server -Force
Enable-WSManCredSSP -Role "Client" -DelegateComputer myserver.domain.com -
↪Force
```



WSMAN/myserver.domain.com

Impersonation of Hyper-V WinAPI Plugin

The impersonation of the **Hyper-V WinAPI plugin** can be achieved in different ways.

- Specify the user name and password locally on the hyper-v node. This is the **recommended method**. In a `bacula-hyperv.pwd` file, located by the `bacula-fd.conf` config file (typically C:\Program Files\Bacula).

  `bacula-hyperv.pwd` contains the user name followed by the user password, separated by a colon.

  ```
  name@domain.com:mypassword
  ```

  or

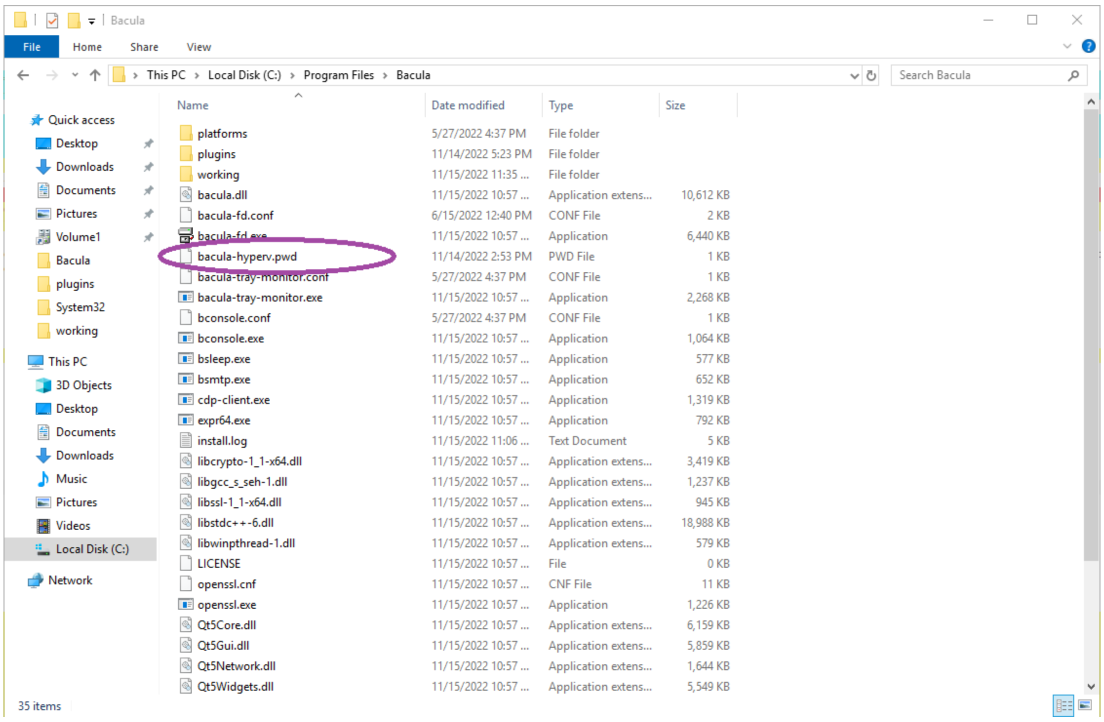
  ```
  DOMAIN\name:mypassword
  ```

- Impersonate the **Hyper-V WinAPI Plugin** by passing user and password, as plugin options. See Job configuration `user_name` and `user_password` options.
- Manually change the Bacula File Daemon default login account:

  Access the Hyper-V server using administrative credentials. Go to the Windows Start menu, type in "Services", and press Enter to display a list of all installed services. Locate the Bacula File Backup Service, right-click on it, and select Properties. Then, navigate to the Log On tab. The settings should appear as follows:

  Toggle the selection from "Local System account" to "This account". Enter the credentials of a Hyper-V administrator (either read only or read-write). Click OK.

  Click on the Bacula File Backup Service entry once more with the right mouse button, then select "Restart" to ensure that the changes take effect.



Bacula

File Home Share View

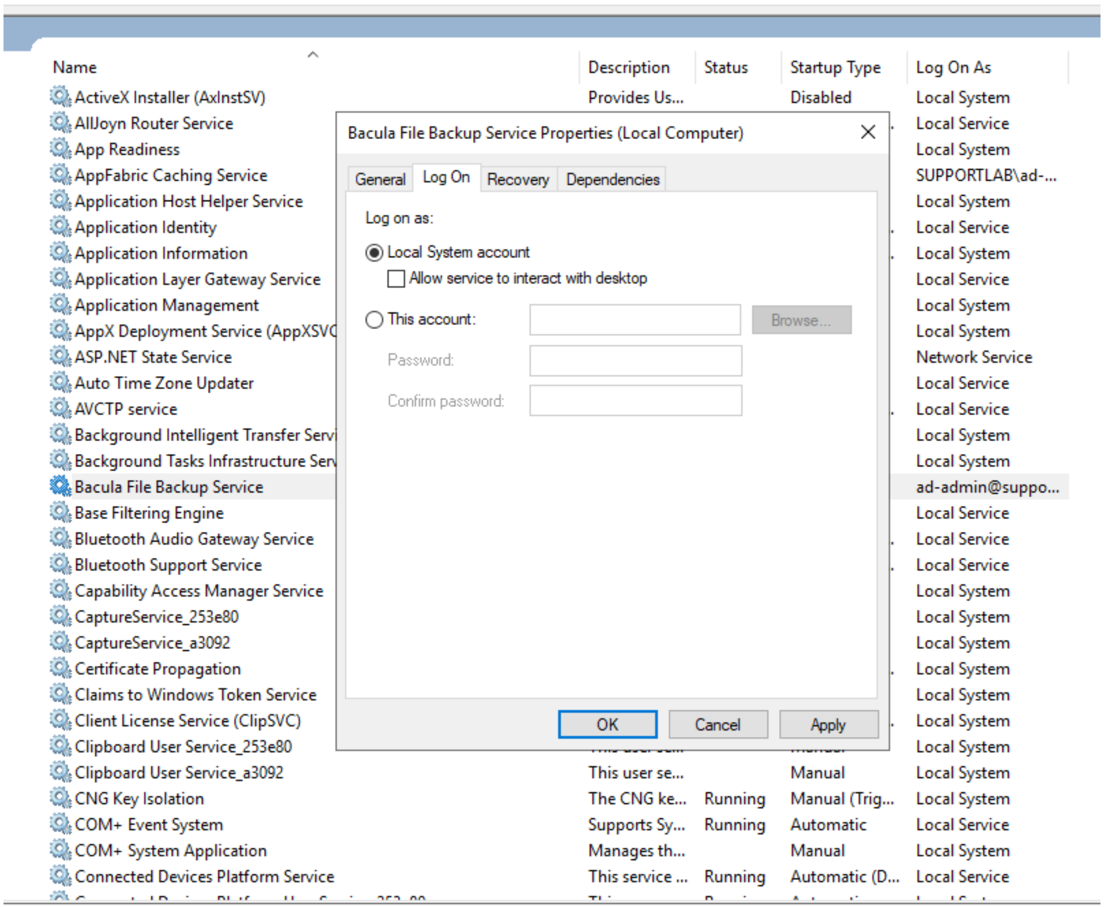
This PC › Local Disk (C:) › Program Files › Bacula

Search Bacula

Quick access, Desktop, Downloads, Documents, Pictures, Volume1, Bacula, plugins, System32, working, This PC, 3D Objects, Desktop, Documents, Downloads, Music, Pictures, Videos, Local Disk (C:), Network

Name	Date modified	Type	Size
platforms	5/27/2022 4:37 PM	File folder	
plugins	11/14/2022 5:23 PM	File folder	
working	11/15/2022 11:35 ...	File folder	
bacula.dll	11/15/2022 10:57 ...	Application extens...	10,612 KB
bacula-fd.conf	6/15/2022 12:40 PM	CONF File	2 KB
bacula-fd.exe	11/15/2022 10:57 ...	Application	6,440 KB
bacula-hyperv.pwd	11/14/2022 2:53 PM	PWD File	1 KB
bacula-tray-monitor.conf	5/27/2022 4:37 PM	CONF File	1 KB
bacula-tray-monitor.exe	11/15/2022 10:57 ...	Application	2,268 KB
bconsole.conf	5/27/2022 4:37 PM	CONF File	1 KB
bconsole.exe	11/15/2022 10:57 ...	Application	1,064 KB
bsleep.exe	11/15/2022 10:57 ...	Application	577 KB
bsmtp.exe	11/15/2022 10:57 ...	Application	652 KB
cdp-client.exe	11/15/2022 10:57 ...	Application	1,319 KB
expr64.exe	11/15/2022 10:57 ...	Application	792 KB
install.log	11/15/2022 11:06 ...	Text Document	5 KB
libcrypto-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	3,419 KB
libgcc_s_seh-1.dll	11/15/2022 10:57 ...	Application extens...	1,237 KB
libssl-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	945 KB
libstdc++-6.dll	11/15/2022 10:57 ...	Application extens...	18,988 KB
libwinpthread-1.dll	11/15/2022 10:57 ...	Application extens...	579 KB
LICENSE	11/15/2022 10:57 ...	File	0 KB
openssl.cnf	11/15/2022 10:57 ...	CNF File	11 KB
openssl.exe	11/15/2022 10:57 ...	Application	1,226 KB
Qt5Core.dll	11/15/2022 10:57 ...	Application extens...	6,159 KB
Qt5Gui.dll	11/15/2022 10:57 ...	Application extens...	5,859 KB
Qt5Network.dll	11/15/2022 10:57 ...	Application extens...	1,644 KB
Qt5Widgets.dll	11/15/2022 10:57 ...	Application extens...	5,549 KB

35 items



Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local System
AllJoyn Router Service				Local Service
App Readiness				Local System
AppFabric Caching Service				SUPPORTLAB\ad-...
Application Host Helper Service				Local System
Application Identity				Local Service
Application Information				Local System
Application Layer Gateway Service				Local Service
Application Management				Local System
AppX Deployment Service (AppXSVC				Local System
ASP.NET State Service				Network Service
Auto Time Zone Updater				Local Service
AVCTP service				Local Service
Background Intelligent Transfer Servi				Local System
Background Tasks Infrastructure Ser				Local System
Bacula File Backup Service				ad-admin@suppo...
Base Filtering Engine				Local Service
Bluetooth Audio Gateway Service				Local Service
Bluetooth Support Service				Local Service
Capability Access Manager Service				Local System
CaptureService_253e80				Local System
CaptureService_a3092				Local System
Certificate Propagation				Local System
Claims to Windows Token Service				Local System
Client License Service (ClipSVC)				Local System
Clipboard User Service_253e80				Local System
Clipboard User Service_a3092	This user se...		Manual	Local System
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local System
COM+ Event System	Supports Sy...	Running	Automatic	Local Service
COM+ System Application	Manages th...		Manual	Local System
Connected Devices Platform Service	This service ...	Running	Automatic (D...	Local Service

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

(●) Local System account

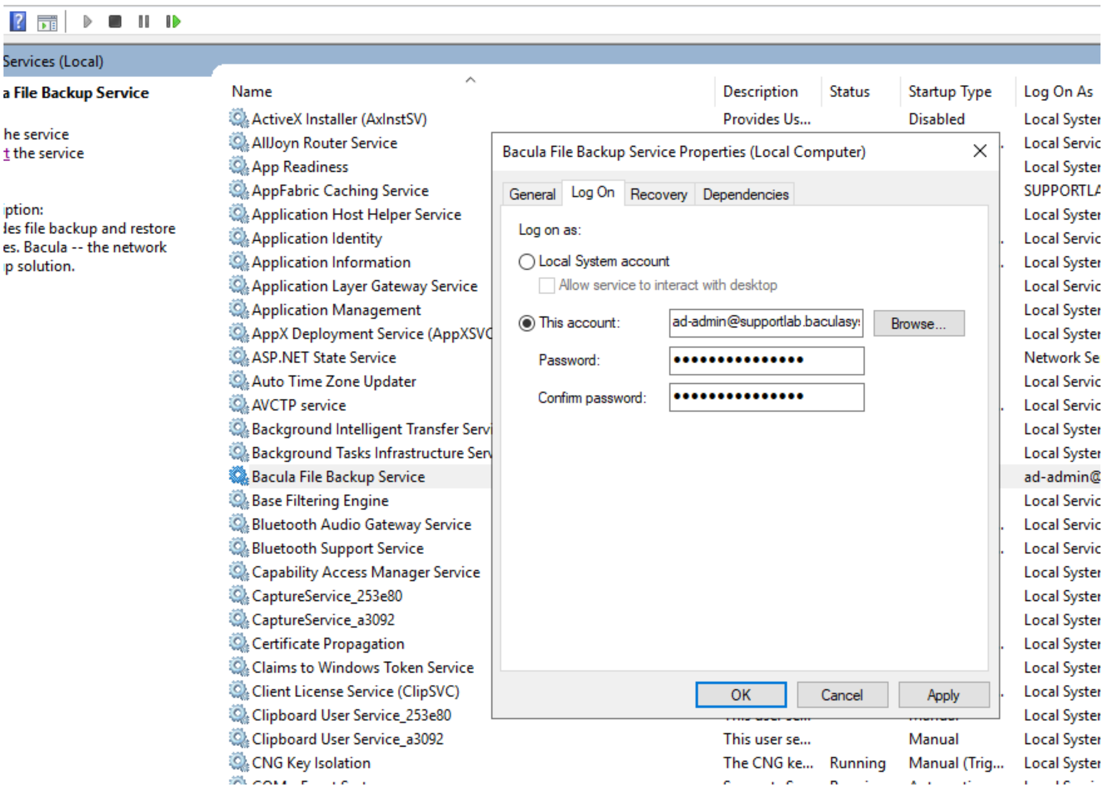
[ ] Allow service to interact with desktop

( ) This account: Browse...

Password:

Confirm password:

OK Cancel Apply



Services (Local)

a File Backup Service

he service
t the service

ption:
les file backup and restore
es. Bacula -- the network
p solution.

Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local Syster
AllJoyn Router Service				Local Servic
App Readiness				Local Syster
AppFabric Caching Service				SUPPORTLA
Application Host Helper Service				Local Syster
Application Identity				Local Servic
Application Information				Local Syster
Application Layer Gateway Service				Local Servic
Application Management				Local Syster
AppX Deployment Service (AppXSVC				Local Syster
ASP.NET State Service				Network Se
Auto Time Zone Updater				Local Servic
AVCTP service				Local Servic
Background Intelligent Transfer Servi				Local Syster
Background Tasks Infrastructure Serv				Local Syster
Bacula File Backup Service				ad-admin@
Base Filtering Engine				Local Servic
Bluetooth Audio Gateway Service				Local Servic
Bluetooth Support Service				Local Servic
Capability Access Manager Service				Local Syster
CaptureService_253e80				Local Syster
CaptureService_a3092				Local Syster
Certificate Propagation				Local Syster
Claims to Windows Token Service				Local Syster
Client License Service (ClipSVC)				Local Syster
Clipboard User Service_253e80				Local Syster
Clipboard User Service_a3092	This user se...		Manual	Local Syster
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local Syster

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

○ Local System account

☐ Allow service to interact with desktop

● This account: ad-admin@supportlab.baculasy:

Password: ••••••••••••••••

Confirm password: ••••••••••••••••

Job Configuration

Once the Bacula File Daemon and the **Hyper-V WinAPI Plugin** are correctly installed and configured, setting a backup job up is as simple as adding the job and the fileset within the Bacula Director configuration file.

Important: The `Enable VSS` parameter must be set to `no` in the Fileset (see examples below).

The following plugin options are supported:

Name	Sta-tus	De-fault	Description
in-clude	Op-tiona	In-clude all (*)	a Unix shell-style wildcards pattern for including VMs by name
ex-clude	Op-tiona	Ex-clude none	a Unix shell-style wildcards pattern for excluding VMs by name
tmp_dir	Op-tiona	`Bacul`	locates the Bacula working repository folder. Make sure there's enought space on this location to create VM's shapshots and exports. Default is a `Bacula-repo` folder in the VHD location.
pre_back	Op-tiona	None	action on the VMs before backup takes place. Can be None, Stop, Save. - None is noop. - Stop stops the VM before backup (useful when VM doesn't support VSS or kernel freeze to maintain consistent backups). - Save saves the VM before stoping it.
post_bac	Op-tiona	None	action on the VMs after backup is completed. Can be None, Restart, ForceRestart. - None is noop. - Restart restarts the VM if it was stopped or saved pre-backup. - ForceRestart restarts the VM unconditionnaly.
consis-tency_le	Op-tiona	Ap-pli-ca-tion	overwrites the consistency level. Can be Application Consistent of Crash Consistent. Application is the recommanded value but some VMs might not support it.
al-low_pre_	Op-tiona	Dis-abled	when enabled, allows retry with `pre_backup_action` set to Save and `post_backup_action` set to Restart, if crash consistency retry backup has failed.
clus-ter_mod	Op-tiona	Dis-abled	when enabled, the plugin will parse and allow migration of VM's that are not on the local node (for Failover Cluster configuration).
abort_or	Op-tiona	Dis-abled	abort immediately the job if a serious error is found (b.e when no VM matches the `include` patterns). By default, a Job error is raised, but the job continues.
wait_on_	Op-tiona	Dis-abled	when the VM is in "migrating" state, the plugin will wait for the VM migration completion before processing with the backup if the option is enabled. Otherwise, it skips the VM.
user_nar	Op-tiona	None	the user name that will run the backup/restore operation. This is **not the recommanded method**. The user name can be specified locally on the hyper-v node in a `bacula-hv.usr` file located in the fd plugins folder.
user_pas	Op-tiona	None	the user password that will run the backup/restore operation. This is **not the recommanded method**. The user password can be specified locally on the hyper-v node in a `bacula-hv.pwd` file located in the fd plugins folder.

Examples

Example 1: backup all vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupAll"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="Simplest-Hyper-V-Fileset"
```

(continues on next page)

(continued from previous page)

```
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}

Fileset {
  Name = "Simplest-Hyper-V-Fileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi:"
  }
}
```

Example 2: backup only «Linux- » prefixed vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupOnlyLinux"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="Linux-Hyper-V-Fileset"
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}
Fileset {
  Name = "Linux-Hyper-V-Fileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi: include=Linux-*"
  }
}
```

Example 3: backup any VM having «Windows» in its name, using a custom working directory

```
Job {
  Name = "Hyper-V-BackupOnlyWindowsOnF"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="WindowsOnF-Hyper-V-Fileset"
  Storage = File
  Messages = Standard
  Pool = Default
  Accurate = yes
}
Fileset {
  Name = "WindowsOnF-Hyper-VFileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
      compression=GZIP
    }
    Plugin = "hyperv-winapi: include=\"*Windows*\" tmp_dir=\"F:/backup\""
  }
}
```

Backup

If you choose not to utilize the **cluster_mode** option in your filesets for a Failover Cluster configuration, your backups will be limited to the VMs on the node where the File Daemon is installed, as long as they match your include/exclude settings. However, you have the option to install a file daemon on each node. On the other hand, if you do specify **cluster_mode**, the **Hyper-V WinAPI Plugin** will identify all VMs that meet your include/exclude settings across the entire cluster, including remote nodes that are accessible. It will then attempt to migrate remote VMs locally before performing the backup. Once the backup is complete, the VM will be automatically migrated back to its original node.

The files backed up from the Hyper-V server will be visible in a **bconsole** or will have the prefix `/@HYPERV-WINAPI/`.

Typically, a VM backup data is organized as follows:

```
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9-test1
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.out
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.bhd
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE00.bmp
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.out
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.bhd
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/IDE01.bmp
/@HYPERV-WINAPI/0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/test1.bvm
```

Where:

- 0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9 is the VM UID of the "test1"

- 0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9-test1 is a convenience empty file reminding us that the content of the VM named test1 is saved into folder /0bc0f01d-b3e5-4b13-b0e3-bf5490c828b9/
- each drive is named after its controller type (IDE, SCSI), its controller number and its location in the controller. IDE00 meaning: IDE controller number 0, location 0.
- each .bhd file contains attributes information on the drive
- each .out file is the actual content of the drive
- each .bmp file gives information on how the content of the .out file maps onto the original drive and is needed for Single Item Restore
- the .bvm file backups information on the virtual machine itself

Incremental-Differential backups: the **Hyper-V WinAPI Plugin** will automatically follow the backup level strategy as scheduled in Bacula.

Consistency Level: A backup can fail when the option “Application Consistent” is required for a VM that doesn’t support it.

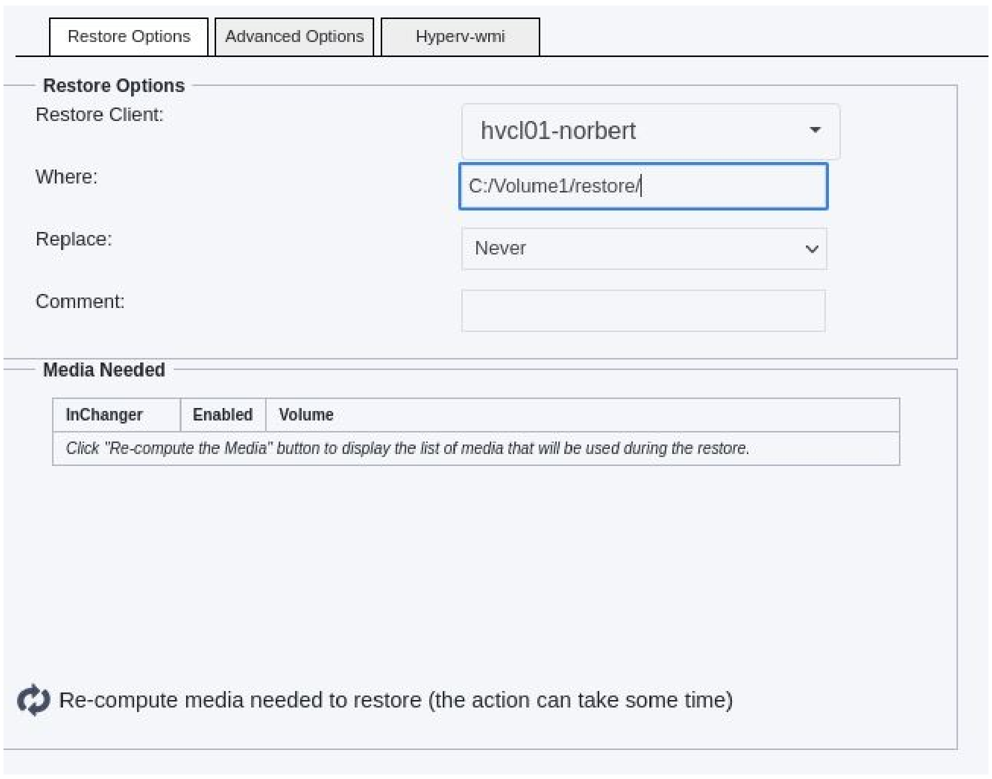
- If an Application Consistent backup fails, the **Hyper-V WinAPI Plugin** will change automatically the Consistency Level to “Crash Consistent” and retry.
- Only if `allow_pre_save` is enabled, when a “Crash Consistent” backup fails, the **Hyper-V WinAPI Plugin** will change the `pre_backup_action` to “Save” and `post_backup_action` to “Restart” and retry.
- Migration from a distant node, current backup checkpoint and pre_backup_action are stored into the corresponding VM notes in a proprietary format. They are removed when the backup is successful. If some interruption occurs during the backup, they can be recovered afterwards.
- If none of the above works, the backup fails with Error.
- Some reference points are maintained on Hyper-V side by Bacula, corresponding to different snapshots backed up. Bacula will prune those reference points when needed to keep at most 4.

Restore

To ensure a successful restore, it is advisable to either choose the entire fileset or restore all files associated with a particular drive, rather than cherry-picking individual files from the backup.

Restore parameters:

- `Where`: Can specify a path for VM restoration. If the content is not a path (does not contain slashes or backslashes), it’s considered to be the new VM restore name.
- `New Virtual Machine Name`: For renaming the restored VM.
- `Restore Path`: Specify the location where Snapshot files are restored.
- `Name used to process restore`: impersonation user name (see Impersonation of the **Hyper-V Winapi Plugin**).
- `Password used to process restore`: impersonation user password (see Impersonation of the **Hyper-V Winapi Plugin**).
- `Reuse the original Mac address`: Generate a new Mac address or not. Having duplicate Mac addresses on the network is not recommended.



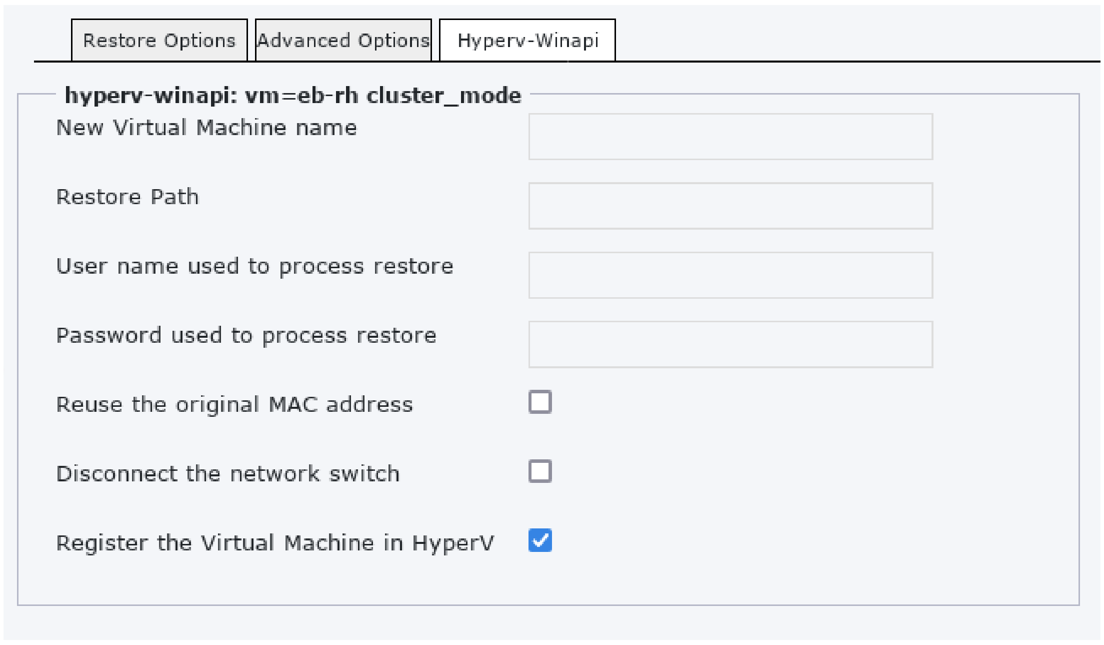
Restore Options

Restore Client:	hvcl01-norbert
Where:	C:/Volume1/restore/
Replace:	Never
Comment:	

Media Needed

InChanger	Enabled	Volume
Click "Re-compute the Media" button to display the list of media that will be used during the restore.		

Re-compute media needed to restore (the action can take some time)



hyperv-winapi: vm=eb-rh cluster_mode

New Virtual Machine name	
Restore Path	
User name used to process restore	
Password used to process restore	
Reuse the original MAC address	☐
Disconnect the network switch	☐
Register the Virtual Machine in HyperV	☑

- `Disconnect the network switch`: Reconnect the network adapter to the network switch at restore time, or choose not to.
- `Register the Virtual Machine in HyperV`: Decide whether to register the restored Virtual Machine in HyperV. It can be used to access only the settings and the raw disks.

VM Renaming:

If the 'New Virtual Machine Name' is specified, the restored VM(s) will be renamed accordingly. In case the 'Where' is not a path value, the 'Where' value will be used for renaming the restored VMs. If neither is set, the restored VM(s) will retain their original name(s), with the restore job name in parenthesis. For instance, a VM named 'tiny-vm' restored by Job 'RestoreFiles.2023-10-18_14.39.28_32' will be named 'tiny-vm (from RestoreFiles.2023-10-18_14.39.28_32)' by default.

Restore Path:

If the 'Restore Path' is specified, it will be utilized as the restore path for all backup files related to the drive. However, if the 'Restore Path' is empty and the 'Where' path is set with a value, then the 'Where' path will be used instead. In the event that neither of the aforementioned conditions are met, the default locations on the hyper-v host will be used, without creating a specific folder named after the Bacula job name. It is important to note that during the restore process, the restore files will not be moved, therefore the Restore Path will serve as the location for the restored VM. In order to facilitate multiple restorations, the files will be restored in a designated folder named after the Bacula job name.

Boot Order:

The original boot order is restored automatically for generation 1 and 2 VMs. For generation 2 VMs, UEFI boot devices will not be visible in the Firmware list before the new VM is booted once.

Examples:

- Defaults:
  - `Where`: Empty
  - `New Virtual Machine Name`: Empty
  - `Restore Path`: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs location, with original name followed by something like (from *RestoreFiles.<date>_<time>*).

- Quick Rename:
  - `Where`: newVMName
  - `New Virtual Machine Name`: Empty
  - `Restore Path`: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs location and renamed newVM-Name.

- Large Restore:
  - `Where`: Empty

- New Virtual Machine Name: NewVMName
- Restore Path: C:\LargeStorage\restore

The restored VM(s) are (re)created and renamed NewVMName. Virtual drive(s) and VM files are located into a folder named after the JobName in C:\LargeStorage\restore. Something like : C:\LargeStorage\restore\RestoreFiles.<date>_<time>.

A typical restore console output will look as follows:

```
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Start Restore Job␣
↪RestoreFiles.2023-10-18_14.39.28_32
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Restoring files from␣
↪JobId(s) 21364
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Connected to Storage
↪"File1" at 10.0.98.5:9103 with TLS
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Using Device "FileChgr1-
↪Dev1" to read.
2023-10-18 14:39:30 bp-vsir-bweb102-dir JobId 21365: Connected to Client
↪"hvcl02-winapi" at 10.0.97.22:9102 with TLS
2023-10-18 14:39:03 hvcl02-winapi JobId 21365: Connected to Storage at 10.0.
↪98.5:9103 with TLS
2023-10-18 14:39:30 bp-vsir-bweb102-sd JobId 21365: Ready to read from volume
↪"Vol-3263" on File device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:03 hvcl02-winapi JobId 21365: StartRestoreJob: Restoring vm␣
↪in c:\Volume1\restore\RestoreFiles.2023-10-18_14.39.28_32\
2023-10-18 14:39:30 bp-vsir-bweb102-sd JobId 21365: Forward spacing Volume
↪"Vol-3263" to addr=3877321766
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: End of Volume "Vol-3263"␣
↪at addr=5368645668 on device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: Ready to read from volume
↪"Vol-0396" on File device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:47 bp-vsir-bweb102-sd JobId 21365: Forward spacing Volume
↪"Vol-0396" to addr=271
2023-10-18 14:39:56 bp-vsir-bweb102-sd JobId 21365: End of Volume "Vol-0396"␣
↪at addr=1732770639 on device "FileChgr1-Dev1" (/bck/a1).
2023-10-18 14:39:56 bp-vsir-bweb102-sd JobId 21365: Elapsed time=00:00:26,␣
↪Transfer rate=123.8 M Bytes/second
2023-10-18 14:40:06 bp-vsir-bweb102-dir JobId 21365: Bacula Enterprise bp-
↪vsir-bweb102-dir 16.0.7 (11Jul23):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Core)
  JobId:                  21365
  Job:                    RestoreFiles.2023-10-18_14.39.28_32
  Restore Client:         "hvcl02-winapi" 16.0.7 (05Oct23) Windows Server␣
↪2019 Standard ServerStandard (build 17763), 64-bit,Cross-compile,Win64
  Where:                  c:\Volume1\restore\
  Replace:                Never
  Start time:             18-Oct-2023 14:39:30
  End time:               18-Oct-2023 14:40:06
  Elapsed time:           36 secs
  Files Expected:         5
  Files Restored:         5
  Bytes Restored:         3,221,302,956 (3.221 GB)
```

(continues on next page)

(continued from previous page)

```
Rate:                   89480.6 KB/s
FD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            Restore OK
```

Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a "multi-VM" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called "abort_on_error" in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 configuration (one VM backed up per job) means that the "abort_on_error" option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.
- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up "serially", one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

Failover Cluster

Backup remains seamless in a Failover Cluster setup, regardless of the hosting node for the VM(s) during backup. As long as the user possesses the appropriate credentials on all nodes, the VMs within the cluster will undergo filtering via include/exclude criteria. Remote VMs will be automatically **Migrated** to the local node, backed up, then restored to it's original node. Hyper-V does not permit VM migration between nodes during backup operations.

Recovery

If for any reason (crash, etc.) a backup is interrupted, some actions can be recovered afterwards by calling the hyperv-recovery.ps1 script with the recovered VM name has parameter:

```
C:\Program Files\Bacula\plugins>powershell -file hyperv-recovery.ps1 my_VM_
↪name
```

so the VM is restored to its pre-backup state: The backup checkpoint is cleaned, the VM state is restored and if the VM has been migrated pre-backup, it's moved back to it's original cluster node.

Hyper-V VSS Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Overview

This white paper presents how to use the Microsoft Hyper-V Server plugin when backing up with **Bacula Enterprise** version 8.2. These solutions are not applicable to prior versions. This document is intended to be used by **Bacula Enterprise** administrators.

Bacula Windows Hyper-V Plugin

Bacula Systems provides a single plugin for Bacula Enterprise named `vss-fd.dll` that permits you to backup a number of different components on Windows machines. One of those components is Microsoft Hyper-V Server, which is the subject of this white paper.

Backing up and restoring Hyper-V virtual machine is supported with Full level backups. It is not possible to do Incremental or Differential backups because Microsoft does not support that backup level for the Hyper-V Server product. Use of the Global Endpoint Deduplication plugin and the `bothsides` Fileset option permits to minimize the data transfer and the storage.

Installation and Configuration

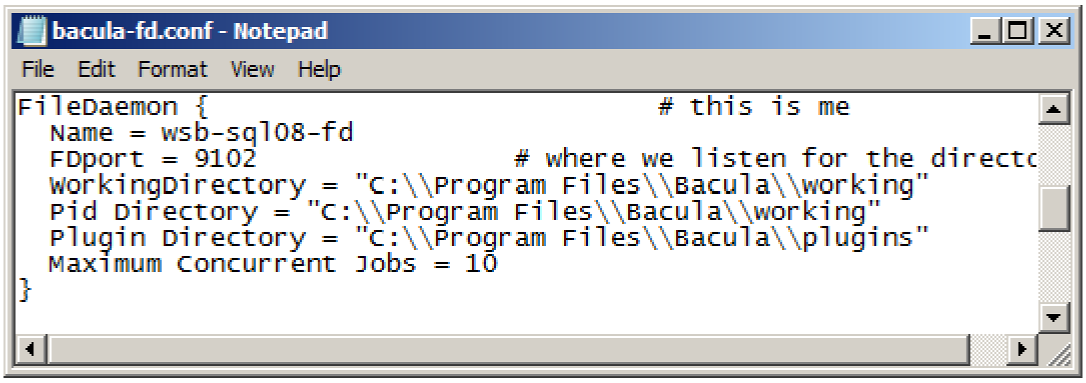
To activate the Hyper-V component you have to put the following into the Include section of the File Set which will be used to back up the Hyper-V Server:

```
Plugin = "vss:/@HYPERV/"
```

This will back up all Hyper-V Virtual Machine. The plugin directive must be specified exactly as shown above. A Job may have one or more of the **vss** plugin components specified.

You must ensure that the `vss-fd.dll` plugin is in the plugins directory on the FD doing the backup (done by default with the installer), and that the `Plugin Directory` directive line is present and enabled in the FD's configuration file `bacula-fd.conf`.

An example of the FD configuration file is shown in the screenshot below:



```
FileDaemon {                          # this is me
  Name = wsb-sql08-fd
  FDport = 9102                  # where we listen for the directo
  WorkingDirectory = "C:\\Program Files\\Bacula\\working"
  Pid Directory = "C:\\Program Files\\Bacula\\working"
  Plugin Directory = "C:\\Program Files\\Bacula\\plugins"
  Maximum Concurrent Jobs = 10
}
```

Fig. 12: File Daemon Configuration Excerpt with "Plugin Directory" Line

The status output of a client with the VSS plugin enabled:

```
*status client=wsb-sql08-fd
Connecting to Client wsb-sql08-fd at wsb-sql08:9102

wsb-sql08-fd Version: 8.2.0 (02 Feb 2015)  VSS Linux Cross-compile Win64
Daemon started 20-Apr-12 13:14. Jobs: run=15 running=0.
Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1 (build 7601),
↪ 64-bit
 Heap: heap=0 smbytes=1,061,455 ...
 Sizes: boffset_t=8 size_t=8 debug=0 ...
 Plugin: vss-fd.dll
```

Backup

If everything is set up correctly as above then the backup will include the Hyper-V server data. The Hyper-V server data files backed up will appear in a **bconsole** or **bat** restore like:

```
/@HYPERV/
...
etc
```

A complete example of a Fileset and Job resource for Hyper-V Server data is shown below. As for all VSS-enabled components, it is the administrator's responsibility to make sure that the required VSS snapshots are created by explicitly mentioning at least one file or directory for each drive where data that is handled by the plugin is stored. In the example, we use the file `c:/backmeup` to ensure this.

```
Fileset {
  Name = HYPERV
  Include {
    Options {
      Signature = SHA1
      Dedup = bothsides
    }
    File = C:/backmeup
    Plugin = "vss:/@HYPERV/"
  }
}

Job {
  Name = HYPERV08
  Accurate = Yes
  File Set = HYPERV
  Client = wsb-hyp08-fd
  Job Defs = DefaultJob
  Level = Full
}
```

Note in the example above that `C:/backmeup` is explicitly included, which is required to ensure that **Bacula** creates the required VSS snapshot of that Windows drive letter. If Hyper-V Server data is also stored on other partitions, you need to create similar `File` =-lines for these drives, too.

Note: Starting with Bacula Enterprise version 12.6, the explicit include of a dummy file (see `File = C:/backmeup` in the fileset example above) is not mandatory anymore

---

```
File Set {
 Name = HYPERV-TestVM
 Include {
  Options {
    Signature = SHA1
  }
 File = C:/backmeup
 # backup only TestVM on the server
 Plugin = "vss:/@HYPERV/ cinclude=\"Host Component\" cinclude=*/TestVM␣
↪cexclude=*"
```

(continues on next page)

(continued from previous page)

```
 }
}
```

Hyper-V uses one of two mechanisms to back up each VM. The default backup mechanism is called the “Saved State” or “Offline” method, where the VM is put into a saved state during the processing of the PrepareForSnapshot event, snapshots are taken of the appropriate volumes, and the VM is returned to the previous state during the processing of the PostSnapshot event.

The other backup mechanism is called the “Child VM Snapshot” or “Online” method, which uses VSS inside the child VM to participate in the backup. For the “Child VM Snapshot” method to be supported, all of the following conditions must be met:

- Backup (volume snapshot) Integration Service is installed and running in the child VM. The service name is “Hyper-V Volume Shadow Copy Requestor”.
- The child VM must be in the running state.
- The Snapshot File Location for the VM is set to be the same volume in the host operating system as the VHD files for the VM.
- All volumes in the child VM are basic disks and there are no dynamic disks.
- All disks in the child VM must use a file system that supports snapshots (for example, NTFS).

To know if your VMs are “Offline” or “Online”, it is possible to use the following windows command on Windows 2012 R2:

```
C:/> echo list writers > t.txt
C:/> diskshadow /s t.txt | find "Caption: O"
                        - Caption: Offline/2012
                        - Caption: Offline/windows
                        - Caption: Online/centos
```

On Windows 2012 and 2008

```
C:/> echo list writers > t.txt
C:/> diskshadow /s t.txt | find /i "Caption: Backup Using"
```

- For Offline backups: Backup Using Saved State/*VMname1*
- For Online backups: Backup Using Child Partition Snapshot/*VMname2*

Restore

Restoring the VMs is done entirely by the host operating system; the VSS writers in the child VMs are not involved.

- During the processing of the PreRestore event, the Hyper-V VSS writer turns off and deletes any VMs that are about to be restored.
- After all VSS writers have processed the PreRestore event, the files are restored.
- For each VM that was restored, the Hyper-V VSS writer registers the VM with the Hyper-V management service. If the VM is restored to a nondefault location, a symbolic link is created in the default location linking to that location.
- For each VHD that was restored, the location is compared with the one specified for that VM. If the location is different, then the configuration is updated with the proper location.

- The network configuration is updated. If the virtual switches that the VM was connected to when it was backed up still exit, new ports are created and connected to the VM.

When restoring a "Offline" VM, the VM will not be re-created by Microsoft Hyper-V vss driver. It is possible to run "New-VM" powershell command to re-create the VM.

```
New-VM -VMName centos -VHDPath C:/VM/centos.vhdx -MemoryStartupBytes 512MB -
↪SwitchName VMNetwork
```

without_vss

With Bacula Enteprise 8.2, it is possible to restore VSS files directly on disk without using the VSS restore framework. In the restore menu, it is possible to configure `Plugin Options` menu and set the `without_vss` option to "true".

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/trusty-amd64-dir.restore.9.bsr
Where:           c:/tmp
Replace:         Always
Fileset:         Full Set
Backup Client:   hyperv
Restore Client:  hyperv
Storage:         dedup
When:            2015-03-03 06:50:22
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*         <------------- Plugin Options menu
```

Example

We assume that a correct backup of Hyper-V data exists and you start the restore with option 5 of the `bconsole` **restore** command, mark the complete tree of data backed up by the Hyper-V component of the VSS plugin, then finally do `lsmark @HYPERV` to show all the files selected to be restored:

```
$ mark *
31 files marked.
$ lsmark
*@HYPERV/
  *Microsoft Hyper-V VSS Writer/
    *Host Component/
      *:component_info_5215da3c
      *c:/
        *programdata/
          *microsoft/
            *windows/
              *hyper-v/
                *initialstore.xml
                *resource types/
                  *06ff76fa-2d58-4baf-9f8d-455773824f37.xml
                  *118c3be5-0d31-4804-85f0-5c6074abea8f.xml
```

(continues on next page)

(continued from previous page)

```
                *146c56a0-3546-469b-9737-fcbcf82428f4.xml
                *dacdcf3f-6f67-4eb8-a4d0-5d93b48a2468.xml
                *f6293891-f32f-4930-b2db-1a8961d9cb75.xml
  *Offline/
    *ubuntu/
      *:component_info_5215da3c
      *c:/
        *programdata/
          *microsoft/
            *windows/
              *hyper-v/
                *virtual machines/
                  *690f5094-ff23-411e-92c0-639fc7ebc598/
                    *690f5094-ff23-411e-92c0-639fc7ebc598.bin
                    *690f5094-ff23-411e-92c0-639fc7ebc598.vsv
                  *690f5094-ff23-411e-92c0-639fc7ebc598.xml
        *vm/
          *ubuntu.vhdx
```

```
$ lsmark
*@HYPERV/
  *Microsoft Hyper-V VSS Writer/
    *Host Component/
      *:component_info_5216cf46
      *c:/
        *programdata/
          *microsoft/
            *windows/
              *hyper-v/
                *initialstore.xml
                *resource types/
                  *06ff76fa-2d58-4baf-9f8d-455773824f37.xml
                  *118c3be5-0d31-4804-85f0-5c6074abea8f.xml
    *Online/
      *centos/
        *:component_info_5216cf46
        *c:/
          *programdata/
            *microsoft/
              *windows/
                *hyper-v/
                  *snapshots/
                    *acc145fb-9566-402d-9434-04f1e325a75f-backupsnapshot.xml
                  *virtual machines/
                    *acc145fb-9566-402d-9434-04f1e325a75f.xml
          *vm/
            *centos-childvhd.avhdx
            *centos.vhdx
```

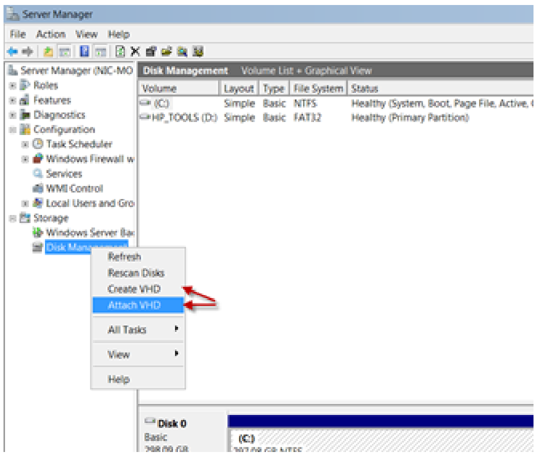
File Level Restore

To restore a set of files from a Hyper-V VM backup without re-importing the entire VM, it is possible to restore VHD files in a directory using the `without_vss` plugin restore option (See sec *without_vss*) and mount them in the system with the Powershell command Mount-VHD (or the Server Manager console (see Fig *Attach/Mount Option in Server Manager* below). Once mounted, the VHD image is accessible like other physical disks on the system.

```
Mount-VHD -Path c:\test\testvhdx.vhdx -ReadOnly
```

More information about the Mount-VHD command can be found here:

https://technet.microsoft.com/en-us/library/hh848551.aspx



Volume	Layout	Type	File System	Status
(C:)	Simple	Basic	NTFS	Healthy (System, Boot, Page File, Active,
HP_TOOLS (D:)	Simple	Basic	FAT32	Healthy (Primary Partition)

Fig. 13: Attach/Mount Option in Server Manager

We advise to restore VHD files on a different system to avoid operational problems during the restore. If the `without_vss` option is not properly set, the original VM would be deleted by Hyper-V automatically during the restore.

Cluster Shared Volumes

Starting with Bacula Enterprise 12.6, Cluster Shared Volumes File System (CSVFS) backup is supported, so VMs located on CSVFS volumes are backuped transparently. However, mixing CSVFS and NTFS in the same backup is not supported due to a Microsoft limitation. Noticeably, Host Component are located on C:/ which is typically a NTFS volume.

- Make sure to backup this kind of system with 2 different jobs:
- One job backups the default selection without the Host Component:

```
File Set {
 Name = HYPERV-CSVFS-1
 Include {
  Options {
    Signature = SHA1
  }
 # backup all defaults but Host Component
```

(continues on next page)

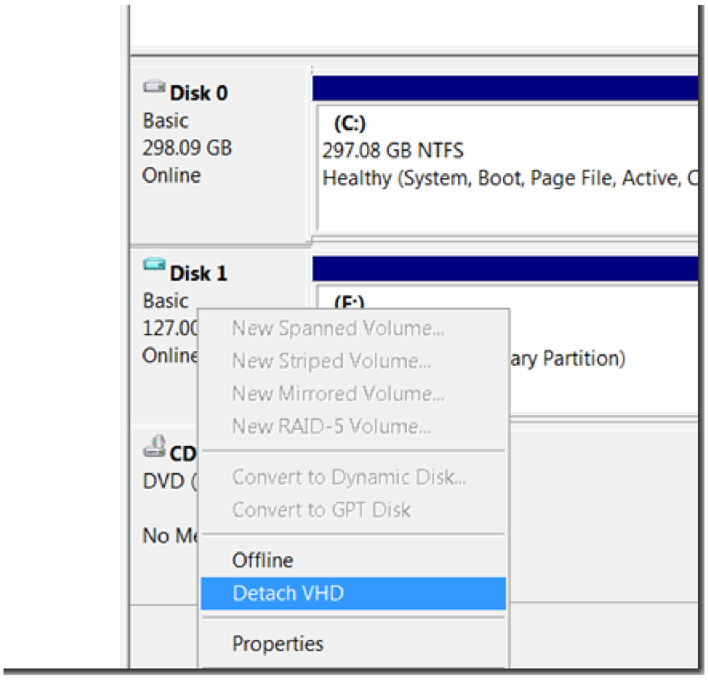


Fig. 14: Detach/Unmount Option in Server Manager

(continued from previous page)

```
  Plugin = "vss:/@HYPERV/ cexclude=\"Host Component\""
  }
}

Job {
Name = HYPERV09-CSVFS-NO-HOSTCOMPONENT
Accurate = Yes
File Set = HYPERV-CSVFS-1
Client = wsb-hyp09-fd
Job Defs = DefaultJob
Level = Full
}
```

- Another job backups specifically the Host Component:

```
File Set {
  Name = HYPERV-CSVFS-2
  Include {
   Options {
      Signature = SHA1
   }
  # backup only Host Component
  Plugin = "vss:/@HYPERV/ cinclude=\"Host Component\" cexclude=*"
  }
}

Job {
```

(continues on next page)

(continued from previous page)

```
Name = HYPERV09-CSVFS-HOSTCOMPONENT
Accurate = Yes
File Set = HYPERV-CSVFS-2
Client = wsb-hyp09-fd
Job Defs = DefaultJob
Level = Full
}
```

Single Item Restore

Single Item Restore

(see separate article about Hyper-V Single Item Restore that you can download on the top of the page of that article)

Plugin Notes

Windows VSS Plugin Items to Note

- One file from each drive needed by the plugins must be explicitly listed in File Set used. This is to ensure that the main **Bacula** code does a snapshot of all the required drives. At a later time, we will find a way to accomplish this automatically.
- When doing a backup that is to be used for Bare Metal Recovery, do **not** use the VSS plugin.

General Plugin Items to Note

- The `estimate` command does not handle plugins. When estimating a job that uses plugins, an error message regarding the plugin will be displayed. However, backup jobs will use the plugin.
- The File Set Include Option `CheckFileChanges = Yes` does not work with plugin-generated data. Thus, you must not use that Option in the Include section of the Fileset where you specify using the Hyper-V plugin.
- When an Offline virtual machine is currently backed up, it is not possible to start it (Hyper-V limitation).
- The Bacula `replace` flag is not respected by the Hyper-V plugin. Virtual machines will be always overwritten during restore.
- Microsoft Hyper-V vss interface doesn't support Differential and Incremental backup. Use of the Global Endpoint Deduplication plugin and the `bothsides` Fileset option permits to minimize the data transfer and the storage.
- When trying to restore Incremental or Differential jobs, Hyper-V VSS writer will print errors on PostRestore and PreRestore events. The virtual disk image (VHDX) should be restored despite these errors.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a "multi-VM" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called "abort_on_error" in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 configuration (one VM backed up per job) means that the "abort_on_error" option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.
- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up "serially", one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

Hyper-V WMI Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Features summary

- Quiescing VSS-based applications can be achieved through VSS-based guest snapshots.
- Microsoft's RCT technology enables Full, Differential, and Incremental image-level backups for virtual machines.
- Complete virtual machine images can be restored effortlessly.

Important notes

- Backups made with the Hyper-V WMI plugin cannot be used with Virtual Full jobs. It is not recommended to mix these backup methods as it may result in difficulties when restoring jobs from Virtual Full backups.
- Single Item Restore is not supported.
- Linux virtual machines cannot be backed up live at Application Consistency level.
- The **Hyper-V WMI Plugin** requires Hyper-V Virtual Machines version 6.2 or above to manage Differential and Incremental backups.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Supported platforms

This documentation presents solutions for **Bacula Enterprise** 16.0.0 and higher, and is not applicable to prior versions of Bacula.

This plugin supports Windows 8, Windows Server 2012 and later.

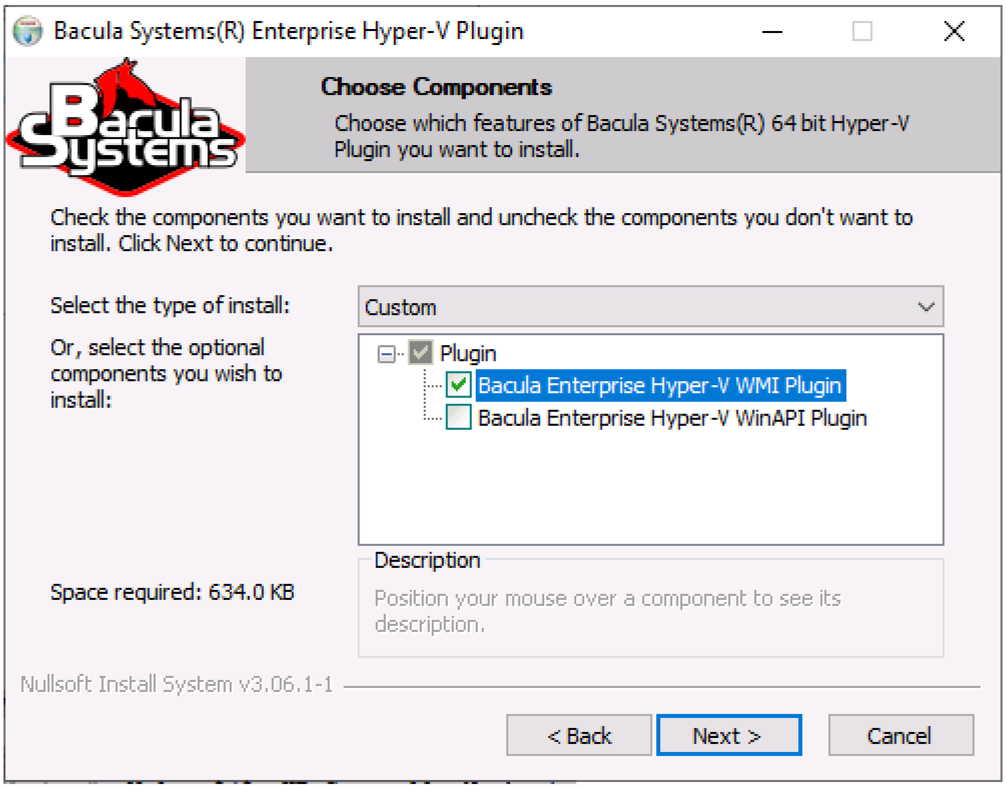
Installation

The Bacula File Daemon and the **Hyper-V WMI Plugin** need to be installed on the Hyper-V host server. The **Hyper-V WMI Plugin** Windows installer is the same as the **Hyper-V Winapi Plugin** installer.

You can choose to install one or the other from the Plugin tree.

It will deploy required components within the Bacula File Daemon plugins directory.

To configure the Bacula File Daemon, refer to the general Bacula installation documentation.


Bacula Systems(R) Enterprise Hyper-V Plugin

Choose Components

Choose which features of Bacula Systems(R) 64 bit Hyper-V Plugin you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Custom

Or, select the optional components you wish to install:

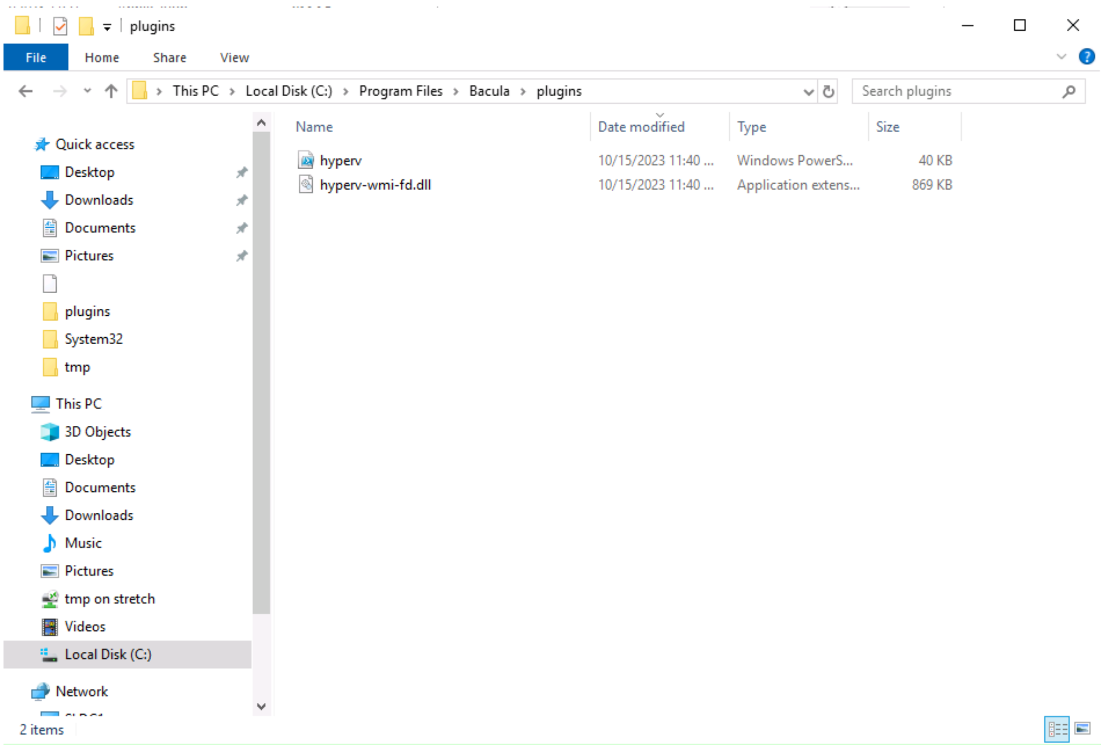
- [✓] Plugin
  - [✓] Bacula Enterprise Hyper-V WMI Plugin
  - [ ] Bacula Enterprise Hyper-V WinAPI Plugin

Space required: 634.0 KB

Description

Position your mouse over a component to see its description.

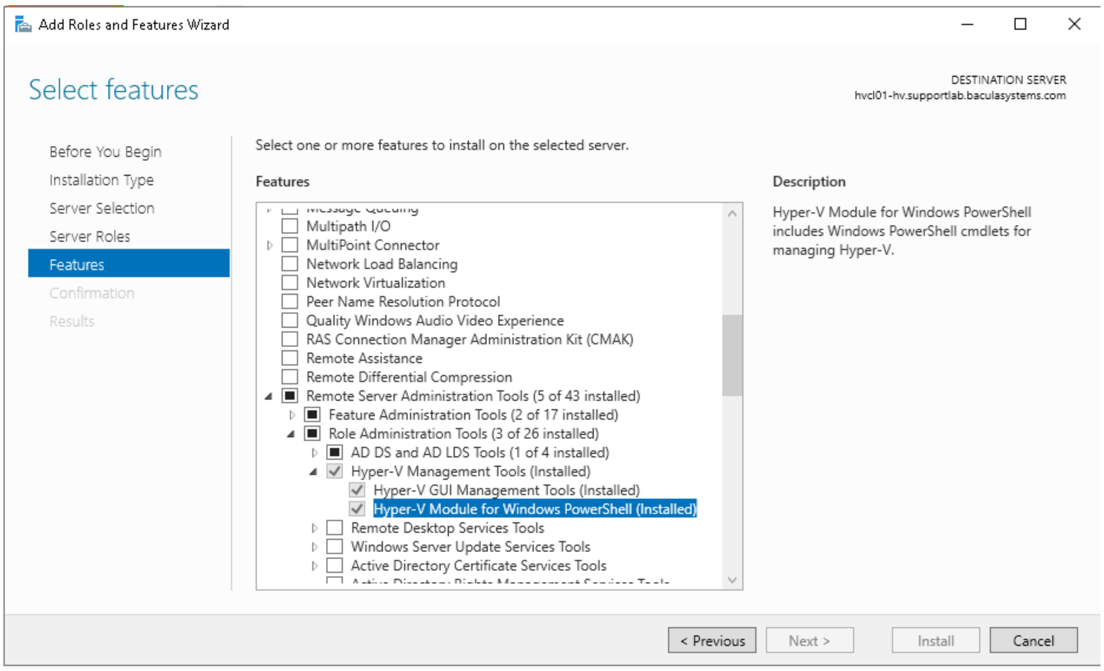
Nullsoft Install System v3.06.1-1


This PC > Local Disk (C:) > Program Files > Bacula > plugins

Name	Date modified	Type	Size
hyperv	10/15/2023 11:40 ...	Windows PowerS...	40 KB
hyperv-wmi-fd.dll	10/15/2023 11:40 ...	Application extens...	869 KB

2 items

On the server side, the *Hyper-V PowerShell Module* needs to be enabled. On Windows Server or Hyper-V server 2012, 2016 and 2019, use Server Manager to install it. It should be located under Remote Server Administration Tools -> Role Administration Tools -> Hyper-V Management Tools and check Hyper-V Module for Windows PowerShell.



Select one or more features to install on the selected server.

- ☐ Multipath I/O
- ☐ MultiPoint Connector
- ☐ Network Load Balancing
- ☐ Network Virtualization
- ☐ Peer Name Resolution Protocol
- ☐ Quality Windows Audio Video Experience
- ☐ RAS Connection Manager Administration Kit (CMAK)
- ☐ Remote Assistance
- ☐ Remote Differential Compression
- ■ Remote Server Administration Tools (5 of 43 installed)
  - ■ Feature Administration Tools (2 of 17 installed)
  - ■ Role Administration Tools (3 of 26 installed)
    - ■ AD DS and AD LDS Tools (1 of 4 installed)
    - ✓ Hyper-V Management Tools (Installed)
      - ✓ Hyper-V GUI Management Tools (Installed)
      - ✓ Hyper-V Module for Windows PowerShell (Installed)
    - ☐ Remote Desktop Services Tools
    - ☐ Windows Server Update Services Tools
    - ☐ Active Directory Certificate Services Tools

Hyper-V Module for Windows PowerShell includes Windows PowerShell cmdlets for managing Hyper-V.

Verify the correct installation of the FD and the **Hyper-V WMI Plugin** by running status client from bconsole or from BWeb.

```
*status client=w2019-hv01-fd
Connecting to Client w2019-hv01-fd at 172.22.22.50:9102
w2019-hv01-fd Version: 12.8.0 (06 April 2021) VSS Linux Cross-compile Win64
Daemon started 19-Jun-21 16:31. Jobs: run=23 running=2.
Microsoft Windows 2012 Standard Edition (build 9200), 64-bit
Priv 0x73f
Memory: WorkingSetSize: 34,168,832 QuotaPagedPoolUsage: 183,768␣
↪QuotaNonPagedPoolUsage: 17,368 PagefileUsage:
43,687,936
APIs=OPT,ATP,LPV,CFA,CFW,
WUL,WMKD,GFAA,GFAW,GFAEA,GFAEW,SFAA,SFAW,BR,BW,SPSP,
WC2MB,MB2WC,FFFA,FFFW,FNFA,FNFW,SCDA,SCDW,
GCDA,GCDW,GVPNW,GVNFVMPW,LZO,EFS
Heap: heap=34,168,832 smbytes=39,489,074 max_bytes=69,259,353 bufs=395 max_
↪bufs=396
Sizes: boffset_t=8 size_t=8 debug=10 trace=1 mode=0,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL
APIs: !GPFS
Plugin: alldrives-fd.dll(1.2) hyperv-wmi-fd.dll(0.1) winbmr-fd.dll(3.1.0)
```

Verify that **hyperv-wmi-fd.dll** is in the “Plugin” line (last line in the above example output).

In the case of a Failover Cluster configuration, the Bacula file deamon and the **Hyper-V WMI plugin** need to be installed on only one node.

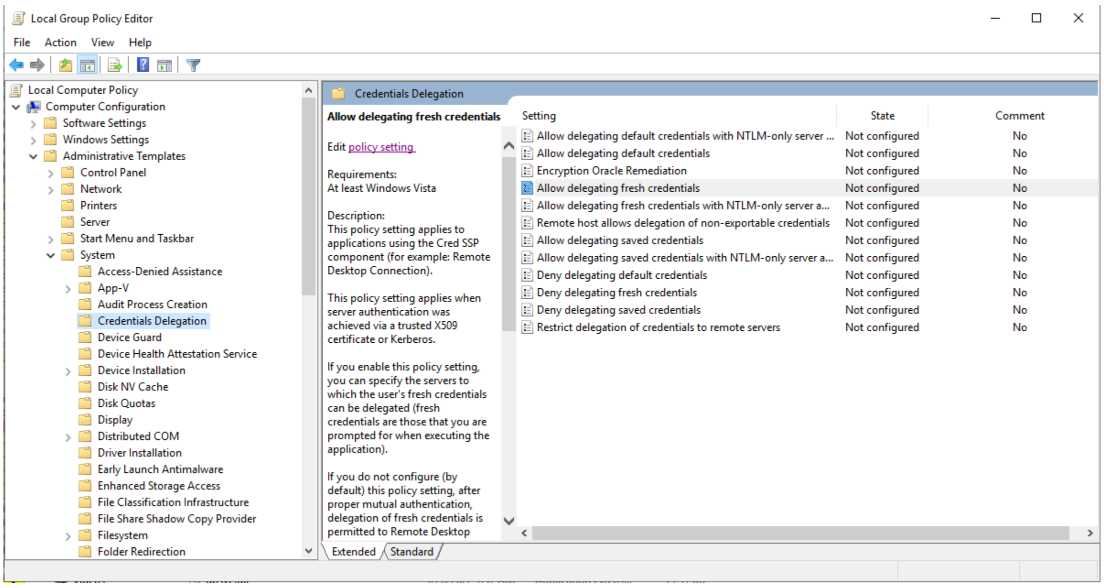
Important considerations regarding credentials settings

Important: In order to access and backup the Hyper-V server, the delegation of the User credentials must be enabled and the Bacula file daemon must be logged as an authorized user within the Hyper-V server.

Enable delegation of user credentials on the Hyper-V server

- Run gpedit.msc (normally in C:\Windows\System32) on the Hyper-V server and look at the

following policy: Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials.



Allow delegating fresh credentials

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop

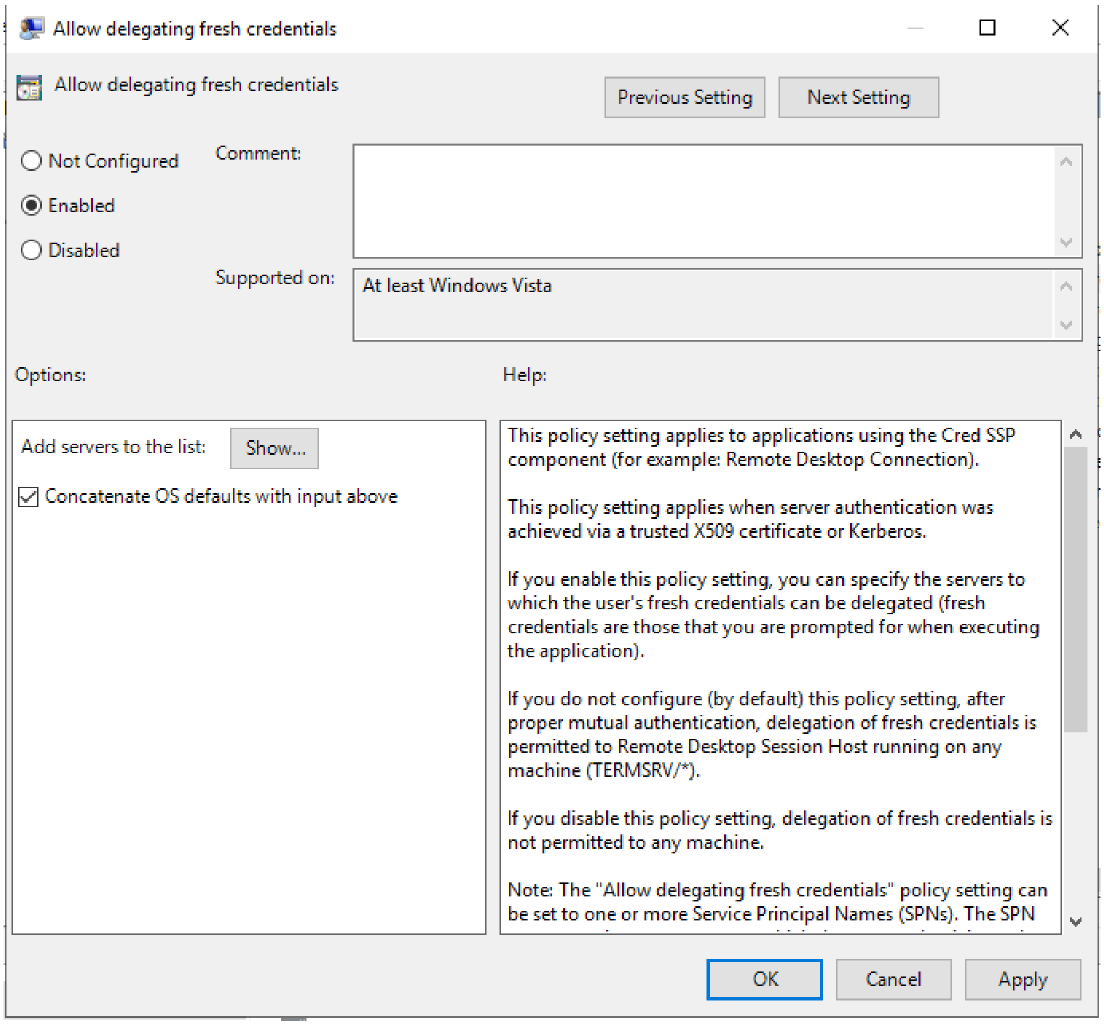
Setting	State	Comment
Allow delegating default credentials with NTLM-only server ...	Not configured	No
Allow delegating default credentials	Not configured	No
Encryption Oracle Remediation	Not configured	No
Allow delegating fresh credentials	Not configured	No
Allow delegating fresh credentials with NTLM-only server a...	Not configured	No
Remote host allows delegation of non-exportable credentials	Not configured	No
Allow delegating saved credentials	Not configured	No
Allow delegating saved credentials with NTLM-only server a...	Not configured	No
Deny delegating default credentials	Not configured	No
Deny delegating fresh credentials	Not configured	No
Deny delegating saved credentials	Not configured	No
Restrict delegation of credentials to remote servers	Not configured	No

- Verify that it is enabled and configured with the WSMAN SPN appropriate for the target computer.

For example, for a target computer name “myserver.domain.com”, the SPN can be one of the following: WSMAN//myserver.domain.com or WSMAN//*.domain.com. Introduce it in the “Add servers to the list” “Show” dialog box.

- Alternatively run a powershell console on the Hyper-V server (normally in C:\Windows\Systeme32\WindowsPowerShellv1.0powershell.exe) and enter the following commands:

```
Enable-WSManCredSSP -Role Server -Force
Enable-WSManCredSSP -Role "Client" -DelegateComputer myserver.domain.com -
↪Force
```



Allow delegating fresh credentials
Allow delegating fresh credentials
Previous Setting
Next Setting

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Windows Vista

Options:

Add servers to the list: Show...

☑ Concatenate OS defaults with input above

Help:

This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

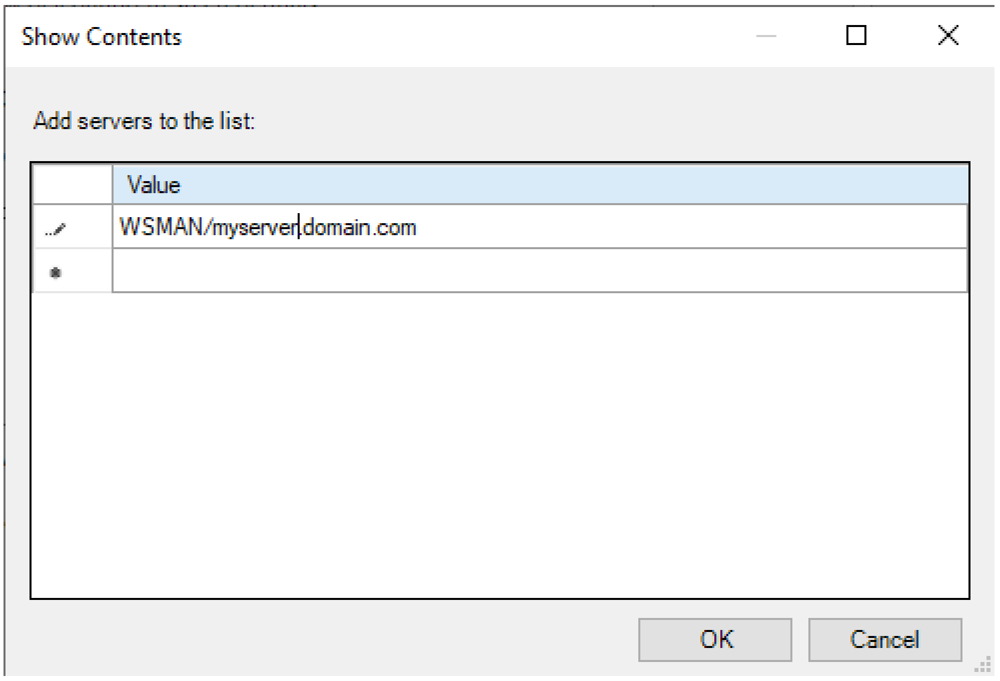
This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop Session Host running on any machine (TERMSRV/*).

If you disable this policy setting, delegation of fresh credentials is not permitted to any machine.

Note: The "Allow delegating fresh credentials" policy setting can be set to one or more Service Principal Names (SPNs). The SPN



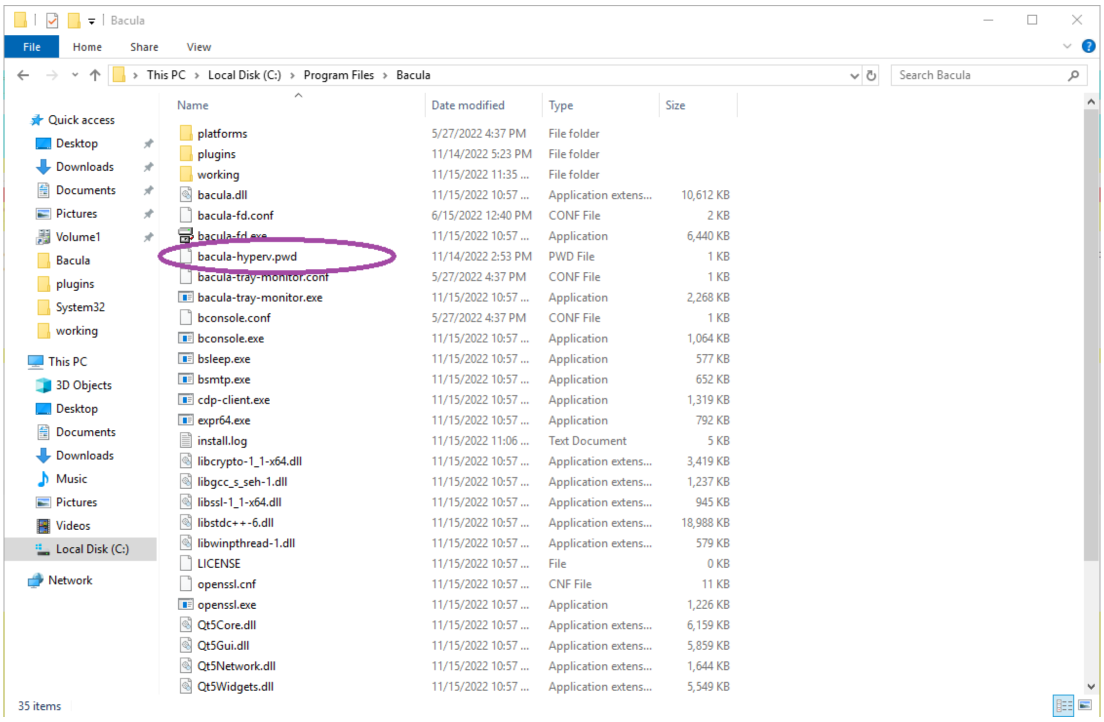
Add servers to the list:

	Value
	WSMAN/myserver.domain.com

Impersonation of Hyper-V WMI Plugin

The impersonation of the **Hyper-V WMI Plugin** can be achieved in different ways.

- Specify the user name and password locally on the hyper-v node. This is the **recommended method**. In a `bacula-hyperv.pwd` file, located by the `bacula-fd.conf` config file (typically C:\Program Files\Bacula).



Name	Date modified	Type	Size
platforms	5/27/2022 4:37 PM	File folder	
plugins	11/14/2022 5:23 PM	File folder	
working	11/15/2022 11:35 ...	File folder	
bacula.dll	11/15/2022 10:57 ...	Application extens...	10,612 KB
bacula-fd.conf	6/15/2022 12:40 PM	CONF File	2 KB
bacula-fd.exe	11/15/2022 10:57 ...	Application	6,440 KB
bacula-hyperv.pwd	11/14/2022 2:53 PM	PWD File	1 KB
bacula-tray-monitor.conf	5/27/2022 4:37 PM	CONF File	1 KB
bacula-tray-monitor.exe	11/15/2022 10:57 ...	Application	2,268 KB
bconsole.conf	5/27/2022 4:37 PM	CONF File	1 KB
bconsole.exe	11/15/2022 10:57 ...	Application	1,064 KB
bsleep.exe	11/15/2022 10:57 ...	Application	577 KB
bsmtp.exe	11/15/2022 10:57 ...	Application	652 KB
cdp-client.exe	11/15/2022 10:57 ...	Application	1,319 KB
expr64.exe	11/15/2022 10:57 ...	Application	792 KB
install.log	11/15/2022 11:06 ...	Text Document	5 KB
libcrypto-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	3,419 KB
libgcc_s_seh-1.dll	11/15/2022 10:57 ...	Application extens...	1,237 KB
libssl-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	945 KB
libstdc++-6.dll	11/15/2022 10:57 ...	Application extens...	18,988 KB
libwinpthread-1.dll	11/15/2022 10:57 ...	Application extens...	579 KB
LICENSE	11/15/2022 10:57 ...	File	0 KB
openssl.cnf	11/15/2022 10:57 ...	CNF File	11 KB
openssl.exe	11/15/2022 10:57 ...	Application	1,226 KB
Qt5Core.dll	11/15/2022 10:57 ...	Application extens...	6,159 KB
Qt5Gui.dll	11/15/2022 10:57 ...	Application extens...	5,859 KB
Qt5Network.dll	11/15/2022 10:57 ...	Application extens...	1,644 KB
Qt5Widgets.dll	11/15/2022 10:57 ...	Application extens...	5,549 KB

  `bacula-hyperv.pwd` contains the user name followed by the user password, separated by a colon.

```
name@domain.com:mypassword
```

  or

```
DOMAIN\name:mypassword
```

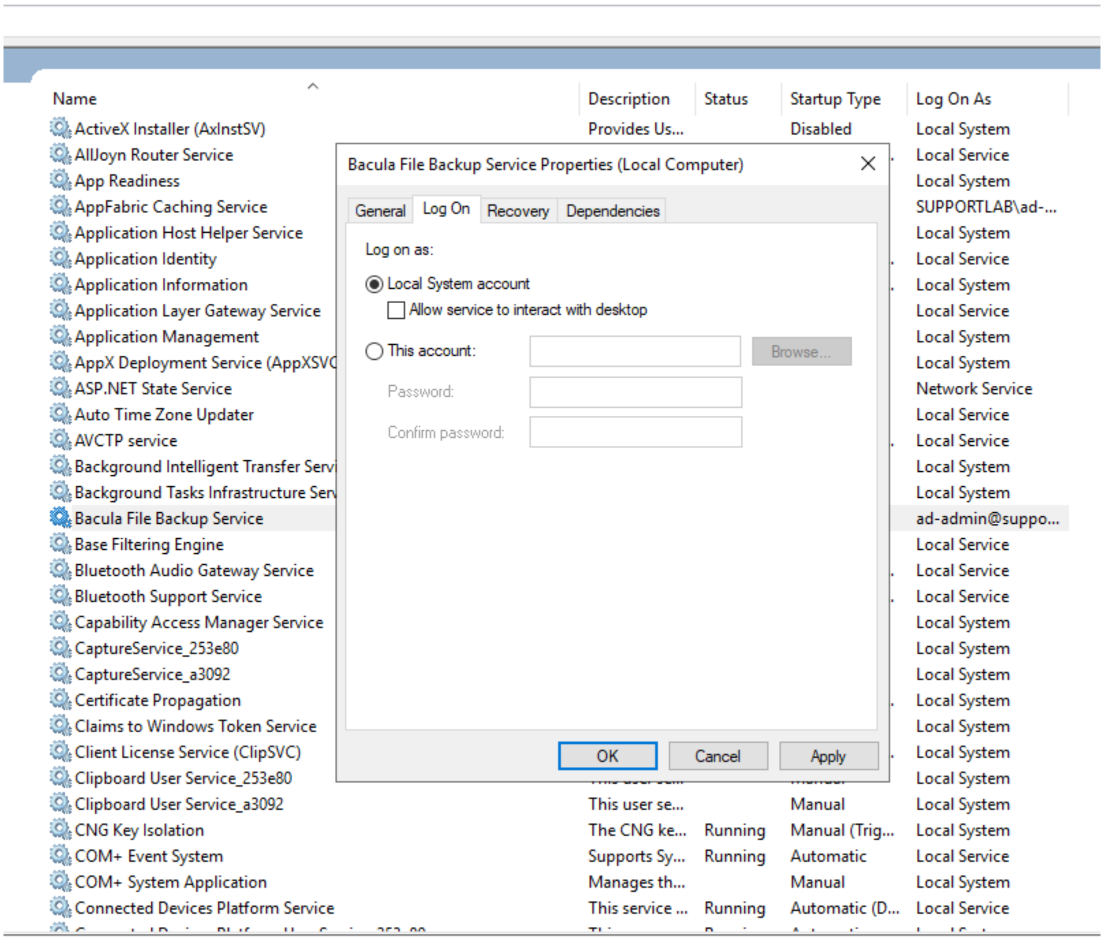
- Impersonate the **Hyper-V WMI Plugin** by passing user and password, as plugin options See Job configuration `user_name` and `user_password` options.

- Manually change the Bacula file daemon default login account:

  Access the Hyper-V server using administrative credentials. Go to the Windows Start menu, enter “Services”, and press Enter to display a list of all installed services. Locate the Bacula File Backup Service, right-click on it, and then select Properties. Navigate to the Log On tab, where the settings should appear as follows:

  Toggle the selection from “Local System account” to “This account”. Enter the credentials of a Hyper-V administrator (either read only or read-write). Click OK.

  Click on the Bacula File Backup Service entry once more with the right mouse button, then select “Restart” to ensure that the changes take effect.



Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local System
AllJoyn Router Service				Local Service
App Readiness				Local System
AppFabric Caching Service				SUPPORTLAB\ad-...
Application Host Helper Service				Local System
Application Identity				Local Service
Application Information				Local System
Application Layer Gateway Service				Local Service
Application Management				Local System
AppX Deployment Service (AppXSVC				Local System
ASP.NET State Service				Network Service
Auto Time Zone Updater				Local Service
AVCTP service				Local Service
Background Intelligent Transfer Servi				Local System
Background Tasks Infrastructure Serv				Local System
Bacula File Backup Service				ad-admin@suppo...
Base Filtering Engine				Local Service
Bluetooth Audio Gateway Service				Local Service
Bluetooth Support Service				Local Service
Capability Access Manager Service				Local System
CaptureService_253e80				Local System
CaptureService_a3092				Local System
Certificate Propagation				Local System
Claims to Windows Token Service				Local System
Client License Service (ClipSVC)				Local System
Clipboard User Service_253e80				Local System
Clipboard User Service_a3092	This user se...		Manual	Local System
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local System
COM+ Event System	Supports Sy...	Running	Automatic	Local Service
COM+ System Application	Manages th...		Manual	Local System
Connected Devices Platform Service	This service ...	Running	Automatic (D...	Local Service

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

(●) Local System account

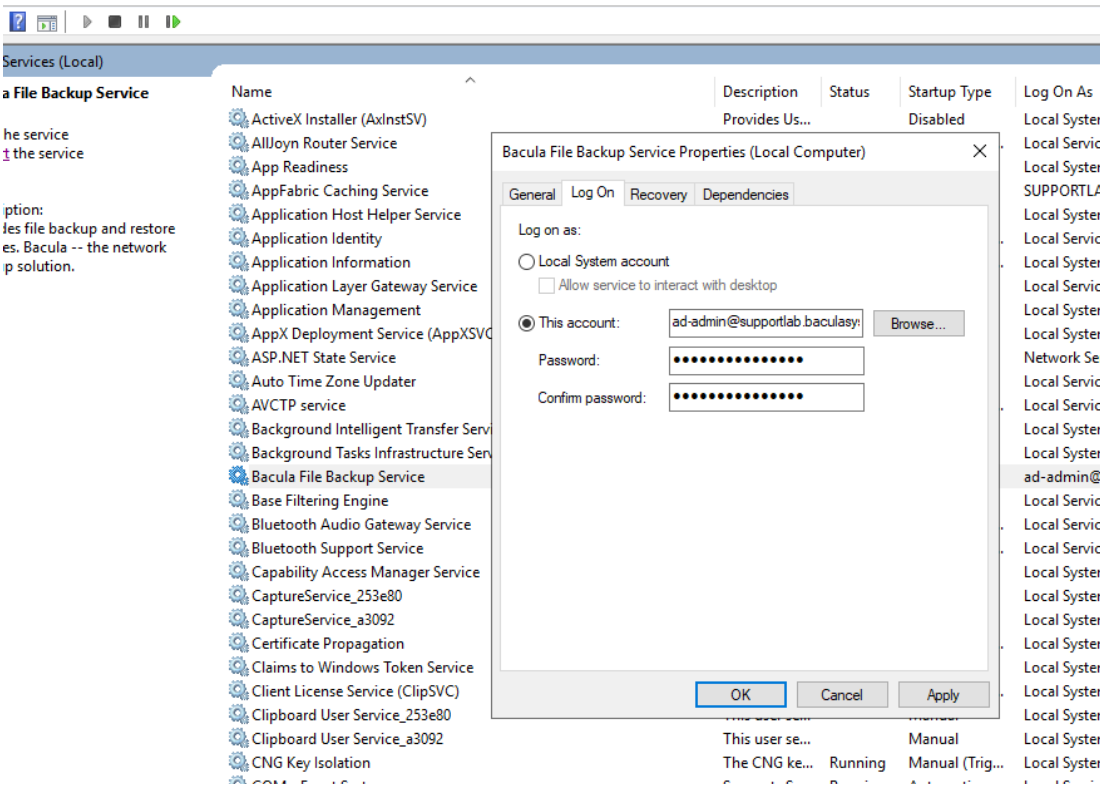
[ ] Allow service to interact with desktop

( ) This account: [ ] Browse...

Password: [ ]

Confirm password: [ ]

OK | Cancel | Apply


Services (Local)

a File Backup Service

he service
t the service

ption:
les file backup and restore
es. Bacula -- the network
p solution.

Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local Syster
AllJoyn Router Service				Local Servic
App Readiness				Local Syster
AppFabric Caching Service				SUPPORTLA
Application Host Helper Service				Local Syster
Application Identity				Local Servic
Application Information				Local Syster
Application Layer Gateway Service				Local Servic
Application Management				Local Syster
AppX Deployment Service (AppXSVC				Local Syster
ASP.NET State Service				Network Se
Auto Time Zone Updater				Local Servic
AVCTP service				Local Servic
Background Intelligent Transfer Servi				Local Syster
Background Tasks Infrastructure Serv				Local Syster
Bacula File Backup Service				ad-admin@
Base Filtering Engine				Local Servic
Bluetooth Audio Gateway Service				Local Servic
Bluetooth Support Service				Local Servic
Capability Access Manager Service				Local Syster
CaptureService_253e80				Local Syster
CaptureService_a3092				Local Syster
Certificate Propagation				Local Syster
Claims to Windows Token Service				Local Syster
Client License Service (ClipSVC)				Local Syster
Clipboard User Service_253e80				Local Syster
Clipboard User Service_a3092	This user se...		Manual	Local Syster
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local Syster

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

( ) Local System account

[ ] Allow service to interact with desktop

[● This account:] ad-admin@supportlab.baculasy:

Password: •••••••••••••••

Confirm password: •••••••••••••••

Job Configuration

Once the Bacula File Daemon and the **Hyper-V WMI plugin** are correctly installed and configured, setting a backup job up is as simple as adding the job and the fileset within the Bacula Director configuration file.

Important: The `Enable VSS` parameter must be set to `no` in the Fileset (see examples below).

The following plugin options are supported:

Name	Status	Default	Description
include	Optiona	Include all (*)	a Unix shell-style wildcards pattern for including VMs by name
exclude	Optiona	Exclude none	a Unix shell-style wildcards pattern for excluding VMs by name
tmp_dir	Optiona	`Bacul`	locates the Bacula working repository folder. Make sure there's enought space on this location to create VM's shapshots and exports. Default is a `Bacula-repo` folder in the VHD location.
pre_bacl	Optiona	None	action on the VMs before backup takes place. Can be None, Stop, Save. - None is noop. - Stop stops the VM before backup (useful when VM doesn't support VSS or kernel freeze to maintain consistent backups). - Save saves the VM before stoping it.
post_bac	Optiona	None	action on the VMs after backup is completed. Can be None, Restart, ForceRestart. - None is noop. - Restart restarts the VM if it was stopped or saved pre-backup. - ForceRestart restarts the VM unconditionnaly.
consistency_le	Optiona	Application	overwrites the consistency level. Can be Application Consistent of Crash Consistent. Application is the recommanded value but some VMs might not support it.
allow_pre_	Optiona	Disabled	when enabled, allows retry with `pre_backup_action` set to Save and `post_backup_action` set to Restart, if crash consistency retry backup has failed.
localhost_onl	Optiona	Disabled	when anabled, restricts all operations to the local node (for Failover Cluster configuration).
abort_or	Optiona	Disabled	abort immediately the job if a serious error is found (b.e when no VM matches the `include` patterns). By default, a Job error is raised, but the job continues.
disable_vm	Optiona	Disabled	when enabled, VMs migration is disabled during backup to avoid collision between backup an migration (for Failover Cluster configuration). Doesn't take any value. To disable, remove keyword.
user_nai	Optiona	None	the user name that will run the backup/restore operation. This is **not the recommanded method**. The user name can be specified locally on the hyper-v node in a `bacula-hv.usr` file located in the fd plugins folder.
user_pas	Optiona	None	the user password that will run the backup/restore operation. This is **not the recommanded method**. The user password can be specified locally on the hyper-v node in a `bacula-hv.pwd` file located in the fd plugins folder.

Examples

Example 1: backup all vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupAll"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="Simplest-Hyper-V-Fileset"
```

(continues on next page)

(continued from previous page)

```
  Storage = File
  Messages = Standard
  Pool = Default
}

Fileset {
  Name = "Simplest-Hyper-V-Fileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "hyperv-wmi:"
  }
}
```

Example 2: backup only «Linux- » prefixed vms using Bacula's default working directory

```
Job {
  Name = "Hyper-V-BackupOnlyLinux"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="Linux-Hyper-V-Fileset"
  Storage = File
  Messages = Standard
  Pool = Default
}
Fileset {
  Name = "Linux-Hyper-V-Fileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "hyperv-wmi: include=\"Linux-*\""
  }
}
```

Example 3: backup any VM having «Windows» in its name, using a custom working directory

```
Job {
  Name = "Hyper-V-BackupOnlyWindowsOnF"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="WindowsOnF-Hyper-V-Fileset"
  Storage = File
  Messages = Standard
```

(continues on next page)

(continued from previous page)

```
  Pool = Default
}
Fileset {
  Name = "WindowsOnF-Hyper-VFileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "hyperv-wmi: include=\"*Windows*\" tmp_dir=\"F:/backup\""
  }
}
```

Backup

The files backed up by the Hyper-V server will be visible in a bconsole or with the prefix `/@HYPERV-WMI/` .

Typically, a VM backup data is organized as follows:

```
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/1ECD8F42-CCCA-462A-AB0F-
↪B7644EA77B9B.vmcx
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/1ECD8F42-CCCA-462A-AB0F-
↪B7644EA77B9B.vmgs
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/1ECD8F42-CCCA-462A-AB0F-
↪B7644EA77B9B.VMRS
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/backup-config.xml
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/test1_1C6A2D1E-9CEF-4F08-
↪A14E-E463F909C94D.avhdx
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b/test1.vhdx
/@HYPERV-WMI/1ecd8f42-ccca-462a-ab0f-b7644ea77b9b-test1
```

Where:

- 1ecd8f42-ccca-462a-ab0f-b7644ea77b9b is the VM UID of the “test1”
- the .vmcx file stores the Vm’s machine settings
- the .vmgs file stores the Vm’s guest state
- the .vmrs file stores the Vm’s running state
- the backup-config.xml contains information on the vm at the backup time
- the vhdx file stores the Vm’s Virtual Hard Drive data
- the avhdx file stores the differential data of the Vm’s Virtual Hard Drive
- 1ecd8f42-ccca-462a-ab0f-b7644ea77b9b-test1 is a convenience file reminding us that the VM name is test1 and its ID 1ecd8f42-ccca-462a-ab0f-b7644ea77b9b

Incremental-Differential backups: the **Hyper-V WMI Plugin** will automatically follow the backup level strategy as scheduled in Bacula.

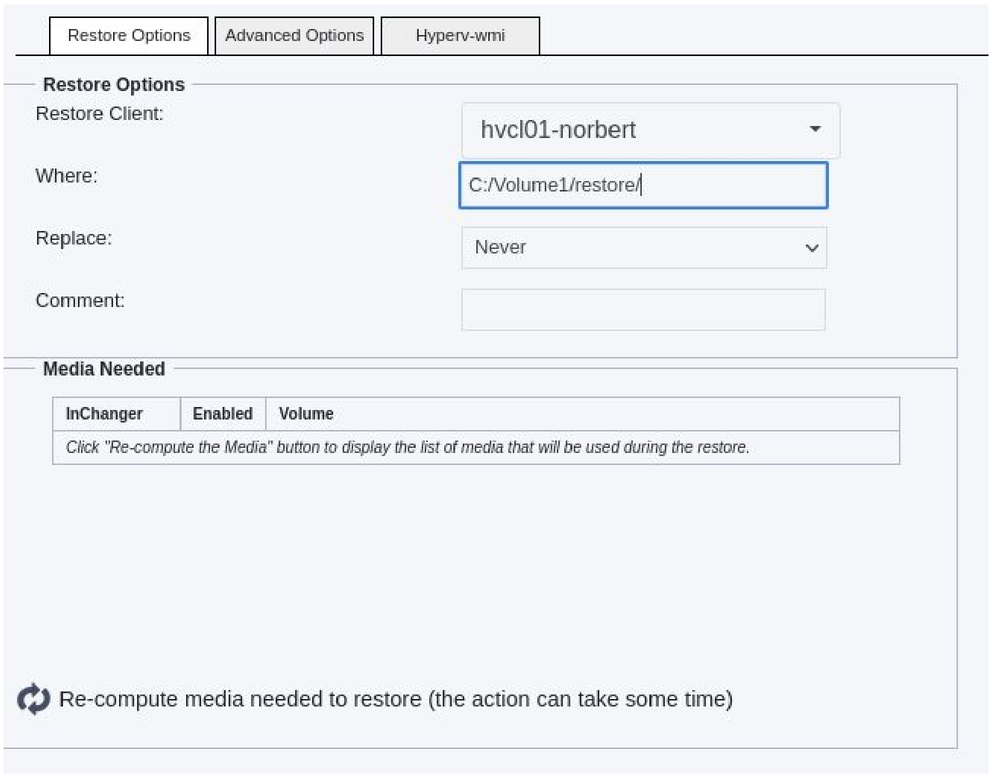
Consistency Level: A backup can fail when the option “Application Consistent” is required for a VM that doesn’t support it.

- If an Application Consistent backup fails, the **Hyper-V WMI Plugin** will change automatically the Consistency Level to “Crash Consistent” and retry.
- If `allow_pre_save` if enabled and a “Crash Consistent” backup fails, the **Hyper-V WMI Plugin** will change the `pre_backup_action` to “Save” and `post_backup_action` to “Restart” and retry.
- If none of the above works, the backup fails with Error.

Restore

It is advisable to choose the entire fileset instead of cherry-picking backed up files, especially for one VM.

Restore parameters:



Restore Options

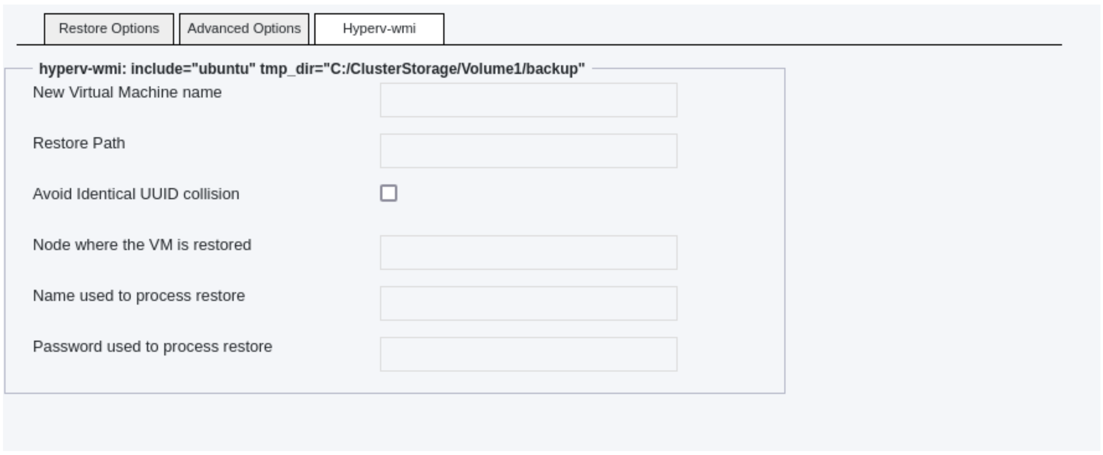
Restore Client:	hvcl01-norbert
Where:	C:/Volume1/restore/
Replace:	Never
Comment:	

Media Needed

InChanger	Enabled	Volume
Click "Re-compute the Media" button to display the list of media that will be used during the restore.		

Re-compute media needed to restore (the action can take some time)

- `Where`: Can specify a path for VM restoration. If the content is not a path (does not contain slashes or backslashes), it’s considered to be the new VM restore name.
- `New Virtual Machine Name`: Specify the restored VM new name
- `Restore Path`: Specify the location where Snapshot files are restored
- `Avoid Identical UUID collision`: Over restoration, the VM UID is regenerated. It avoids issue when the original VM is still existing on the Hyper-V host.



hyperv-wmi: include="ubuntu" tmp_dir="C:/ClusterStorage/Volume1/backup"

New Virtual Machine name	
Restore Path	
Avoid Identical UUID collision	☐
Node where the VM is restored	
Name used to process restore	
Password used to process restore	

- `Node where the VM is restored`: Specify the name of the node where the VM is to be restored (clustered configuration). Note: the Restore Path need to be shared between the local host and the remote node, for this option to work (on a clustered storage b.e.)
- `Name used to process restore`: impersonation user name (see Impersonation of the **Hyper-V WMI plugin**)
- `Password used to process restore`: impersonation user password (see Impersonation of the **Hyper-V WMI plugin**)

VM Renaming:

If the 'New Virtual Machine Name' is specified, the restored VM(s) will be renamed using this value. However, if the 'Where' value is not a path, then the 'Where' value itself will be used for renaming the restored VMs. In case neither of these options are set, the restored VM(s) will retain their original name(s).

Restore Path:

If the 'Restore Path' is provided, it will be used as the restore path for all the drive-related backup files (vhdx, avhdx) and the VM-related backup files (vmcx, vmgs, vmrs). However, if the 'Restore Path' is empty and the 'Where' value is set with a path, then the 'Where' path will be used instead. To facilitate multiple restorations, the files will be restored in a specific folder named after the Bacula job name. If none of the above options are set, the default locations of the Hyper-V host will be used, and no specific folder named after the Bacula job name will be created. It is important to note that the restore files are not moved during the restoration process, so the Restore Path will be the location of the restored VM.

VMs duplication:

If the original VM still exists on the node, attempting to restore the same VM with the same unique identifier will be rejected. In such cases, the “Avoid Identical UUID collision” option can be used to assign a new unique identifier to the restored machine. If a restoration without the “Avoid Identical UUID collision” option fails, it will automatically be retried with this option enabled, and a warning will be issued.

Retries:

In the event of a VM restoration failure, the import process will be retried. If the “Avoid Identical UUID collision” option is turned off, it will be automatically enabled to prevent the most common cause of restoration errors: duplication of unique identifiers. Alternatively, an attempt will be made to rename the VM. By default, the rename will follow the pattern “<JobName>_<originalVMName>”.

Examples:

- Defaults:
  - `Where`: Empty
  - `New Virtual Machine Name`: Empty
  - `Restore Path`: Empty
  - `Avoid Identical UUID collision`: Off
  - `Node where the VM is restored`: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs and VirtualHardDrives locations with original names are Unique Identifiers.

- Quick Rename:
  - `Where`: newVMName
  - `New Virtual Machine Name`: Empty
  - `Restore Path`: Empty
  - `Avoid Identical UUID collision`: Off
  - `Node where the VM is restored`: Empty

The restored VM(s) are (re)created in the local Hyper-V host default VMs and VirtualHardDrives locations and renamed newVMName.

- Large Restore:
  - `Where`: Empty
  - `New Virtual Machine Name`: NewVMName
  - `Restore Path`: C:\LargeStorage\restore
  - `Avoid Identical UUID collision`: Off
  - `Node where the VM is restored`: Empty

The restored VM(s) are (re)created and renamed NewVMName. Virtual drive(s) and VM files are located into a folder named after the JovName in C:\LargeStorage\restore. Something like : C:\LargeStorage\restore\RestoreFiles.<date>_<time>.

Best Practices

While it is technically possible to backup multiple VMs in one Bacula hypervisor plugin backup job (VMware, Hyper-V, RHV, Proxmox, etc), this is not necessarily the best way to perform VM backups. It is strongly recommended that one backup Job is created for each VM being backed up for the following reasons:

- By default, if one of your VMs fails to backup in a "multi-VM" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some VM backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called "abort_on_error" in each of the Bacula hypervisor plugins, which causes Bacula to immediately fail the job as soon as an error is detected while backing up a VM. However, if you use this option, and the backup of VM number 11 in a list of 50 VMs fails, then the whole job will be failed, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 configuration (one VM backed up per job) means that the "abort_on_error" option will make more sense to enable in each job so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.
- With a 1:1 VM/Job configuration, re-running a specific VM backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 50 VMs, without a 1:1 configuration, there is no way to re-run a backup of just the one VM that failed to backup.
- Additionally, with a 1:1 VM/Job configuration, job metrics will have more meaning because each VM will be one job, and you will know to expect a specific number of jobs each night with each job representing one VM.
- With a multi-VM per job configuration, each VM will be backed up "serially", one at a time, disk by disk, VM by VM. A 1:1 configuration will allow several VM backups to be run concurrently which will reduce the overall time to perform the VM backups. Of course, you will need to pay close attention to SD and ESXi storage and networking resources, and adjust the number of concurrent jobs accordingly.
- For some hypervisors (VMware, Proxmox, etc) Bacula provides automation scripts (eg: scan_datacenter.pl for VMware). These scripts are designed so that they will create 1:1 VM/Job configurations. If you plan to make use of these automation scripts, it is a good idea to already be thinking this way, and having your hypervisor plugin backup configurations in a 1:1 configuration from the beginning.

Failover Cluster

Backup operations are seamlessly executed in a Failover Cluster setup, regardless of the node responsible for hosting the VM(s) at the time of backup. As long as the user possesses the correct credentials on all nodes, the VMs within the cluster will undergo filtering via include/exclude criteria. The tmp_dir directory must be situated on the Cluster Shared Volume and be accessible to the user through an identical path on each node. By default, the tmp_dir is designated as a "Bacula-repo" folder within the VHD default directory, ensuring optimal performance as long as sufficient disk space is allocated for snapshots on the Shared Volume. It is important to note that transferring a VM from one node to another during a backup process is prohibited by Hyper-V.

Single Item Restore

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This article presents how to use the Hyper-V VSS Single File Restore feature with **Bacula Enterprise** and the Hyper-V Plugin.

Features Summary

The **Bacula Enterprise** Hyper-V VSS Single File Restore provides the following main features:

- Console interface
- Compatible with Hyper-V Server 2019
- Support for Linux filesystems (ext3, ext4, btrfs, lvm, xfs)
- Support for Windows filesystems (FAT, NTFS)

Hyper-V SIR is available starting with Bacula Enterprise 12.8

This document will present solutions for **Bacula Enterprise** 12.8 and later, which are not applicable to prior versions. The Hyper-V VSS Single File Restore has been tested and is supported on RHEL 7.x, RHEL 8.x, Ubuntu Focal and Debian Stretch. SELinux is currently not supported.

Installation

Packages for the Hyper-V VSS Single File Restore plugin are available for supported platforms. Please contact Bacula Systems to get them.

Download the plugin package to your **Storage Daemon** server and then install using the package manager like so:

```
rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the Hyper-V VSS Single File Restore plugin and will install dependencies. On RHEL, it will be necessary to install the `perl-JSON` package from **rpmforge** and the `libguestfs-winsupport` package.

Note: On RHEL 7.x, it is necessary to install a custom version of the libguestfs packages from our repository to support NTFS devices. Those should not be updated with a newer version from official repositories. The YUM package manager has plugins to prevent package updates, try **yum-plugin-versionlock** or **yum-plugin-priorities**.

Additionally, the `ntfs-3g` package from the EPEL repository is needed for NTFS support. To install the EPEL respository, please follow the official instructions on the EPEL website to install the "epel-release" package here:

https://fedoraproject.org/wiki/EPEL

Note: On RHEL 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag] name = Red Hat Enterprise - RPMFORGE
baseurl = https://www.baculasystems.com/dl/DAG/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0

# cat /etc/yum.repos.d/baculasystems.repo
[single_file_restore_hyper-v]
name = Red Hat Enterprise - RPMFORGE
baseurl = https://www.baculasystems.com/dl/<xxx>/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0
```

Note: This following repository is required on RHEL7:

```
[Bacula-Enterprise-DAG-Guestfish]
name = Bacula Enterprise - DAG for Guestfish
```

(continues on next page)

(continued from previous page)

```
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64/guestfish/
enabled = 1
protect = 0
gpgcheck = 0
```

```
yum install bacula-enterprise-single-item-restore perl-JSON
```

If BWeb Management Suite is used, you need to run:

```
service bweb restart
```

Notes about the “bacula” Account on RHEL

All commands in this document use the “bacula” unix account to run.

On RHEL, the Unix “bacula” account is locked by default. It means that it’s not possible by default to execute a command such as “su - bacula”.

It is possible to unlock the “bacula” account, or to use “sudo -u bacula” to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

Can be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using

```
root@storage# sudo -u bacula /bin/bash
```

Or unlock the “bacula” unix account and use “su -” with a command such as:

```
root@storage# chsh -s /bin/bash bacula root@storage# su - bacula
bacula@storage# whoami bacula
```

Fuse FileSystem

If a restore session is not properly cleaned up, some directories might still be mounted with the Bacula Fuse FileSystem.

```
baculafs on /opt/bacula/working/cat-ro type fuse.baculafs
(ro,user=bacula) backend0 on /opt/bacula/working/test-vm-0 type
fuse.backend0 (ro,user=bacula) /dev/fuse on
/opt/bacula/working/test-vm type fuse (rw,nosuid,nodev,user=bacula)
```

It is possible to unmount directories with the `fusermount -u` command.

```
bacula@storage# fusermount -z -u /opt/bacula/working/26
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm-0
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm
```

Samba SMB Shares

The **Bacula Enterprise** Hyper-V VSS Single File Restore plugin can automatically set up Samba SMB shares from the console program or the BWeb Management Suite.

To enable Samba SMB network shares, installing and configuring the "samba" package is mandatory. To configure the `/etc/samba/smb.conf` file correctly, you need to run `install-single-item-restore.sh` script.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
include = /etc/samba/conf.d/all
```

At this point, it is possible to modify `/etc/samba/smb.conf` to add your own configuration directives.

Network share descriptions will be stored in the directory `/etc/samba/conf.d`. It is possible to create and customize the template used by Bacula to generate configuration files.

```
root@storage# cat /etc/samba/conf.d/custom.tpl
[__share__]
path = __path__
follow symlinks = yes
wide links = yes
writable = yes
```

Configuration

On the **Storage Daemon** host server, the `bconsole` program should be configured properly to let the "bacula" user connect to the Director with `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 12.8.0
Enter a period to cancel a command.
* version
mydir-dir Version: 12.8.0 x86_64-redhat-linux-gnu
* quit
```

The package contains a script to test the connection with the Director and to test if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage# /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
```

(continues on next page)

(continued from previous page)

```
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you may not see any files or files may have strange permissions, ownerships and sizes and will inaccessible even to the root user.

Restore Scenario With Text Console Interface

The Hyper-V VSS Single File Restore plugin provides a simple console program that provides access to files inside VMs.

```
bacula@storage# /opt/bacula/bin/mount-vm
Automatically Selected Catalog: MyCatalog

Client list:
1: 127.0.0.1-fd
2: win2008-fd
3: rhel7-fd Select a Client: 1
Selected Client: 127.0.0.1-fd

Job list:
1: HYPERV08.2021-02-15_19.12.51_34
2: HYPERV08.2021-02-16_12.12.29_39
3: HYPERV08.2021-02-16_12.37.54_03
4: HYPERV08.2021-02-16_14.23.47_03
5: HYPERV08.2021-02-16_15.45.32_03
6: HYPERV08.2021-02-16_17.00.47_52
Select a Job: 6
Selected HYPERV08.2021-02-16_17.00.47_52

Virtual Machine:
1: squeeze2
2: win2008
3: rhel7
4: sir-test-vm
Select a Virtual Machine: 4
Selected sir-test-vm

Actions list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Cleanup
Select a Actions: 1
Selected Mount guest filesystem locally

I: Files are available under /opt/bacula/working/mount-vm-6434/disks/sir-test-
↪vm
```

(continues on next page)

(continued from previous page)

```
I: Press enter to finish and cleanup the session
```

In this step, the virtual machine filesystem is mounted locally (in the example above, files are available under `/opt/bacula/working/mount-vm-6434/disks/sir-test-vm`. It is possible to browse directories and copy files (with `cp`, `scp`, `ftp`) as with a standard filesystem from another terminal session with the Unix "root" and "bacula" accounts. If you need to use another Unix account to operate on files, use the "-o allow_other" option when starting the `mount-vm` script.

```
bacula@storage# ls /opt/bacula/working/mount-vm-6434/disks/sir-test-vm
bin   dev  home        lib         media  opt   root  selinux  sys  usr ␣
↪vmlinuz
boot  etc  initrd.img  lost+found  mnt    proc  sbin  srv      tmp  var
```

To clean up the session, just press "Enter" in the terminal session where the `mount-vm` script was started.

It is possible to limit the Job list with the following command line options:

- `-s=<days>` Limit the job list to the last *days*
- `-l=<number>` Limit the job list to the last *number* entries
- `-f=<filter>` Specify an advanced filter based on the Job name, the Fileset name or the JobId

```
# Limit the job output to the last 100 jobs
bacula@storage# /opt/bacula/bin/mount-vm -l 100

# Limit the job output to the last 30 days
bacula@storage# /opt/bacula/bin/mount-vm -s 30

# Limit the job output to jobs that start with "MyHyper-V"
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyHyper-V*'

# BAD USAGE for the filter option, it will search for a job named "MyHyper-V"
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyHyper-V'

# Limit the job output to jobs that start with "MyHyper-V"
# and that use the Fileset Test1
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyHyper-V\*␣
↪fileset=Test1'

# Limit the job to the jobid XX
bacula@storage# /opt/bacula/bin/mount-vm -f jobid=XX
```

On some cases, the device detection doesn't work properly. It is possible to use the `-m` option to mount recognized disks in a simple way. The option is automatically set when only one disk is selected during the restore.

```
bacula@storage# /opt/bacula/bin/mount-vm -m
```

Notes

Cache Directory

To speed up future Hyper-V Single File restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
bacula@storage# ls /opt/bacula/working/mount-cache
sir-test-vm-0.bmp  sir-test-vm-2.bmp    MyCatalog-2.idx  MyCatalog-5.idx ␣
↪MyCatalog-8.idx
sir-test-vm-1.bmp  sir-test-vm.profile  MyCatalog-4.idx  MyCatalog-6.idx ␣
↪MyCatalog-9.idx
```

It is possible to remove files in the cache after some time; they will be re-generated if needed.

Support

The `install-single-item-restore.sh` script can collect traces automatically when a `mount-vm` session is running.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh support
```

Limitations

- The Hyper-V VSS Single File Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with the MySQL catalog backend due to internal MySQL limitations with indexes on TEXT colums. For Hyper-V Single Item Restore there should not be too much impact on performance (the backup structure is usually quite small) but we advise to use the PostgreSQL backend for the best experience.
- The Hyper-V VSS Single File Restore performance may vary depending on various factors. For instance, Bacula will have to read more data if the Volume was created with a large number of concurrent jobs.
- The Storage Daemon where the Hyper-V VSS Single File Restore is installed should be have a CPU with the VT-x/EPT extensions. If these extensions are not available, you may experience delays (from 20 seconds up to 10 minutes in our lab).
- RHEL 7 does not support mounting NTFS disks with the libguestfs provided with their system. To mount Microsoft NTFS disks on RHEL 7, it is required to install a patched version of the libguestfs packages. Please see notes in the "Installation" section of this document for more information.
- The Hyper-V VSS Single File Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.

Hyper-V Plugins to Cover All Backup Needs

Microsoft has created various technologies for backing up Hyper-V virtual machines. Thus, multiple Bacula Plugins have been designed to maximize the benefits of each solution.

Hyper-V is equipped with a VSS writer on all compatible versions of Windows Server. This VSS writer enables developers to utilize the existing VSS infrastructure for backing up virtual machines to Bacula using the Bacula Enterprise VSS Plugins. This technology is supported by the original **Bacula Hyper-V plugin**. Although it doesn't allow incremental and differential backups and other more granular VM-based options, it can still cover any Hyper-V version. Therefore, it is highly recommended for small standalone Hyper-V servers (single node, no Failover Cluster) and older Hyper-V versions that do not offer Virtual Disk Service or Hyper-V WMI API.

The Virtual Disk Service (VDS) is a service provided by Microsoft Windows that handles query and configuration tasks upon request from end users, scripts, and applications. This service is compatible with Windows Server 2003, Windows Vista, and newer versions. The **Hyper-V Winapi Plugin** utilizes this technology to backup and restore virtual machines. It supports incremental and differential backups, making it the recommended solution for more intricate Hyper-V servers, such as Failover Clusters with multiple nodes, where local disk space is a critical resource. Depending on the relocation of virtual machines within the Cluster, it may be necessary to migrate the backed-up VMs across specific nodes.

Starting in Windows 8 and Windows Server 2012, Hyper-V supports backup via the Hyper-V WMI API. This feature enables individual Guest VMs to be backed up separately and incrementally, offering a more scalable solution compared to using VSS in the host. The Bacula Enterprise **Hyper-V WMI Plugin** uses this technology for backup and restore to/from Bacula. It backups/restores VM in the recommended Microsoft format. By utilizing the Microsoft snapshot format from/to disk, it is essential to have sufficient disk space available for the process to proceed smoothly. In scenarios where disk space may be limited due to busy configurations, the Hyper-V Winapi Plugin can be utilized as an alternative solution.

Backups created using the **Hyper-V WMI Plugin**, **Hyper-V Winapi Plugin** and **Hyper-V Plugin** are not compatible with each other.

RHV Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

Scope

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. Virtualization technologies are an important and widely used element of the resulting strategies and accordingly found in most data centers.

Naturally, reliable solutions to protect the underlying data are a critical topic. This white paper presents the Bacula Enterprise plugin and strategy to protect Red Hat Virtualization environments.

The plugin provides several ways to backup virtual machines, image level with a full backup or proxy backup, incremental or differential backup, with a set of options to select different backup sets. To later, recovery them with a set of options to customize the restore process.

Red Hat Virtualization Technology

Red Hat Virtualization (RHV) is a full virtualization platform capable of managing the full set of features that make up a virtualized data center: Hosts, networks, local, remote, or distributed storage. Management, accounting and monitoring are done through an API-based Web interface, which implements users and roles to implement access restrictions.

Ovirt

RHV is based on the open source OVirt virtualization platform. OVirt is available for Linux distributions related to RHEL. OVirt is under active development, and RHEL integrates releases they consider mature. Therefore, Ovirt itself can typically exist in newer versions than what is part of RHV.

Architecture

RHV is composed of the following elements:

RHV Manager
: is the service offering management through a web interface and APIs.

Virtualization Hosts
: are the hypervisors providing their resources to the virtual machines. They are managed through the RHV Manager, and are based on the KVM system.

  Two kinds of hosts are available:

  RHEL hypervisors
  : being standard RHEL server systems

  oVirt nodes
  : are minimal servers which are distributed as ISO images to install a lean virtualization platform solely for RHV usage.

There are 2 ways to deploy the RHV Manager: Standalone, with the RHV Manager running on a host outside of the virtualized environment, which does not provide native high availability, but is easily deployed and managed, and self-hosted, where RHV Manager runs inside the virtualized environment, in a dedicated VM.

The latter approach can be a highly available service by making use of the `ovirt-ha-agent` and `virt-ha-broker` services.

In figure *RHV Host Architecture*, the architecture of a virtualization host is shown. The stack of technologies employed by a host to provide a VM consists of the following elements:

VDSM
: is the host agent service running on every virtualization host to provide the hypervisor service to the RHV system. This services listens on the TCP port 54321 on the network.

libvirt
: is the toolkit that manages virtual machines.

QEMU
: is the multi-platform emulator which provides emulation of full systems, including CPU capabilities not available on the underlying system.

KVM
: is the kernel module which provides the system integration component to QEMU. It allows to

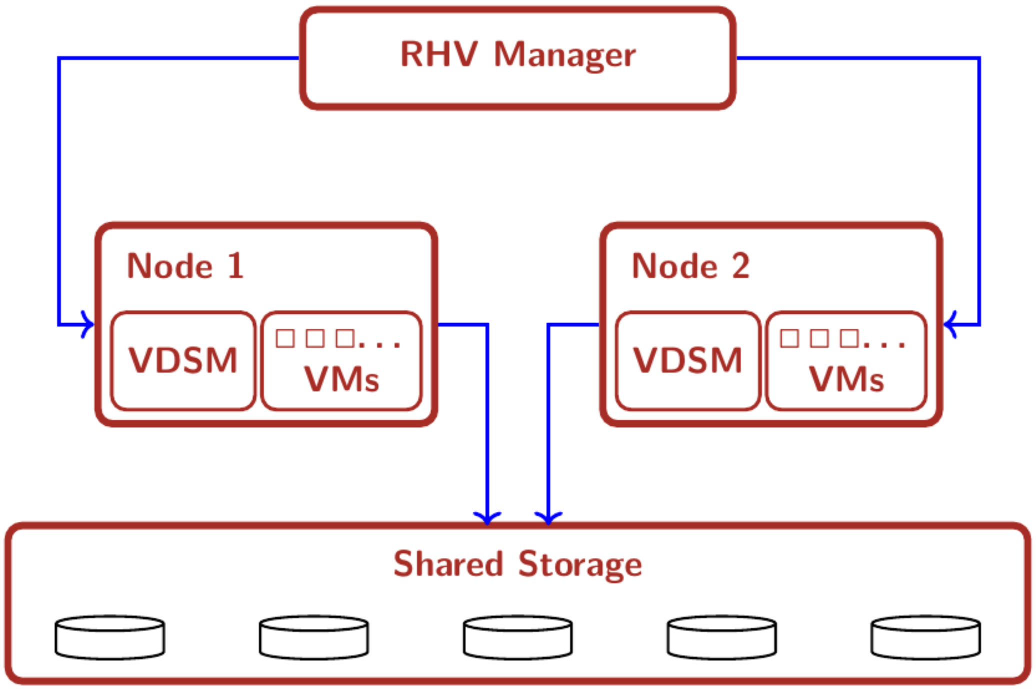


Fig. 15: General Architecture (Standalone mode)

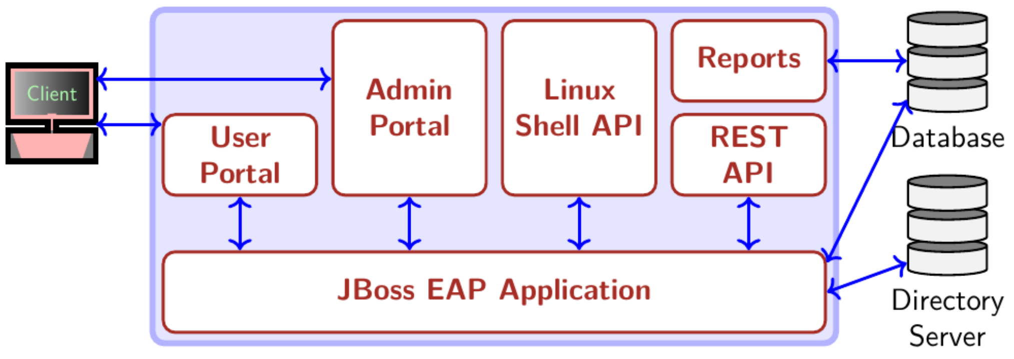


Fig. 16: RHV Manager Architecture

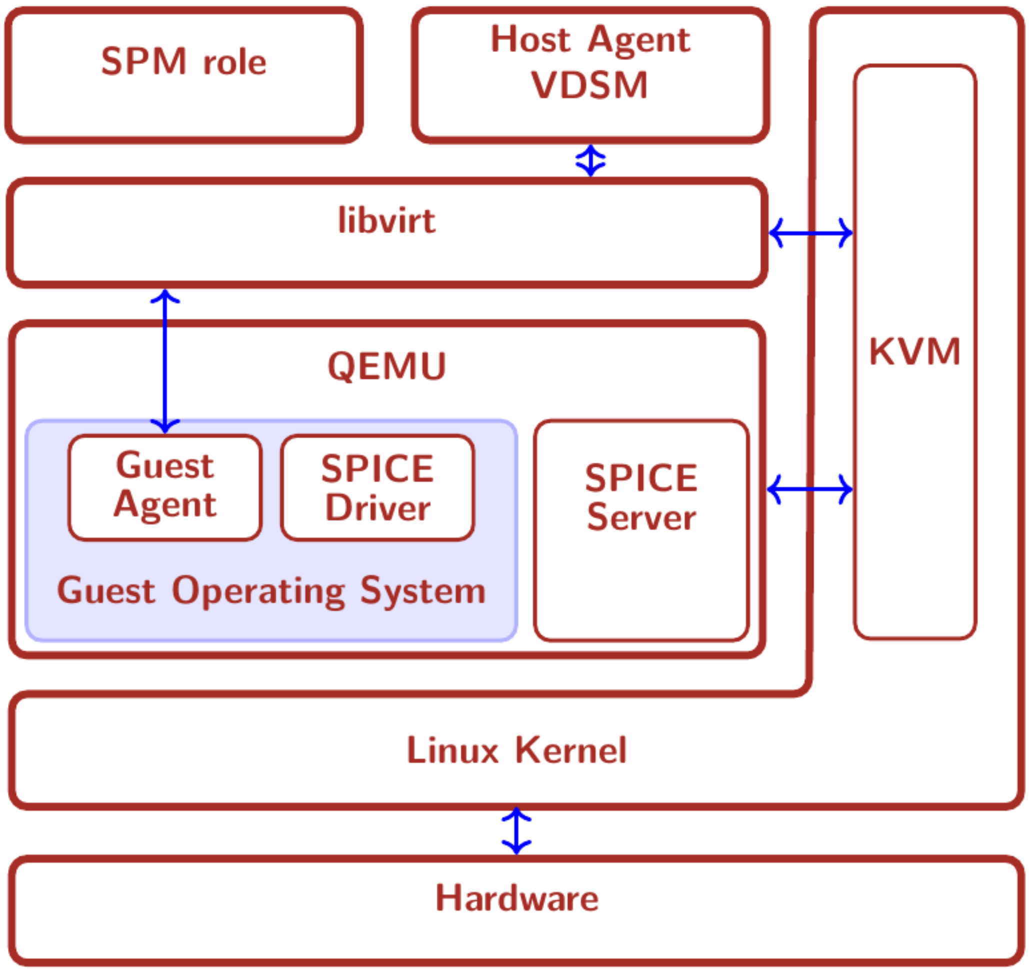


Fig. 17: RHV Host Architecture

execute guest VMs in user space, ensuring full segregation of VMs from each other and the host system.

SPICE
is the interface between virtualized user-facing components and remote interfaces, providing input, display, and sound services to remote interfaces (the "viewer" program).

Other important concepts to understand a RHV system are

Datacenter
is the most high-level group of virtualization systems, the VMs managed in it, and all the resources available.

Cluster
refers to a group of virtualization hosts (sharing networks, storage, and some other infrastructure properties). A Cluster is always part of a distinct Datacenter, and a Datacenter can consist of more than one Cluster.

Storage Domain
indicates a logical entity providing storage capacity to Virtual Machines of a Datacenter.

Storage Pool Manager
is the role of a certain host in the datacenter which creates, manages, and removes virtual disk images.

Template
is the base configuration of vms. This configuration can be from the architecture of the processor that will use the vm to the disks attached to it.

Host
is a physical server on which the virtual machines run. Alternatively, host is also called hypervisor. Both RHEL Virtualization Hypervisors and RHEL hosts interact with the rest of the virtualized environment in the same way.

Template

The templates have the ability to create virtual machines quickly and conveniently. To create a template we need a vm to copy its configuration, its networks and content of its disks.

The templates have a specific configuration and unique content on the disks. Once the template is created the content of the disks is immutable and RHV does not allow to add or remove disks. However, it is possible to modify, add or modify the networks attached to the template.

Storage Domain

Storage Domains are distinct storage subsystems used to host different kinds of information.

Two Storage Domains are used by RHV:

The Data Domain
stores virtual hard disks of VMs and templates in a Data Center. Data Domains are exclusive to one Data Center.

Data Domains can be backed by different storage attachment technologies, such as NFS, Fibre Channel, FCoE, GlusterFS, Ceph, iSCSI, and any (local) POSIX-compliant file system.

The ISO Domain
hosts images of media such as CDs and DVDs to deploy software on VMs. There can only be one

ISO Domain in a Data Center, an ISO Domain can be shared among Data Centers, and NFS is the only available backing store for this type of storage domain.

Data Warehouse

RHV employs a database of historical data where all the management and auditing data is stored. This function is provided by the `ovirt_engine_history` service. It makes use of the PostgreSQL database for storing its data.

Networking

RHV allows the definition of different logical networks to isolate traffic types and paths. These networks are created, maintained, and destroyed by the RHV Manager, which also handles VLANs, routing, and firewalling.

Usually, there will be several networks in use in a RHV Datacenter. Some pre-defined network types are available to segregate different types of traffic: A management network which should be used exclusively by RHV management communications, VM networks for use by the virtual machines, storage networks for example for iSCSI or NFS traffic, and storage migration networks.

Networks for dedicated purposes can be created as needed, and they can be bound to specific network hardware (and VLANs) available on the virtualization hosts.

Red Hat Virtualization APIs

RHV provides different APIs to access its functionality:

- Shell: 4.1 4.2
- REST API: 4.1 4.2 4.3
- SDKs (based on the REST API)
  - Java: 4.1 4.2 4.3
  - Python: 4.1 4.2 4.3
  - Ruby: 4.1 4.2 4.3

The Bacula Enterprise Red Hat Virtualization Plugin is based on the Java SDK 4.3

Connection Modes

RHV allows 2 different authentication schemes:

- OAuth Authentication
- HTTP Basic Authentication

The plugin supports both connection and authentication modes.

More Information

The information of this chapter is based upon the official RHEL Documentation available in the following links:

General Documentation
: redhat.com/.../red-hat-virtualization/

Version Information
: redhat.com/.../updates/rhev

oVirt
: https://ovirt.org/

Architecture and Design

The Red Hat Virtualization Plugin is a File Daemon plugin. It may be installed on a machine inside or outside of your RHV environment. Is based on a Java SDK, therefore is non-machine dependent and is compatible with any Operating System where the Bacula File Daemon can run and the Java Virtual Machine is available.

Backup Process

The backup process allows you to perform in two different environments: external or internal. The external process allows full clone backups or snapshots, and incremental or differential backups. The internal process allows full backups to be performed faster than the external processes.

Backup from RHV environment external machine

The next backups methods (clone, snapshot, template) perform a download of the VM's disks through the API. These methods are slower than the 'Proxy VM' method, but they do not depend on a virtual machine in the RHV environment.

The general backup process of a virtual machine is as follows (check also Figure *External backup diagram*):

1. Check for the existence of the virtual machine to protect and the compatibility of the RHV environment with the chosen backup method (check Section *Limitations*).
2. If it is not a template backup:
   - Check the snapshots:
     - If the last backup was incomplete and the generated snapshot was not removed successfully.
     - If apply a full backup in a virtual machine with incremental snapshots.
   - Lauch a snapshot.
3. If it is a clone backup:
   - Clone the virtual machine using the created snapshot.
   - Download configuration of the original virtual machine and cloned machine (XML).
   - Download all disks of cloned machine.
4. If it is a template backup:
   - Download configuration of template (XML).

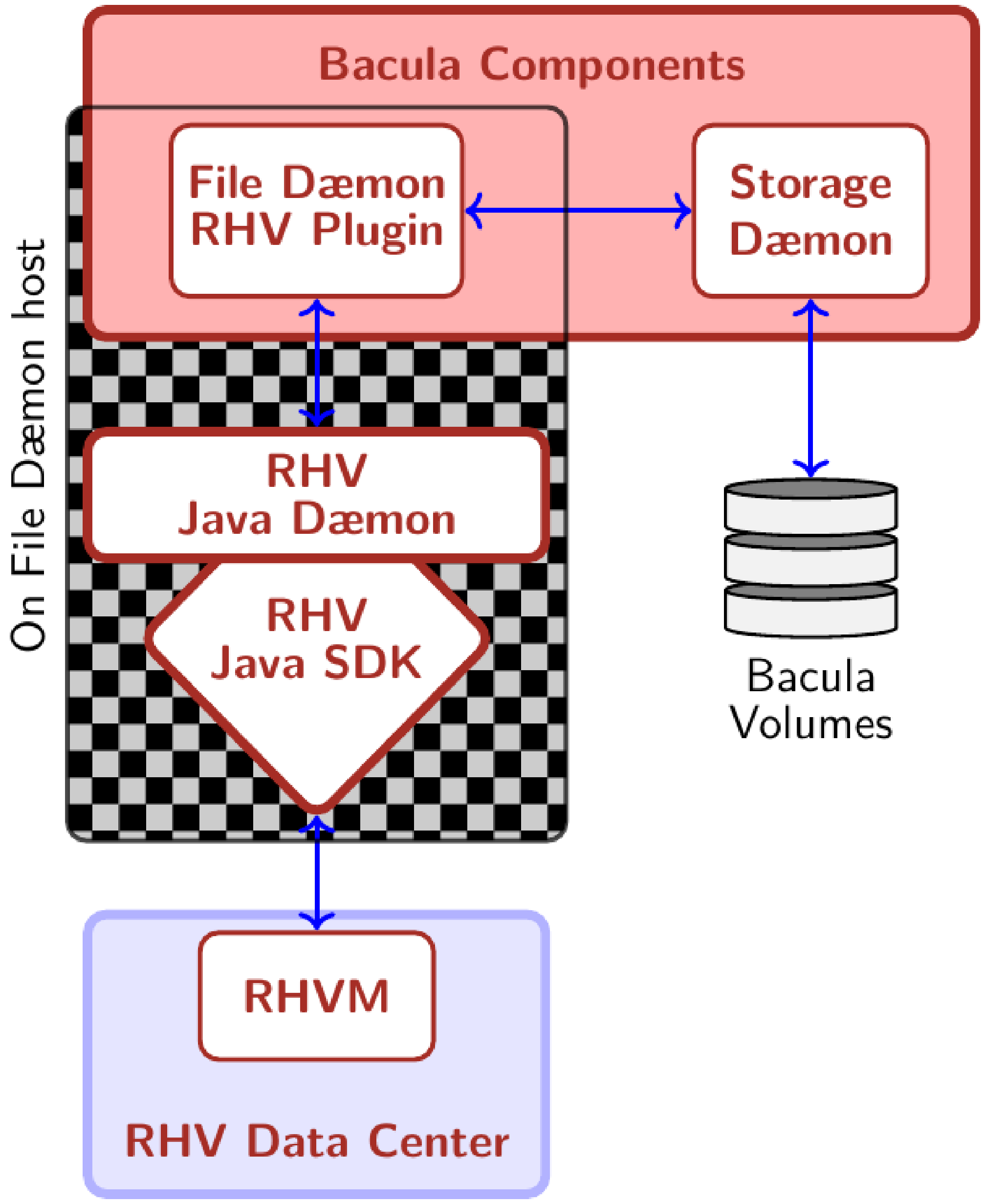


Fig. 18: Plugin Architecture

- Download all disks of template.

5. If it is a snapshot backup:

- Download configuration of virtual machine (XML).
- Download all virtual machine snapshot disks.

6. If it is a backup clone:

- If necessary, remove clone machine.

7. If it is not a template backup and **cbt is off**:

- Remove snapshot.

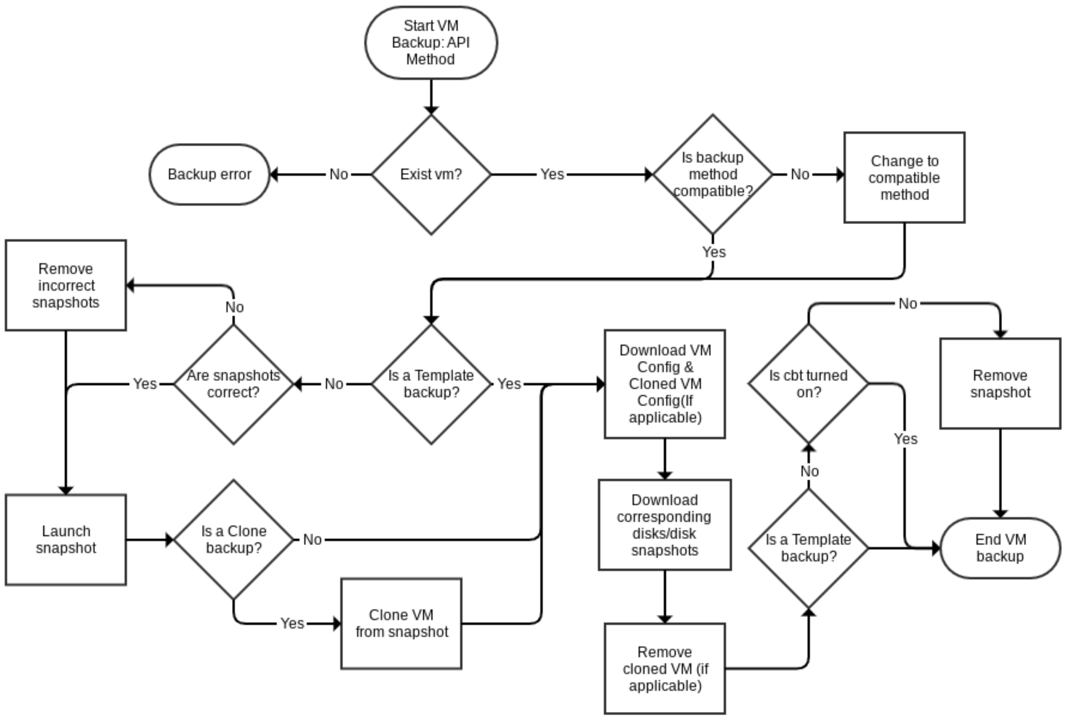

Start VM Backup: API Method
Backup error
No
Exist vm?
Yes
Is backup method compatible?
No
Change to compatible method
Yes
Remove incorrect snapshots
No
Yes
Are snapshots correct?
No
Is a Template backup?
Yes
Download VM Config & Cloned VM Config(If applicable)
No
Is cbt turned on?
Remove snapshot
Yes
No
Launch snapshot
Is a Clone backup?
No
Download corresponding disks/disk snapshots
Is a Template backup?
End VM backup
Yes
Clone VM from snapshot
Remove cloned VM (if applicable)

Fig. 19: External backup diagram

Incremental and Differential Backups

Incremental Backup Process

An incremental backup is one in which successive copies of the data will contain only the portion of the disk(s) that has changed since the previous backup was performed. When a full recovery is needed, the restoration process will need the last full backup plus all the incremental backups up to the point of restoration. Incremental backups are often desirable as they reduce storage space usage, and are quicker to perform than full or differential backups.

The method to perform incremental backups used by this plugin is as follows:

- CBT function is activated in the fileset.
- The first FULL will leave the snapshot, which it created during backup, in the machine.

- Each Incremental will perform a new snapshot, take the data from that new snapshot (which will contain the difference between the current state and the snapshot immediately before) and eliminate the snapshot created by the previous Incremental backup.

The disadvantages of this process are the following:

- A longer time is required during a restore since the Full and all Incrementals will need to be read.
- It is recommended to perform closed cycles that are not very long with the following sequence: 1 FULL + X INCREMENTAL, with 'X' being the number of backups in the chain. This recommendation is due to the fact that in each iteration that performs an incremental backup the snapshot that is maintained in the virtual machine is larger, and therefore the operations carried out with it will take longer and longer, and more data will be stored. (See Table *A normal incremental backup*).

Differential Backup Process

A differential backup is a type of data backup which only backs up the difference in the data since the last full backup. The rationale in this is that since changes to data are generally few compared to the entire amount of data in the data repository, the amount of time required to complete the backup will be smaller than if a full backup was performed every time that the organization or data owner wishes to back up changes since the last full backup. Another advantage, at least as compared to the incremental backup method of data backup, is that at data restoration time, at most two backup media are ever needed to restore all the data (The last Full + the last Differential). This simplifies data restores as well as increases the likelihood of shortening data restoration time.
The method to perform incremental backups used by this plugin is as follows:

- CBT function is activated in the fileset.
- The first FULL will leave the snapshot, which it created during backup on the machine. This full snapshot will be present in all differential backup processes until a new full is performed.
- Each differential will make a new snapshot. It will take the data of that new snapshot (which will contain the difference between the Full and the current state) and eliminate the snapshot at the end.

The advantages of this level of backup over INCREMENTAL are:

- Faster restorations.
- Unlimited iteration of differential backups without influencing backup performance.

In Figure **fig:incr-diff-backup-diagram** you can see 3 marks with the letters A, B and C. These marks correspond with:

1. Check previous backups with internal RhvCatalog stored in */opt/bacula/ working/rhv/catalog/{server_parameter}/{vm_id}*.json. The function of this catalog is to control the backups of each VM to check if they finish correctly. If this catalog entry doesn't exist for the VM, the plugin creates it with a **FULL** backup.
2. Incremental/Differential backups need a chain of snapshots. The base snapshot is the snapshot of the last Full backup done by the RHV plugin for a virtual machine. The description of the snapshots created by the RHV plugin is *backup_{jobId}_{jobName}*.
3. If the snapshot chain is interrupted by any snapshot created outside of the RHV plugin, the RHV plugin must make a **COMPLETE** backup. The RHV plugin will always keep foreign snapshots.

Use cases in snapshot maintenance

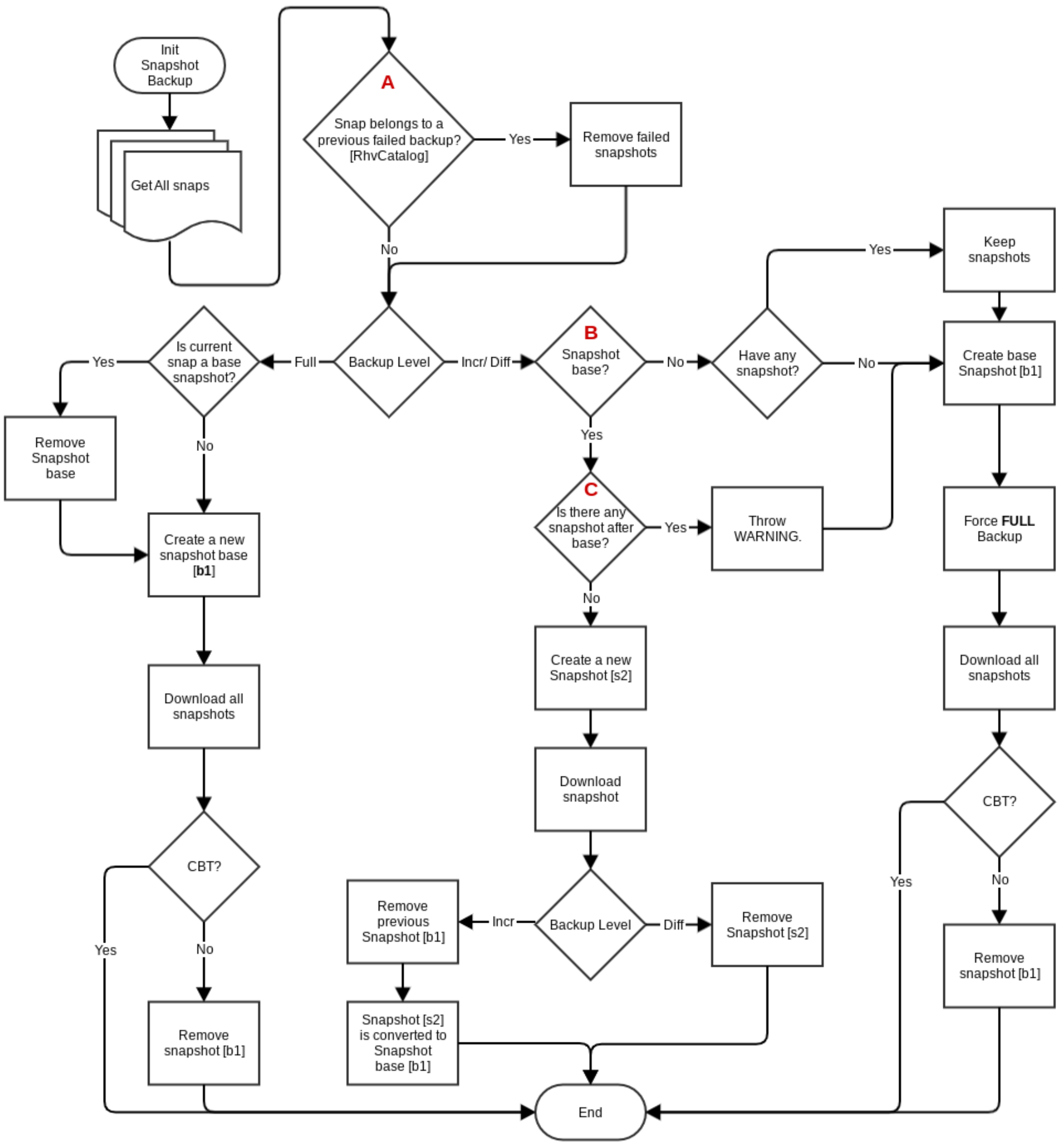


Fig. 20: Incremental/Differential backup diagram

Table 16: A normal incremental backup

	T1	T2	T3	T4	T5
Type Backup	F	I	I	I	I
Create snapshot	S1	S2	S3	S4	S5
Snapshots in VM	S1	S2	S3	S4	S5

Table 17: A normal differential backup

	T1	T2	T3	T4	T5
Type Backup	F	D	D	D	D
Create snapshot	S1	S2	S3	S4	S5
Snapshots in VM	S1	S1	S1	S1	S1

Table 18: A change of type backup

	T1	T2	T3	T4	T5
Type Backup	F	D	D	F	I
Create snapshot	S1	S2	S3	S4	S5
Snapshots in VM	S1	S1	S1	S4	S5

Table 19: An external snapshot in T1

	T1	T2	T3	T4	T5
Type Backup		F	I	I	I
Create snapshot	E1	S2	S3	S4	S5
Snapshots in VM	E1	E1 + S2	E1 + S3	E1 + S4	E1 + S5

Table 20: An external snapshot between backups

	T1	T2	T3	T4	T5
Type Backup	F	I		I *	I
Create snapshot	S1	S2	E3	S4	S5
Snapshots in VM	S1	S2	S2 + E3	S2 + E3 + S4	S2 + E3 + S5

I *: Internally, the RHV plugin will do a FULL backup and log a warning.

Backup from RHV environment internal machine

A proxyVM backup is a method where backup is done through a special VM inside the RHV environment. Disks belonging to VMs that need to be backed up will be attached to this special VM and backed up via this proxyVM. This method only allows FULL backups, but is much faster than the other REST API based methods.

The backup process consists of binding to each of the virtual machine disks to perform the backup, backing up of the contents of the disks, then unbinding from them. The binding process is made through a snapshot and an attach operation of the snapshot to the proxy VM.

Backup Cache

This feature keeps VM backups in different storage domains to be used for future restores. This functionality allows faster restores of virtual machines, but requires more space available in the storage domains.

Quiescing VMs

If the guest agent for RHV is installed in a VM, the system will be quiesced automatically and transparently when a snapshot is created. Red Hat's documentation at redhat.com/.../Live_Snapshots_in_Red_Hat....html provides full information.

Restore Process

The general restore process of a virtual machine consists of

1. Create a new VM using an XML configuration.
2. Create all of the VM's disks.
3. Create every snapshot, if necessary.
4. Upload disks.
5. Create the VM's network interfaces.
6. Create the template from the virtual machine, if necesary.
7. Switch on the virtual machine if necessary.

The disk upload operations are performed using the ImageTransfer service of oVirt.

Features

The following lists present the general features of this plugin.

Backup Features

- Full image-level backup.
- Incremental backup.
- Differential backup.
- Proxy backup.
- Agentless backup.
- RHV snapshot integration to create a snapshot to back up, and always delete this snapshot when the backup finishes, when the plugin finishes its work, or when the next backup is initiated if a prior backup error occurred.
- Snapshot consistency through quiescing VMs for backup.
- Backup storage cache in Red Hat Virtualization Manager (Where we keep cloned VMs).
- Individual VM Backups.
- VM derivated of template Backups.
- Individual Template Backups.
- Optional VM configuration only backups.
- Disk exclusion.

- Selection of VMs to back up by tags, regular expression matching on name, or per any structure RHV *Querying RHV*.
- Backup of virtual machines in a “running”, “paused” or “shut off” state.
- VM exclusion.
- Password obfuscation.
- Failed backup control:
  - Find and remove previous failed backups / snapshots with every execution.
  - Backup cancellation and connection loss controls to try to remove snapshots or clones of current backup.
  - Control of snapshots by an internal catalog.

Restore Features

- Restore to original or different location in RHV environment (Cluster or Storage Domain).
- Replace original VM or create a new VM with a new name.
- Restore templates.
- Change destination VM name.
- Local restore of disk images and VM XML configuration to allow manual processing.
- Exclude disks from restore.
- Turn VM on after restore, or leave turned off.
- Restore disk configuration:
  - Virtual disk interface (ide, virtio, virtio_scsi, spart).
  - “Active” state of disk.
  - “Boot disk” flag.
  - Template to name restored disks.
  - Name list of restored disks.
- Restore NIC configurations (providing MAC address list).

Limitations

The plugin currently does not support the following features:

- The backup snapshot method is not compatible with RHVM4.1.
- Virtual Machines in Suspended status cannot be backed up. This is oVirt/RHV limitation as do a snapshot is not allowed with that state.
- The Incremental/Differential backup is only compatible with snapshot backups.
- The Incremental/Differential backup is compatible with RHV Plugin version >= 4.0.0.
- The proxy backup is compatible only with Bacula Enterprise version >= 12.4.0.
- Virtual Full backups.

- Using an external OAuth / SSO service different from the one provided directly by RHV.
- Stop or Resume incomplete Jobs. For this to be possible with the guarantee of consistent backups, snapshots of virtual disks and cloned VMs would need to be kept.
- Optimized backup with thin disks. The Ovirt API doesn't provide any information about thin disk usage, except with thin disks located in iSCSI storage.
- Restores from a backup sequences based on a Full and Differential and/or Incremental snapshots of VMs that contain at least one disk in a **Block Based Storage Domain** are possible only with RHV versions 4.4.5 or newer.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Compatibility and Requirements

Compatibility

The plugin is currently developed exclusively for RHEL Virtualization 4.1 and above environments (RHEL 7-based). It has been tested only on those platforms. Other virtualization platforms based on oVirt, and sufficiently similar to the RHV platform, may also be suitable.

Bacula Systems will add compatible platforms to this list when interoperability has been demonstrated through its development and QA processes. Please contact Bacula Systems Support if you have questions about the currently supported RHV versions.

Requirements

Bacula

The plugin works with version 10.1 or newer of Bacula Enterprise with **non Accurate mode** (The **Accurate** Job directive must be disabled).

Java

The server hosting the Bacula File Daemon must have the Java Virtual Machine version 8 or newer installed.

Red Hat Virtualization Considerations

Disk Download

In order to allow correct disk image downloads during the backup processes, and due to an existing bug of RHV, the following configuration changes must be made in RHV:

For RHV 4.1 (Compatible with RHV 4.2)

Connect to the RHV Manager host using `ssh`, then

```
$ su - postgres
$ psql -U postgres -d engine
# Get the existing value for future reference
psql (12.3)
Type "help" for help.

```

(continues on next page)

(continued from previous page)

```
postgres=# SELECT * FROM vdc_options WHERE
option_name='ImageTransferClientTicketValidityInSeconds';
UPDATE vdc_options SET option_value=999999
WHERE option_name='ImageTransferClientTicketValidityInSeconds';
```

The value of `option_value` represents the validity time in seconds of the token to access disk images for download or upload. We recommend to set the life time to be virtually unlimited with the value 999999 in the above query.

RHV 4.2

Connect to the RHV Manager host using `ssh`, then

```
# Get the existing value for future reference
engine-config --get=ImageTransferClientTicketValidityInSeconds
# Set value (we recommend 999999 seconds, virtually unlimited)
engine-config --set=ImageTransferClientTicketValidityInSeconds=999999
```

Storage Space

If RHV version 4.2 is used, additional backup storage space is not required, but if RHV version 4.1 is used, or the parameter 'apply clone' is set, as discussed in section *Backup Process*, backups need to perform a complete clone of a snapshot of each virtual machine being backed up. This clone is removed after a backup or at the start of a subsequent backup in case the previous one failed without the plugin having a chance to clean up. Therefore it is necessary to have sufficient storage space in the Storage Domain the backup runs in. It is also important to consider how many backups are going to be run in parallel and reserve enough space for those temporary clones.

Use of computer resources

Proxy Backup

The process will run in a virtual machine that it is placed inside the same RHV environment we are protecting. It will be necessary to deploy this VM which contains the Bacula agent and proxy VM. This virtual machine has some minimal requirements:

- **OS:** There is no specific OS requirement and this mode should work with any updated OS. However, it is recommended to use Debian 10 (Buster), as it is the only one where proxy mode has been tested extensively.
- **RAM:** 2 GB.
- **Network:** Access to logical network RHV.
- **Storage:** 4GB.
- **Bacula Enterprise version:** 12.4.0.
- **Java Virtual Machine:** 8.

Calculations have been made that determine that the machine with the minimum requirements can process up to 20 backups in parallel. It will be necessary to add 1 GB of RAM for every 10 backups in parallel.

API Backup

This process will execute in a external machine, and it has been calculated that 1 GB of available RAM is needed for every 10 paralel backups for this process.

Configuration

Plugin Installation

The **Bacula Enterprise** RHV plugin is distributed as a standard package for every supported operating system.

After installing the package, a few aspects must be considered: The Plugin Directory directive of the File Daemon resource in `/opt/bacula/etc/bacula-fd.conf` has to point to where the `rhv-fd.so` plugin is installed. The usual Bacula plugin directory is `/opt/bacula/plugins`. Thus, the configuration file `bacula-fd.conf` should look like in the following example:

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Certificate and Truststore

There are two ways to create the truststore that allows the Bacula plugin to connect to RHV in secure mode.

Automatic

Call plugin

The plugin has an option to create the truststore automatically. To create the truststore run the following command:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --server=myrhv.com --
↪operation=system
  --create_truststore=true --truststore_file=/opt/bacula/working/rhv_
↪truststore
  --truststore_password=changeit
```

Example command to create the truststore

```
user@host:~/\$ java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --server=
↪{server}
  --operation=system --create_truststore=true --truststore_file={truststore_
↪path}
  --truststore_password={truststore_password}
```

Interactive script

There is an interactive script in `/opt/bacula/scripts/rhv_config.sh`. When the script is executed it requests parameters such as: server, truststore path, truststore password, alias in truststore and keytool Java path. Only the parameter 'server' is required. The others parameters by default are:

- Truststore path: `/opt/bacula/etc/rhv.cacerts`
- Truststore password: changeit
- Truststore internal alias: rhvPluginX<randomNumber(1-100000)>
- Path to Java's keytool: `/usr/bin/keytool`

Example of a script execution:

```
[root@rhv-client tmp]# /opt/bacula/scripts/rhv_config.sh
Welcome wizard to create TrustStore File
Enter FQDM Red Hat Virtualization Manager:rhv-mgmnt.local.lan
Path truststore (/opt/bacula/etc/rhv.cacerts):
Password truststore (changeit):
Alias (rhvPluginX89803):
Path keytool ('/usr/bin/keytool'):


Resume:
Server:  rhv-mgmnt.local.lan
Path Truststore:  /opt/bacula/etc/rhv.cacerts
Pass Truststore: default
Alias:  rhvPluginX89803
Keytool:  /usr/bin/keytool


Are you sure? [no] yes
Truststore generate successfully
[root@rhv-client tmp]#
```

Manual

To generate the truststore that allows the Bacula plugin to connect to RHV, the RHV server certificate needs to be downloaded and added to the existing Java truststore or a new one used by the plugin.

In the following examples, the address will be used; it needs to be replaced with the proper one.

To install the RHV server certificate on the File Daemon host, the certificate can be prepared in different ways:

- Download using http, as in

```
# beware - line broken below!
curl -o rhvm.cer 'http://rhv.example.com/ovirt-engine/services/
pki-resource?resource=ca-certificate&format=X509-PEM-CA'
```

- Edit a new certificate file and paste the needed content:
  1. Log in to the Bacula File Daemon host.
  2. Use `ssh` to connect to the RHV Manager.
  3. `cat /etc/pki/ovirt-engine/ca.pem`
  4. Copy the contents from the “BEGIN CERTIFICATE” line, up to and including the “END CERTIFICATE” one.
  5. Log out from the RHV host.
  6. Edit a new text file: For example `vim rhvm.cer`
  7. Paste the copied text. (With `vim` in a typical Linux terminal session, the key strokes would be “i shift-ctrl-v escape : x”.)

Once the certificate file, called `rhvm.cer` in the example above, is available, it be added to a Java truststore file. The truststore path and password are general required parameters of this plugin.

It is possible to use an existing Java truststore (including the local default file), or to create a new one. For this example, a new one is created:

```
keytool -import -alias "mirhvm.crt_truststore" \
 -file rhvm.cer -keystore rhvm.truststore
# A prompt will ask for a password
```

To use the existing, default Java truststore, the command would be as follows:

```
keytool -import -alias "mirhvm.crt_truststore" \
 -file rhvm.cer -keystore $JAVA_HOME/jre/lib/security/cacerts
# This will ask for a password, default value is: changeit
```

Note: By default, the Java truststore path is: `$JAVA_HOME/jre/lib/security/cacerts`

Please refer to the relevant Java documentation for an explanation of the `keytool` command, or Java truststore in general.

Note: The full path to the truststore file used, as well as the password, are needed for further plugin configuration steps, as well as for its operation.

Without Certificate

The plugin also allows a connection without certificate. It is discouraged to use this mode in production environments. To run the plugin in this mode, the plugin must be execute with the option "insecure", explained in *Plugin Parameters*.

User Permissions

It is possible to use the RHV "admin" user for the plugin. But, in general, IT organizations should prefer using a specific backup user account with more restricted permissions. In order to have full functionality of this plugin, the user account that will be used must have the following RHV permissions:

Table 21: RHV User Permissions

Category	Required setting
RHV Special Roles	ExternalEventsCreator
	UserVmManager
System, *Configure System*	Login Permissions
Template, *Provisioning Operations*	Import/Export
VM	Run VM
VM, *Provisioning Operations*	Edit properties
	Create
	Create Instance
	Delete
	Edit Storage
	Edit properties
Disk, *Provisioning Operations*	Create
	Delete
Disk, *Provisioning Operations*, *Edit Storage*	Attach
	Manipulate SCSI I/O Privileges

Plugin Parameters

Below all options and parameters that can be used in a Fileset plugin line with the RHV plugin are described.

Options which have default values don't need to be defined as long as their default behavior is desirable.

Note: To create obfuscated passwords, the Bacula administrator should execute the command `@encode password` in `bconsole` and copy the result, or execute it directly via the plugin. This application is explained in section *Plugin System*.

Warning: When orthogonal VM selection schemes are combined, the plugin only will back them up once. For example: **target_datacenter=dc1** and **target_virtualmachine=vm1,vm2** where **vm1 is in dc1**, and **vm2 is in dc2**, the plugin will backup **dc1** and **vm2**, and **vm1 only backup once**.

General plugin parameters

server

- Required: Yes
- Default:
- Info: DNS name of RHV~Manager (the server parameter requires a domain name, not an URI, nor are IP addresses allowed).
- Example: `server=rhv.example.com`

truststore_file

- Required: Yes
- Default:
- Info: File name of truststore
- Example: `truststore\_file=/opt/bacula/etc/rhv-trust`

truststore_password

- Required: No
- Default: changeit
- Info: Truststore password
- Example: `truststore\_password=changeit`

truststore_hpassword

- Required: No
- Default: changeit
- Info: Obfuscated truststore password, use either the obfuscated or plain password
- Example: `truststore_hpassword=NTUyOjU0NDpRR1xaR0VJWwA`

insecure

- Required: No
- Default: false
- Info: Mode insecure, without truststore. Not recommended in production environment
- Example: `insecure=yes`

user

- Required: No
- Default: admin
- Info: RHV user account to authenticate as
- Example: `user=bacula`

password

- Required: Yes
- Default:
- Info: User password
- Example: `password=rhvpass123`

hpassword

- Required: Yes
- Default:
- Info: Obfuscated user password
- Example: `hpassword=NTUyOjU0NDpRR1xaR0VJWwA`

auth

- Required: No
- Default: http
- Info: Authentication type, `http` or `oauth`
- Example: `auth=oauth`

config_file

- Required: No
- Default:
- Info: Path to configuration file
- Example: `config_file=/opt/bacula/etc/rhv.conf`

profile

- Required: No
- Default: internal
- Info: Connection user profile
- Example: `profile=internal`

abort_on_error

- Required: No

- Default: 1
- Info: Abort the backup in case of any problem with disks implied or previous backup files (0 or 1)
- Example: `abort_on_error=0`

output

- Required: No
- Default: json
- Info: Helper program output format, one of `json`, `bacula`, `console`
- Example: `output=console`

log

- Required: No
- Default: ./rhv-plugin.log
- Info: Log to specified file
- Example: `log=/tmp/rhvbkup.log`

debug

- Required: No
- Default: 0
- Info: Enable debug output: 0 (informational), 1 (debug), 2 (deep debug)
- Example: `debug=1`

Tuning parameters

system_num_tries_connection

- Required: No
- Default: 5
- Info: Number of failed attemps in http requests to RHV/oVirt server before failing
- Example: `system_num_tries_connection=10`

system_interval_request

- Required: No
- Default: 120
- Info: Interval between http request to RHV/oVirt server (seconds)
- Example: `system_interval_request=360`

system_timeout_request

- Required: No
- Default: 120
- Info: Timeout for http requests done to RHV/oVirt server (seconds)
- Example: `system_timeout_request=360`

Backup specific parameters

target_datacenter

- Required: No
- Default:
- Info: Back up VMs from Data Center
- Example: `target_datacenter=production`

target_cluster

- Required: No
- Default:
- Info: Back up VMs from specified Cluster
- Example: `target_cluster=webfarm`

target_tag

- Required: No
- Default:
- Info: Back up VMs with specified tag
- Example: `target_tag=bkup`

target_virtualmachine

- Required: No
- Default:
- Info: Comma-separated list of VM names to back up
- Example: `target_virtualmachine=web1,web2,db1`

target_path

- Required: No
- Default:
- Info: Backup all vms of children path. Support comma-separated list.
- Example: `target_path=/datacenters/Default/clusters/Default/,/tags/tag1/`

target_template

- Required: No
- Default:
- Info: Comma-separated list of Template names to back up
- Example: `target_template=template1,temp2,temp3`

target_exclude_disks

- Required: No
- Default:
- Info: **In testing**, comma-separated list of disk ids to exclude

- Example: `target_exclude_disks=65ccb526-30c0-4bebb348-4395e70d6e32, 3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c`

target_exclude_vms

- Required: No
- Default:
- Info: Comma-separated list of VM names to exclude
- Example: `target_exclude_vms=web,db2`

target_configuration_only

- Required: No
- Default: false
- Info: Back up only VM configuration, not disk images, `true` or `false`
- Example: `target_configuration_only=true`

clone_storage

- Required: No
- Default: false
- Info: Enable clone mode backups
- Example: `clone_storage=on`

restorecache_retention

- Required: No
- Default: 0
- Info: Number of clones to keep in RHVM
- Example: `restorecache_retention=3`

restorecache_storage_domain

- Required: No
- Default:
- Info: Target storage to keep backup clone
- Example: `restorecache_storage_domain=dataMaster2`

cbt

- Required: No
- Default: false
- Info: Active incremental or differential backup, available since Bacula Enterprise version 12.0.2 or newer
- Example: `cbt=true`

proxy_vm

- Required: No
- Default:

- Info: Virtual machine to proxy backup
- Example: `proxy_vm=ProxyVmBackup`

proxy_block

- Required: No
- Default: true
- Info: Safely attach/detach disks in proxy backup
- Example: `proxy_block=true`

persist_memory_state

- Required: No
- Default: false
- Info: Save the memory state (or not) when taking a live snapshot for the backup. During a live snapshot with memory creation, the VM will be locked to prevent it from being changed. When the memory dump is being saved, the VM will be switched to the 'paused' state. Available since Bacula Enterprise 14.0.5. previously, version memory was persisted by default.
- Example: `persist_memory_state=true`

Restore-Specific Parameters

Note: These are usually set interactively during restore, **not** in a plugin line in a file set.

The correct compatibility [vm, cluster, storageDomain] is the responsibility of the administrator.

---

vm_name

- Required: No
- Default: suffix
- Info: Name of restored VM, `suffix` and `original` have special meaning (see chapter *Restore* for details)
- Example: `vm_name=restoredVM`

template_name

- Required: No
- Default: suffix
- Info: Name of restored Template, `suffix` and `original` have special meaning (see chapter *Restore* for details)
- Example: `template_name=restoredTemplate`

vm_disks

- Required: No
- Default:
- Info: Comma-separated list of disks to restore (skip all others)
- Example: `vm_disks=3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c`

vm_exclude_disks

- Required: No
- Default:
- Info: **In testing**, comma-separated list of disks to exclude
- Example: `vm_exclude_disks=3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c`

cluster

- Required: No
- Default: original
- Info: Cluster to restore to, uses value from backup
- Example: `cluster=secondary`

storage_domain

- Required: No
- Default: original
- Info: Storage Domain name to restore to: For specific cases that affect only one disk, you should use the id of the disk (you can use the restore prompt) delimited by ':' and the value. For the general case, no id or ':' should be used.Separate each case by ','.
- Example: `storage_domain=3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c:secondary, primary`

vm_force_overwrite

- Required: No
- Default: false
- Info: Overwrite existing target VM, `true` or `false`
- Example: `vm_force_overwrite=true`

vm_switchon

- Required: No
- Default: false
- Info: Turn on restored VM, `true` or `false`
- Example: `vm_switchon=true`

remove_base_vm

- Required: No
- Default: true
- Info: In template restore or vm restore with create template, remove vm restored after create template (see chapter *Restore* for details)
- Example: `remove_base_vm=false`

disk_interface

- Required: No
- Default: original

- Info: Interface types for restored disks; comma-separated list of `original`, `ide`, `virtio`, `virtio_scsi`, `spart`
- Example: `disk_interface=3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c:original,ide`

disk_active

- Required: No
- Default: original
- Info: Comma-separated list of `original`, `true`, `false` flags to make disks active
- Example: `disk_active=true,3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c:false`

disk_boot

- Required: No
- Default: originalfootref{fn:source}
- Info: Comma-separated list of `original`, `true`, `false` flags to make disks bootable
- Example: `disk_boot=true,3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c:false`

disk_names

- Required: No
- Default: original
- Info: Comma-separated list of names for restored disks
- Example: `disk_names=3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c:DiskInactive1, restOfDisks`

nics

- Required: No
- Default: new
- Info: Comma-separated list (one entry per virtual NIC) of MAC addresses or `original`, `new`
- Example: `nics=C0:BA:C0:7A:CA:FE,original`

System-Specific Parameters

create_truststore

- Required: No
- Default: false
- Info: Create truststore of server RHVM (only use with Java Daemon), see *Plugin System* for details
- Example: `create_truststore=true`

system_num_tries_connection

- Required: No
- Default: 5
- Info: Number failed connection tries before throw an Error more important, see section *System operations* for details}
- Example: `system_num_tries_connection=10`

system_interval_request~footref{fn:systemopts}

- Required: No
- Default: 120
- Info: Time(seconds) between failed try and other try
- Example: `system_interval_request=60`

system_timeout_request

- Required: No
- Default: 120
- Info: Timeout(seconds) to request server
- Example: `system_timeout_request=60`

Configuration File

The plugin parameter `config_file` allows for the creation of a file which can be used to provide all necessary options instead of putting them into the Fileset's RHV plugin line. These configuration files are plain text files containing one setting per line where the name of the setting and its value are separated by an equals sign "=". An example configuration file could look like this:

```
server=rhv.example.com
user=admin
truststore_file=/opt/bacula/etc/rhvm.truststore
target_datacenters=myDatacenter
operation=backup
```

If a parameter is mentioned more than once, the last instance is used.

Fileset Configuration

A Fileset to back up RHV VMs will usually contain one **plugin=rhv:...** line, and may contain **File=...** directives.

It is recommended to have only a single RHV plugin line per Fileset, which will ensure that no unpredictable behaviour in case of errors such as failure to proceed with one plugin call, but success with all others.

Bacula Systems strongly advises against the use of other plugins in Filesets which use the RHV plugin.

Backup Configuration Examples

The minimal Fileset. This Fileset will backup all vms in RHV environment:

```
Fileset {
  Name = minimalFileset
  Include {
    Plugin = "rhv: server=rhv.example.com insecure=true password=rhvpass123"
  }
}
```

The minimal Fileset using proxy backup method, using the *PROXY_BACKUP_VM*. This Fileset will backup all VMs in the RHV environment (includes proxy backup VM):

```
Fileset {
  Name = minimalFilesetProxy
  Include {
    Plugin = "rhv: server=rhv.example.com insecure=true password=rhvpass123␣
↪proxy_vm=PROXY_BACKUP_VM"
  }
}
```

The minimal backup Job. This Job uses the minimal Fileset above, so it will backup all VMs in the RHV environment:

```
Job {
  Name = minimalJob
  Type = Backup
  Level = Full
  Write Bootstrap = "/var/bacula/working/minimalJob.bsr"
  Accurate = no
  Client = Client1
  Maximum Concurrent Jobs = 1
  Allow Duplicate Jobs = no
  Fileset = minimalFilesetProxy
  Storage = Storage1
  Pool = Pool1
  Messages = Standard
  Enabled = yes
}
```

Backing up only "vmExample" can be achieved like this:

```
Fileset {
  Name = ExampleFileset0
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv: server=rhv.example.com insecure=true password=rhvpass123␣
↪target_virtualmachine=vmExample"
  }
}
```

A backup of the VM named "vm1" and all vms that the datacenter "dc1" contains, excluding the disks with ids 65ccb526-30c0-4beb-b348-4395e70d6e32 and 3b874e8c-ddbb-4e1e-a4fc-3cdf0b76ec1c, would be configured like this:

```
Fileset {
  Name = ExampleFileset1
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv:server=rhv.example.com
```

(continues on next page)

(continued from previous page)

```
      truststore_file=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/
↪cacerts/rhvm.truststore
      auth=http user=admin password=rhvpass123 target_virtualmachine=vm1␣
↪target_datacenter=dc1
      target_exclude_disks=65ccb526-30c0-4beb-b348-4395e70d6e32,3b874e8c-ddbb-
↪4e1e-a4fc-3cdf0b76ec1c"
  }
}
```

The backup of all VMs with names starting with “vm”, excluding “vm1”:

```
Fileset {
  Name = ExampleFileset2
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv:server=rhv.example.com truststore_password=changeit␣
↪auth=http profile=internal
    truststore_file=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/
↪cacerts/rhvm.truststore
    user=admin password=rhvpass123 target_virtualmachine=vm* target_exclude_
↪vms=vm1"
  }
}
```

The backup of all templates with names starting with “template”:

```
Fileset {
  Name = ExampleFileset3
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv:server=rhv.example.com truststore_password=changeit␣
↪auth=http profile=internal
    truststore_file=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/
↪cacerts/rhvm.truststore
    user=admin password=rhvpass123 target_template=template*"
  }
}
```

Backing up only “vm1” with backup proxy method, using the vm *PROXY_BACKUP_VM*:

```
Fileset {
  Name = ExampleProxyFileset
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv: server=rhv.example.com insecure=true password=rhvpass123␣
↪target_virtualmachine=vm1 proxy_vm=PROXY_BACKUP_VM"
```

(continues on next page)

(continued from previous page)

```
  }
}
```

Restore

Restores of backups done using the RHV plugin are initiated like restores of any file-based backups. The File Daemon used for the restore, indicated by the Client selected as restore target, must have the RHV plugin installed in order to allow restores to RHV directly.

From version 3.0.0, the plugin allows restores of templates. This integration is compatible with restore of vms. And also it allows create a template from a restored vm, directly. Only the appropriate parameters should be used.

From version 4.0.0 (**Bacula Enterprise** version 12.4.0), the plugin allows restores of incremental and differential backups, and proxy backups.

If the `where` parameter of a restore of a RHV plugin backup is set to indicate the original location, the restore will be passed directly to RHV Manager. An example would be a `bconsole` command line such as:

```
restore job=one-RHV-VM where=/
```

If the restore target location is set to anything else, the selected data will be restored to the path indicated. VM or TEMPLATE configuration will be in an XML file, and virtual disk images will be stored in the format applicable, depending on format at backup time and restore options set. This operation is not necessary use the truststore file.

When selecting VMs or TEMPLATEs to restore, these will be represented as sub-directories in a common root directory `@rhv/`. Only one VM or one TEMPLATE (one subdirectory) should be restored at a time, and for restores to RHV directly (as opposed to a plain file), the whole subdirectory needs to be `marked`.

The treatment between templates and vms is done transparently to the user. Therefore, the user should know if it is a job with a template backup or vm backup. The only way to know if a backup was from a vm or a template is by looking at the name of a file inside the directory. This file will be named as: "nameVm|nameTemplate_vm|template_BaculaRhvBackup.xml". Such as:

```
exampleVm_vm_BaculaRhvBackup.xml
exampleTemplate_template_BaculaRhvBackup.xml
```

The options that were used during a backup job are stored in Bacula's catalog database and they will applied during a restore session. To provide full flexibility, these options can be modified, and some options specific to restores are also available. These options are documented, along with all other plugin options, in table *Restore-Specific Parameters*.

Note that some options will be used to pass lists of values to the plugin. Such lists consist of comma-separated values. If a list does not contain enough values to be applied to all related objects, the default value as shown in table *Restore-Specific Parameters* is used for the missing ones.

Often, special values are available to cause certain behaviour; such options are explained below, in detail.

When using `bconsole`, the options can be modified using the option 13 ("Plugin Opts") of the "modify" menu.

An example `bconsole` restore session of a RHV backup job that saved a single VM, with one disk won't be active but all of them yes, is shown below:

```
Connecting to Director 127.0.0.1:8101
1000 OK: 103 127.0.0.1-dir Version: 9.0.6 (20 November 2017)
Enter a period to cancel a command.
restore jobid=926 Client=RHV-fd where=
Using Catalog "MyCatalog"
You have selected the following JobId: 926
Building directory tree for JobId(s) 926 ...
6 files inserted into the tree.
You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.
cwd is: /
$ cd @rhv/
cwd is: /@rhv/
$ ls
vmTest/
$ cd vmTest/
cwd is: /@rhv/vmTest/
$ ls
DiskSnap_513901a8-095e-4bbc-a6d3-b082e91fe64c_151736db-dba5-491b-af39-
↪537914f4a176_0.img
DiskSnap_513901a8-095e-4bbc-a6d3-b082e91fe64c_e3d2062a-d463-4271-9c8a-
↪ff85a8d73d3c_1.img
DiskSnap_58efd04d-48e8-4ab6-9d9c-42247798fd31_151736db-dba5-491b-af39-
↪537914f4a176_4.img
DiskSnap_58efd04d-48e8-4ab6-9d9c-42247798fd31_e3d2062a-d463-4271-9c8a-
↪ff85a8d73d3c_5.img
DiskSnap_d382cd50-f052-4621-950a-4692200012e7_151736db-dba5-491b-af39-
↪537914f4a176_2.img
DiskSnap_d382cd50-f052-4621-950a-4692200012e7_e3d2062a-d463-4271-9c8a-
↪ff85a8d73d3c_3.img
vmTest_attachs_BaculaRhvBackup.xml
vmTest_config_BaculaRhvBackup.json
vmTest_disks_BaculaRhvBackup.xml
vmTest_dsnaps_BaculaRhvBackup.xml
vmTest_nics_BaculaRhvBackup.xml
vmTest_snapshots_BaculaRhvBackup.xml
vmTest_vm_BaculaRhvBackup.xml
$ cd ..
cwd is: /@rhv/
$ mark vmTest
13 files marked.
$ done
$
Bootstrap records written to
/opt/bacula/working/bacula-dir.restore.6.bsr
The Job will require the following (*=>InChanger):
Volume(s)                    Storage(s)            SD Device(s)
===========================================================================
poolrhv-0415                RestoreStorage
Volumes marked with "*" are in the Autochanger.
```

(continues on next page)

(continued from previous page)

```
6 files selected to be restored.
Using Catalog "MyCatalog"
Run Restore job
JobName:          RestoreFiles
Bootstrap:        /opt/bacula/working/bacula-dir.restore.6.bsr
Where:
Replace:          Always
Fileset:          Full Set
Backup Client:    RHV-fd
Restore Client:   RHV-fd
Storage:          File
When:             2018-05-22 10:32:51
Catalog:          MyCatalog
Priority:         10
Plugin Options:   *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
1: Level
2: Storage
3: Job
4: Fileset
5: Restore Client
6: When
7: Priority
8: Bootstrap
9: Where
10: File Relocation
11: Replace
12: JobId
13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : rhv:"abort_on_error=0"
"server=cubietruck1.dtic.ua.es"
"truststore_file=/opt/bacula/etc/rhvm.truststore"
"truststore_password=sosecret!" "auth=http" "profile=internal"
"user=admin" "password=alsogood" "target_virtualmachine=vmB3"
Plugin Restore Options
server:              *None*              (*None*)
truststore_file:     *None*              (*None*)
truststore_password: *None*              (*None*)
system_num_tries_connection: *None*              (*None*)
system_interval_request: *None*              (*None*)
system_timeout_request: *None*              (*None*)
auth:                *None*              (*None*)
profile:             *None*              (*None*)
user:                *None*              (*None*)
password:            *None*              (*None*)
vm_disks:            *None*              (*None*)
vm_exclude_disks:    *None*              (*None*)
cluster:             *None*              (*None*)
storage_domain:      *None*              (*None*)
vm_name:             *None*              (*None*)
```

(continues on next page)

(continued from previous page)

```
vm_force_overwrite:  *None*              (*None*)
vm_switchon:         *None*              (*None*)
disk_interface:      *None*              (*None*)
disk_active:         *None*              (*None*)
disk_boot:           *None*              (*None*)
nics:                *None*              (*None*)
disk_names:          *None*              (*None*)
template_name:       *None*              (*None*)
remove_base_vm:      *None*              (*None*)
log:                 *None*              (*None*)
debug:               *None*              (*None*)
debug:Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
1: server (Restore server)
2: truststore_file (Restore truststore file)
3: truststore_password (Restore truststore password)
4: system_num_tries_connection (Number failed tries before abort)
5: system_interval_request (Time (s) between failed tries)
6: system_timeout_request (Timeout(s) to request server)
7: auth (Mode authentication(http|auth2))
8: profile (Profile connection user)
9: user (Restore user name)
10: password (Password user)
11: vm_disks (List id disks to restore)
12: vm_exclude_disks (Exclude id disks in restore)
13: cluster (Target restore cluster)
14: storage_domain (Target restore domain)
15: vm_name (New name Vm restore)
16: vm_force_overwrite (Force overwrite if exist vm)
17: vm_switchon (Turn on vm when retore finished)
18: disk_interface (List interface disk)
19: disk_active (List if want active disks)
20: disk_boot (List if want bootable disks)
21: nics (List restore MACS)
22: disk_names (LIst disks' names)
23: template_name (New name Template restore)
24: remove_base_vm (Remove base vm of template)
25: log (Log path in restore phase)
26: debug (Level debug in restore phase)
Select parameter to modify (1-24): 15
Please enter a value for vm_name: TestBaculaActive
Plugin Restore Options
server:              *None*               (*None*)
...
storage_domain:      *None*               (*None*)
vm_name:             TestBaculaActive     (*None*)
vm_force_overwrite:  *None*               (*None*)
vm_switchon:         *None*              (*None*)
disk_interface:      *None*              (*None*)
disk_names:          *None*               (*None*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
```

(continues on next page)

(continued from previous page)

```
1: server (Restore server)
...
16: vm_force_overwrite (Force overwrite if exist vm)
17: vm_switchon (Turn on vm when retore finished)
18: disk_interface (List interface disk)
19: disk_active (List if want active disks)
20: disk_boot (List if want bootable disks)
21: nics (List restore MACS)
22: disk_names (LIst disks' names)
23: template_name (New name Template restore)
24: remove_base_vm (Remove base vm of template)
25: log (Log path in restore phase)
26: debug (Level debug in restore phase)
Select parameter to modify (1-24): 16
Please enter a value for vm_force_overwrite: true
Plugin Restore Options
server:             *None*              (*None*)
...
storage_domain:     *None*              (*None*)
vm_name:            TestBaculaActive    (*None*)
vm_force_overwrite: true                (*None*)
vm_switchon:        *None*              (*None*)
disk_interface:     *None*              (*None*)
disk_active:        *None*              (*None*)
disk_boot:          *None*              (*None*)
nics:               *None*              (*None*)
disk_names:         *None*              (*None*)
template_name:      *None*              (*None*)
remove_base_vm:     *None*              (*None*)
log:                *None*              (*None*)
debug:              *None*              (*None*)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
...
OK to run? (yes/mod/no): yes
Job queued. JobId=935
```

File-Level Restore

It is possible to perform file listing and extraction functions over the virtual disk images that are restored on the Bacula client's file system (as opposite to restore directly to the RHV manager) by utilizing the set of tools developed and maintained as a part of a libguestfs project (http://libguestfs.org).

Prerequisites are to have libguestfs tools installed on the host that the virtual disk image file is going to be restored on and to have the appropriate daemon started.

Example of commands that should be run in order to meet these prerequisites are, in the case of RHEL7 host:

```
yum install libguestfs-tools
```

```
systemctl start libvirtd
```

In order to be able to list the contents of the virtual disk image, the disk image itself needs to be restored on the filesystem of the local host and command `virt-ls` executed.

Example of the command usage (where the virtual disk has been restored to the file `/tmp/bacula-restore/vm1-cos7/Disk1-249226e-c38c-44f0-a121.img` in order to list the contents of the / directory):

```
# virt-ls -R -a /tmp/bacula-restore/vm1-cos7/Disk1-249226e-c38c-44f0-a121.img␣
↪/
```

In order to be able to extract the contents of the virtual disk image, the disk image itself needs to be restored on the filesystem of the local host and command `virt-copy-out` executed.

Example of the command usage (where the virtual disk has been restored to the file `/tmp/bacula-restore/vm1-cos7/Disk1-249226e-c38c-44f0-a121.img` in order to extract the contents of the `/home/` directory to the `/tmp` directory):

```
# virt-copy-out -a /tmp/bacula-restore/vm1-cos7/Disk1-249226e-c38c-44f0-a121.
↪img \
 /home/ /tmp
```

Extra notes

- In RHV, inactive disks are ignored when a clone is made from a snapshot.
- In a snapshot backup, if there is a inactive disks without any snapshots, the content of this won't be backed. But the disk will be created, without content, in the restore.
- The templates always use the disk format "thick-provisioned", although the format disks of base vm was "thin-provisioned". Can not modify templates' disks.
- The disks out of the template behave like disks of independent vms.
- The backup of virtual machines derived from independent templates behaves like independent virtual machines.
- The backup of vms derivated of dependent templates, only the snapshots of the machine will be backed up, without those of the template.
- The networks of the templates can be modified(name, connected, etc..) after the template was created. So if the network name is changed in restores from previous backups to this change as result will be more networks, as many networks as have been modified.

Diagnostic Operations

For diagnostic purposes, it is possible to interact with the RHV plugin java program on the shell to list information it retrieves from the RHV system as well as to actually transfer data.

Querying RHV

Assuming the Java environment is properly configured, the general approach is as follows:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --server=<RHVM-DNS> \
--truststore_file=<path-to-truststore> --truststore_password=<password> \
--auth=<auth> --profile=<profile> \
--user=<user-name> --password=<password> \
--operation=list --path=<RHVM structure>
```

Results will be returned in JSON format.

The required options indicated above are directly representing the plugin options as described in table *Plugin Parameters*.

The following list shows all the available RHVM structure, by default “/vms”:

```
/
  api/
    version/
    compatibility/cluster_storage/"name_cluster"/"name_storage_domain"
  clusters/"name"/
    hosts -> ../hosts
    vms -> ../vms
    templates -> ../templates
  datacenters/"name"/
    clusters -> ../clusters
    disks -> ../disks
    hosts -> ../hosts
    storages -> ../storages
    vms -> ../vms
    templates -> ../templates/
  disks/"name"/
  hosts/"name"/
    storages -> ../storages/
    vms -> ../vms
  tags/"name"/
    vms -> ../vms
    vms/"name"/
      disks -> ../disks
      snapshots
    storages/"name"/
      disks -> ../disks
      vms -> ../vms
      templates -> ../templates
    templates/"name"/
      disks -> ../disks
      vms -> ../vms
```

It is also possible to list RHV information using using the `.ls`-command of `bconsole`. Below, some examples are provided.

The meaning of the required and optional parameters are the same as shown in table *Plugin Parameters* and above for the query functionality.

Note also that `bconsole` commands need to be entered without any line breaks.

Verifying compatible backups of VMs than they starts with vm1*:

```
*.ls client=RHV-fd plugin="rhv:server=myrhv.com \
insecure=true password=rhvPass123" path=/vms/vm1*

#Answer:
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 4f5f9abd-f407-4da6-ba39-
↪272aed709c80->vm1\
                        [vm] [BACKUP_CLONE,BACKUP_SNAP]
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 a8bf6b18-a116-4067-b884-
↪172777fec446->vm11\
                        [vm] [BACKUP_SNAP]
```

Listing all disks attached to "vm1":

```
.ls client=RHV-fd plugin="rhv:server=myrhv.com \
insecure=true user=bacula password=alsogood" path=/vms/vm1/disks/

# Answer:
#-rw-r----- 0 root root 0 1970-01-01 01:00:00 c43a1865-13ba-483e-b38b-
↪6b929d6b4653->\
                        Preassigned_Disk1 [disk]
#-rw-r----- 0 root root 1 1970-01-01 01:00:00 b0b349ea-72d3-405f-9429-
↪c1cb009f30ce->\
                        Preassigned_Disk2 [disk]
#-rw-r----- 0 root root 2 1970-01-01 01:00:00 3f993fb1-3157-4916-aad1-
↪2f323a3cdd4f->\
                        Preassigned_Disk3 [disk]
#-rw-r----- 0 root root 3 1970-01-01 01:00:00 e9ac9bd8-920f-4eb8-b4d0-
↪b405d675b16f->\
                        Preassigned_Disk4 [disk]
```

To list all datacenters:

```
.ls client=RHV-fd plugin="rhv:server=RHV.example.com \
insecure=true user=bacula password=alsogood" path=/datacenters/

# Answers:
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 3e54cf31-7a9a-46ad-8a8d-
↪7843b6d7a145->\
                        dcTest [datacenter]
# -rw-r----- 0 root root 1 1970-01-01 01:00:00 4cdcc978-73af-11e8-835d-
↪b827eb7e5e47->\
                        Default [datacenter]
```

To list all options in datacenter named "dcTest":

```
.ls client=RHV-fd plugin="rhv:server=RHV.example.com \
insecure=true user=bacula password=alsogood" path=/datacenters/dcTest/

```

(continues on next page)

(continued from previous page)

```
# Answer:
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 clusters
# -rw-r----- 0 root root 1 1970-01-01 01:00:00 disks
# -rw-r----- 0 root root 2 1970-01-01 01:00:00 hosts
# -rw-r----- 0 root root 3 1970-01-01 01:00:00 storages
# -rw-r----- 0 root root 4 1970-01-01 01:00:00 templates
# -rw-r----- 0 root root 5 1970-01-01 01:00:00 vms
```

Check if the Cluster “clusterTest” is compatible with the Storage Domain “iSCSIStorage”

```
.ls client=RHV-fd plugin="rhv:server=RHV.example.com \
user=bacula password=alsogood" insecure=true \
path=/api/compatibility/clone_storage/clusterTest[Name_cluster]/
↪iSCSIStorage[Name_storage]
# Answer:
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 true
```

Get version of RHVM:

```
.ls client=RHV-fd plugin="rhv:server=RHV.example.com \
user=bacula password=alsogood insecure=true" path=/api/version
# Answer:
# -rw-r----- 0 root root 0 1970-01-01 01:00:00 4.2.0
```

System operations

Plugin System

These operations are designed to offer some functionalities without affecting the RHVM’s state. Among these are check connection with RHVM, create truststore automatically and encode password using same pattern as Bacula. The possible parameters are defined in table *Plugin Parameters*

Warning: Call plugin directly!

The plugin generates a new truststore:

```
java -jar rhv-backup.jar --truststore_file=/opt/bacula/etc/rhvOwn.truststore \
--truststore_password=sosecret --log=./log_debugff.log --server=myrhvm.com \
--hpassword=Mjk1M2QwRnPCncKT --operation=system --create_truststore=true
# Result: Truststore file in /opt/bacula/etc/rhvOwn.truststore with password␣
↪sosecret
```

Warning: Advanced use only

These allow control the disk request timeout, the number tries before reporting an error and more.

```
java -jar rhv-backup.jar --log=./log_debugff.log --server=myrhv.com \
--hpassword=Mjk1M2QwRnPCncKT --operation=list --path=/datacenters/ \
--num_tries_connection=10 --system_timeout_request=120
```

```
Fileset {
  Name = ExampleFileset3
  Include {
    Options {
      Signature = MD5
    }
    Plugin = "rhv:server=rhv.example.com insecure=true
    system_num_tries_connection=100 system_interval_request=1200 system_
↪timeout_request=2000
    user=admin password=rhvpass123 target_template=template*"
  }
}
```

Transferring Data

The Java program interacting with RHV can also be used to transfer data between the RHV infrastructure, in which case additional parameters are available and the `operation=list` parameter-value-pair is not used.

In this mode, the following general considerations apply:

- Either backup, restore, or list mode must be chosen by using appropriate options. Modes can not be mixed.
- Options have to start with double dashes, as in `--server`.
- The `--output` defines the output format and should be explicitly set to either `console` or `json`.
- These modes are not fully supported, but exist mostly to assist in trouble shooting with **Bacula Systems** support team. As such, they are not fully documented and should not be used for regular operations or maintenance of a RHV environment.

Some examples are provided below.

Backup of datacenter “OneDatacenter” configuration:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --output=console --debug=1 \
--server=rhv.example.com \
--truststore_file=/opt/bacula/etc/rhvm.truststore \
--truststore_password=sosecure --auth=http --profile=internal \
--user=bacula --password=alsogood \
--operation=backup \
--target_datacenter=OneDatacenter \
--target_configuration_only=true \
--path=/tmp/rhvPlugins/
```

Backup of all Virtual Machines whose names match the regular expression “vm.*”, but excluding “vm1”. Note the quoting of the asterisk character to avoid the shell expaning it:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --output=console --debug=1 \
--server=rhv.example.com \
--truststore_file=/opt/bacula/etc/rhvm.truststore \
--truststore_password=sosecure --auth=http --profile=internal \
--user=bacula --password=alsogood \
--operation=backup \
--target_path=/vms/vm.* \
--target_exclude_vms=vm1 \
--path=/tmp/rhvPlugins/
```

Restoring the Virtual Machine “vm1”:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --output=console --debug=1 \
--server=rhv.example.com \
--truststore_file=/opt/bacula/etc/rhvm.truststore \
--truststore_password=sosecure --auth=http --profile=internal \
--user=bacula --password=alsogood \
--operation=restore \
--vm=vm1 \
--path=/tmp/rhvPlugins/
```

Restoring a VM, renaming disks:

```
java -jar /opt/bacula/lib/bacula-rhv-plugin.jar --output=console --debug=1 \
--server=rhv.example.com \
--truststore_file=/opt/bacula/etc/rhvm.truststore \
--truststore_password=sosecure --auth=http --profile=internal \
--user=bacula --password=alsogood \
--operation=restore --restore_vm=vm3 \
--vm_disk_names=65ccb526-30c0-4beb-b348-4395e70d6e32:vm3_restore_disks1,vm3_
↪restore_disks_general
--path=/tmp/rhvPlugins/
```

Troubleshooting

Bacula environment

This plugin stores the logs depending of the number of paralel executions it has. In other words, if the operations are only performed synchronously, it will always be sotred in the same file, under the suffix of ’-0.log’. However, when a backup is running and another backup(or a restore) is started, the first backup will continue to store the log in the same file(under suffix of ’-0.log’), but the one just started will be stored in the file under the suffix of ’-1.log’.

The plugin indicates the log file in a message in job log. By default, the location of log file will be: */opt/bacula/working/rhv/rhv-debug-X.log*, but the user can modify this default value in file */opt/bacula/rhv_backend*, or in each job with param *log*.

It’s recommended that the log file always be in */opt/bacula/working/rhv/*.

RHV environment

The RHV Manager can return errors in some situations. The RHV REST API return code and messages do not always precisely reflect the cause of the problem. Given that general HTTP codes indicating rather broad error conditions are used, the plugin can not always react in a perfect manner. Instead, the plugin returns all available information about errors and warnings, but in many situations it will be necessary to check RHV logs to understand the cause of an operational error.

The most informative logs of RHV are usually `vdc-log.txt` and `/var/log/ovirt-engine/engine.log` (for RHV Manager) and `/var/log/vdsm/vdsm.log` (for virtualization hosts) since those reflect the main flow of the application and if any error occurs, it will generally be tracked in those logs.

Once the exact component that failed is known, it is possible to check the specific log file. The full list of RHV logs is described at redhat.com/…/17587#RHEV_Logs_Overview

Below we present some common errors and associated actions to resolve them.

Failed Transfer

If a disconnection happens when a disk image is being transferred by the ImageTransferService, the virtual disk remains in the state “Transfering via API” even if the backup job has terminated. This state blocks snapshot deletion. If a backup has suffered from non recoverable connection loss to the RHV Manager and this state remains for a virtual disk, a manual action on the RHV Manager can be performed to unblock the situation:

```
# log in to the RHV Manager shell, as root user
su postgres
psql -U engine -d engine
SELECT command_id FROM image_transfers;
# Get the command_id that is aborted
DELETE FROM image_transfers WHERE command_id='<UUID from above>';
# The state 'Transfering via API' will change to 'Locked' (2).
# Query the images table for locked images
SELECT image_group_id FROM images WHERE imagestatus = '2';
# Identify which one needs to be unlocked
UPDATE images SET imagestatus = '1' WHERE imagestatus = '2';
```

This will make the virtual disk inoperative, but it will allow either delete or commits the snapshots created previously.

In the RHV GUI, navigate to “Virtual Machines”, select the faulty one, select “Snapshots”, and perform a preview of a stable snapshot. Afterwards, “Commit” the action.

The plugin will automatically clear this situation with the following backup. This manual intervention is only necessary if there is no option of running a subsequent backup.

DSM <host> command ExtendImageTicketVDS Failed

The full message provided would be similar to

```
DSM <host> command ExtendImageTicketVDS failed: Image daemon request
failed: u'
status=404, code=404,
title=Not Found,
explanation=The resource could not be found.,
detail=No such ticket: a035901e-c3d6-4033-b0fa-8e127d422ccf,
reason=Not Found'
```

This error can be solved manually in the same way as the previous problem, and the plugin will also clear this situation with the following backup.

javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException

Here, the full error message would look like

```
javax.net.ssl.SSLHandshakeException:sun.security.validator.ValidatorException:
  PKIX path building failed:
    sun.security.provider.certpath.SunCertPathBuilderException:
      unable to find valid certification path to requested target
at
sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
at
sun.security.ssl.SSLSocketImpl.fatal(SSLSocketImpl.java:1959)
at
sun.security.ssl.Handshaker.fatalSE(Handshaker.java:302)
at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:296)
...
```

This error is related to the certificates used by RHV. A possible solution is shown here: stackoverflow.com/…/pkix-path-building-failed.

org.apache.http.NoHttpResponseException: <RHVM>:443 failed to respond

The full error is

```
org.apache.http.NoHttpResponseException: rhv.example.com:443 failed to respond
org.ovirt.engine.sdk4.Error
org.ovirt.engine.sdk4.Error: Failed to send request at
org.ovirt.engine.sdk4.internal.HttpConnection.send(HttpConnection.java:213)
at org.ovirt.engine.sdk4.internal.services.
  EventsServiceImpl\$AddRequestImpl.send(EventsServiceImpl.java:83)
...
```

This error happens when no certificate is present. The solution can be found in the SSLHandshakeException problem resolution.

HTTP Response Code is 409

The full error is:

```
org.ovirt.engine.sdk4.Error: HTTP response code is "409".
  HTTP response message is "Conflict".
at org.ovirt.engine.sdk4.internal.services.ServiceImpl.
  throwError(ServiceImpl.java:113)
at org.ovirt.engine.sdk4.internal.services.ServiceImpl.
  checkFault(ServiceImpl.java:40)
```

This can happen if a snapshot could not correctly get deleted.

As explained in section **ch:fixlockedsnapshot**, this can be manually resolved by commiting a snapshot preview in the RHV GUI.

The plugin will solve this situation manually with the following backup. A manual intervention is only necessary if there is no option of running that following backup.

HTTP response is 401. HTTP response message is "Unauthorized"

This error happens when RHVM credentials are incorrect.

HTTP 503 Code at Disk Download

This happens if the connection between RHV Manager and RHV virtualization host daemons is not working correctly. Usually this indicates a firewalling issue.

The solution is to ensure that the firewall service (`firewalld` is active and correctly configured. For RHV operations, two ports–54321 for `vdsm` and 54322 for `ovirt-imageio`– need to be accessible. To ensure this, execute the follow commands on RHV virtualization hosts:

```
systemctl start firewalld
firewall-cmd --permanent --add-port=54321/tcp
firewall-cmd --permanent --add-port=54322/tcp
```

To check if this command has been effective:

```
ssystemctl restart firewalld
firewall-cmd --list-ports
```

The ports 54321/tcp and 54322/tcp must appear in the list of open ports.

In a production environment, it is reasonable to limit access to the daemons affected only from trusted hosts or network segments. Please refer to RHEL documentation of the `firewall-cmd` program, which is available using `man firewall-cmd`.

VDSM hostTest command GetCapabilitiesVDS failed: Vds timeout occured

This error can happen if a host is not registered. It shouldn't impact operations on other hosts.

To fix this situation it is usually necessary to confirm the host has been rebooted or shutdown and put it into maintenance mode after.

Can't create any snapshot in a specific vm, but the vm can run

The origin of this error is unknown, this case is reported here. The simple explanation is that an attempted snapshot removal (which we have no logs of), corrupted the chain.

See also:

Best Practices for Hypervisors

Future Development Directions

Some features are planned for further development of this plugin:

- Backup and restore of RHV Manager Configuration and database.
- Implement an option to control Storage Domain overcommitment.

RHV Single Item Restore

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This article presents how to use the RHV Single File Restore feature with **Bacula Enterprise** and the RHV Plugin.

Features Summary

The **Bacula Enterprise** RHV Single File Restore provides the following main features:

- Console interface and BWeb integration
- Support only for Full level jobs
- Support for Linux filesystems (ext3, ext4, btrfs, lvm, xfs)
- Support for Windows filesystems (FAT, NTFS)

RHV SIR is available starting with Bacula Enterprise 12.2

This document will present solutions for **Bacula Enterprise** 12.2 and later, which are not applicable to prior versions. The RHV Single File Restore has been tested and is supported on RHEL 7.x, RHEL 8.x, Ubuntu Xenial, Debian Stretch and Debian Buster. SELinux is currently not supported.

Warning: Only Full level backups are currently supported with RHV Single Item Restore.

Installation

Packages for the RHV Single File Restore plugin are available for supported platforms. Please contact Bacula Systems to get them.

Download the plugin package to your **Storage Daemon** server and then install using the package manager like so:

```
rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the RHV Single File Restore plugin and will install dependencies. On RHEL, it will be needed to install `perl-JSON` package from **rpmforge** and the `libguestfs-winsupport` package.

Note: On RHEL 7/8.x, it is necessary to install a custom version of the libguestfs packages from our repository to support NTFS devices. Those should not be updated with a newer version from official repositories. The YUM package manager has plugins to prevent package updates, try **yum-plugin-versionlock** or **yum-plugin-priorities**.

Additionally, the `ntfs-3g` package from the EPEL repository is needed for NTFS support. To install the EPEL respository, please follow the official instructions on the EPEL website to install the “epel-release” package here:

https://fedoraproject.org/wiki/EPEL

Note: On RHEL 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag]
name = Bacula - RPMFORGE - DAG
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64
enabled = 1
protect = 0
gpgcheck = 0

# cat /etc/yum.repos.d/baculasystems.repo
[baculasystems-single_item_restore]
name = Bacula Enterprise - SIR
baseurl = https://www.baculasystems.com/dl/<xxx>/rpms/single-item-restore/
↪<version>/rhel7-64/
enabled = 1
protect = 0
gpgcheck = 0


Note: This last repository is required on RHEL7:

[baculasystems-dag-guestfish]
name = Bacula Enterprise - DAG for Guestfish
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64/guestfish/
enabled = 1
protect = 0
gpgcheck = 0
```

```
# yum install bacula-enterprise-single-item-restore
```

If BWeb Management Suite is used:

```
# service bweb restart
```

Notes about the "bacula" Account on RHEL

All commands in this document use the "bacula" unix account to run.

On RHEL, the Unix "bacula" account is locked by default. It means that it's not possible by default to execute a command such as "su - bacula".

It is possible to unlock the "bacula" account, or to use "sudo -u bacula" to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

Can be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using

```
root@storage# sudo -u bacula /bin/bash
```

Or unlock the "bacula" unix account and use "su -" with a command such as:

```
root@storage# chsh -s /bin/bash bacula
root@storage# su - bacula
bacula@storage# whoami
bacula
```

Fuse FileSystem

If a restore session is not properly cleaned up, some directories might still be mounted with the Bacula Fuse FileSystem.

```
baculafs on /opt/bacula/working/cat-ro type fuse.baculafs (ro,user=bacula)
backend0 on /opt/bacula/working/test-vm-0 type fuse.backend0 (ro,user=bacula)
/dev/fuse on /opt/bacula/working/test-vm type fuse (rw,nosuid,nodev,
↪user=bacula)
```

It is possible to unmount directories with the `fusermount -u` command.

```
bacula@storage# fusermount -z -u /opt/bacula/working/26
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm-0
bacula@storage# fusermount -z -u /opt/bacula/working/test-vm
```

Samba SMB Shares

The **Bacula Enterprise** RHV Single File Restore plugin can automatically set up Samba SMB shares from the console program or the BWeb Management Suite.

To enable Samba SMB network shares, installing and configuring the "samba" package is mandatory. To configure the `/etc/samba/smb.conf` file correctly, you need to run `install-single-item-restore.sh` script.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
include = /etc/samba/conf.d/all
```

At this point, it is possible to modify `/etc/samba/smb.conf` to add your own configuration directives.

Network share descriptions will be stored in the directory `/etc/samba/conf.d`. It is possible to create and customize the template used by Bacula to generate configuration files.

```
root@storage# cat /etc/samba/conf.d/custom.tpl
[__share__]
   path = __path__
   follow symlinks = yes
   wide links = yes
   writable = yes
```

Configuration

On the **Storage Daemon** host server, the `bconsole` program should be configured properly to let the "bacula" user connect to the Director with `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 12.8.0
Enter a period to cancel a command.
* version
mydir-dir Version: 12.8.0 x86_64-redhat-linux-gnu
* quit
```

The package contains a script to test the connection with the Director and to test if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage#  /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you may not see any files or files may have strange permissions, ownerships and sizes and will inaccessible even to the root user.

Restore Scenario With Text Console Interface

The RHV Single File Restore plugin provides a simple console program that provides access to files inside VMs.

```
bacula@storage# /opt/bacula/bin/mount-vm
Automatically Selected Catalog: MyCatalog

Client list:
1: 127.0.0.1-fd
2: win2008-fd
3: rhel7-fd
Select a Client: 1
Selected Client: 127.0.0.1-fd

Job list:
1: RHVSERVER.2021-02-15_19.12.51_34
2: RHVSERVER.2021-02-16_12.12.29_39
3: RHVSERVER.2021-02-16_12.37.54_03
4: RHVSERVER.2021-02-16_14.23.47_03
5: RHVSERVER.2021-02-16_15.45.32_03
6: RHVSERVER.2021-02-16_17.00.47_52
Select a Job: 6
Selected RHVSERVER.2021-02-16_17.00.47_52

Virtual Machine:
1: squeeze2
2: win2008
3: rhel7
4: sir-test-vm
Select a Virtual Machine: 4
Selected sir-test-vm

Actions list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Cleanup
Select a Actions: 1
Selected Mount guest filesystem locally

I: Files are available under /opt/bacula/working/mount-vm-6434/disks/sir-test-
↪vm
I: Press enter to finish and cleanup the session
```

In this step, the virtual machine filesystem is mounted locally (in the example above, files are available under `/opt/bacula/working/mount-vm-6434/disks/sir-test-vm`. It is possible to browse directories and copy files (with `cp`, `scp`, `ftp`) as with a standard filesystem from another terminal session with the Unix "root" and "bacula" accounts. If you need to use another Unix account to operate on files, use the "-o allow_other" option when starting the `mount-vm` script.

```
bacula@storage# ls /opt/bacula/working/mount-vm-6434/disks/sir-test-vm
bin   dev  home        lib         media  opt   root  selinux  sys  usr ␣
↪vmlinuz
```

(continues on next page)

(continued from previous page)

```
boot  etc  initrd.img  lost+found  mnt    proc  sbin  srv      tmp  var
```

To clean up the session, just press "Enter" in the terminal session where the `mount-vm` script was started.

It is possible to limit the Job list with the following command line options:

- `-s=<days>` Limit the job list to the last *days*
- `-l=<number>` Limit the job list to the last *number* entries
- `-f=<filter>` Specify an advanced filter based on the Job name, the Fileset name or the JobId

```
# Limit the job output to the last 100 jobs
bacula@storage# /opt/bacula/bin/mount-vm -l 100

# Limit the job output to the last 30 days
bacula@storage# /opt/bacula/bin/mount-vm -s 30

# Limit the job output to jobs that start with ``MyRHVServer''
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyRHVServer*'

# BAD USAGE for the filter option, it will search for a job named␣
↪``MyRHVServer''
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyRHVServer'

# Limit the job output to jobs that start with ``MyRHVServer''
# and that use the Fileset Test1
bacula@storage# /opt/bacula/bin/mount-vm -f 'jobname=MyRHVServer*␣
↪fileset=Test1'

# Limit the job to the jobid XX
bacula@storage# /opt/bacula/bin/mount-vm -f jobid=XX
```

In some cases, the device detection doesn't work properly. It is possible to use the `-m` option to mount recognized disks in a simple way. The option is automatically set when only one disk is selected during the restore.

```
bacula@storage# /opt/bacula/bin/mount-vm -m
```

Notes

Cache Directory

To speed up future RHV Single File restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
bacula@storage# ls /opt/bacula/working/mount-cache
sir-test-vm-0.bmp  sir-test-vm-2.bmp    MyCatalog-2.idx  MyCatalog-5.idx ␣
↪MyCatalog-8.idx
sir-test-vm-1.bmp  sir-test-vm.profile  MyCatalog-4.idx  MyCatalog-6.idx ␣
↪MyCatalog-9.idx
```

It is possible to remove files in the cache after some time; they will be re-generated if needed.

Support

The `install-single-item-restore.sh` script can collect traces automatically when a `mount-vm` session is running.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh support
```

Limitations

- The RHV Single File Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backend due to internal MySQL limitations with indexes on TEXT colums. For RHV Single Item Restore there should not be too much impact on performances (the backup structure is usually quite small) but we advise using the PostgreSQL backend for the best experience.
- The RHV Single File Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was created with a large number of concurrent jobs.
- The Storage Daemon where the RHV Single File Restore is installed should have a CPU with the VT-x/EPT extensions. If these extensions are not available, the performance will be degraded. (From 20s to 10mins in our lab).
- RHEL 7/8 does not support mounting NTFS disks with the libguestfs provided with their system. To mount Microsoft NTFS disks on RHEL 7/8, it is required to install a patched version of the libguestfs packages. Please see notes in the "Installation" section of this document for more information.
- The RHV Single File Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.

vSphere Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This document aims at presenting the reader with information about various techniques and strategies to backup VMware virtual machine using **Bacula Enterprise vSphere Plugin**. It describes the Plugin, defines the scope of its operations, and presents its main features. It also covers how to restore specific files (Single File Recovery) or instantly restore VM (VM Instant Recovery) from backups made with Bacula Enterprise vSphere Plugin. Make sure you have familiarized yourself with the VM discovery automation tool that simplifies Backup Administrator tasks in adding new VMs or managing decommissioned VMs automatically.

Scope

This Plugin is available since **Bacula Enterprise** 8.0, and is not applicable to prior versions of Bacula.

It supports the following versions of vSphere:

- ESX/ESXi 5.0, 5.1, 5.5, 6.0, 6.5, 6.7, 7.0 and 8.0
- vCenter 6.0, 6.5, 6.7, 7.0 (Bacula Enterprise version 12.4.3 and higher) and 8.0 (Bacula Enterprise version 16.0.0 and higher)
- Virtual machines: VM hardware version 7 and higher.

The vSphere Plugin has been tested with and is supported on the following Linux platforms:

- RHEL 64bit versions 7, 8 and 9
- Oracle Linux version 8
- SLES 64bit version 12.5

Features

The vSphere Plugin provides virtual machine bare metal recovery, while the backup at the guest level simplifies data protection of critical applications.

The Plugin integrates VMware's Changed Block Tracking (CBT) technology to ensure only blocks that have changed since the initial Full, and/or the last Incremental or Differential backup are sent to the current Incremental or Differential backup stream to give you more efficient backups and reduced network load.

See the features list:

- vSphere Storage APIs – Data Protection based online backups
- VSS-based guest snapshots for quiescing VSS-based applications
- Full, Differential and Incremental image-level backups of virtual machines
- Restores complete virtual machine images
- Restores vmdk files to alternate directory
- Supports both TCP/IP and SAN (FC/iSCSI) VMware datastore access
- NBD (Network Block Device), HotAdd, or SAN access
- The vSphere Plugin is compatible with Copy/Migration jobs. In order for Incremental vSphere Plugin backup jobs to be compatible with Copy Jobs, the devices in the destination Storage Daemon **must** have the setting "Maximum Concurrent Jobs = 1". If this option is not set, then restores from Incremental copied vSphere backup jobs may not be possible. Read the CopyMigrationJobsReplication documentation for more information.

Along with the vSphere Plugin, Bacula Systems provides additional packages which allows:

- Single Item Restore
- Instant Recovery

from vSphere Plugin backups of VMware VMs. It also integrates with the autodiscovery of new VM with the *Scan Plugin*

Backup Strategies

This article aims at presenting possible VM backup strategies with Bacula Enterprise.

Image Backup with the vSphere Plugin

With the image backup level strategy, the **Bacula Enterprise** vSphere Plugin will save the virtual machine's disks at the *raw* level, in the VMware/vSphere context. For this to work, you don't need a Bacula File Daemon on each guest VM. You only need one FD with the vSphere Plugin installed. It is recommended that this FD be installed on the same machine as a Storage Daemon (SD) so that data from the ESXi servers traverses a network link only once. The vSphere Plugin will contact your VMware ESXi server to read and save the contents of your virtual machine disks using Network Block Device (NBD), HotAdd, or SAN access. When directly accessing a **vmdk** image stored on your **Datastore**, Bacula doesn't need to walk through the Client Filesystem to open/read/close/stat files, so it consumes fewer resources on your ESXi infrastructure than a backup with a File Daemon on each guest machine would. On the other hand, Bacula will also read and save useless data such as swap files or temporary files.

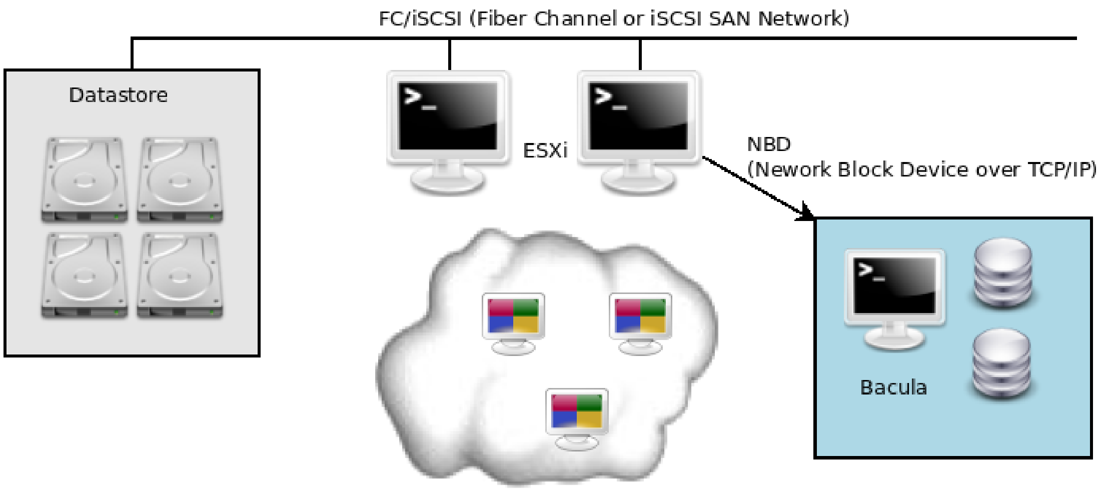

FC/iSCSI (Fiber Channel or iSCSI SAN Network)
Datastore
ESXi
NBD
(Nework Block Device over TCP/IP)
Bacula

Fig. 21: Backup through TCP Network using NBD (Network Block Device)

When the vSphere Plugin is using the NBD transport method for the backup, the data is streamed to the backup server via the ESXi system's VMkernel port.

The Bacula Enterprise vSphere Plugin can also use your SAN infrastructure to minimize the I/O load on your ESXi servers. Using this method of access, even fewer resources are consumed on the ESXi server, but the data still needs to be read from your datastore, thus it is still possible to experience I/O contention.

When using block differential techniques such as those used by the vSphere Plugin, you need to **ensure** that all Incremental backups are available for restore. If one of your Incremental Jobs is missing at the restore time, Bacula will not be able to create a consistent image. Using the Differential level reduces the number of Jobs that are required for restore, and thus reduces the risk that something might be lost. To avoid losing important Incremental Jobs, you must ensure that your Volume retention periods are long enough to recover all of your data.

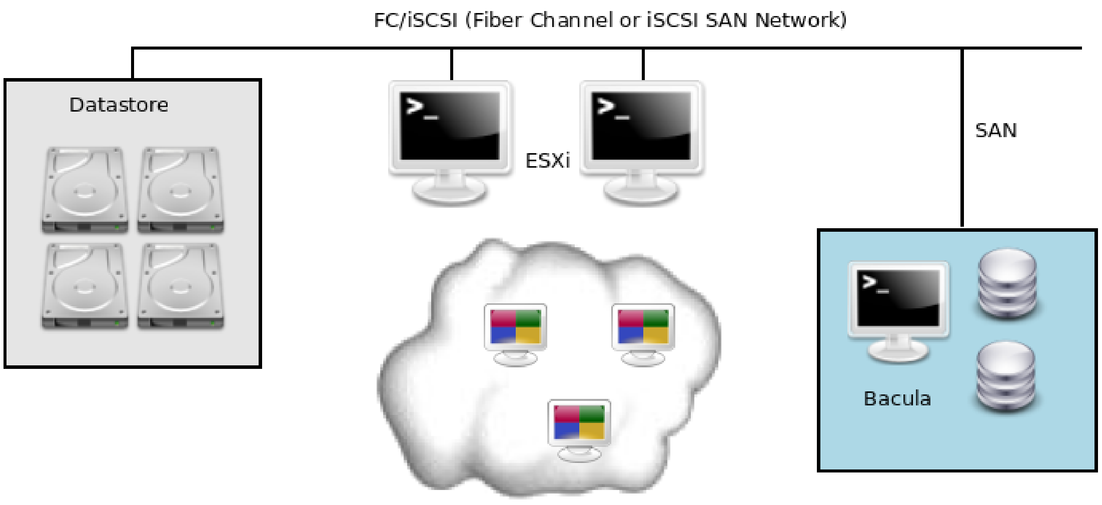


Fig. 22: Backup through SAN Network

Installing Bacula Client on Each Guest

This strategy doesn't use the Bacula Enterprise vSphere Plugin, but instead installs a Bacula Enterprise File Daemon (FD) on every virtual machine as if they were physical servers. To optimize the I/O usage on your VMware ESX/ESXi server, you will use **Schedule**, **Priority** and **Maximum Concurrent Jobs** to spread your backup jobs over your backup window. Since all servers use the same set of disks, running all your backup jobs at the same time could create a bottleneck on the disk/network subsystem.

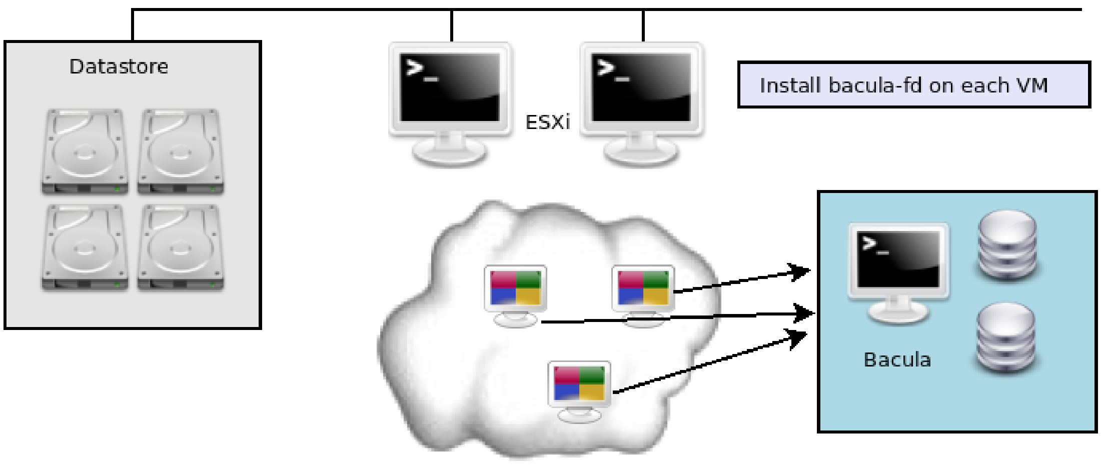


Fig. 23: Installing bacula-fd on each guest

Installing a Bacula Enterprise File Daemon on each virtual machine allows you to manage your virtual servers like physical servers, and to use all of Bacula Enterprise's features such as:

- quick restores of individual files
- checksum of individual files for virus and spyware detection

- Verify Job
- file/directory exclusion (such as swap or temporary files)
- file level compression
- and others.

Strategies Comparison

Table 22: Backup Strategies Comparison

Features	Inside Guest	vSphere VADP
Incremental backup	Yes	Yes
Fileset options	Yes	No
Block level backup	No	Yes
Accurate support	Yes	Yes
Speed	Slow	Fast
I/O Load	High	Low
LAN free backup	No	Yes

Installation

This article describes how to install Bacula Enterprise vSphere Plugin.

Prerequisites

Note: The `Plugin Directory` is set by default, however, it is recommended to double-check if the `Plugin Directory` directive of the `File Daemon` resource in */opt/bacula/etc/bacula-fd.conf* points to the directory where the `vsphere-fd.so` plugin is installed. The standard Bacula plugin directory is: `/opt/bacula/plugins`.

Your File Daemon must have access to the vCenter/vSphere management network, port TCP/443 for API calls, and port TCP/902 for NBD data transfer or direct SAN access to vSphere datastores for SAN transport mode.

vSphere Installation with BIM

In order to install the vSphere Plugin with BIM, install the File Daemon with BIM and choose to install the vSphere Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

Installation with Package Manager

Due to VMware vSphere library dependencies, vSphere Plugin packages are available for a limited number of platforms (check the list here). Make a request from your Customer Portal to access the vSphere Plugin.

Installation can be performed using apt-get or yum/dnf with the package names below if you have added the vSphere Plugin to your Bacula repositories. Otherwise, download the packages and install them manually as below:

```
rpm -ivh bacula-enterprise-vsphere-vixdisk*.rpm
```

```
rpm -ivh bacula-enterprise-vsphere-16.*.rpm
```

These packages will ensure that your **Bacula Enterprise** version is compatible with the vSphere plugin and will install the `vsphere-ctl`, `vddk` and `vsphere-fd` programs.

```
# cd /opt/bacula
/opt/bacula # ls plugins/vsphere-fd.so bin/vsphere-ctl* bin/vddk
-rwxr-xr-x 1 root root  551890 10 nov.  15:13 plugins/vsphere-fd.so
-rwxr-xr-x 1 root root 3551890 10 nov.  15:13 bin/vsphere-ctl.jar
-rwxr-xr-x 1 root root    4096 10 nov.  15:13 bin/vsphere-ctl
-rwxr-xr-x 1 root root 3551890 10 nov.  15:13 bin/vddk
```

Configuration

The following chapter presents the information on how to configure the vSphere Plugin and how to configure a vSphere backup job.

Enable VMware Integration Feature

To enable VMware Integration support, use `yum` with the package names listed below, provided that you have incorporated the vSphere Plugin into your Bacula repositories, as shown:

```
# yum install bacula-enterprise-vsphere-vixdisk
# yum install bacula-enterprise-vsphere
```

These packages will guarantee that your Bacula Enterprise Edition version is compatible with the vSphere Plugin and will install the `vsphere-ctl`, `vddk`, and `vsphere-fd` programs.

```
# cd /opt/bacula
/opt/bacula # ls plugins/vsphere-fd.so bin/vsphere-ctl* bin/vddk
-rwxr-xr-x 1 root root 551890 10 nov. 15:13 plugins/vsphere-fd.so
-rwxr-xr-x 1 root root 3551890 10 nov. 15:13 bin/vsphere-ctl.jar
-rwxr-xr-x 1 root root 4096 10 nov. 15:13 bin/vsphere-ctl
-rwxr-xr-x 1 root root 3551890 10 nov. 15:13 bin/vddk
```

To verify that the VMware Integration Feature is available, examine the BWeb "Configuration -> Health Overview" and scroll down the page to the section titled "Checking Available Features". The "VMware Integration Support" should be listed as available.

vSphere Plugin Configuration

The following chapter presents the information on how to configure the vSphere Plugin.

Prepare Your ESXI or vCenter Environment

The following article gathers the necessary information on how to prepare your ESXI or vCenter environment.

Enabling Block Level Incremental Backup

Note: Changed Block Tracking (CBT) is not supported when the virtual hardware version is 6 or earlier, or when the virtual disk is attached to a shared virtual SCSI bus.

For CBT to identify altered disk sectors since the last change ID, the following items are required:

- The host must be ESX/ESXi 4.0 or later.
- The virtual machine owning the disks to be tracked must be hardware version 7 or later.
- I/O operations must go through the ESX/ESXi storage stack. So NFS is supported, as is RDM in virtual compatibility mode, but not RDM in physical compatibility mode. VMFS is supported, whether backed by SAN, iSCSI, or local disk.
- CBT must be enabled for the virtual machine (see below).
- Virtual machine storage must not be (persistent or non-persistent) independent disk, meaning unaffected by snapshots.

For CBT to identify disk sectors using Full backup, the following items are required:

- The virtual disk must be located on a VMFS volume, backed by SAN, iSCSI, or local disk.
- The virtual machine must have zero (0) snapshots when CBT is enabled, for a clean start.

When using “Thick Provisioned Eager Zeroed” disks, the VMware CBT will report all blocks as “used” during the Full backup.

For virtual machines that do not support CBT, the vSphere Plugin will always perform a Full backup of the virtual disks.

To check if a virtual disk has CBT enabled, open the vSphere Client, select a **powered-off** virtual machine **without snapshots**.

- Right-click the virtual machine and click Edit Settings.
- Click the Options tab.
- Click General under the Advanced section and then click Configuration Parameters. The Configuration Parameters dialog opens.
- Click Add Row.
- Add the `ctkEnabled` parameter and then set its value to `true`.
- Click Add Row, add `scsi0:0.ctkEnabled`,and set its value to `true`.

Note: `scsi0:0` in `scsi0:0.ctkEnabled` indicates the SCSI device assigned to the hard disk that is added to the virtual machine. Every hard disk added to the virtual machine is given a SCSI device that appears similar to `scsi0:0`, `scsi0:1`, or `scsi1:1`.

Note: VMWare FAQ articles may help:

- https://kb.vmware.com/s/article/1020128
- http://kb.vmware.com/kb/1033816

During the first Full backup, the vSphere Plugin will try to enable CBT automatically. To disable this feature, use the `keep_cbt` option in the Plugin command string.

```
Plugin = "vsphere: keep_cbt host=guest1"
```

No snapshots must exist on a virtual machine at the time of enabling CBT on it.

If you notice large Full backup jobs despite small real disk usage, contact your Bacula Systems support team to assess the situation and possibly shrink the backup size by resetting CBT.

Detecting when CBT Is Not Available

When the CBT option is not available for a disk, the `vsphere-ctl*log` file may contain the following notice:

```
com.cybozu.vmbkp.soap.SnapshotManager, getChangedBlocksOfDisk]
      com.vmware.vim25.FileFault
```

When the vSphere Plugin receives this notice, it will automatically do a Full backup of the disk image.

Enabling SAN Access

To use the SAN transport method, your backup server where the vSphere Plugin is installed should have access to all LUNs which are exported to your ESXi server(s). The `multipathd` tools will avoid problems with multiple device paths to SAN devices.

Once your SAN LUNs are visible to your backup server as `/dev/sda`, `/dev/sdb`, … The vSphere Plugin will open each LUN to get the UUID and compare it with what the ESXi server is providing. For example, when using iSCSI, the `lsscsi` command will display them as:

```
% lsscsi
[5:0:0:0]    disk    IET      VIRTUAL-DISK     0     /dev/sdb
```

You can verify that SAN transport is used during backup by using the **debug** option in the vSphere Plugin command line. If SAN transport mode is used, the **vddk** trace file will have an entry showing "Opened san://…" like in the following example:

```
% grep san: /opt/bacula/working/vsphere/<moref>/<seq>/0.log
DISKLIB-LIB   : Opened "san://3-snapshot-18[datastore2] test/test_2.vmdk...
```

When the SAN mode is not available, the vSphere Plugin will automatically switch to the `NBD` transport mode.

Check User Permissions

Attention: Starting with version 12.8.0 of Bacula Enterprise, it is possible to use bconsole to query vSphere and see if a user has all required permissions to perform backups and restores.

Example

```
[root@localhost bin]# ./bconsole
Connecting to Director localhost:9101
1000 OK: 10002 localhost-dir Version: 12.6.1 (05 March 2021)
Enter a period to cancel a command.
*.query client=localhost-fd plugin="vsphere: server=vcenter_192_168_0_8"␣
↪parameter=permissions
missing=VirtualMachine.Provisioning.DiskRandomAccess
missing=VirtualMachine.Provisioning.DiskRandomRead
missing=VirtualMachine.Provisioning.FileRandomAccess
missing=VirtualMachine.Provisioning.GetVmFiles
missing=VirtualMachine.State.CreateSnapshot
missing=VirtualMachine.State.RemoveSnapshot
missing=VirtualMachine.State.RenameSnapshot
missing=VirtualMachine.State.RevertToSnapshot
missing=VirtualMachine.Interact.PowerOff
missing=VirtualMachine.Interact.PowerOn
...
```

If the list is not empty, the listed permissions must be configured properly.

Note: In addition to the above `bconsole` command, you can also use the `vsphere-ctl` command to check the permissions of the current user on the vCenter system and diagnose issues if any:

```
/opt/bacula/bin/vsphere-ctl query list_missing_permissions
```

The following privileges can be allocated to a role and assigned to a Bacula user to perform vStorage backups and restores. These are the minimum required permissions that have been found to be sufficient in the tests performed by Bacula Systems for a basic vSphere environment.

This list may change in the future. The permissions are best propagated downwards from the root of the vSphere level. Additional privileges might be required if advanced features are in use.

Set the following permissions in your vSphere/vCenter environment:

Privilege Level	Permissions
Datastore	• Allocate space • Browse Datastores • Configure Datastores • Delete • Low level file operations • Remove File • Update Virtual Machine Files • Update Virtual Machine Meta Data
Distributed Virtual Switch	• Host operation
Folder	• Create Folder
Global	• Cancel Task • Disable Methods • Enable Methods • Licenses • Log Event • Manage Custom Attributes • Set Custom Attributes • Settings
Host: Configuration	• Advanced Settings • Storage Partition Configuration
Host: Local Operations	• Create Virtual Machine • Delete Virtual Machine • Reconfigure Virtual Machine
Network	• Assign Network
Resource	• Assign Vapp to resource pool • Assign Virtual Machine to resource pool • Query Vmotion
Tasks	• Create task • Update task
vApp	• Add Virtual Machine • Assign Resource Pool • Assign Virtual Machine • Create • Export • Import • vApp application configuration • vApp instance configuration • vApp resource configuration • View OVF Environment
Virtual Machine: Configuration	• Add Existing Disk • Add New Disk • Add or Remove Device

Query Information about vSphere Environment

Attention: New in version 16.0.12

Using a similar mechanism to what was described in the previous section about user permisisons, it is possible to query VMware to get different kinds of information. Below some examples:

```
// List networks
.query client=my-fd plugin="vsphere:" parameter=network

// List resource pools
.query client=my-fd plugin="vsphere:" parameter=pool

// List datastore
.query client=my-fd plugin="vsphere:" parameter=datastore

// List current configuration
.query client=my-fd plugin="vsphere:" parameter=config_list

// Check some configuration section
.query client=my-fd plugin="vsphere: sectionname=vsphere" parameter=config_
↪check
```

Connect to ESXi or vCenter Server

Automated Configuration

The article describes the recommended automated configuration using `vsphere-ctl config` command. If you have access to BWeb and would like to use it to configure the connection to VMware infrastructure and your backup jobs, skip ahead to vspherebweb.

Attention: New in version 16.0.8

The automated configuration is available with version 16.0.12.

Set up the vSphere configuration in the console:

```
[root@localhost bin]# ./vsphere-ctl config create
Enter ESXi/vCenter url: 192.168.0.15
Enter user: administrator@vsphere.local
Enter password:
Connecting to "https://192.168.0.15/sdk"...
OK: successful connection
OK: user has all necessary permissions to perform backups and restores
Select the ESXi host that contains the VMs you wish to backup:
    1) 192.168.0.8
    2) 192.168.0.26
Select host: 1
Computing thumbprint of host "192.168.0.8"
```

(continues on next page)

(continued from previous page)

```
OK: thumbprint for "192.168.0.8" is␣
↪04:24:24:13:3C:AD:63:84:A1:9F:E5:14:82:7D:5C:31:25:A8:FA:89
OK: added entry [vcenter_192_168_0_8] to ../etc/vsphere_global.conf
```

It's also possible to list configurations that are already present:

```
[root@localhost bin]# ./vsphere-ctl config list
[esxi_192_168_0_26]
     root_directory = /opt/bacula/working/esxi_192_168_0_26
     server = 192.168.0.26
     thumbprint = A0:6D:75:53:9E:30:85:BB:99:63:6E:33:C4:B8:64:E9:06:AD:BF:CF
     url = https://192.168.0.26/sdk
     username = root
[esxi_192_168_0_8]
     root_directory = /opt/bacula/working/esxi_192_168_0_8
     server = 192.168.0.8
     thumbprint = 04:24:24:13:3C:AD:63:84:A1:9F:E5:14:82:7D:5C:31:25:A8:FA:89
     url = https://192.168.0.8/sdk
     username = root
[vcenter_192_168_0_26]
     root_directory = /opt/bacula/working/vcenter_192_168_0_26
     server = 192.168.0.26
     thumbprint = A0:6D:75:53:9E:30:85:BB:99:63:6E:33:C4:B8:64:E9:06:AD:BF:CF
     url = https://192.168.0.15/sdk
     username = administrator@vsphere.local
[vcenter_192_168_0_8]
     root_directory = /opt/bacula/working/vcenter_192_168_0_8
     server = 192.168.0.8
     thumbprint = 04:24:24:13:3C:AD:63:84:A1:9F:E5:14:82:7D:5C:31:25:A8:FA:89
     url = https://192.168.0.15/sdk
     username = administrator@vsphere.local
```

Existing configuration entries can be validated with:

```
[root@localhost bin]# ./vsphere-ctl config check vcenter_192_168_0_26
Connecting to "https://192.168.0.15/sdk"...
OK: successful connection
Computing thumbprint of host "192.168.0.26"
OK: local thumbprint matches server thumbprint
Checking user privileges...
OK: user has all necessary privileges
```

Finally, you can delete configuration entries with:

```
[root@localhost bin]# ./vsphere-ctl config delete --entry vcenter_192_168_0_8
OK: deleted entry "vcenter_192_168_0_8".
```

Manual Configuration

If you have access to BWeb and would like to use it to configure the connection to VMware infrastructure and to configure your backup jobs, skip ahead to vspherebweb.

> **Attention: New in version 12.8.0**
>
> Starting with version 12.8.0 it is possible to use the config command described in Usingvsphere-ctlConfigCommand to interactively create a configuration in the command-line. Otherwise continue from here to configure the plugin directly with your text editor.

The vSphere network access to your ESXi or vCenter server is configured in `/opt/bacula/etc/vsphere_global.conf`.

```
% cat /opt/bacula/etc/vsphere_global.conf
[vsphere]
        username = root
        password = vspherepassword
        server = 192.168.0.1
        url = https://192.168.0.1/sdk
        thumbprint =␣
↪34:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:09
        root_directory = /opt/bacula/working/vsphere
```

Click here (recommended to open in a new tab) to see all available directives for the `vsphere_global.conf` file.

To get the `thumbprint` value, you can copy it from the ESXi console screen. Hit F2 then log in. The thumbprint is displayed in "View Support Information" under the "SSL Thumbprint (SHA1)".

Starting with Bacula Enterprise 8.6, it is also possible to use the `vsphere-ctl` 'thumbprint' command to display the `thumbprint` from the Bacula client.

```
# /opt/bacula/bin/vsphere-ctl thumbprint 192.168.0.1
```

The thumbprint may also be obtained via an ssh session on the ESXi host:

```
# openssl x509 -sha1 -in   \
    /etc/vmware/ssl/rui.crt -noout -fingerprint | cut -d '=' -f 2 "
```

Or on a vCenter server:

```
# openssl x509 -sha1 -in \
    /etc/vmware-vpx/ssl/rui.crt -noout -fingerprint | cut -d '=' -f 2 "
```

Using Multiple vSphere Servers

You may specify multiple vSphere servers in the `vsphere_global.conf` file. When using this feature, you will need to specify the `server=xxx` option in the Plugin Command line. If you have access to BWeb and would like to use it to configure the connection to VMware infrastructure and to configure your backup jobs, skip ahead to vspherebweb. You may also use the CLI automated configuration tool starting with version 12.8.0 in Usingvsphere-ctlConfigCommand.

Click here (recommended to open in a new tab) to see all available directives for the `vsphere_global.conf` file.

It is also mandatory to specify a unique root_directory for each section so that information about VMs from one vCenter or ESXi server is not overwritten with information from a different one.

```
% cat /opt/bacula/etc/vsphere_global.conf
[vsphere]
    username = root
    password = vspherepassword
    server = 192.168.0.1
    url = https://192.168.0.1/sdk
    thumbprint = 01:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:01
    default_datastore = datastore1
    default_restore_host = esx1
    root_directory = /opt/bacula/working/vsphere

[other]
    username = root
    password = vspherepassword
    server = 192.168.0.2
    url = https://192.168.0.2/sdk
    thumbprint = 02:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:01
    default_datastore = abigdatastore
    root_directory = /opt/bacula/working/other
    vddk_backup_transport_mode = san:nbdssl
    vddk_restore_transport_mode = san:nbd

[secure]
    username = root
    hpassword = MTEyOjEyNzoGAwAYFQIVABEDAwcfAhQA
    server = 192.168.0.3
    url = https://192.168.0.3/sdk
    thumbprint = 03:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:01
    default_datastore = abigdatastore
    root_directory = /opt/bacula/working/secure
```

The `[vsphere]` section is optional in the `vsphere_global.conf` file. If not present, make sure the *–server* options is always used for backup (or vsphere-ctl operations).

```
% cat /opt/bacula/etc/conf.d/Fileset-other.conf
Fileset {
  Name = Fileset-other
  Include {
    Plugin = "vsphere: server=other"
  }
```

(continues on next page)

(continued from previous page)

```
}

% /opt/bacula/bin/vsphere-ctl --server other update
1: 1 vm1
2: 2 vm2
```

Click here to see all the vSphere Fileset plugin command options.

Obscure vSphere Password

Starting with the 8.0.3 version of the vSphere Plugin, it is now possible to obscure the vSphere password in the `vsphere_global.conf` file. The obscured password field is called `hpassword`.

Click here (recommended to open in a new tab) to see all available directives for the `vsphere_global.conf` file.

The bconsole `@encode` command can be used to generate the obscured password. Note that if the string you want to encode contains "=", you must use the `string`= keyword in the command.

```
# /opt/bacula/bin/bconsole
* @encode vspherepassword
MTEyOjEyNzoGAwAYFQIVABEDAwcfAhQA

* @encode string="passwordwith="
NTMwOjU0Mzpic2FhZX1gdmV7ZnovAA
```

```
# cat /opt/bacula/etc/vsphere_global.conf
[vsphere]
    username = root
    hpassword = MTEyOjEyNzoGAwAYFQIVABEDAwcfAhQA
    server = 192.168.0.1
    url = https://192.168.0.1/sdk
    thumbprint = 01:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:01
```

vsphere_global.conf Options

Table 23: Sphere Connection Configuration vsphere_global.conf

Option	Re-quire	Default	Info
keep_generat	No	100	Maximum number of Backup between two Full backups
pro-file_all_vm	No	vsphere_all	Internal filename used to store virtual machine profile information
root_director	No	/opt/bacula/	Working directory of the vSphere plugin
vddk_path	No	/opt/bacula/	
vddk_backup	No	ho-tadd:nbdssl	Specify the different transport method to try with the VDDK service during backup. Available with 14.0.1. See the VMWare Disk Transport Library documentation for the list of values.
vddk_restore_	No	ho-tadd:nbdssl	Specify the different transport method to try with the VDDK service during restore Available with 14.0.1. See the VMWare Disk Transport Library documentation for the list of values.
username	Yes		Username allowed to connect to vSphere
password	Yes		Username password allowed to connect to vSphere
hpassword	No		Username obscured password allowed to connect to vSphere. Read more here.
timeout	No	60 seconds	Connection timeout (available since version 8.2.9) used to contact the vSphere server in seconds. The timeout with internal file locking is 10x the value (available since version 8.4.15).
thumbprint	Yes		SSL Thumbprint of the vSphere server (required for vSphere 6.0 and above)
server	Yes		vSphere ESXi server used for Backup
url	Yes		vSphere ESXi or vCenter server URL used for SOAP call
de-fault_datastor	No	datas-tore1	Default datastore for restore
de-fault_restore_	No		Default ESX server used for restore if multiple ones are available in the vCenter
default_ovf	No		Default OVF description used when current OVF fails to be loaded in VMware (available in version 6.2.3-2 and later).
root_director	No	/opt/bacula/	Directory used to store internal plugin data
datas-tore_minimu	No		Minimum space to keep on a Datastore. ex: 5GB
datas-tore_allow_o	No	Yes	Allow to restore a VM using Over Provisioning. When set to `no`, the restore process will ensure that all `full` disks can fit on the Datastore.
datas-tore_refresh_	No	600 seconds	Specify the interval used to refresh storage statistics of the Datastore.
nfc_host_por	No	902	Specify the NFC TCP port to contact the ESX server. Available with 8.4.12.
server_port	No	443	Specify the HTTPS TCP port to contact the ESX server. Available with 8.4.12.
checkssl	No	No	Check SSL certificate with vSphere server. Available with 16.0.3.

The `index` feature will generate records in the Catalog to quickly seek to a given block in the backup stream. The granularity of the index can be controlled with the Storage Daemon device directive `MaximumFileIndex`. The default value is 100MB.

Test vSphere Configuration

To test the vSphere Plugin, you can use the following command as the user that the File Daemon runs as - (usually `root`):

```
# /opt/bacula/bin/vsphere-ctl update
1: 3 squeeze2
2: 4 squeeze.esx
```

The `vsphere-ctl update` command should print a list of all virtual machines that are defined in your ESXi server. If not, check if your credentials in `vsphere_global.conf` are properly set.

Note: When a VM is removed from inventory and then re-imported - even if it is re-imported to the same ESXi host or same vCenter Server - its MoRef will be changed from its previous value. This will require that the `vsphere-ctl update` command is run prior to attempting backup of this VM again. Additionally, a Full backup must be performed when a VM is removed and re-imported to inventory.

If you foresee a use case where VMs are often removed from inventory only to be re-imported at some point, we recommend that the `vsphere-ctl update` command be triggered in an Admin Job's Run-Script which is set to run before the normal nightly backups of your vSphere infrastructure.

The `list` command displays information that is detected on the ESXi hosts and datastores.

```
# /opt/bacula/bin/vsphere-ctl list
Display host list available and their datastores:
 esxi.lan
     datastore1
     datastore2

Will now display configured settings for restore:

No default_restore_host defined in vsphere_global.conf file, trying to
get it from vSphere. Will use restore host esxi.lan

No default_datastore defined in vsphere_global.conf file, trying to
get it from vSphere. Will use datastore datastore1
```

BWeb VMware Center Module

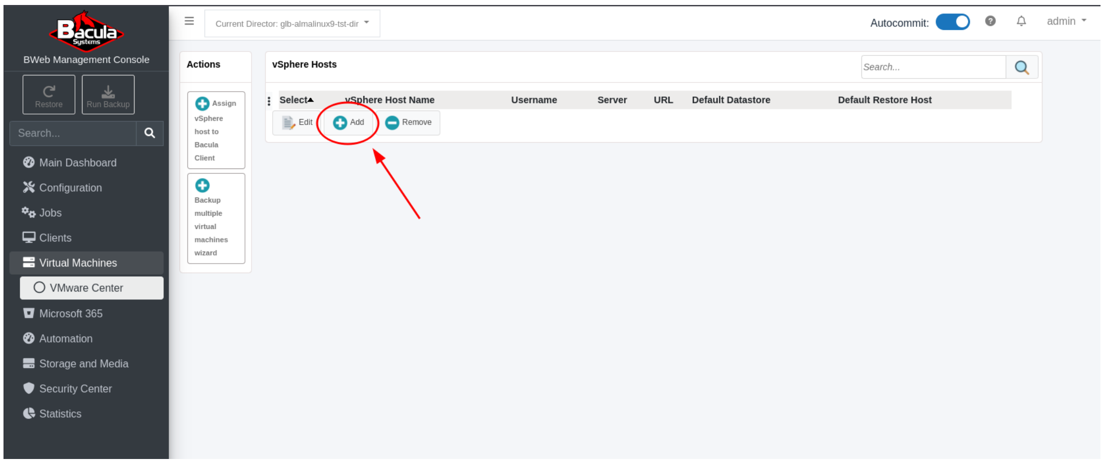
BWeb Management Suite's configuration mode includes a "VMware Center" module which is devoted to integrating a Bacula environment with a vSphere environment.

In VMware Center, all vCenter and/or ESXi hosts may be defined and configured. Once configured, these vSphere hosts may be assigned to individual or multiple Bacula Clients having the vSphere Plugin installed.

Once vSphere hosts are assigned to Bacula Clients, vSphere Plugin Jobs and associated Filesets may be automatically configured so that as VMware VMs appear in a VMware environment, new Jobs/Filesets will be automatically added. Additionally, as VMs are removed from a VMware environment, their Jobs/Filesets may be automatically disabled, or completely removed depending on specific requirements. If preferred, Job and Fileset creation and removal may be performed manually instead.

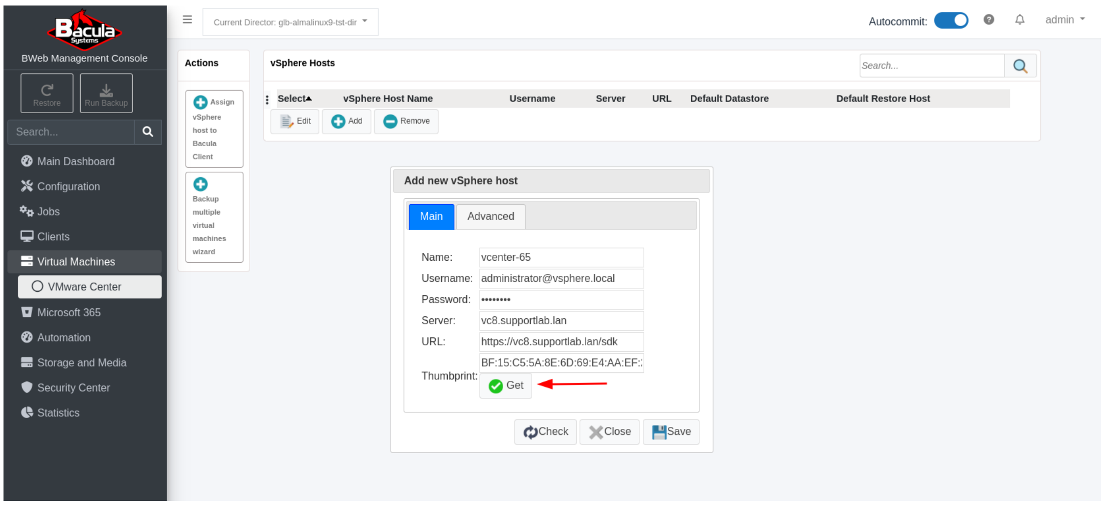
Configuring New vSphere Host

On the BWeb main menu (left panel), expand “Virtual Machines” and click “VMware Center”. You will be taken to the page with the “vSphere Hosts” listing, which should currently be empty.



To add a new vCenter server or specific ESXi host to the list, click on the “Add” icon.

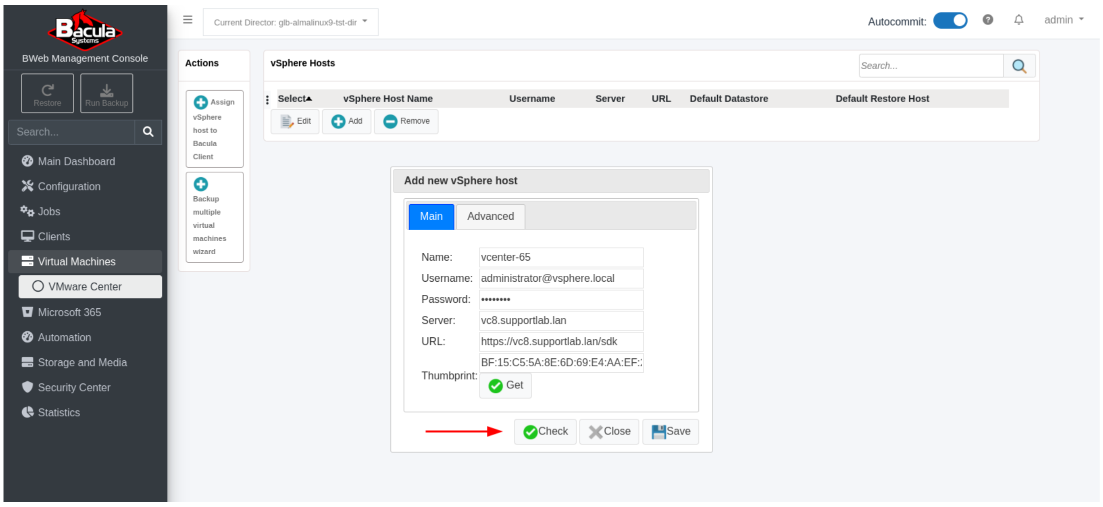
Fill in the fields in the pop-up form.



Name:	vcenter-65
Username:	administrator@vsphere.local
Password:	••••••••
Server:	vc8.supportlab.lan
URL:	https://vc8.supportlab.lan/sdk
Thumbprint:	BF:15:C5:5A:8E:6D:69:E4:AA:EF::

To get the “Thumbprint”, just click on the “Get” icon next to the Thumbprint field. Be sure that the “Server” IP address (or FQDN) is correct. It is important to verify this Thumbprint information now to prevent future issues or potential miscommunications with the wrong vSphere server.

To verify that the information in the form has been filled in correctly for this vSphere host, click on the circular arrows (check) icon, and after a few seconds, the icon should turn into a green check mark as in the image below.

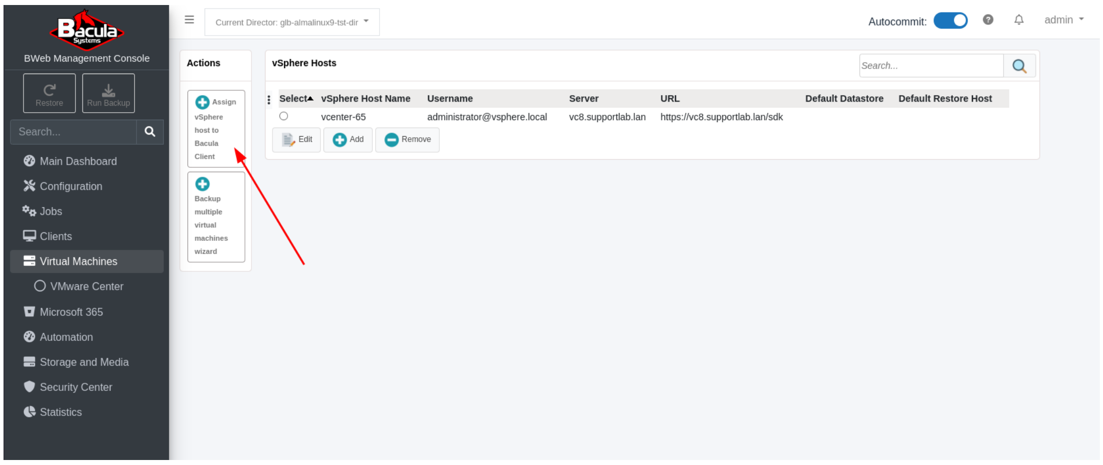


Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host

Add new vSphere host

Name: vcenter-65
Username: administrator@vsphere.local
Password: ••••••••
Server: vc8.supportlab.lan
URL: https://vc8.supportlab.lan/sdk
Thumbprint: BF:15:C5:5A:8E:6D:69:E4:AA:EF:

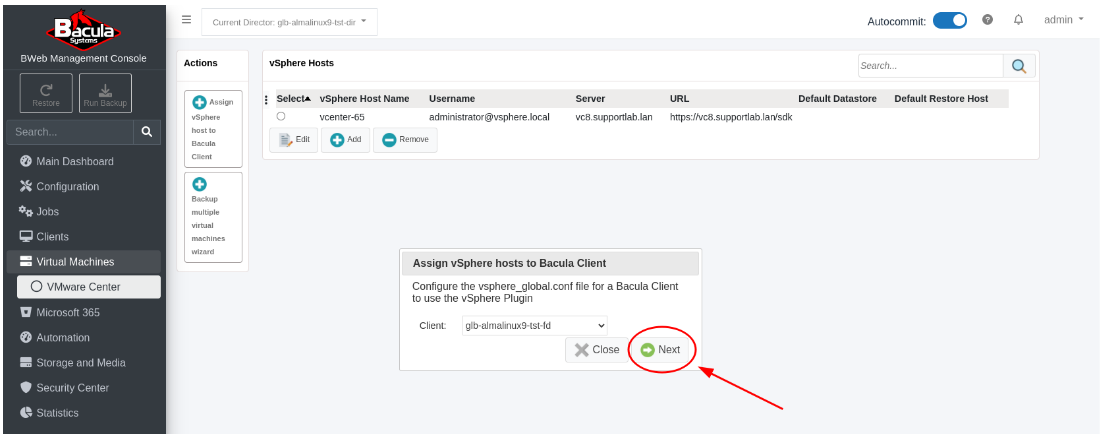
Click “Save” and you will be returned to the “vSphere Hosts” page, which should now be populated with this one (in this case called “vcenter-65”) vCenter host just created.



vSphere Hosts

Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host
○	vcenter-65	administrator@vsphere.local	vc8.supportlab.lan	https://vc8.supportlab.lan/sdk		

Next, we need to assign this vSphere (vCenter) host which we named “vcenter-65” to a Bacula FD with the vSphere plugin installed. To do this, click the “Assign vSphere host to Bacula Client” button in the upper left corner.



vSphere Hosts

Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host
○	vcenter-65	administrator@vsphere.local	vc8.supportlab.lan	https://vc8.supportlab.lan/sdk		

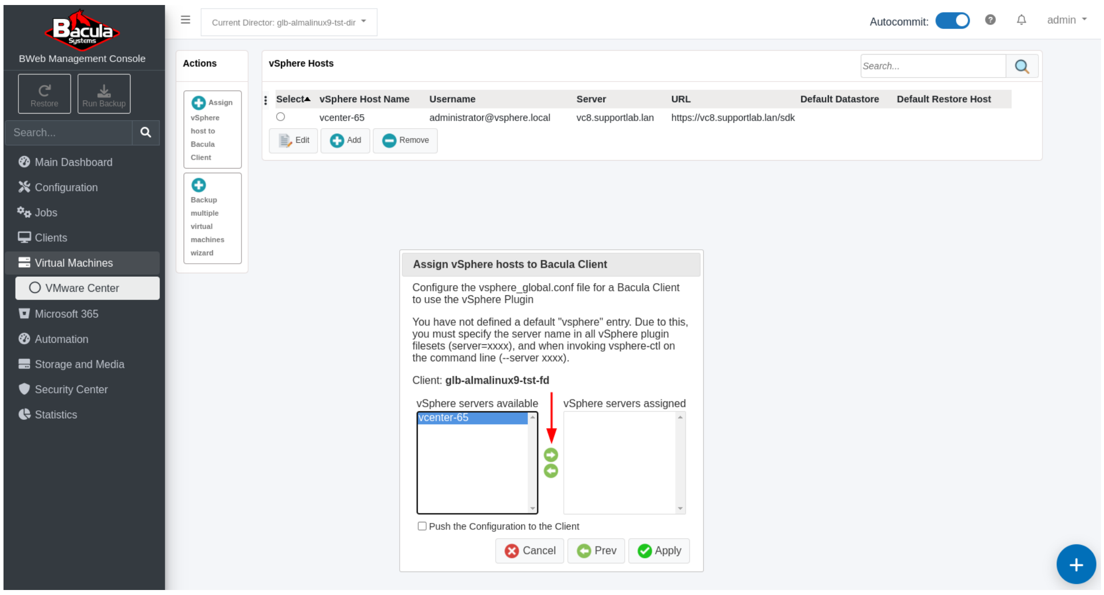
Assign vSphere hosts to Bacula Client

Configure the vsphere_global.conf file for a Bacula Client to use the vSphere Plugin

Client: glb-almalinux9-tst-fd

Select the correct Bacula Client from the “Client” drop-down list. In the screenshot, there is only one Bacula Client configured. A fully configured Bacula environment will have all of the Clients listed here to choose from.

Click “Next”, and you will be presented with a dialog box where the vSphere host(s) (shown here as vcenter-65) are shown in the “vSphere servers available” box. To assign this vSphere host to this FD, highlight it, then click the green arrow pointing to the “vSphere servers assigned” box.



vSphere Hosts

Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host
○	vcenter-65	administrator@vsphere.local	vc8.supportlab.lan	https://vc8.supportlab.lan/sdk		

Assign vSphere hosts to Bacula Client

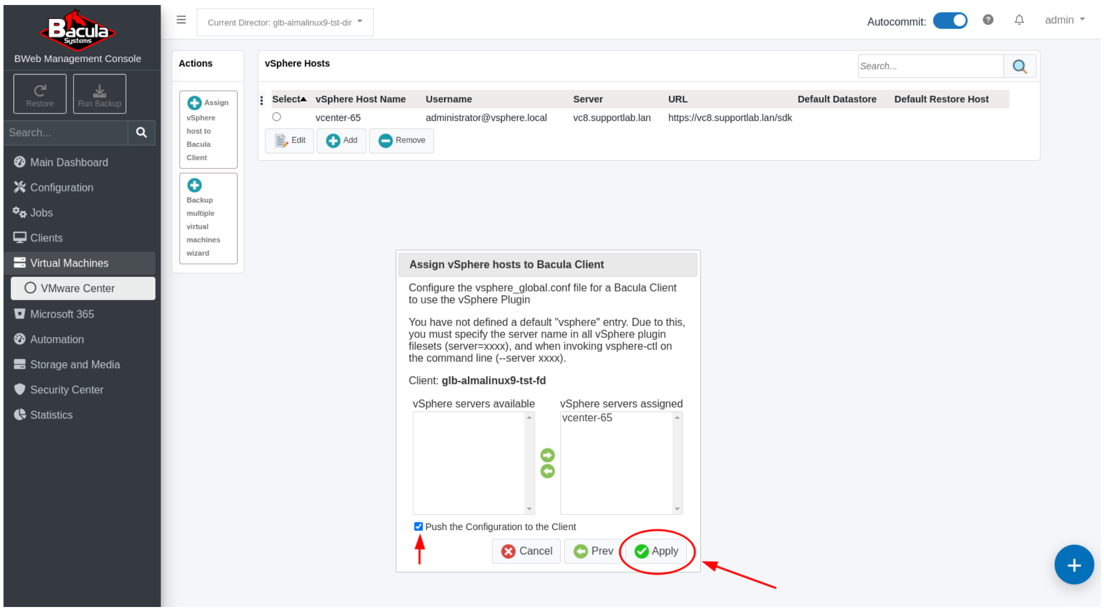
Configure the vsphere_global.conf file for a Bacula Client to use the vSphere Plugin

You have not defined a default "vsphere" entry. Due to this, you must specify the server name in all vSphere plugin filesets (server=xxxx), and when invoking vsphere-ctl on the command line (--server xxxx).

Client: **glb-almalinux9-tst-fd**

vSphere servers available	vSphere servers assigned
vcenter-65	

The “vcenter-65” vSphere host will be added to the “vSphere servers assigned” box on the right.



vSphere Hosts

Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host
○	vcenter-65	administrator@vsphere.local	vc8.supportlab.lan	https://vc8.supportlab.lan/sdk		

Assign vSphere hosts to Bacula Client

Configure the vsphere_global.conf file for a Bacula Client to use the vSphere Plugin

You have not defined a default "vsphere" entry. Due to this, you must specify the server name in all vSphere plugin filesets (server=xxxx), and when invoking vsphere-ctl on the command line (--server xxxx).

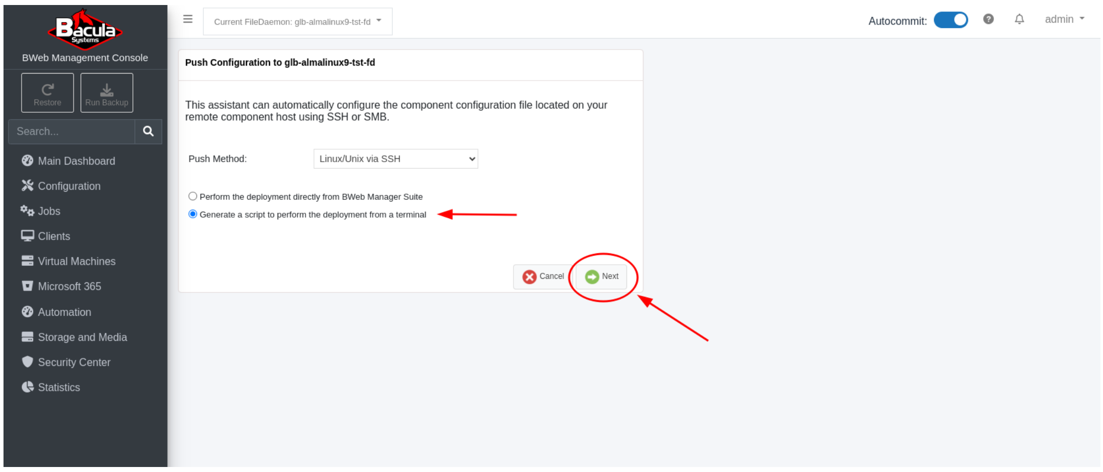
Client: **glb-almalinux9-tst-fd**

vSphere servers available	vSphere servers assigned
	vcenter-65

[✓] Push the Configuration to the Client

Be sure to check the “Push the Configuration to the Client” check box.

Click “Apply” and you will be taken to the “Push Configuration to the Client” page.



Push Configuration to glb-almalinux9-tst-fd

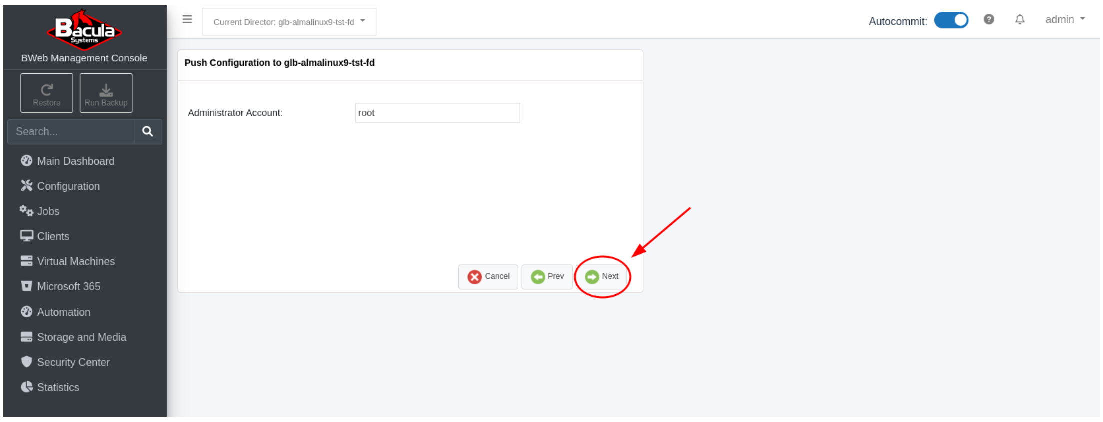
This assistant can automatically configure the component configuration file located on your remote component host using SSH or SMB.

Push Method: Linux/Unix via SSH

○ Perform the deployment directly from BWeb Manager Suite
● Generate a script to perform the deployment from a terminal

The settings on this page refer to the Bacula Client where the FD is running the vSphere plugin. It is the “Client” system we chose previously, and its name is displayed in the header of this dialog box.

Choose the Push Method, set it to Linux and check “Generate a script to perform the deployment from a terminal”. Click “Next”.

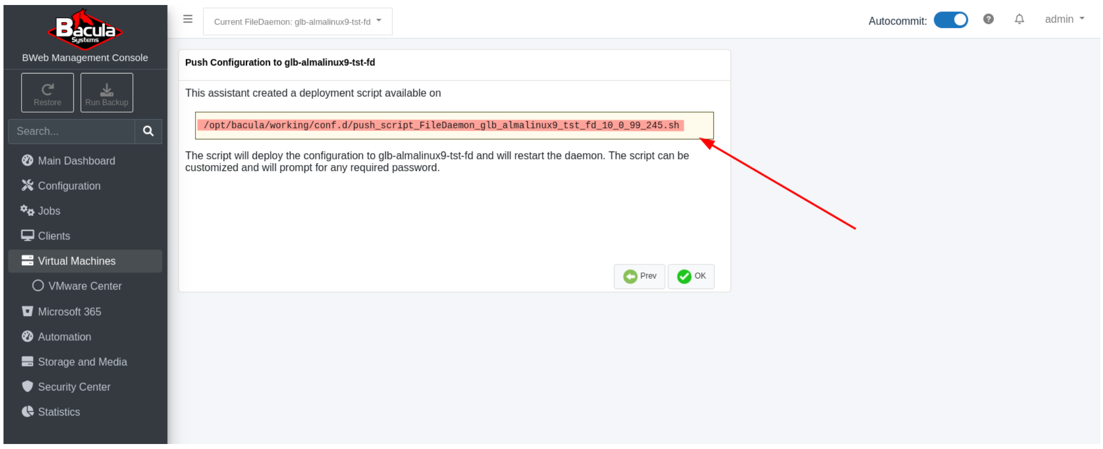


Push Configuration to glb-almalinux9-tst-fd

Administrator Account: root

Set the “Administrator Account” to “root” for this Bacula Client FD.

Click “Next” and you will be presented with the “Push Configuration” dialog box where there will be a Linux command line script in the yellow box.



Push Configuration to glb-almalinux9-tst-fd

This assistant created a deployment script available on

```
/opt/bacula/working/conf.d/push_script_FileDaemon_glb_almalinux9_tst_fd_10_0_99_245.sh
```

The script will deploy the configuration to glb-almalinux9-tst-fd and will restart the daemon. The script can be customized and will prompt for any required password.

Copy the script path to the clipboard.

Next, ssh into your Bacula Director/BWeb server as root:

```
$ ssh root@10.0.99.245
root@10.0.99.245's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Feb  7 10:27:54 2024 from 10.255.0.6
[root@glb-almalinux9-tst ~]#
```

Paste the command copied previously:

```
[root@glb-almalinux9-tst ~]# /opt/bacula/working/conf.d/push_script_
↪FileDaemon_glb_almalinux9_tst_fd_10_0_99_245.sh
```

Hit <enter> and you will see something similar to the following output. This output shows that the script run on the Bacula Director/BWeb server copies a second script to the Bacula Client and then runs that script on the Client via ssh. This second script copies a correctly configured file into the "/opt/bacula/etc" directory on the Client and then restarts the Bacula FD:

```
INFO: Execute user script '/opt/bweb/bin/deploy_script_linux.sh'
INFO: Checking required files on 10.0.99.245
INFO: Copy configuration files
The authenticity of host '10.0.99.245 (10.0.99.245)' can't be established.
ED25519 key fingerprint is SHA256:B+W7gybwB/qYcXWWAMLwR1hElPS6gcupykIgMFtk3PU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.99.245' (ED25519) to the list of known hosts.
root@10.0.99.245's password:
bacula-push-50944.tar                         100%   10KB   9.1MB/s   00:00
INFO: Extract configuration files
INFO: Backing up original configuration from 10.0.99.245
tar: Removing leading `/' from member names
/opt/bacula/working/vsphere_vcenter-65/
tmp.ou8uyzscYk                                100%   36    70.9KB/s   00:00
tar: Removing leading `/' from member names
/opt/bacula/etc/vsphere_global.conf
tar: Removing leading `/' from hard link targets
Shared connection to 10.0.99.245 closed.
Shared connection to 10.0.99.245 closed.
tar: Removing leading `/' from member names
/opt/bacula/working/vsphere_vcenter-65/
/opt/bacula/etc/vsphere_global.conf
Shared connection to 10.0.99.245 closed.
INFO: Clean up temporary config files
Shared connection to 10.0.99.245 closed.
INFO: Restarting  service on 10.0.99.245
INFO: Clean up
```

The Bacula Client is now ready to run backup Jobs of VMware VMs managed by the vCenter server that we named "vcenter-65" in our "vSphere Hosts" listing.

At this point, you may manually create Jobs/Filesets to backup one or more VMs using the vSphere plugin on the Client. See the Creating a vSphere Plugin Backup Job and Fileset in BWeb section.

Alternatively, by clicking the "Backup multiple virtual machines wizard" button on the "VMware Center" page, you may create an automated configuration whereby Bacula will create a new Job and associated Fileset for each VM managed by this “vcenter-65” vCenter server. These Jobs/Filesets may be based on all VMs, some specific VM names, VM names using wildcards, VMs based on “VM Tags”, all VMs on one or more Datastores, or one or more Resource Pools. See the Creating An Automated vSphere Backup Environment Using BWeb section for more information about configuring automatic job creation.

When configured using this automated method, Bacula will automatically add a new Job and associated Fileset for each new VM found, and will disable (or remove) the Job/Fileset for any VM that has been decommissioned and is no longer available on this vCenter server.

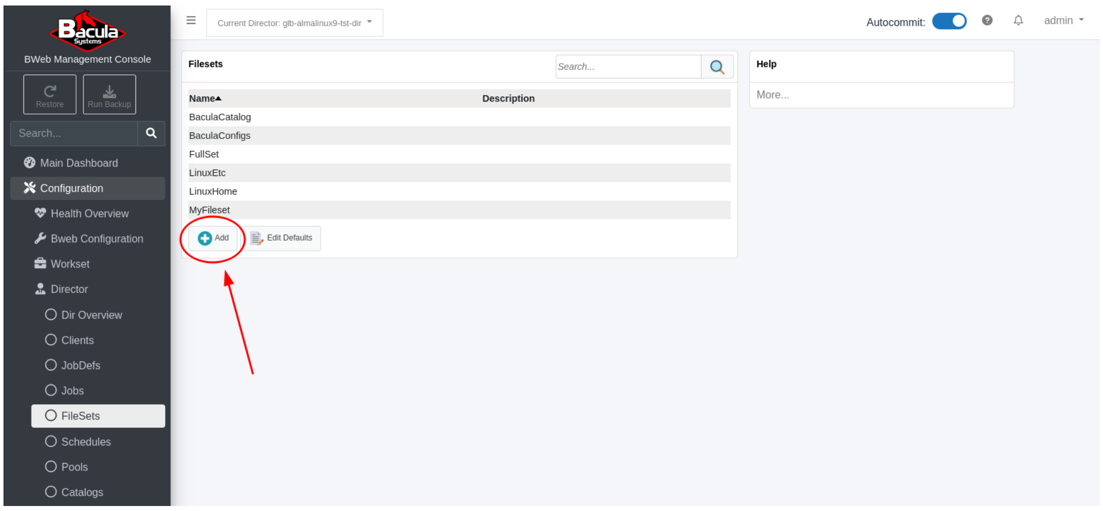
Creating vSphere Plugin Backup Job and Fileset in BWeb

Manually Creating vSphere Plugin Backups With BWeb

Before proceeding with this section, be sure to follow Configuring New vSphere Host article guide where you will be shown how to add vSphere hosts to the BWeb “VMware Center”. The purpose of this section is to demonstrate how to manually create Bacula vSphere Plugin backup Jobs and their associated Filesets to backup your VMware VM(s) with the use of BWeb.

Creating New vSphere Plugin Fileset

On the BWeb main menu, expand “Configuration -> Director” and click “Filesets”. You will be taken a page where all of the currently configured Filesets are listed.

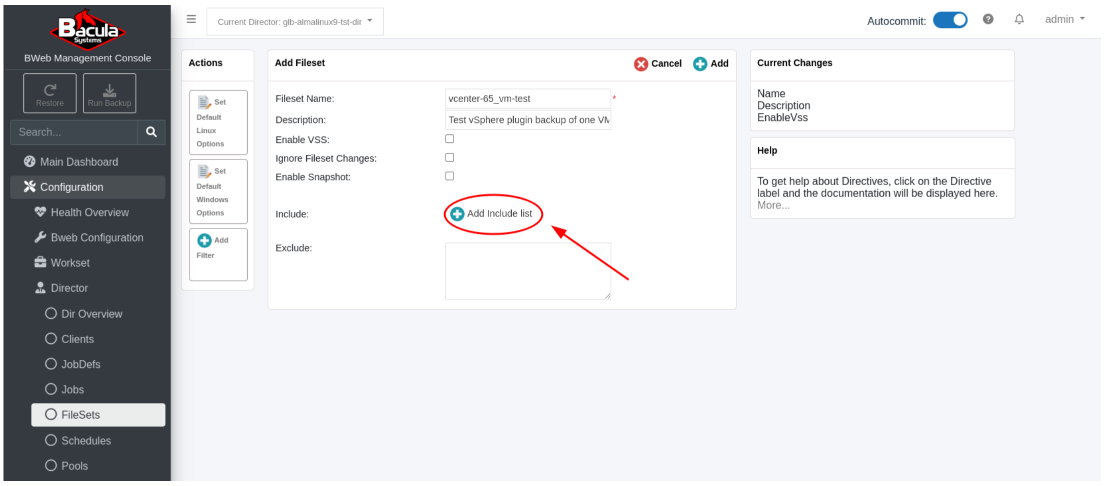


Name▲	Description
BaculaCatalog	
BaculaConfigs	
FullSet	
LinuxEtc	
LinuxHome	
MyFileset	

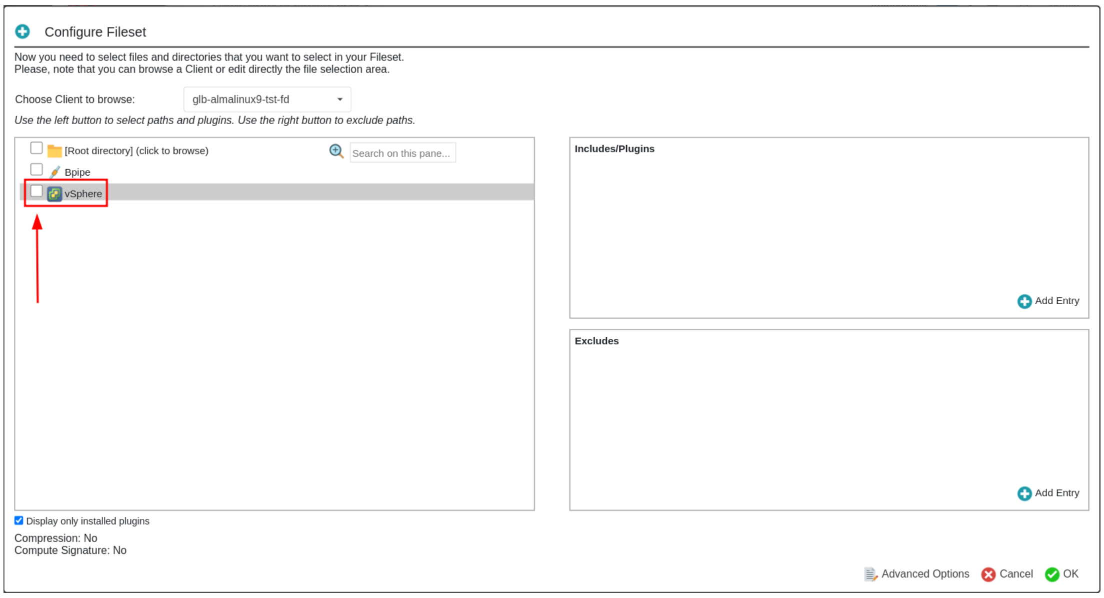
Here, we will start by creating a Fileset which will be used to backup one VMware VM managed by a vCenter server. We will use the ’vcenter-65’ vCenter server which was configured in the “BWeb VMware Center” section of this document.

Click the “+” next to the word “Filesets” in the middle dialog box. You will be taken to a form where this new Fileset will be configured.

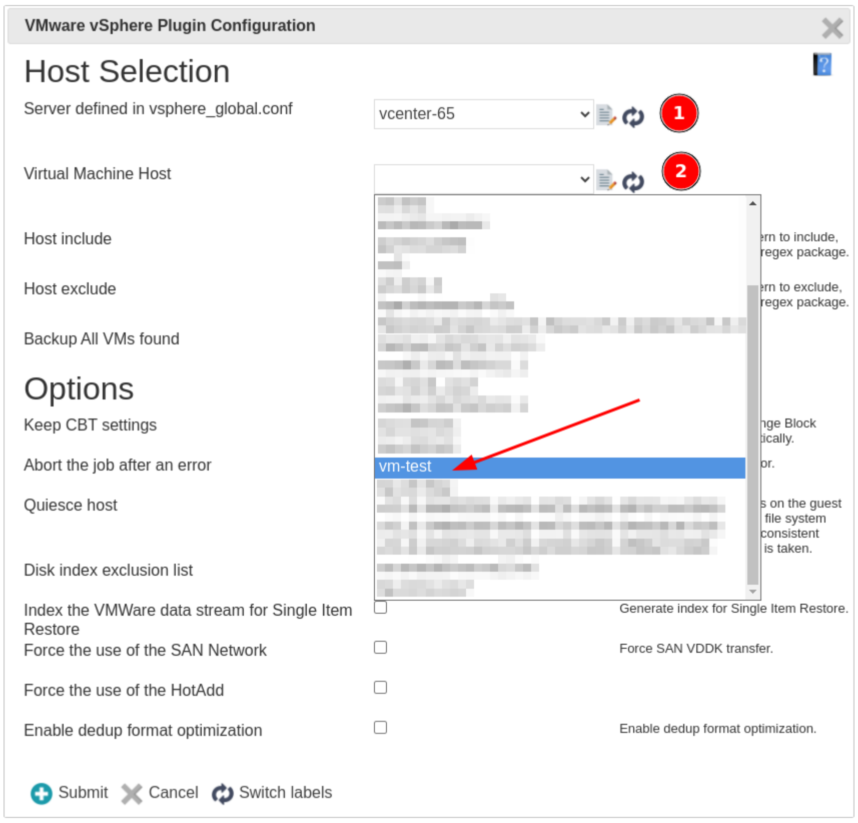
Fill in the “Fileset Name:” field with an appropriate name. In this example, the name “vcenter-65_vm-test” clearly indicates that the VM being backed up is called ‘vm-test’ and it is managed by a vCenter server called ’vcenter-65’. Optionally, you may also fill in the “Description:” field.



Click "+ Add include list" and you will be taken to the "Configure Fileset" dialog box. In this dialog box, select the client from the drop-down menu. Click the "vSphere" checkbox.


Configure Fileset
Now you need to select files and directories that you want to select in your Fileset.
Please, note that you can browse a Client or edit directly the file selection area.
Choose Client to browse:
glb-almalinux9-tst-fd
Use the left button to select paths and plugins. Use the right button to exclude paths.
[Root directory] (click to browse)
Search on this pane...
Bpipe
vSphere
Includes/Plugins
Add Entry
Excludes
Add Entry
Display only installed plugins
Compression: No
Compute Signature: No
Advanced Options
Cancel
OK

You will be taken to the "Plugin Configuration" dialog box. Here, after refreshing clicking the circular arrows 1, choose the 'vcenter-65' server from the "Server:" drop-down menu. Again, refresh the "Virtual Machine" drop-down menu 2 and click the down arrow. You will see a listing of all of the VMs managed by this vCenter server.



Host Selection

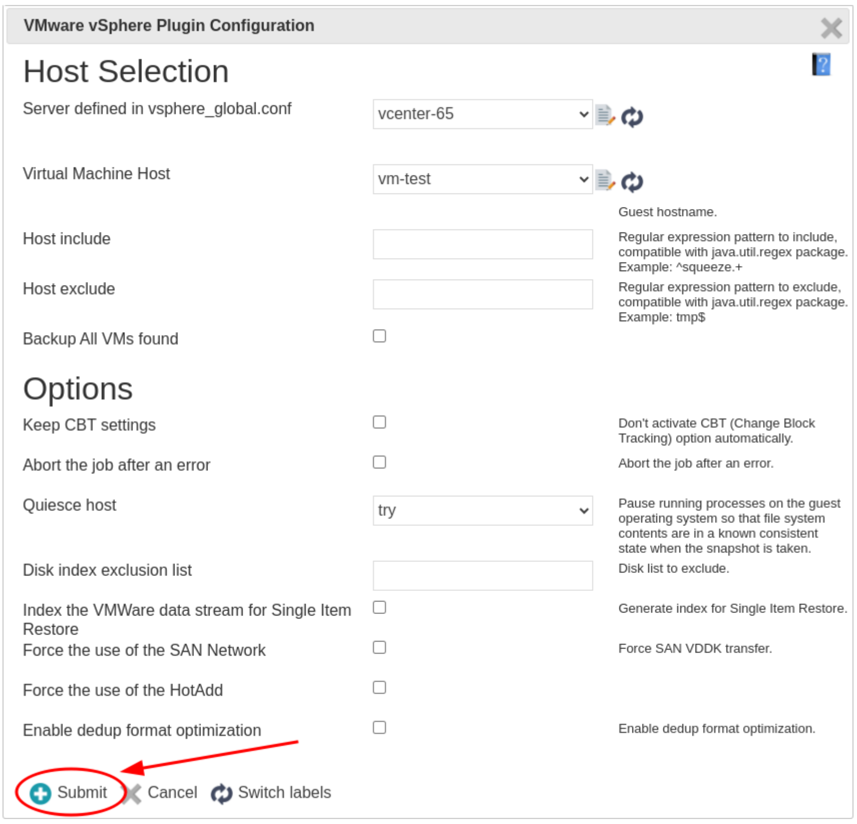
Server defined in vsphere_global.conf	vcenter-65	1
Virtual Machine Host		2
Host include		rn to include, regex package.
Host exclude		rn to exclude, regex package.
Backup All VMs found		

[redacted]
vm-test
[redacted]

Options

Keep CBT settings		nge Block ically.
Abort the job after an error		or.
Quiesce host		s on the guest file system consistent is taken.
Disk index exclusion list		
Index the VMWare data stream for Single Item Restore	☐	Generate index for Single Item Restore.
Force the use of the SAN Network	☐	Force SAN VDDK transfer.
Force the use of the HotAdd	☐	
Enable dedup format optimization	☐	Enable dedup format optimization.

Since we are configuring a new Fileset and Job to backup one specific VM managed by this vCenter server, we will not use the “Host Include” and “Host Exclude” fields. These fields can be used used to backup or exclude VMs based on wildcards like: debian9-* or www-*.



VMware vSphere Plugin Configuration

Host Selection

Server defined in vsphere_global.conf	vcenter-65	
Virtual Machine Host	vm-test	Guest hostname.
Host include		Regular expression pattern to include, compatible with java.util.regex package. Example: ^squeeze.+
Host exclude		Regular expression pattern to exclude, compatible with java.util.regex package. Example: tmp$
Backup All VMs found	☐	

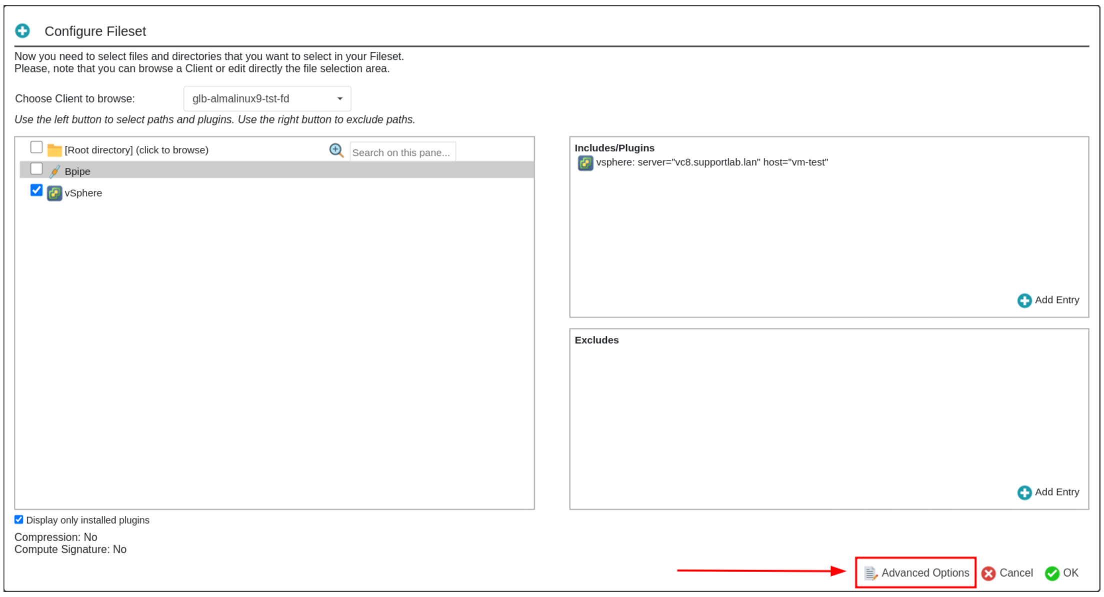
Options

Keep CBT settings	☐	Don't activate CBT (Change Block Tracking) option automatically.
Abort the job after an error	☐	Abort the job after an error.
Quiesce host	try	Pause running processes on the guest operating system so that file system contents are in a known consistent state when the snapshot is taken.
Disk index exclusion list		Disk list to exclude.
Index the VMWare data stream for Single Item Restore	☐	Generate index for Single Item Restore.
Force the use of the SAN Network	☐	Force SAN VDDK transfer.
Force the use of the HotAdd	☐	
Enable dedup format optimization	☐	Enable dedup format optimization.

In this example, we will select the VM named "vm-test", then click "Submit" at the bottom of the dialog box.

You will be taken to the "Configure Fileset" dialog box where you can see in the "Includes/Plugins" box what will be added to the Fileset's "Include:" section. This is called the "vSphere plugin command line". In this case, there are only three simple parameters:

1. The name of the Bacula plugin to use: "vsphere:"
2. The ESXi host or vCenter server to communicate with: "server=vc8.supportlab.lan"
3. The VM to be backed up: "host=vm-test"



Configure Fileset

Now you need to select files and directories that you want to select in your Fileset.
Please, note that you can browse a Client or edit directly the file selection area.

Choose Client to browse: glb-almalinux9-tst-fd

Use the left button to select paths and plugins. Use the right button to exclude paths.

- [ ] [Root directory] (click to browse)
- [ ] Bpipe
- [x] vSphere

Includes/Plugins

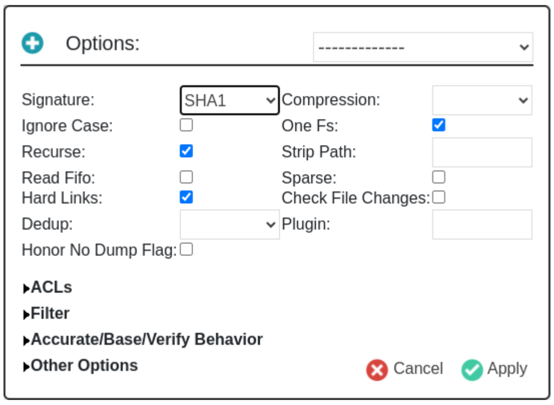
vsphere: server="vc8.supportlab.lan" host="vm-test"

Excludes

- [x] Display only installed plugins

Compression: No
Compute Signature: No

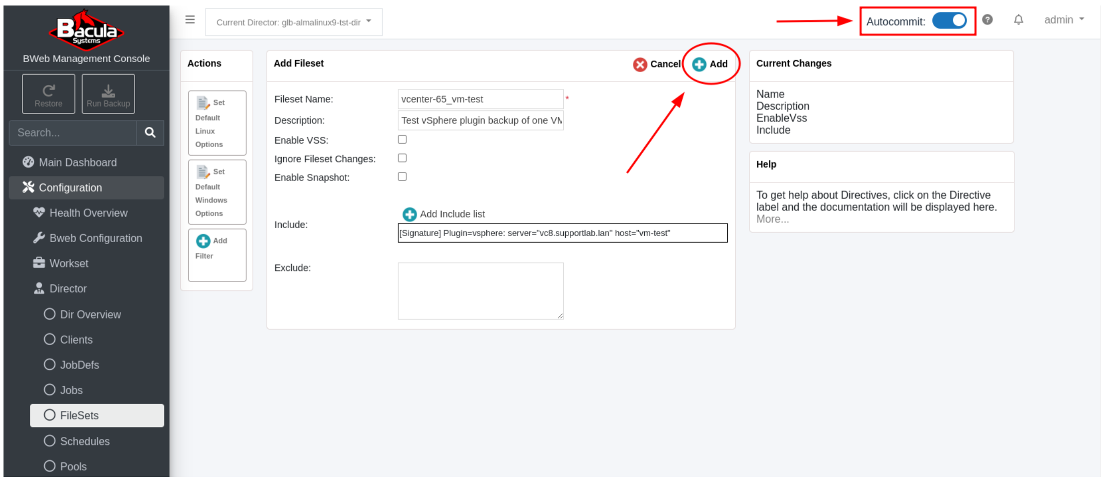
Just click "Advanced Options" here, and you will be taken to the new Fileset's "Options:" dialog box. Here we will set an SHA1 "Signature" for all files backed up by this vSphere plugin backup Fileset. The rest of the options may be left as-is.



Options: -------------

Signature:	SHA1	Compression:	
Ignore Case:	☐	One Fs:	✓
Recurse:	✓	Strip Path:	
Read Fifo:	☐	Sparse:	☐
Hard Links:	✓	Check File Changes:	☐
Dedup:		Plugin:	
Honor No Dump Flag:	☐		

Click "Apply", and you will be taken back to the "Add Fileset" dialog where you can see the newly created Fileset with the vSphere plugin command line and the optional SHA1 "Signature" which was added to the Fileset:

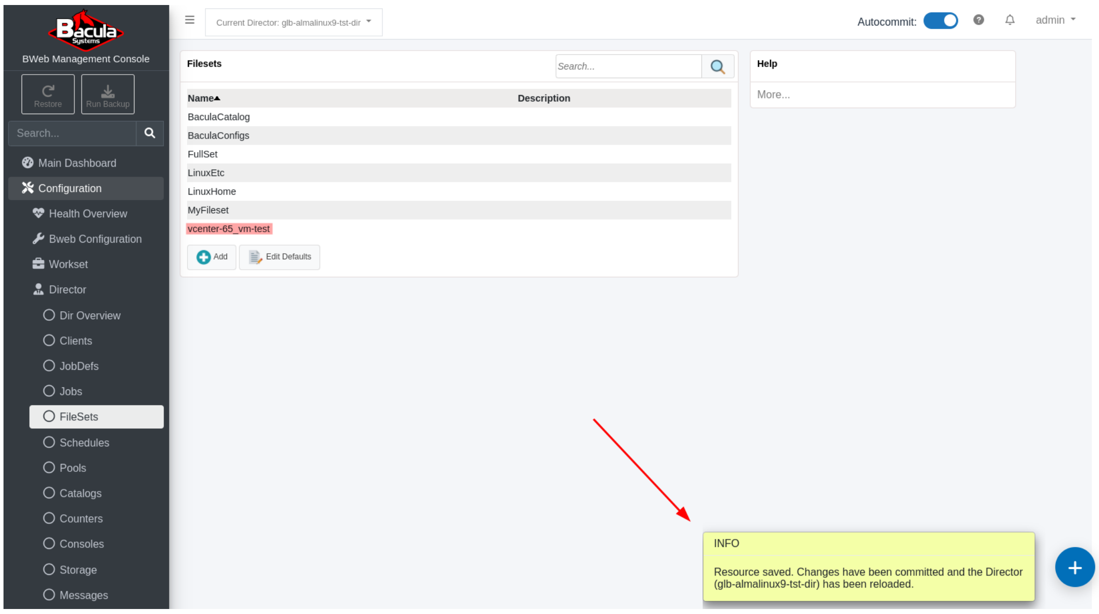


Fileset Name:	vcenter-65_vm-test
Description:	Test vSphere plugin backup of one VM
Enable VSS:	☐
Ignore Fileset Changes:	☐
Enable Snapshot:	☐
Include:	[Signature] Plugin=vsphere: server="vc8.supportlab.lan" host="vm-test"
Exclude:	

Current Changes: Name, Description, EnableVss, Include

To get help about Directives, click on the Directive label and the documentation will be displayed here. More...

Make sure to toggle “Autocommit” in the top-right corner and click “+ Add” at the top of the “Add Fileset” dialog box. You will be taken to the full listing of all configured Filesets where the new ’vcenter-65_vm-test’ Fileset should be listed:



Name▲	Description
BaculaCatalog	
BaculaConfigs	
FullSet	
LinuxEtc	
LinuxHome	
MyFileset	
vcenter-65_vm-test	

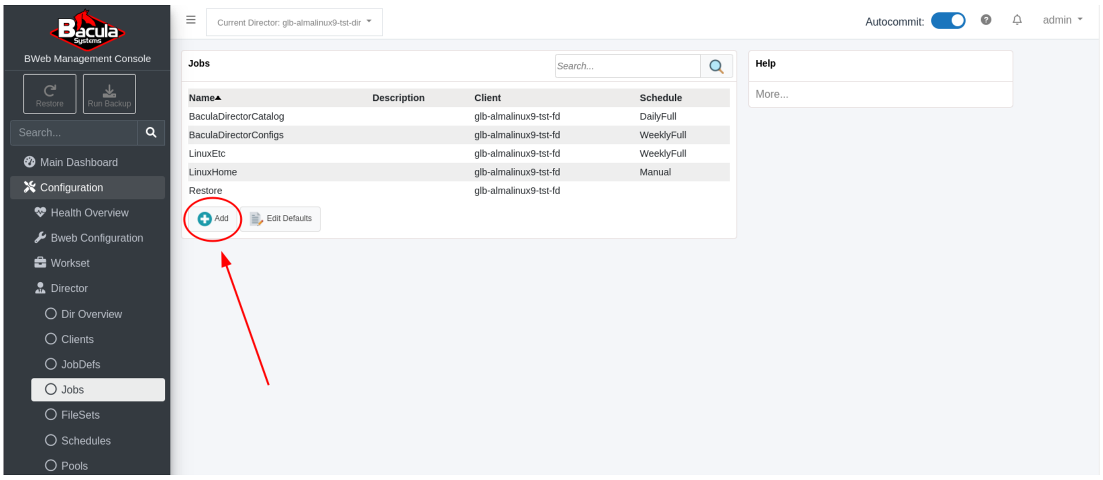
INFO

Resource saved. Changes have been committed and the Director (glb-almalinux9-tst-dir) has been reloaded.

A pop-up window in the bottom-right corner will inform about the successful operation.

Add New Backup Job Which Uses New 'vcenter-65_vm-test' Fileset

On the BWeb main menu, expand "Configuration -> Director" and click "Jobs". You will be taken a page where all of the currently configured Jobs are listed.

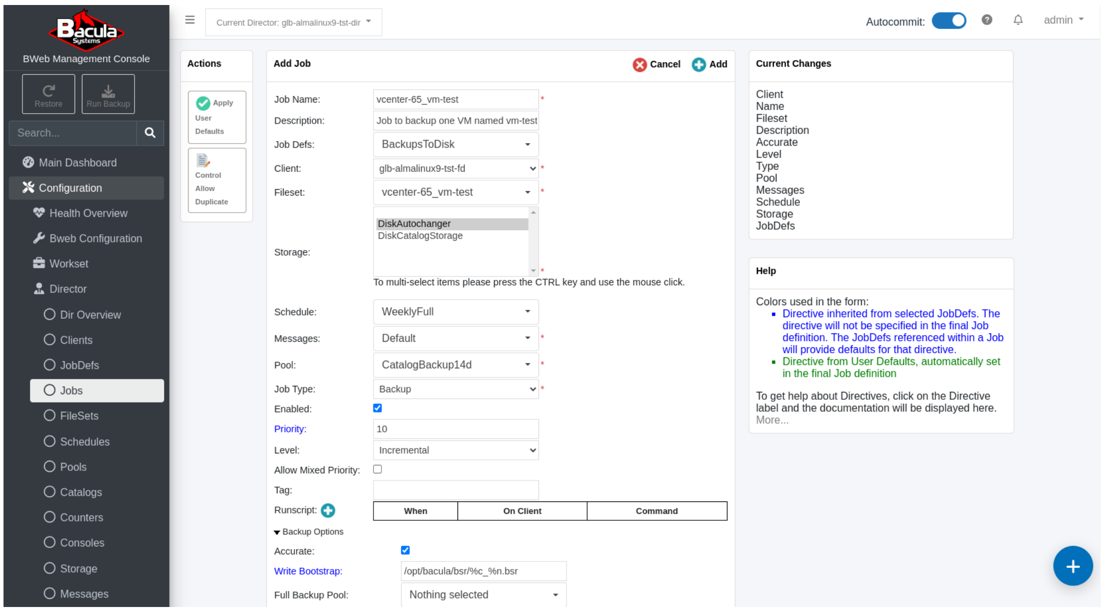


Current Director: glb-almalinux9-tst-dir

Jobs

Name	Description	Client	Schedule
BaculaDirectorCatalog		glb-almalinux9-tst-fd	DailyFull
BaculaDirectorConfigs		glb-almalinux9-tst-fd	WeeklyFull
LinuxEtc		glb-almalinux9-tst-fd	WeeklyFull
LinuxHome		glb-almalinux9-tst-fd	Manual
Restore		glb-almalinux9-tst-fd	

Click the "+ Add" at the bottom of the dialog box. You will be taken to a form where this new Job will be configured.



Current Director: glb-almalinux9-tst-dir

Add Job

Job Name:	vcenter-65_vm-test
Description:	Job to backup one VM named vm-test
Job Defs:	BackupsToDisk
Client:	glb-almalinux9-tst-fd
Fileset:	vcenter-65_vm-test
Storage:	DiskAutochanger, DiskCatalogStorage
Schedule:	WeeklyFull
Messages:	Default
Pool:	CatalogBackup14d
Job Type:	Backup
Enabled:	✓
Priority:	10
Level:	Incremental
Allow Mixed Priority:	
Tag:	
Accurate:	✓
Write Bootstrap:	/opt/bacula/bsr/%c_%n.bsr
Full Backup Pool:	Nothing selected

To multi-select items please press the CTRL key and use the mouse click.

Current Changes
Client
Name
Fileset
Description
Accurate
Level
Type
Pool
Messages
Schedule
Storage
JobDefs

Help

Colors used in the form:
- Directive inherited from selected JobDefs. The directive will not be specified in the final Job definition. The JobDefs referenced within a Job will provide defaults for that directive.
- Directive from User Defaults, automatically set in the final Job definition

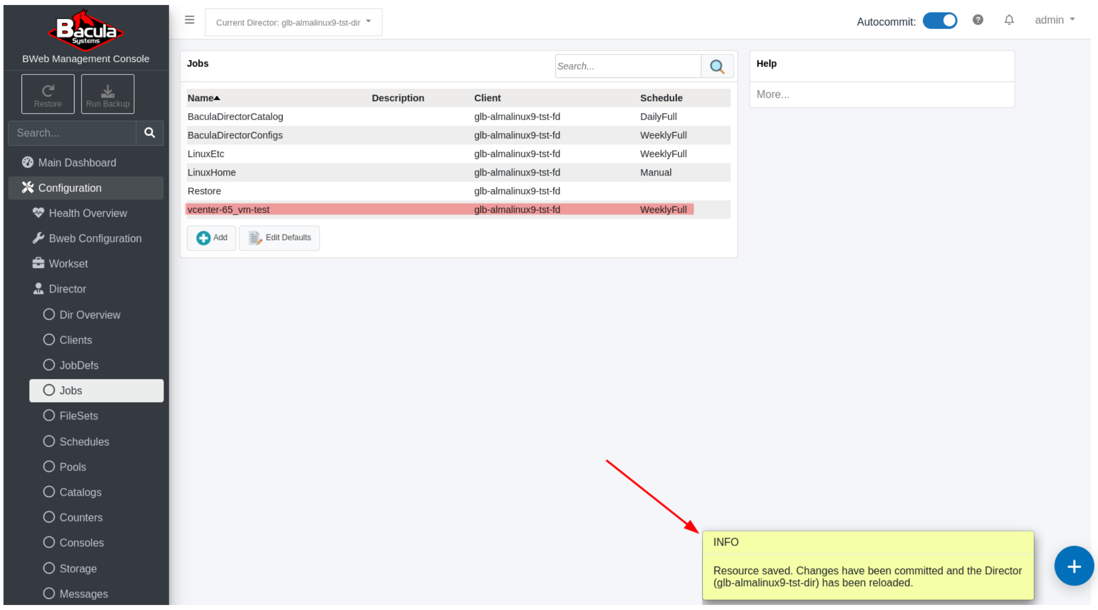
To get help about Directives, click on the Directive label and the documentation will be displayed here.
More...

In this dialog box, the "Job Name:" field is filled in with a name that matches the Fileset we have previously configured. This is not required but can be a good practice to implement that makes it easy to understand what this Job does.

You may choose a pre-configured "JobDef" from the "JobDefs:" drop-down menu. This will automatically populate several of the other fields with some default settings inherited from the selected JobDef. Take a look at all of these and verify that they are OK for your needs. All the defaults from the JobDef can be overridden here.

NOTE: Since we are configuring a vSphere plugin backup job, the "Accurate:" option must be enabled. To display this option, click on "Backup Options" to open and expand this normally hidden section.

Click the checkbox next to “Accurate:”, then click the “+ Add” at the top of this dialog box to save this new Job. You should be taken back to the list of all configured jobs where this new job should be listed.



Name▲	Description	Client	Schedule
BaculaDirectorCatalog		glb-almalinux9-tst-fd	DailyFull
BaculaDirectorConfigs		glb-almalinux9-tst-fd	WeeklyFull
LinuxEtc		glb-almalinux9-tst-fd	WeeklyFull
LinuxHome		glb-almalinux9-tst-fd	Manual
Restore		glb-almalinux9-tst-fd	
vcenter-65_vm-test		glb-almalinux9-tst-fd	WeeklyFull

INFO

Resource saved. Changes have been committed and the Director (glb-almalinux9-tst-dir) has been reloaded.

At this time, the new vSphere VM backup Job is available to be run manually, or via a schedule.

Running New vSphere Plugin Job

On the BWeb main menu, click “Run Backup” in the upper-left corner. In the “Job name:” drop-down menu, you should be able to select your new VMware backup job. Follow the steps in the manual job run wizard to run this job.

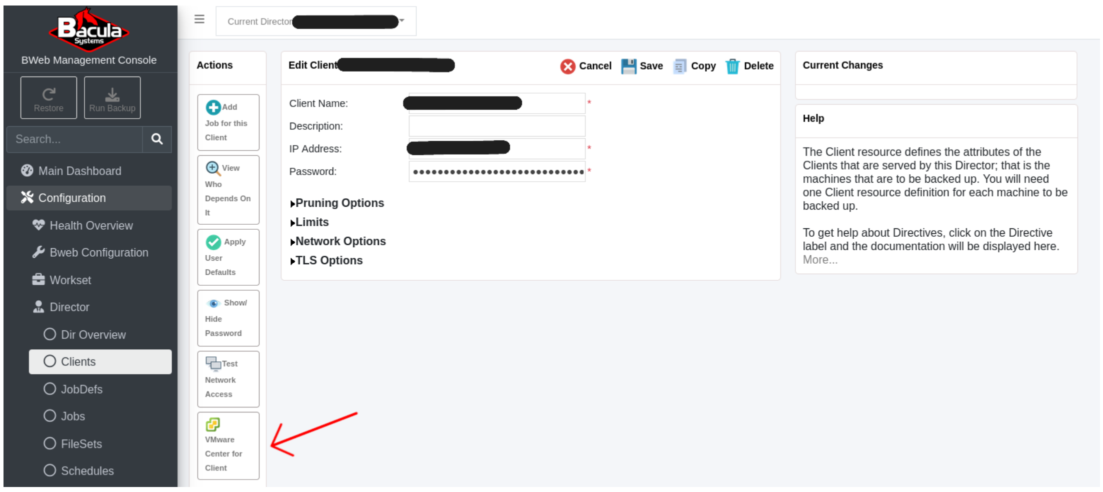
Creating Automated vSphere Backup Environment Using BWeb

The purpose of this wizard is to create an automated configuration whereby all VMs managed by a particular ESXi host or vCenter server are backed up. As new VMs are added, they will automatically be backed up, and as VMs are decommissioned, they will be disabled (or removed) from the backup configuration.

It is possible to exclude VMs from this automatic configuration based on their names, or VM tags, etc., but the steps to do this will come after creation of the default “Backup all VMs” configuration is complete.

On the BWeb main menu, expand “Virtual Machines” and click “VMware Center”. You will be taken to the page with the vSphere Hosts listing, which should currently have at least one vSphere host defined. If there are no hosts in this list, follow the Configuring New vSphere Host article to create one and assign it to a Bacula Client first. Click on the “Backup multiple virtual machines wizard” button on the left.

Note: The menu for VMware Center will appear immediately after the initial VMware backup job has been executed, or by selecting the “VMware Center for Client” button located on the left side under “Configuration -> Director -> Clients”.



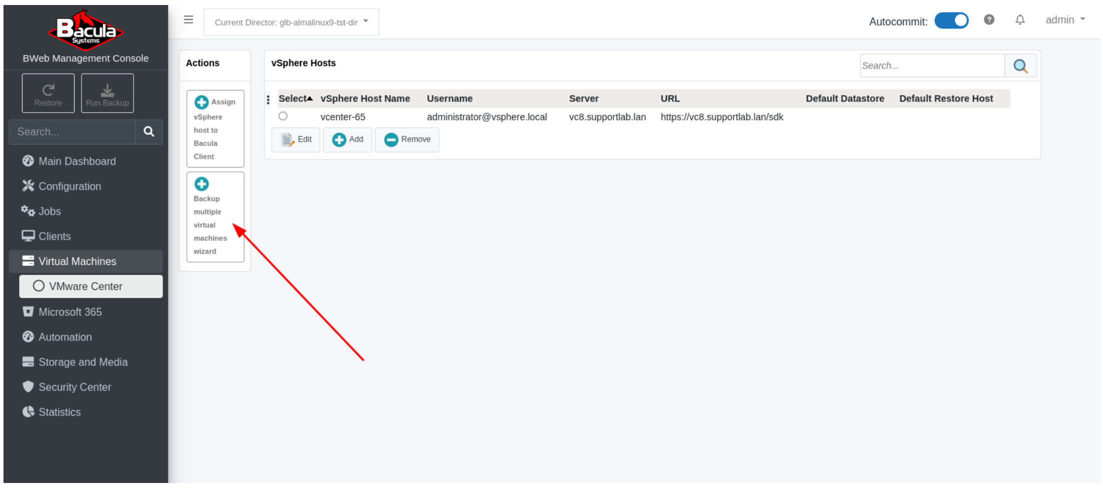
Edit Client [redacted]

Client Name:	[redacted] *
Description:	
IP Address:	[redacted] *
Password:	•••••••••••••••••••••••••••••• *

Help

The Client resource defines the attributes of the Clients that are served by this Director; that is the machines that are to be backed up. You will need one Client resource definition for each machine to be backed up.

To get help about Directives, click on the Directive label and the documentation will be displayed here. More...

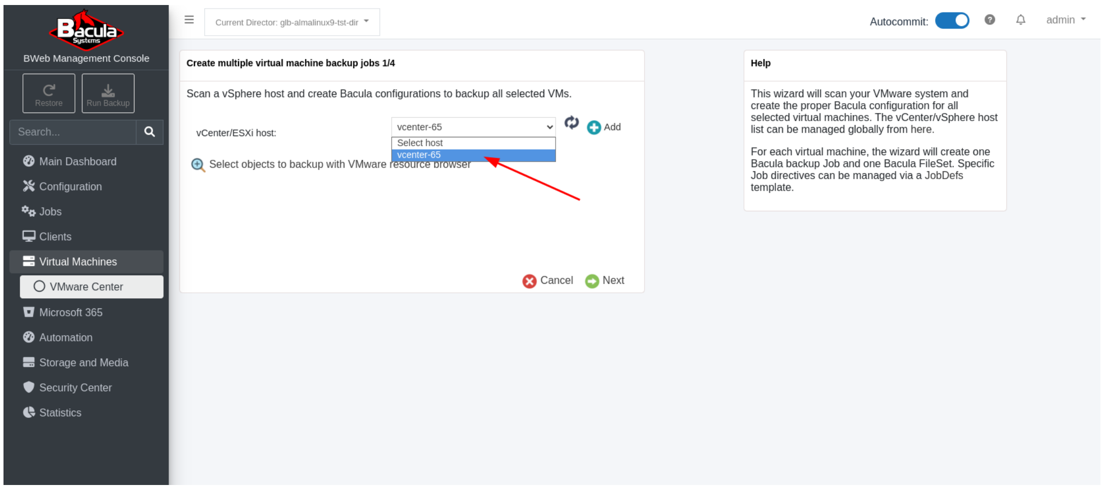


vSphere Hosts

Select	vSphere Host Name	Username	Server	URL	Default Datastore	Default Restore Host
○	vcenter-65	administrator@vsphere.local	vc8.supportlab.lan	https://vc8.supportlab.lan/sdk		

You will be taken to Step 1 of 4 of the Wizard.

Select a vSphere host from the “vCenter/ESXi host” drop-down list.



Create multiple virtual machine backup jobs 1/4

Scan a vSphere host and create Bacula configurations to backup all selected VMs.

vCenter/ESXi host: vcenter-65

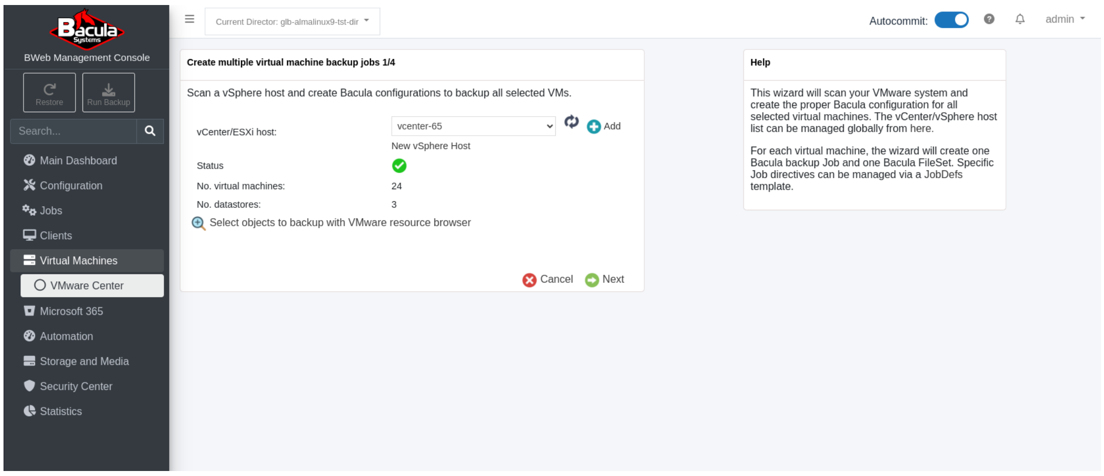
Select objects to backup with VMware resource browser

Help

This wizard will scan your VMware system and create the proper Bacula configuration for all selected virtual machines. The vCenter/vSphere host list can be managed globally from here.

For each virtual machine, the wizard will create one Bacula backup Job and one Bacula FileSet. Specific Job directives can be managed via a JobDefs template.

Click the circular arrows icon to obtain information about the VMs and Storages managed by this vCenter server.


Create multiple virtual machine backup jobs 1/4

Scan a vSphere host and create Bacula configurations to backup all selected VMs.

vCenter/ESXi host: vcenter-65

Status ✓

No. virtual machines: 24

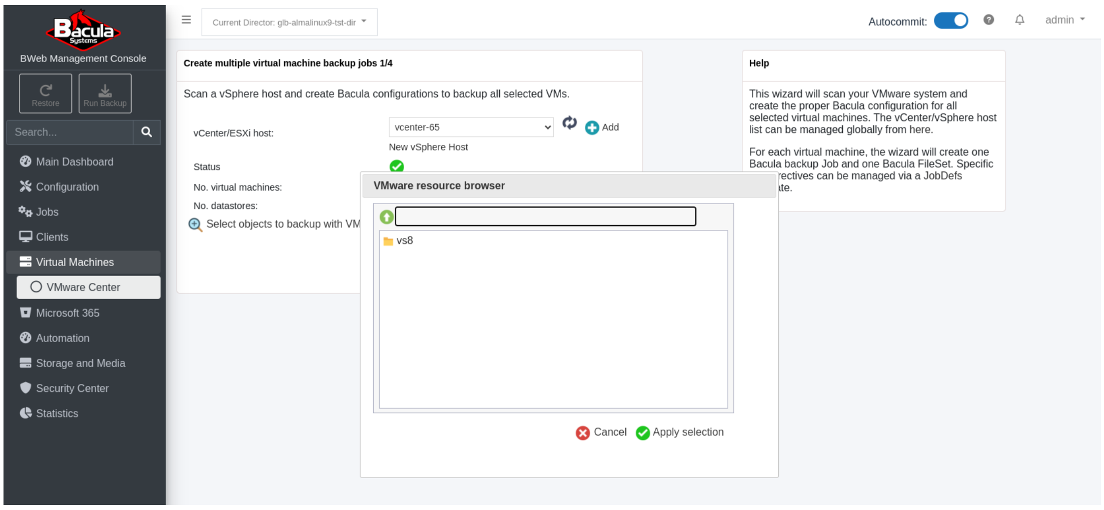
No. datastores: 3

Help

This wizard will scan your VMware system and create the proper Bacula configuration for all selected virtual machines. The vCenter/vSphere host list can be managed globally from here.

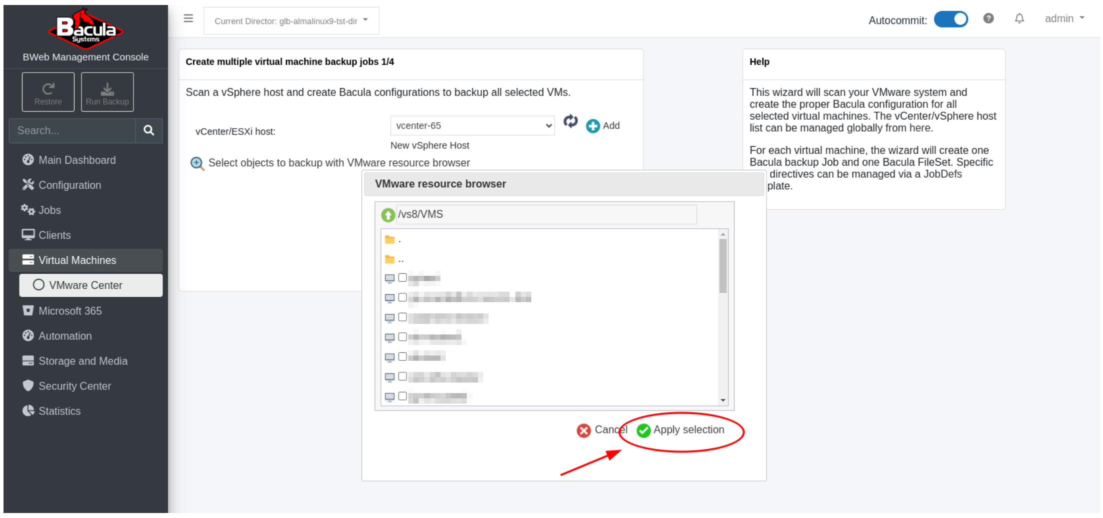
For each virtual machine, the wizard will create one Bacula backup Job and one Bacula FileSet. Specific Job directives can be managed via a JobDefs template.

Click “Select objects to backup with VMware resource browser” and you will be presented with a “VMware resource browser” pop-up dialog. Since this “vcenter-65” vSphere host is a vCenter server, there will initially be a top-level “Datacenter” listed. In this case the Datacenter is called “vc8”.



Click on the Datacenter icon to reveal the VM, ResourcePool, and Datastore folders.

In this example we will be focusing on backing up all VMs managed by this vCenter server.



Create multiple virtual machine backup jobs 1/4

Scan a vSphere host and create Bacula configurations to backup all selected VMs.

vCenter/ESXi host: vcenter-65

Select objects to backup with VMware resource browser

Help

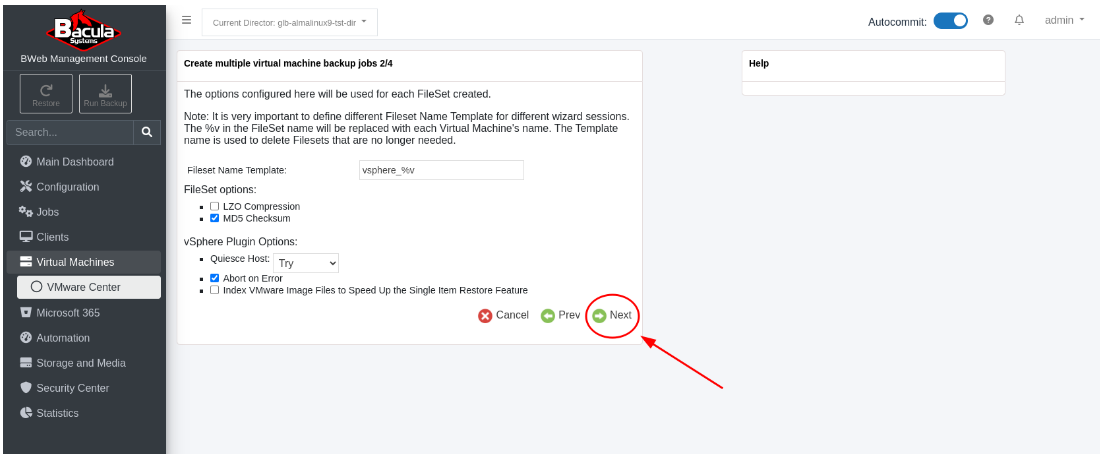
This wizard will scan your VMware system and create the proper Bacula configuration for all selected virtual machines. The vCenter/vSphere host list can be managed globally from here.

For each virtual machine, the wizard will create one Bacula backup Job and one Bacula FileSet. Specific directives can be managed via a JobDefs plate.

Click on the VMs folder icon to see a list of VMs managed by this vCenter host. Do not check any boxes, and simply click the "Apply selection" green check icon.

You will be taken back to the "Create multiple virtual machine backup jobs 1/4" dialog box. Just click "Next" here.

You will be taken to the step 2/4 dialog box where the options for the Filesets to be created may be set.



Create multiple virtual machine backup jobs 2/4

The options configured here will be used for each FileSet created.

Note: It is very important to define different Fileset Name Template for different wizard sessions. The %v in the FileSet name will be replaced with each Virtual Machine's name. The Template name is used to delete Filesets that are no longer needed.

Fileset Name Template: vsphere_%v

FileSet options:

- ☐ LZO Compression
- ☑ MD5 Checksum

vSphere Plugin Options:

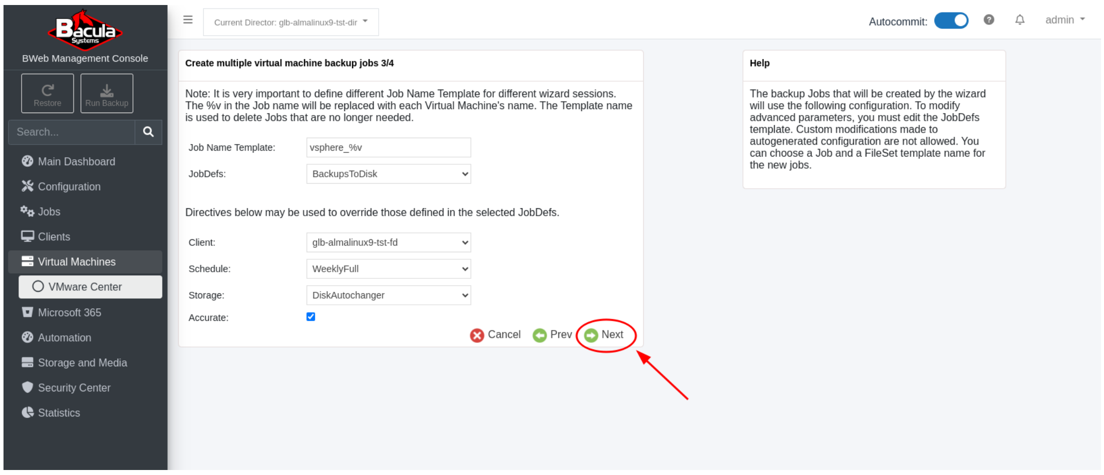
- Quiesce Host: Try
- ☑ Abort on Error
- ☐ Index VMware Image Files to Speed Up the Single Item Restore Feature

Help

In the "Fileset Name Template:", the "%v" will be replaced with the name of each VM found.

Leave the settings at their defaults for now and click "Next".

You will be taken to the step 3/4 dialog box where the options for the Jobs to be created may be set.



Create multiple virtual machine backup jobs 3/4

Note: It is very important to define different Job Name Template for different wizard sessions. The %v in the Job name will be replaced with each Virtual Machine's name. The Template name is used to delete Jobs that are no longer needed.

Job Name Template: vsphere_%v

JobDefs: BackupsToDisk

Directives below may be used to override those defined in the selected JobDefs.

Client: glb-almalinux9-tst-fd

Schedule: WeeklyFull

Storage: DiskAutochanger

Accurate: ✓

Cancel Prev Next

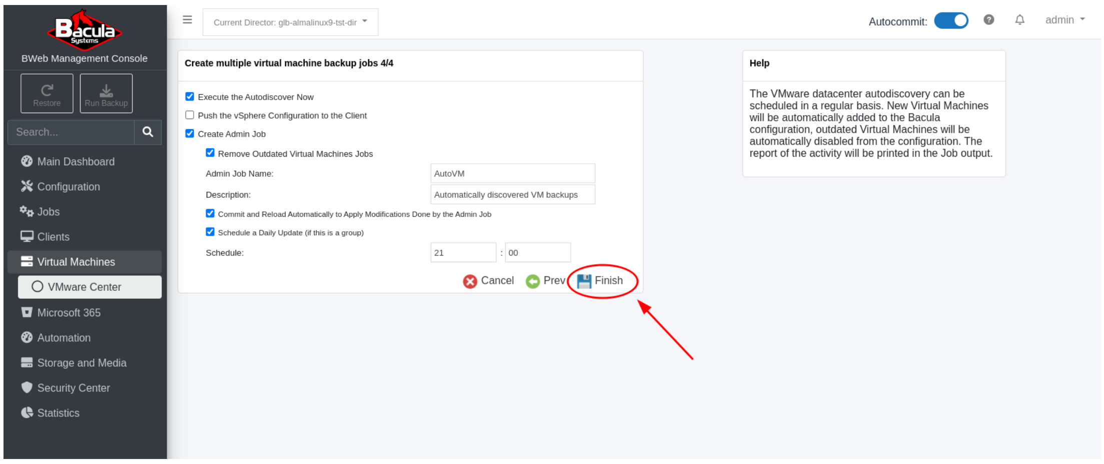
Help

The backup Jobs that will be created by the wizard will use the following configuration. To modify advanced parameters, you must edit the JobDefs template. Custom modifications made to autogenerated configuration are not allowed. You can choose a Job and a FileSet template name for the new jobs.

In the "Job Name Template:", the "%v" will be replaced with the name of each VM found.

Be sure to choose the appropriate Client, Schedule, and Storage for these jobs. If you are performing Incremental or Differential backup of your VMware VMs, then the "Accurate" mode must be enabled, so normally it is recommended to leave this checkbox checked. Click "Next".

You will be take to the step 4/4 dialog box.



Create multiple virtual machine backup jobs 4/4

[✓] Execute the Autodiscover Now

[ ] Push the vSphere Configuration to the Client

[✓] Create Admin Job

[✓] Remove Outdated Virtual Machines Jobs

Admin Job Name: AutoVM

Description: Automatically discovered VM backups

[✓] Commit and Reload Automatically to Apply Modifications Done by the Admin Job

[✓] Schedule a Daily Update (if this is a group)

Schedule: 21 : 00

Cancel Prev Finish

Help

The VMware datacenter autodiscovery can be scheduled in a regular basis. New Virtual Machines will be automatically added to the Bacula configuration, outdated Virtual Machines will be automatically disabled from the configuration. The report of the activity will be printed in the Job output.

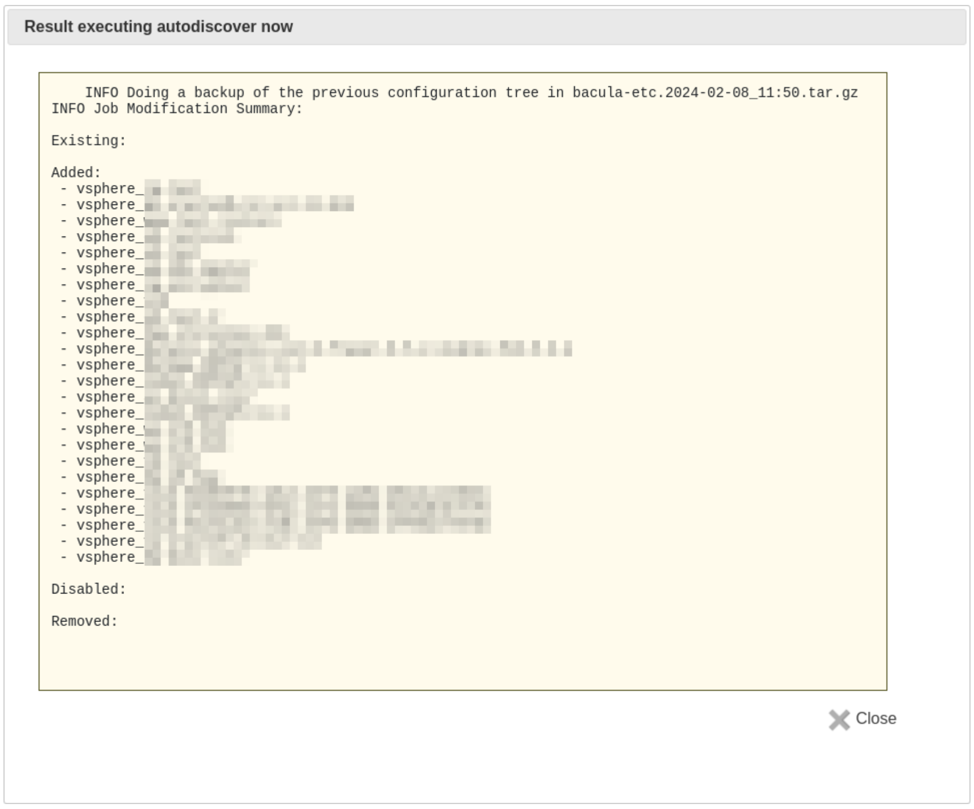
Here, leave the "Execute the Autodiscover Now" checkbox checked, leave the "Push the vSphere Configuration to the Client" Checkbox unchecked, and then check the "Create Admin Job" checkbox - more options will be revealed.

Enter an appropriate "Admin Job Name" and "Description" for the automatic discovery job, then check the "Commit and Reload..." and "Schedule a Daily Update" box - more options will be revealed.

Set the "Schedule" time before your nightly VMware plugin backup jobs are expected to run. This will ensure that the work of creating new Jobs/Filesets for newly discovered VMs and disabling jobs for decommissioned VMs happens before the VMware backup jobs are queued.

Click "Finish".

After a few seconds, the "Result executing autodiscover now" dialog box will pop up showing what has been done.



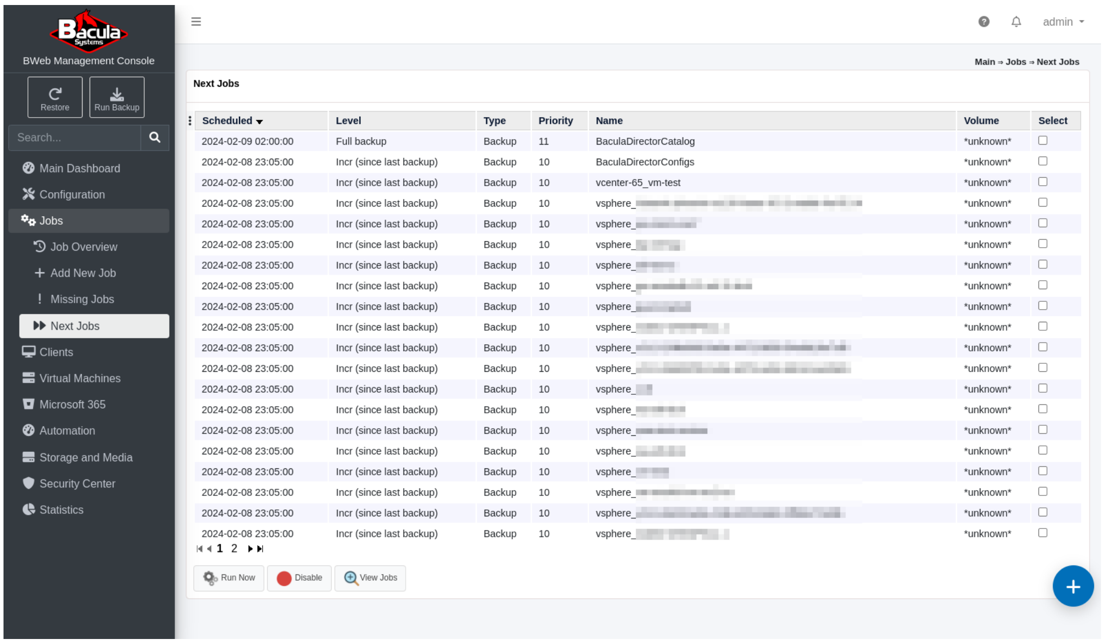
```
    INFO Doing a backup of the previous configuration tree in bacula-etc.2024-02-08_11:50.tar.gz
INFO Job Modification Summary:

Existing:

Added:
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]
 - vsphere_[redacted]

Disabled:

Removed:
```

Notice that BWeb has added Jobs and Filesets for All VMs managed by this vCenter server called “vcenter-65”, and there were no existing Jobs/Filesets discovered, and none removed, nor disabled.

At this time, you may click on “Jobs” or “Filesets”, or “Schedules” to see the new resources created by the Autodiscovery process.

Exit configuration mode by clicking on the “BWeb Management Console” at the top left of the main menu.

Expand “Jobs” and click “Next Jobs”. If you had selected a schedule in step 4/4 above, then you should see your new “vsphere_XXXX” jobs in this list of “Next Jobs”.



Next Jobs

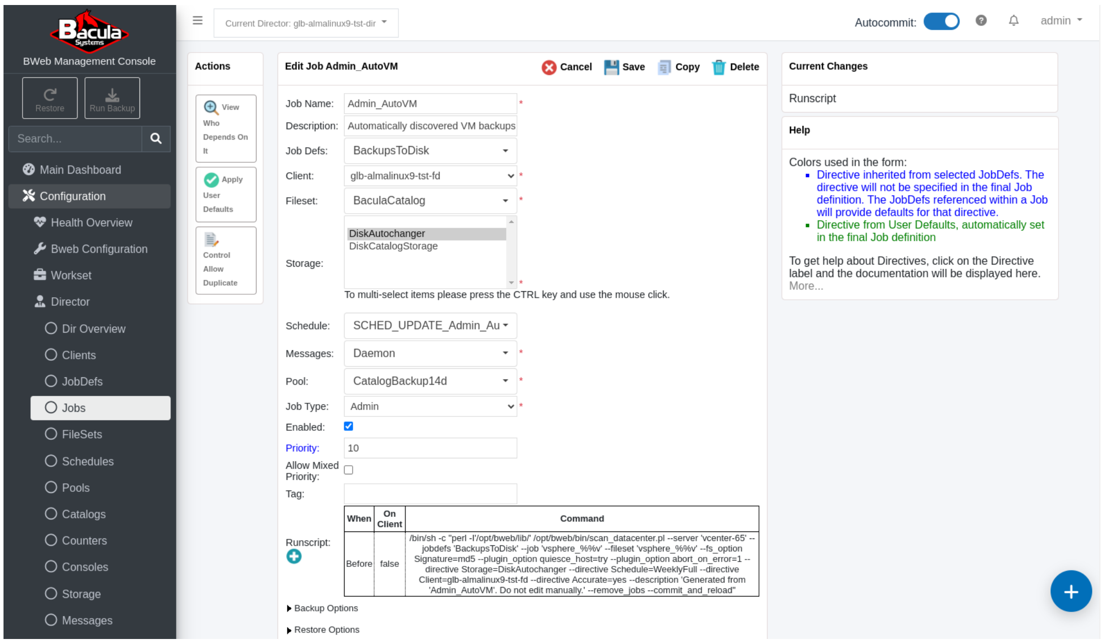
Scheduled	Level	Type	Priority	Name	Volume	Select
2024-02-09 02:00:00	Full backup	Backup	11	BaculaDirectorCatalog	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	BaculaDirectorConfigs	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vcenter-65_vm-test	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐
2024-02-08 23:05:00	Incr (since last backup)	Backup	10	vsphere_[redacted]	*unknown*	☐

Creating Automated vSphere Backup Environment Using Command Line Scripting

Important: Since Bacula version 16.0.7, a new solution has been introduced to replace the scan_datacenter tool. It is highly recommended to use the new solution - *Automatic Objects Integration (Scan Plugin)* as the scan_datacenter will soon be deprecated. See an example for vSphere.

There is a script called "scan_datacenter.pl" that is integral to BWeb's "VMware Center" located in "/opt/bweb/bin". In Step 4 of 4 in the BWeb Automated VMware Backups section of this document, we have already created an Admin type job called "Admin_AutoVM", and scheduled it to run daily at 21:00.

If we take a look at this Job in BWeb (Configuration –> Director –> Jobs –> Admin_AutoVM), we can see that the scan_datacenter.pl script has been automatically configured to be called in a RunScript section of this Admin job. Runscripts are job resources that can run any shell script before or after a backup job or administrative task as in this case.



Edit Job Admin_AutoVM

Job Name:	Admin_AutoVM *
Description:	Automatically discovered VM backups
Job Defs:	BackupsToDisk
Client:	glb-almalinux9-tst-fd *
Fileset:	BaculaCatalog *
Storage:	DiskAutochanger DiskCatalogStorage *
	To multi-select items please press the CTRL key and use the mouse click.
Schedule:	SCHED_UPDATE_Admin_Au
Messages:	Daemon *
Pool:	CatalogBackup14d *
Job Type:	Admin *
Enabled:	✓
Priority:	10
Allow Mixed Priority:	☐
Tag:	

Runscript:

When	On Client	Command
Before	false	/bin/sh -c "perl -I'/opt/bweb/lib/' /opt/bweb/bin/scan_datacenter.pl --server 'vcenter-65' --jobdefs 'BackupsToDisk' --job 'vsphere_%%v' --fileset 'vsphere_%%v' --fs_option Signature=md5 --plugin_option quiesce_host=try --plugin_option abort_on_error=1 --directive Storage=DiskAutochanger --directive Schedule=WeeklyFull --directive Client=glb-almalinux9-tst-fd --directive Accurate=yes --description 'Generated from 'Admin_AutoVM'. Do not edit manually.' --remove_jobs --commit_and_reload"

Current Changes

Runscript

Help

Colors used in the form:
- Directive inherited from selected JobDefs. The directive will not be specified in the final Job definition. The JobDefs referenced within a Job will provide defaults for that directive.
- Directive from User Defaults, automatically set in the final Job definition

To get help about Directives, click on the Directive label and the documentation will be displayed here.
More...

The scan_datacenter.pl script is called with some specific command line parameters that are used to create the new Filesets and Jobs for each VM found.

We can simply cut and paste this working example into a new custom script which can be run manually, run automatically via cron, or via a Bacula Admin type Job as in the Admin_AutoVM Admin Job example.

Here is the command in the Admin Job's RunScript section as it was created by BWeb's VMware Center's "Backup multiple virtual machines wizard" in the previous section of this guide:

```
/bin/sh -c "perl -I'/opt/bweb/lib/' /opt/bweb/bin/scan_datacenter.pl
--server 'vcenter-65' --jobdefs 'DefaultJob' --job 'vsphere_%%v'
--fileset 'vsphere_%%v' --fs_option Signature=md5 --plugin_option
quiesce_host=try --plugin_option abort_on_error=1 --directive
Storage=File1 --directive Schedule=WeeklyFull --directive
Client=glb-almalinux9-tst-fd --directive Accurate=yes --description
'Generated from 'Admin_AutoVM'. Do not edit manually.' --remove_jobs
--commit_and_reload"
```

In this example, there are a number of command line options being used. For example, we can see that for the Fileset name and the Job name, each will be prefaced by "vmware_" - the name of the VM found. This means that each Job/Fileset pair will have the same name, making the correlation between Jobs and Filesets for each VM backup easy.

There are a few Job-specific "–directive" settings specified too. Importantly, "Accurate mode" is enabled for these Jobs - a requirement for Differential and Incremental VMware VM backups. For the "–directive" values, use the upper camel case (camel case with the first letter capitalised) writing method, for example:

```
--directive AllowDuplicateJobs=yes --directive SpoolSize=100000000
```

Also, we can see there are a few vSphere plugin-specific settings specified by using the "–plugin_option" command line parameter multiple times.

And finally, because this Admin type Job will run once per day at 21:00, there are two options specified so that new Jobs and Filesets are automatically committed to the Director's configuration "–commit_and_reload", and also, any VMs that no longer exist have their Fileset and Job resources removed from the Director's configuration due to the "–remove_jobs" option.

A full listing of all available command line options to the scan_datacenter.pl script can be seen simply by running this script with no command line parameters.

The `scan_datacenter.pl` script was introduced with **BWeb** version 8.10. Interim versions required to download and install the Perl SDK from VMware (not longer required from version 12.2.4).

vSphere Plugin Backup Job Configuration

The following chapter presents the information on how to configure a vSphere Plugin Backup Job.

VMware Automatic Integration

Important: Since Bacula version 16.0.7, a new solution has been introduced to replace the scan_datacenter tool. It is highly recommended to use the new solution - *Automatic Objects Integration (Scan Plugin)* as the scan_datacenter will be deprecated. See an example for vSphere.

The `scan_datacenter` BWeb Management Suite module can help you set up the Bacula configuration needed, and protect your VMs in a highly automated way. The `scan_datacenter` module will connect the vCenter/ESXi server to list the VMs to backup.

Note: The BWeb VMware Center article describes in detail how to set up the `scan_datacenter` module.

Job Example

The `Accurate` option is mandatory in the backup Job resource when running Incremental/Differential backup jobs with the vSphere plugin.

```
Job {
  Name = vSphereBackup
  Fileset = vSphere
  Accurate = yes
  ...
}
```

Fileset Example

This section presents various Fileset examples.

```
Fileset {
 Name = vSphere
 Include {
  Options { Signature=MD5 }
  Plugin = "vsphere: host=guest1"
```

(continues on next page)

(continued from previous page)

```
 }
}
```

```
Fileset {
 Name = vSphere
 Include {
  Options { Signature=MD5 }
  Plugin = "vsphere: host=vm-401"
 }
}
```

Important: The vSphere Plugin is not compatible with the `sparse` Fileset option.

Testing Fileset

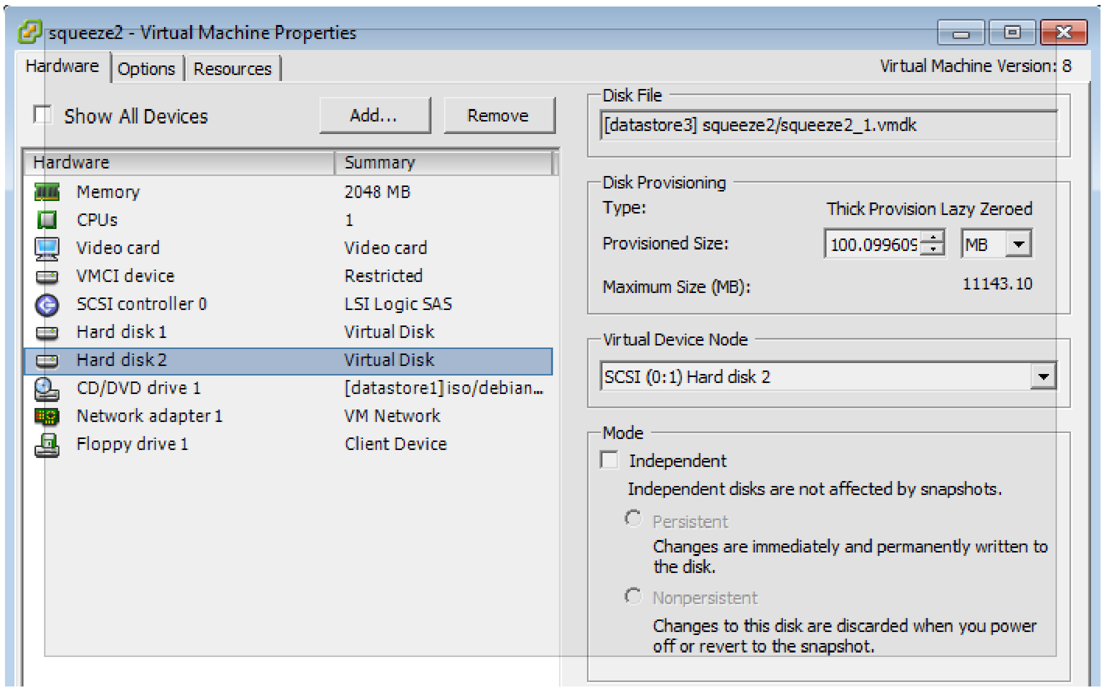
You can use the bconsole `estimate` command to test your Fileset.

```
*estimate listing job=vmware1
Using Catalog "MyCatalog"
Connecting to Client 127.0.0.1-fd at 127.0.0.1:9102
-rwx------  0 root  root           0 18:22:16  /@vsphere/3-squeeze2
-rw-r-----  1 root  root        1923 18:22:19  /@vsphere/3/vmbkp_generation.
↪profile
-rw-r-----  1 root  root        4693 18:22:19  /@vsphere/3/3.ovf
-rw-------  1 root  root  2147483648 18:22:20  /@vsphere/3/0.bvmdk
-rw-------  1 root  root   104857600 18:22:21  /@vsphere/3/1.bvmdk
-rwx------  1 root  root           0 18:22:26  /@vsphere/79-squeeze.esx
-rw-r-----  1 root  root        1806 18:22:29  /@vsphere/79/vmbkp_generation.
↪profile
-rw-r-----  1 root  root        4704 18:22:28  /@vsphere/79/79.ovf
-rw-------  1 root  root   104857600 18:22:29  /@vsphere/79/0.bvmdk
-rw-------  1 root  root   104857600 18:22:30  /@vsphere/79/1.bvmdk
2000 OK estimate files=10 bytes=2,462,069,574
```

Excluding Disk

Exclude Disk with vSphere Console

To exclude a specific disk, you can activate the **independent** mode for the disk in the vSphere console.



squeeze2 - Virtual Machine Properties

Hardware | Options | Resources — Virtual Machine Version: 8

Show All Devices — Add... — Remove

Hardware	Summary
Memory	2048 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
SCSI controller 0	LSI Logic SAS
Hard disk 1	Virtual Disk
Hard disk 2	Virtual Disk
CD/DVD drive 1	[datastore1] iso/debian...
Network adapter 1	VM Network
Floppy drive 1	Client Device

Disk File: [datastore3] squeeze2/squeeze2_1.vmdk

Disk Provisioning
Type: Thick Provision Lazy Zeroed
Provisioned Size: 100.099609 MB
Maximum Size (MB): 11143.10

Virtual Device Node: SCSI (0:1) Hard disk 2

Mode
Independent
Independent disks are not affected by snapshots.
Persistent
Changes are immediately and permanently written to the disk.
Nonpersistent
Changes to this disk are discarded when you power off or revert to the snapshot.

Exclude Disk with Fileset

To find the diskid to use in the disk_exclude option (see the options table), it is possible to use the estimate listing command. 0.bvmdk is the image of the diskid 0.

```
# Will exclude the disk 0 and the disk 2 from "myvm"
Plugin = "vsphere: disk_exclude=0,2 host=myvm"
```

vSphere Fileset Plugin Command Options

Table 24: vSphere Fileset Plugin Command Options

Option	Required	Default	Info	Example
host	No		Guest hostname or a MoRef	host=srv1, host=vm-25
host_inclu	No		Guest pattern to include	host_include=srv3
host_exclu	No		Guest pattern to exclude	host_exclude=srv[12]
disk_exclu	No		Disk list to exclude (available since version 8.4.8)	disk_exclude=0,2,4
disk_inclu	No		Disk list to include (available since version 16.0.8). If used, other disks will be exluded	disk_include=0,2,4
keep_cbt	No		Don't try to activate CBT	keep_cbt
dedup_for	No	No	Control VDDK analyzer for Global Endpoint Deduplication (available since version 12.6).	dedup_format=yes
quiesce_host	Yes	Try	Choose to quiesce guest before the snapshot (Try, yes, no)	quiesce_host=no
server	No	vsphere	Specify a vsphere server	server=vsrv2
debug	No		Enable debug	debug
abort_on_	No		Abort the job after an error (available since version 8.2.5)	abort_on_error
update_time	No	900 seconds	Change initial update timeout	
index	No		Generate index for Single Item Restore (available since version 8.6)	index
force_san	No		Force SAN VDDK transfer (available since version 10.2.3)	

Note that `host_include` and `host_exclude` are Java regexp patterns.

The `index` feature will generate records in the Catalog to quickly seek to a given block in the backup stream. The granularity of the index can be controlled with the Storage Daemon device directive `MaximumFileIndex`. The default value is 100MB.

If used together disk_include and disk_exclude affecting the same disk, disk_include takes precedence and the disk will be considered into the backup.

Going back to the Configuration chapter

To go back to the main Configuration chapter, click here.

Operations

This article describes details regarding backup, restore, quiescing guests, VM Instant Recovery and Single Item Restore with Bacula Enterprise vSphere Plugin.

Backup

Example of a Backup Job

```
*list joblog jobid=21509
+-----------------------------------------------------------------------------
↪-----------------------+
| logtext                                                                    ␣
↪                       |
+-----------------------------------------------------------------------------
↪-----------------------+
| bp-vsir-bweb102-dir JobId 21509: No prior or suitable Full backup found in␣
↪catalog. Doing FULL backup. |
| bp-vsir-bweb102-dir JobId 21509: Start Backup JobId 21509, Job=000_bp-o9-
↪hap_vsphere.2023-11-15_13.45.06_49 |
| bp-vsir-bweb102-dir JobId 21509: Connected to Storage "LocalGED" at 10.0.98.
↪5:9103 with TLS          |
| bp-vsir-bweb102-dir JobId 21509: Using Device "LocalGED-04" to write.      ␣
↪                         |
| bp-vsir-bweb102-dir JobId 21509: Connected to Client "bp-vsir-bweb1004-fd"␣
↪at bp-vsir-bweb1004:9102 with TLS |
| bp-vsir-bweb102-fd JobId 21509: Connected to Storage at 10.0.98.5:9103 with␣
↪TLS                      |
| bp-vsir-bweb102-sd JobId 21509: Volume "ged-3474" previously written,␣
↪moving to end of data.         |
| bp-vsir-bweb102-sd JobId 21509: Ready to append to end of Volume "ged-3474"␣
↪size=5,347,408           |
| bp-vsir-bweb102-fd JobId 21509: Backup "bp-o9-hap" (vm-7253) start.        ␣
↪                         |
| bp-vsir-bweb102-fd JobId 21509: Activating vSphere "Change Tracking System"␣
↪                         |
| bp-vsir-bweb102-fd JobId 21509: To activate properly CBT, a Snapshot␣
↪creation/deletion cycle will now be performed. |
| bp-vsir-bweb102-fd JobId 21509: Create snapshot name 000_bp-o9-hap_vsphere.
↪2023-11-15_13.45.06_49 succeeded. |
| bp-vsir-bweb102-fd JobId 21509: There are 2 disks.                         ␣
↪                         |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3474" Bytes=1,
↪137,843,500 Blocks=88 at 15-Nov-2023 13:47. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3478"                ␣
↪                         |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3478" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3478" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 13:47. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3478" Bytes=1,
```

(continues on next page)

(continued from previous page)

```
↪077,411,747 Blocks=15 at 15-Nov-2023 13:53. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3479"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3479" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3479" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 13:53. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3479" Bytes=1,
↪113,849,764 Blocks=15 at 15-Nov-2023 13:58. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3480"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3480" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3480" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 13:58. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3480" Bytes=1,
↪100,546,012 Blocks=18 at 15-Nov-2023 14:03. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3484"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3484" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3484" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:03. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3484" Bytes=1,
↪075,511,280 Blocks=16 at 15-Nov-2023 14:07. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3485"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3485" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3485" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:07. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3485" Bytes=1,
↪129,119,668 Blocks=15 at 15-Nov-2023 14:11. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3486"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3486" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3486" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:11. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3486" Bytes=1,
↪091,633,114 Blocks=15 at 15-Nov-2023 14:15. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3487"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3487" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3487" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:15. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3487" Bytes=1,
↪089,667,036 Blocks=20 at 15-Nov-2023 14:18. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3488"                    ␣
↪                            |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3488" on Dedup device
```

(continues on next page)

(continued from previous page)

```
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3488" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:18. |
| bp-vsir-bweb102-sd JobId 21509: End of medium on Volume "ged-3488" Bytes=1,
↪127,284,756 Blocks=22 at 15-Nov-2023 14:22. |
| bp-vsir-bweb102-dir JobId 21509: Recycled volume "ged-3489"                    ␣
↪                           |
| bp-vsir-bweb102-sd JobId 21509: Recycled volume "ged-3489" on Dedup device
↪"LocalGED-04" (/bck_2/ged), all previous data lost. |
| bp-vsir-bweb102-sd JobId 21509: New volume "ged-3489" mounted on device
↪"LocalGED-04" (/bck_2/ged) at 15-Nov-2023 14:22. |
| bp-vsir-bweb102-fd JobId 21509: VDDK Transport "nbdssl" selected               ␣
↪                           |
| bp-vsir-bweb102-fd JobId 21509: Dump vmdk 6000C291-8cef-5868-9f15-
↪bea1f333c165 succeeded.           |
| bp-vsir-bweb102-fd JobId 21509: VDDK Transport "nbdssl" selected               ␣
↪                           |
| bp-vsir-bweb102-fd JobId 21509: Dump vmdk 6000C299-88b4-26f3-3145-
↪c18324d30940 succeeded.           |
| bp-vsir-bweb102-fd JobId 21509: Delete snapshot 000_bp-o9-hap_vsphere.2023-
↪11-15_13.45.06_49 succeeded. |
| bp-vsir-bweb102-fd JobId 21509: BACKUP OK bp-o9-hap (vm-7253)                  ␣
↪                           |
| bp-vsir-bweb102-sd JobId 21509: Elapsed time=00:39:32, Transfer rate=4.110␣
↪M Bytes/second             |
| bp-vsir-bweb102-sd JobId 21509: Sending spooled attrs to the Director.␣
↪Despooling 2,219 bytes ...    |
| bp-vsir-bweb102-dir JobId 21509: Bacula Enterprise bp-vsir-bweb102-dir 16.0.
↪7 (11Jul23):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Core)
  JobId:                  21509
  Job:                    000_bp-o9-hap_vsphere.2023-11-15_13.45.06_49
  Backup Level:           Full (upgraded from Incremental)
  Client:                 "bp-vsir-bweb1004-fd" 16.0.7 (11Jul23) x86_64-
↪redhat-linux-gnu-bacula-enterprise,redhat,(Core)
  Fileset:                "000_bp-vsir-bweb1004-fd_bp-o9-hap_vsphere" 2023-10-
↪25 09:59:57
  Pool:                   "GED2D" (From Command input)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "LocalGED" (From Pool resource)
  Scheduled time:         15-Nov-2023 13:45:06
  Start time:             15-Nov-2023 13:45:08
  End time:               15-Nov-2023 14:24:42
  Elapsed time:           39 mins 34 secs
  Priority:               10
  FD Files Written:       9
  SD Files Written:       9
  FD Bytes Written:       10,081,812,251 (10.08 GB)
  SD Bytes Written:       9,749,179,049 (9.749 GB)
  Rate:                   4246.8 KB/s
  Software Compression:   3.3% 1.0:1
```

(continues on next page)

(continued from previous page)

```
  Comm Line Compression:  None
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               yes
  Volume name(s):         ged-3474|ged-3478|ged-3479|ged-3480|ged-3484|ged-
↪3485|ged-3486|ged-3487|ged-3488|ged-3489
  Volume Session Id:      44
  Volume Session Time:    1698051447
  Last Volume Bytes:      meta: 574,097 (574.0 KB) aligned: 562,429,952 (562.
↪4 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK |
+------------------------------------------------------------------------------
↪-----------------------+
+---------+-----------------------+---------------------+------+-------+-------
↪---+----------------+-----------+
| jobid  | name                  | starttime           | type | level |␣
↪jobfiles | jobbytes       | jobstatus |
+---------+-----------------------+---------------------+------+-------+-------
↪---+----------------+-----------+
| 21,509 | 000_bp-o9-hap_vsphere | 2023-11-15 13:45:08 | B    | F     |      ␣
↪ 9 | 10,081,812,251 | T         |
+---------+-----------------------+---------------------+------+-------+-------
↪---+----------------+-----------+
```

NVRAM and VMX Files

Since version 16.0.12, the NVRAM and VMX files are automatically included as part of the backup process. They are downloaded from the datastore and from the directory where the given Virtual Machine has its disks and other files stored.

During the restore process, the NVRAM and VMX files are automatically restored to the host where the FD and the vSphere Plugin are running. NVRAM file is also uploaded to the destination folder of the Virtual Machine that is being restored.

The NVRAM file contains BIOS information and can help in some restore cases. The VMX file contains configuration information of Virtual Machines and the purpose of having it is to allow the user to read this information if necessary.

Restore

This article describes restore with the use of the vSphere Plugin.

Restore to New VMware Guest

If you run a restore of a VM backup using the `where=/` restore option and select all files under the VM's directory, the vSphere Plugin will create a new VM with the same attributes (disks, controller, CPU type, …) on your ESXi host and restore the disks to this new VM. If you do not specify a new name for the restored VM, then the new VM's name will be the original VM's name with the restore job's jobid appended like: **originalName-123**.

```
* lsmark
/@vsphere/3/vsphere_generation.profile
/@vsphere/3/3.ovf
/@vsphere/3/1.bvmdk
/@vsphere/3/2.bvmdk
```

The SAN advanced transport mode is currently unsupported for restore. The vSphere Plugin will use NBD for VM restores.

The ESXi host and the datastore that will be used to restore your guest VM will be detected automatically. However, you can change the default destination by modifying the plugin restore options in the bconsole menu:

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.1.bsr
Where:           /tmp/regress/tmp/bacula-restores
...
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
...
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : vsphere: host=squeeze2
Plugin Restore Options
datastore:              *None*
restore_host:           *None*
new_hostname:           *None*
vsphere_server:         *None*
datastore_allow_overprovisioning: *None*               (yes)
datastore_minimum_space:    *None*
override_vm:            *None*              (no)
power_on:               *None*              (no)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: datastore (Datastore to use for restore)
     2: restore_host (ESXi host to use for restore)
     3: new_hostname (Restore guest VM to specified name)
```

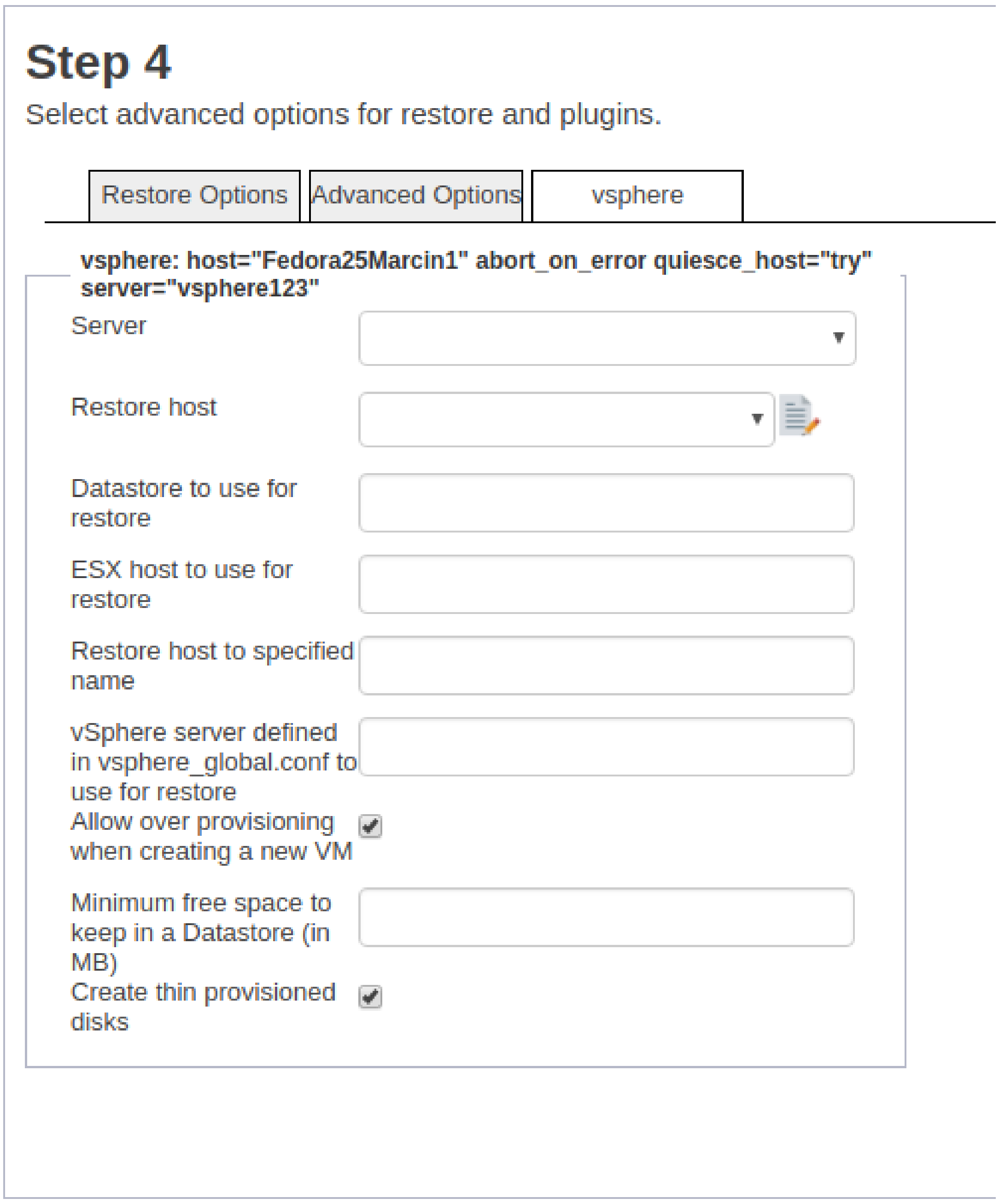
(continues on next page)

(continued from previous page)

```
     4: vsphere_server (vSphere server defined in vsphere_global.conf to use␣
↪for restore)
     5: datastore_allow_overprovisioning (Allow over provisioning when␣
↪creating a new VM)
     6: datastore_minimum_space (Minimum free space to keep in a Datastore␣
↪(in MB))
     7: override_vm (Restore to original VM, overriding it's disks (new_
↪hostname value will be ignored))
     8: power_on (Power on VM after restoration)
Select parameter to modify (1-6): 3
Please enter a value for new_hostname: test
Plugin Restore Options
datastore:           *None*
restore_host:        *None*
new_hostname:        test
vsphere_server:      *None*
datastore_allow_overprovisioning: *None*               (yes)
datastore_minimum_space: *None*
override_vm:         *None*                (no)
power_on:            *None*                (no)
Use above plugin configuration? (yes/mod/no): yes
```

The restore options may also be modified with using the BWeb restore interface.

Supported restore options are listed and detailed below:



Step 4

Select advanced options for restore and plugins.

vsphere: host="Fedora25Marcin1" abort_on_error quiesce_host="try" server="vsphere123"

Server	
Restore host	
Datastore to use for restore	
ESX host to use for restore	
Restore host to specified name	
vSphere server defined in vsphere_global.conf to use for restore	
Allow over provisioning when creating a new VM	✓
Minimum free space to keep in a Datastore (in MB)	
Create thin provisioned disks	✓

Fig. 24: Choose datastore, ESXi or new VM name at restore time

Option	Re-quire	De-fault	Info	Ex-am-ple
datastore	No		Datastore to use for restore	my-Data-store
re-store_host	No		ESX host to use for restore	host.mydc.com
new_hostnan	No		Name to apply to the new restored host (VM)	myVM-Re-stored1
vsphere_serv	No		vSphere server defined in vsphere_global.conf to use for restore	host3
datas-tore_allow_o	No	yes	Allow over provisioning when creating a new VM	no
datas-tore_minimu	No		Minimum free space to keep in a Datastore (in MB)	430899200
thin_provisio	No	yes	Create thin provisioned disks	no
over-ride_vm	No	no	Restore to original VM, overriding it's disks (new_hostname value will be ignored)	yes
power_on	No	no	Power on VM after restoration	yes
no_vmdk	No	no	Restore except vmdk contents	yes
force_san	No	no	Force the use of the SAN transfer	yes
no_network	No	no	Do not activate the network	yes
new_network	No		Name or Moref of a new network resource to apply, overriding the original one, during the restore.	network-7583
new_network	No		Name of the original network resource that will be overridden (by the new_network defined value), during the restore. If not specified, all original network resources will be overridden.	vlan-1

Note: You need to have at least one VM configured on your ESXi server to restore a VM from Bacula automatically. We plan to remove this limitation in a future version.

The vSphere Plugin can check the space available in the datastore during restore. It is possible to disallow *Over Provisioning* and reserve a minimum amount of space in the datastore. These two options can be set in the `vsphere_global.conf` file but can be overwritten from the restore menu.

Click here to see all available directives for the `vsphere_global.conf` file.

```
[vsphere]
    username = root
    password = xxxx
    server = 192.168.0.68
    url = https://192.168.0.68/sdk
    thumbprint = 34:F5:0F:10:82:59:EF:2D:DB:96:CC:5B:C4:66:33:83:DC:91:AF:01

    datastore_minimum_space = 64MB
    datastore_refresh_interval = 10
    datastore_allow_overprovisioning = false
```

Additional Information

Starting with Bacula Enterprise 12.2, the vSphere Plugin includes the vApp options in the OVF description of the virtual machine.

Starting with Bacula Enterprise 12.3, the vSphere Plugin can power on the virtual machine after a successful restore. Just select the option `power_on` in the bconsole plugin restore options.

Restore to Existing VMware Guest

Important: This option is available for Bacula 12.3 and above.

If you run a restore of a VM backup using the `where=/` restore option, select all files under the VM's directory and select the `override_vm` Plugin Option, the vSphere Plugin will look for the original guest in the ESXi host and restore those disks that are part of the backup. All other disks remain unchanged. If the Guest is still running, it will be powered off during restore. If the restore was successful, the guest will be powered on again.

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.1.bsr
Where:           /tmp/regress/tmp/bacula-restores
...
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
...
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : vsphere: host=squeeze2
Plugin Restore Options
datastore:              *None*
restore_host:           *None*
new_hostname:           *None*
vsphere_server:         *None*
datastore_allow_overprovisioning: *None*               (yes)
datastore_minimum_space:    *None*
override_vm:            *None*              (no)
power_on:               *None*              (no)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: datastore (Datastore to use for restore)
     2: restore_host (ESXi host to use for restore)
     3: new_hostname (Restore guest VM to specified name)
     4: vsphere_server (vSphere server defined in vsphere_global.conf to use␣
↪for restore)
     5: datastore_allow_overprovisioning (Allow over provisioning when␣
↪creating a new VM)
     6: datastore_minimum_space (Minimum free space to keep in a Datastore␣
↪(in MB))
```

(continues on next page)

(continued from previous page)

```
     7: override_vm (Restore to original VM, overriding it's disks (new_
↪hostname value will be ignored))
     8: power_on (Power on VM after restoration)
Select parameter to modify (1-6): 7
Please enter a value for override_vm: yes
Plugin Restore Options
datastore:            *None*
restore_host:         *None*
new_hostname:         *None*
vsphere_server:       *None*
datastore_allow_overprovisioning: *None*               (yes)
datastore_minimum_space: *None*
override_vm:          yes
power_on:             *None*              (no)
Use above plugin configuration? (yes/mod/no): yes
```

The restore options may also be modified with using the BWeb restore interface.

Restore to Local Disk with vSphere Plugin

Bacula Enterprise allows restoring any file (bvmdk, ovf, . . .) to your File Daemon's local disks. Then, you may mount the image locally using the VMware `vmware-mount` tool or `qemu-nbd` and perform file level restores.

By using `where=/path/to/dir` in the restore options, the Plugin will automatically restore selected files to this location on your File Daemon's local disk.

```
% qemu-img convert -O vmdk /tmp/0.bvmdk /tmp/0.vmdk
```

It is also possible to copy the raw image to any device or to mount it and restore files directly.

```
 # modprobe nbd max_part=16
 # qemu-nbd -c /dev/nbd0 /tmp/0.bvmdk
 # partprobe /dev/nbd0
 # fdisk -l /dev/nbd0

Disk /dev/nbd0: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders, total 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000a7154

     Device Boot      Start         End      Blocks   Id  System
/dev/nbd0p1   *          63     1686824      843381   83  Linux
/dev/nbd0p2         1686825     4192964     1253070    5  Extended
/dev/nbd0p5         1686888     1959929      136521   82  Linux swap
/dev/nbd0p6         1959993     4192964     1116486   83  Linux

 # mount /dev/nbd0p1 /mnt/image
 # ls /mnt/image
bin   cdrom  etc      initrd.img  lost+found  mnt    proc   sbin
```

(continues on next page)

(continued from previous page)

```
tmp   var    boot     dev          home         lib    media  opt
sys   usr    vmlinuz  srv          selinux      root

 # umount /mnt/image
 # qemu-nbd -d /dev/nbd0
```

Restore Without the vSphere Plugin

If you try to restore to a File Daemon where the Bacula Enterprise vSphere Plugin is not installed, you will have to convert the bvmdk files to raw files using the vddk tool in command line:

```
% vddk -C -i 0.bvmdk -o 0.raw
```

The bvmdk format is used internally by the vSphere Plugin to ensure data integrity and handle Changed Block Tracking (CBT) sparse information efficiently. It may happen that Bacula reports a false error during the restore about the file size.

Quiescing Guests

To properly quiesce the guest machines, VMware Tools **must** be installed and up to date in your Linux/Windows virtual machines.

The `quiesce_host=try/yes/no` Plugin command line option allows control of how the vSphere Plugin should try to quiesce guest VMs before a snapshot. The default value is `try`. In this mode, the plugin will try to quiesce the guest when creating the snapshot. If the quiesce operation fails, the plugin will make a second attempt to create the snapshot without quiescing the guest. The first attempt will be reported in the job log as an error and the job termination status will terminate “with warnings”.

```
Fileset {
  Name = guestvm
  Include {
    # Will try to quiesce the VM and skip the VM
    # from the backup if pre/post scripts are
    # returning an error
    Plugin = "vsphere: host=guestvm quiesce_host=yes"
  }
}
```

More information about the exact error message can be found in the vSphere console log.

```
Warning message from ESXi: the guest OS has reported an error during
quiescing. Error code was: 2 the error message was: custom quiesce script
failed.
```

Or

```
An error occurred while saving the snapshot: Failed to quiesce the virtual␣
↪machine.
```

Linux

By creating a special script located in `/usr/sbin/pre-freeze-script`, you will be able to quiesce your system automatically when vSphere creates a Snapshot.

The vSphere Plugin will also try to execute `/usr/sbin/post-thaw-script` script if present on the guest VM.

Windows VSS notes

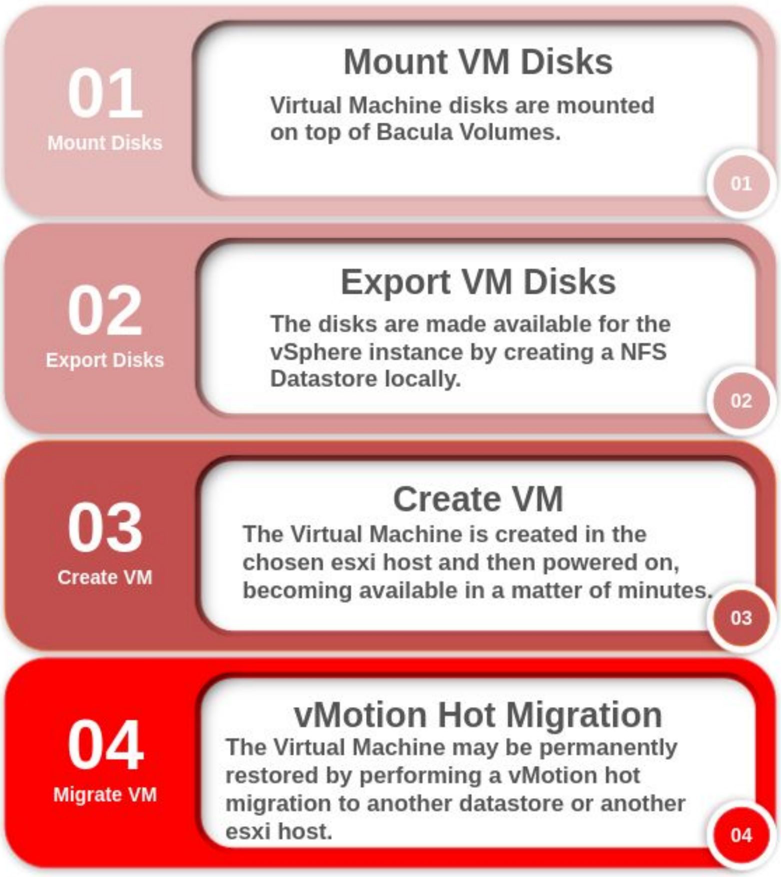
The plugin enhances Windows protection by performing VSS-based snapshots to quiesce VSS-enabled applications before backup.

VSS Pre-freeze and Post-thaw Scripts

VMware Tools first looks in `C:/Program Files/VMware/VMware Tools/backupScripts.d` for scripts in alphabetic order, calling them all with freeze argument, and afterward in reverse alphabetic order, calling them with `thaw` argument (or `freezeFail` if quiescing failed).

VM Instant Recovery

Starting With Bacula Enterprise 12.6, it is now possible to recover a vSphere Virtual Machine in a matter of minutes by running it directly from a Bacula Volume.


01
Mount Disks
Mount VM Disks
Virtual Machine disks are mounted on top of Bacula Volumes.
01
02
Export Disks
Export VM Disks
The disks are made available for the vSphere instance by creating a NFS Datastore locally.
02
03
Create VM
Create VM
The Virtual Machine is created in the chosen esxi host and then powered on, becoming available in a matter of minutes.
03
04
Migrate VM
vMotion Hot Migration
The Virtual Machine may be permanently restored by performing a vMotion hot migration to another datastore or another esxi host.
04

Any changes made to the VM disks are virtual and temporary. This means that the disks remain in a read-only state. The users may write to the VM disks without the fear of corrupting their backups. Once the Virtual Machine is started, it is then possible via VMotion to migrate the temporary Virtual Machine to a production datastore.

The feature is available inside the `mount-vm` script and follows this workflow:

1. User chooses a VM backup.
2. Script mounts the VM disks locally.
3. User chooses the ESXi host that will own the VM.
4. Script creates a temporary NFS Datastore locally.
5. Script creates and powers on the VM in this temporary NFS Datastore.
6. User chooses to keep the VM permanently or to discard it.

VM Instant Recovery Installation

Installation: Bacula Storage Daemon

In addition to installing the **bacula-enterprise-single-item-restore** package as described in installvspheresir, some extra actions are needed to enable the Instant Recovery feature:

1. Install and configure the vSphere Plugin

Refer to Bacula's *vSphere plugin documentation* for instructions on its installation and configuration.

2. Install NFS service

To install and configure the NFS server service automatically, run the `install-single-item-restore` script:

```
$ /opt/bacula/scripts/install-single-item-restore.sh install_ir
```

Installation: vSphere side

1. Test NFS connection on an ESXi host

Since Instant Recovery creates a temporary **NFS Datastore** on the machine containing the Bacula volumes, we need to make sure that the **ESXi hosts** can reach it through the NFS ports:

```
$ ssh <ESXi-host>
$ ping <nfs-datastore>
$ nc -z <nfs-datastore> 2049
```

For more information, refer to https://kb.vmware.com/s/article/1003967.

2. Set up vMotion

Migration with vMotion requires a vMotion network interface on each ESXi host where you plan to migrate VMs. The vMotion interface can be configured from either a vSphere client or a vSphere Web Client. The steps are:

- Navigate to the desired ESXi host
- Navigate to the host network settings
- Under network settings, click to **add networking**
- Select **VMkernel Network Adapter** as connection type
- Select either an existing switch or a new one
- Under port properties, set it to **allow vMotion traffic**

For more information and best practices please refer to https://kb.vmware.com/s/article/2054994.

Recovery Scenarios

This article aims at presenting recovery scenarios of VM Instant Recovery.

Temporary Recovery for Testing Purposes

In this scenario, we recover the VM called `yVM` to the ESXi host located at 192.168.0.26:

```
[root@localhost bin]# sudo -u bacula /opt/bacula/bin/mount-vm
Automatically Selected Catalog: BaculaCatalog
Automatically Selected Client: localhost-fd

Job list:
1: LinuxEtc.2020-08-31_07.08.04_04
2: vsphere_hbck-centos7.2020-08-27_06.39.21_03
3: vsphere_hbck-centos7.2020-08-31_08.01.58_03
4: vsphere_yVM.2020-09-22_08.03.07_08
Select a Job: 4
Selected vsphere_yVM.2020-09-22_08.03.07_08
Automatically Selected Virtual Machine: yVM (14)
Automatically Selected Disks: 0

Action list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Export guest virtual machine to vSphere instance (Instant Recovery)
4: Cleanup
Select an Action: 3
Selected Export guest virtual machine to vSphere instance (Instant Recovery)
I: Instant Recovery Mode
I: NFSv4 detected.
Select where you want to mount the Virtual Machine

ESXi Host list:
1: [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

2: [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select an ESXi Host: 1
Selected [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

[sudo] password for bacula:
Creating NFS Datastore...
OK: Registered this machine as a NFS Datastore (name=bir-14-6959, host=192.
↪168.0.26)
Creating Virtual Machine...
OK: The Virtual Machine is now available (name=yVM-6959, host:192.168.0.26,␣
↪datastore:bir-14-6959)
```

(continues on next page)

(continued from previous page)

```
Action list:
1: Migrate Virtual Machine to a permanent Datastore
2: Cleanup
Select an Action: 2
Selected Cleanup
end
```

Recovery and Migration in the Same ESXi Host

In this scenario, we recover the VM called `yVM` to the ESXi host located at 192.168.0.26 and migrate it permanently to the datastore "datastore 1 (1)".

```
[root@localhost bin]# sudo -u bacula /opt/bacula/bin/mount-vm
Automatically Selected Catalog: BaculaCatalog
Automatically Selected Client: localhost-fd

Job list:
1: LinuxEtc.2020-08-31_07.08.04_04
2: vsphere_hbck-centos7.2020-08-27_06.39.21_03
3: vsphere_hbck-centos7.2020-08-31_08.01.58_03
4: vsphere_yVM.2020-09-22_08.03.07_08
Select a Job: 4
Selected vsphere_yVM.2020-09-22_08.03.07_08
Automatically Selected Virtual Machine: yVM (14)
Automatically Selected Disks: 0

Action list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Export guest virtual machine to vSphere instance (Instant Recovery)
4: Cleanup
Select an Action: 3
Selected Export guest virtual machine to vSphere instance (Instant Recovery)
I: Instant Recovery Mode
I: NFSv4 detected.
Select where you want to mount the Virtual Machine

ESXi Host list:
1: [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

2: [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select an ESXi Host: 1
Selected [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

[sudo] password for bacula:
Creating NFS Datastore...
```

(continues on next page)

(continued from previous page)

```
OK: Registered this machine as a NFS Datastore (name=bir-14-7454, host=192.
↪168.0.26)
Creating Virtual Machine...
OK: The Virtual Machine is now available (name=yVM-7454, host:192.168.0.26,␣
↪datastore:bir-14-7454)

Action list:
1: Migrate Virtual Machine to a permanent Datastore
2: Cleanup
Select an Action: 1
Selected Migrate Virtual Machine to a permanent Datastore
Select which host you want to migrate the Virtual Machine to:

ESXi Host list:
1: [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

2: [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select an ESXi Host: 1
Selected [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select which datastore you want to migrate the Virtual Machine to:
Automatically Selected Datastore: datastore1 (1)
Migrating Virtual Machine to datastore1 (1) (192.168.0.26). This may take␣
↪some time...
OK: The Virtual Machine was migrated.
I: Press enter to finish and cleanup the session

I: End of session
```

Recovery and Migration to Another ESXi Host

In this scenario, we recover the VM called yVM to the ESXi host located at 192.168.0.26 and migrate it permanently to a datastore in another ESXi host (192.168.0.8).

```
[root@localhost bin]# sudo -u bacula /opt/bacula/bin/mount-vm
Automatically Selected Catalog: BaculaCatalog
Automatically Selected Client: localhost-fd

Job list:
1: LinuxEtc.2020-08-31_07.08.04_04
2: vsphere_hbck-centos7.2020-08-27_06.39.21_03
3: vsphere_hbck-centos7.2020-08-31_08.01.58_03
4: vsphere_yVM.2020-09-22_08.03.07_08
Select a Job: 4
Selected vsphere_yVM.2020-09-22_08.03.07_08
Automatically Selected Virtual Machine: yVM (14)
```

(continues on next page)

(continued from previous page)

```
Automatically Selected Disks: 0

Action list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Export guest virtual machine to vSphere instance (Instant Recovery)
4: Cleanup
Select an Action: 3
Selected Export guest virtual machine to vSphere instance (Instant Recovery)
I: Instant Recovery Mode
I: NFSv4 detected.
Select where you want to mount the Virtual Machine

ESXi Host list:
1: [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

2: [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select an ESXi Host: 2
Selected [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Creating NFS Datastore...
OK: Registered this machine as a NFS Datastore (name=bir-14-8028, host=192.
↪168.0.8)
Creating Virtual Machine...
OK: The Virtual Machine is now available (name=yVM-8028, host:192.168.0.8,␣
↪datastore:bir-14-8028)

Action list:
1: Migrate Virtual Machine to a permanent Datastore
2: Cleanup
Select an Action: 1
Selected Migrate Virtual Machine to a permanent Datastore
Select which host you want to migrate the Virtual Machine to:

ESXi Host list:
1: [192.168.0.26]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

2: [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select an ESXi Host: 2
Selected [192.168.0.8]
    (administrator@vsphere.local, https://192.168.0.15/sdk)

Select which datastore you want to migrate the Virtual Machine to:
Automatically Selected Datastore: datastore1
Migrating Virtual Machine to datastore1 (192.168.0.8). This may take some␣
```

(continues on next page)

(continued from previous page)

```
↪time...
OK: The Virtual Machine was migrated.
I: Press enter to finish and cleanup the session

I: End of session
```

Instant Recovery with Network Card Disconnected

Starting with version 14.0.1, it is possible to create the temporary VM with network cards not connected. To disconnect network cards at boot, use the `-N` option in the `mount-vm` command line.

```
[root@localhost bin]# sudo -u bacula /opt/bacula/bin/mount-vm -N
Automatically Selected Catalog: BaculaCatalog
Automatically Selected Client: localhost-fd

Job list:
1: LinuxEtc.2020-08-31_07.08.04_04
...
```

Manually Cleaning an Instant Recovery Session

It is possible to manually clean an Instant Recovery session by purging its associated datastore:

```
bacula@storage# /opt/bacula/bin/vsphere-ctl purge_ds bir-41-4018 --server␣
↪vcenter_70
I: Successfully purged Datastore bir-41-4018
```

VM Instant Recovery Limitations

- The VMware Single Item Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backend due to internal MySQL limitations with indexes on TEXT columns. For VMware and Exchange Single Item Restore there should not be too much impact on performances (the backup structure is usually quite small) but we advise using the PostgreSQL backend for the best experience.
- The VMware Single Item Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was created with a large number of concurrent jobs.
- The Storage Daemon where the VMware Single Item Restore is installed should be have a CPU with the VT-x/EPT extensions. If these extensions are not available, the performance will be degraded. (From 20s to 10mins in our lab).
- The VMware Single Item Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc.). Tape devices are not supported.
- To perform VM Instant Recovery from a Copy/Migration job make sure destination Pool has `Maximum Volume Jobs` set to 1. Note that when you use `MaximumVolumeJobs = 1` in the Pool resource, you must use `MaximumConcurrentJobs = 1` in the Device resource(s).

- The Instant Recovery hot migration only works with the VMware vMotion technology.
- All volumes needed for the VMware Single Item Restore must reside on the single Storage Daemon instance where the SIR session is started. A storage group policy can conflict with this requirement.
- To avoid heavy network traffic and prevent jobs from failing, do not run the vSphere plugin along the FD running on the SD with a storage group.
- Client side PKI Encryption is currently not compatible with vSphere IR features. Use the Volume Encryption feature if needed.

VMware Single Item Restore

This section presents how to use the VMware Single Item Restore feature with Bacula Enterprise and the vSphere Plugin.

Single Item Restore Features

The **Bacula Enterprise** VMware Single Item Restore provides the following main features:

- Console interface
- BWeb Management Suite interface
- Support for Full/Differential/Incremental jobs
- Support for Windows NTFS
- Support for Linux (ext3, ext4, btrfs, lvm, xfs)
- Support for ESX 5.x, 6.x, 7.0 and 8.0.

Single Item Restore Scope

This document will present solutions for **Bacula Enterprise** 8.4 and later, which are not applicable to prior versions. The VMware Single Item Restore has been tested and is supported on RHEL 7.x, 8.x, 9.x. SELinux is currently not supported. The vSphere Plugin might not be supported on all platforms where VMware Single Item Restore is supported.

Warning: RHEL decided to stop supporting Windows NTFS devices starting with RHEL version 7 and they have removed the “ntfs-ng” package from the official RHEL repository. The “ntfs-ng” package is a required dependency of the “libguestfs” package and will need to be installed separately from a repository such as EPEL (see installvspheresir)

Single Item Restore Installation

This article describes how to install Single Item Restore for the vSphere Plugin.

SIR Installation with BIM

The SIR features are installed along the SD. In consequence, to install SIR for the vSphere Plugin with BIM, run BIM with the *-t SD* option and choose to install SIR. If the SD is already installed on you system, Bacula configuration and SD packages will not be modified.

SIR Installation with Package Manager

Packages for the VMware Single Item Restore plugin are available for supported platforms. Please contact Bacula Systems to get them.

Download the plugin package to your **Storage Daemon** server and then install using the package manager like so:

```
rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the VMware Single Item Restore Plugin and will install dependencies. On RHEL, it will be needed to install `perl-JSON` package from **rpmforge** and the `libguestfs-winsupport` package.

Note: On RHEL 8.x or 9.x, it is necessary to install a custom version of the libguestfs packages from our repository to support NTFS devices. Those should not be updated with a newer version from official repositories. The YUM package manager has plugins to prevent package updates, try **yum-plugin-versionlock** or **yum-plugin-priorities**. Additionally, the `ntfs-3g` package from the EPEL repository is needed for NTFS support. To install the EPEL respository, please follow the official instructions on the EPEL website to install the "epel-release" package here: https://fedoraproject.org/wiki/EPEL

Note: The *DAG* repo below must be set up in order to install *libguestfs* needed for SIR.

Note: On RHEL 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/DAG/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0
```

(continues on next page)

(continued from previous page)

```
# cat /etc/yum.repos.d/baculasystems.repo
[single_file_restore_vmware]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/<xxx>/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0


Note: This last repository is required on RHEL7:

[Bacula-Enterprise-DAG-Guestfish]
name = Bacula Enterprise - DAG for Guestfish
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64/guestfish/
enabled = 1
protect = 0
gpgcheck = 0
```

```
yum install bacula-enterprise-single-item-restore perl-JSON
```

If BWeb Management Suite is used:

```
service bweb restart
```

Samba SMB Shares

The **Bacula Enterprise** VMware Single Item Restore plugin can automatically set up Samba SMB shares from the console program or the BWeb Management Suite.

To enable Samba SMB network shares, installing and configuring the “samba” package is mandatory. To configure the `/etc/samba/smb.conf` file correctly, you need to run `install-single-item-restore.sh` script.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
include = /etc/samba/conf.d/all
```

At this point, it is possible to modify `/etc/samba/smb.conf` to add your own configuration directives.

Network share descriptions will be stored in the directory `/etc/samba/conf.d`. It is possible to create and customize the template used by Bacula to generate configuration files.

```
root@storage# cat /etc/samba/conf.d/custom.tpl
[__share__]
   path = __path__
   follow symlinks = yes
```

(continues on next page)

(continued from previous page)

```
   wide links = yes
   writable = yes
```

BWeb Integration with Single Item Recovery

Installation

To use the BWeb Management Suite graphical GUI with the VMware Single File Restore option it is currently necessary to install and configure BWeb Management Suite on the Storage Daemon where vSphere jobs are stored. If the Director is not on the same machine than the Storage Daemon, remember that the administrator needs to connect to the right BWeb Management Suite instance to use specific VMware Single Item Restore screens.

After the installation of the Single Item Restore package, the BWeb service "bweb" must be restarted to take in account the `bacula` unix user group modification.

```
service bweb restart
```

HTTP Server Extra Configuration

To let the end user access the virtual machine files, it is necessary to set up the `lighttpd` daemon correctly. In this case, we advise using both SSL and user authentication. Example `/opt/bweb/etc/httpd.conf`:

```
################################################################
# To enable SSL
# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout server.pem -out␣
↪server.pem
# chown bacula:bacula server.pem
# chmod 400 server.pem

ssl.engine = "enable"
ssl.pemfile = "/opt/bweb/etc/server.pem"


################################################################
# To enable Auth login http://redmine.lighttpd.net/projects/1/wiki/Docs_
↪ModAuth
# htpasswd -c /opt/bweb/etc/htpasswd.bweb

auth.backend = "htpasswd"
auth.backend.htpasswd.userfile =  "/opt/bweb/etc/htpasswd.bweb"
auth.require = ( "/" =>
        (
        "method" => "basic",
        "realm" => "Password protected area",
        "require" => "valid-user"
        )
   )
```

Single Item Restore Configuration

Storage Daemon Configuration

On the **Storage Daemon** host server, the `bconsole` program should be configured properly to let the "bacula" user connect to the Director with `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 8.4.0 (11 August 2015)
Enter a period to cancel a command.
* version
mydir-dir Version: 8.4.0 (11 August 2015) x86_64-redhat-linux-gnu
* quit
```

The package contains a script to test the connection with the Director and to test if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage#  /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you may not see any files or files may have strange permissions, ownerships and sizes and will inaccessible even to the root user.

Single Item Restore: Restore Scenarios

"bacula" Account on RHEL

All commands in this document use the "bacula" unix account to run.

On RHEL, the Unix "bacula" account is locked by default. It means that it's not possible by default to execute a command such as `su - bacula`.

It is possible to unlock the "bacula" account, or to use `sudo -u bacula` to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

It can be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using:

```
root@storage# sudo -u bacula /bin/bash
```

or unlock the "bacula" unix account and use `su -` with a command such as:

```
root@storage# chsh -s /bin/bash bacula
root@storage# su - bacula
bacula@storage# whoami
bacula
```

Fuse FileSystem

If a restore session is not properly cleaned up, the `mount` command may show some directories mounted with the Bacula Fuse FileSystem.

```
baculafs on /opt/bacula/working/cat-ro type fuse.baculafs (ro,user=bacula)
backend0 on /opt/bacula/working/26-0 type fuse.backend0 (ro,user=bacula)
/dev/fuse on /opt/bacula/working/26 type fuse (rw,nosuid,nodev,user=bacula)
```

It is possible to unmount directories with the `fusermount -u` command.

```
bacula@storage# fusermount -z -u /opt/bacula/working/26
bacula@storage# fusermount -z -u /opt/bacula/working/26-0
bacula@storage# fusermount -z -u /opt/bacula/working/cat-ro
```

Cache Directory

To speed up future VMware Single Item restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
bacula@storage# ls /opt/bacula/working/mount-cache
1-5-0.bmp  1-5-2.bmp    MyCatalog-2.idx  MyCatalog-5.idx  MyCatalog-8.idx
1-5-1.bmp  1-5.profile  MyCatalog-4.idx  MyCatalog-6.idx  MyCatalog-9.idx
```

It is possible to remove files in the cache after some time - they will be re-generated if needed.

With Text Console Interface

The VMware Single Item Restore plugin provides a simple console program that provides access to files inside VMs.

```
bacula@storage# /opt/bacula/bin/mount-vmware
Automatically Selected Catalog: MyCatalog

Client list:
1: 127.0.0.1-fd
2: win2008-fd
3: rhel7-fd
Select a Client: 1
```

(continues on next page)

(continued from previous page)

```
Selected Client: 127.0.0.1-fd

Job list:
1: NightlySave.2015-09-01_10.49.18_39
2: pluginTest.2015-09-01_10.40.20_04
3: pluginTest.2015-09-01_10.46.19_08
Select a Job: 2
Selected pluginTest.2015-09-01_10.40.20_04

Virtual Machine:
1: squeeze2 (5)
2: win2008 (6)
3: rhel7 (7)
Select a Virtual Machine: 1
Selected squeeze2 (5)

Actions list:
1: Mount guest filesystem locally
2: Export guest filesystem through SMB
3: Cleanup
Select a Actions: 1
Selected Mount guest filesystem locally

I: Files are available under /opt/bacula/working/vmware/5
I: Press enter to finish and cleanup the session
```

In this step, the virtual machine filesystem is mounted locally (in the example above, files are available under `/opt/bacula/working/vmware/5`. It is possible to browse directories and copy files (with `cp`, `scp`, `ftp`) as with a standard filesystem from another terminal session with the Unix "root" and "bacula" accounts. If you need to use another Unix account to operate on files, use the "-o allow_other" option when starting the `mount-vmware` script.

```
bacula@storage# ls /opt/bacula/working/vmware/5
bin   dev  home        lib         media  opt   root  selinux  sys  usr ␣
↪vmlinuz
boot  etc  initrd.img  lost+found  mnt    proc  sbin  srv      tmp  var
```

To clean up the session, just press "Enter" in the terminal session where the `mount-vmware` script was started.

It is possible to limit the Job list with the following command line options:

- `-s=<days>` Limit the job list to the last *days*
- `-l=<number>` Limit the job list to the last *number* entries
- `-f=<filter>` Specify an advanced filter based on the Job name, the Fileset name or the JobId

```
# Limit the job output to the last 100 jobs
bacula@storage# /opt/bacula/bin/mount-vmware -l 100

# Limit the job output to the last 30 days
bacula@storage# /opt/bacula/bin/mount-vmware -s 30

```

(continues on next page)

(continued from previous page)

```
# Limit the job output to jobs that start with ``MyVMware''
bacula@storage# /opt/bacula/bin/mount-vmware -f 'jobname=MyVMware*'

# BAD USAGE for the filter option, it will search for a job named ``MyVMware''
bacula@storage# /opt/bacula/bin/mount-vmware -f 'jobname=MyVMware'

# Limit the job output to jobs that start with ``MyVMware''
# and that use the Fileset Test1
bacula@storage# /opt/bacula/bin/mount-vmware -f 'jobname=MyVMware*␣
↪fileset=Test1'

# Limit the job to the jobid XX
bacula@storage# /opt/bacula/bin/mount-vmware -f jobid=XX
```

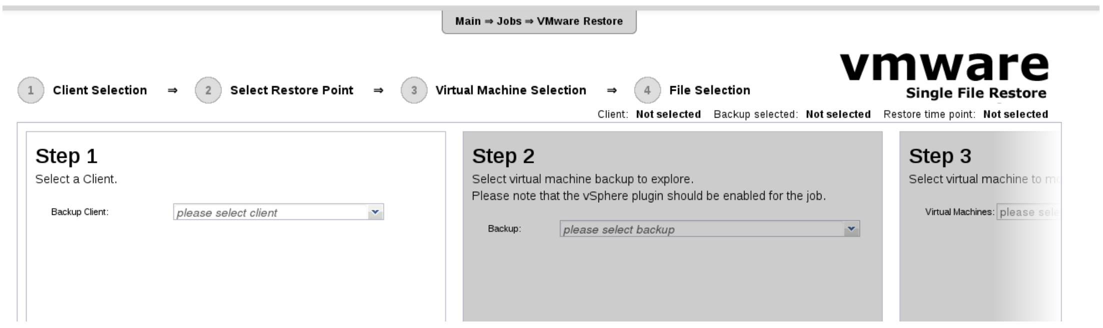
On some cases, the device detection doesn't work properly. It is possible to use the -m option to mount recognized disks in a simple way. The option is automatically set when only one disk is selected during the restore.

```
bacula@storage# /opt/bacula/bin/mount-vmware -m
```

With BWeb Management Suite Interface

The VMware Single Item Restore option in BWeb Management Suite is a wizard that provides easy restoration of files from a VMware guest. The integration of BWeb within the Single Item Restore is necessary following the below steps.

The first step is to select the Client where the vSphere backup job was done.


Main ⇒ Jobs ⇒ VMware Restore
1 Client Selection ⇒ 2 Select Restore Point ⇒ 3 Virtual Machine Selection ⇒ 4 File Selection
vmware
Single File Restore

Client: Not selected Backup selected: Not selected Restore time point: Not selected

Step 1
Select a Client.
Backup Client: please select client

Step 2
Select virtual machine backup to explore.
Please note that the vSphere plugin should be enabled for the job.
Backup: please select backup

Step 3
Select virtual machine to m
Virtual Machines: please sel

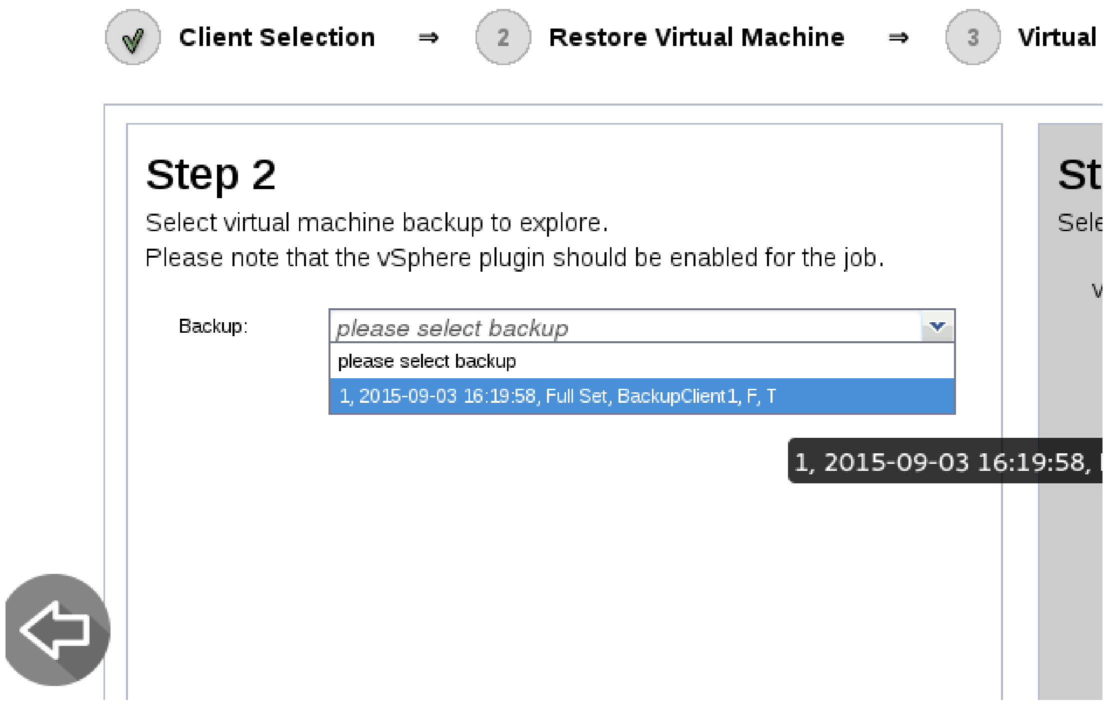
Fig. 25: Client Selection

Once the Client is selected, the administrator needs to select the Job (a Restore Point) to restore.

If the selected Job is a valid vSphere job, the third step will display a list of all virtual machines included in the Fileset.

At this point, Bacula needs to build a virtual image of the selected virtual machine. A couple of small files need to be restored from each Job that makes up the selected *Restore Point*. Once done, Bacula needs to mount the disk of the selected virtual machine on the system. These steps are usually quite fast, but the time depends a lot on the configuration used. Indexes are created and kept during this phase to speed up any further restore requests.

Available with Bacula Enterprise 8.6



Step 2

Select virtual machine backup to explore.
Please note that the vSphere plugin should be enabled for the job.

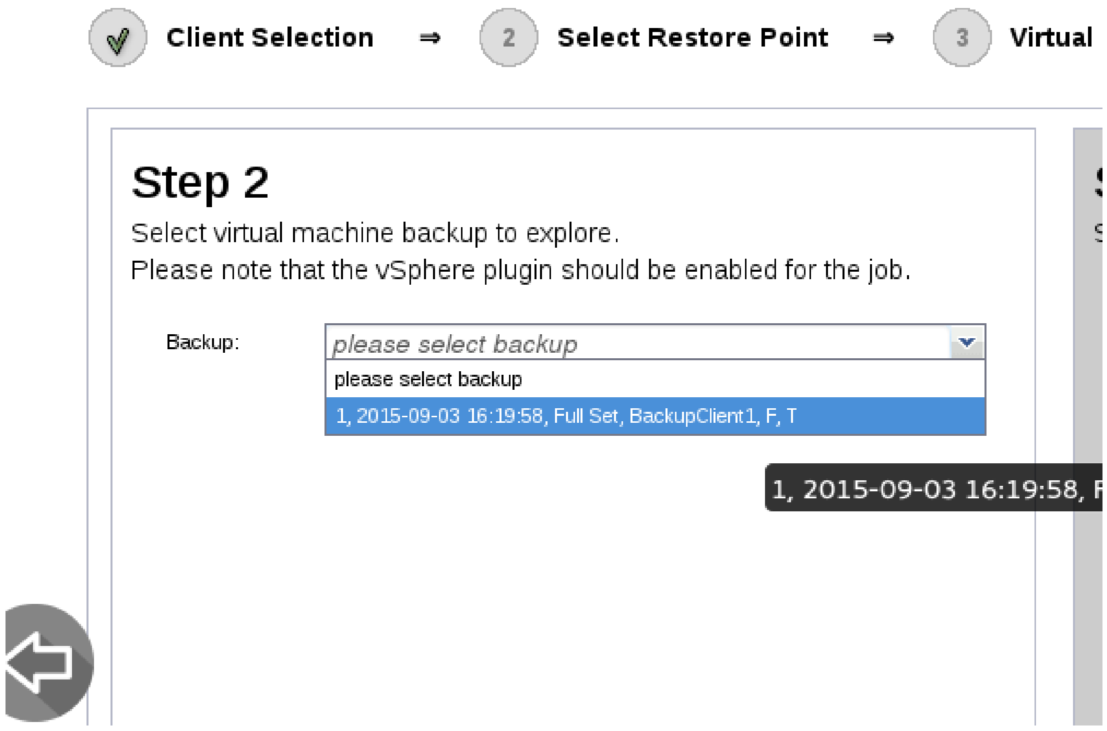
Backup: please select backup

please select backup

1, 2015-09-03 16:19:58, Full Set, BackupClient1, F, T

1, 2015-09-03 16:19:58,

Fig. 26: Restore Point Selection



Step 2

Select virtual machine backup to explore.
Please note that the vSphere plugin should be enabled for the job.

Backup: please select backup

please select backup

1, 2015-09-03 16:19:58, Full Set, BackupClient1, F, T

1, 2015-09-03 16:19:58,

Fig. 27: Virtual Machine Selection

To create the index during the backup phase, the Fileset plugin option `index` can be used.

```
Plugin = "vsphere: host=myhost index"
```

Once mounted, the selected virtual machine files will be displayed in a file browser where it is possible to select files or directories to restore (figure below). The administrator can then choose to generate a ZIP or a TAR archive. The archive will be generated automatically and will be stored in `/opt/bacula/working`. A secure HTTP download link will be generated, and the administrator can provide this link to the end user.

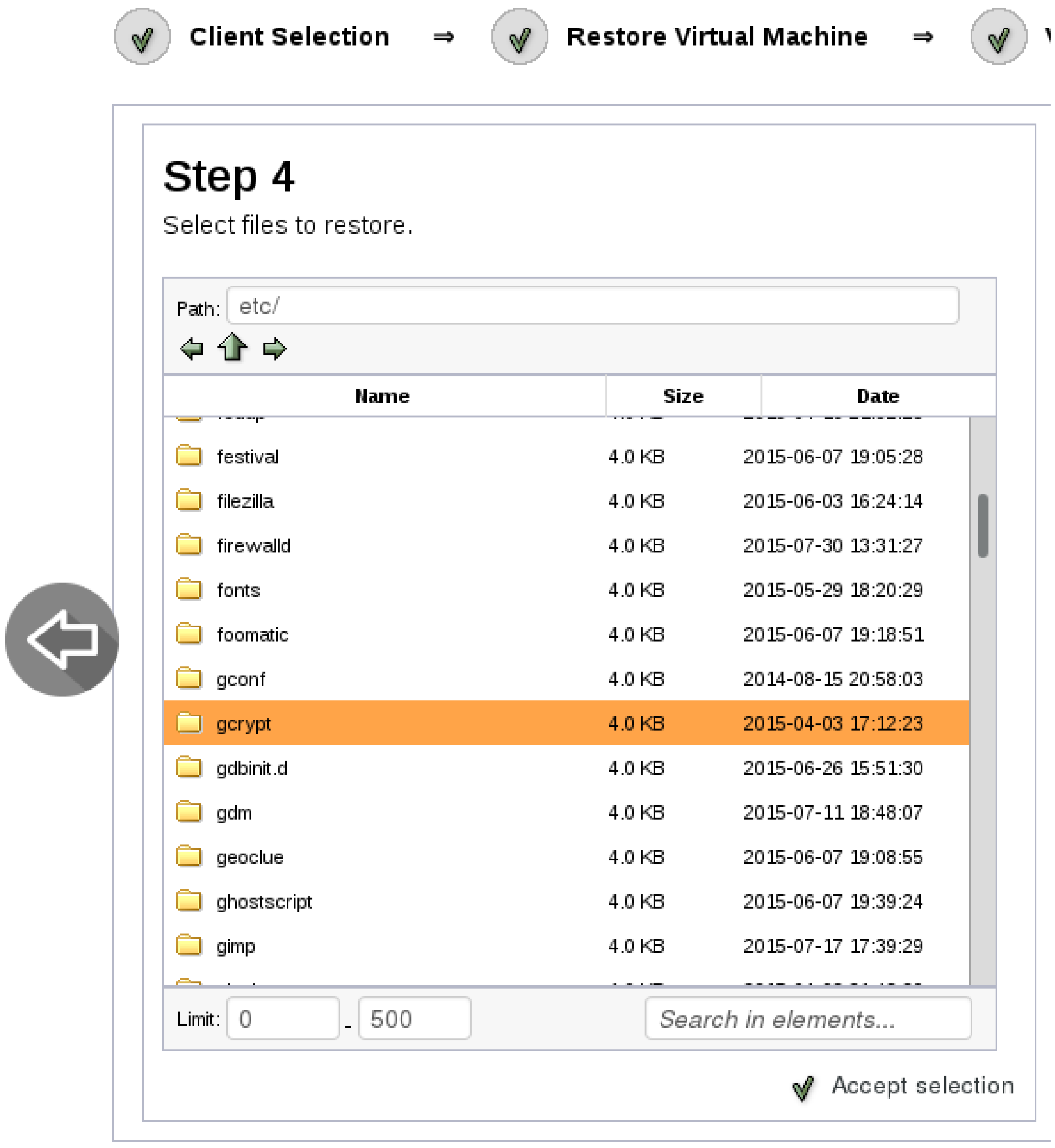
If BWeb Management Suite is configured to use HTTP Authentication, it is necessary to configure `lighttpd` properly to allow "anonymous" users to download their files. (See config-httpd)

For each selection, the administrator can choose how to retrieve the files directly, compressed as a tar file or a zip file.

Once the restore has taken place it is important to terminate the

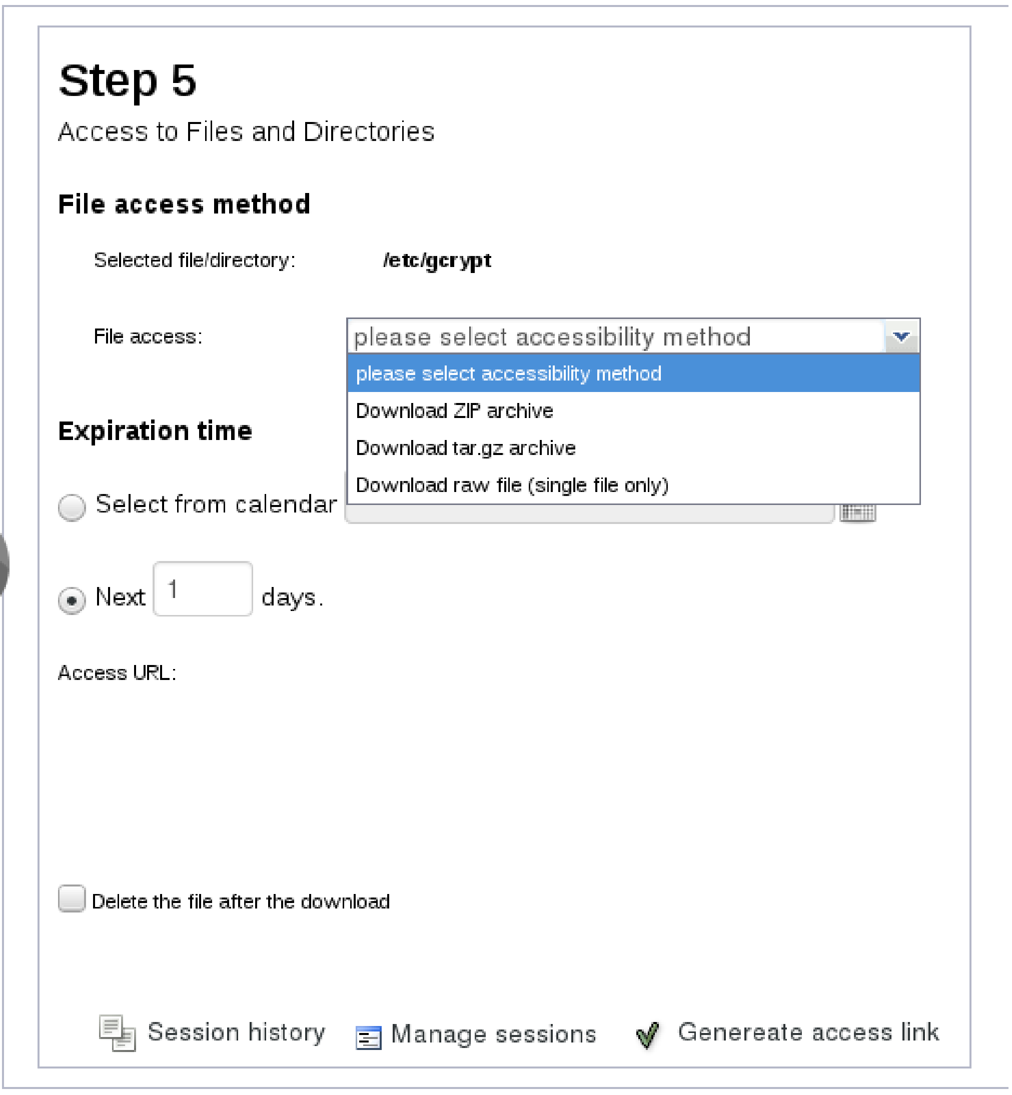
Single Item Restore Limitations

- The VMware Single Item Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backend due to internal MySQL limitations with indexes on TEXT colums. For VMware and Exchange Single Item Restore there should not be too much impact on performances (the backup structure is usually quite small) but we advise using the PostgreSQL backend for the best experience.
- The VMware Single Item Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was created with a large number of concurrent jobs.
- The Storage Daemon where the VMware Single Item Restore is installed should be have a CPU with the VT-x/EPT extensions. If these extensions are not available, the performance will be degraded. (From 20s to 10 min in our lab).
- RHEL 7 and later does not support mounting NTFS disks with the libguestfs provided with their system. To mount Microsoft NTFS disks on RHEL 7 or later, it is required to install a patched version of the libguestfs packages. Please see notes in installvspheresir of this document for more information.
- The VMware Single Item Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.
- To perform Single Item Restore from a Copy/Migration job make sure destination Pool has `Maximum Volume Jobs` set to 1. Note that when you use `MaximumVolumeJobs = 1` in the Pool resource, you must use `MaximumConcurrentJobs = 1` in the Device resource(s).
- All volumes needed for the VMware Single Item Restore must reside on the single Storage Daemon instance where the SIR session is started. A storage group policy can conflict with this requirement.
- To avoid heavy network traffic and prevent jobs from failing, do not run the vSphere plugin along the FD running on the SD with a storage group.
- Volume encryption is compatible with vSphere SIR features. PKI Encryption is not compatible.



Name	Size	Date
festival	4.0 KB	2015-06-07 19:05:28
filezilla	4.0 KB	2015-06-03 16:24:14
firewalld	4.0 KB	2015-07-30 13:31:27
fonts	4.0 KB	2015-05-29 18:20:29
foomatic	4.0 KB	2015-06-07 19:18:51
gconf	4.0 KB	2014-08-15 20:58:03
gcrypt	4.0 KB	2015-04-03 17:12:23
gdbinit.d	4.0 KB	2015-06-26 15:51:30
gdm	4.0 KB	2015-07-11 18:48:07
geoclue	4.0 KB	2015-06-07 19:08:55
ghostscript	4.0 KB	2015-06-07 19:39:24
gimp	4.0 KB	2015-07-17 17:39:29

Fig. 28: File Selection


Step 5

Access to Files and Directories

File access method

Selected file/directory: **/etc/gcrypt**

File access: please select accessibility method

please select accessibility method
Download ZIP archive
Download tar.gz archive
Download raw file (single file only)

Expiration time

● Next 1 days.

Access URL:

☐ Delete the file after the download

Fig. 29: Setup File Access

Single Item Restore Troubleshooting

Collecting Traces Automatically

The `install-single-item-restore.sh` script can collect traces automatically when a `mount-vmware` session is running.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh support
```

List Host Operations

Attention: New in version 16.0.12

It is possible to list Vmware hosts and apply different filters:

```
// List all hosts
.query client=my-fd plugin="vsphere:" parameter=host

// List hosts by tag name
.query client=my-fd plugin="vsphere: filter=tag filter_value=tag1"␣
↪parameter=host

// List hosts by tag id
.query client=my-fd plugin="vsphere: filter=tag_id filter_value=id1"␣
↪parameter=host

// List hosts by resource pool
.query client=my-fd plugin="vsphere: filter=pool filter_value=myPoolMoref1"␣
↪parameter=host

// List hosts by datastore
.query client=my-fd plugin="vsphere: filter=datastore filter_
↪value=myDatastore1" parameter=host

// List hosts by host ESXi
.query client=my-fd plugin="vsphere: filter=host filter_value=myHost1"␣
↪parameter=host

// List hosts by datacenter
.query client=my-fd plugin="vsphere: filter=datacenter filter_value=dc1"␣
↪parameter=host
```

Limitations

- vSphere 5.5 and older versions are supported only with the 10.0.6 Bacula FD or older.
- Backups created using the vSphere Plugin are not compatible with Virtual Full jobs. Do not attempt to combine these two backup strategies as you will not be able to properly restore vSphere Plugin jobs from Virtual Full backups.
- The vSphere Plugin uses the vStorage API to manipulate files and snapshots, this extension requires a valid, non-free VMware license.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

Troubleshooting: Backup

The following article presents ways to troubleshoot issues regarding backup.

Cleanup Old Snapshots

Using the vSphere Plugin version 6.6.3 and later, if the VMware system contains snapshots that were not deleted automatically by the vSphere Plugin, the following commands are useful to clean up your system.

```
Cleanup old snapshots and previous failed generation
  vsphere-ctl clean-snapshot --snapshot myhost

Cleanup old snapshots with a name starting with a string
  vsphere-ctl clean-snapshot --snapshot-base pluginTest myhost

Cleanup snapshots with all children (probably faster)
  vsphere-ctl clean-snapshot --snapshot --snapshot-delete-child myhost
```

When starting a new backup job, the vSphere Plugin will automatically check if the previous job had a problem and will delete the old snapshot if required.

Working Files

The vSphere Plugin creates special files in the working directory. These files are needed to use the Changed Block Tracking (CBT) VMware feature. To clean up the vSphere Plugin working directory, you can schedule the `vsphere-ctl` command as:

```
# /opt/bacula/bin/vsphere-ctl clean 30
```

It will remove files and directories after a period of 30 days. This period should correspond at the minimum to the Full interval period plus additional days for safety reasons. During the backup, if the vSphere Plugin is not able to find working files created during the last Backup, the vSphere Plugin will create the necessary directories and upgrade the backup job to a Full backup of all disks.

Troubleshooting: Restore

The following article presents ways to troubleshoot issues regarding restore.

Not Loading OVF Guest Description into vSphere or vCenter Server

Sometimes, Bacula is not able to load the OVF guest description into your vSphere or vCenter server. This is mainly due to some limitations of VMware, such as “you can’t deploy an OVF that contains references to a mounted CDROM”, etc. The vSphere Plugin implements workarounds for well-known issues, but the plugin doesn’t cover all of them. If you are facing this problem, you can use the `default_ovf` parameter in `vsphere_global.conf` file. Basically, you will need to configure the `default_ovf` parameter to refer to an existing simple OVF template. The restore process will use it automatically, and you will have to configure your VM afterward for properties such as CPU number, amount of RAM, etc.

```
[vsphere]
...
   default_ovf=/opt/bacula/etc/default.ovf
```

Possible Additional Tasks Required after Restore on Windows

On Windows systems, in some cases, after the actual restore process is finished, some additional tasks maybe required. For example, if the recovered system does not boot, the restored VM may need to be repaired using the repair options of the original Windows installation media. Also, for Active Directory servers, it may be necessary to follow Microsoft’s guidelines to get a consistent state of the AD databases and synchronize with other AD servers. If your setup includes dynamic disks, you must import them in the freshly restored system after the reboot. You can do that from the disk manager or using “diskpart” by selecting one of the dynamic disk and using the “import” command.

```
select disk <XX>
import
```

Amount of Space Returned by EXSi or vCenter Accuracy

The “uncommitted” amount of space returned by the EXSi or vCenter server is not always accurate, the refresh frequency can be changed using the method described in: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2008367

vSphere Plugin Logs

The vSphere Plugin uses many different technologies and third party libraries. The result is that traces are spread among different directories on the backup server. You will be able to consult the following files:

Table 25: vSphere plugin traces

File	Component	Note
/opt/bacula//*vsphere-ctl-*.log	vsphere-ctl	This file is produced by the Java vsphere-ctl program that sends commands to the ESXi/vCenter server.
/opt/bacula/working/vspł	vddk	This file is produced by the C++ vddk program that read/write to the VMDK.
/opt/bacula/working/*.tra	bacula-fd	This file is produced Bacula File Daemon when activating the debug.

To extract the `bvmdk` file without converting it with `vddk` on the fly during a restore, you need to set the File Daemon debug level to 1. Bacula may report a false error during the restore about the file size. This is normal.

KVM Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This document aims at presenting the reader with information about the **Bacula Enterprise Kernel-based Virtual Machine (KVM) Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features, including clusters.

Scope

Bacula Enterprise KVM module is supported on RHEL. The KVM Plugin is designed to be used when the hypervisor uses local storage for virtual machine disks and **libvirtd** for virtual machine management.

The current version of the KVM Plugin is compatible with:

- Proxmox (note that as of **Bacula Enterprise** 10, a specific *Proxmox Plugin* is available)
- *Redhat RHV* or *Ovirt*
- *OpenStack*

Kernel-based virtual machine (KVM) is an open source virtualization infrastructure built for Linux OS. It requires x86-based processor architecture to operate. KVM consists of two technology components: kernel and user-space. Kernel has two loadable modules: kvm.ko, and either kvm-intel.ko or kvm-amd.ko. The kvm.ko module is responsible for core architecture-independent virtualization processing. The latter two correspond to Intel and AMD processor-specific modules. These modules were merged into the Linux kernel as of version 2.6.20.

Features

Bacula Enterprise is an especially secure, scalable and modular backup and recovery software solution with a wide range of capabilities. Backup up KVM hypervisor(s) and its data in a comprehensive manner is one of its many qualities.

Bacula System's KVM Module is fast, reliable, especially simple to use and establishes confidence in the backup system administrator.

The **Bacula Enterprise** KVM Plugin provides the following main features:

- Automatic virtual machine discovery
- File level backup of virtual machines
- Full, Differential, and Incremental backup level support
- The ability to handle inclusion/exclusion of files
- Backup of virtual machines in a 'running', 'paused' or 'shut off' state

KVM enables Linux to do the system grunt work, so it can focus on handling new virtualization instructions exposed by the hardware. Another key benefit for KVM is that it inherits any continuing system improvement from upstream in the larger Linux community.

It is crucial that the kernel modules do not emulate the virtual machine hardware that the guest OS runs on. KVM uses QEMU to build the virtual machines (VM) that interact with guest OS. Each VM is a regular Linux process, which means familiar Linux commands like 'top' and 'kill' can be used to manage and monitor the VMs.

KVM transforms Linux into a type 1 hypervisor, also known as a bare-metal hypervisor. Type 1 hypervisors directly access the computer's hardware and replace the host OS. They are also considered secure because there is nothing between a hypervisor and a CPU that a potential attacker could tamper with. As part of the Linux kernel, KVM has all the OS-level components. This means that each VM works like a regular Linux process and has dedicated virtual components like memory, disks, CPU and network card.

Bacula Enterprise data backup and recovery allows you to back up not only KVM environments, but covers your entire IT estate through a single plane of glass. It brings unprecedented levels of security, business value benefits and low cost of ownership.

Installation

This article describes how to install Bacula Enterprise Kernel-based Virtual Machine (KVM) Plugin.

KVM Installation with BIM

In order to install the KVM Plugin with BIM, install the File Daemon with BIM and choose to install the KVM Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

KVM Installation with Package Manager

Packages of the KVM Plugin are available for supported platforms. Make a request from your Customer Portal to access the KVM Plugin. The KVM Plugin package needs to be installed onot the KVM host server (hypervisor host), on which **Bacula File Daemon** is already installed.

Note: The KVM Plugin uses snapshots while backing up guest VMs. During a snapshot, blocks modified by the guest VM need to be copied in temporary space, the space required depends on the guest disk activity. By default the space is allocated under `/var/tmp`.

Installation with Package Manager on Debian/Ubuntu

1. Configure the package manager:

Add the following to a file `/etc/apt/sources.list.d/bacula.list`

```
# Bacula Enterprise - KVM
deb https://www.baculasystems.com/dl/@@customer-id@@/debs/kvm/@@bee-version@@/
↪@@os-version@@-@@arch@@/ @@os-version@@ kvm

**@@bee-version@@** should be replaced by the version of Bacula
Enterprise you purchased (16.x.y, 14.x.y)

**@@os-version@@** is the code name of the distribution
(buster/stretch/jessie)

**@@arch@@** Architecture: 32 or 64 bit
```

Note: On Ubuntu 64 bit systems you will need to write **deb [arch=amd64]** instead of **deb**.

A complete example might look like this:

Debian:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/Customer-123456/debs/kvm/16.0.7/buster-
↪64/ buster kvm
```

Ubuntu:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/Customer-123456/debs/kvm/16.0.7/bionic-
↪64/ bionic kvm
```

2. Update your package manager and verify your **Bacula Enterprise** repositories are correctly configured.

```
apt-get update
```

3. Run this command to install the **Bacula Enterprise** packages:

```
apt-get install bacula-enterprise-kvm-plugin
```

Installation with Package Manager on RHEL

1. Configure the package manager:

Add the following to a file `/etc/yum.repos.d/bacula.repo`

```
#  [Bacula-Enterprise-KVM]
name= Bacula Enterprise KVM
baseurl=https://www.baculasystems.com/dl/@@customer-id@@/rpms/kvm/@@bee-
↪version@@/@@rhel@@-@@arch@@/
enabled=1
protect=0
gpgcheck=1

**@@bee-version@@** should be replaced by the version of Bacula
Enterprise you purchased (16.x.y, 14.x.y)

**@@rhel@@** is the version of your RHEL distribution (9/8/7)

**@@arch@@** Architecture: 32 or 64 bit
```

A complete example might look like this:

```
[Bacula-Enterprise]
name=Red Hat Enterprise - Bacula - Enterprise
baseurl=https://www.baculasystems.com/dl/Customer-123456/rpms/kvm/16.0.7/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=1
```

2. Update your package manager and verify your **Bacula Enterprise** repositories are correctly configured.

```
yum update
```

3. Run this command to install the **Bacula Enterprise** packages:

```
yum install bacula-enterprise-kvm-plugin
```

Configuration

This article describes how to configure Bacula Enterprise KVM Plugin.

File Daemon Configuration

On the KVM host server, the **Plugin Directory** option of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` has to point to where the `kvm-fd.so` plugin is installed. The standard directory for Bacula plugins is `/opt/bacula/plugins`

```
FileDaemon {
   Name = bacula-fd
   Plugin Directory = /opt/bacula/plugins
   ...
}
```

The `libvirtd` daemon should be started and accessible. The KVM host server's File Daemon should have local, network-mounted, or SAN access to where KVM images are stored. On the KVM host server, the following command should list all local VMs:

```
# virsh list --all
 Id    Name                           State
----------------------------------------------------
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off
```

If specification of a URI using the virsh `-c` parameter is required, the corresponding **uri=** parameter will also be needed in the plugin command.

```
# virsh -c qemu:///system list --all
 Id    Name                           State
----------------------------------------------------
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off
```

The `install-kvm.sh` script is designed to test the KVM setup of the hypervisor. The script should report an "OK" at the end. If not, any problems reported need to be corrected, and the resulting output should be sent to the Bacula Systems support team if you need assistance.

```
# /opt/bacula/scripts/install-kvm.sh check
Enter the :term:`libvirt` URI to connect libvirtd [qemu:///system]:

Trying to list VMs using virsh -r -c 'qemu:///system' list --all, it should␣
↪not ask for a password.
Id    Name                           State
----------------------------------------------------
 1     gentoo                         running
 2     centos                         paused
 -     debian                         shut off

Did you have to enter a password to get the VM list? [y/N]: N

Enter the name of a guest that will be used to test the :term:`KVM` plugin␣
↪requirements:
gentoo
```

(continues on next page)

(continued from previous page)

```
Trying to mount gentoo filesystem as /tmp/bee-kvm-gentoo.2vsYz
Mount OK.
Attempting to list 10 files from gentoo root filesystem.
/tmp/bee-kvm-gentoo.2vsYz/
/tmp/bee-kvm-gentoo.2vsYz/bin
/tmp/bee-kvm-gentoo.2vsYz/bin/bb
/tmp/bee-kvm-gentoo.2vsYz/bin/dd
/tmp/bee-kvm-gentoo.2vsYz/bin/cp
/tmp/bee-kvm-gentoo.2vsYz/bin/df
/tmp/bee-kvm-gentoo.2vsYz/bin/du
/tmp/bee-kvm-gentoo.2vsYz/bin/ip
/tmp/bee-kvm-gentoo.2vsYz/bin/ln
/tmp/bee-kvm-gentoo.2vsYz/bin/ls

Unmounting gentoo filesystem.
OK: All tests are good.
```

Director Configuration

KVM Plugin Options

Table 26: KVM Plugin Parameters

Option	Default	Description
uri		The URI parameter specifies how to connect to the hypervisor. The documentation page at http://libvirt.org/uri.html list the values supported.
host		Virtual Machine to be backed up. It is possible to specify a list of of hosts separated by a ‘,’ (without spaces).
include		Specify files or directories to backup. It is possible to specify multiple include= parameters in the plugin command line. See <includeexclude>
exclude		Specify files or directories to exclude. It is possible to specify multiple exclude= parameters in the plugin command line. See <includeexclude>
host_pre	No	Prefix all files with the virtual machine name (mandatory when doing multiple virtual machine backup in the same job).
host_sep	,	Specify a host separator used in the host parameter. Example: `host_sep=:`, `host=h1:h2:h3`
abort_on	No	Abort the job if an error is occurring during the job. By default, if a VM is not accessible for example, the Job will end with JobStatus OK (T) and some JobErrors.
timeout	300 seconds	Number of seconds used to run various commands or wait to connect KVM during during a backup (starting with Bacula Enterprise 12.6.1).
unix_use		Unix user used to execute KVM commands via sudo (starting with Bacula Enterprise 12.6.1).

Fileset Examples

The Fileset example below will backup all files from both the centos and gentoo virtual machines.

```
Fileset {
  Name = FS_KVM
  Include {
    Options {
      Signature = MD5
      Compression = LZO
    }
    Plugin = "kvm: host=centos,gentoo"
  }
}
```

If a KVM Fileset contains multiple virtual machines, each file's path in the Catalog will begin with the name of the virtual machine as shown in the example listing below.

```
* list files jobid=100
+---------------------------------+
| filename                        |
+---------------------------------+
| ...                             |
| /centos/boot                    |
| /centos/boot/grub               |
| /centos/boot/grub/menu.lst      |
| ...                             |
| /gentoo/etc/passwd              |
| /gentoo/etc/group               |
| /gentoo/etc/hosts               |
| ...                             |
+ --------------------------------+
```

If a KVM Fileset contains only one virtual machine, each file's path will not be prefixed with the virtual machine's name, as shown below. This default behavior can be overridden by using the **host_prefix** KVM plugin parameter listed in table kvm-options.

```
Fileset {
  Name = FS_KVM_centos
  Include {
    Plugin = "kvm: host=centos"
  }
}
```

```
* list files jobid=101
+---------------------------------+
| filename                        |
+---------------------------------+
| ...                             |
| /boot                           |
| /boot/grub                      |
| /boot/grub/menu.lst             |
| ...                             |
```

(continues on next page)

(continued from previous page)

```
+ --------------------------------+
```

Including and Excluding Files

With the KVM Plugin, by default all files in the virtual machine are backed up and the Fileset's Include and Exclude directives are ignored.

```
Fileset {
  Name = FS_KVM_broken
  Include {
    Plugin = "kvm: host=centos"  # all files on centos VM will be backed up
    File = /etc                  # will be ignored due to KVM plugin
    File = /home                 # will be ignored due to KVM plugin
  }
  Exclude {
    File = /tmp                  # will be ignored due to KVM plugin
  }
}
```

Note: Selection rules defined in Options blocks are still applied.

If certain directories should be excluded from the virtual machine's backup, the **include** and **exclude** plugin options can be used. The Fileset example below will backup files under `/etc` and `/home`. Files below `/home/tmp` will be explicitly excluded.

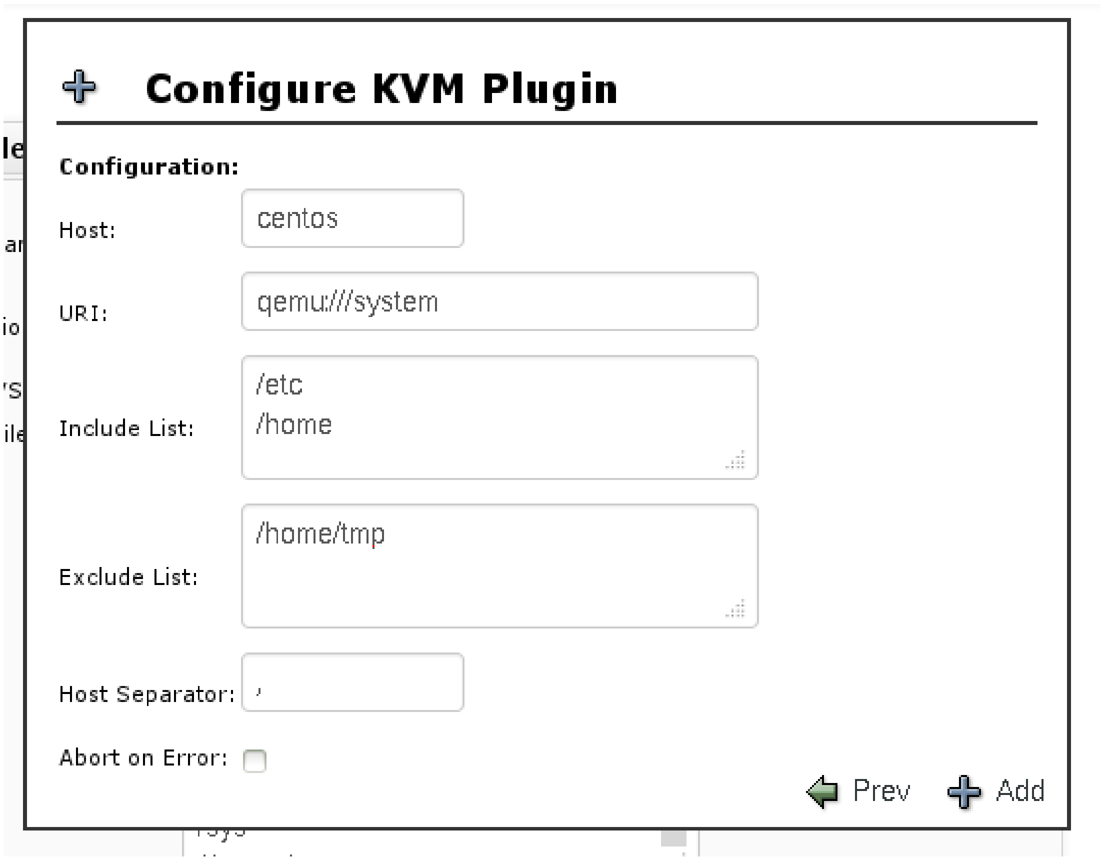
```
Fileset {
  Name = FS_KVM_etc_home
  Include {
    Plugin = "kvm: host=centos include=/etc include=/home exclude=/home/tmp"
  }
}
```

When the **include** plugin option is specified, the default behavior of backing up all files in the VM is overridden and **only** directories specified by the **include** plugin options are backed up.

To manage exclude lists, it is also possible to use the **Exclude Dir Containing** Fileset directive. In the example below, all files and directories on the virtual machine named "centos" will be backed up except for directories containing a file named `.excludeme`

```
Fileset {
  Name = FS_KVM_centos
  Include {
    Options {
    Signature = MD5
    }
    Exclude Dir Containing = ".excludeme"
    Plugin = "kvm: host=centos"
  }
}
```

If more complex Include or Exclude capabilities are required, it may be useful to consider installing a Bacula File Daemon inside the virtual machine instead of using the KVM Plugin.



Configure KVM Plugin

Configuration:

Host:	centos
URI:	qemu:///system
Include List:	/etc /home
Exclude List:	/home/tmp
Host Separator:	,
Abort on Error:	☐

Fig. 30: KVM Configuration

Operations

This article describes details regarding backup and restore with Bacula Enterprise KVM Plugin.

Backup

Common Approaches

To prevent data loss and facilitate system recovery in case the VM is compromised, it is crucial to back up KVM. A common backup strategy is to use a custom script like perl or something else. This type of script can be flexible and can be configured to suit the setup and backup needs of the VM. This process generally involves identifying the VM and for each VM, taking a snapshot of the disks. The snapshot is then dumped on a remote server or a backup storage.

Some scripts may require temporary suspension of the VM's operations. A risk to using scripts is that, while it is easy to access and execute, it can also open up VMs to security vulnerabilities.

Another way to use the snapshot tool is to install all the necessary components on the VM and use this as a base snapshot when it is time to restore.

A backup software may be installed on each VM. The backup process itself can affect the performance of the VM on the server.

This module removes the need to have a client installed in each VM. The module works at the Hypervisor Level, in the same way as Bacula's XEN and PROXMOX modules. The module is a fully integrated, agentless technology.

To back up, the user simply defines a file set.

However, in the instance of needing to restore a file to a running VM, the user would need to have a Bacula Enterprise client installed.

Backup with Bacula Enterprise

Bacula Enterprise enables easy backup and restoration of KVMs at a file level. There is no software installation and scripting required for individual clients. Bacula Enterprise's high level of granularity allows users a high level of control over the data they need to (or don't need to) backup and recover. Auto-detection capabilities also enable automatic VM discovery, which means there is no need to define which VM to backup after initial configuration.

With Bacula Enterprise, it is possible to conduct KVM backup whether the VM is running, paused or shut off. Backing up VMs is a resource-intensive process that can reduce resource allocation to the OS, application files, user data and settings, and consequently, affect the performance of the VM. Bacula's backup tool enables KVM backup without service interruption and data consistency.

Snapshot backup is also available. When this is combined with Bacula Systems' *Global Endpoint Deduplication* engine and comm-line compression, users can save storage and reduce use of network bandwidth.

Restore

Virtual machines are backed up at the file level. To restore a set of files to a virtual machine, a Bacula File Daemon must be installed inside the virtual machine. It is also possible to restore files to a different Bacula client and copy the files to the intended virtual machine afterward.

To restore a file, simply use the bconsole `restore` command, select the backup Job and run a traditional restore Job.

The simplicity of this module can be of great benefit to a System Administrator. Imagine a scenario where we have a "hacked" server that is sending DDOS to the network, and the Sys Admin needs a file from it. She does not want to put her organization's network in danger or at risk by starting a VM to restore a file. Instead, she only selects the specific file to restore, without the need to restore the full VM.

Restoring to a Different Bacula Client

It is also possible to restore files to a different Bacula client and copy the files to the intended virtual machine afterward. Some of the advantages of being able to restore a single VM guest file instead the whole VM are:

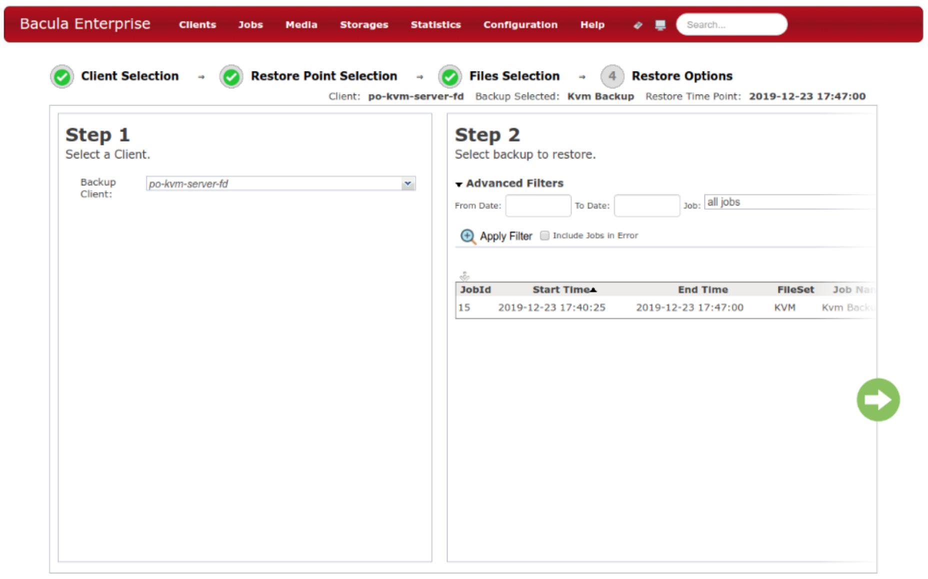
- Saves Time, Network Bandwidth, and Hard Disk Space
- Security
- Facilitates the Restore Workflow
- Removes the need to stop or start a new virtual machine

Single File Restore

Single File Restore is possible with all VM guests where the "libguestfs" library is compatible. KVM and Bacula's KVM module use the library to access the guest's file system.

Restoring with BWeb

The images below demonstrate the sequence to recover KVM with Bacula's BWeb GUI interface

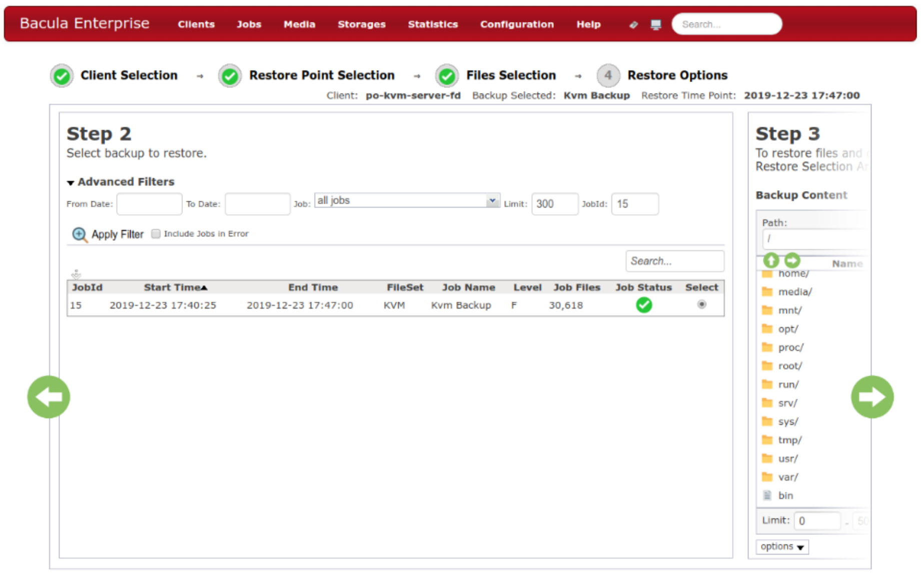


Client: po-kvm-server-fd Backup Selected: Kvm Backup Restore Time Point: 2019-12-23 17:47:00

Step 1
Select a Client.
Backup Client: po-kvm-server-fd

Step 2
Select backup to restore.
From Date: To Date: Job: all jobs

JobId	Start Time	End Time	FileSet	Job Na
15	2019-12-23 17:40:25	2019-12-23 17:47:00	KVM	Kvm Back



Client: po-kvm-server-fd Backup Selected: Kvm Backup Restore Time Point: 2019-12-23 17:47:00

Step 2
Select backup to restore.
From Date: To Date: Job: all jobs Limit: 300 JobId: 15

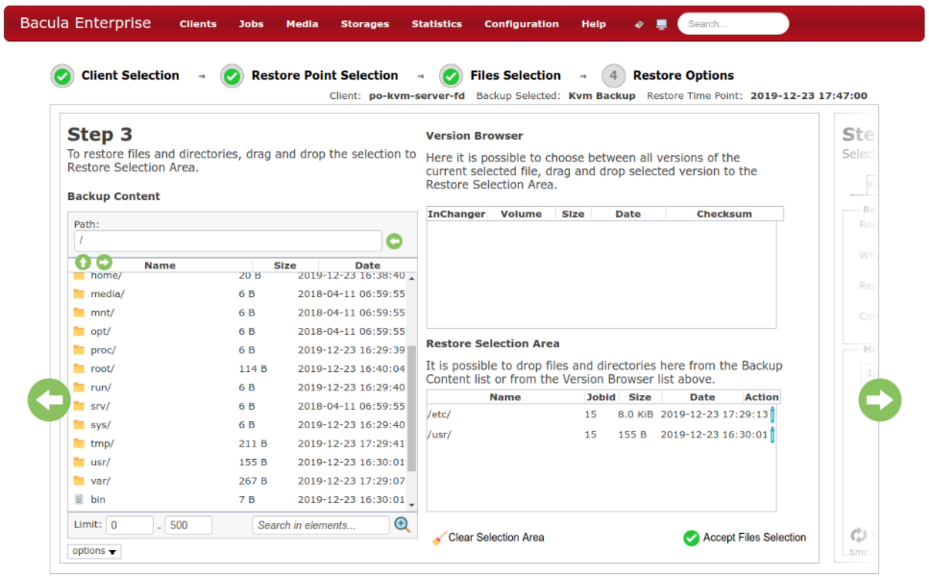
JobId	Start Time	End Time	FileSet	Job Name	Level	Job Files	Job Status	Select
15	2019-12-23 17:40:25	2019-12-23 17:47:00	KVM	Kvm Backup	F	30,618	✓	●

Step 3
To restore files and
Restore Selection

Backup Content
Path: /

Name
home/
media/
mnt/
opt/
proc/
root/
run/
srv/
sys/
tmp/
usr/
var/
bin

Limit: 0



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

Step 3

To restore files and directories, drag and drop the selection to Restore Selection Area.

Backup Content

Path: /

Name	Size	Date
home/	20 B	2019-12-23 16:38:40
media/	6 B	2018-04-11 06:59:55
mnt/	6 B	2018-04-11 06:59:55
opt/	6 B	2018-04-11 06:59:55
proc/	6 B	2019-12-23 16:29:39
root/	114 B	2019-12-23 16:40:04
run/	6 B	2019-12-23 16:29:40
srv/	6 B	2018-04-11 06:59:55
sys/	6 B	2019-12-23 16:29:40
tmp/	211 B	2019-12-23 17:29:41
usr/	155 B	2019-12-23 16:30:01
var/	267 B	2019-12-23 17:29:07
bin	7 B	2019-12-23 16:30:01

Limit: 0 - 500 Search in elements...

options

Version Browser

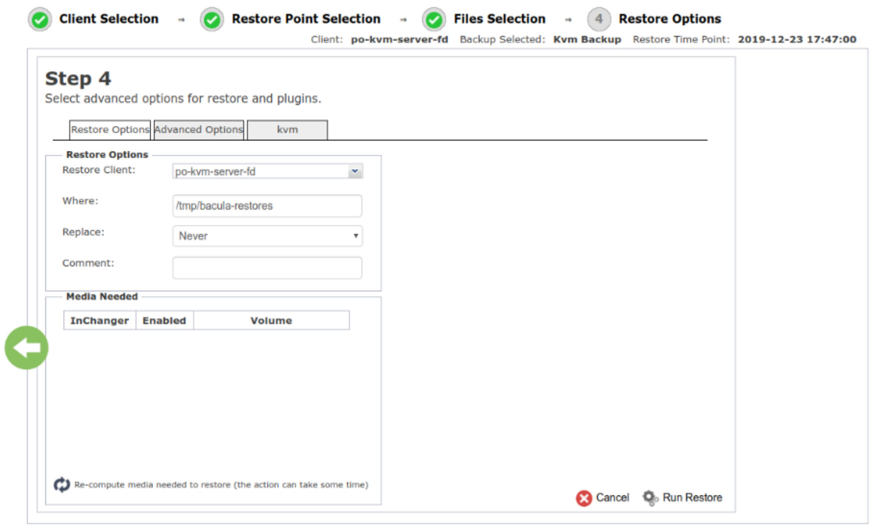
Here it is possible to choose between all versions of the current selected file, drag and drop selected version to the Restore Selection Area.

InChanger	Volume	Size	Date	Checksum

Restore Selection Area

It is possible to drop files and directories here from the Backup Content list or from the Version Browser list above.

Name	Jobid	Size	Date	Action
/etc/	15	8.0 KiB	2019-12-23 17:29:13	
/usr/	15	155 B	2019-12-23 16:30:01	



Client: **po-kvm-server-fd** Backup Selected: **Kvm Backup** Restore Time Point: **2019-12-23 17:47:00**

Step 4

Select advanced options for restore and plugins.

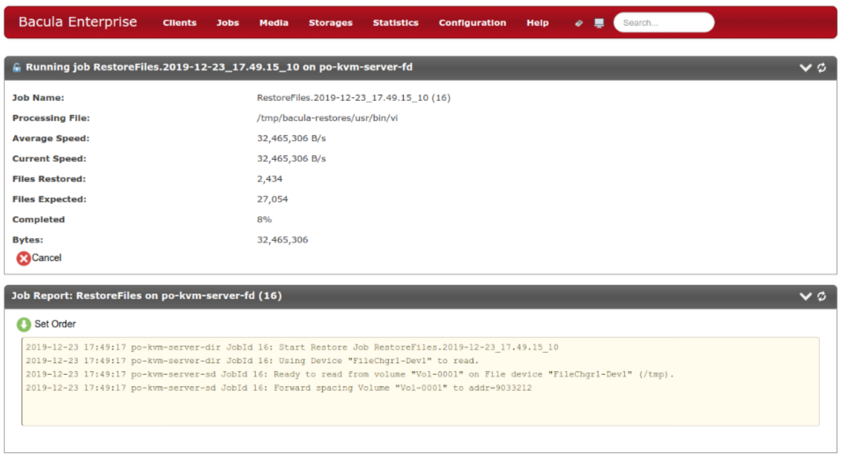
Restore Options

Restore Client:	po-kvm-server-fd
Where:	/tmp/bacula-restores
Replace:	Never
Comment:	

Media Needed

InChanger	Enabled	Volume

Re-compute media needed to restore (the action can take some time)



Running job RestoreFiles.2019-12-23_17.49.15_10 on po-kvm-server-fd

Job Name:	RestoreFiles.2019-12-23_17.49.15_10 (16)
Processing File:	/tmp/bacula-restores/usr/bin/vi
Average Speed:	32,465,306 B/s
Current Speed:	32,465,306 B/s
Files Restored:	2,434
Files Expected:	27,054
Completed	8%
Bytes:	32,465,306

Job Report: RestoreFiles on po-kvm-server-fd (16)

```
2019-12-23 17:49:17 po-kvm-server-dir JobId 16: Start Restore Job RestoreFiles.2019-12-23_17.49.15_10
2019-12-23 17:49:17 po-kvm-server-dir JobId 16: Using Device "FileChgr1-Dev1" to read.
2019-12-23 17:49:17 po-kvm-server-sd JobId 16: Ready to read from volume "Vol-0001" on File device "FileChgr1-Dev1" (/tmp).
2019-12-23 17:49:17 po-kvm-server-sd JobId 16: Forward spacing Volume "Vol-0001" to addr=9033212
```

Limitations

- When performing backup at the KVM domain controller (hypervisor) level, some Bacula features such as client RunScripts will not execute commands at the guest level. Dumping a MySQL database on a guest VM before the backup will require the use of custom scripts, using SSH for example or simply make us of the Bacula Enterprise *MySQL Plugin*. The same can be applied to any application running on the VM with any Bacula Enterprise plugin.
- Virtualization tools (qemu-ga) must be installed on the guest, and you may have to configure the guest to quiesce the file system properly during the snapshot phase.
- Virtual machines cannot be restored in a bare metal recovery way using the KVM Plugin.
- CD ROMS *must* be disconnected from VMs before attempting backup with the KVM Plugin. Else, the system might report the error "guestmount: multi-boot operating systems are not supported".
- The KVM Plugin uses snapshots while backing up guest VMs. During a snapshot, blocks modified by the guest VM need to be copied in temporary space, the space required depends on the guest disk activity. By default the space is allocated under `/var/tmp`.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Proxmox Plugin

Important: Remember to read the *Best Practices chapter* common for all of our *hypervisor plugins*.

This document aims at presenting the reader with information about the **Bacula Enterprise Proxmox Plugin**. The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features, including clusters.

Scope

The Proxmox Plugin was originally introduced with Bacula Enterprise version 10.0.

Our plugin is designed to work seamlessly with the latest releases of Proxmox VE. Bacula is compatible with the supported Proxmox 8.x and earlier versions. If you encounter any problems with your Proxmox VE version, reach out to Bacula Systems Support for assistance.

Features

In order to deliver high quality solutions with reduced total cost of ownership, Bacula Enterprise represents the opportunity to do exactly this, by offering a particularly broad range of features. Its two Proxmox modules are just some of these many features.

Bacula has two different plugins for Proxmox. The first is a Proxmox module where FULL backups are supported, and both virtual machines and LXC containers can be backed up very efficiently. The second is a *QEMU module*, where FULL and INCREMENTAL backups of QEMU virtual machines are supported. DIFFERENTIAL backups are not supported yet. This module does not backup LXC containers.

Both modules provide fast, effective virtual machine bare metal recovery including QEMU and LXC guests.

Detailed list of features of the Proxmox module:

- Snapshot-based online backup of any guest VM including QEMU and LXC guests
- Full image-level backup
- Ability to restore complete virtual machine image
- Ability to restore QEMU VM archive (.vma) to an alternate directory
- Ability to restore LXC VM archive (.tar) and configuration to an alternate directory
- Ability to scan a Proxmox cluster to automatically create a backup configuration for each virtual machine

Backup and Restore Strategies

Image Backup

With the image backup level strategy, the Bacula Enterprise Proxmox Plugin will save the Client disks as raw images for QEMU VMs and as a `tar` archives for LXC VMs, in the Proxmox context.

For this to work, it is not needed a Bacula File Daemon in each guest VM. The Bacula Proxmox Plugin will contact the Proxmox hypervisor to read and save the contents of virtual machine disks using snapshots (default behavior) and dump them using the Proxmox API.

Bacula does not need to walk through the Client filesystems to open, read, close and `stat` files, so it consumes less resources on the Proxmox infrastructure than a file level backup on each guest machine would. On the other hand, Bacula will also read and save useless data such as swap files or temporary files.

The Proxmox Plugin will save not only the disk images of the guest VM, but also guest VM configurations which allows for very easy guest VM restores.

Installing Bacula Client on Each Guest

With this first strategy, you do not use the Bacula Enterprise Proxmox Plugin, but instead install a Bacula Enterprise File Daemon inside every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on your Proxmox system, you will use Bacula's Schedules, Priorities, and Job concurrency settings to spread backup jobs throughout the backup window. Since all guest VMs could use the same storage on the Proxmox hypervisor, running all your backup jobs at the same time can create a bottleneck on the disk/network subsystem.

Installing a Bacula Enterprise File Daemon in each virtual machine permits you to manage your virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files.
- Checksum of individual files for Virus and Spyware detection.
- Verify Jobs.
- File or Directory exclusion (such as swap or temporary files).
- File level compression.
- Accurate backups.

Installation

This article describes how to install Bacula Enterprise Proxmox Plugin.

The Bacula File Daemon and its Proxmox Plugin need to be installed on the host of the Proxmox hypervisor which runs the guest VMs that are to be backed up. Proxmox uses a customized Debian distribution, so the Bacula Enterprise File Daemon for that platform has to be used.

Installation of the Bacula Enterprise Proxmox Plugin is most easily done by adding the repository file suitable for the existing subscription and the Debian version (the distributions package manager configuration) that Proxmox is built upon.

Prerequisites

The **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `proxmox-fd.so` plugin file is installed. The standard Bacula plugin directory is `/opt/bacula/plugins`.

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
...
}
```

Steps

Installation example:

1. Insert the following content to the `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪stretch-64/ stretch main
deb https://www.baculasystems.com/dl/@customer-string@/debs/proxmox/@version@/
↪stretch-64/ stretch proxmox
```

2. Run `apt-get update`.

```
apt-get update
```

3. Then, to install the Plugin, run: `apt-get install bacula-enterprise-proxmox-plugin`

```
apt-get install bacula-enterprise-proxmox-plugin
```

Manual installation of the packages can be done after downloading the right files from the Bacula Systems provided download area, and then using the package manager to install.

Configuration

This article describes how to configure Bacula Enterprise Proxmox Plugin.

The plugin is configured using **Plugin Parameters** defined in a Filesets **Include** section of the Bacula Enterprise Director configuration.

General Parameters

The following Proxmox plugin parameter effects any type of Job (Backup, Estimation, or Restore).

abort_on_error[=<0 or 1>]
: specifies whether or not the plugin should abort execution (and the Bacula Job) if a fatal error occurs during a Backup, Estimation, or Restore operation. This parameter is optional. The default value is 0.

Estimation and Backup Parameters

These plugin parameters are relevant only for Backup and Estimation jobs:

vm=<name-label>
: specifies a guest VM name to backup. All guest VMs with a <`name-label`> provided will be selected for backup. Multiple `vm=...` parameters are allowed. If guest VM with <`name-label`> cannot be found, then a single job error will be generated, and the backup will proceed to the next VM unless `abort_on_error` is set, which will cause the backup job to be aborted. This parameter is optional.

vmid=<vmid>
: specifies a guest VM VMID to backup. Multiple `vmid=...` parameters may be provided. If a guest VM with <`vmid`> cannot be found, a job error will be generated and the backup will proceed

to the next VM unless `abort_on_error` is set which will cause the backup job to be aborted. This parameter is optional.

include=<name-label-regex>
specifies a list of a guest VM names to backup using regular expression syntax. All guest VMs with names matching the name regular expression provided will be selected for backup. Multiple `include=...` parameters may be provided. The match is performed case insensitive.

If no guest VMs are matching the name expression provided, the backup will proceed to the next parameters or finish successfully without backing up any VMs. The `abort_on_error` parameter will not abort the job when no guest VMs are found using name matching.

This parameter is optional.

exclude=<name-label-regex>
specifies a list of guest VMs names which will be excluded from backup using regular expression matching. All guest VMs with names matching the provided regular expression, and selected for backup using the `include=...` parameter will be excluded. The match is performed case insensitive.

This parameter does not affect any guest VM selected to be backed up using `vm=...` or `vmid=...` parameters.

Multiple `exclude=...` parameters may be provided.

This parameter is optional.

mode=<snapshot|suspend|stop>
specifies the default backup mode to use. The **snapshot** mode will generate the live backup using snapshots which minimizes the downtime of a VM during the the backup process. The **suspend** mode will suspend the guest VM during a backup to achieve a consistent backup. The guest VM will remain suspended during backup and will resume running once the backup of this guest VM finishes. The **stop** mode will shut down the guest VM before backup and the VM will be restarted when its backup finishes.

This parameter is optional. If not set, then **snapshot** mode will be used by default.

vzdump_storage=<proxmox_storage>
specifies the Proxmox storage resource <`proxmox_storage`> used by `vzdump` command during backup when the default *'local'* Proxmox storage resource has no backup content type available. The <`proxmox_storage`> should point to the Proxmox storage resource which has the valid backup content type added to the resource. See: `vzdumpstorage`. This parameter is optional. If not set, then a default *'local'* Proxmox storage resource will be used.

bwlimit=<N>
specifies if the backup should be performed with an I/O bandwidth limitation (in KBytes per second) on the hypervisor side. This limitation is independent from Bacula's own bandwidth limitation feature, and is performed on a different level.

This parameter is optional. If not set, no bandwidth limitation on the hypervisor will be applied.

verbose
specifies if the backup should be display more verbose information during the vzdump operation.

This parameter is optional. If not set, the vzdump step is not verbose.

If none of the parameters `vm=...`, `vmid=...`, `include=...` and `exclude=...` are specified, all available guest VMs on the Proxmox hypervisor will be backed up.

Restore Parameters

During restore, the Proxmox Plugin will use the same parameters which were set for the backup job and saved in the Catalog. Some of them may be changed during the restore process if required.

storage: <storage>
specifies a Proxmox Storage where restored guest VMs will be restored to. If not set, then a guest VM will be restored to the Proxmox Storage it was backed up from. If this parameter points to a nonexistent Storage, the original Storage of the guest VM will be used.

This parameter is optional.

pool: <resourcepool>
specifies a Proxmox Resource Pool to which a restored guest VM will be attached. If not set, a restored guest VM will have no Resource Pool attached. When this parameter indicates a nonexistent pool, no pool will be attached.

This parameter is optional.

sequentialvmid: <yes or no>
specifies what new VMID allocation procedure to use. If set to **yes**, the new guest VM will get the first available VMID based on the maximum VMID found on the Proxmox hypervisor. Otherwise, the restored guest VM will get a randomly generated VMID in a range of 10 above the highest used VMID found on the Proxmox hypervisor. This is the default.

Setting `sequentialvmid: yes` mitigates some conflicts which may occur during concurrent restores.

Fileset Examples

Example 1

In the example below, all guest VMs will be backed up.

```
Fileset {
  Name = FS_ProxmoxAll
  Include {
    Plugin = "proxmox:"
  }
}
```

Example 2

In the example below, a single guest VM with name-label of “VM1” will be backed up.

```
Fileset {
  Name = FS_Proxmox_VM1
  Include {
    Plugin = "proxmox: vm=VM1"
  }
}
```

Example 3

The same example as above, but using **vmid** instead:

```
Fileset {
  Name = FS_Proxmox_VM1
  Include {
    Plugin = "proxmox: vmid=101"
  }
}
```

Example 4

In the following example, all guest VMs which contain “Prod” in their names will be backed up.

```
Fileset {
  Name = FS_Proxmox_ProdAll
  Include {
    Plugin = "proxmox: include=Prod"
  }
}
```

Example 5

In this final example, all guest VMs except VMs whose name begins with “Test” will be backed up.

```
Fileset {
  Name = FS_Proxmox_AllbutTest
  Include {
    Plugin = "proxmox: include=.* exclude=^Test"
  }
}
```

Operations

This article describes details regarding backup, restore, resource listing, and cluster support with Bacula Enterprise Proxmox Plugin.

Backup

The backup of a single guest VM consists of the following steps:

1. Save guest VM configuration (for LXC guest VMs).
2. Create a new backup snapshot (default), suspending or stopping the guest VM as requested.
3. Execute `vzdump` and save data.

Backups can be performed for guest VMs in any power state (running or stopped). The Proxmox hypervisor will automatically create the required backup snapshot and remove it after the backup. Any other guest VMs snapshots will be unaffected.

The Proxmox Plugin will inform you about every guest VM backup start and finish:

```
 JobId 68: Start Backup JobId 68, Job=proxmox.2018-01-25_11.25.05_21
 JobId 68: Using Device "FileChgr1-Dev2" to write.
 JobId 68: Volume "Vol-0002" previously written, moving to end of data.
 JobId 68: proxmox: Start Backup vm: ubuntu-container (101)
```

(continues on next page)

(continued from previous page)

```
 JobId 68: proxmox: Backup of vm: ubuntu-container (101) OK.
...
```

The backup will create a single (`.vma`) file for any QEMU guest VM and two files (`.conf` and `.tar`) for any LXC guest VM which is saved. Inside Bacula, those are represented as follows.

- `/@proxmox/qm/<name-label>/VM<vmid>.vma` for QEMU guest VMs
- `/@proxmox/lxc/<name-label>/VM<vmid>.conf` and
- `/@proxmox/lxc/<name-label>/VM<vmid>.tar` for LXC guest VMs

Multiple files will be created during a backup if multiple guest VMs are backed up with one job. However, note that backing up multiple VMs goes against *hypervisor best practices*. The distinct file names as shown above allow to locate the proper guest VM archive for restore.

Tip: You can exclude machine disks from Proxmox image backup in the Proxmox Console VM Disk Settings screen.

Restore

The Proxmox Plugin provides two targets for restore operations:

- Restore to Proxmox hypervisor as new or original guest VM.
- Restore to a local directory as Proxmox archive files (`.vma` or the pair of `.tar` and `.conf` files).

Restore to Proxmox

To use this restore method, the `where=` parameter of a Bacula restore is used. The guest VM archive will be sent to the Proxmox hypervisor and restored as a new guest VM if the VMID of the restored VM is already allocated. Otherwise, the guest VM will be restored with its original Proxmox.

To restore a VM or VMs to a Proxmox hypervisor, the administrator should execute the restore command and specify the `where` parameter as in this example:

```
* restore where=/
```

and then set any other required restore plugin parameters for the restore.

In the following restore session example, the `sequentialvmid` plugin restore option is set to “yes”:

```
* restore where=/
...
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/pve-dir.restore.1.bsr
Where:           /opt/bacula/archive/bacula-restores
Replace:         Always
Fileset:         Full Set
Backup Client:   pve-fd
Restore Client:  pve-fd
```

(continues on next page)

(continued from previous page)

```
Storage:         File1
When:            2018-01-30 14:58:21
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : proxmox: vmid=108 abort_on_error
Plugin Restore Options
storage:             *None*               (*Default locations*)
pool:                *None*               (*Default pool*)
sequentialvmid:      *None*               (*No*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: storage (Storage location for restore)
     2: pool (Add the VM to the specified pool on restore)
     3: sequentialvmid (Allocate new vmid in sequential order)
Select parameter to modify (1-3): 3
Please enter a value for sequentialvmid: yes
Plugin Restore Options
storage:             *None*               (*Default locations*)
pool:                *None*               (*Default pool*)
sequentialvmid:      yes                  (*No*)
Use above plugin configuration? (yes/mod/no):
```

The restore job log will indicate which guest VM is restored and which new guest VM was created:

```
JobId 76: Start Restore Job RestoreFiles.2018-01-25_13.50.31_29
JobId 76: Using Device "FileChgr1-Dev1" to read.
JobId 76: Ready to read from volume "Vol-0004" on File device "FileChgr1-Dev1
↪" (/opt/bacula/archive).
JobId 76: proxmox: VM restore: lxc/ubuntu-container/VM101 as VM222
JobId 76: End of Volume "Vol-0004" at addr=47137166325 on device "FileChgr1-
↪Dev1" (/opt/bacula/archive).
```

The new guest VM created during the restore will get a new VMID (if the original VMID is not available anymore) but the name/hostname will stay the same as it was with the original VM.

All other guest VM configuration parameters will be restored as they were backed up, including network MAC Addresses. For this reason, it is recommended to inspect and possibly update a guest VMs configuration before it is started. Otherwise, resource conflicts may arise.

New guest VMs will get a new VMID which will be allocated by the Bacula Proxmox Plugin as a random value in a range between the maximum VMID already allocated $+1$ and $+11$. This procedure is intended to mitigate possible resource allocation conflicts which could be occur when two or more guest VM restores or creations are executed at the same time, as Proxmox has no conflict resolution mechanism itself for this situation.

Using the plugin restore option `sequentialvmid` this behavior can be modified so that the newly restore guest VM gets the next available VMID, which will help limiting the range of VMID assigned over time, but will make concurrent restores unreliable.

The Storage target to be used for the restored guest VM disk(s) can be set using the **storage** plugin restore option. If this option is not set, all guest VM disks will be restored to their original Storage.

To list available Storages, a listing mode is available.

If an improper (e. g., non-existent) Storage is chosen for a restore, the restore process will create the guest VM in the Proxmox hypervisors default Storage.

Restore to Local Directory

It is possible to restore the guest VM data to a file and not to pass the data to the hypervisor. To do so, the``where`` restore option should point to a directory where the Proxmox plugin is installed:

```
* restore where=/tmp/bacula/restores
```

If the path does not exist, it will be created by the Bacula Proxmox Plugin.

Check the following example for the test “VM local restore”:

```
JobId 90: Start Restore Job RestoreFiles.2018-01-30_15.04.12_05
JobId 90: Using Device "FileChgr1-Dev1" to read.
JobId 90: Forward spacing Volume "Vol-0001" to addr=45406565308
JobId 90: proxmox: VM local restore: qm/ubuntu-server/VM108
```

The restore job log will show that the restore was done to a local directory.

Resource Listing

The Bacula Enterprise Proxmox Plugin supports the new Plugin Listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available Proxmox resources such as:

- List of guest VM name-labels
- List of guest VM VMIDs
- List of Proxmox Storages
- List of Proxmox Resource Pools

The new feature uses the special `.ls` command with a new `plugin=<plugin>` parameter. The command requires the following parameters to be set:

client=<client>
    A Bacula Client name with the Proxmox Plugin installed.

plugin=<plugin>
: A Plugin name, which would be **proxmox:** in this case, with optional plugin parameters as described in section **genericparameters**.

path=<path>
: An object path to display.

The supported values for a `path=<path>` parameter are:

/
: to display object types available to list.

vm
: to display a list of guest VM name-labels.

vmid
: to display a list of guest VM VMIDs and name-label pointers.

storage
: to show the list of available Storages.

pool
: to display the list of Resource Pools.

To display available object types, follow the following command example:

```
*.ls client=pve-fd plugin=proxmox: path=/
Connecting to Client pve-fd at pve:9102
drwxr-x---   1 root     root               0 2018-01-30 15:08:08  vm
drwxr-x---   1 root     root               0 2018-01-30 15:08:08  vmid
drwxr-x---   1 root     root               0 2018-01-30 15:08:08  storage
drwxr-x---   1 root     root               0 2018-01-30 15:08:08  pool
2000 OK estimate files=4 bytes=0
```

To display the list of all available guest VMs, the following command example can be used:

```
*.ls client=pve-fd plugin=proxmox: path=vm
Connecting to Client pve-fd at pve:9102
-rw-r-----   1 root     root      2251187813 2018-01-30 15:08:49  ubuntu-
↪container
-rw-r-----   1 root     root       594332876 2018-01-30 15:08:50  fedora-
↪container
-rw-r-----   1 root     root      1288490188 2018-01-30 15:08:52  openhab-test
-rw-r-----   1 root     root       680735539 2018-01-30 15:08:53  vm1
-rw-r-----   1 root     root       490733568 2018-01-30 15:08:54  testvm1
-rw-r-----   1 root     root     34359738368 2018-01-30 15:08:54  ubuntu-
↪server-template
-rw-r-----   1 root     root     34359738368 2018-01-30 15:08:55  rhel-server
-rw-r-----   1 root     root     68719476736 2018-01-30 15:08:56  testvm2
-rw-r-----   1 root     root     34359738368 2018-01-30 15:08:56  ubuntu-
↪server
-rw-r-----   1 root     root     34359738368 2018-01-30 15:08:57  vm2
2000 OK estimate files=10 bytes=211,463,910,192
```

To display the list of guest VM VMIDs, use the following command example:

```
*.ls client=pve-fd plugin=proxmox: path=vmid
Connecting to Client pve-fd at pve:9102
 root     root      2251187813 2018-01-30 15:09:37  101 -> ubuntu-container
 root     root       594332876 2018-01-30 15:09:38  102 -> fedora-container
 root     root      1288490188 2018-01-30 15:09:40  103 -> openhab-test
 root     root       680735539 2018-01-30 15:09:41  106 -> vm1
 root     root       490733568 2018-01-30 15:09:42  107 -> testvm1
 root     root     34359738368 2018-01-30 15:09:42  100 -> ubuntu-server-
↪template
 root     root     34359738368 2018-01-30 15:09:43  104 -> rhel-server
 root     root     68719476736 2018-01-30 15:09:43  105 -> testvm2
 root     root     34359738368 2018-01-30 15:09:44  108 -> ubuntu-server
 root     root     34359738368 2018-01-30 15:09:45  201 -> vm2
2000 OK estimate files=10 bytes=211,463,910,192
```

The VM and VMID lists display an estimated size of the guest VM.

To display available Proxmox Storages, the following command example can be used:

```
*.ls client=pve-fd plugin=proxmox: path=storage
Connecting to Client pve-fd at pve:9102
brw-r-----    1 root     root                 0 2018-01-30 15:11:18  local-lvm
brw-r-----    1 root     root                 0 2018-01-30 15:11:18  local
brw-r-----    1 root     root                 0 2018-01-30 15:11:18  sunnfs
2000 OK estimate files=3 bytes=0
```

And, finally, use the following to show available Proxmox Resource Pools:

```
*.ls client=pve-fd plugin=proxmox: path=pool
Connecting to Client pve-fd at pve:9102
brw-r-----    1 root     root                 0 2018-01-30 15:12:27  testpool
brw-r-----    1 root     root                 0 2018-01-30 15:12:27  Development
brw-r-----    1 root     root                 0 2018-01-30 15:12:27  Production
brw-r-----    1 root     root                 0 2018-01-30 15:12:27  Sales
brw-r-----    1 root     root                 0 2018-01-30 15:12:27  Marketing
2000 OK estimate files=5 bytes=0
```

Cluster Support

Important: Since Bacula version 16.0.7, a new solution has been introduced to replace the scan_proxmox_cluster tool. It is highly recommended to use the new solution - *Automatic Object Integration (Scan Plugin)* as the scan_proxmox_tool will soon be deprecated. See an example for Proxmox.

At a regular interval, Bacula can contact a Proxmox cluster member, list all virtual machines that are hosted, and adapt the configuration dynamically.

The `scan_proxmox_cluster` tool is used to analyze a Proxmox cluster and create individual Filesets and Backup Jobs for each virtual machine or container found. The virtual machines selected by this tool may be based on the virtual machine names, or by using regular expressions.

```
cluster: cluster1
node:    proxmox1
VM:      debian1    -> Job=j_cluster1_debian1
                       Fileset=fs_cluster1_debian1
                       Client=proxmox1
```

The Job's Client directive will be set to the Proxmox cluster node member.

If a virtual machine is no longer detected, the Job resource will be disabled or removed from the configuration.

If a virtual machine is on an other member of the cluster, the Job Client directive will be adapted automatically.

Cluster Support Installation

Bacula's Proxmox cluster support requires the installation of the library `Net::Proxmox::VE` which is available at: http://github.com/mrproper/proxmox-ve-api-perl

The Bacula Enterprise DAG repository includes a package for this library named `perl-Net-Proxmox-VE` for RHEL.

On RHEL, the Proxmox library needs to have the package `perl-LWP-Protocol-https` installed, otherwise the message `Login failed. Protocol scheme 'https' is not supported` will be displayed.

Cluster Support Configuration

Bacula's Proxmox cluster support is available with BWeb Management Console.

In this example, we have a cluster named `cluster1` with 3 nodes, `prox1`, `prox2` and `prox3`. 5 virtual machines are defined, `vm1`, `vm2`, `vm3`, `vm4`, `vm5`.

The following steps are needed to activate the support:

- Install a Bacula Enterprise FileDaemon on each node of the Proxmox cluster (prox1, prox2 and prox3)
- Install and configure BWeb Enterprise Management Configuration Module (Configuration -> Configure Bacula -> Complete the Configuration Wizard)
- Define one Client resource per cluster member in the Director configuration file `bacula-dir.conf`. For simplicity, the Bacula Client name can match the DNS Promox server name (example "prox1").
- Define a JobDefs with the parameters shared by the Proxmox jobs (Schedule, Storage, Priority, . . .)
- Configure Proxmox access in `/opt/bacula/etc/pve.conf` (click *here* for more information)

```
# chown bacula /opt/bacula/etc/pve.conf
# chmod 600 /opt/bacula/etc/pve.conf
# cat /opt/bacula/etc/pve.conf
[default]
username=root
password=password_of_root
```

(continues on next page)

(continued from previous page)

```
host=prox1
skip_certificate
realm=pam
```

- Execute the `scan_proxmox_cluster` as the unix user `bacula`

  ```
  # /opt/bweb/bin/scan_proxmox_cluster --cluster cluster1 --node prox1 --
  ↪jobdefs ProxmoxJobs
  INFO Using Job Client directive to prox1
  tar: Removing leading `/' from member names
  INFO Doing a backup of the previous configuration tree in bacula-etc.
  ↪2019-05-06_10:15.tar.gz
  INFO Job Modification Summary:

  Added:
  - j_cluster1_vm1
  - j_cluster1_vm2
  - j_cluster1_vm3
  - j_cluster1_vm4
  - j_cluster1_vm5

  Disabled:

  Existing:

  Removed:
  ```

  The `cluster` option is used to name the resources that will be created. The `node` option is used to select the virtual machines to create. The list of the changes will be displayed in the output of the script. In the BWeb Management Configuration Module, a workset for the Proxmox cluster will be displayed with the list of the changes to apply to the configuration. The option `--commit_and_reload` can activate the changes automatically.

  ```
  scan_proxmox_cluster --cluster cluster1 --node prox1 --jobdefs␣
  ↪ProxmoxJobs --commit_and_reload
  ```

- Schedule the `scan_proxmox_cluster` for each node of the Proxmox cluster at a regular interval with a Bacula Admin Job for example.

It is possible to set various options for the Job or the Fileset.

If the Proxmox node name is not equal to the Bacula Client name, it is possible to specify `--directive Client=name` in the command line.

Example:

```
scan_proxmox_cluster --cluster cluster1 --node prox1 --jobdefs DefaultJob --
↪directive Client=prox1-fd
```

Example of a complete configuration will be:

```
#### Configuration for the Cluster
Client {
  Name = prox1
```

(continues on next page)

(continued from previous page)

```
  Password = xxx
  Address = prox1
  File Retention = 5 years
  Job Retention = 5 years
  Catalog = MyCatalog
}
Client {
  Name = prox2
  Password = xxx
  Address = prox2
  File Retention = 5 years
  Job Retention = 5 years
  Catalog = MyCatalog
}
JobDefs {
  Name = PromoxJobs
  Priority = 10
  Type = Backup

  Storage = File
  Schedule = AtNight
  Pool = Default
  Messages = Standard
}
Job {
  Name = UpdateCluster1
  Type = Admin
  JobDefs = ProxmoxJobs
  Schedule = BeforeNight
  RunBeforeJob = "/opt/bweb/bin/scan_proxmox_cluster --cluster cluster1 --
↪node prox1 --commit_and_reload --jobdefs ProxmoxJobs"
  RunBeforeJob = "/opt/bweb/bin/scan_proxmox_cluster --cluster cluster1 --
↪node prox2 --commit_and_reload --jobdefs ProxmoxJobs"
  RunBeforeJob = "/opt/bweb/bin/scan_proxmox_cluster --cluster cluster1 --
↪node prox3 --commit_and_reload --jobdefs ProxmoxJobs"

}
################################################################
#### The following example configuration snippet was generated by the scan_
↪proxmox_cluster program
Fileset {
  Name = fs_pve_vm1
  Include {
    Plugin = "proxmox: vm=vm1"
  }
}
Job {
  Name = j_pve_vm1
  Client = prox1
  Fileset = fs_pve_vm1
  JobDefs = ProxmoxJobs
}
```

(continues on next page)

(continued from previous page)

```
Fileset {
  Name = fs_pve_vm2
  Include {
    Plugin = "proxmox: vm=vm2"
  }
}
Job {
  Name = j_pve_vm2
  Client = prox1
  Fileset = fs_pve_vm2
  JobDefs = ProxmoxJobs
}
...
```

Command Line Options

- `--pvefile file` Get credentials from a file (default: /opt/bacula/etc/pve.conf)
- `--profile=name` Profile name to use in the pvefile
- `--username string` Proxmox server username
- `--password string` Proxmox server password
- `--host string` Proxmox server host
- `--realm` pam or pve
- `--director string` Director name. If not provided, the first Director is used
- `--jobdefs string` JobDefs resource name
- `--job 'job_%v'` Jobs resource name pattern (
- `--fileset 'fs_%v'` Filesets resource name pattern (
- `--directive key=value` (storage|client|schedule …)=value definition
- `--fs_option key=value` Fileset options such as: signature, compression …
- `--plugin_option key=value` Fileset plugin options such as abort_on_error, index …
- `--vm value` Virtual machine to backup (don't use with vms_(include|exclude))
- `--pool value` Pool value to backup
- `--node value` Node to backup
- `--type value` Host type (lxc, qemu)
- `--vms_include pattern` Regular expression to include virtual machines
- `--vms_exclude pattern` Regular expression to exclude virtual machines
- `--commit_and_reload` Commit changes and reload just after resources are created
- `--description` New jobs' description
- `--remove_jobs` Remove jobs instead of disabling them
- `--json` Print output as JSON string

Storing Proxmox Credentials on Disk

The --pvefile argument can refer to a configuration file stored on disk. This file may contain the information used to connect to the Proxmox Cluster. It is possible to store multiple Proxmox server profiles and reference them with the --profile option.

- username=string Proxmox server username (default “root”)
- password=string Proxmox server password
- host=string Proxmox server host
- realm=string pam or pve
- cluster=string Name of the cluster (used to generate resources)
- skip_certificate Do not check SSL certificates
- Other parameters used in pve_get_address program
- failback=string Address returned when the virtual machine is not found
- base=string Used to determine the Virtual Machine name from a pattern

```
# cat /opt/bacula/etc/pve.conf
[cluster1]
password=pass1
host=prox1
realm=pam
skip_certificates
cluster=cluster1

[cluster2]
password=pass3
host=prox3
realm=pam
skip_certificate
cluster=test
```

Troubleshooting

If you encounter a following error during backup job:

```
Fatal error: proxmox: Error closing backend. Err=Unknown error during program␣
↪execvp
```

You should check the Proxmox Task log for the following error:

```
TASK ERROR: could not get storage information for 'local':
       can't use storage 'local' for backups - wrong content type
```

Then your ’local’ Proxmox storage resource is missing the required ’backup’ content type. This kind of configuration is expected by vzdump Proxmox command to save a backup task log and guest vm configuration. As a remedy you can add a missing ’backup’ content type to the ’local’ Proxmox storage resource or configure your backup job to use a different Proxmox storage resource for that. In the latter case you should add a vzdump_storage=... parameter to your configuration.

Limitations

This article presents limitations of the Proxmox Plugin.

- Granular restore (*Single Item Restore*) is not available yet. A file level backup of the VM with the Bacula FD inside the Guest VM is needed to enable single file restores of a Proxmox guest VM. Read more here.
- Concurrent backups of the same guest VM are not possible.
- Only Full level backups are possible. This is a Proxmox limitation as its API does not provide methods suitable for other backup levels. This limitation is described in details in the Features chapter, which also describes another module, *QEMU* that is free of that limitation.
- VM templates available on the Proxmox system can not be backed up. This is also a Proxmox limitation.
- In listing mode with the **vm** or **vmid** parameters, the plugin will display both guest VMs and VM templates. This limitation will be removed in the future.
- Backup of LXC created in **unprivileged** mode could fail with **Permission denied** error in Proxmox Task execution log and a following job messages error log:

  ```
  Error: proxmox: Error closing backend. Err=Unknown error during program␣
  ↪execvp
  ```

  This is a known issue which will be removed in future release of the plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

3.2 Containers

Kubernetes Plugin

Containers provide a lightweight system-level virtualization solution with minimal overhead, as applications within virtual partitions utilize the operating system's standard system call interface without requiring emulation or running in a separate virtual machine. Kubernetes manages a set of containers to create a flexible execution environment for various applications and services

The Bacula Enterprise Kubernetes Plugin is designed to securely store all critical Kubernetes objects and meta-data that are essential for the functioning of the application or service. This includes the following name-spaced objects:

- Config Map
- Daemon Set
- Deployment
- Endpoint
- Ingress (From Bacula Enterprise version **16.0.14**)
- Limit Range
- Pod
- Persistent Volume Claim

- Pod Template
- Replica Set
- Replication Controller
- Resource Quota
- Secret
- Service
- Service Account
- Stateful Set
- PVC Data Archive

Note: The PVC Data does not precisely correspond to a Kubernetes Object, but rather serves as a repository of actual data that exists on the chosen PVC.

---

and non name-spaced objects:

- Namespace
- Persistent Volume
- Storage Class

All name-spaced objects which belong to a particular namespace are grouped together for easy backup data browsing and recovery.

Proper authorization is necessary for users and service accounts to access the server API. This involves going through authentication, authorization, and admission control. In order to effectively back up Kubernetes objects, it is essential to have a user or service account with the appropriate permissions and rights to be authenticated and authorized to access the API server and the objects that need to be backed up.

To ensure successful object configuration backups, it is essential for the user or service account to have the capability to read and list the objects. Additionally, for PVC Data backup, the user or service account must also possess the ability to create and delete pods. This is necessary as the plugin will need to create and delete the Bacula Backup Proxy Pod during the backup process.

See the Kubernetes documentation for more details.

Features

The artile aims at presenting the Kubernetes Plugin features.

Note: Only Full level backups are supported.

They are:

- Kubernetes cluster objects configuration backup
- Ability to restore single Kubernetes configuration object
- Ability to restore Kubernetes object configuration to local directory
- Kubernetes Persistent Volumes data backup and restore

- Ability to restore Kubernetes Persistent Volumes data to local directory
- Ability to use Kubernetes CSI driver volume snapshot and cloning features to perform Persistent Volume data backup
- Ability to adapt the best technique to each Persistent Volumes data backup
- Ability to execute user defined commands on required Pod containers to prepare and clean data backup
- Configure Kubernetes workload backup requirements directly with Pod annotations.
- Cleanup pods and cloned pvcs from old backup jobs that, for any reason, did not finish correctly. This feature is automatic.

Installation

The Bacula File Daemon and the Kubernetes Plugin can be installed outside of the Kubernetes cluster on a server which has access to the Kubernetes API, or inside a protected cluster in a Container/Pod. The Kubernetes Plugin can be installed on different operating systems and distributions, so the Bacula Enterprise File Daemon for the correct operating system and platform has to be used.

There is no need, or in some solutions (when `K8S` is a cloud service like `GKE` or `EKS`), it is even a possible to install a Bacula File Daemon and the Kubernetes Plugin on a Kubernetes Master Server (`etcd`, `control pane`).

Prerequisites

The **Plugin Directory** directive of **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `kubernetes-fd.so` plugin file is installed. The standard Bacula plugin directory is `/opt/bacula/plugins`

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Kubernetes Plugin Installation with Package Manager

Installation of the Bacula Enterprise Kubernetes Plugin is most easily done by adding the Kubernetes repository from your Bacula Enterprise subscription to your Linux distribution's package manager configuration.

An example would be `/etc/apt/sources.list.d/bacula.list` for Debian-based Linux distributions. The contents of this file would look as follows:

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@cust@/debs/bin/@ver@/stretch-64/␣
↪bookworm main
deb https://www.baculasystems.com/dl/@cust@/debs/kubernetes/@ver@/stretch-64/␣
↪bookworm kubernetes
```

Where “@cust@” would be your individual download area identification string found in your Welcome Package.

After that, a run of `apt-get update` is needed. Then, the Kubernetes Plugin can be installed using `apt-get install bacula-enterprise-kubernetes-plugin`

On RHEL, extend the repository file for your package manager to contain a section for the plugin - `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@cust@/rpms/bin/@version@/rhel8-64/
enabled=1
protect=0
gpgcheck=1
[Bacula EnterpriseDockerPlugin]
name=Bacula Enterprise Kubernetes Plugin
baseurl=https://www.baculasystems.com/dl/@cust@/rpms/kubernetes/@version@/
↪rhel8-64/
enabled=1
protect=0
gpgcheck=1
```

Then perform a `yum update` and the Kubernetes Plugin package can be installed with `yum install bacula-enterprise-kubernetes-plugin`.

Manual installation of the packages can be performed after downloading the required files from your Bacula Systems download area, and then using the low-level package manager tools (`rpm` or `dpkg`) to perform the plugin installation.

bacula-backup Proxy Pod Image Installation

For Kubernetes PVC Data backup functionality, the Bacula Enterprise Kubernetes Plugin requires a dedicated Bacula Backup Proxy Pod which is deployed using an image that is available in the `bacula-enterprise-kubernetes-tools` package.

This image should be installed manually on your local Docker images registry service which is available on your Kubernetes cluster as a source for application images.

Installation of the image can be performed with the following example commands:

```
# cd /opt/bacula/lib
# docker load -i bacula-backup-<timestamp>.tar
# docker image tag bacula-backup:<timestamp> <registry>/bacula-backup:
↪<timestamp>
# docker push <registry>/bacula-backup:<timestamp>
```

where `<timestamp>` is the image version shipped with above package and `<registry>` is the location of your Docker images registry service. The exact procedure depends on your Kubernetes cluster deployment, so make sure to verify the above before attempting to run the Docker commands.

You can use any registry service available for your cluster, public or private, i.e. `gcr.io/`.

Depending on your cluster configuration it may be necessary to set the **baculaimage=<name>** plugin parameter (see section k8srbackupparameters for details) to define which repository and Container image to use. The default for this parameter is `bacula-backup:<timestamp>` which may not be correct for your deployment.

Another example where you will need to modify the Bacula Backup Proxy Pod Image is in the case where your registry requires authentication. See advancedpoddeployment for more details.

Advanced Bacula Backup Proxy Pod Deployment

Warning: This is an advanced topic related to Kubernetes clusters. It is strongly advised against attempting to modify or configure the Bacula Kubernetes Plugin unless you possess a deep understanding of the process.

You can customize the service parameters used for deploying Bacula backup Pods dedicated to Persistent Volume Claim data backup to suit your needs. The plugin uses the following Pod service deployment YAML template to execute the proxy operation pod on the cluster.

```
apiVersion: v1
kind: Pod
metadata:
  name: {podname}
  namespace: {namespace}
  labels:
    app: {podname}
spec:
  hostname: {podname}
  {nodenameparam}
  containers:
  - name: {podname}
    resources:
      limits:
        cpu: "1"
        memory: "64Mi"
      requests:
        cpu: "100m"
        memory: "16Mi"
    image: {image}
    env:
    - name: PLUGINMODE
      value: "{mode}"
    - name: PLUGINHOST
      value: "{host}"
    - name: PLUGINPORT
      value: "{port}"
    - name: PLUGINTOKEN
      value: "{token}"
    imagePullPolicy: {imagepullpolicy}
    volumeMounts:
      - name: {podname}-storage
        mountPath: /{mode}
  restartPolicy: Never
  volumes:
    - name: {podname}-storage
      persistentVolumeClaim:
```

(continues on next page)

(continued from previous page)

```
        claimName: {pvcname}
```

The above template uses a number of predefined placeholders which will be replaced by corresponding variables during Pod execution preparation. To customize proxy Pod deployment you can change or tune template variables or the template body. Below is a list of all supported variables with short descriptions and requirement conditions.

podname
: This is the predefined Pod name used by a plugin. This variable is required and cannot be customized.

namespace
: This is a Namespace name for which the PVC Data backup is performed. This variable is required and cannot be customized.

nodenameparam
: This is a placeholder for Cluster node name parameter (`nodeName: ...`) used to mount an existing Volume Claim for backup or restore if required for the selected PVC. In most cases this variable is required and cannot be customized.

image
: This is a Pod Container image name to execute. You can customize or omit this variable as long as you provide a Container image name required by the cluster.

mode
: This is an operation mode for the Proxy Pod. The supported values are: `backup` and `restore`. This variable is required and cannot be customized.

host
: This is an endpoint address which corresponds to the `pluginport=...` Kubernetes plugin parameter. This variable is required. You can customize or omit this variable as long as you provide a value for the **PLUGINHOST** Container environment.

port
: This is an endpoint address which corresponds to the `pluginport=...` Kubernetes plugin parameter. This variable is required. You can customize or omit this variable as long as you provide a value for the **PLUGINPORT** Container environment.

token
: This is an Authorization Token (randomly generated). This variable is required and cannot be customized.

pvcname
: This is the name of the PVC for backup or restore operations. This variable is required and cannot be customized.

You can create the required file: `/opt/bacula/scripts/bacula-backup.yaml` or point to the custom one using the `$DEFAULTPODYAML` environment variable.

Configuration

The plugin is configured using **Plugin Parameters** defined in a **Filesets** -> **Include** section of the Bacula Enterprise Director configuration. Starting from Bacula Enterprise version **16.0.7**, it is now possible to enclose particular parameters of this plugin within single quotation marks for delimitation purposes.

Generic Plugin Parameters

The following Kubernetes Plugin parameters affect any type of Job (Backup, Estimate, or Restore).

abort_on_error[=<0 or 1>]
: specifies whether or not the plugin should abort execution (and the Bacula Job) if a fatal error occurs during a Backup, Estimate, or Restore operation. If not specified, the default value is 0.

  This parameter is optional.

config=</path/to/file>
: points to the location of the config file which defines how to connect to the Kubernetes cluster. Please check the Kubernetes documentation for more details about the *kubeconfig* file. If this directive is omitted and no other access method is configured then a default config file location will be used - `$HOME/.kube/config`.

  This parameter is optional.

incluster
: if this directive is defined then a standard in-cluster access to the Kubernetes API will be used. This option should be used only when the Bacula File Daemon and Kubernetes Plugin are deployed as a Kubernetes cluster service in a Pod. In this case, Kubernetes itself will provide the required access credentials. This option won't work when executed outside of a Kubernetes cluster and services.

  This parameter is optional.

host=<url-k8s-api>
: defines a Kubernetes API url. This option is used only with **token** parameter described below. If this option is specified, then both parameters are required.

  This parameter is optional.

token=<bearer-token>
: defines a `Bearertoken` used for authorization to the Kubernetes API available at **host=<url-k8s-api>**. This option is used only with **host** parameter described above. You can read more about this type of authentication at: https://swagger.io/docs/specification/authentication/bearer-authentication/

  This parameter is optional.

verify_ssl[=<0 or 1>]
: specifies whether or not the plugin should verify a Kubernetes API certificate when the connection uses SSL/TLS. If set to **verify_ssl=0** then verification will be disabled. This is useful when connecting to a Kubernetes API server with a self-signed certificate. The default behavior if this parameter if not set is to perform proper certificate validation.

  This parameter is optional.

ssl_ca_cert=</path/to/file>
: specifies a file with the CA certificate used to customize the Kubernetes API server identity to verify the peer.

  This parameter is optional.

timeout=<seconds>

specifies the number of seconds for various network operations. Examples include: waiting for Bacula Backup Proxy Pod connection or Kubernetes API operations, waiting for Pod execution or removal. The default is 600 seconds. The minimum timeout you may set is 1 second. When an invalid value is set the default will be used.

This parameter is optional.

debug[=1,2,3]

specifies that the plugin backend will generate an execution debug file at location `/bacula/working/backendplugin/`. This file can help with troubleshooting the job execution if something goes wrong. If not defined then no debug file will be generated. From version **16.0.14**, users have the ability to choose from various debug levels, with the highest number encompassing the previous level.

1: Save debug messages of k8s server interactions. **2:** Save communication messages with bacula core except data packages. **3:** Save all debug messages.

This parameter is optional. Normally, the level 2 is enough to open support cases.

Estimate and Backup Plugin Parameters

These plugin parameters are relevant only for Backup and Estimate jobs:

namespace=<name>

specifies a Kubernetes namespace name which you want to backup. Multiple `namespace=<name>` parameters are allowed if you want to backup multiple namespaces. If a namespace with `name` can not be found its backup will be silently ignored. If this parameter is not set, all the objects in all namespaces will be saved. If doing a pvcdata backup, the namespace(s) must be specified, and the persistent volume(s) in the specified namespace(s) are included in the backup.

This parameter is required for pvcdata backup. Otherwise, this parameter is optional.

persistentvolume=<name>

specifies a Kubernetes persistent volume configuration you want to backup. Multiple `persistentvolume=<name>` parameters are allowed if you want to backup multiple volumes. You can use standard shell wildcard pattern matching to select multiple volumes using a single `persistentvolume` parameter. If a persistent volume with `name` can not be found its backup will be silently ignored. If this parameter is not set, all persistent volume configurations will be saved unless **pvconfig=0** is set as described below.

This parameter is optional.

pvconfig=[0|1]

this option is used to disable persistent volume configuration backups when **pvconfig=0** is set. The default is to backup persistent volume configurations.

This parameter is optional.

storageclass=<name>

specifies a Kubernetes Storage Class configuration to be backed up. Multiple `storageclass=<name>` parameters are allowed if you want to backup multiple objects. You can use standard shell wildcard pattern matching to select multiple volumes using a single `storageclass` parameter. If a storage class with `name` can not be found, its backup will be silently ignored. If this parameter is not set, all storage class configuration will be saved unless **scconfig=0** is set as described below.

This parameter is optional.

scconfig=[0|1]
: this option is used to disable Storage Class configuration backups when **scconfig=0** is set. The default is to backup Storage Class object configurations.

  This parameter is optional.

pvcdata[=<pvcname1>,<pvcname2>]
: specifies a Kubernetes Persistent Volume Claim names you want to make PVC Data archive for. As all PVCs are namespaced objects, to use this option you should specify the required namespace(s) with **namespace=<name>** parameter. If you define a simple **pvcdata** parameter without the equals sign ("=") and subsequent value, all detected persistent volume claims in the specified namespace(s) will be selected for the PVC Data archive backup. You can select several Persistent Volume Claim in this parameter.

  This parameter is optional.

backup_mode=[snapshot|clone|standard]
: specifies the general backup mode you want to do backup.

  Note: The **bacula/backup.mode** pod annotation has precendence over this parameter in the defined pvcs in the pod.

If none of the parameters above are specified, then all available Namespaces and Persistent Volume Configurations will be backed up. However, the plugin does not force a PVC Data archive backup in this case.

Backup and Restore Plugin Parameters

These plugin parameters are relevant only for Backup and Restore jobs when the PVC Data archive functionality is used:

fdaddress=<IP or name>
: is the IP address or host name where the Kubernetes Plugin should listen for incoming connections from a Bacula Backup Proxy Pod. This parameter, combined with **fdport=<number>** below will define a socket to listen on. The default is to listen on all available interfaces (`0.0.0.0`) which should be fine in most cases.

  This parameter is optional if **pluginhost=<IP or name>** is defined.

fdport=<number>
: is a port number on which Kubernetes Plugin should listen for incoming connections from a Bacula Backup Proxy Pod. This parameter, combined with **fdaddress=<IP or name>** above will define a socket to listen on. The default is to listen on port `9104`.

  This parameter is optional.

pluginhost=<IP or name>
: is the entry point address where a Bacula Backup Proxy Pod should connect during backup or restore operations. The name should be resolvable on the Kubernetes cluster, otherwise an IP address must be used here. This with **pluginport=<number>** parameter below will define the exact service entry point. The default is to use the value from **fdaddress=<IP or name>** parameter above.

  This parameter is required when **fdaddress=<IP or name>** is not defined. Otherwise it is optional.

pluginport=<number>
: is the port number for service entry point address where the Bacula Backup Proxy Pod should

connect during backup or restore operations. This, combined with the **pluginhost=<IP or name>** parameter above define the exact service entry point. The default is to use the value from the **fdaddress=<IP or name>** parameter above. When neither is defined the default `9104` port number will be used.

This parameter is optional.

fdcertfile=<path>
is the SSL certificate file location for the Kubernetes Plugin endpoint service data connection for a Bacula Backup Proxy Pod. The certificate and key (see **fdkeyfile=<path>** below) files are required for proper Bacula Backup Proxy Pod. You can create and use any valid SSL certificate including a self-signed one. You can even just use the certificate generated during the Kubernetes Plugin installation located at `/opt/bacula/etc/bacula-backup.cert` The default is to use the certificate generated during plugin installation.

This parameter is optional.

fdkeyfile=<path>
is an SSL private key file location for the SSL certificate defined by **fdcertfile=<path>** above. An unencrypted private key must be used for this purpose. The default is to use a private key file created during plugin installation at `/opt/bacula/etc/bacula-backup.key` when **fdcertfile=<path>** above is not defined (the default). Otherwise the plugin will refer to the same certificate file from **fdcertfile=<path>** and use it as a `.pem` Container.

This parameter is optional.

baculaimage=<name>
is a Bacula Backup Proxy Pod Container image registry location for your cluster as described in the image installation chapter **baculatarimage**. In most cases this parameter will consist of your Docker images registry location with the tag of the required image. The default for this parameter is `bacula-backup:<timestamp>` and may not match your cluster configuration.

This parameter is optional.

imagepullpolicy=<Always|Never|ifNotPresent>
sets the Kubernetes pod image pull policy during Bacula Backup Proxy Pod Container deployment. You can configure Kubernetes to **Always** or **Never** pull the image from the repository, or pull it only when not available with **ifNotPresent**. The option value is case insensitive. If this parameter is not defined, the default **ifNotPresent** will be used.

This parameter is optional.

backup_mode=<Standard|Clone|Snapshot>
sets the mode to perform backup of the persistent volume data (pvcdata backup). When you specify this parameter, you are limited with compatibility of the storage using this mode. If the persistent volume is not compatible with the backup mode specified (snapshot or clone), the plugin will try another backup mode in this order: Snapshot, Clone and Standard.

Fileset Examples

In the example below, all Kubernetes Namespaces, Objects and Persistent Volume Configurations will be backed up using the default Kubernetes API access credentials.

```
Fileset {
  Name = FS_Kubernetes_All
  Include {
    Plugin = "kubernetes:"
  }
}
```

In this example, we will backup a single Kubernetes Namespace using the Bearer Token authorization method.

```
Fileset {
  Name = FS_Kubernetes_plugintest
  Include {
    Plugin = "kubernetes: host=http://10.0.0.1/k8s/clusters/test \
        token=kubeconfig-user:cbhssdxq8vv8hrcw8jdxs2 namespace=plugintest"
  }
}
```

The same example as above, but with a Persistent Volume:

```
Fileset {
  Name = FS_Kubernetes_mcache1
  Include {
    Plugin = "kubernetes: host=http://10.0.0.1/k8s/clusters/test \
        token=kubeconfig-user:cbhssdxq8vv8hrcw8jdxs2 \
        namespace=plugintest persistentvolume=myvol"
  }
}
```

This example backs up a single Namespace and all detected PVCs in this Namespace using a defined listening and entry point address and the default connection port:

```
Fileset {
  Name = FS_Kubernetes_test_namespace
  Include {
    Plugin = "kubernetes: namespace=test pvcdata fdaddress=10.0.10.10"
  }
}
```

The same example as above, but using different listening and entry point addresses as may be found when the service is behind a firewall using port forwarding features:

```
Fileset {
  Name = FS_Kubernetes_test_namespace_through_firewall
  Include {
    Plugin = "kubernetes: namespace=test pvcdata=plugin-storage fdaddress=10.
↪0.10.10 \
        pluginhost=backup.example.com pluginport=8080"
```

(continues on next page)

(continued from previous page)

```
  }
}
```

This example shows PVC Data archive backup with the Bacula File Daemon inside a Kubernetes cluster:

```
Fileset {
  Name = FS_Kubernetes_incluster
  Include {
    Plugin = "kubernetes: incluster namespace=test pvcdata \
        pluginhost=backup.bacula.svc.cluster.local"
  }
}
```

The configuration above is designed for use in situations where the Bacula server components are located on-premise and behind a firewall with no external ports allowed in, but must back up data on an external Kubernetes cluster.

Restore Plugin Parameters

During restore, the Kubernetes Plugin will use the same parameters which were set for the backup job and saved in the catalog. During restore, you may change any of the parameters described in chapter generick8spluginparameters and k8srbackupparameters. In addition to the options used for backups, the **outputformat** option can be used during restore. This option specifies the file format when restoring to a local filesystem. You can choose the restore output in `JSON` or `YAML`. If not defined the restored files will be saved in `YAML`.

outputformat: <JSON or YAML>
specifies the file format when restoring to a local filesystem as described above.

This parameter is optional.

Operations

The following article aims at describing Kubernetes Plugin operations.

Backup

The plugin can back up a number of Kubernetes Objects including: Deployments, Pods, Services or Persistent Volume Claims, check the *following chapter* for a complete list.

The backup will create a single (`.yaml`) file for any Kubernetes Object which is saved. For PVC Data backup functionality the Kubernetes Plugin generates a data archive as a single `<pvcname>.tar` archive file. The objects are organized inside the Bacula catalog to facilitate browsing and restore operations. In the Bacula catalog, Kubernetes objects are represented as follows:

- `/@kubernetes/namespaces/<namespace>.yaml` - Namespace definition
- `/@kubernetes/namespaces/<namespace>/<objecttype>/<name>.yaml` - Object definitions in the namespace
- `/@kubernetes/namespaces/<namespace>/persistentvolumeclaim/<pvcname>.tar` - PVC Data backup in the selected namespace

- `/@kubernetes/persistentvolumes/<pvname>.yaml` - Persistent Volume definition
- `/@kubernetes/storageclass/<scname>.yaml` - Storage Class definition

All supported Kubernetes Objects will be saved if no filter options are set. You may limit which objects are saved using filtering options described in the k8sbackupparameters chapter. By default, if no filter options are set, all supported Kubernetes Objects will be saved. To see the Kubernetes Objects that may be filtered, a listing mode is available. This mode is described in the k8sobjectlisting chapter.

Persistent Volume Claim Backup

All Pods in Kubernetes are ephemeral and may be destroyed manually or by operations from controllers. Pods do not store data locally because stored data would be destroyed with a pod's life cycle management, so data is saved on Persistent Volumes using Persistent Volume Claim objects to control Volume Space availability.

This brings a new challenge to data backup. Fortunately most of the challenges found here are similar to standard bare metal or virtualized environments. As with bare metal and virtual machine environments, data stored in databases should be protected with dedicated Bacula Enterprise plugins that take advantage of the database backup routines.

See the appropriate *Bacula Enterprise plugin documentation* for more details on database backups.

On the other hand, most non-database applications store data as simple flat files we can backup as-is without forcing complicated transactions or data consistency procedures. This use case is handled directly with the Kubernetes Plugin using a dedicated Bacula Backup Proxy Pod executed in the cluster.

If the container application is more complex, it is possible to execute commands inside the container to quiesce the application:

- before the volume cloning or snapshot
- after the volume cloning or snapshot
- after the backup of the container.

The execution of a command may cause the backup of the container to be terminated due to an issue with the `run.*.failonerror` annotation. You can find detailed description of this feature here.

A Bacula Backup Proxy Pod is a service executed automatically by the Kubernetes Plugin which manages secure access to Persistent Volume data for the plugin. It is executed on the Kubernetes cluster infrastructure and requires a network connection to the Kubernetes Plugin for data exchange on backup and restore operations. No external cluster service like `NodePort`, `LoadBalancer`, `Ingress` or `Host Based Networking` configuration is required to use this feature.

It is also not required to permanently deploy and run this service on the cluster itself as it is executed on demand. The Bacula Backup Proxy Pod does not consume any valuable compute resources outside of the backup window. You can even operate your Kubernetes backup solution (Bacula Enterprise service with Kubernetes Plugin) directly from your on-premise backup infrastructure to backup a public Kubernetes cluster (it requires a simple port forwarding firewall rule) or use public backup infrastructure to back up on-premise Kubernetes cluster(s). Support for these varied architecture modes is built into the Kubernetes Plugin. It is designed to be a *One-Click* solution for Kubernetes backups.

Starting from Bacula Enterprise version **12.6.0**, you can back up and restore any PVC data including PVCs not attached to any running Kubernetes Pod. This removes a previous limitation in this area.

PVC Backup Modes

To ensure the backup of PVC data, there are three available modes:

- Standard Mode: In this mode, the tar program generates a tar pipe from the original data in the PVC, and then the plugin backs it up. a drawback of this method is that if a file is being written while the tar pipe is generating, it may cause issues with the backup and compromise data integrity. On the positive side, this backup mode is slightly faster than the others.
- Clone Mode: The plugin instructs the Kubernetes cluster to clone a persistent volume, followed by backing up all data from this PVC. However, this backup method lacks data consistency guarantee. The persistent volume CSI driver needs to support Volume Cloning in order to utilize this backup mode.
- Snapshot Mode: The plugin initiates a request to the Kubernetes Cluster for a snapshot of the persistent volume, followed by a request to create a PVC from the snapshot volume. Subsequently, the plugin proceeds to back up all data, ensuring data consistency. To utilize this backup mode, the persistent volume CSI driver must be compatible with Volume Snapshot.

In terms of data consistency, the order is as follows: snapshot, clone and standard. Nevertheless, not all persistent volumes are capable of supporting snapshot technology. It is crucial to verify the compatibility of this technology. In case Snapshot is not supported, the plugin will seamlessly transition to Clone, and if Volume Cloning is not supported, it will then move to Standard.

PVC Backup Flow

The plugin has the ability to incorporate multiple pathways in order to enhance reader comprehension. They are:

1. **Pod annotations:**
    - If the parameter **pvcdata** is defined in fileset and a pod has **bacula** annotations referencing to pvc backup, these pvcs in the pod annotations will be backed up, even if they are not defined in **pvcdata** parameter.
    - The plugin looks for all the pods in the namespace specified in the fileset, if they have pod annotation, the plugin includes the persistent volumes in pod annotation with their respective backup_mode.
    - If a pvc is defined in a pod annotation, the plugin will use the backup mode specified in the pod annotation.
    - If **pvcdata*** parameter is defined in fileset without any pvc specified, the plugin will perform backup of all pvc inside the defined namespaces.
    - If a persistent volume is included in both pod annotation and the Fileset pvcdata option, the pod annotation takes precedence over the Fileset pvcdata option.
2. **Backup mode:**
    - By default, the backup mode is snapshot.
    - The priority of the backup mode defined in the pod annotation takes precedence over the backup mode defined in the fileset. This priority is applicable only to the persistent volume claim (PVC) that is annotated in the pod.
    - In case that the pvc is not compatible with snapshot, the plugin will try to perform the backup with clone mode.

- If the pvc clone does not have any data (the backup result of the pvc clone is empty), the plugin will attempt to perform the backup using the standard mode, as long as the pvc is not compatible with the clone.
- The best option to get consistent data is the snapshot mode.

From Bacula Enterprise version **16.0.14**, the plugin does not perform a backup of pvc data when they are in `Pending` state.

Kubernetes Pod Annotations

This feature allows you to define what volumes (PVC Data) to back up, where and what commands to execute, and how to react to some failures to achieve the best results from data volume backup functionality. You can select which volumes mounted at the Pod you want to backup, the preferred backup mode for the Pod, and what commands you want to execute.

The Pod Annotation feature allows administrators of a Kubernetes Cluster, or any Kubernetes cluster user with enough permissions, to configure a persistent volume to be included in the backup.

The supported annotations are:

bacula/backup.mode:[snapshot|clone|standard]
defines how to handle PVC Data backups. If not defined, the default is **snapshot**.

> **Note:** This type of backup follows the flow described here.

bacula/backup.volumes:<pvcname[,pvcname2…]>
defines what volumes will be backed up from this Pod. Multiple PVC names may be selected as a comma separated list. This annotation is required if you want to backup volumes on this Pod. Any other annotations are optional.

bacula/run.before.job.container.command:<container>/<command>
defines what command to execute on which container of the Pod before the volume cloning (or snapshot) and data backup of this Pod occurs. An asterisk (*) as <`container`> name means to execute <`command`> on all containers.

bacula/run.before.job.failjobonerror:[yes|no]
defines whether or not to fail the job when the exit code of the executed command is not zero. The default is yes.

bacula/run.after.job.container.command:<container>/<command>
defines what command to execute on which container of the Pod after volume cloning(or snapshot) and data backup of this Pod. An asterisk (*) as <`container`> name means to execute <`command`> on all containers.

bacula/run.after.job.failjobonerror:[yes|no]
defines whether or not to fail the job when exit code of the executed command is not zero. The default is no.

bacula/run.after.snapshot.container.command:<container>/<command>
defines what command to execute on which container of the Pod after volume cloning (or snapshot) and just before any data backup. An asterisk (*) as <`container`> name means to execute <`command`> on all containers. Note that the **snapshot** mode will not perform a volume snapshot but rather a volume clone.

bacula/run.after.snapshot.failjobonerror:[yes|no]
defines to fail the job when exit code of the executed command is not zero. The default is no.

Pod annotations is an extension to the current PVC Data backup feature available with the **pvcdata=...** plugin parameter as described in k8sbackupparameters. This is an independent function which may be used together with the functionality described above, especially since both use the same data archive stream handling with Bacula Backup Pod.

All you need to use a new feature is to configure selected Pod annotations and make sure that the backup for a required Kubernetes namespace is properly configured. There is no need for any plugin configuration modifications.

Note: Even if the PVC is not listed in the parameter **pvcdata**, the backup will always be performed if the pod has pod annotations for PVC backup.

Pod Annotations Examples

Here are a few examples demonstrating how to set up Bacula annotations within Kubernetes Pods.

When dealing with Kubernetes deployments, the annotations must be placed in the Pod template's specification, or `.template.spec` field:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
    tier: mysql
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
      annotations:
        bacula/backup.mode: standard
        bacula/backup.volumes: mysql-pv-claim
    spec:
      containers:
        - image: mysql:5.6
          name: mysql
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: changeme
          ports:
            - containerPort: 3306
              name: mysql
          volumeMounts:
```

(continues on next page)

(continued from previous page)

```
            - name: mysql-persistent-storage
              mountPath: /var/lib/mysql
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
            claimName: mysql-pv-claim
```

In the example below you will use a simple Linux command `sync` to synchronize cached writes to persistent storage before volume cloning.

```
apiVersion: v1
kind: Pod
metadata:
   name: app1
   namespace: default
   annotations:
   bacula/backup.mode: clone
   bacula/run.before.job.container.command: "*/sync -f /data1; sync -f /data2"
   bacula/run.before.job.failjobonerror: "no"
   bacula/backup.volumes: "pvc1,pvc2"
spec:
   containers:
   - image: ubuntu:latest
     name: test-container
     volumeMounts:
       - name: pvc1
         mountPath: /data1
       - name: pvc2
         mountPath: /data2
   volumes:
     - name: pvc1
       persistentVolumeClaim:
         claimName: pvc1
     - name: pvc2
       persistentVolumeClaim:
         claimName: pvc2
```

In the example below you will use PostgreSQL's database data files quiesce feature to perform consistent backup with volume cloning.

Note: The final PostgreSQL backup solution requires more configuration and preparation which was skipped in this example to make it clearer.

The initial directive (*run.before.job.container.command*) halts any updates to the database files, while the subsequent instruction (*run.after.snapshot.container.command*) restores normal database functionality once the PVC volume cloning process is completed.

```
apiVersion: v1
kind: Pod
metadata:
  name: postgresql13
```

(continues on next page)

(continued from previous page)

```
  namespace: default
  annotations:
    bacula/backup.mode: standard
    bacula/run.before.job.container.command: "*//bin/startpgsqlbackup.sh"
    bacula/run.after.snapshot.container.command: "*//bin/stoppgsqlbackup.sh"
    bacula/run.after.snapshot.failjobonerror: "yes"
    bacula/backup.volumes: "pgsql"
spec:
   containers:
   - image: postgresql:13
     name: postgresql-server
     volumeMounts:
       - name: pgsql
         mountPath: /var/lib/pgsql
     volumes:
       - name: pgsql
         persistentVolumeClaim:
           claimName: pgsql-volume
```

All PVCs defined with this storage class will perform backups using snapshots as per this storage definition.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-hostpath-sc
provisioner: hostpath.csi.k8s.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: plugintest-persistent-volume-claim-csi
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-hostpath-sc
```

Run Container Command Annotation

All flavors of the Run Container Command parameters are remotely executed using the Kubernetes Pod remote execution API. Every command is prepared to execute with a standard Linux shell `/bin/sh`. This requires that a container image has to have the specified shell available. Using command shell execution gives flexibility to command execution or even allows for preparation of small scripts without additional container image customization.

CSI Volume Features Support

Starting from Bacula Enterprise version **12.6.0**, the support for Kubernetes CSI Volume Cloning functionality was introduced, along with various other features. Subsequently, from Bacula Enterprise version **16.0.8** Kubernetes CSI Volume Snapshot functionality was included. The plugin now automatically detects its ability to perform this functionality starting from this version. As of version **16.0.13**, users have the option to select the backup mode that suits their needs while ensuring compatibility.

Starting from Bacula Enterprise version **16.0.8**, users can use CSI Volume Cloning or CSI Volume Snapshot to acquire a consistent data view of selected Volumes.

Additionally, you can configure remote command execution on a selected Container of the Pod. This command execution can be configured just before or after a Pod backup and just after volume snapshot/clone creation. More details in here.

CSI drivers may or may not have implemented the volume cloning or snapshot functionality. The reference to clone feature is CSI Volume Cloning. The reference to snapshot feature is Volume Snapshots.

The main distinction between CSI Volume Cloning and CSI Volume Snapshot is that Volume Cloning creates an exact duplicate of the specified volume, meanwhile, the Volume Snapshot represents a point-in-time copy of a volume. Therefore, a snapshot-based backup offers greater consistency.

When performing persistent volume backups, our plugin uses the volume clone and snapshot api. Given the higher level of consistency associated with snapshotting, our plugin uses this technique by default as long as it is supported by the persistent volume CSI driver.

Restore

The Kubernetes Plugin provides two targets for restore operations:

Restore to Kubernetes Cluster

To use this restore method, the **where=/** parameter of a Bacula `restore` command is used. You can select any supported Kubernetes Object to restore, or batch restore the whole namespace or even multiple namespaces. If you select a single object to restore it will be restored as is without any dependent objects. In most cases, for (`Config Maps`, `Secrets`, `Services`, etc.) this is fine and restore will always succeed. When you restore compound objects (`Pods`, `Deployments`, `Ingress`, etc.), they won't be ready unless all dependencies are resolved. In this case you should make sure that you select all required objects to restore. Furthermore, it is important to note that certain containers such as Postgresql, MySQL, etc., require pvc data. In the event that these containers do not locate data within pvc, they will generate new ones. From Bacula Enterprise version **16.0.14**, there is no need for concern as the plugin will handle the restore seamlessly.

In Kubernetes, a successful object restore doesn't necessarily result in the service successfully coming online. In some cases further monitoring and investigation will be required. For example:

- *Container image is unavailable.*
- *Volume Claim cannot provision new volume.*
- *Untrusted Image Repository.*
- *Infrastructure limits exceeded, i.e. a number of Pods or Memory allocations.*
- *etc...*

All example cases above must be resolved by the Kubernetes administrator. When all issues are resolved, the object should automatically come online. If not, it may be necessary to repeat a restore to redeploy the Object configuration.

The Kubernetes Plugin does not wait for a Object to come online during restore. It checks the Object creation or replace operation status and reports any errors in the job log. The only exception to this is PVC Data restore, when the Kubernetes Plugin will wait for a successful archive data restore. From Bacula Enterprise version **16.0.14**, this operation is always executed before create compound components (`Pods`, `Deployments`, etc.).

Restore to Local Directory

To use this mode, the **where=/some/path** Bacula `restore` parameter is set to a full path on a server where the Bacula File Daemon and Kubernetes Plugin are installed. If the path does not exist, it will be created by the Kubernetes Plugin. With this restore mode, you can restore any saved Kubernetes Object including PVC Data archive file to a location on disk.

Restore PVC Data

Here we describe functionalities and requirements related to pvcdata restore.

Local Directory Restore

The procedure to restore a PVC Data archive file to a local directory is basically the same as restoring the Kubernetes Object configuration file as described in k8slocaldirrestore. However, output transformation is unavailable and ignored when restoring PVC data. Restore of this data will create a `tar` archive file you can manually inspect and use.

Restore to PVC

This procedure is similar to the one described in PVC Data backup and uses the same Bacula Backup Proxy Pod image. During restore, the plugin uses the same endpoint configuration parameters so it is not necessary to setup it again. If your endpoint parameters have changed you can update them using Bacula plugin restore options modification as in example below:

```
*restore select all done where=/
(...)
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
```

(continues on next page)

(continued from previous page)

```
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : kubernetes: namespace=plugintest pvcdata␣
↪pluginhost=example.com
Plugin Restore Options
config:              *None*              (*None*)
host:                *None*              (*None*)
token:               *None*              (*None*)
verify_ssl:          *None*              (True)
ssl_ca_cert:         *None*              (*None*)
outputformat:        *None*              (RAW)
fdaddress:           *None*              (*FDAddress*)
fdport:              *None*              (9104)
pluginhost:          *None*              (*FDAddress*)
pluginport:          *None*              (9104)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-10): 9
Please enter a value for pluginhost: newbackup.example.com
Plugin Restore Options
config:              *None*              (*None*)
host:                *None*              (*None*)
token:               *None*              (*None*)
verify_ssl:          *None*              (True)
ssl_ca_cert:         *None*              (*None*)
outputformat:        *None*              (RAW)
fdaddress:           *None*              (*FDAddress*)
fdport:              *None*              (9104)
pluginhost:          newbackup.example.com (*FDAddress*)
pluginport:          *None*              (9104)
Use above plugin configuration? (yes/mod/no): yes
```

You can restore all data available from the backup archive for a selected Persistent Volume Claim and all data will be overwritten, ignoring the `Replace` job parameter. Please take note of this behavior, which may change in the future.

Restore Examples

The following articles aims at presenting the examples of restore configuration.

Restore to Kubernetes Cluster Example

To restore Kubernetes objects to a Kubernetes cluster, the administrator should execute the restore command and specify the **where** parameter as in this example:

```
* restore where=/
```

and then set any other required restore plugin parameters for the restore.

```
* restore where=/
...
$ cd /@kubernetes/namespaces/plugintest/configmaps/
cwd is: /@kubernetes/namespaces/plugintest/configmaps/
$ ls
plugintest-configmap.yaml
$ add *
1 file marked.
$ done
Bootstrap records written to /opt/bacula/working/bacula-devel-dir.restore.1.
↪bsr

The Job will require the following (*=>InChanger):
Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

Vol005                    File1                     FileChgr1

Volumes marked with "*" are in the Autochanger.


1 file selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-devel-dir.restore.1.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-devel-fd
Restore Client:  bacula-devel-fd
Storage:         File1
When:            2019-09-30 12:39:13
Catalog:         MyCatalog
Priority:        10
```

(continues on next page)

(continued from previous page)

```
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : kubernetes: config=/home/radekk/.kube/config
Plugin Restore Options
config:             radekk/.kube/config  (*None*)
host:               *None*               (*None*)
token:              *None*               (*None*)
username:           *None*               (*None*)
password:           *None*               (*None*)
verify_ssl:         *None*               (True)
ssl_ca_cert:        *None*               (*None*)
outputformat:       *None*               (RAW)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-8): 1
Please enter a value for config: /root/.kube/config
Plugin Restore Options
config:             /root/.kube/config   (*None*)
host:               *None*               (*None*)
token:              *None*               (*None*)
verify_ssl:         *None*               (True)
ssl_ca_cert:        *None*               (*None*)
outputformat:       *None*               (RAW)
fdaddress:          *None*               (*FDAddress*)
fdport:             *None*               (9104)
pluginhost:         *None*               (*FDAddress*)
pluginport:         *None*               (9104)
```

(continues on next page)

(continued from previous page)

```
Use above plugin configuration? (yes/mod/no): yes
Job queued. JobId=1084
```

The plugin does not wait for Kubernetes Objects to become ready and online in the same way as the `kubectl` or the `oc` commands.

Restore to Local Directory Example

It is possible to restore any Kubernetes Object(s) to file without loading them into a cluster. To do so, the **where** restore option should point to the local directory:

```
* restore where=/tmp/bacula/restores
...
$ cd /@kubernetes/namespaces/
cwd is: /@kubernetes/namespaces/
$ ls
bacula/
cattle-system/
default/
graphite/
ingress/
plugintest/
$ add plugintest
25 files marked.
$ done
Bootstrap records written to /opt/bacula/working/bacula-devel-dir.restore.2.
↪bsr

The Job will require the following (*=>InChanger):
Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

Vol005                    File1                     FileChgr1

Volumes marked with "*" are in the Autochanger.


25 files selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-devel-dir.restore.2.bsr
Where:           /tmp/bacula/restores
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-devel-fd
Restore Client:  bacula-devel-fd
Storage:         File1
When:            2019-09-30 12:58:16
Catalog:         MyCatalog
Priority:        10
```

(continues on next page)

(continued from previous page)

```
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : kubernetes: config=/home/radekk/.kube/config debug=1
Plugin Restore Options
config:              *None*               (*None*)
host:                *None*               (*None*)
token:               *None*               (*None*)
verify_ssl:          *None*               (True)
ssl_ca_cert:         *None*               (*None*)
outputformat:        *None*               (RAW)
fdaddress:           *None*               (*FDAddress*)
fdport:              *None*               (9104)
pluginhost:          *None*               (*FDAddress*)
pluginport:          *None*               (9104)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-8): 8
Please enter a value for outputformat: JSON
Plugin Restore Options
config:              *None*               (*None*)
host:                *None*               (*None*)
token:               *None*               (*None*)
verify_ssl:          *None*               (True)
ssl_ca_cert:         *None*               (*None*)
outputformat:        *None*               (RAW)
fdaddress:           *None*               (*FDAddress*)
fdport:              *None*               (9104)
```

(continues on next page)

(continued from previous page)

```
pluginhost:          *None*              (*FDAddress*)
pluginport:          JSON                (9104)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-devel-dir.restore.2.bsr
Where:           /tmp/bacula/restores
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-devel-fd
Restore Client:  bacula-devel-fd
Storage:         File1
When:            2019-09-30 12:58:16
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (yes/mod/no):
Job queued. JobId=1085
```

Output format conversion at restore time will format all data in a human readable format. You can find an example of this restore below.

```
# cat /tmp/bacula/restores/namespaces/plugintest/plugintest.json
{
    "apiVersion": "v1",
    "kind": "Namespace",
    "metadata": {
        "annotations": {
            "field.cattle.io/projectId": "c-hb9ls:p-bm6cw",
            "lifecycle.cattle.io/create.namespace-auth": "true"
        },
        "cluster_name": null,
        "creation_timestamp": "2019-09-25T16:31:03",
        "deletion_grace_period_seconds": null,
        "deletion_timestamp": null,
        "finalizers": [
        "controller.cattle.io/namespace-auth"
        ],
        "generate_name": null,
        "generation": null,
        "initializers": null,
        "labels": {
            "field.cattle.io/projectId": "p-bm6cw"
        },
        "name": "plugintest",
        "namespace": null,
        "owner_references": null,
        "resource_version": "11622",
        "self_link": "/api/v1/namespaces/plugintest",
        "uid": "dd873930-dfb1-11e9-aad0-022014368e80"
    },
    "spec": {
```

(continues on next page)

(continued from previous page)

```
        "finalizers": [
        "kubernetes"
        ]
    },
    "status": {
        "phase": "Active"
    }
}
```

The supported output transformations are: `JSON` and `YAML`.

Listing

The Bacula Enterprise Kubernetes Plugin supports the “plugin listing” feature of Bacula Enterprise 8.x or newer. This mode allows the plugin to display some useful information about available Kubernetes Objects such as:

- List of Kubernetes Namespaces and their resources.
- List of Kubernetes Persistent Volumes.
- List of Kubernetes Storage Classes Objects.
- List of Kubernetes Cluster Roles Objects.
- List of Kubernetes Cluster Roles Bindings Objects.
- List of Kubernetes VOlume Snapshot Classes Objects.

The feature uses the special **.ls** command with a **plugin=<plugin>** parameter.

The command requires the following parameters to be set:

client=<client>
: A Bacula Client name with the Kubernetes plugin installed.

plugin=<plugin>
: A Plugin name, which would be **kubernetes:** in this case, with optional plugin parameters as described in section generick8spluginparameters.

path=<path>
: An object path to display.

The supported values for a **path=<path>** parameter are:

/
: to display resources types available to list.

/namespaces
: to display a list of Namespaces.

/namespaces/<namespace>/
: to display resources types available to list.

/namespaces/<namespace>/<resources>/
: to display Kubernetes Objects of this Kubernetes Resource.

/persistentvolumes
: to display a list of Persistent Volumes

/storageclasses
to display a list of Storage Class Objects.

/clusterroles
to display a list of Cluster Roles Objects.

/clusterrolebindings
to display a list of Cluster Role Bindings Objects.

/volumesnapshotclasses
to display a list of Volume Snapshot Classes Objects.

To display available Kubernetes Resources, follow the following command example:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/
Connecting to Client kubernetes-fd at localhost:9102
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪clusterroles/
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪clusterrolebindings/
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪namespaces/
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪persistentvolumes/
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪storageclasses/
drwxr-xr-x   1 root     root                          0 2024-10-18 13:56:06  /
↪volumesnapshotclasses/
2000 OK estimate files=6 bytes=0
```

To display the list of all available Kubernetes Namespaces, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/namespaces
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/cattle-provisioning-capi-system
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/cattle-system
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/cattle-ui-plugin-system
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/cert-manager
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/cluster-fleet-local-local-1a3d67d0a899
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/default
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/fleet-default
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/fleet-local
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/harbor-system
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
↪namespaces/kube-node-lease
-rw-r-----   1 root     root                       1024 2024-10-18 13:59:45  /
```

(continues on next page)

(continued from previous page)

```
↪namespaces/kube-public
-rw-r-----   1 root     root                   1024 2024-10-18 13:59:45  /
↪namespaces/kube-system
-rw-r-----   1 root     root                   1024 2024-10-18 13:59:45  /
↪namespaces/local
-rw-r-----   1 root     root                   1024 2024-10-18 13:59:45  /
↪namespaces/local-path-storage
-rw-r-----   1 root     root                   1024 2024-10-18 13:59:45  /
↪namespaces/longhorn-system
[...]
2000 OK estimate files=36 bytes=36,864
```

To display available Kubernetes Resources belong to a namespace, follow the following command example:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/namespaces/default/
Connecting to Client kubernetes-fd at localhost:9102
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/configmaps/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/daemonsets/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/deployments/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/endpoints/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/ingresses/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/limitranges/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/persistentvolumeclaims/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/pods/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/podtemplates/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/replicasets/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/replicationcontrollers/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/resourcequotas/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/rolebindings/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/roles/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/secrets/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/services/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/serviceaccounts/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
```

(continues on next page)

(continued from previous page)

```
↪namespaces/default/statefulsets/
drwxr-xr-x   1 root     root                      0 2024-10-18 22:53:11  /
↪namespaces/default/volumesnapshots/
2000 OK estimate files=19 bytes=0
```

To display the list of available Namespaced Kubernetes Objects, use the next command(or similar, it depends on Namespaced Kubernetes Resource indicated):

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/namespaces/default/pods/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                   1024 2024-10-18 22:56:08  /
↪namespaces/default/pods/web-server-nginx
2000 OK estimate files=1 bytes=1,024
```

To display the list of available Persistent Volume Claims which could be used for PVC Data archive feature selection, you can use the following example command for the `mysql` namespace:

```
*.ls client=bacula-devel-fd plugin="kubernetes:" path=/namespaces/mysql/
↪persitentvolumeclaims/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root             2147483648 2024-10-18 23:12:09  /
↪namespaces/mysql/persistentvolumeclaims/web-server-mysql
2000 OK estimate files=1 bytes=2,147,483,648
```

Note: The volume lists display a Volume Storage size which does not reflect the actual configuration size during backup.

To display the list of all available Persistent Volumes, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/persistentvolumes/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-cd6d3b67-f72b-4451-ae6e-6e9eb5fc9d2e
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-d5fe7d2b-31b2-469d-962f-89ec87fb5c90
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-e712b02c-472e-4fa1-beae-a7dc0db7afbf
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-e8ad52b4-0d9b-4efd-b98f-5334eae34302
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-ed57261f-096e-4b88-b7cb-7d02e896c602
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-f4f9e72c-7e00-4dd3-bc84-9bedc99097d7
-rw-r-----   1 root     root                   1024 2024-10-18 23:15:14  /
↪persistentvolumes/pvc-fa10cbde-56c2-4268-b5ce-4c2afdc7a868
[...]
2000 OK estimate files=32 bytes=32,768
```

To display the list of all defined Storage Class Objects, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/storageclasses/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/ceph-block
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/ceph-bucket
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/ceph-filesystem
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/local-path
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/local-storage
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/longhorn
-rw-r-----   1 root     root                   1024 2024-10-18 23:21:50  /
↪storageclasses/nfs-client
2000 OK estimate files=7 bytes=7,168
```

To display the list of all defined Cluster Roles Objects, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/clusterroles/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/admin
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/calico-node
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/capi-aggregated-manager-role
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/capi-manager-role
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/cattle-fleet-local-system-fleet-agent-role
-rw-r-----   1 root     root                   1024 2024-10-18 23:22:57  /
↪clusterroles/cattle-globalrole-admin
[...]
2000 OK estimate files=164 bytes=167,936
```

To display the list of all defined Cluster Role Bindings Objects, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/clusterrolesbindings/
Connecting to Client kubernetes-fd at localhost:9102
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
↪clusterrolebindings/canal-calico
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
↪clusterrolebindings/canal-flannel
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
↪clusterrolebindings/capi-manager-rolebinding
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
↪clusterrolebindings/cattle-fleet-local-system-fleet-agent-role-binding
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
↪clusterrolebindings/cattle-globalrolebinding-globalrolebinding-nxl4c
-rw-r-----   1 root     root                   1024 2024-10-18 23:23:06  /
```

(continues on next page)

(continued from previous page)

```
↪clusterrolebindings/cattle-globalrolebinding-grb-5jhgc
[...]
2000 OK estimate files=133 bytes=136,192
```

To display the list of all defined Volume Snapshot Classes Objects, the following command example can be used:

```
*.ls plugin=kubernetes: client=kubernetes-fd path=/volumesnapshotclasses/
Connecting to Client kubernetes-fd at localhost:9102
kubernetes: Listing returned an empty result
2000 OK estimate files=0 bytes=0
```

Query Commands

The Bacula Enterprise Kubernetes Plugin supports the "query command". This type of command allows the plugin to display useful information about something beyond its main function, such as:

- User permissions on cluster

The feature uses the special **.query** command on bconsole.

This type of commands require the following parameters to be set:

client=<client>
A Bacula Client name with the Kubernetes plugin installed.

plugin="<plugin>"
A Plugin name, which would be **kubernetes:** in this case, with optional plugin parameters as described in section generick8spluginparameters or k8sbackupparameters. It depends on the function.

parameter=<function>
The functions will be explained in the following sections.

Note: We can change the output to **JSON** format. To do that, we write **json** before function and separated them with **|**, like: **json|<function>**.

user_permissions

This function allows us to know if the indicated user in k8s config file has correct permissions to work with this plugin or if this user misses some permission.

In this command, the more important parameters are:

config=<path>
Path to k8s config file connection.

namespace=<namespace>
We indicate a namespace if we want to know permissions that the user has inside in that namespace. If you don't use this parameter, you only query for cluster permissions.

Note: We can define several **namespace** in the same command

Examples:

Listing 33: **Example of the function without namespace**

```
*.query client=kubernetes-fd plugin="kubernetes:" parameter=user_permissions
Allowed to: Cluster list clusterroles -> True
Allowed to: Cluster get clusterroles -> True
Allowed to: Cluster create clusterroles -> True
Allowed to: Cluster update clusterroles -> True
Allowed to: Cluster list clusterrolebindings -> True
Allowed to: Cluster get clusterrolebindings -> True
Allowed to: Cluster create clusterrolebindings -> True
Allowed to: Cluster list namespaces -> True
Allowed to: Cluster get namespaces -> True
Allowed to: Cluster create namespaces -> True
Allowed to: Cluster list storageclasses -> True
Allowed to: Cluster get storageclasses -> True
Allowed to: Cluster create storageclasses -> True
Allowed to: Cluster list volumesnapshotclasses -> True
Allowed to: Cluster get volumesnapshotclasses -> True
Allowed to: Cluster create volumesnapshotclasses -> True
Summary:
Your user has all necessary permissions
```

Listing 34: **Example of the function with some misses permissions**

```
*.query client=kubernetes-fd plugin="kubernetes:" parameter=user_permissions
Allowed to: Cluster list clusterroles -> False
Allowed to: Cluster get clusterroles -> False
Allowed to: Cluster create clusterroles -> False
Allowed to: Cluster update clusterroles -> False
Allowed to: Cluster list clusterrolebindings -> True
Allowed to: Cluster get clusterrolebindings -> True
Allowed to: Cluster create clusterrolebindings -> True
Allowed to: Cluster list namespaces -> True
Allowed to: Cluster get namespaces -> True
Allowed to: Cluster create namespaces -> True
Allowed to: Cluster list storageclasses -> True
Allowed to: Cluster get storageclasses -> True
Allowed to: Cluster create storageclasses -> True
Allowed to: Cluster list volumesnapshotclasses -> True
Allowed to: Cluster get volumesnapshotclasses -> True
Allowed to: Cluster create volumesnapshotclasses -> True
You must grant your user to the next permissions:
- Scope: `Cluster` Resource: `clusterroles` Action: `list`
- Scope: `Cluster` Resource: `clusterroles` Action: `get`
- Scope: `Cluster` Resource: `clusterroles` Action: `create`
- Scope: `Cluster` Resource: `clusterroles` Action: `update`
```

Listing 35: **Example of the function with a namespace**

```
*.query client=kubernetes-fd plugin="kubernetes: debug=1 namespace=\
↪"namespace-1\"" parameter=user_permissions
```

(continues on next page)

(continued from previous page)

```
Allowed to: Cluster list clusterroles -> True
Allowed to: Cluster get clusterroles -> True
Allowed to: Cluster create clusterroles -> True
Allowed to: Cluster update clusterroles -> True
Allowed to: Cluster list clusterrolebindings -> True
Allowed to: Cluster get clusterrolebindings -> True
Allowed to: Cluster create clusterrolebindings -> True
Allowed to: Cluster list namespaces -> True
Allowed to: Cluster get namespaces -> True
Allowed to: Cluster create namespaces -> True
Allowed to: Cluster list storageclasses -> True
Allowed to: Cluster get storageclasses -> True
Allowed to: Cluster create storageclasses -> True
Allowed to: Cluster list volumesnapshotclasses -> True
Allowed to: Cluster get volumesnapshotclasses -> True
Allowed to: Cluster create volumesnapshotclasses -> True
Allowed to: namespace/namespace-1 list configmaps -> True
Allowed to: namespace/namespace-1 get configmaps -> True
Allowed to: namespace/namespace-1 create configmaps -> True
Allowed to: namespace/namespace-1 update configmaps -> True
Allowed to: namespace/namespace-1 list daemonsets -> True
Allowed to: namespace/namespace-1 get daemonsets -> True
Allowed to: namespace/namespace-1 create daemonsets -> True
Allowed to: namespace/namespace-1 update daemonsets -> True
Allowed to: namespace/namespace-1 list deployments -> True
Allowed to: namespace/namespace-1 get deployments -> True
Allowed to: namespace/namespace-1 create deployments -> True
Allowed to: namespace/namespace-1 update deployments -> True
Allowed to: namespace/namespace-1 list endpoints -> True
Allowed to: namespace/namespace-1 get endpoints -> True
Allowed to: namespace/namespace-1 create endpoints -> True
Allowed to: namespace/namespace-1 list ingresses -> True
Allowed to: namespace/namespace-1 get ingresses -> True
Allowed to: namespace/namespace-1 create ingresses -> True
Allowed to: namespace/namespace-1 list limitranges -> True
Allowed to: namespace/namespace-1 get limitranges -> True
Allowed to: namespace/namespace-1 create limitranges -> True
Allowed to: namespace/namespace-1 list persistentvolumes -> True
Allowed to: namespace/namespace-1 get persistentvolumes -> True
Allowed to: namespace/namespace-1 create persistentvolumes -> True
Allowed to: namespace/namespace-1 list persistentvolumeclaims -> True
Allowed to: namespace/namespace-1 get persistentvolumeclaims -> True
Allowed to: namespace/namespace-1 create persistentvolumeclaims -> True
Allowed to: namespace/namespace-1 list pods -> True
Allowed to: namespace/namespace-1 get pods -> True
Allowed to: namespace/namespace-1 create pods -> True
Allowed to: namespace/namespace-1 delete pods -> True
Allowed to: namespace/namespace-1 list podtemplates -> True
Allowed to: namespace/namespace-1 get podtemplates -> True
Allowed to: namespace/namespace-1 create podtemplates -> True
Allowed to: namespace/namespace-1 list replicasets -> True
Allowed to: namespace/namespace-1 get replicasets -> True
```

(continues on next page)

(continued from previous page)

```
Allowed to: namespace/namespace-1 create replicasets -> True
Allowed to: namespace/namespace-1 update replicasets -> True
Allowed to: namespace/namespace-1 list replicationcontrollers -> True
Allowed to: namespace/namespace-1 get replicationcontrollers -> True
Allowed to: namespace/namespace-1 create replicationcontrollers -> True
Allowed to: namespace/namespace-1 list resourcequotas -> True
Allowed to: namespace/namespace-1 get resourcequotas -> True
Allowed to: namespace/namespace-1 create resourcequotas -> True
Allowed to: namespace/namespace-1 update resourcequotas -> True
Allowed to: namespace/namespace-1 list roles -> True
Allowed to: namespace/namespace-1 get roles -> True
Allowed to: namespace/namespace-1 create roles -> True
Allowed to: namespace/namespace-1 list rolebindings -> True
Allowed to: namespace/namespace-1 get rolebindings -> True
Allowed to: namespace/namespace-1 create rolebindings -> True
Allowed to: namespace/namespace-1 list secrets -> True
Allowed to: namespace/namespace-1 get secrets -> True
Allowed to: namespace/namespace-1 create secrets -> True
Allowed to: namespace/namespace-1 list services -> True
Allowed to: namespace/namespace-1 get services -> True
Allowed to: namespace/namespace-1 create services -> True
Allowed to: namespace/namespace-1 list serviceaccounts -> True
Allowed to: namespace/namespace-1 get serviceaccounts -> True
Allowed to: namespace/namespace-1 create serviceaccounts -> True
Allowed to: namespace/namespace-1 list statefulsets -> True
Allowed to: namespace/namespace-1 get statefulsets -> True
Allowed to: namespace/namespace-1 create statefulsets -> True
Allowed to: namespace/namespace-1 list volumesnapshots -> True
Allowed to: namespace/namespace-1 get volumesnapshots -> True
Allowed to: namespace/namespace-1 create volumesnapshots -> True
Allowed to: namespace/namespace-1 delete volumesnapshots -> True
Summary:
Your user has all necessary permissions
```

Listing 36: **Same command that before but with json output format**

```
*.query client=kubernetes-fd plugin="kubernetes: namespace=\"namespace-1\"" ␣
↪parameter=json|user_permissions
{"cluster": {"clusterrolebindings":
{"create": true, "get": true, "list": true},
"clusterroles": {"create": true,
"get": true, "list": true, "update": true},
"namespaces": {"create": true, "get": true, "list": true},
"storageclasses": {"create": true, "get": true, "list": true},
"volumesnapshotclasses": {"create": true, "get": true, "list": true}},
"namespaced": {"namespace-1": {
"configmaps": {"create": true, "get": true, "list": true, "update": true},
"daemonsets": {"create": true, "get": true, "list": true, "update": true},
"deployments": {"create": true, "get": true, "list": true, "update": true},
"endpoints": {"create": true, "get": true, "list": true},
"ingresses": {"create": true, "get": true, "list": true},
```

(continues on next page)

(continued from previous page)

```
"limitranges": {"create": true, "get": true, "list": true},
"persistentvolumeclaims": {"create": true, "get": true, "list": true},
"persistentvolumes": {"create": true, "get": true, "list": true},
"pods": {"create": true, "delete": true, "get": true, "list": true},
"podtemplates": {"create": true, "get": true, "list": true},
"replicasets": {"create": true, "get": true, "list": true, "update": true},
"replicationcontrollers": {"create": true, "get": true, "list": true},
"resourcequotas": {"create": true, "get": true, "list": true, "update": true},
"rolebindings": {"create": true, "get": true, "list": true},
"roles": {"create": true, "get": true, "list": true},
"secrets": {"create": true, "get": true, "list": true},
"serviceaccounts": {"create": true, "get": true, "list": true},
"services": {"create": true, "get": true, "list": true},
"statefulsets": {"create": true, "get": true, "list": true},
"volumesnapshots": {"create": true, "delete": true, "get": true, "list": true}
}}}
```

Use Cases

The following article presents use cases for the Kubernetes Plugin.

The Kubernetes Plugin InCluster Option

This document aims at explaining how to use the Bacula Kubernetes Plugin, running in a container, to back up the resources in the Kubernetes Cluster.

The Software versions used in this use case:

- Rancher Kubernetes Engine (RKE): *Provider: RKE2, Kubernetes Version: v1.30.4 +rke2r1, Architecture: Amd64*
- Bacula Enterprise 18.0.5: *Bacula Client 18.0.5, and Bacula Kubernetes Plugin 18.0.5*

The Bacula File Daemon and the Kubernetes Plugin Deployment

Dockerfile

Use the following Dockerfile to build a Bacula File Daemon with the MySQL Plugin installed image.

```
# cat Dockerfile
FROM debian:bullseye
# Replace "@@customer-id@@" and "@@bee-version@@" with the customer download␣
↪area and Bacula Enterprise version to use
ARG CUSTOMER_AREA="@@customer-id@@"
ARG BEE_VERSION="@@bee-version@@"
LABEL maintainer="Bacula Systems SA"
LABEL version="${BEE_VERSION}"
LABEL name="Bacula Enterprise Edition Client"
LABEL vendor="BACULA SYSTEMS SA"
```

(continues on next page)

(continued from previous page)

```
LABEL summary="This is a Bacula File Daemon with the Kubernetes plugin"
LABEL description="This image contains a Bacula File Daemon which allows␣
↪connection between this pod resources and the Bacula Director."
# Update image
RUN apt update
RUN apt-get -y install curl
RUN mkdir -p /etc/apt/keyrings
RUN curl -fsSL https://www.baculasystems.com/dl/${CUSTOMER_AREA}/
↪BaculaSystems-Public-Signature-08-2017.asc -o /etc/apt/keyrings/bacula.asc
RUN chmod a+r /etc/apt/keyrings/bacula.asc
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
↪bacula.asc] https://www.baculasystems.com/dl/${CUSTOMER_AREA}/debs/bin/$
↪{BEE_VERSION}/bullseye-64/ bullseye main" > /etc/apt/sources.list.d/Bacula-
↪Enterprise-Edition.list
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
↪bacula.asc] https://www.baculasystems.com/dl/${CUSTOMER_AREA}/debs/
↪kubernetes/${BEE_VERSION}/bullseye-64/ bullseye kubernetes" > /etc/apt/
↪sources.list.d/Bacula-Enterprise-Edition-kubernetes-plugin.list
RUN apt-get update
RUN apt-get install -y bacula-enterprise-client bacula-enterprise-kubernetes-
↪plugin
RUN apt autoremove
# use the bacula-fd.conf file previously configured for a specific Bacula␣
↪Director / configuration
RUN rm /opt/bacula/etc/bacula-fd.conf
COPY bacula-fd.conf /opt/bacula/etc/bacula-fd.conf
# copy the kubeconfig file
COPY config /opt/bacula/etc/config
#  expose bacula-fd port
EXPOSE 9102
USER root
# Start the Bacula File Daemon service
CMD ["/opt/bacula/bin/bacula-fd", "-f"]
```

Build an image using the Dockerfile, and tag/push it to a local registry in the Kubernetes Cluster.

```
docker build \
 -t <image>:<tag> .
```

Important Notes

- It is possible to use any base image, Debian Bullseye is used in this Use Case, but the dependencies and external program versions could change.
- Have the `bacula-fd.conf` file previously configured (the Director the File Daemon will use, for example), or use a Kubernetes ConfigMap for a valid `bacula-fd.conf` file.

For example:

```
# kubectl create configmap bacula-fd-configmap --from-file=/path/to/bacula-fd.
↪conf
```

And, in the bacula-fd deployment definition:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: bacula-fd
  namespace: default
  labels:
    app.kubernetes.io/name: bacula-fd
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: bacula-fd
  template:
    metadata:
      labels:
        app.kubernetes.io/name: bacula-fd
      namespace: default
    spec:
      containers:
        - name: bacula-fd
          imagePullPolicy: Always
          image: <registry>/<image>:<tag>
          ports:
            - containerPort: 9102
              name: bacula-fd
              protocol: TCP
            volumeMounts:
            - name: bacula-fd-configmap-volume
              mountPath: /opt/bacula/etc/bacula-fd.conf
              subPath: bacula-fd.conf
      volumes:
        - name: bacula-fd-configmap-volume
          configMap:
            name: bacula-fd-configmap
```

Create the bacula-fd Deployment in the Kubernetes Cluster

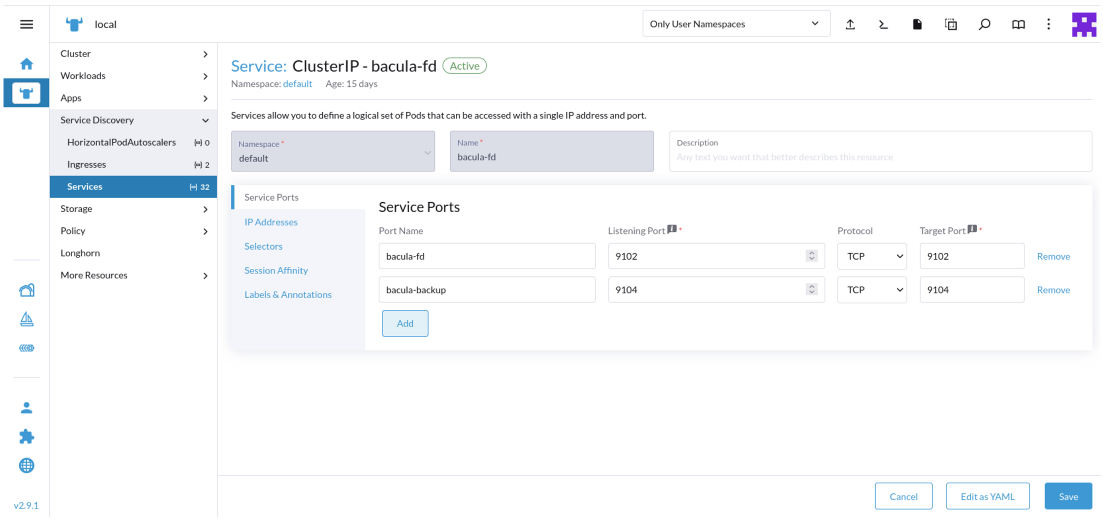
After creating the *bacula-fd* deployment in the Kubernetes Cluster, it is important to provide an external access for the communication between the Bacula File Daemon in the Kubernetes Cluster and both the Director and the Storage Daemon, if their services are running outside the Kubernetes Cluster.

In this use case, the *EXTERNAL-IP* configured for the *bacula-fd* service is the ip address of one of the Kubernetes Master nodes:

```
# kubectl get service -n default
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    ␣
↪          AGE
bacula-fd       ClusterIP   10.43.94.186    10.0.97.201   9102/TCP,
↪9104/TCP   15d
```

An External Load Balancer or Ingress can be used as well.

The Kubernetes Plugin uses both the 9102 port for the File Daemon, and the 9104 port for the bacula-backup proxy pod to backup persistent volume data. Thus, you must add these two ports to the bacula-fd service:



Service: ClusterIP - bacula-fd Active
Namespace: default Age: 15 days

Services allow you to define a logical set of Pods that can be accessed with a single IP address and port.

Namespace *: default
Name *: bacula-fd
Description: Any text you want that better describes this resource

Service Ports

Port Name	Listening Port *	Protocol	Target Port *
bacula-fd	9102	TCP	9102
bacula-backup	9104	TCP	9104

Backup and Restore using the bacula-fd service in the Kubernetes Cluster

Using the Kubernetes Plugin, it is possible to backup all the resources in the Kubernetes cluster, and also include the backup of persistent volumes data.

For details about the Kubernetes Plugin backup configuration, refer to the *Kubernetes Plugin page*.

Backup and Restore of All the Resources in the Kubernetes Cluster

RBAC Configuration

To back up all the resources, in all namespaces and non-namespaced objects, ideally we can create a cluster role to all the Kubernetes cluster resources with the verbs list and get. The admin cluster role rules can be copied, but having the verbs get and list only for the backup purpose.

In addition to the RBAC configuration used to backup all the resources in the Kubernetes cluster, to perform pvcdata backups, our plugin needs to create/delete the snapshot/clone of persistent volumes, and to create/delete the bacula-backup pod to perform pvcdata backups.

This is the `bacula-backup-default ClusterRole` used, rules copied from the `admin ClusterRole`, but only get and list verbs kept:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: bacula-backup-default
rules:
  - apiGroups:
      - cert-manager.io
    resources:
      - certificates
```

(continues on next page)

(continued from previous page)

```
      - certificaterequests
      - issuers
      - challenges
      - orders
    verbs:
      - get
      - list
  - apiGroups:
      - longhorn.io
    resources:
      - volumes
      - engines
      - replicas
      - settings
      - engineimages
      - nodes
      - instancemanagers
      - sharemanagers
      - backingimages
      - backingimagemanagers
      - backingimagedatasources
      - backupbackingimages
      - backuptargets
      - backupvolumes
      - backups
      - recurringjobs
      - orphans
      - snapshots
      - supportbundles
      - systembackups
      - systemrestores
      - volumeattachments
    verbs:
      - get
      - list
  - apiGroups:
      - apps
    resources:
      - controllerrevisions
      - daemonsets
      - deployments
      - replicasets
      - statefulsets
    verbs:
      - get
      - list
  - apiGroups:
      - ''
    resources:
      - namespaces
      - secrets
      - configmaps
```

(continues on next page)

(continued from previous page)

```
      - events
      - replicationcontrollers
      - secrets
      - serviceaccounts
      - services
      - endpoints
      - persistentvolumes
      - bindings
      - events
      - limitranges
      - resourcequotas
    verbs:
      - get
      - list
  - apiGroups:
      - autoscaling
    resources:
      - horizontalpodautoscalers
    verbs:
      - get
      - list
  - apiGroups:
      - batch
    resources:
      - cronjobs
      - jobs
    verbs:
      - get
      - list
  - apiGroups:
      - extensions
    resources:
      - daemonsets
      - deployments
      - ingresses
      - networkpolicies
      - replicasets
    verbs:
      - get
      - list
  - apiGroups:
      - policy
    resources:
      - poddisruptionbudgets
    verbs:
      - get
      - list
  - apiGroups:
      - networking.k8s.io
    resources:
      - ingresses
      - networkpolicies
```

(continues on next page)

(continued from previous page)

```
    verbs:
      - get
      - list
  - apiGroups:
      - coordination.k8s.io
    resources:
      - leases
    verbs:
      - get
      - list
  - apiGroups:
      - metrics.k8s.io
    resources:
      - pods
      - nodes
    verbs:
      - get
      - list
  - apiGroups:
      - discovery.k8s.io
    resources:
      - endpointslices
    verbs:
      - get
      - list
  - apiGroups:
      - authorization.k8s.io
    resources:
      - localsubjectaccessreviews
      - rolebindings
      - roles
    verbs:
      - get
      - list
  - apiGroups:
      - storage.k8s.io
    resources:
      - storageclasses
    verbs:
      - get
      - list
      - patch
  - apiGroups:
      - ''
    resources:
      - persistentvolumeclaims
    verbs:
      - create
      - delete
      - get
      - list
  - apiGroups:
```

(continues on next page)

(continued from previous page)

```
      - ''
    resources:
      - pods
    verbs:
      - create
      - delete
      - get
      - list
  - apiGroups:
      - ''
    resources:
      - persistentvolumeclaims/status
    verbs:
      - get
      - list
  - apiGroups:
      - ''
    resources:
      - pods/status
    verbs:
      - get
      - list
```

Note: The `bacula-backup-default ClusterRole` in this Use Case is an example. You may need a different set of permissions/rules to allow the backup and restore of other resources in your Kubernetes cluster.

See the Kubernetes documentation about using RBAC authorization for more details: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

And the *bacula-backup-default-binding* ClusterRoleBinding:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: bacula-backup-default-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: bacula-backup-default
subjects:
  - kind: ServiceAccount
    name: default
    namespace: default
```

Fileset Configuration

The Fileset configuration used in this use case for the backup of all the resources in the Kubernetes Cluster, and another example to backup the persistent volumes data in the *testing-ns-0010-1* namespace:

```
Fileset {
  Name = "kubernetes-all-resources-incluster-fileset"
  Include {
   Plugin = "kubernetes: incluster"
  }
}
```

```
Fileset {
  Name = "kubernetes-pvcdata-testing-ns-0010-1-incluster-fileset"
  Include {
   Plugin = "kubernetes: incluster pluginhost=bacula-fd.default.svc.
↪cluster.local namespace=testing-ns-0010-1 pvcdata␣
↪baculaimage=harbor.supportlab.lan/library/bacula-backup:30Nov23"
  }
}
```

The value for the fdaddress or the pluginhost options is the FQDN for the bacula-fd service configured to listen in both ports 9102 and 9104:

```
# kubectl get svc -n default
NAME             TYPE         CLUSTER-IP      EXTERNAL-IP    PORT(S)    ␣
↪          AGE
bacula-fd        ClusterIP    10.43.94.186    10.0.97.201    9102/TCP,
↪9104/TCP   14d
```

For more details about pvcdata backup and restore, see Persistent Volume Claim Backup page and Backup and Restore Plugin Parameters page.

Limitations

- Only full level backups are possible. This is a Kubernetes limitation.
- You can perform a single PVC Data backup or restore with a single Bacula File Daemon installation associated with single **fdaddress=<name>**. This limitation may be removed in a future release of the Kubernetes Plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

In this section we describe common problems and ways to solve them.

Error: kubernetes: incluster error: Service host/port is not set

You have set the **incluster** parameter in your Job but you have the following error:

```
Error: kubernetes: incluster error: Service host/port is not set.
```

This means you are running the Bacula File Daemon and Kubernetes plugin not in a Pod, or Kubernetes does not provide default service access in your installation. In the latter case you should use a standard Kubernetes access method in a prepared kubeconfig file.

OpenShift Plugin

Features Summary

- OpenShift cluster resources configuration backup
- Ability to restore single OpenShift configuration resource
- Ability to restore OpenShift resource configuration to local directory
- OpenShift Persistent Volumes data backup and restore
- Ability to restore OpenShift Persistent Volumes data to local directory
- Ability to use new OpenShift CSI driver snapshot features to perform Persistent Volume data backup
- Ability to execute user defined commands on required Pod containers to prepare and clean data backup
- Configure OpenShift workload backup requirements directly with Pod annotations

OpenShift Backup Overview

Containers are very light system level virtualization with less overhead because programs in virtual partitions use the operating system's normal system call interface and do not need to be subjected to emulation or run in an intermediate virtual machine. OpenShift manages a set of containers to create a flexible execution environment for applications and services.

The Bacula Enterprise OpenShift plugin will save all the important Kubernetes resources which make up the application or service. This includes the following namespaced objects:

- Config Map
- Daemon Set
- Deployment
- Endpoint
- Limit Range
- Pod
- Persistent Volume Claim
- Pod Template
- Replica Set
- Replication Controller
- Resource Quota
- Secret
- Service
- Service Account
- Stateful Set

- PVC Data Archive

Note: The PVC Data is not exact Kubernetes Object but represents archive of real data existed on selected PVC.

and non namespaced objects:

- Namespace
- Persistent Volume
- Storage Class

All namespaced objects which belong to the particular namespace are grouped together for easy backup data browsing and recovery.

Users and service accounts can be authorized to access the server API. This process goes through authentication, authorization and admission control. To be able to successfully backup the Kubernetes resources, it is required to have a user or service account with the correct permissions and rights to be successfully authenticated and authorized to access the API server and resources to be backed up.

For resource configuration backups, the user or service account must be able to read and list the resources. In the case of PVC Data backup, it is also required that the user or service account can create and delete pods because the plugin will need to create and delete the Bacula Backup Proxy Pod during the backup.

Please see the OpenShift documentation for more details.

OpenShift Persistent Volume Claim Backup

All Pods in OpenShift are ephemeral and may be destroyed manually or by operations from controllers. Pods do not store data locally because stored data would be destroyed with a pod's life cycle management, so data is saved on Persistent Volumes using Persistent Volume Claim objects to control Volume Space availability.

This brings a new challenge to data backup. Fortunately most of the challenges found here are similar to standard bare metal or virtualized environments. As with bare metal and virtual machine environments, data stored in databases should be protected with dedicated Bacula Enterprise plugins that take advantage of the database backup routines.

Please refer to the appropriate Bacula Enterprise plugin whitepapers for more details on database backups.

On the other hand, most non-database applications store data as simple flat files we can backup as-is without forcing complicated transactions or data consistency procedures. This use case is handled directly with the OpenShift plugin using a dedicated Bacula Backup Proxy Pod executed in the cluster.

If the container application is more complex, it is possible to execute commands inside the container to quiesce the application.

- before the snapshot
- after the snapshot
- after the backup of the container

A problem with command execution can abort the backup of the container with the **run.*.failonerror** annotation. You can find detailed description of this feature at *CSI Snapshot Support*.

A Bacula Backup Proxy Pod is a service executed automatically by the OpenShift plugin which manages secure access to Persistent Volume data for the plugin. It is executed on the OpenShift cluster infrastructure and requires a network connection to the OpenShift plugin for data exchange on backup and restore operations. No external cluster service like `NodePort`, `LoadBalancer`, `Ingress` or `Host Based Networking` configuration is required to use this feature.

It is also not required to permanently deploy and run this service on the cluster itself as it is executed on demand. The Bacula Backup Proxy Pod does not consume any valuable compute resources outside of the backup window. You can even operate your OpenShift backup solution (Bacula Enterprise service with OpenShift plugin) directly from your on-premise backup infrastructure to backup a public Kubernetes cluster (it requires a simple port forwarding firewall rule) or use public backup infrastructure to backup on-premise Kubernetes cluster(s). Support for these varied architecture modes is built into the OpenShift plugin. It is designed to be a *One-Click* solution for OpenShift backups.

Starting from version `1.1.0` of the OpenShift plugin, you can backup and restore any PVC data including PVCs not attached to any running Kubernetes Pod. This removes a previous limitation in this area.

CSI Snapshot Support

Starting from Bacula Enterprise version 12.6 and OpenShift plugin version 2.0, OpenShift CSI Snapshot functionality support together with a bunch of other features was added. Starting from this version you can use CSI Snapshots to acquire a consistent data view of selected Volume. Additionally, you can configure remote command execution on a selected Container of the Pod. You can configure command execution just before or after a Pod backup and just after snapshot creation.

Our plugin uses the volume clone api when doing a volume snapshot. CSI drivers may or may not have implemented the volume cloning functionality.

OpenShift Pod Annotations

The **CSI Snapshot Support** feature described in *CSI Snapshot Support* comes with a configuration of Volume data backup using Pod annotations. This feature allows you to define what volumes (PVC Data) to backup, where and what commands to execute, and how to react to some failures to achieve the best results from data snapshot functionality. You can select which volumes mounted at the Pod you want to backup, the preferred backup mode for the Pod, and what commands you want to execute.

The supported annotations are:

bacula/backup.mode:[snapshot|standard]
: defines how to handle PVC Data backups. If not defined, the default is **snapshot**.

bacula/backup.volumes:<pvcname[,pvcname2. . .]>
: defines what volumes will be backed up from this Pod. Multiple PVC names may be selected as a comma separated list. This annotation is required if you want to backup volumes on this Pod. Any other annotations are optional.

bacula/run.before.job.container.command:<container>/<command>
: defines what command to execute on which container of the Pod before any snapshot and data backup of this Pod occurs. An asterisk (`*`) as <`container`> name means to execute <`command`> on all containers.

bacula/run.before.job.failjobonerror:[yes|no]
: defines whether or not to fail the job when the exit code of the executed command is not zero. The default is yes.

bacula/run.after.job.container.command:<container>/<command>
: defines what command to execute on which container of the Pod after all snapshot and data backup of this Pod. An asterisk (`*`) as <`container`> name means to execute <`command`> on all containers.

bacula/run.after.job.failjobonerror:[yes|no]
defines whether or not to fail the job when exit code of the executed command is not zero. The default is no.

bacula/run.after.snapshot.container.command:<container>/<command>
defines what command to execute on which container of the Pod after all snapshot creations and just before any data backup. An asterisk (*) as <`container`> name means to execute <`command`> on all containers.

bacula/run.after.snapshot.failjobonerror:[yes|no]
defines to fail the job when exit code of the executed command is not zero. The default is no.

Pod annotations is an extension to the current PVC Data backup feature available with the **pvcdata=...** plugin parameter as described in *Estimate and Backup Plugin Parameters*. This is an independent function which may be used together with the functionality described above, especially since both use the same data archive stream handling with Bacula Backup Pod.

All you need to use a new feature is to configure selected Pod annotations and make sure that the backup for a required OpenShift namespace is properly configured. There is no need for any plugin configuration modifications. A Pod's volumes will be backed up automatically.

Examples

Below you can find some examples how to configure Bacula annotations in OpenShift Pods.

In the example below you will use a simple Linux command `sync` to synchronize cached writes to persistent storage before volume snapshot.

```
apiVersion: v1
  kind: Pod
  metadata:
    name: app1
    namespace: default
    annotations:
      bacula/backup.mode: snapshot
      bacula/run.before.job.container.command: "*/sync -f /data1; sync -f /
↪data2"
      bacula/run.before.job.failjobonerror: "no"
      bacula/backup.volumes: "pvc1,  pvc2"
  spec:
    containers:
    - image: ubuntu:latest
      name: test-container
      volumeMounts:
      - name: pvc1
        mountPath: /data1
      - name: pvc2
        mountPath: /data2
      volumes:
      - name: pvc1
        persistentVolumeClaim:
          claimName: pvc1
      - name: pvc2
        persistentVolumeClaim:
          claimName: pvc2
```

In the example below you will use PostgreSQL's database data files quiesce feature to perform consistent backup with snapshot.

Note: The final PostgreSQL backup solution requires more configuration and preparation which was skipped in this example to make it clear

The first command (*run.before.job.container.command*) freezes writes to database files and the second (*run.after.snapshot.container.command*) resumes standard database operation as soon as PVC snapshot becomes ready.

```
apiVersion: v1
kind: Pod
metadata:
  name: postgresql13
  namespace: default
  annotations:
    bacula/backup.mode: standard
    bacula/run.before.job.container.command: "*//bin/startpgsqlbackup.sh"
    bacula/run.after.snapshot.container.command: "*//bin/stoppgsqlbackup.sh"
    bacula/run.after.snapshot.failjobonerror: "yes"
    bacula/backup.volumes: "pgsql"
  spec:
    containers:
    - image: postgresql:13
      name: postgresql-server
      volumeMounts:
      - name: pgsql
        mountPath: /var/lib/pgsql
      volumes:
      - name: pgsql
        persistentVolumeClaim:
          claimName: pgsql-volume
```

Run Container Command annotation

All flavors of the Run Container Command parameters are remotely executed using the OpenShift Pod remote execution API. Every command is prepared to execute with a standard Linux shell `/bin/sh`. This requires that a container image has to have the specified shell available. Using command shell execution gives flexibility to command execution or even allows for preparation of small scripts without additional container image customization.

Installation

Please create the following bacula-fd.yaml pod configuration file to be used for deployment:

```
apiVersion: v1
kind: Pod
metadata:
  name: bacula-fd
  labels:
    app: bacula-fd
spec:
  containers:
  - name: web
    image: registry.connect.redhat.com/bacula-enterprise/bacula-enterprise-
↪openshift-plugin-1260
    env:
      - name: MASTER
        value: "true"
    volumeMounts:
    - name: bacula-conf
      mountPath: /opt/bacula/etc
  volumes:
    - name: bacula-conf
      configMap:
        name: bacula-configmap
```

Once you have the pod configuration file, please use the following procedure to deploy it:

Login to the OpenShift cluster using an account/service account with enough permissions to deploy the pod:

```
oc login -u kubeadmin -p <password> <cluster URI>
```

Create a bacula-fd.conf file that will be used to create the configmap resource for the Bacula File Daemon pod. The bacula-fd.conf file must be in your OpenShift cluster file system. For example:

```
Director {
    Name = "bacula-dir"
    Password = "bacula-dir-fd-password"
}
FileDaemon {
    Name = "openshift-fd"
    Description = "OpenShift Plugin File Daemon"
    PidDirectory = "/opt/bacula/working"
    PluginDirectory = "/opt/bacula/plugins"
    WorkingDirectory = "/opt/bacula/working"
}
Messages {
    Name = "Standard"
    Director  = "bacula-dir" = All,!Skipped,!Restored
}
```

Please modify the above example with the Director name and password to be used in your Bacula Enterprise environment.

Then create the configmap resource:

```
oc create configmap <configmapname> --from-file=/path/to/bacula-fd.conf --
↪from-file=/path/to/cluster/kubeconfig
```

And create the bacula-fd pod:

```
oc apply -f ./bacula-fd.yaml
```

Bacula Backup Proxy Pod Image

For OpenShift PVC Data backup functionality, the Bacula Enterprise **OpenShift** Plugin requires a dedicated Bacula Backup Proxy Pod which is automatically deployed using an image that is available in the `bacula-enterprise-openshift-tools` package.

This image should be installed manually on your local Docker images registry service which is available on your OpenShift cluster as a source for application images.

Installation of the image can be performed with the following example commands:

```
# cd /opt/bacula/lib
# docker load -i bacula-backup-<timestamp>.tar
# docker image tag bacula-backup:<timestamp> <registry>/bacula-backup:
↪<timestamp>
# docker push <registry>/bacula-backup:<timestamp>
```

Where `<timestamp>` is the image version shipped with the above package and `<registry>` is the location of your Docker images registry service. The exact procedure depends on your Kubernetes cluster deployment, so please verify the above steps before attempting to run the docker commands.

You can use any registry service available for your cluster, public or private, i.e. `gcr.io/`.

Depending on your cluster configuration it may be necessary to set the **baculaimage=<name>** plugin parameter (see section *Backup and Restore Plugin Parameters* for details) to define which repository and container image to use. The default for this parameter is `bacula-backup:<timestamp>` which may not be correct for your deployment.

Another example where you will need to modify the Bacula Backup Proxy Pod Image is in the case where your registry requires authentication. Please see the section *Advanced Bacula Backup Proxy Pod Deployment* for more details.

Backup and Restore Operations

Backup

The plugin can backup a number of Kubernetes Resources including: Deployments, Pods, Services or Persistent Volume Claims, check chapter *OpenShift Backup Overview* for a complete list.

The backup will create a single (`.yaml`) file for any Kubernetes Resource which is saved. For PVC Data backup functionality the OpenShift plugin generates a data archive as a single `<pvcname>.tar` archive file. The resources are organized inside the Bacula catalog to facilitate browsing and restore operations. In the Bacula catalog, Kubernetes resources are represented as follows:

- `/@openshift/namespaces/<namespace>.yaml` - Namespace definition

- `/@openshift/namespaces/<namespace>/<resource>/<name>.yaml` - Resource definitions in the namespace
- `/@openshift/namespaces/<namespace>/persistentvolumeclaim/<pvcname>.tar` - PVC Data backup in the selected namespace
- `/@openshift/persistentvolumes/<pvname>.yaml` - Persistent Volume definition
- `/@openshift/storageclass/<scname>.yaml` - Storage Class definition

All supported Kubernetes Resources will be saved if no filter options are set. You may limit which resources are saved using filtering options described in chapter *Estimate and Backup Plugin Parameters*. By default, if no filter options are set, all supported Kubernetes Resources will be saved. To see the Kubernetes Resources that may be filtered, a listing mode is available. This mode is described in chapter *Resource listing*.

Restore

The OpenShift plugin provides two targets for restore operations:

- Restore to an OpenShift cluster
- Restore to a local directory

Restore to an OpenShift Cluster

To use this restore method, the **where=/** parameter of a Bacula `restore` command is used. You can select any supported Kubernetes Resource to restore, or batch restore the whole namespace or even multiple namespaces. If you select a single resource to restore it will be restored as is without any dependent objects. In most cases, for (`Config Maps`, `Secrets`, `Services`, etc.) this is fine and restore will always succeed. On the other hand, compound objects (`Pods`, `Deployments`, etc.) won't be ready unless all dependencies are resolved during the restore. In this case you should make sure that you select all required resources to restore.

In OpenShift, a successful resource restore doesn't necessarily result in the service successfully coming online. In some cases further monitoring and investigation will be required. For example:

- *Container image is unavailable.*
- *Volume Claim cannot provision new volume.*
- *Untrusted Image Repository.*
- *Infrastructure limits exceeded, i.e. a number of Pods or Memory allocations.*
- *etc…*

All example cases above must be resolved by the OpenShift administrator. When all issues are resolved, the resource should automatically come online. If not, it may be necessary to repeat a restore to redeploy the Resource configuration.

The OpenShift plugin does not wait for a Resource to come online during restore. It checks the Resource creation or replace operation status and reports any errors in the job log. The only exception to this is PVC Data restore, when the OpenShift plugin will wait for a successful archive data restore. This operation is always executed at the end of the namespace recovery (when pvcdata is restored with other K8S objects) and should wait for proper PVC mount.

Restore to a Local Directory

To use this mode, the **where=/some/path** Bacula `restore` parameter is set to a full path on a server where the Bacula File Daemon and OpenShift plugin are installed. If the path does not exist, it will be created by the OpenShift plugin. With this restore mode you can restore any saved Kubernetes Resource including PVC Data archive file to a location on disk.

Please note that the OpenShift Client may not have enough space or sufficient permissions to restore in local directories of the bacula-fd pod. Thus It may be required to provide a persistent volume to the bacula-fd pod to perform the restore to a local directory.

Plugin Configuration

The plugin is configured using **Plugin Parameters** defined in a **Filesets** -> **Include** section of the Bacula Enterprise Director configuration.

Generic Plugin Parameters

The following OpenShift plugin parameters affect any type of Job (Backup, Estimate, or Restore).

abort_on_error[=<0 or 1>]
: specifies whether or not the plugin should abort execution (and the Bacula Job) if a fatal error occurs during a Backup, Estimate, or Restore operation. If not specified, the default value is 0.

  This parameter is optional.

config=</path/to/file>
: points to the location of the config file which defines how to connect to the OpenShift cluster. Please check the OpenShift documentation for more details about the *kubeconfig* file. If this directive is omitted and no other access method is configured then a default config file location will be used - `$HOME/.kube/config`.

  This parameter is optional.

incluster
: if this directive is defined then a standard in-cluster access to the OpenShift API will be used. This option should be used only when the Bacula File Daemon and OpenShift plugin are deployed as an OpenShift cluster service in a Pod. In this case, OpenShift itself will provide the required access credentials. This option won't work when executed outside of an OpenShift cluster and services.

  This parameter is optional.

host=<url-openshift-api>
: defines an OpenShift API url. This option is used only with **token** parameter described below. If this option is specified, then both parameters are required.

  This parameter is optional.

token=<bearer-token>
: defines a `Bearertoken` used for authorization to the OpenShift API available at **host=<url-openshift-api>**. This option is used only with **host** parameter described above. You can read more about this type of authentication at: https://swagger.io/docs/specification/authentication/bearer-authentication/

  This parameter is optional.

verify_ssl[=<0 or 1>]
: specifies whether or not the plugin should verify an OpenShift API certificate when the connection uses SSL/TLS. If set to **verify_ssl=0** then verification will be disabled. This is useful when connecting to an OpenShift API server with a self-signed certificate. The default behavior if this parameter if not set is to perform proper certificate validation.

  This parameter is optional.

ssl_ca_cert=</path/to/file>
: specifies a file with the CA certificate used to customize the OpenShift API server identity to verify the peer.

  This parameter is optional.

timeout=<seconds>
: specifies the number of seconds for various network operations. Examples include: waiting for Bacula Backup Proxy Pod connection or OpenShift API operations, waiting for Pod execution or removal. The default is 600 seconds. The minimum timeout you may set is 1 second. When an invalid value is set the default will be used.

  This parameter is optional.

debug[=1]
: specifies that the plugin backend will generate an execution debug file at location `/bacula/working/backendplugin/`. This file can help with troubleshooting the job execution if something goes wrong. If not defined then no debug file will be generated.

  This parameter is optional.

Estimate and Backup Plugin Parameters

These plugin parameters are relevant only for Backup and Estimate jobs:

namespace=<name>
: specifies an OpenShift namespace name which you want to backup. Multiple `namespace=<name>` parameters are allowed if you want to backup multiple namespaces. If a namespace with `name` can not be found its backup will be silently ignored. If this parameter is not set, all namespaces will be saved.

  This parameter is optional.

persistentvolume=<name>
: specifies an OpenShift persistent volume configuration you want to backup. Multiple `persistentvolume=<name>` parameters are allowed if you want to backup multiple volumes. You can use standard shell wildcard pattern matching to select multiple volumes using a single `persistentvolume` parameter. If a persistent volume with `name` can not be found its backup will be silently ignored. If this parameter is not set, all persistent volume configurations will be saved unless **pvconfig=0** is set as described below.

  This parameter is optional.

pvconfig=[0|1]
: this option is used to disable persistent volume configuration backups when **pvconfig=0** is set. The default is to backup persistent volume configurations.

  This parameter is optional.

storageclass=<name>
: specifies an OpenShift Storage Class configuration to be backed up. Multiple `storageclass=<name>` parameters are allowed if you want to backup multiple resources.

You can use standard shell wildcard pattern matching to select multiple volumes using a single `storageclass` parameter. If a storage class with `name` can not be found, its backup will be silently ignored. If this parameter is not set, all storage class configuration will be saved unless **scconfig=0** is set as described below.

This parameter is optional.

scconfig=[0|1]
this option is used to disable Storage Class configuration backups when **scconfig=0** is set. The default is to backup Storage Class resource configurations.

This parameter is optional.

pvcdata[=<pvcname>]
specifies an OpenShift Persistent Volume Claim name you want to make a PVC Data archive for. As all PVCs are namespaced objects, to use this option you should select a required namespace with **namespace=<name>** parameter. If you define a simple **pvcdata** parameter without the equals sign ("=") and subsequent value, all detected persistent volume claims will be selected for the PVC Data archive backup.

This parameter is optional.

If none of the parameters above are specified, then all available Namespaces and Persistent Volume Configurations will be backed up. However, the plugin does not force a PVC Data archive backup in this case.

Backup and Restore Plugin Parameters

These plugin parameters are relevant only for Backup and Restore jobs when the PVC Data archive functionality is used:

fdaddress=<IP or name>
is the IP address or host name where the OpenShift plugin should listen for incoming connections from a Bacula Backup Proxy Pod. This parameter, combined with **fdport=<number>** below will define a socket to listen on. The default is to listen on all available interfaces (`0.0.0.0`) which should be fine in most cases.

This parameter is optional if **pluginhost=<IP or name>** is defined.

fdport=<number>
is a port number on which OpenShift plugin should listen for incoming connections from a Bacula Backup Proxy Pod. This parameter, combined with **fdaddress=<IP or name>** above will define a socket to listen on. The default is to listen on port `9104`.

This parameter is optional.

pluginhost=<IP or name>
is the entry point address where a Bacula Backup Proxy Pod should connect during backup or restore operations. The name should be resolvable on the OpenShift cluster, otherwise an IP address must be used here. This with **pluginport=<number>** parameter below will define the exact service entry point. The default is to use the value from **fdaddress=<IP or name>** parameter above.

This parameter is required when **fdaddress=<IP or name>** is not defined. Otherwise it is optional.

pluginport=<number>
is the port number for service entry point address where the Bacula Backup Proxy Pod should connect during backup or restore operations. This, combined with the **pluginhost=<IP or name>** parameter above define the exact service entry point. The default is to use the value from the

fdaddress=<IP or name> parameter above. When neither is defined the default `9104` port number will be used.

This parameter is optional.

fdcertfile=<path>
is the SSL certificate file location for the OpenShift plugin endpoint service data connection for a Bacula Backup Proxy Pod. The certificate and key (see **fdkeyfile=<path>** below) files are required for proper Bacula Backup Proxy Pod. You can create and use any valid SSL certificate including a self-signed one. You can even just use the certificate generated during the OpenShift plugin installation located at `/opt/bacula/etc/bacula-backup.cert` The default is to use the certificate generated during plugin installation.

This parameter is optional.

fdkeyfile=<path>
is an SSL private key file location for the SSL certificate defined by **fdcertfile=<path>** above. An unencrypted private key must be used for this purpose. The default is to use a private key file created during plugin installation at `/opt/bacula/etc/bacula-backup.key` when **fdcertfile=<path>** above is not defined (the default). Otherwise the plugin will refer to the same certificate file from **fdcertfile=<path>** and use it as a `.pem` Container.

This parameter is optional.

baculaimage=<name>
is a Bacula Backup Proxy Pod Container image registry location for your cluster as described in the image installation chapter *Bacula Backup Proxy Pod Image*. In most cases this parameter will consist of your Docker images registry location with the tag of the required image. The default for this parameter is `bacula-backup:<timestamp>` and may not match your cluster configuration.

This parameter is optional.

imagepullpolicy=<Always|Never|ifNotPresent>
sets the OpenShift pod image pull policy during Bacula Backup Proxy Pod Container deployment. You can configure OpenShift to **Always** or **Never** pull the image from the repository, or pull it only when not available with **ifNotPresent**. The option value is case insensitive. If this parameter is not defined, the default **ifNotPresent** will be used.

This parameter is optional.

Restore Plugin Parameters

During restore, the OpenShift plugin will use the same parameters which were set for the backup job and saved in the catalog. During restore, you may change any of the parameters described in chapter *Generic Plugin Parameters* and *Backup and Restore Plugin Parameters*. In addition to the options used for backups, the **outputformat** option can be used during restore. This option specifies the file format when restoring to a local filesystem. You can choose the restore output in `JSON` or `YAML`. If not defined the restored files will be saved in `YAML`.

outputformat: <JSON or YAML>
specifies the file format when restoring to a local filesystem as described above.

This parameter is optional.

Fileset Examples

In the example below, all OpenShift Namespaces, Resources and Persistent Volume Configurations will be backed up using the default OpenShift API access credentials.

```
Fileset {
  Name = FS_OpenShift_All
  Include {
    Plugin = "openshift:"
  }
}
```

In this example, we will backup a single Namespace using the kube config file authentiaton method.

```
Fileset {
  Name = FS_OpenShift_default_namespace
  Include {
    Plugin = "openshift: namespace=default config=/opt/bacula/etc/kube_cofig"
  }
}
```

The same example as above, but with a Persistent Volume:

```
Fileset {
  Name = FS_OpenShift_default_namespace
  Include {
    Plugin = "openshift: namespace=default config=/opt/bacula/etc/kube_cofig \
              persistentvolume=myvol"
  }
}
```

This example backs up a single Namespace and all detected PVCs in this Namespace using a defined listening and entry point address and the default connection port:

```
Fileset {
  Name = FS_OpenShift_test_namespace
  Include {
    Plugin = "openshift: namespace=test pvcdata fdaddress=10.0.10.10"
  }
}
```

The same example as above, but using different listening and entry point addresses as may be found when the service is behind a firewall using port forwarding features:

```
Fileset {
  Name = FS_OpenShift_test_namespace_through_firewall
  Include {
    Plugin = "openshift: namespace=test pvcdata=plugin-storage fdaddress=10.0.
↪10.10 \
        pluginhost=backup.example.com pluginport=8080"
  }
}
```

The configuration above is designed for use in situations where the Bacula server components are located on-premise and behind a firewall with no external ports allowed in, but must back up data on an external OpenShift cluster.

Restore examples

Restore to OpenShift Cluster

To restore Kubernetes resources to an OpenShift cluster, the administrator should execute the restore command and specify the **where** parameter as in this example:

```
* restore where=/
```

and then set any other required restore plugin parameters for the restore.

```
* restore where=/
...
$ cd /@openshift/namespaces/plugintest/configmaps/
cwd is: /@openshift/namespaces/plugintest/configmaps/
$ ls
plugintest-configmap.yaml
$ add *
1 file marked.
$ done
Bootstrap records written to /opt/bacula/working/bacula-dir.restore.1.bsr

The Job will require the following (*=>InChanger):
Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

Vol005                    File1                     FileChgr1

Volumes marked with "*" are in the Autochanger.


1 file selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-dir.restore.1.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-fd
Restore Client:  bacula-fd
Storage:         File1
When:            2019-09-30 12:39:13
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
```

(continues on next page)

(continued from previous page)

```
    2: Storage
    3: Job
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : openshift: config=/home/test_user/.kube/config
Plugin Restore Options
config:             radekk/.kube/config  (*None*)
host:               *None*               (*None*)
token:              *None*               (*None*)
username:           *None*               (*None*)
password:           *None*               (*None*)
verify_ssl:         *None*               (True)
ssl_ca_cert:        *None*               (*None*)
outputformat:       *None*               (RAW)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-8): 1
Please enter a value for config: /root/.kube/config
Plugin Restore Options
config:             /root/.kube/config   (*None*)
host:               *None*               (*None*)
token:              *None*               (*None*)
verify_ssl:         *None*               (True)
ssl_ca_cert:        *None*               (*None*)
outputformat:       *None*               (RAW)
fdaddress:          *None*               (*FDAddress*)
fdport:             *None*               (9104)
pluginhost:         *None*               (*FDAddress*)
pluginport:         *None*               (9104)
Use above plugin configuration? (yes/mod/no): yes
Job queued. JobId=1084
```

The plugin does not wait for Kubernetes Resources to become ready and online in the same way as the kubectl or the oc commands.

Restore to a Local Directory

It is possible to restore any Kubernetes Resource(s) to file without loading them into a cluster. To do so, the **where** restore option should point to the local directory:

```
* restore where=/tmp/bacula/restores
...
$ cd /@openshift/namespaces/
cwd is: /@openshift/namespaces/
$ ls
bacula/
cattle-system/
default/
graphite/
ingress/
plugintest/
$ add plugintest
25 files marked.
$ done
Bootstrap records written to /opt/bacula/working/bacula-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

Vol005                    File1                     FileChgr1

Volumes marked with "*" are in the Autochanger.


25 files selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-dir.restore.2.bsr
Where:           /tmp/bacula/restores
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-fd
Restore Client:  bacula-fd
Storage:         File1
When:            2019-09-30 12:58:16
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
```

(continues on next page)

(continued from previous page)

```
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : openshift: config=/home/radekk/.kube/config debug=1
Plugin Restore Options
config:              *None*              (*None*)
host:                *None*              (*None*)
token:               *None*              (*None*)
verify_ssl:          *None*              (True)
ssl_ca_cert:         *None*              (*None*)
outputformat:        *None*              (RAW)
fdaddress:           *None*              (*FDAddress*)
fdport:              *None*              (9104)
pluginhost:          *None*              (*FDAddress*)
pluginport:          *None*              (9104)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-8): 8
Please enter a value for outputformat: JSON
Plugin Restore Options
config:              *None*              (*None*)
host:                *None*              (*None*)
token:               *None*              (*None*)
verify_ssl:          *None*              (True)
ssl_ca_cert:         *None*              (*None*)
outputformat:        *None*              (RAW)
fdaddress:           *None*              (*FDAddress*)
fdport:              *None*              (9104)
pluginhost:          *None*              (*FDAddress*)
pluginport:          JSON                (9104)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/bacula-dir.restore.2.bsr
```

(continues on next page)

(continued from previous page)

```
Where:           /tmp/bacula/restores
Replace:         Always
Fileset:         Full Set
Backup Client:   bacula-fd
Restore Client:  bacula-fd
Storage:         File1
When:            2019-09-30 12:58:16
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (yes/mod/no):
Job queued. JobId=1085
```

Output format conversion at restore time will format all data in a human readable format. You can find an example of this restore below.

```
# cat /tmp/bacula/restores/namespaces/plugintest/plugintest.json
{
    "apiVersion": "v1",
    "kind": "Namespace",
    "metadata": {
        "annotations": {
            "field.cattle.io/projectId": "c-hb9ls:p-bm6cw",
            "lifecycle.cattle.io/create.namespace-auth": "true"
        },
        "cluster_name": null,
        "creation_timestamp": "2019-09-25T16:31:03",
        "deletion_grace_period_seconds": null,
        "deletion_timestamp": null,
        "finalizers": [
        "controller.cattle.io/namespace-auth"
        ],
        "generate_name": null,
        "generation": null,
        "initializers": null,
        "labels": {
            "field.cattle.io/projectId": "p-bm6cw"
        },
        "name": "plugintest",
        "namespace": null,
        "owner_references": null,
        "resource_version": "11622",
        "self_link": "/api/v1/namespaces/plugintest",
        "uid": "dd873930-dfb1-11e9-aad0-022014368e80"
    },
    "spec": {
        "finalizers": [
        "kubernetes"
        ]
    },
    "status": {
        "phase": "Active"
```

(continues on next page)

(continued from previous page)

```
    }
}
```

The supported output transformations are: `JSON` and `YAML`.

Restore PVC Data Archive

Here we describe functionalities and requirements related to pvcdata restore.

Local Directory Restore

The procedure to restore a PVC Data archive file to a local directory is basically the same as restoring the Kubernetes Resource configuration file as described in *Restore to a Local Directory*. However, output transformation is unavailable and ignored when restoring PVC data. Restore of this data will create a `tar` archive file you can manually inspect and use.

Restore to PVC

This procedure is similar to the one described in PVC Data backup and uses the same Bacula Backup Proxy Pod image. During restore, the plugin uses the same endpoint configuration parameters so it is not necessary to setup it again. If your endpoint parameters have changed you can update them using Bacula plugin restore options modification as in example below:

```
*restore select all done where=/
(...)
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : openshift: namespace=plugintest pvcdata␣
↪pluginhost=example.com
Plugin Restore Options
config:              *None*              (*None*)
host:                *None*              (*None*)
token:               *None*              (*None*)
verify_ssl:          *None*              (True)
ssl_ca_cert:         *None*              (*None*)
```

(continues on next page)

(continued from previous page)

```
outputformat:        *None*               (RAW)
fdaddress:           *None*               (*FDAddress*)
fdport:              *None*               (9104)
pluginhost:          *None*               (*FDAddress*)
pluginport:          *None*               (9104)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: config (K8S config file)
    2: host (K8S API server URL/Host)
    3: token (K8S Bearertoken)
    4: verify_ssl (K8S API server cert verification)
    5: ssl_ca_cert (Custom CA Certs file to use)
    6: outputformat (Output format when saving to file (JSON, YAML))
    7: fdaddress (The address for listen to incoming backup pod data)
    8: fdport (The port for opening socket for listen)
    9: pluginhost (The endpoint address for backup pod to connect)
    10: pluginport (The endpoint port to connect)
Select parameter to modify (1-10): 9
Please enter a value for pluginhost: newbackup.example.com
Plugin Restore Options
config:              *None*               (*None*)
host:                *None*               (*None*)
token:               *None*               (*None*)
verify_ssl:          *None*               (True)
ssl_ca_cert:         *None*               (*None*)
outputformat:        *None*               (RAW)
fdaddress:           *None*               (*FDAddress*)
fdport:              *None*               (9104)
pluginhost:          newbackup.example.com (*FDAddress*)
pluginport:          *None*               (9104)
Use above plugin configuration? (yes/mod/no): yes
```

You can restore all data available from the backup archive for a selected Persistent Volume Claim and all data will be overwritten, ignoring the `Replace` job parameter. Please take note of this behavior, which may change in the future.

Other

Resource listing

The Bacula Enterprise OpenShift plugin supports the "plugin listing" feature of Bacula Enterprise 8.x or newer. This mode allows the plugin to display some useful information about available Kubernetes resources such as:

- List of OpenShift namespaces
- List of OpenShift Persistent Volumes
- List of OpenShift storage class resources

The feature uses the special **.ls** command with a **plugin=<plugin>** parameter.

The command requires the following parameters to be set:

client=<client>
A Bacula Client name with the OpenShift plugin installed.

plugin=<plugin>
A plugin name, which would be **openshift:** in this case, with optional plugin parameters as described in section *Generic Plugin Parameters*.

path=<path>
An object path to display.

The supported values for a **path=<path>** parameter are:

/
to display Object types available to list

namespaces
to display a list of Namespaces

persistentvolumes
to display a list of Persistent Volumes

storageclass
to display a list of Storage Class Resources

namespaces/<name>/pvcdata
to display all available Persistent Volume Claims available in Namespace <`name`>

To display available Object types, follow the following command example:

```
*.ls plugin=openshift: client=openshift-fd path=/
Connecting to Client openshift-fd at localhost:9102
drwxr-x---   1 root     root 2018-09-28 14:32:20  /namespaces
drwxr-x---   1 root     root 2018-09-28 14:32:20  /persistentvolumes
drwxr-x---   1 root     root 2018-09-28 14:32:20  /storageclass
2000 OK estimate files=2 bytes=0
```

To display the list of all available Kubernetes namespaces, the following command example can be used:

```
*.ls plugin=openshift: client=openshift-fd path=namespaces
Connecting to Client openshift-fd at localhost:9102
drwxr-xr-x   1 root     root 2019-09-25 16:39:56  /namespaces/default
drwxr-xr-x   1 root     root 2019-09-25 16:39:56  /namespaces/kube-public
drwxr-xr-x   1 root     root 2019-09-25 16:39:56  /namespaces/kube-system
drwxr-xr-x   1 root     root 2019-09-25 16:46:19  /namespaces/cattle-system
drwxr-xr-x   1 root     root 2019-09-27 13:04:01  /namespaces/plugintest
2000 OK estimate files=5 bytes=0
```

To display the list of available Persistent Volume Claims which could be used for PVC Data archive feature selection, you can use the following example command for the `mysql` namespace:

```
*.ls client=openshift-fd plugin="openshift:" path=/namespaces/mysql/pvcdata
Connecting to Client openshift-fd at openshift:9102
-rw-r-----   1 root     root 2019-10-16 14:29:38  /namespaces/mysql/pvcdata/
↪mysql-mysql
2000 OK estimate files=1 bytes=0
```

To display the list of all available Persistent Volumes, the following command example can be used:

```
*.ls plugin=openshift: client=openshift-fd path=persistentvolumes
Connecting to Client openshift-fd at localhost:9102
-rw-r-----  1073741824 2019-09-25 /persistentvolumes/pvc-bfaebd0d-dfad-11e9-
↪a2cc-42010a8e0174
-rw-r-----  1073741824 2019-09-25 /persistentvolumes/pvc-b1a49497-dfad-11e9-
↪a2cc-42010a8e0174
-rw-r-----  1073741824 2019-09-25 /persistentvolumes/pvc-949cb638-dfad-11e9-
↪a2cc-42010a8e0174
-rw-r-----  1073741824 2019-09-25 /persistentvolumes/pvc-9313388c-dfad-11e9-
↪a2cc-42010a8e0174
-rw-r----- 10737418240 2019-09-24 /persistentvolumes/myvolume
2000 OK estimate files=5 bytes=15,032,385,536
```

The volume lists display a Volume Storage size which does not reflect the actual configuration size during backup.

To display the list of all defined Storage Class Resources, the following command example can be used:

```
*.ls plugin=openshift: client=openshift-fd path=storageclass
Connecting to Client openshift-fd at openshift:9102
-rw-r----- 1024 2020-07-27 13:39:48  /storageclass/local-storage
-rw-r----- 1024 2020-07-23 16:14:13  /storageclass/default-postgresql-1
-rw-r----- 1024 2020-07-24 11:47:02  /storageclass/local-storage-default
-rw-r----- 1024 2020-07-23 12:00:02  /storageclass/standard
2000 OK estimate files=4 bytes=4,096
```

Advanced Bacula Backup Proxy Pod Deployment

Warning: This is an advanced topic related to OpenShift clusters. You should **NOT** try to implement or customize the Bacula OpenShift Plugin behavior unless you REALLY know what you are doing.*

You can customize the service parameters used for deploying Bacula backup Pods dedicated to Persistent Volume Claim data backup to suit your needs. The plugin uses the following Pod service deployment YAML template to execute the proxy operation pod on the cluster.

```
apiVersion: v1
kind: Pod
metadata:
  name: {podname}
  namespace: {namespace}
  labels:
    app: {podname}
spec:
  hostname: {podname}
  {nodenameparam}
  containers:
  - name: {podname}
    resources:
      limits:
```

(continues on next page)

(continued from previous page)

```
          cpu: "1"
          memory: "64Mi"
        requests:
          cpu: "100m"
          memory: "16Mi"
      image: {image}
      env:
      - name: PLUGINMODE
        value: "{mode}"
      - name: PLUGINHOST
        value: "{host}"
      - name: PLUGINPORT
        value: "{port}"
      - name: PLUGINTOKEN
        value: "{token}"
      imagePullPolicy: {imagepullpolicy}
      volumeMounts:
        - name: {podname}-storage
          mountPath: /{mode}
    restartPolicy: Never
    volumes:
      - name: {podname}-storage
        persistentVolumeClaim:
          claimName: {pvcname}
```

The above template uses a number of predefined placeholders which will be replaced by corresponding variables during Pod execution preparation. To customize proxy Pod deployment you can change or tune template variables or the template body. Below is a list of all supported variables with short descriptions and requirement conditions.

podname
: This is the predefined Pod name used by a plugin. This variable is required and cannot be customized.

namespace
: This is a Namespace name for which the PVC Data backup is performed. This variable is required and cannot be customized.

nodenameparam
: This is a placeholder for Cluster node name parameter (`nodeName: ...`) used to mount an existing Volume Claim for backup or restore if required for the selected PVC. In most cases this variable is required and cannot be customized.

image
: This is a Pod Container image name to execute. You can customize or omit this variable as long as you provide a Container image name required by the cluster.

mode
: This is an operation mode for the Proxy Pod. The supported values are: `backup` and `restore`. This variable is required and cannot be customized.

host
: This is an endpoint address which corresponds to the `pluginport=...` OpenShift plugin parameter. This variable is required. You can customize or omit this variable as long as you provide a value for the **PLUGINHOST** Container environment.

port

This is an endpoint address which corresponds to the `pluginport=...` Kubernetes plugin parameter. This variable is required. You can customize or omit this variable as long as you provide a value for the **PLUGINPORT** Container environment.

token

This is an Authorization Token (randomly generated). This variable is required and cannot be customized.

pvcname

This is the name of the PVC for backup or restore operations. This variable is required and cannot be customized.

You can create the required file: `/opt/bacula/scripts/bacula-backup.yaml` or point to the custom one using the `$DEFAULTPODYAML` environment variable.

Limitations

- Only full level backups are possible. This is an OpenShift limitation.
- You can perform a single PVC Data backup or restore with a single Bacula File Daemon installation associated with single **fdaddress=<name>**. This limitation may be removed in a future release of the OpenShift plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Common Problems

In this section we describe the common problems you may encounter when you first deploy the Bacula OpenShift plugin or when you are not very familiar with this plugin.

- You have set the **incluster** parameter in your Job but you have the following error:

  ```
  Error: openshift: incluster error: Service host/port is not set.
  ```

  This means you are running the Bacula File Daemon and OpenShift plugin not in a Pod, or Kubernetes does not provide default service access in your installation. In the latter case you should use a standard OpenShift access method in a prepared kubeconfig file.

Docker Plugin

Features Summary

- Docker container image-based backup
- Docker system images backup
- Docker volumes files data backup
- Ability to restore single Docker container, image and volume
- Ability to restore Docker container as a new Docker system image
- Ability to create or run restored Docker container
- Ability to restore image, container or volume archives to local directory

Backup with Docker Plugin

Containers are very light system level virtualization with less overhead because programs in virtual partitions use the operating system's normal system call interface and do not need to be subjected to emulation or be run in an intermediate virtual machine. Especially Docker containers rely on sophisticated fs-level data abstraction with a number of read-only images which creates a template used for container initialization. When initialized a container include a writable layer where all file modifications are stored.

With Docker Plugin, the Bacula Enterprise will save the full container image including all read-only and writable layers into a single image archive. It is possible to backup a defined Docker images only which will be used to create new containers when required.

It is not needed to install a Bacula File daemon in each container, so you can backup containers based on common image repository. The Bacula Docker Plugin will contact the Docker service to read and save the contents of any system image or container image using snapshots (default behavior) and dump them using the Docker API.

Bacula does not need to walk through the container file-system to open, read, close and `stat` files, so it consumes less resources on the Docker infrastructure than a standard file level backup.

Docker Persistent Volume Backup

All Containers as seen in Docker are ephemeral. You can execute a container with auto destroy option (**—-rm**) which will remove all changes made in docker images during runtime by default. This feature is very useful when you want to execute your computations in never changing environment but is useless when computation result should be preserved instead of removed. In this case you will use a Docker Persistent Volumes.

This brings a new challenge to data backup solution. Fortunate most of the challenges we can meet here are almost the same as we have in standard bare metal or virtualized environments. All data stored in different kind of databases should be saved with dedicated Bacula Enterprise Plugins exploiting network backup data dump.

You should refer to appropriate Bacula Enterprise Plugin Whitepapers for more details.

On the other hand most non-database applications stores its data as a simple flat files we can backup as-is without forcing complicated transactions or data consistency procedures. This case is handled directly with Docker Plugin using a dedicated Bacula Archive container executed from a prepared image.

Installation

The Bacula File Daemon and its Docker Plugin need to be installed on the host for the Docker service. Docker could be installed on different operating systems and distributions, so the Bacula Enterprise File Daemon for this operating system and platform has to be used. You can backup or restore Docker objects on remote host using **docker_host=...** plugin parameter, check **genericparameters**.

Installation of the Bacula Enterprise Docker Plugin is most easily done by adding the repository file suitable for the existing subscription and the distributions package manager configuration. An example would be `/etc/apt/sources.list.d/bacula.list` for deb based Linux distributions with the following content:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪stretch-64/ stretch main
deb https://www.baculasystems.com/dl/@customer-string@/debs/docker/@version@/
↪stretch-64/ stretch docker
```

After that, a run of `apt-get update` is needed. Then, the Plugin can be installed using `apt-get install bacula-enterprise-docker-plugin`

On RHEL, extend the repository file for your package manager to contain a section for the plugin - `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer@/rpms/bin/@version@/rhel7-
↪64/
enabled=1
protect=0
gpgcheck=0
[Bacula EnterpriseDockerPlugin]
```

(continues on next page)

(continued from previous page)

```
name=Bacula Enterprise Docker Plugin
baseurl=https://www.baculasystems.com/dl/@customer@/rpms/docker/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

Then perform a `yum update` and after that the package `bacula-enterprise-docker-plugin` can be installed with `yum install`.

The package `bacula-enterprise-docker-plugin` must be installed for Docker volume backup. Please see *Bacula Archive docker image* section for more details.

Manual installation of the packages, can be done after downloading the right files from the Bacula Systems provided download area, and then using the low-level package manager (`rpm` or `dpkg`) to do the plugin installation.

File Daemon Configuration

The **Plugin Directory** directive of the resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `docker-fd.so` plugin file is installed. The standard Bacula plugin directory is `/opt/bacula/plugins`

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Bacula Archive docker image

For proper Docker volume backup Bacula Enterprise Docker Plugin requires a dedicated Bacula Archive Container image that is available in the bacula-enterprise-docker-tools package.

The docker image should be installed automatically, but if not, it is possible to load manually the `baculatar` to local Docker installation with the following command:

```
# cd /opt/bacula/lib
# docker load -i baculatar-docker-07Dec22.tar.gz
```

Backup and Restore Operations

Backup

Plugin can backup a three distinct Docker objects: *Docker Container*, *Image* and *Persistent Volume* files data.

The backup of a single container consists of the following steps:

1. Save current container state to new image (container commit - snapshot).

2. Execute `docker` utility and save data.
3. Remove saved snapshot to free not needed resources.

The backup of a docker system images does not make a snapshot as every system image is a read-only template used for container creation.

The Persistent Volume files data backup is performed with a dedicated helper container and consist of the following steps:

1. Prepare a local log archive Docker Volume.
2. Execute a dedicated Bacula Archive container which will access *Docker Persistent Volume* data files for backup.
3. Save Volume data files stream from container.
4. Terminate Bacula Archive container when done.

Backups can be performed for container in any state (created, running or stopped).

The Docker Plugin will inform you about every container or image backup start and finish:

```
JobId 127: docker: Start Backup Docker Container: myubuntu (4d0a4fadb50d)
JobId 127: dkcommctx: Commit created: myubuntu/4d0a4fadb50d/127:backup
JobId 127: dkcommctx: Commit removed: myubuntu/4d0a4fadb50d/127:backup
JobId 127: docker: Backup of Docker Container: myubuntu (4d0a4fadb50d) OK.
...
```

The backup will create a single (`.tar`) file for any container, image or Docker volume which is saved. Inside Bacula, those are represented as follows.

- `/@docker/container/<name-label>/<id>.tar` for container backup
- `/@docker/container/<name-label>/volume: <name> -> <mountpoint>` during volume backup for this container
- `/@docker/image/<repository:tag>/<id>.tar` for image backup
- `/@docker/volume/<name>.tar` for volume backup

Multiple files will be created during a backup if multiple containers images or volumes are selected for backed up with one job. The distinct file names as shown above allow to locate the proper container, system image or volume archive for restore. You can list a special support file (symbolic link) generated at container backup directory tree which represent a volume name and mount point for this volume for easy data recovery.

To list available Docker containers, system images or defined volumes, a listing mode is available, described in chapter **listing**.

Restore

The Docker Plugin provides two targets for restore operations:

- Restore to Docker service
- Restore to a local directory as archive files

Restore to Docker

To use this restore method, the **where=** or parameter of a Bacula `restore` command is used.

The Docker container archive will be sent to the Docker service and restored as a new image and then created as container if the `container_create` restore parameter is set (this is a default). If the restore parameter `container_create` is not set then any container will be restored to image level only and user has to create or run the container manually. If restore parameter `container_run` is set (default is no) then restored container will be started immediate after successful restore.

The Docker image archive will be loaded into the Docker service as the original docker image overwriting the one already exist. This is a default behavior and cannot be changed. You can skip docker image restore of already existent image with **Replace** option of `restore` command.

Docker volume restore will extract all files into the same Docker volume. If volume is not defined in Docker then a default local volume will be created. Volume data files restore will always overwrite restored data. This is a current limitation of the plugin and will be removed in the future.

You can change default container name or container image label during restore with `container_defaultnames` or `container_imageid` restore parameters (see `container_defaultnames: <Yes|No>` or `container_imageid: <Yes|No>`).

Restore To Local Directory

To use this mode, the **where=/some/path** Bacula `restore` parameter is set to a full path on the server where the Docker Plugin is installed. If the path does not exist, it will be created by the Bacula Docker Plugin. With this restore mode you can restore any saved Docker object including containers, images and volumes.

Plugin Configuration

The plugin is configured using **Plugin Parameters** defined in a Filesets **Include** section of the Bacula Enterprise Director configuration.

Generic Plugin Parameters

The following Docker plugin parameter affects any type of Job (Backup, Estimation, or Restore).

abort_on_error[=<0 or 1>]
: specifies whether or not the plugin should abort execution (and the Bacula Job) if a fatal error occurs during a Backup, Estimation, or Restore operation. The default value is 0.

  This parameter is optional.

docker_host=tcp://<host:port>
: if specified then all operations will be executed using **<host:port>** endpoint. The default is to use a local default socket for local `dockerd`. This parameter can be set or overloaded for restore operations using restore plugin options. Single **docker_host=...** option is supported. If multiple defined the first one will be used.

  When this parameter is set then plugin will disable volume backup or restore functionality as this feature is not supported with remote Docker. This is a current plugin limitation and will be removed in the future.

  This parameter is optional.

Estimation and Backup Plugin Parameters

These plugin parameters are relevant only for Backup and Estimation jobs:

container=<name-label>|<id>
: specifies a Docker container to backup. Multiple `container=...` parameters are allowed. If container with `name-label` can not be found, then a single job error will be generated and the backup will proceed to the next container unless `abort_on_error` is set which will cause the backup job to be aborted. You can use Docker container names (`name-label`) or container id () to select required container to backup.

  This parameter is optional.

image=<repository:tag>|<id>
: specifies a Docker image to backup. Multiple `image=...` parameters are allowed. If image with `repository:tag` can not be found, then a single job error will be generated and the backup will proceed to the next image unless `abort_on_error` is set which will cause the backup job to be aborted. You can use Docker image repository and tag names (`repository:tag`) or image id () to select required image to backup. This parameter is optional.

include_container=<name-label-regex>
: specifies a list of Docker container names to backup using regular expression syntax. All containers with names matching the regular expression provided will be selected for backup. Multiple **include_container=...** parameters may be provided.

  If no containers are matching the name expression provided, the backup will proceed to the next parameter or finish successfully without backing up any containers. The **abort_on_error** parameter will not abort the job when no containers are found using name matching. This parameter is optional.

include_image=<repository-regex>
: specifies a list of Docker image repository names to backup using regular expression syntax. All images with repository name matching the regular expression provided will be selected for backup. Multiple **include_images=...** parameters may be provided.

  If no images are matching the repository name expression provided, the backup will proceed to the next parameter or finish successfully without backing up any images. The **abort_on_error** parameter will not abort the job when no images are found using repository name matching. This parameter is optional.

exclude_container=<name-label-regex>
: specifies a list of a Docker container names which will be excluded from backup using regular expression matching. All containers with names matching the provided regular expression, and selected for backup using the **include_container=...** parameter will be excluded. This parameter does not affect any containers selected to be backed up using **container=...** parameter. Multiple **exclude_container=...** parameters may be provided. This parameter is optional.

exclude_image=<repository-regex>
: specifies a list of a Docker image repository names which will be excluded from backup using regular expression matching. All images with repository names matching the provided regular expression, and selected for backup using the **include_image=...** parameter will be excluded. This parameter does not affect any images selected to be backed up using **image=...** parameter. Multiple **exclude_image=...** parameters may be provided. This parameter is optional.

volume=<name>
: specifies the Docker volume to backup. Multiple `volume=...` parameters are allowed. If volume with `name` can not be found, then a single job error will be generated and the backup will proceed to the next volume unless `abort_on_error` is set which will cause the backup job to be aborted.

This parameter is optional.

allvolumes
specifies the automatic Docker volumes selection for backup for every container selected. You should configure a single **allvolumes** parameter if required.

This parameter is optional.

mode=<pause|nopause>
specifies the default backup mode for containers to use. The **pause** mode will generate backup using container pause commit which quiesce I/O operations during backup and allow to achieve a consistent backup with some possible downtime. The container will remain suspended during backup commit phase and will resume once the commit operation finishes. The **nopause** mode will commit container without quiesce I/O operations. This parameter is optional. If not set then **pause** mode will be used by default.

timeout=nn
specifies the default timeout for connecting to Bacula Archive container during volume backup. The default is 30 seconds if parameter is not specified. The value **<nn>** specifies a number of seconds how long plugin will wait. The value zero (0) is invalid.

This parameter is optional.

If none of the parameters **container=...**, **image=...**, **include_container=...** , **include_image=...**, and **exclude=...** are specified, all available containers and images on Docker will be backed up.

Plugin Restore Parameters

During restore, the Docker plugin will use the same parameters which were set for the backup job and saved in the catalog. Some of them may be changed during the restore process if required.

container_create: <Yes|No>
specifies if Docker container restore should automatically create container. The default option is to create container on restore. If you want to restore container as an image only then you should set this parameter to `No`.

This parameter is optional.

container_run: <Yes|No>
specifies if Docker container restore should automatically create and run container. The default option is to not run container on restore. If you want to automatically run container on restore then you should set this parameter to `Yes`. If you set this parameter to `Yes` then **container_create** parameter will be ignored.

This parameter is optional.

container_imageid: <Yes|No>
specifies if Docker Plugin should use image id value to create or run restored container. The default is to use image repository and tag values for that. This parameter will be ignored when both **container_create** and **container_run** will be set to `No` as no container create or run operation will be performed.

This parameter is optional.

container_defaultnames: <Yes|No>
specifies if Docker Plugin should setup a container names based on original container name and JobId values which is a default. If parameter is set to `Yes` then Docker service will setup default names for created or run container.

This parameter is optional.

timeout: nn

specifies the default timeout for connecting to Bacula Archive container during volume restore. The default is 30 seconds. The value zero (0) is invalid.

This parameter is optional.

docker_host: tcp://<host:port>

check **genericparameters** for more info.

Fileset Examples

In the example below, all Docker containers, images and volumes will be backed up.

```
Fileset {
  Name = FS_DockerAll
  Include {
    Plugin = "docker:"
  }
}
```

In this example, a single Docker container with name-label of "mcache1" will be backed up.

```
Fileset {
  Name = FS_Docker_mcache1
  Include {
    Plugin = "docker: container=mcache1"
  }
}
```

The same example as above, but using container id instead:

```
Fileset {
  Name = FS_Docker_mcache1
  Include {
    Plugin = "docker: container=cd77eb89e59a"
  }
}
```

In the following example, all Docker containers which contain "ngnix" in their names will be backed up.

```
Fileset {
  Name = FS_Docker_nginixAll
  Include {
    Plugin = "docker: include_container=ngnix"
  }
}
```

In this final example, all Docker containers except whose names begins with "test" will be backed up.

```
Fileset {
  Name = FS_Docker_AllbutTest
  Include {
    Plugin = "docker: include_container=.* exclude_container=^test"
```

(continues on next page)

(continued from previous page)

```
  }
}
```

In this final example, a single Docker volume will be backed up.

```
Fileset {
  Name = FS_Docker_Volume
  Include {
    Plugin = "docker: volume=myvolume"
  }
}
```

In this final example, a single Docker container with all mounted volumes will be backed up.

```
Fileset {
  Name = FS_Docker_Container_VolumeAll
  Include {
    Plugin = "docker: container=mycontainer allvolumes"
  }
}
```

In the example below, all Docker objects will be backed up using remote docker host.

```
Fileset {
  Name = FS_RemoteDockerAll
  Include {
   Plugin = "docker: docker_host=tcp://10.0.0.1:2376"
  }
}
```

Restore examples

Restore to a Docker service

To restore a container or image to a Docker service, the administrator should execute the restore command and specify the **where** parameter as in this example:

```
* restore where=/
```

and then set any other required restore plugin parameters for the restore.

In the following restore session example, the **container_run** plugin restore option is set to "Yes":

```
* restore where=/
...
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/docker-test-dir.restore.1.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
```

(continues on next page)

(continued from previous page)

```
Backup Client:   docker-test-fd
Restore Client:  docker-test-fd
Storage:         File1
When:            2018-09-28 14:09:30
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
    1: Level
    2: Storage
    3: Job
    4: Fileset
    5: Restore Client
    6: When
    7: Priority
    8: Bootstrap
    9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : docker: container=mcache1 abort_on_error
Plugin Restore Options
container_create:    *None*               (*Yes*)
container_run:       *None*               (*No*)
container_imageid:   *None*               (*No*)
container_defaultnames: *None*               (*No*)
docker_host:         *None*               (*local*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
    1: container_create (Create container on restore)
    2: container_run (Run container on restore)
    3: container_imageid (Use Image Id for container creation/start)
    4: container_defaultnames (Use default docker Names on container creation)
    5: docker_host (Use defined docker host to restore)
Select parameter to modify (1-4): 2
Please enter a value for container_run: yes
Plugin Restore Options
container_create:    *None*               (*Yes*)
container_run:       yes                  (*No*)
container_imageid:   *None*               (*No*)
container_defaultnames: *None*               (*No*)
docker_host:         *None*               (*local*)
Use above plugin configuration? (yes/mod/no): yes
```

The restore job log will indicate which container is restored and which new container id was created:

```
JobId 139: Start Restore Job RestoreFiles.2018-09-28_14.13.31_03
JobId 139: Using Device "FileChgr1-Dev1" to read.
JobId 139: Ready to read from volume "vol001" on File device "FileChgr1-Dev1"␣
```

(continues on next page)

(continued from previous page)

```
↪(/opt/bacula/archive).
JobId 139: Forward spacing Volume "vol001" to addr=225
JobId 139: docker: Docker Container restore: mcache1/b97d4dd88063
JobId 139: End of Volume "vol001" at addr=62177197 on device "FileChgr1-Dev1"␣
↪(/opt/bacula/archive).
JobId 139: dkcommctx: Successfully run container as: ef48c6b5b867
```

The new container created during the restore will get a new container id. You can check it running with a following command:

```
# docker ps -a
CONTAINER ID        IMAGE                              CREATED                  ␣
↪STATUS
ef48c6b5b867        mcache1/b97d4dd88063/139:restore   4 minutes ago       Up␣
↪4 minutes
```

Restore to Local Directory

It is possible to restore the container images, Docker images and volume archives to a file without loading them into Docker service. To do so, the **where** restore option should point to the local directory:

```
* restore where=/tmp/bacula/restores
```

Please check the following example for the message “Docker local restore”:

```
JobId 141: Start Restore Job RestoreFiles.2018-09-28_14.26.34_03
JobId 141: Using Device "FileChgr1-Dev1" to read.
JobId 141: Ready to read from volume "vol001" on File device "FileChgr1-Dev1"␣
↪(/opt/bacula/archive).
JobId 141: docker: Docker local restore: container/mcache1/b97d4dd8806(...)
```

The restore job log will show that the restore was done to a local directory. The log above was truncated for a clear view.

Other

Resource listing

The Bacula Enterprise Docker Plugin supports the new Plugin Listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available Docker resources such as:

- List of Docker containers name-labels
- List of Docker images name-repository
- List of Docker volumes

The new feature uses the special **.ls** command with a new **plugin=<plugin>** parameter. The command requires the following parameters to be set:

client=<client>
  A Bacula Client name with the docker Plugin installed.

plugin=<plugin>
A Plugin name, which would be **docker:** in this case, with optional plugin parameters as described in section **genericparameters**.

path=<path>
An object path to display.

The supported values for a **path=<path>** parameter are:

/
to display object types available to list

container
to display a list of containers name-labels data

image
to display a list of images name-repository data

volume
to display a list of volume names data

To display available object types, follow the following command example:

```
*.ls plugin=docker: client=docker-test-fd path=/
Connecting to Client docker-test-fd at docker-test:9102
drwxr-x---   1 root     root              0 2018-09-28 14:32:20  image
drwxr-x---   1 root     root              0 2018-09-28 14:32:20  container
drwxr-x---   1 root     root              0 2018-09-28 14:32:20  volume
2000 OK estimate files=3 bytes=0
```

To display the list of all available Docker containers, the following command example can be used:

```
*.ls plugin=docker: client=docker-test-fd path=container
Connecting to Client docker-test-fd at docker-test:9102
root     root        88185293 2018-10-01 09:18:00  myubuntu -> 4d0a4fadb50d
root     root        61656268 2018-10-01 09:18:00  mcache1_98 -> cb0c2e54dd00
root     root        88185240 2018-10-01 09:18:00  my_docker -> 4eefcf7d61ee
root     root        88185240 2018-10-01 09:18:00  amazing_hamilton ->␣
↪73ce08ad3d59
root     root        88185240 2018-10-01 09:18:00  with.label -> 1f476fd3c1b1
root     root        61656268 2018-10-01 09:18:00  mcache1 -> b97d4dd88063
root     root       239075328 2018-10-01 09:18:00  brave_edison ->␣
↪66f45d8601ba
root     root       239075391 2018-10-01 09:18:00  some-postgres ->␣
↪28e4c3a3cd27
root     root       580911104 2018-10-01 09:18:00  my-ubuntu -> a6ba1cb597d5
root     root        88185245 2018-10-01 09:18:00  sharp_visvesvaraya ->␣
↪ce5844df6842
root     root       239075391 2018-10-01 09:18:00  my-postgres -> a7c9518405e8
root     root        88185240 2018-10-01 09:18:00  frosty_kowalevski ->␣
↪0f601bcb1ef5
root     root        88185302 2018-10-01 09:18:00  infallible_bose ->␣
↪00571da76da6
root     root            1894 2018-10-01 09:18:00  romantic_hermann ->␣
↪37285d94347a
2000 OK estimate files=14 bytes=2,038,748,444
```

To display the list of all available Docker images, use the following command example:

```
*.ls plugin="docker:" client=docker-test-fd path=image
Connecting to Client docker-test-fd at docker-test:9102
root     root        61656268 2018-10-01 09:19:34  memcached:latest ->␣
↪80256dbd25ae
root     root       239075328 2018-10-01 09:19:34  postgres:latest ->␣
↪ac25c2bac3c4
root     root        88185240 2018-10-01 09:19:34  ubuntu:latest ->␣
↪16508e5c265d
root     root            1894 2018-10-01 09:19:34  hello-world:latest ->␣
↪2cb0d9787c4d
2000 OK estimate files=4 bytes=0
```

The **container** and lists display an estimated size of Docker objects. The final size of backup could be different.

All above Docker objects lists use a short (truncated) object ids during display. It is technically plausible that you'll get the same truncated id for two or more objects. To properly distinguish them you can use `notrunc` plugin parameter. In this case all Docker objects ids will be displayed in full sha256 form, i.e.

```
*.ls plugin="docker: notrunc" client=docker-test-fd path=image
Connecting to Client docker-test-fd at docker-test:9102
(...) memcached:latest ->␣
↪80256dbd25aedb4c025d495f18e0e513dd011005726326f529c296e4141811f6
(...) postgres:latest ->␣
↪ac25c2bac3c4e56f949c60ca343e1c4cd95f493db28bbf29b8ce466b4171cc8f
(...) ubuntu:latest ->␣
↪16508e5c265dcb5c05017a2a8a8228ae12b7b56b2cda0197ed5411bda200a961
(...) hello-world:latest ->␣
↪2cb0d9787c4dd17ef9eb03e512923bc4db10add190d3f84af63b744e353a9b34
2000 OK estimate files=4 bytes=0
```

This plugin parameter works only in plugin listing mode and does not affect any backup, estimate or restore jobs. In this case you can get *Invalid parameter* job error.

To display the list of all available Docker volumes, the following command example can be used:

```
*.ls plugin=docker: client=docker-test-fd path=volume
Connecting to Client docker-test-fd at docker-test:9102
root  root 2019-07-19 18:13:41 ␣
↪a563ae0edf55a01a0cdb3165d854a7326f13793119a708c44a5e49bd72a0286d
root  root 2019-07-19 18:13:41 ␣
↪aa9d3074f8c65a5afafddc6eaaf9827e99bb51f676aafaacc05cfca0188e65bf
root  root 2019-07-19 18:13:41 ␣
↪b49a3607eb04a4d3d00ed9dc0910e12201ad85f6ca38cb8fc7b43333207203a9
root  root 2019-07-19 18:13:41 ␣
↪bd80f277bd000d493dde4de763b0e82f1c0d5fd760acc07cb5c971f74a314471
root  root 2019-07-19 18:13:41 ␣
↪c0a478d317195ba27dda1370b73e5cb94a7773f2a611142d7dff690abdcfdcbf
root  root 2019-07-19 18:13:41 ␣
↪e3f25066e7957a12f084de87686c37fdb954ab2c15068369d887d3718b860a4c
root  root 2019-07-19 18:13:41  my-vol
root  root 2019-07-19 18:13:41  testvolume
2000 OK estimate files=8 bytes=0
```

The **volume** list display a volume real size if this information is available from Docker else it display simple 0. The final size of backup could be different.

Limitations

- Granular restore (*Single Item Restore*) is not supported. This feature could be implemented in the future.
- Only Full level backups are possible. This is a Docker limitation for containers and images. For volume backup this limitation could be removed in the future.
- You can restore volume files data into the same volume name only. This limitation could be removed in the future.
- You will always overwrite restored volume files data regardless of **Replace** parameter value. This limitation will be removed in the future.
- Backup or restore of the Volume data files for remote Docker service is unsupported. In this case you should get a following warning when want to explicitly backup a such data:

  ```
  Warning: dkcommctx: Docker Volume backup with docker_host is unsupported!
  ```

  and a following warning when want to restore:

  ```
  Warning: docker: Docker Volume restore with docker_host is unsupported! \
  All volumes restore skipped.
  ```

  If you access the remote Docker service outside plugin configuration then your job might hung. This limitation will be removed in the future.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

3.3 IaaS

Azure Virtual Machine Plugin

Features Summary

- Snapshot-based online backup of any guest VM.
- Full, Incremental and Differential block level image backup
- Ability to restore complete virtual machine image.
- The Azure-VM plugin is compatible with Copy/Migration jobs. Please read the ProtectMigrationAndCopyJobs and CopyMigrationJobsReplication for more information.

Guest VM Backup Strategies

Installing Bacula Client on Each Guest

This strategy works by installing a Bacula Enterprise File Daemon on every virtual machine as if they were normal physical clients. In order to optimize the I/O usage on the Azure-VM hypervisor, the user will use Bacula's Schedules, Priorities, and Maximum Concurrent Jobs to spread backup jobs over the backup window. Since all VMs could use the same storage on the Microsoft Azure hypervisor, running all backup jobs at the same time could create a bottleneck on the disk/network subsystem since Bacula will walk through all filesystems to open/read/close/stat files.

Installing a Bacula Enterprise File Daemon on each virtual machine permits to manage virtual servers like physical servers and also to use all Bacula Enterprise's features such as:

- Quick restores of individual files.
- Checksum of individual files for Virus and Spyware detection.
- Verify Jobs.
- File/Directory exclusion (such as swap or temporary files).
- File level compression.
- Accurate backups.

Image Backup With Azure-Vm Plugin

With the image backup level strategy, the Bacula Enterprise Azure-VM Plugin will save the Client disks at the raw level, in the Azure context.

Bacula's Azure-VM plugin will read and save content of virtual machines disks using snapshots.

During backups, Azure plugin will save the integrity of disks images and also guest VM configurations to allow guest VM restores with their original parameters.

Backup and Restore Operations

Backup

The backup of a single guest VM takes the following steps:

- Export guest VM metadata configuration for future restore.
- Backup OS disk.
- Backup all data disks
- Wait for all procedures to finish
- Add new snapshot to a tracker and check for older snapshot to delete

Disk backup procedure

The same process is used to backup OS or data disk, it follows the following steps:

- Create a snapshot of the disk
- Issue an SAS token on the snapshot just created.
- Move snapshot data to a blob
- Revoke SAS token
- Stream disk data or incremental changes to Bacula

Bacula files

The backup will create the following backup files for each guest VM:

- A single empty file to match a guest VM name to its UUID /@azure-vm/<vmUuid>_<vmName>.name
- A single configuration metadata file: /@azure-vm/<vmUuid>/<epoch>_conf
- A unique raw data file for the guest VM OS disk /@azure-vm/<vmUuid>/<epoch>_osdisk
- A raw data file for each data disk: /@azure-vm/<vmUuid>/<epoch>_<index>

At restore time the user can identify the guest VM using the UUID to mark the corresponding files:

```
+---------------------------------------------------------------------+
| filename                                                            |
+---------------------------------------------------------------------+
| /@azure-vm/690d74f9-9ce4-45fb-9b23-149afebe18f7_VmName.name         |
| /@azure-vm/690d74f9-9ce4-45fb-9b23-149afebe18f7/1661160371_conf     |
| /@azure-vm/690d74f9-9ce4-45fb-9b23-149afebe18f7/1661161330_0        |
| /@azure-vm/690d74f9-9ce4-45fb-9b23-149afebe18f7/1661167810_osdisk |
+---------------------------------------------------------------------+
```

Restore

The Azure plugin restores data as a new guest VM.

The restore process goes through the following steps.

- Receive configuration metadata file.
- Restore OS disk.
- Restore data disks.
- Create guest VM from OS disk and information from configuration metadata file.
- Attach data disk to newly created guest VM
- Delete locally generated files

Note: If debug level (debug parameter) is at its maximum level (9) locally generated file are not deleted

---

Disk restore process

This section will describe the restore process for any type of disk.

- Restore disk locally
- Apply incremental changes if any
- Create empty disk on Azure side.
- Issue SAS on the newly created disk.
- Upload locally restored data to disk.
- Revoke SAS

Note: Since disk are restored locally before being upload to azure there is a free space requirement of at least the size of the full VM

Restore to Local Disk with Azure-VM Plugin

Bacula Enterprise allows restoring any file (`bvmdk`, `conf`, etc.) to your File Daemon's local disks. Then, you may attach the disk to a running Azure Virtual Machine and perform file level restores from the running VM.

By using `where=/path/to/dir` in the restore options, the Plugin will automatically restore selected files to this location on your File Daemon's local disk.

After the disk is restored to a local directory, one can create a new disk in Azure and upload the restored disk data. The following guide assumes the user is logged to Azure account as described in *the installation section*.

The first step is to get the size of a restored VM disk:

```
ls -ln /path/to/dir/
```

The new Azure VM disk has to be created with the exact same size of a restored disk:

```
az disk create -n <DiskName> -g <ResourceGroupName> --upload-type Upload --
↪upload-size-bytes <DiskSizeInBytes>
```

Write permissions need to be granted to a new placeholder disk:

```
az disk grant-access -n <DiskName> -g <ResourceGroupName> --access-level␣
↪Write  --duration-in-seconds 86400
```

The above command will return an ``accessSAS` token that is needed for the next step:

```
azcopy copy /path/to/dir/<LocallyRestoredDisk>.bvmdk <accessSAS> --blob-type␣
↪PageBlob
```

After the upload is completed, the write permissions need to be revoked:

```
az disk revoke-access <DiskName> -g <ResourceGroupName>
```

Get the uploaded disk ID and attach it to Azure Virtual Machine:

```
diskId=$(az disk show -g <ResourceGroupName> -n <DiskName> --query 'id' -o␣
↪tsv)
az vm disk attach -g <ResourceGroupName> --vm-name <VmName> --name $diskId
```

Then, one can SSH to Azure Virtual Machine and inspect the content of the restored disk or perform file level restores.

Finally, the restored disk can be detached from the VM and deleted if needed:

```
az vm disk detach -g <ResourceGroupName> --vm-name <VmName> -n <DiskName>
az disk delete --name <DiskName> -g <ResourceGroupName>
```

Incremental backup fallback process

During incremental backup if the plugin fails for any reason to compute incremental changes for a disk the disk will be backed up as a full image instead. A message will be displayed and the resulting backup will end with a `Backup OK with warning` status. At restore time the user must carefully mark the files for restore by ignoring older full/incremental images of said disk.

Snapshot safekeeping and data usage

Microsoft Azure uses a disk by disk snapshot policy. Bacula will keep track of snapshot relation between one another and delete snapshots that are no longer relevant. However, multiple snapshots of the same disk can be expected on the storage at any given time.

On Microsoft Azure incremental snapshots appear to have the same size as the full disk. However, the data contained inside the snapshot will be the differential data. Per Azure documentation incremental snapshots are billed for the used size only.

Plugin Installation

The Microsoft Azure plugin, can work on any machine with a Bacula File Daemon.

Since all backup/restore interactions are network based, any Bacula Enterprise File Daemon with access to the necessary Microsoft Azure endpoints can be used to run the plugin.

Dependencies

The Microsoft Azure plugin needs two additional software to work

- Microsoft Azure CLI
- AzCopy

After the installation the final step is to log into `Azure CLI` by running `az login` or `az login --use-device-code` if the plugin runs on a server with no GUI.

Note: When using `az login --use-device-code`, Azure-CLI may prompt the user to visit: https://microsoft.com/devicelogin for login, however, the page is no longer available. The user should instead go to: https://aka.ms/devicelogin.

Configuration of the Bacula File Daemon

The `Plugin Directory` directive of the `File Daemon` resource in */opt/bacula/etc/bacula-fd.conf* should point to the location where the `azure-vm-fd.so` plugin is installed. The default directory is: */opt/bacula/plugins*

```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Installation of the Plugin

For more information about plugin installation see LinuxInstallFileDaemon.

Plugin Configuration

The plugin is configured using `Plugin Parameters` defined in the "Include" section of a Fileset resource (Bacula Director configuration).

Generic Plugin Parameters

The following Microsoft Azure plugin parameters impact any type of Job (Backup, Restore, Query).

abort_on_error[= <0 or 1>] Specifies whether or not the plugin should abort its execution if a fatal error happens during backup or restore. This parameter is optional. The default value is `0`.

tenant_id=<string> Will create, with subscription_id, an Azure profile to gain access to Azure sdk functionalities. This parameter is mandatory.

subscription_id=<string> Will create, with tenant_id, an Azure profile to gain access to Azure sdk functionalities. This parameter is mandatory.

application_id=<string> Another parameter to gain access to Azure sdk functionalities. This parameter is mandatory.

application_secret=<string> Another parameter used to gain access to Azure sdk functionalities. This parameter is mandatory

connection_key=<string> Is used to create a `BlobServiceClientBuilder` to interact with blobs. This is a connection string from an Azure storage account. This parameter is mandatory.

debug=[0,9] Specifies the level of debug with `0` being no debug and `9` being the highest level of debug. Warnings and errors are always sent to the joblog and if any debug level is set those messages are sent to the debug file as well. For the Microsoft Azure plugin `1` displays debug level message, `2` displays trace level message. Any value higher than `2` displays additional information about external libraries that handle those values on their own. When the debug level is at its maximum, locally generated disk file at restore will not be deleted. This parameter is optional. The default value is `0`.

group=<string> Specifies which from which group the parser should retrieve the parameters. This parameter is optional. If this parameter is not set and a config file exsist at `/opt/bacula/etc/azure-vm.conf` the first available group will be chosen by default.

See the `Configuration File` section for more informations and examples about configuration files for the Azure plugin.

Note: `tenant_id`, `application_id` and `application_secret` parameters can be manually generated by running the following command `az ad sp create-for-rbac --role Owner --scopes /subscriptions/<subscription-id>`. If successful the command will output four different values.

- `tenant` = `tenant_id`
- `appId` = `application_id`
- `password` = `application_secret`
- `displayName` = Does not correspond to any of the plugin parameter

Note: Parameters from the fileset have precedence over parameters provided by a configuration file.

Backup Plugin Parameters

include=<Java Regexp> Specifies a list of guest VM names to backup.

exclude=<Java Regexp> Specifies a list of guest VM names to not backup.

vm=<guest VM name> Specifies the name of a single guest VM to backup.

The use of regular expressions in the parameters `include=` and `exclude=` must be a Java compatible regular expression.

In order to be backed up the guest VM must match the `include=...` predicate and not match the `exclude=...`. A guest VM that matches the `vm=...` will be backed up regardless of the include/exclude specifications.

By default all guest VMs match the include predicate and not the exclude. Therefore, if none of the parameters `vm=...`, `include=...` and `exclude=...` are provided, all available guest VMs hosted on the Microsoft Azure hypervisor will be backed up.

On the other hand, if the parameter `vm=...` is specified, all guest VMs will no longer match `include=...` predicate. This means that if only `vm=...` parameter is specified, no other guest VM will be backed up.

See Fileset Examples section for examples of include/exclude/vm setups.

Restore Plugin Parameters

new_hostname=<String> Specified when a guest VM should be restored with a specific name.

A restore job can only restore one guest VM at a time. To select the relevant VM the user should interact with Bacula *bconsole* and `mark` all files in the guest VM folder identified by its UUID.

Note: If the Bacula restore parameter: `where=<path>` contains data, the plugin will restore the disks locally at `<path>`.

```
cwd is: /
$ ls
$ @azure-vm/
$ cd @azure-vm
cwd is: /@azure-vm/
$ ls
010a9727-ec67-4f3c-ac8b-129310764aa1/
010a9727-ec67-4f3c-ac8b-129310764aa1_testvm.name
$ mark 010a9727-ec67-4f3c-ac8b-129310764aa1*
4 files marked.
$ done
```

Configuration File

For this plugin it is possible to pass arguments from a configuration file. Configuration file follows the `.ini` format. Each combination of parameters has to be identified by a bracketed group name then each parameters are set via a `key = value` format.

The azure plugin will look for a configuration file located at `/opt/bacula/etc/azure-vm.conf`. If such file is present the plugin will then look for the relevant group to get parameters from. If the `group` parameter is not set then the first available group will be chosen.

Parameters provided by the Fileset have precedence over the parameters provided by the configuration file.

Configuration File Example

Here is an example of a configuration file at `/opt/bacula/etc/azure-vm.conf` for two different azure configurations.

```
[main]
tenant_id = 11111111-aaaa-bbbb-cccc-000000000000
subscription_id = 12345678-abcd-efgh-ijkl-01234567890ab
connection_key = DefaultEndpointsProtocol=[...]
application_id = 12345678-2222-3333-4444-555555555555
application_secret = qqqq~---------------------qwertzuiopasd


[second]
tenant_id = aaaaaaaa-1111-2222-3333-zzzzzzzzzzzz
subscription_id = abcdefgh-1234-5678-9012-abcdefghijkl
connection_key = DefaultEndpointsProtocol=[...]
application_id = 87654321-1111-2222-3333-44444444444
application_secret = 9qqqq~---------------------qw123456789d
```

For the following fileset example since no `group` parameter is provided the plugin will look for the fist available group. In this case it will be `main`.

```
Fileset {
Name = "FS_AZURE_VM"
  Include {
    Options {
      signature = MD5
      compression = LZO
    }
    Plugin = "azure-vm: vm=vm"
  }
}
```

For the following fileset example the parameter group `second` will be selected.

```
Fileset {
Name = "FS_AZURE_VM"
  Include {
    Options {
```

(continues on next page)

(continued from previous page)

```
      signature = MD5
      compression = LZO
    }
    Plugin = "azure-vm: vm=vm group=second"
  }
}
```

Fileset Examples

In the example below, all guest VMs will be backed up:

```
Fileset {
     Name= Azure-all-params-all-vms
     Include {
       Plugin="azure-vm: tenant_id=12345678-abcd-1234-efgh-0123456789ab␣
↪subscription_id=abcdefgh-1234-abcd-1234-abcdefghijkl connection_
↪key=DefaultEndpointsProtocol=https;AccountName=example;
↪AccountKey=aBcdEfGhijKlMNOpqrstuvwxyZ012345;EndpointSuffix=core.windows.net␣
↪application_id=87654321-1111-2222-3333-44444444444 application_secret=9qqqq~
↪---------------------qw123456789d"
     }
}
```

In the example below, all guest VMs will be backed up but this time the `tenant_id`, `subscription_id`, `application_id`, `secret_id` and `connection_key` are set inside a configuration file located at `/opt/bacula/etc/bacula-vm.conf` into a group named `main`

Note: In all the future examples, it will be assumed for less verbose Filesets, that `tenant_id`, `subscription_id`, `application_id`, `secret_id` and `connection_key` are set by the same configuration file and that the group `main` is the first available group.

```
Fileset {
      Name= Azure-no-params
      Include {
        Plugin="azure-vm: group=main"
      }
}
```

In the example below, a single guest VM with a name of "VM1" will be backed up.

```
Fileset {
    Name= Azure-One-Vm
    Include {
        Plugin="azure-vm: vm=VM1"
    }
}
```

In the example below, all guest VMs which have `prod` in their name will be backed up.

```
Fileset {
    Name= Azure-prod
    Include {
        Plugin="azure-vm: include=(.*)prod(.*)"
    }
}
```

In the example below, all guest VMs which have `prod` in their name but do not start with `test` will be backed up.

```
Fileset {
    Name= Azure_no_test
    Include {
        Plugin="azure-vm: include=(.*)prod(.*) exclude=^test(.*)"
    }
}
```

In the example below, all `prod` VMs will be backed up. All `test` VMs will be ignored except for a VM named `exception.test`

```
Fileset {
    Name= Azure_prod_no_test_except
    Include {
        Plugin="azure-vm: include=(.*)prod(.*) exclude=(.*)test(.*)␣
↪vm=exception.test"
    }
}
```

Specific bconsole Query Commands

The Bacula Enterprise Azure-vm plugin supports also the query parameter.

The plugin answer the query in the form of tuple `key=value`.

The plugin supports two operators if none of them is passed the plugin will answer with `Query not recognized, possible value={CONNECTION, VM}`

CONNECTION

This query check if the system is logged to a Microsoft Azure account

When the query parameter is `CONNECTION`, the returned value will be either `OK` or `NOK`, indicating whether the connection was successful or not.

```
.query plugin="azure-vm:" client=client-fd parameter=CONNECTION
info=Query
CONNECTION=OK
```

VM

This query sends a request to Azure that lists all available guest VMs. This query displays one `key=value` per virtual machine where `key` is `VM` and `value` is the name of the virtual machine.

The example below shows the execution of a correctly formatted `VM` query that finds three different guest VMs.

```
.query plugin="azure-vm:" client=client-fd parameter=VM
info=Query
VM=Virtual1
VM=Virtual2
VM=Virtual3
```

GROUP

This query looks for a configuration file located at `/opt/bacula/etc/azure-vm.conf` and prints all available parameter groups in a `GROUP=groupName`

```
.query plugin="azure-vm:" client=client-fd parameter=GROUP
info=Query
GROUP=main
GROUP=second
```

VERSION

Print information relative to azure and `azcopy` CLI

```
.query plugin="azure-vm:" client=client-fd parameter=VERSION
info=Query
azure-cli=2.40.0
azure-cli-core=2.40.0
azure-cli-telemetry=1.0.8
azcopy=10.16.0
```

Limitations

- Restore job needs to have the size of the full VM in free space
- The plugin may restore only one guest VM per restore job
- Virtual full jobs are not supported

Amazon EC2 Plugin

The following article aims at presenting the reader with information about the **Bacula Enterprise Amazon EC2 Plugin**.

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 or any other Cloud computing service is usually done to reduce hardware costs, to gain flexibility and being more agile in order to deploy applications faster. Amazon EC2 can be used to launch as many or as few virtual servers as the target service needs, to configure security and networking, and to manage storage. It is possible to add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, it is possible to reduce capacity (scale down) again and orchestrate all these operations automatically.

Bacula Enterprise Amazon EC2 Plugin provides backup and restore of the instances running in Amazon EC2 service and the persistent data associated with them. It does it in an agent-less manner, working at the block level, using snapshots to represent the state of the information at the moment of the backup and supporting Full, Incremental or Differential backups. It offers a very high level of flexibility to perform its different operations and a high level of performance when the underlying network is appropriate to the backup target. Bacula Enterprise not only protects persistent data of the storage, but also all the configuration of the instance and the volumes, providing the ability to restore it as it was or to modify them as desired at restore time.

To provide its services, Amazon EC2 is directly linked with various AWS Cloud services, such as IAM (Identity and Access Management Service), EBS (Elastic Block Storage Service) and S3 (Amazon Simple Storage Service). Below, we outline the functions of each service and explain how they work together. This will help us understand how Bacula Enterprise manages the backup and restore of data stored in these services. The information is extracted directly from the official AWS Documentation, where it is possible to find more in-depth details:

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes designed for utilization with EC2 instances. EBS volumes behave like raw, unformatted block devices. EBS volumes function as unformatted, raw block devices and can be easily mounted as devices on EC2 instances. When attached to an instance, EBS volumes are presented as storage volumes that remain persistent even if the instance is terminated. A file system can be created on top of these volumes, or use them in any way a regular block device can be used (such as a hard drive). It is possible to dynamically change the configuration of a volume attached to an instance. As explained earlier, Bacula Enterprise not only protects the data of EBS Volumes, but also all the configuration they use, allowing for seamless restoration to its original state or customization as needed during the restore process. Bacula Enterprise uses Amazon EBS High performance Direct APIs to fetch the data of the target volumes for backup. It is important to consider the pricing structure associated with the entire Amazon EBS layer:

- https://aws.amazon.com/ebs/pricing/

AWS Identity and Access Management (IAM) is a web service provided by Amazon that enables users to securely manage access to AWS resources. IAM plays a crucial role in controlling user access to various AWS resources by managing permissions. It is utilized to determine authentication (sign-in) and authorization (permissions) for resource usage. The Bacula Enterprise Amazon EC2 Plugin necessitates the configuration of an access key within IAM, along with a specific set of permissions, to facilitate listing, backup, and restore operations.

Amazon Simple Storage Service (Amazon S3) is a highly scalable object storage service that offers top-notch data availability, security, and performance. It caters to customers of various sizes and industries, allowing them to securely store and safeguard any volume of data for a wide range of purposes. These purposes include data lakes, websites, mobile applications, backup and restore operations, archiving, enterprise applications, IoT devices, and big data analytics. With Amazon S3, users can efficiently manage, organize, and control access to their data, ensuring it aligns with their specific business, organizational, and compliance requirements. Bacula Enterprise Amazon EC2 service utilizes S3 to store snapshots, which are then analyzed to identify changes in Incremental or Differential backups before being downloaded. The Bacula Enterprise Amazon EC2 Plugin takes care of cleaning up these snapshots, both before (for failed backups) and after (for unnecessary snapshots) backup operations. However, certain snapshots need to be retained in S3 to facilitate Differential and Incremental backups, and the Bacula Enterprise Amazon EC2 Plugin handles this process seamlessly.

Through subchapters, more in-depth information can be found about the following topics:

Scope

Bacula Enterprise Amazon EC2 Plugin currently supports backup and restore instances deployed in the Amazon EC2 service when the data is accessible through the Amazon EBS Direct API. For more information about this API, consult:

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-accessing-snapshot.html

This plugin is available since **Bacula Enterprise 18.0**, and needs to be deployed in a Linux host.

Features

The main feature of the **Bacula Enterprise Amazon EC2 Plugin** is to offer backup and restore for Amazon EC2 instances (also known as Virtual Machines). This includes the instance configuration data (or metadata), the associated volumes (also known as disks) configuration data (or metadata), as the actual data stored on those volumes.

In addition tits main purpose, this Plugin permits users to customize its functions to meet their specific needs. It offers a high level of flexibility to select the target information to protect, to parallelize the operations efficiently through different strategies or to restore the information from or to different locations, allowing the modification of many parameters at that restore time.

General Features

Below, there is a list of general features this Plugin offers:

- Backup and restore of Amazon EC2 instances
- Backup and restore of Amazon EBS volumes
- Amazon AWS API based backups: Amazon EBS Direct API based backups to access volume blocks
- Automatic multi-threaded download or upload operations
- Network resiliency mechanisms
- Instances and volume discovery capabilities
- List instances and disk through query or list commands
- Auto-generation capabilities if combined with Scan Plugin
- Restore objects to Amazon EC2
  - To the original location (region, availability zone . . .)

  - To a different location (region, availability zone . . .)
  - With the same original instance and/or disks configurations (name, tags, instance type, network options, security groups . . .)
  - With different instance and/or disks configurations (volume type, volume iops, volume throughput . . .)
- Full, Incremental & Differential backups
  - Change block tracking based on snapshot differences
- Advanced instance selection for backup, filtering by any parameter
- Advanced disk selection for backup, including or excluding boot volumes or any disk by id
- Automatic snapshot cleanup before and after backups
- Command line mode to work from or to a given local file system path.

Architecture

Bacula Enterprise Amazon EC2 Plugin is a Bacula File Daemon plugin that utilizes the Amazon Web Services APIs to interact with the Amazon Cloud (retrieve data from Amazon Cloud and to feed it at restore time). The plugin runs a Java Daemon which uses the official Amazon AWS SDK version 2.x built by Amazon.

All the information is obtained using secure and encrypted HTTPS queries to AWS from the File Daemon (and through the mentioned Java Daemon), where the plugin is installed. The specific URLs will depend on the region and the operation being performed.

To get more information about AWS implied APIs, visit: https://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html

Metadata of every backed up element is stored in Bacula in JSON format. The blocks of the volumes are stored in memory before being streamed directly to the Storage Daemon.

Backup and restore processes use different parallelization techniques in order to maximize performance, and overcome latency times when communicating with AWS Parallelization. The Plugin also supports parallelization of multiple backup jobs.

Every block of data is 512KiB in size, so it is important to consider the total size of data blocks when configuring parallel operations (you could need at least that size multiplied by the number of parallel operations configured). For example, in the worst-case scenario of backing up one Virtual Machine with 500 operations, the total size would be 250MB.

Below, there is a simplified vision of the architecture of this plugin within a generic **Bacula Enterprise** deployment:

Installation

This article describes how to install Bacula Enterprise Amazon EC2 Plugin.

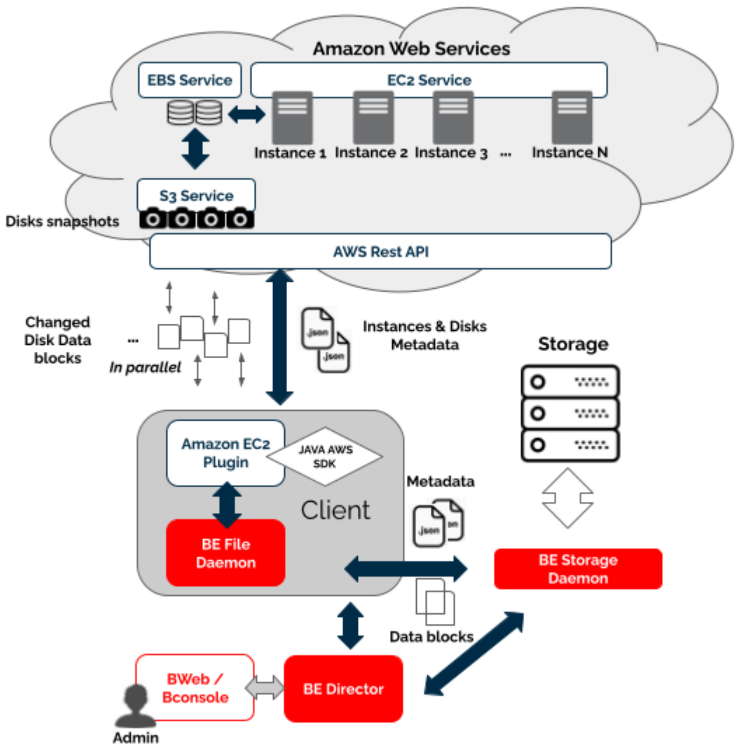


Fig. 31: Amazon EC2 Plugin Architecture

Prerequisites

- The Bacula File Daemon and the Amazon EC2 Plugin need to be installed on the host that will connect to the AWS Cloud.
- The plugin is implemented over a Java layer and it must be deployed in a host running Linux. It is possible to use any of the supported **Linux** distributions of Bacula Enterprise, including RHEL, Debian, Ubuntu or Suse Linux Enterprise Server as some examples.
- The plugin works through a Java daemon, therefore Java needs to be installed into the host using a JRE or JDK package (openjdk-11-jre for example). The Java environment should be version 11 or higher, and the Java binary must be accessible in the system PATH.
- Memory and computational requirements completely vary based on the plugin's configuration and usage, such as parallelization and data size for backup etc. It is recommended to have a minimum of **4GB RAM** on the server where the File Daemon is running. By default, each job may use up to 512Mb of RAM in demanding scenarios, although it is typically less. In some situations this could be higher. Memory limits can be adjusted (see Out of memory).
- Amazon Web Services REST APIs are used to perform all plugin operations. Therefore, they must be accessible through HTTPS from the host where Bacula FD and the Plugin will be deployed.
- In order to fetch data, tan access key with key and secret is used to connect to AWS. Proper EC2, EBS and S3 permissions need to be associated to that access key. Details about how to configure such access key are given in the next sections.

Installation Methods

- AmazonEC2InstallationWithBIM (recommended)
- AmazonEC2InstallationPackageManagers
- AmazonEC2InstallationManual

Amazon EC2 Plugin Installation with BIM

The recommended way to install any Bacula component, including any Daemon and any plugin is using the Bacula Installation Manager (BIM).

Steps

1. Install File Daemon.
2. Select the amazon-ec2 key needs in the plugin selection step.

Result

Amazon EC2 Plugin is installed. Click here for more details.

For more information, visit Linux: Bacula Enterprise Installation with BIM.

Amazon EC2 Plugin Installation with Package Managers

Another way to install this Plugin is using the package manager of the used distribution.

Here, Debian Bullseye is taken as an example base operative system. The process will be very similar in any version of Debian, or any other Debian-based distribution (e.g. such as Ubuntu). In case of using RPM-based distributions, like RHEL, Oracle Linux or Suse Linux, the main difference is to switch to the proper package manager of that distribution, usually *yum*.

Steps

1. Add the repository file suitable for the existing customer subscription, and the Debian version utilized. An example would be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 37: **APT repositories**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪bullseye-64/ bullseye main
deb https://www.baculasystems.com/dl/@customer-string@/debs/amazon-ec2/
↪@version@/bullseye-64/ bullseye amazon-ec2
```

2. Run of apt update:

Listing 38: **APT update**

```
apt update
```

3. Install the plugin using:

Listing 39: **APT install**

```
apt install bacula-enterprise-amazon-ec2-plugin
```

The plugin has two different packages implied that should be installed automatically with the command shown:

- bacula-enterprise-amazon-ec2-plugin
- bacula-enterprise-amazon-ec2-plugin-libs

Result

Amazon EC2 Plugin is installed. Click here for more details.

Amazon EC2 Plugin Manual Installation

Alternately, manual installation of the packages may be done after downloading the packages, and then using the package manager to install them.

1. Download the packages from your Bacula Systems download area.

There are the packages you need to download for this plugin:

- bacula-enterprise-amazon-ec2-plugin
- bacula-enterprise-amazon-ec2-plugin-libs

2. After downloading, install them with the command:

Listing 40: **Manual dpkg install**

```
dpkg -i bacula-enterprise-*
```

Result

Amazon EC2 Plugin is installed. Click here for more details.

Result

The package installs the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-amazon-ec2-plugin-x.x.x.jar and bacula-amazon-ec2-plugin-libs-x.x.x.jar). Note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.
- Plugin connection file (amazon-ec2-fd.so) in the plugins directory (usually /opt/bacula/plugins). Note that amazon-ec2 acronym means Amazon Elastic Compute Cloud.
- Backend file (amazon-ec2-backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-amazon-ec2-plugin-x.x.x.jar file in order to launch it, even thought usually we should have only one file.
- Bacula ec2 script (bec2) that invokes the jar files in /opt/bacula/bin in 'direct mode'. This script also searches for the most recent bacula-amazon-ec2-plugin-x.x.x.jar file in order to launch it, even thought usually we should have only one file. Its purpose is to invoke the plugin directly from/to the filesystem and not through a Bacula job.

Once the plugin is installed, it should be possible to see it loaded through a status client command in bconsole ('Plugin:' line must contain 'amazon-ec2'):

Listing 41: **Status client**

```
*st client
Automatically selected Client: 127.0.0.1-fd
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102

127.0.0.1-fd Version: 18.0.0 (20 Nov 2023)  x86_64-pc-linux-gnu ubuntu 22.04
Daemon started 14-abr-23 10:14. Jobs: run=2 running=0 max=100.
Ulimits: nofile=1024 memlock=2026356736 status=ok
Heap: heap=827,392 smbytes=436,939 max_bytes=5,100,087 bufs=153 max_bufs=248
Sizes: boffset_t=8 size_t=8 debug=600 trace=1 mode=1,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL 3.0.2 15 Mar 2022
APIs: !GPFS
Plugin: bpipe(2) amazon-ec2(1.0.0)
```

Configuration

The following chapter presents the information on how to configure the Plugin. Backup jobs with Bacula Enterprise Amazon EC2 Plugin function similarly to other Bacula Enterprise backup jobs after the appropriate fileset has been established and the access key with necessary permissions is provided.

Important: Backups using this plugin and Incremental backups are required to use Accurate = yes option in the job configuration

In the following sections we present how to configure an AWS access key with the associated permissions to use this plugin, and then the parameters to set up a fileset to invoke Amazon EC2 Plugin through a job.

Restore parameters are discussed in the Restore section of Operations chapter.

Access Key Configuration

The utilization of Bacula Enterprise Amazon EC2 Plugin requires the involvement of an AWS IAM service user who possesses a specific set of permissions outlined below.

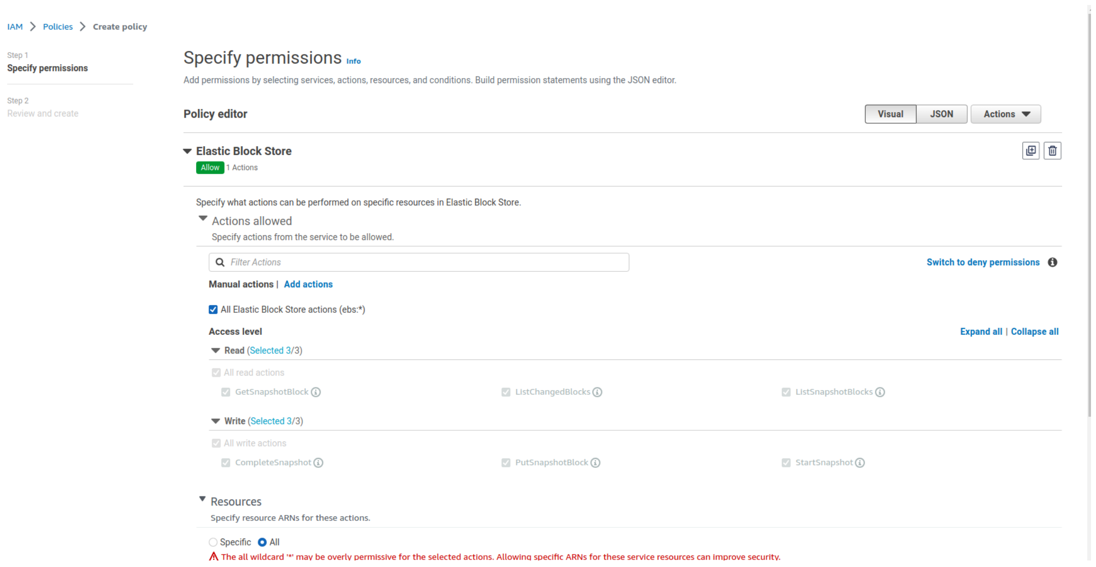
Subsequently, this user must establish an access key, which should be configured as fileset parameters. This configuration enables the plugin to effectively retrieve or write data during backup or restore operations.

This plugin needs the following set of permissions to work appropriately: - Full EC2 service permissions for the target instances - Full EBS service permissions for the target volumes - Full S3 service permissions, so snapshots of the volumes can be created and removed - Full SSM service permissions, so images can be discovered using their associated tag paths (ssm:image/tag/path), and not only by their ids

To obtain complete EBS permissions, it is typically required to create a new policy via the IAM > Policies menu, as illustrated in the image below.

If we look at the policy contents in JSON, they should be like the following image shows:

Upon obtaining the EBS Policy, we can combine it with the existing policies for S3 and EC2 in order to grant a user the relevant permissions. The newly created EBS policy can be easily located using the search toolbar, along with the other two:


IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor — Visual | JSON | Actions

Elastic Block Store
Allow 1 Actions

Specify what actions can be performed on specific resources in Elastic Block Store.

Actions allowed
Specify actions from the service to be allowed.

Filter Actions — Switch to deny permissions

Manual actions | Add actions

[✓] All Elastic Block Store actions (ebs:*)

Access level — Expand all | Collapse all

Read (Selected 3/3)
[✓] All read actions
[✓] GetSnapshotBlock [✓] ListChangedBlocks [✓] ListSnapshotBlocks

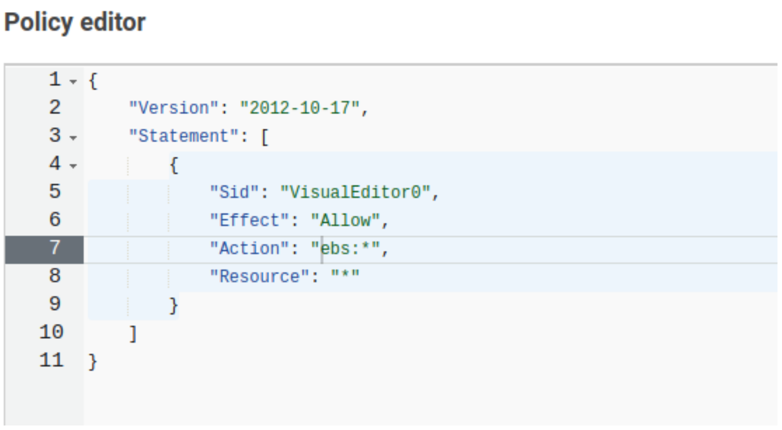
Write (Selected 3/3)
[✓] All write actions
[✓] CompleteSnapshot [✓] PutSnapshotBlock [✓] StartSnapshot

Resources
Specify resource ARNs for these actions.

Specific [✓ All]

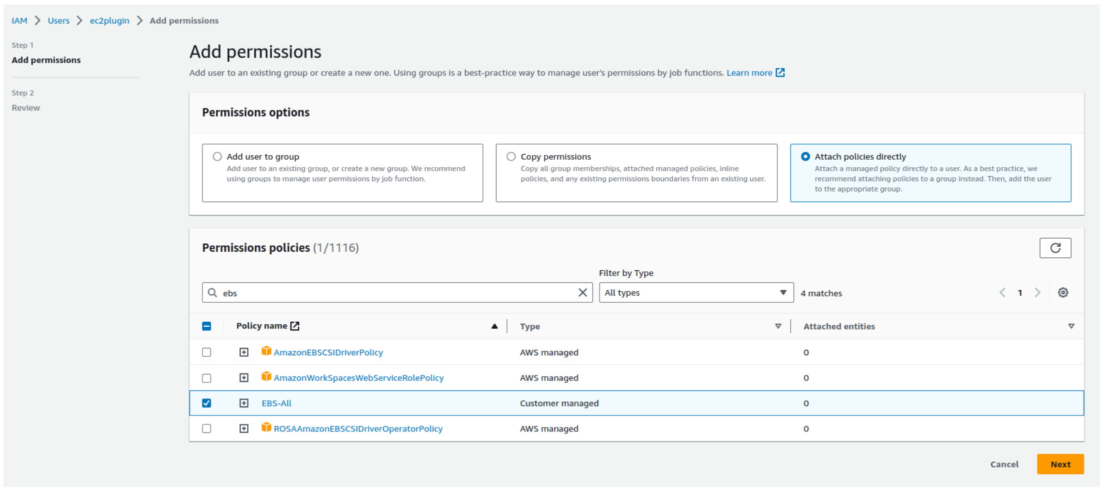
The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.

Fig. 32: New Policy for all EBS Permissions


Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*"
        }
    ]
}
```

Fig. 33: EBS Policy contents in JSON


IAM > Users > ec2plugin > Add permissions

Step 1
Add permissions

Step 2
Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

[✓ Attach policies directly]
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1116)

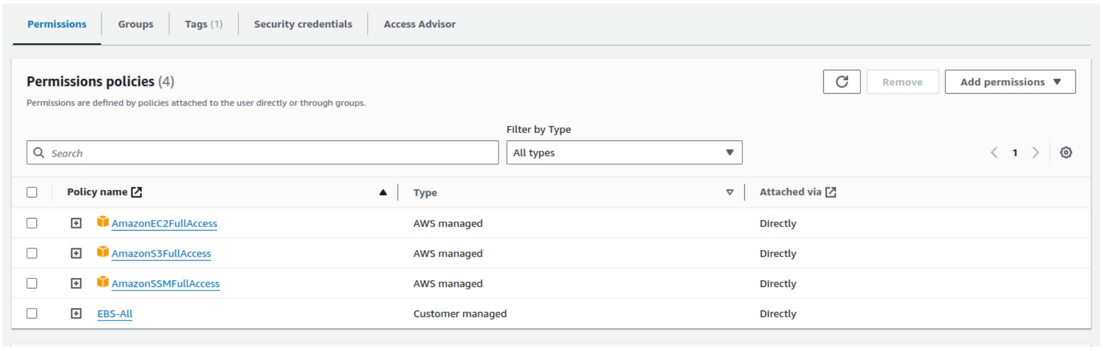
Filter by Type
ebs — All types — 4 matches

	Policy name	Type	Attached entities
	AmazonEBSCSIDriverPolicy	AWS managed	0
	AmazonWorkSpacesWebServiceRolePolicy	AWS managed	0
✓	EBS-All	Customer managed	0
	ROSAAmazonEBSCSIDriverOperatorPolicy	AWS managed	0

Cancel Next

Note: It is recommended to create a new specific user for this plugin.

Permissions for the configured user should look like the following image shows:



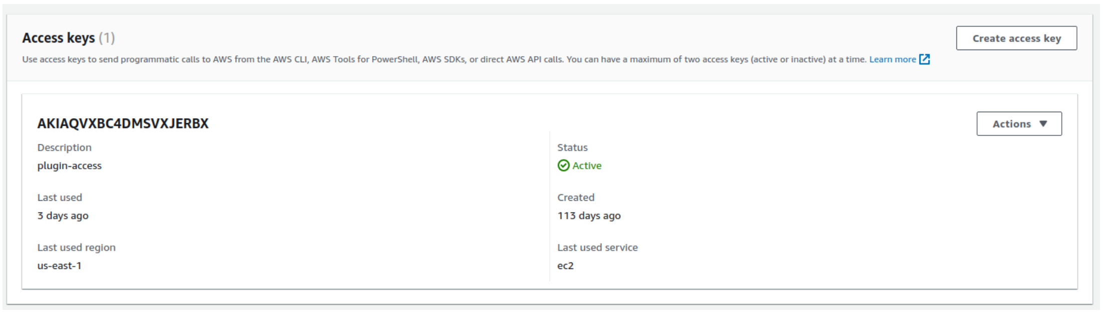
Permissions policies (4)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Directly
AmazonS3FullAccess	AWS managed	Directly
AmazonSSMFullAccess	AWS managed	Directly
EBS-All	Customer managed	Directly

Fig. 34: Amazon EC2 IAM User Permissions

Once we have the user, it is necessary to go to 'Access keys' and create a new one:



Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

AKIAQVXBC4DMSVXJERBX

Description plugin-access	Status Active
Last used 3 days ago	Created 113 days ago
Last used region us-east-1	Last used service ec2

Fig. 35: Amazon EC2 Access Key

The Id of the key (AKIAQV… in the image) and the associated secret are the parameters to configure in the plugin fileset in 'access_key' and 'access_secret' parameters. Adding also the 'region' parameter should be enough to allow the plugin to connect to the target dataset to protect.

Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to invoke the Amazon EC2 Plugin:

Listing 42: **Fileset EC2**

```
Fileset {
   Name = FS_EC2
   Include {
      Options {
        signature = MD5
        ...
      }
```

(continues on next page)

(continued from previous page)

```
      Plugin = "amazon-ec2: <amazon-ec2-parameter-1>=<amazon-ec2-value-1>
↪<amazon-ec2-parameter-2>=<amazon-ec2-value-2> ..."
   }
}
```

It is **strongly recommended** to use only one 'Plugin' line in every fileset. The plugin offers the needed flexibility to combine different modules backup inside the same plugin line. Different ec2 servers, in case of existing, should be using different filesets and different jobs.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ','.

Below, in the subsections, there are lists that present all the parameters you can use to control Amazon EC2 Plugin behavior.

Fileset Connection Parameters

The following parameters control the connection of the Amazon EC2 Plugin AWS.

Option	Require	Default	Values	Example	Description
access_	**Yes**		String: access key with proper permissions	AKIAIOSFODNN7EXAMPL	Access key defined in the desired AWS account with access to the target instances and having the right permissions
secret_	**Yes**		String: secret key associated to provided access key	wJalrXUtnFEMI/K7MDENG/b	Secret associated to the access key
region	No	eu-west-1	String: region code	eu-east-1	Existing region in AWS where the instances to backup reside

Fileset Backup Parameters

The following list of parameters control what is going to be included into the associated backup:

Option	Requir	Default	Values	Example	Description
instance	No		Valid instance names or ids separated by ','	myInstance1, myInstance2	Backup each instance containing the tag 'Name' with the provided values or the provided instance ids. If no instance is provided the plugin will list all the instances and will backup them
instance	No		Valid instance names or ids separated by ','	exclude-Example, i-1233sabll	Exclude selected instance containing the tag 'Name' with the provided values or the provided instance ids. If this is the only parameter found for selection, all elements will be included and this list will be excluded
instance	No		Valid regex	.*-included$	Backup matching instances by name. Please, only provide list parameters (instances + instances_exclude) or regex ones. But do not try to combine them
instance	No		Valid regex	.*-excluded$	Exclude matching instances by name. Please, only provide list parameters (instances + instances_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for user selection, all instances will be included and matching instances will be excluded
instance	No		Tag keys or values (format: tagkey1=value	backup=y	Backup instances containing the specified tags
instance	No		Tag keys or values (format: tagkey1=value	backup=r	Exclude instances containing the specified tags
disks	No		Disk ids separated by ','	vol-123fadfaf	Backup only the list of ebs volumes indicated by this parameter from the selected instances. If not specified, all disks from each instance will be backed up
disks_	No		Disk ids separated by ','	vol-456fadfaf vol-78904ads	Exclude selected volumes from the selectd instances. If this is the only parameter found for disk selection, all disks will be included and this list will be excluded
disks_	No		Valid regex	.*232323	Backup matching volumes by name. Please, only provide list parameters or regex ones. But do not try to combine them
disks_	No		Valid regex	.*vol-34.*	Exclude matching volumes by name. Please, only provide list parameters or regex ones. But do not try to combine them. If this is the only parameter found for volume selection, all volumes will be included and matching volumes will be excluded
instance	No		String representing a full path of a .json file	/opt/bacu filter.json	Use the file to interpret all the filters and backup only the matching instances. See next section to have more information about this advanced filter
keep_f	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Incremental backups will normally remove the Full backup, making the combination with Differential backups not possible. If that combination is desired, enable this parameter and the Full snapshot wont be removed
shutdown	No	No	0, no, No, false, FALSE, false, off ;	Yes	Shutdown the selected instance(s) before doing the snapshot of the backup operation and start them just after. This will ensure application consistency, but be aware of the inherent service disruption with the

Advanced Instance Filter

The `instances_filter_file` parameter provides users with a highly adaptable method to filter the selection of backup target instances according to their preferences.

This parameter requires a file path that points to a JSON file. The file must be locally accessible to the File Daemon and have the necessary permissions to be read.

The supported parameters are all the ones described in this AWS article in the section associated to the parameter 'filters' that are not already present as other direct parameters for the plugin: https://awscli.amazonaws.com/v2/documentation/api/2.0.34/reference/ec2/describe-instances.html

The article refers to the sub-routine 'describe-instances' of the command line tool AWS CLI, so it should be familiar to users running workloads over AWS. The possible parameters are also listed below:

Listing 43: **Possible filters**

```
affinity - The affinity setting for an instance running on a Dedicated Host␣
↪(default | host ).

architecture - The instance architecture (i386 | x86_64 | arm64 ).

availability-zone - The Availability Zone of the instance.

block-device-mapping.attach-time - The attach time for an EBS volume mapped␣
↪to the instance, for example, 2010-09-15T17:15:20.000Z .

block-device-mapping.delete-on-termination - A Boolean that indicates␣
↪whether the EBS volume is deleted on instance termination.

block-device-mapping.device-name - The device name specified in the block␣
↪device mapping (for example, /dev/sdh or xvdh ).

block-device-mapping.status - The status for the EBS volume (attaching |␣
↪attached | detaching | detached ).

block-device-mapping.volume-id - The volume ID of the EBS volume.

client-token - The idempotency token you provided when you launched the␣
↪instance.

dns-name - The public DNS name of the instance.

group-id - The ID of the security group for the instance. EC2-Classic only.

group-name - The name of the security group for the instance. EC2-Classic␣
↪only.

hibernation-options.configured - A Boolean that indicates whether the␣
↪instance is enabled for hibernation. A value of true means that the␣
↪instance is enabled for hibernation.

host-id - The ID of the Dedicated Host on which the instance is running, if␣
↪applicable.
```

(continues on next page)

(continued from previous page)

```
hypervisor - The hypervisor type of the instance (ovm | xen ). The value xen␣
↪is used for both Xen and Nitro hypervisors.

iam-instance-profile.arn - The instance profile associated with the instance.
↪ Specified as an ARN.

image-id - The ID of the image used to launch the instance.

instance-lifecycle - Indicates whether this is a Spot Instance or a␣
↪Scheduled Instance (spot | scheduled ).

instance-state-code - The state of the instance, as a 16-bit unsigned␣
↪integer. The high byte is used for internal purposes and should be ignored.␣
↪The low byte is set based on the state represented. The valid values are: 0␣
↪(pending), 16 (running), 32 (shutting-down), 48 (terminated), 64 (stopping),
↪ and 80 (stopped).

instance-state-name - The state of the instance (pending | running |␣
↪shutting-down | terminated | stopping | stopped ).

instance-type - The type of instance (for example, t2.micro ).

instance.group-id - The ID of the security group for the instance.

instance.group-name - The name of the security group for the instance.

ip-address - The public IPv4 address of the instance.

kernel-id - The kernel ID.

key-name - The name of the key pair used when the instance was launched.

launch-index - When launching multiple instances, this is the index for the␣
↪instance in the launch group (for example, 0, 1, 2, and so on).

launch-time - The time when the instance was launched.

metadata-options.http-tokens - The metadata request authorization state␣
↪(optional | required )

metadata-options.http-put-response-hop-limit - The http metadata request put␣
↪response hop limit (integer, possible values 1 to 64 )

metadata-options.http-endpoint - Enable or disable metadata access on http␣
↪endpoint (enabled | disabled )

monitoring-state - Indicates whether detailed monitoring is enabled␣
↪(disabled | enabled ).

network-interface.addresses.private-ip-address - The private IPv4 address␣
↪associated with the network interface.
```

(continues on next page)

(continued from previous page)

```
network-interface.addresses.primary - Specifies whether the IPv4 address of␣
↪the network interface is the primary private IPv4 address.

network-interface.addresses.association.public-ip - The ID of the␣
↪association of an Elastic IP address (IPv4) with a network interface.

network-interface.addresses.association.ip-owner-id - The owner ID of the␣
↪private IPv4 address associated with the network interface.

network-interface.association.public-ip - The address of the Elastic IP␣
↪address (IPv4) bound to the network interface.

network-interface.association.ip-owner-id - The owner of the Elastic IP␣
↪address (IPv4) associated with the network interface.

network-interface.association.allocation-id - The allocation ID returned␣
↪when you allocated the Elastic IP address (IPv4) for your network interface.

network-interface.association.association-id - The association ID returned␣
↪when the network interface was associated with an IPv4 address.

network-interface.attachment.attachment-id - The ID of the interface␣
↪attachment.

network-interface.attachment.instance-id - The ID of the instance to which␣
↪the network interface is attached.

network-interface.attachment.instance-owner-id - The owner ID of the␣
↪instance to which the network interface is attached.

network-interface.attachment.device-index - The device index to which the␣
↪network interface is attached.

network-interface.attachment.status - The status of the attachment␣
↪(attaching | attached | detaching | detached ).

network-interface.attachment.attach-time - The time that the network␣
↪interface was attached to an instance.

network-interface.attachment.delete-on-termination - Specifies whether the␣
↪attachment is deleted when an instance is terminated.

network-interface.availability-zone - The Availability Zone for the network␣
↪interface.

network-interface.description - The description of the network interface.

network-interface.group-id - The ID of a security group associated with the␣
↪network interface.

network-interface.group-name - The name of a security group associated with␣
```

(continues on next page)

(continued from previous page)

```
↪the network interface.

network-interface.ipv6-addresses.ipv6-address - The IPv6 address associated␣
↪with the network interface.

network-interface.mac-address - The MAC address of the network interface.

network-interface.network-interface-id - The ID of the network interface.

network-interface.owner-id - The ID of the owner of the network interface.

network-interface.private-dns-name - The private DNS name of the network␣
↪interface.

network-interface.requester-id - The requester ID for the network interface.

network-interface.requester-managed - Indicates whether the network␣
↪interface is being managed by AWS.

network-interface.status - The status of the network interface (available )␣
↪| in-use ).

network-interface.source-dest-check - Whether the network interface performs␣
↪source/destination checking. A value of true means that checking is enabled,
↪ and false means that checking is disabled. The value must be false for the␣
↪network interface to perform network address translation (NAT) in your VPC.

network-interface.subnet-id - The ID of the subnet for the network interface.

network-interface.vpc-id - The ID of the VPC for the network interface.

owner-id - The AWS account ID of the instance owner.

placement-group-name - The name of the placement group for the instance.

placement-partition-number - The partition in which the instance is located.

platform - The platform. To list only Windows instances, use windows .

private-dns-name - The private IPv4 DNS name of the instance.

private-ip-address - The private IPv4 address of the instance.

product-code - The product code associated with the AMI used to launch the␣
↪instance.

product-code.type - The type of product code (devpay | marketplace ).

ramdisk-id - The RAM disk ID.

reason - The reason for the current state of the instance (for example,␣
↪shows “User Initiated [date]” when you stop or terminate the instance).␣
```

(continues on next page)

(continued from previous page)

```
↪Similar to the state-reason-code filter.

 requester-id - The ID of the entity that launched the instance on your␣
↪behalf (for example, AWS Management Console, Auto Scaling, and so on).

 reservation-id - The ID of the instance's reservation. A reservation ID is␣
↪created any time you launch an instance. A reservation ID has a one-to-one␣
↪relationship with an instance launch request, but can be associated with␣
↪more than one instance if you launch multiple instances using the same␣
↪launch request. For example, if you launch one instance, you get one␣
↪reservation ID. If you launch ten instances using the same launch request,␣
↪you also get one reservation ID.

 root-device-name - The device name of the root device volume (for example, /
↪dev/sda1 ).

 root-device-type - The type of the root device volume (ebs | instance-store␣
↪).

 source-dest-check - Indicates whether the instance performs source/
↪destination checking. A value of true means that checking is enabled, and␣
↪false means that checking is disabled. The value must be false for the␣
↪instance to perform network address translation (NAT) in your VPC.

 spot-instance-request-id - The ID of the Spot Instance request.

 state-reason-code - The reason code for the state change.

 state-reason-message - A message that describes the state change.

 subnet-id - The ID of the subnet for the instance.

 tenancy - The tenancy of an instance (dedicated | default | host ).

 virtualization-type - The virtualization type of the instance (paravirtual |␣
↪hvm ).

 vpc-id - The ID of the VPC that the instance is running in.
```

This information needs to be formatted in the file with a JSON structure. Below is an example:

Listing 44: **Advanced Filter: File Contents Example**

```
{
     "instance-type" : "t2.micro",
     "architecture" : "x86_64"
}
```

Fileset Common Parameters

These parameters are controlling generic aspects of the behavior of the Amazon EC2 Plugin, it is possible to find also these parameters in other Bacula Enterprise Plugins with similar effects, so you may be familiar with them.

Option	Require	Default	Values	Example	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/ ec2.settings	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset
log	No	/opt/bacula ec2/amazo ec2-debug.log	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/amazo ec2.log	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/amazon-ec2/amazon-ec2-debug.log* files containing debut information which is more complete with a greater debug number
path	No	/opt/bacula	An existing path with enough permissions for File Daemon to create any internal (and usually temporary) plugin file	/mnt/my-vol/	Uses this path to store metadata and temporary files

Fileset Tuning Parameters

The following parameters can be utilized to adjust the plugin's behavior for increased flexibility in challenging network conditions, high job concurrency scenarios, and similar situations.

It is not necessary to modify them for the great majority of the cases:

Option	Require	Default	Values	Example	Description
concurrent_threads	No	500	1-1000	100	Number of maximum concurrent backup threads running in parallel in order to download blocks from a given EBS volume
api_timeout	No	9000	Positive integer (seconds)	2000	Total timeout for AWS API calls
api_read_timeo	No	600	Positive integer (seconds)	5000	Timeout for AWS API calls to respond
general_network_r	No	600	Positive integer (number of retries)	10	Number of retries for failed requests to the AWS API
general_network_d	No	50	Positive integer (seconds)	100	Delay between retries to the AWS API
snapshot_operations	No	1800	Positive integer (seconds)	3600	Timeout for snapshot related operations (like make snapshot) before giving up
instance_operatio	No	1800	Positive integer (seconds)	3600	Timeout for instance related operations (like booting an instance) before giving up
volume_operations	No	1800	Positive integer (seconds)	3600	Timeout for volume related operations (like changing the status of a volume) before giving up

Fileset Advanced Parameters

The following parameters are advanced ones, and they should not be modified in the great majority of cases:

Option	Re-quire	Default	Values	Ex-am-ple	Description
stream_sl	No	1	Positive integer (1/10 sec-conds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_m	No	120	Positive integer (sec-onds)	360	Max wait time for FD to answer packet requests
time_max	No	86400	Positive integer (sec-onds)	4320	Maximum time to wait to ovewrite a debug log that was marked as being used by other process
log-ging_max	No	50MB	String size	300M	Maximum size of a single debug log fileGenerates the working/amazon-ec2/amazon-ec2-debug.log* files containing debut information which is more complete with a greater debug number
log-ging_max	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rollin	No	amazon-ec2.log.%d{ MMM}.log.	String file name pattern	…	Log patter for rotated log files
split_conf	No	=	Charac-ter	:	Character to be used in config_file parameter as separator for keys and values
pub-lisher_qu	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The ".log" files are rotated automatically. Currently, each file can be 50Mb at maximum and the plugin will keep 25 files.
  - This behavior can be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The ".err" file can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like 'logrotate'.
- Backups in parallel and also failed backups will generate several log files. For example: amazon-ec2-debug-0.log, amazon-ec2-debug-1.log…

Fileset Examples

In this section, some fileset examples are presented:

Listing 45: **Fileset: for a selection of instances by name**

```
Fileset {
   Name = fs-amazon-ec2-instances-a-b
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: region=us-east-1 access_key=AKIAQTESTKEY12134g␣
↪secret_key=m23480ahpqwre894qwrffsfdSecretExample instances=myInstanceA,
↪myInstanceB"
   }
}
```

Listing 46: **Fileset: using a config file**

```
Fileset {
   Name = fs-amazon-ec2-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances=myInstanceA"
   }
}

Config file contents in stored in the same File Daemon host in /opt/bacula/
↪etc/amazon-ec2.settings:
region=us-east-1
access_key=AKIAQTESTKEY12134g
secret_key=m23480ahpqwre894qwrffsfdSecretExample
```

Listing 47: **Fileset: Exclude boot volume**

```
Fileset {
   Name = fs-amazon-ec2-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances=myInstanceA exclude_boot_volume=true"
   }
}
```

Listing 48: **Fileset: Backup instances contaning tag bacula**

```
Fileset {
   Name = fs-amazon-ec2-instances
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances_tags=\"bacula\""
```

(continues on next page)

(continued from previous page)

```
   }
}
```

Listing 49: **Fileset: Backup instances contaning tag backup=yes**

```
Fileset {
   Name = fs-amazon-ec2-instances
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances_tags=\"backup=yes\""
   }
}
```

Listing 50: **Fileset: using instance id**

```
Fileset {
   Name = fs-amazon-ec2-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances=i-xxxxmyInstanceId"
   }
}
```

Listing 51: **Fileset: by tag, but exclude A**

```
Fileset {
   Name = fs-amazon-ec2-instances
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances_tags=\"backup=yes\" instances_exclude=myInstanceA"
   }
}
```

Listing 52: **Fileset: Backup specific volume of instance**

```
Fileset {
   Name = fs-amazon-ec2-instance-a-vol-x123
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances=myInstanceA disks=vol-x123asfafafexample"
   }
}
```

Listing 53: **Fileset: Backup one instance, but reduce concurrency**

```
Fileset {
   Name = fs-amazon-ec2-instance-a
```

(continues on next page)

(continued from previous page)

```
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-ec2: config_file=/opt/bacula/etc/amazon-ec2.settings␣
↪instances=myInstanceA concurrent_threads=50"
   }
}
```

Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise Amazon EC2 Plugin**.

Backup

The general backup operation consists on the following procedure:

1. Find all selected target instances.
2. Find all selected target disks for each instance and do for each disk.
3. Cleaning previous snapshots.
4. Check for base snapshots (Incremental and Differential backups).
5. Launch a snapshot without predecessor (Full) or using a found predecessor (Incremental and Differential) for all the target disks at the same time.
6. Download block by block (EBS Direct API) and in parallel, all the blocks belonging to each disk.

Backup File Structure

One instance with one single volume will usually produce the following contents during a backup operation:

`/@amazon-ec2/i-xxxxxx/i-xxxxxx.json`

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy-zzzz-wG.json`

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy-zzzz-wG.bimg`

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy.bmp`

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy.abmp`

Being i-xxxxxx an instance id, all the information of each backed up instance will be contained into a folder with that name. Inside files are explained below:

`/@amazon-ec2/i-xxxxxx/i-xxxxxx.json`

It is a JSON file containing all the configuration data of the instance, such as device mapping, tags or networking among many others.

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy-zzzz-wG.json`

Being vol-yyyyyy the EBS volume id, it is a JSON file containing all the configuration data of the associated volume, such as volume type, iops or throughput among many others. The name will contain also the name of the device inside the instance in ‘zzzz’ (sda for instance) and the size in Gigabytes in ‘w’.

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy-wG.bimg`

It contains the associated volume data in the instant the backup snapshot was done. This file contains headers with offset and size.

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy.bmp`

`/@amazon-ec2/i-xxxxxx/vol-yyyyyy.abmp`

Contains all the headers (offset and size) of the associated disk (.bmp), as well as the headers of the empty blocks with the same format (.abmp).

Changed Block Tracking

The Amazon EC2 API provides a diff function to compare changes at the block level between two Snapshots. That function is what this plugin uses to perform Incremental or Differential backups. As a result, Bacula will need to keep one snapshot at least (so changes can be compared later) in the cloud.

By default, Full backups will keep the snapshot just created during backup. Differential backups won't keep any snapshot. Incremental backups will keep also the last snapshot, but will remove the previous one (also if it's from a Full backup). This makes the combination of Incremental and Differential backups not possible in the same backup chain. If you wish to combine then, then you need to use the parameter: keep_full_snapshot=yes.

Backup with bec2

In addition to a regular backup job, this plugin allows to invoke the same functions through the command line, using the provided ‘bec2’ script. This script will accept the same parameters as the plugin command line and will store the backup in a local folder. Below an example on how to use it:

Listing 54: **BEC2 backup example**

```
bec2 --operation=backup  --region=us-east-1 --access_key=ABCXXXEXAMPLEKEY --
↪secret_key=xxdrperefkafasfdExampleSecret123 --instances=my-instance --
↪bitmap_headers=false
```

That invocation will download a backup of the ‘my-instance’ instance into the working directory.

Restore

Restore Procedure

A restore operation of this plugin will typically involve the following steps:

1. For each involved volume, a new empty snapshot will be created
2. Data will be uploaded to those empty snapshots in an incremental way (Full first, Incremental data after)
3. Snapshots will be completed once everything was uploaded
4. Instance will be created mixing original instance configuration and user specified information during restore session

5. The creation operation will already include the creation of any associated EBS Volume coming from the restored snapshots (except boot volume)
6. Instance will be switched off
7. If boot volume is being restored, default boot volume of created instance (coming from base AMI) will be de-attached and removed
8. Restored snapshot of boot volume will be used to create a new volume that will be attached to the restored instance
9. Instance will be switched on if necessary and all the snapshots used by the restore will be cleaned up

Restore Parameters

Amazon EC2 Plugin is able to do raw restore to any local filesystem mounted over the host where the File Daemon is running, or to the Amazon EC2 environment. The restore method is selected based on the value of the `where` parameter at restore time:

- Empty or '/' (example: `where=/`) → Amazon EC2 restore will be triggered
- Any other path for where (example: `where=/tmp`) → Raw restore to the local file system will be triggered.

This principle generally applies to other Bacula plugins as well. Nevertheless, in this specific plugin, it is advised against using the raw restore method by the File Daemon. This is because the raw restore method retains the headers generated during the backup process in the volume disk files. If you want to restore the filesystem, you can trigger an Amazon EC2 restore using the where=/ parameter. However, you should utilize the restore variable to_local_path and specify the desired destination. By doing so, the mentioned headers will be automatically removed, and the resulting disk image will be suitable for any future purposes.

When using the Amazon EC2 restore method, the following parameters are available to control the restore behavior under 'Plugin Options' menu during a BConsole restore session:

Option	Required	Default	Values
instance_name	No	Original backup value	Existing email address on the target
instance_switchon	No	no	0, no, No, false, FALSE, false, off ;
instance_type	No	Original backup value	Accepted instance type in Amazon
instance_tags	No	Original backup value	List of key value strings: key1=valu
instance_key_name	No	Original backup value	The name of the access key pair tha
instance_generate_new_key	No	no	0, no, No, false, FALSE, false, off ;
instance_no_network	No	no	0, no, No, false, FALSE, false, off ;
instance_keep_ip_addresses	No	no	0, no, No, false, FALSE, false, off ;
instance_security_groups	No	Original backup value	Id of an existing Security Group in t
instance_no_licenses	No	no	0, no, No, false, FALSE, false, off ;
instance_licenses	No	no	List of license key strings separated
instance_placement_tenancy	No	Original backup value	shared ; dedicated ; host ; default
instance_placement_affinity	No	Original backup value	default ; host
instance_placement_groupid	No	Original backup value	String
instance_image_id	No	Original backup value	Id of an existing image in the destin
instance_root_volume_id	No	Original backup value	Id of the volume being restored (or
volume_type	No	Original backup value	Accepted volume type in Amazon E
volume_encrypted	No	Original backup value	0, no, No, false, FALSE, false, off ;

Option	Required	Default	Values
volume_iops	No	Original backup value	Positive integer
volume_throughput	No	Original backup value	Positive integer
availability_zone	No	Original backup value	String: availability zone
image_generate_new	No	No	0, no, No, false, FALSE,
image_name	No		String
image_architecture	No	Original backup value or instance type value	i386 ; x86_64 ; arm64
image_boot_mode	No	Original backup value	uefi ; legacy-bios ; uefi-p
image_virtualization_type	No	Original backup value or instance type value	hvm ; paravirtual
image_ena_support	No	Yes	0, no, No, false, FALSE,
image_imds_support	No	Original backup value	0, no, No, false, FALSE,
image_tpm_support	No	Original backup value	0, no, No, false, FALSE,
access_key	No	Original backup value	String
secret_key	No	Original backup value	String
region	No	Original backup value	String: region code existi
to_local_path	No		Accessible and existing l
from_local_path	No		Accessible and existing l
dry_run	No	no	0, no, No, false, FALSE,
debug	No	0	0, 1, 2 ,3, 4, 5, 6, 7, 8, 9

Restore Use Cases

The following restore scenarios are supported, and steps to execute them are described:

a) Restore an instance to Amazon EC2, to its original location, and keeping all its configuration :

   1. Run a restore session selecting appropriate backup jobs
   2. Select all the contents of the instance directory (named with the instance id: i-xxxxxxxx/)
   3. Use `Where=/`

Important: Amazon EC2 Plugin uses the name (the value of the tag 'Name') to decide if an instance is existing. It means that for a) case to happen it is needed that the original instance with the original name is no more there or to have a different name.

b) Restore an instance to Amazon EC2 with a different name:

   - Follow previously described 'a' scenario.
   - Set `instance_name` with the value of the desired new name

c) Restore an instance to Amazon EC2, but generate a new access key for it:

- Follow previously described 'a' or 'b' scenarios
- Before confirming the restore operation, set `instance_generate_new_key` with `yes`
- Run the restore. The joblog will show the associated secret to the generated new key

d) Restore an instance to Amazon EC2, but adjust any configuration parameter of it (advanced):

- Follow previously described 'a', 'b' or 'c' scenarios.
- Adjust any desired parameter among below list:
- Underlying hardware parameters: `instance_type`, `instance_placement_tenancy`, `instance_placement_affinity`, `instance_placement_groupid`

- Networking and security: `instance_security_groups`, `instance_key_name`, `instance_keep_ip_addresses`
- Customization of the instance: `image_id`, `instance_tags`, `instance_no_licenses`
- Please, note that the configuration you set here will be applied as you put it. Therefore, the consistency of that configuration will depend on everything being correct (existing and consistent) at Amazon EC2 side for your particular infrastructure

e) Restore an instance to Amazon EC2, but to a different location:

- Follow previously described 'a', 'b', 'c' or 'd' scenarios.
- Adjust `region`, `access_key`, `secret_key` with the destination values

f) Restore an instance to Amazon EC2, but adjust any configuration parameter of the selected EBS volumes (advanced):

- Follow previously described 'a', 'b', 'c', 'd' or 'e' scenarios.
- Adjust `volume_type`, `volume_encrypted`, `volume_iops`, `volume_throughput` with the desired values
- Please, note that the configuration you set here will be applied as you put it. Therefore, the consistency of that configuration will depend on everything being correct (existing and consistent) at Amazon EC2 side for your particular infrastructure

g) Restore a volume to an existing instance:

1. Run a restore session selecting appropriate backup jobs
2. Select the desired volume yyyyy.bimg file(s) inside a given i-xxxxxxx/ folder
3. Use `Where=/`
4. Set `instance_name` with the value of an existing instance in the destination region that will hold the restored volumes

h) Restore instance files or volume files to a local directory:

1. Run a restore session selecting appropriate backup jobs
2. Select the desired files (or all of them) inside a given i-xxxxxxx/ folder
3. Use `Where=/`
4. Adjust `to_local_path` to the desired path

i) Restore just an instance without any disk (use image root disk):

1. Run a restore session selecting appropriate backup jobs
2. Select the just the i-xxxxx.json file
3. Use `Where=/`

j) Restore a complete instance from a local directory with bec2 script:

1. Run bec2 command using a local folder containing a full instance backup like the example below, and using the desired name for the restored instance:
2. bec2 –operation=restore –region=us-east-1 –access_key=ABCXXXEXAMPLEKEY –secret_key=xxdrperefkafasfdExampleSecret123 –from_local_path=/my-local-path/amazon-ec2/i-xxxxxxx__my-instance –instance_name=RestoredInstanceName

Restore Example Session

In the following restore example session, we restore a given instance with a new name.

Note: It is also possible to run backup or restore operations from any of the Bacula Graphical User Interfaces.

Listing 55: **Restore bconsole session**

```
restore jobid=1 Client=127.0.0.1-fd where="/"
Using Catalog "MyCatalog"
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
5 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd "/@amazon-ec2/"
cwd is: /@amazon-ec2/
$ dir
----------   0 root     root               0  1970-01-01 01:00:00  /@amazon-
↪ec2/i-0c762632709fa24fa__REGRESS_20231115155818/
$ mark *
5 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

   TEST-2023-11-15:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


5 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
```

(continues on next page)

(continued from previous page)

```
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-11-15 16:04:18
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: Fileset
   5: Restore Client
   6: When
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : amazon-ec2: region="us-east-1" access_key=
↪"AKIAQVXBC4DMSVXJERBX" secret_key="mETUHxdz8vjA8DyWLmzFSaJDudXz8Nyh1J4FW3vP
↪" instances="REGRESS_20231115155818" debug=6
Plugin Restore Options
Option                         Current Value        Default Value
instance_name:                 *None*               (*None*)
instance_switchon:             *None*               (*None*)
instance_type:                 *None*               (*none*)
instance_security_groups:      *None*               (*none*)
instance_tags:                 *None*               (*none*)
instance_key_name:             *None*               (*none*)
instance_generate_new_key:     *None*               (*none*)
instance_no_network:           *None*               (*none*)
instance_keep_ip_addresses:    *None*               (*none*)
instance_no_licenses:          *None*               (*none*)
instance_placement_tenancy:    *None*               (*none*)
instance_placement_affinity:   *None*               (*none*)
instance_placement_groupid:    *None*               (*none*)
instance_image_id:             *None*               (*none*)
volume_type:                   *None*               (*none*)
volume_encrypted:              *None*               (*none*)
volume_iops:                   *None*               (*none*)
volume_throughput:             *None*               (*none*)
availability_zone:             *None*               (*None*)
access_key:                    *None*               (*None*)
secret_key:                    *None*               (*None*)
region:                        *None*               (*None*)
to_local_path:                 *None*               (*None*)
from_local_path:               *None*               (*None*)
dry_run:                       *None*               (*None*)
```

(continues on next page)

(continued from previous page)

```
debug:                          *None*               (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_name (Set the name for a new restored instance, or specify an␣
↪existing name or id where you want to attach the volumes)
   2: instance_switchon (Start the instance just after restoring it)
   3: instance_type (Set the instance type (c1.medium, a1.large...))
   4: instance_security_groups (Set a new set of security groups (ids␣
↪separated by ','))
   5: instance_tags (Add extra tags to the restored instance(s)␣
↪(tag1key:tag1value,tag2key:tag2value...))
   6: instance_key_name (Set the name of the access key to access the new␣
↪created instance(s))
   7: instance_generate_new_key (Generate a new access key to access the new␣
↪created instance(s))
   8: instance_no_network (Do not generate any network configuration for the␣
↪restore instance(s))
   9: instance_keep_ip_addresses (Keep original IPv4 or IPv6 addresses of the␣
↪instance(s))
   10: instance_no_licenses (Do not restore the licenses configuration to the␣
↪restored instance(s))
   11: instance_placement_tenancy (Set up the tenancy value for the␣
↪destination placement of the restored instance(s))
   12: instance_placement_affinity (Set up the affinity value for the␣
↪destination placement of the restored instance(s))
   13: instance_placement_groupid (Set up the group value for the destination␣
↪placement of the restored instance(s))
   14: instance_image_id (Set a different base imageId)
   15: volume_type (Set up the volume type for the restored disk image(s))
   16: volume_encrypted (Encrypt the restored volume(s))
   17: volume_iops (Iops value for the restored volume(s))
   18: volume_throughput (Throughput value for ther restored volume(s))
   19: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   20: access_key (Set a different access key to access to the destination)
   21: secret_key (Set a different secret key to access to the destination)
   22: region (Set the destination region)
   23: to_local_path (Local path to restore the information, including the␣
↪disk images without bitmap headers)
   24: from_local_path (Local path to restore the information from it,␣
↪instead of the Bacula job)
   25: dry_run (Do not actually create any object in Amazon EC2)
   26: debug (Change debug level)
Select parameter to modify (1-26): 1
Please enter a value for instance_name: REGRESS_20231115160418
Plugin Restore Options
Option                          Current Value        Default Value
instance_name:                  REGRESS_20231115160418 (*None*)
instance_switchon:              *None*               (*None*)
instance_type:                  *None*               (*none*)
instance_security_groups:       *None*               (*none*)
instance_tags:                  *None*               (*none*)
```

(continues on next page)

(continued from previous page)

```
instance_key_name:              *None*                  (*none*)
instance_generate_new_key:      *None*                  (*none*)
instance_no_network:            *None*                  (*none*)
instance_keep_ip_addresses:     *None*                  (*none*)
instance_no_licenses:           *None*                  (*none*)
instance_placement_tenancy:     *None*                  (*none*)
instance_placement_affinity:    *None*                  (*none*)
instance_placement_groupid:     *None*                  (*none*)
instance_image_id:              *None*                  (*none*)
volume_type:                    *None*                  (*none*)
volume_encrypted:               *None*                  (*none*)
volume_iops:                    *None*                  (*none*)
volume_throughput:              *None*                  (*none*)
availability_zone:              *None*                  (*None*)
access_key:                     *None*                  (*None*)
secret_key:                     *None*                  (*None*)
region:                         *None*                  (*None*)
to_local_path:                  *None*                  (*None*)
from_local_path:                *None*                  (*None*)
dry_run:                        *None*                  (*None*)
debug:                          *None*                  (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_name (Set the name for a new restored instance, or specify an␣
↪existing name or id where you want to attach the volumes)
   2: instance_switchon (Start the instance just after restoring it)
   3: instance_type (Set the instance type (c1.medium, a1.large...))
   4: instance_security_groups (Set a new set of security groups (ids␣
↪separated by ','))
   5: instance_tags (Add extra tags to the restored instance(s)␣
↪(tag1key:tag1value,tag2key:tag2value...))
   6: instance_key_name (Set the name of the access key to access the new␣
↪created instance(s))
   7: instance_generate_new_key (Generate a new access key to access the new␣
↪created instance(s))
   8: instance_no_network (Do not generate any network configuration for the␣
↪restore instance(s))
   9: instance_keep_ip_addresses (Keep original IPv4 or IPv6 addresses of the␣
↪instance(s))
   10: instance_no_licenses (Do not restore the licenses configuration to the␣
↪restored instance(s))
   11: instance_placement_tenancy (Set up the tenancy value for the␣
↪destination placement of the restored instance(s))
   12: instance_placement_affinity (Set up the affinity value for the␣
↪destination placement of the restored instance(s))
   13: instance_placement_groupid (Set up the group value for the destination␣
↪placement of the restored instance(s))
   14: instance_image_id (Set a different base imageId)
   15: volume_type (Set up the volume type for the restored disk image(s))
   16: volume_encrypted (Encrypt the restored volume(s))
   17: volume_iops (Iops value for the restored volume(s))
   18: volume_throughput (Throughput value for ther restored volume(s))
```

(continues on next page)

(continued from previous page)

```
   19: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   20: access_key (Set a different access key to access to the destination)
   21: secret_key (Set a different secret key to access to the destination)
   22: region (Set the destination region)
   23: to_local_path (Local path to restore the information, including the␣
↪disk images without bitmap headers)
   24: from_local_path (Local path to restore the information from it,␣
↪instead of the Bacula job)
   25: dry_run (Do not actually create any object in Amazon EC2)
   26: debug (Change debug level)
Select parameter to modify (1-26): 2
Please enter a value for instance_switchon: true
Plugin Restore Options
Option                         Current Value        Default Value
instance_name:                REGRESS_20231115160418 (*None*)
instance_switchon:             true                 (*None*)
instance_type:                 *None*               (*none*)
instance_security_groups:      *None*               (*none*)
instance_tags:                 *None*               (*none*)
instance_key_name:             *None*               (*none*)
instance_generate_new_key:     *None*               (*none*)
instance_no_network:           *None*               (*none*)
instance_keep_ip_addresses:    *None*               (*none*)
instance_no_licenses:          *None*               (*none*)
instance_placement_tenancy:    *None*               (*none*)
instance_placement_affinity:   *None*               (*none*)
instance_placement_groupid:    *None*               (*none*)
instance_image_id:             *None*               (*none*)
volume_type:                   *None*               (*none*)
volume_encrypted:              *None*               (*none*)
volume_iops:                   *None*               (*none*)
volume_throughput:             *None*               (*none*)
availability_zone:             *None*               (*None*)
access_key:                    *None*               (*None*)
secret_key:                    *None*               (*None*)
region:                        *None*               (*None*)
to_local_path:                 *None*               (*None*)
from_local_path:               *None*               (*None*)
dry_run:                       *None*               (*None*)
debug:                         *None*               (*None*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-11-15 16:04:18
```

(continues on next page)

(continued from previous page)

```
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
Job queued. JobId=3
```

Listing 56: **Restore job result**

```
llist joblog jobid=3
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-dir JobId 3: Start Restore Job RestoreFiles.2023-11-15_16.
↪04.18_09

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-dir JobId 3: Restoring files from JobId(s) 1

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-dir JobId 3: Connected to Storage "File" at 127.0.0.1:8103␣
↪with TLS

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-dir JobId 3: Using Device "FileStorage" to read.

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-dir JobId 3: Connected to Client "127.0.0.1-fd" at 127.0.0.
↪1:8102 with TLS

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-fd JobId 3: Connected to Storage at 127.0.0.1:8103 with TLS

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-sd JobId 3: Ready to read from volume "TEST-2023-11-15:0"␣
↪on Dedup device "FileStorage" (/tmp/regress/tmp).

   time: 2023-11-15 16:04:20
logtext: 127.0.0.1-sd JobId 3: Forward spacing Volume "TEST-2023-11-15:0" to␣
↪addr=268

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Plugin log of this job available␣
↪in: /tmp/regress/working/amazon-ec2/amazon-ec2-debug-0.log

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Jar Version: 1.0.0

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Starting backend restore process

   time: 2023-11-15 16:04:21
```

(continues on next page)

(continued from previous page)

```
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Restoring to Amazon EC2 service

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Restoring from Bacula stream

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Instance name: REGRESS_
↪20231115160418

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Instance type was not set. We will␣
↪use the original one from each instance of the backup

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Instance switch-on: enabled

   time: 2023-11-15 16:04:21
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Availability Zone was not set. We␣
↪will use the original one from each volume

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Volume Encrypted not set, we will␣
↪use the value from each original volume

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Volume Iops was not set. We will␣
↪use the value from each original volume

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Volume Throughput was not set. We␣
↪will use the value from each original volume

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Volume Type was not set. We will␣
↪use the value from each original volume

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Generate New Key: disabled

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Key Name was not set. We will use␣
↪the value from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Image Id was not set. We will use␣
↪the value from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: No network: disabled

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: No licenses: disabled
```

(continues on next page)

(continued from previous page)

```
   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Placement Affinity was not set. We␣
↪will use the value from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Placement Group Id was not set. We␣
↪will use the value from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Placement Tenancy was not set. We␣
↪will use the value from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Security Groups were not set. We␣
↪will use the values from each original instance

   time: 2023-11-15 16:04:22
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: No extra tags will be added to␣
↪restored instances

   time: 2023-11-15 16:04:36
logtext: 127.0.0.1-sd JobId 3: Elapsed time=00:00:16, Transfer rate=110.3 M␣
↪Bytes/second

   time: 2023-11-15 16:04:36
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Restoring disk:vol-
↪0fcff2bdcafb6cdf5 data to new snapshot

   time: 2023-11-15 16:04:36
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Started snapshot:snap-
↪0efed219eca86c531 for original volume:vol-0fcff2bdcafb6cdf5

   time: 2023-11-15 16:04:36
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Sending data to destination for␣
↪original volume:vol-0fcff2bdcafb6cdf5 ...

   time: 2023-11-15 16:04:36
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: 3365 blocks were uploaded to␣
↪volume:vol-0fcff2bdcafb6cdf5

   time: 2023-11-15 16:04:42
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Completing snapshot:snap-
↪0efed219eca86c531 with checksum:EzDgqUll+NZrc9sE8GivEP8l4Nj+C/
↪fQAKGoAKvwSpE= ...

   time: 2023-11-15 16:05:13
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Snapshot snap-0efed219eca86c531:␣
↪is completed

   time: 2023-11-15 16:05:58
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Finalizing volume restore for␣
```

(continues on next page)

(continued from previous page)

```
↪original volume:vol-0fcff2bdcafb6cdf5 ...

   time: 2023-11-15 16:05:58
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Creating new instance from␣
↪instance:i-0c762632709fa24fa ...

   time: 2023-11-15 16:06:30
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Instance i-0a267ee7035475860: is␣
↪running

   time: 2023-11-15 16:06:30
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Stopping instance:i-
↪0a267ee7035475860 ...

   time: 2023-11-15 16:07:02
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Successfully stopped instance: i-
↪0a267ee7035475860

   time: 2023-11-15 16:07:02
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Dettaching and deleting␣
↪automatically generated root volume for instance:i-0a267ee7035475860 ...

   time: 2023-11-15 16:07:02
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Root volume of instance:i-
↪0a267ee7035475860 is:vol-0f5649a3d860669ce

   time: 2023-11-15 16:07:13
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Successfully deleted volume: vol-
↪0f5649a3d860669ce

   time: 2023-11-15 16:07:13
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Creating and attaching root volume␣
↪from snapshot:snap-0efed219eca86c531 to instance:i-0a267ee7035475860

   time: 2023-11-15 16:07:19
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Volume vol-0192896b074dad9e2: is␣
↪available

   time: 2023-11-15 16:07:19
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Cleaning created snapshots during␣
↪the restore process

   time: 2023-11-15 16:07:19
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Deleting snapshot:snap-
↪0efed219eca86c531 ...

   time: 2023-11-15 16:07:20
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Starting instance:i-
↪0a267ee7035475860...

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: Successfully started instance: i-
```

(continues on next page)

(continued from previous page)

```
↪0a267ee7035475860

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-fd JobId 3: amazon-ec2: No more items to restore. Restore␣
↪ended

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: Bacula 127.0.0.1-dir 16.0.7 (05Oct23):
Build OS:               x86_64-pc-linux-gnu ubuntu 22.04
JobId:                  3
Job:                    RestoreFiles.2023-11-15_16.04.18_09
Restore Client:         "127.0.0.1-fd" 16.0.7 (05Oct23) x86_64-pc-linux-gnu,
↪ubuntu,22.04
Where:                  /
Replace:                Always
Start time:             15-nov-2023 16:04:20
End time:               15-nov-2023 16:07:51
Elapsed time:           3 mins 31 secs
Files Expected:         5
Files Restored:         5
Bytes Restored:         1,764,313,223 (1.764 GB)
Rate:                   8361.7 KB/s
FD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            Restore OK

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: Begin pruning Jobs older than 6 months .

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: No Jobs found to prune.

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: Begin pruning Files.

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: No Files found to prune.

   time: 2023-11-15 16:07:51
logtext: 127.0.0.1-dir JobId 3: End auto prune.

         jobid: 3
            job: RestoreFiles.2023-11-15_16.04.18_09
            name: RestoreFiles
   purgedfiles: 0
            type: R
         level: F
      clientid: 1
      clientname: 127.0.0.1-fd
      jobstatus: T
   jobstatuslong: Completed successfully
```

(continues on next page)

(continued from previous page)

```
      schedtime: 2023-11-15 16:04:18
      starttime: 2023-11-15 16:04:20
         endtime: 2023-11-15 16:07:51
   realendtime: 2023-11-15 16:07:51
   realstarttime: 2023-11-15 16:04:20
      jobtdate: 1,700,060,871
   volsessionid: 3
volsessiontime: 1,700,060,507
      jobfiles: 5
      jobbytes: 1,764,313,223
      readbytes: 1,764,312,106
      joberrors: 0
jobmissingfiles: 0
         poolid: 0
      poolname:
      priorjobid: 0
      priorjob:
      filesetid: 0
         fileset:
      hascache: 0
         comment:
      reviewed: 0
   isvirtualfull: 0
            rate: 8361.7
   compressratio: 0
      statusinfo:
   writestorage:
   writedevice:
lastreadstorage: File
lastreaddevice: FileStorage
```

Cross Hypervisor Restore and Virtual Machine Migration

Bacula Amazon EC2 Plugin is designed to provide also generic restore operations using RAW data from disk images. Those data disks can come from different Bacula Enterprise plugins working at disk image level, this is IaaS or Hypervisor Plugins, as well as any other technology producing disks in this RAW format.

Bacula Amazon EC2 Plugin works at the block level, so it is capable of reading and uploading just the blocks where the real information resides. As a result, migrations done with it are *much more faster* than any other done through the import method described by Amazon, where all the data needs to be uploaded first to a S3 bucket, before the import operation is called. More information: https://aws.amazon.com/ec2/vm-import/

A migration operaton is just a restore operation calling the bec2 script, where the parameter 'from_local_path' points to the folder or file where the data disks are placed. Then, using all the instance_* and image_* parameters, the migrated instance can be configured. Below an example:

Listing 57: **Restore job result**

```
/opt/bacula/bin/bec2 --operation=restore  --region=us-east-1 --access_
```

(continues on next page)

(continued from previous page)

```
↪key=XXXXXX --secret_key=YYYYYYY --from_local_path=disk-0.bvmdk --instance_
↪name=FromVMware1 --instance_switchon=true --instance_security_groups=sg-
↪XXXXX --instance_root_volume_id=disk-0.bvmdk --image_generate_new=true
```

BEC2 will try to use files with '.img' or '.bvmdk' extensions. Contents must be in RAW format. If the format is different, normally it's possible to convert it using tools like 'qemu-img convert': https://qemu-project.gitlab.io/qemu/tools/qemu-img.html

If you plan to use bec2 tool to migrate Virtual Machines coming from other virtualization systems, some planning and preparation is needed:

The safest restore method for migration is to restore instances in Amazon EC2 using supported images and default root/boot volumes that come with them. After, data to be migrated can be attached as a new volume. To use this method, bec2 restore operation needs to leave emtpy the parameter 'instance_root_volume_id'.

When it comes to root volumes, it is possible to find inconveniences with booting or the networking after the instance was created in Amazon EC2. Amazon EC2 has some prerequisites to have Virtual Machines correctly working when they are created outside the service, detailed here: https://docs.aws.amazon.com/vm-import/latest/userguide/prerequisites.html

On the other hand, the internal configuration of the Virtual Machine has also to comply with some conditions, detailed here: https://docs.aws.amazon.com/vm-import/latest/userguide/prepare-vm-image.html

It is key to understand that if you plan to use your backups from a different environment as something to migrate to Amazon EC2, you must prepare the target Virtual Machines *before* the backup is performed. For example, if the VM has LVM (Logical Volume Manager) or some other layer of complexity, it may not boot properly once migrated.

We recommend to check these requirements as they can be changed by Amazon anytime. We list here an example on the actions that could be done on a given Linux VM to be migrated:

Listing 58: **Linux VM Preparation**

```
1. Check fstab entries are using UUID=xxx

2. Check dhcp client is enabled

3. Disable predictable Network interfaces (isn0...):

   In /etc/default/grub
        GRUB_CMDLINE_LINUX_DEFAULT="net.ifnames=0 biosdevname=0"

   In /etc/sysconfig/network-scripts
        cp ifcfg-ens192 ifcfg-eth0
   -> Adapt the new script with the proper network interface name (eth0 here)

4. Check grub config exists and it's done with UUID

     grep root /boot/grub2/grub.cfg

5. Disable any antivirus

6. Uninstall VMWare tools
```

(continues on next page)

(continued from previous page)

```
7. Add xen, nvme and ena modules to initram

   cat /etc/dracut.conf
   # additional kernel modules to the default
   add_drivers+="xen-blkfront xen-netfront nvme-core nvme ena virtio"

   Update grub and initram
   grub2-mkconfig -o /boot/grub2/grub.cfg
   dracut -f -v


8. Enable access for non-root user with public ssh keys

   ssh-keygen -t rsa -b 4096
   touch .ssh/authorized_keys
   chmod 600 .ssh/authorized_keys
   # Enable SSH service
   # Copy .ssh/id_rsa to source machine from which we want to connect
   ...
```

If a Virtual Machine was not prepared and you are still trying to migrate it with bec2, it is recommended to not use the original boot disk as root volume and just use a native boot volume from a standard Amazon AMI while attaching the other disks to it. Once connecting to the new instance, data can be moved manually to the new boot disk. Another alternative would be to try to modify the data contained in the RAW disk before attempting the restore with a tool like 'virt-customize' (https://libguestfs.org/virt-customize.1.html).

Once you have your virtual machines correctly prepared, a migration operation will typically consist on:

1. Restore your Virtual Machine to the local filesystem (in general, use where=/local/fs/path).
2. Convert the data with qemu-img to RAW format. This is not always needed (for instance, VMWare Plugin restores will be directly in RAW)
3. Run bec2 tool pointing to the restored disks

Image(AMI) and root volume options

For Virtual Machine migration operations root volume and image management options are more relevant compared to regular restores coming from Amazon EC2 Plugin backups.

The different options to restore images are described below:

a. Use an existing image in Amazon EC2 and attach migrated data disks to it:

- Use 'instance_root_volume_id' and 'instance_image_id' options.
- Use 'instance_name' with a non existing instance name in Amazon EC2

b. Create a new image in Amazon EC2, using one of the volumes as root disk, while attaching the others after to the generated instance:

- Use 'instance_root_volume_id', enable 'instance_generate_new' and any ohter 'instance_*' desired option.
- Use 'instance_name' with a non existing instance name in Amazon EC2

c. Use one of the data disks as root disk, while attaching the others, but do not create a new image:
   - Use 'instance_root_volume_id' with the volume that should be root
   - Use 'instance_name' with a non existing instance name in Amazon EC2

d. Use an existing instance and attach our data disks to without modifying root volume
   - Use 'instance_name' with the existing instance name in Amazon EC2

e. Use an existing instance and attach our data disks to while replacing root volume
   - Use 'instance_name' with the existing instance name in Amazon EC2
   - Use 'instance_root_volume_id' with the volume that should be root

Connection key options

When restoring a root disk, all the information contained for booting is holded by the data coming from the disk image provided. Therefore, in all associated restore scenarios, any option related with the SSH keys won't take any effect (instance_key_name, instance_generate_new_key).

Networking options

Bacula Amazon EC2 Plugin will take backup values by default, but they can be fully personalized. Similarly, when using a RAW disk without any other metadata, default values will be used for networking, such as default subnet and default security group for the selected destination region. If you don't specify your own values, please be aware that normally default Security Groups have no SSH or any other access rule allowed, so you may find difficulties to connect to your new instance.

See also:

Go back to:

- EC2OperationBackup

Go to:

- EC2ListAndQuery

Go back to the AmazonEC2Operations article.

Go back to the *Amazon EC2 plugin main page*.

Go back to the *main Dedicated Backup Solutions page*.

List & Query

It is possible to list instances using the bconsole `.ls` command or through a `.query` command.

We can use any fileset parameter in the plugin="" value in order to filter the results, same way the backup does.

Below, there are some examples:

List instances:

Listing 59: **Query example: Get one instance details**

```
.query plugin="amazon-ec2: instances=MyTestInstance region=us-east-1 access_
↪key=AKIAQTESTKEY12134g secret_key=m23480ahpqwre894qwrffsfdSecretExample"␣
↪client=127.0.0.1-fd parameter="instance"
instance=MyTestInstance
instanceId=i-0a267ee7035475860
imageId=ami-0f34c5ae932e6f0e4
instanceType=t2.micro
hypervisor=xen
state=running
publicIpAddress=34.207.94.177
launchTime=2023-11-15 15:07:20
```

List all instances:

Listing 60: **Query example: Get all instance details**

```
*.query client=127.0.0.1-fd parameter="instance" plugin="amazon-ec2:␣
↪region=us-east-1 access_key=AKIAQTESTKEY12134g secret_
↪key=m23480ahpqwre894qwrffsfdSecretExample"
instance=jg-aws
instanceId=i-0bc4a027596016c44
imageId=ami-0f34c5ae932e6f0e4
instanceType=t2.micro
hypervisor=xen
state=stopped
launchTime=2023-11-06 11:48:08
instance=va-dedup2-aws
instanceId=i-098e49a028ef38588
imageId=ami-026ebd4cfe2c043b2
instanceType=t2.micro
hypervisor=xen
state=stopped
launchTime=2023-11-10 23:16:37
instance=REGRESS_20231129105019
instanceId=i-0028cbc698389b835
imageId=ami-0f34c5ae932e6f0e4
instanceType=t2.micro
hypervisor=xen
state=pending
publicIpAddress=3.82.114.237
launchTime=2023-11-29 09:52:08
instance=va-dedup2-t3-tst
instanceId=i-0e4d9a67884a2cb50
imageId=ami-05a5f6298acdb05b6
instanceType=t3a.medium
hypervisor=xen
state=running
publicIpAddress=3.94.202.65
launchTime=2023-11-09 17:20:22
...
```

Best Practices

The following article presents best practices regarding jobs distribution, concurrency and performance.

Jobs Distribution

It is recommended to split the target backup between different instances, selecting them by type, tags or even having one job per instance.

Following this recommendation, errors in one job will not invalidate a whole backup cycle, where some instances will be backed up successfully, even if some others experienced errors. This also makes it easier to identify the cause of any error.

Concurrency

When using Amazon EC2 APIs, it is possible to find a variety of boundaries that need to be considered. We highlight some of them below:

- Amazon EC2 Throttling: https://docs.aws.amazon.com/AWSEC2/latest/APIReference/throttling.html
- Capabilities of the host serving the EC2 Service
- Usage of the service during the backup window

If a boundary is crossed, the corresponding request will usually fail. Bacula Amazon EC2 Plugin is prepared to wait some amount of time and retry it, so it has a certain level of resiliency. However, it is crucial to plan an adequate strategy to backup all the elements without needing to reach any boundary on a regular basis. This means to control how many concurrent requests are done during the backup window.

A single job implements parallelism which can be reduced or increased if necessary, using the following parameter:

- `concurrent_threads` which controls the number of simultaneous processes fetching and downloading data (EBS blocks). This can be reduced or increased to directly affect the concurrency level of a single job.

The recommended strategy to backup a new environment is to plan a step-by-step testing scenario before putting it into production, where the number of instances and the concurrency of the jobs are increased progressively. Other important point is the timing schedule as some boundaries are related to time-frames (number of request per amount of time). If you detect you reach boundaries when running all your backups during a single day of the week, try to increase the time window, and spread the load through it in order to achieve better performance results.

Note that from an architectural and AWS service point of view, you can also consider to:

- Run your File Daemon directly in the cloud (if your SD is also in the cloud)
- Run your Storage Daemon and File Daemon in the same host, so you skip one network hop in the process (recommended)
- Use a dedicated AWS connection (https://aws.amazon.com/directconnect/)
- Increase throttling limits in AWS (https://docs.aws.amazon.com/AWSEC2/latest/APIReference/throttling.html#throttling-increase)

Performance

The performance of this plugin is highly dependent on many external factors:

- Latency and bandwidth to AWS
- Network infrastructure
- FD Host hardware
- FD Load
- Ratio number of elements/size
- And many more.

In summary, it is not possible to establish an exact reference about how much time a backup will need to complete.

As a reference and regarding the number of instances and their size:

- Many small instance volumes to protect: More volumes per time period, but smaller speed (MB/s).
- Big instance volumes to protect: Fewer volumes per time period, but greater speed (MB/s).

Consider also that any instance backup needs some time to make snapshot cleanup previously and after the backup, as well as the time Amazon EC2 needs to make the snapshots, list elements or calculate differences between two snapshots.

It is recommended to benchmark your own environment in base to your requirements and needs.

The automatic parallelization mechanism (using `concurrent_threads=x`) should work well for most scenarios, however, fine-tune is possible if we define one job per instance. We could control how many jobs to run in parallel, and then decrease the `concurrent_threads` value, if needed, in order to avoid throttling from EBS service.

There are many possible strategies to use this plugin, so it is recommended to study what suits your needs best before deploying the jobs in your entire environment, so you can get the best possible results:

- You can have a job per instance
- You can have multiple instances per job or even all your instances in the same job (not recommended)
- You can split your workload through a schedule, or try to run all your jobs together
- You can run jobs in parallel or take advantage of `concurrent_threads` and so, run less jobs in parallel
- You can specifically select the instances you want to backup or backup them all
- You can specifically select the volumes you want to backup or backup them all
- You can include boot volumes in your backup or exclude them.
- You can run your File Daemon on premise or in the cloud
- You can use default internet connection to AWS or use a dedicated AWS connection (https://aws.amazon.com/directconnect/)
- You can use default throttling limits, or ask AWS for more (https://docs.aws.amazon.com/AWSEC2/latest/APIReference/throttling.html#throttling-increase)
- You can run your Storage Daemon and File Daemon in the same host, so you skip one network hop in the process (recommended).

Limitations

The following article presents limitations of Amazon EC2 Plugin.

Application consistency

By default, Amazon EBS snapshots are crash consistent, not application consistent. However, AWS allows for configuration to achieve application consistency, as detailed in the following article: https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/application-consistent-snapshots.html

An alternative method to ensure application consistency is by enabling the backup parameter "shutdown". This will shut down the associated instances before taking the snapshot and restart them afterward. It is important to note that this process will result in a service disruption for the associated instances during the snapshot creation period. Additionally, if any issues occur with the Bacula service, File Daemon operations are terminated, or network connectivity is lost, the instances may not be able to restart. Therefore, this feature should be utilized with caution and careful consideration.

Troubleshooting

In this article, there are suggested solutions to common situations that can cause trouble during the usage of the Amazon EC2 Plugin.

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (you will find them in the `amazon-ec2-debug.err` file), you are very likely using a high level of concurrency through internal `concurrent_threads` parameter and/or parallel jobs.

To overcome this situation you can:

a) Reduce `concurrent_threads` parameter.

b) Reduce the number of jobs running in parallel.

c) If you cannot do that, you should increase JVM memory (you may need to also increase the underlying host physical memory)

To increase JVM memory, you will need to:

Create this file: `/opt/bacula/etc/AMAZON_EC2_backend.conf`.

Below, an example of the contents:

Listing 61: **Limits file contents**

```
AMAZON_EC2_JVM_MIN=2G
AMAZON_EC2_JVM_MAX=8G
```

Those values will define the MIN (AMAZON_EC2_JVM_MIN) and MAX (AMAZON_EC2_JVM_MAX) memory values assigned to the JVM Heap size. In Be careful if you are running jobs in parallel, as very big values and several jobs at a time could quickly eat all the memory of your host.

The `/opt/bacula/etc/AMAZON_EC2_backend.conf` won't be modified through package upgrades, so your memory settings will be persistent.

Throttling

If throttling usually happens in your backup window you are using more concurrency than the one you are allowed to in your EC2 environment.

Possible solutions are to:

- Reduce concurrency with the different strategies mentioned in Best Practices section
- Increase throttling limits: https://docs.aws.amazon.com/AWSEC2/latest/APIReference/throttling.html#throttling-increase

4 Applications

Important: Application solutions are used with the File Daemon.

4.1 On-premise

Exchange Plugins

Microsoft Exchange is a messaging and collaborative platform that enables businesses to communicate, share information and collaborate securely and efficiently. It provides email, calendaring, task management, and contacts functionalities, as well as allows users to access their data from anywhere with Internet connection. Exchange is used by organizations of all sizes, from small businesses to large enterprises, and is a critical part of many companies' communication infrastructure. It is available both as an on-premises solution and as a cloud-based service through Microsoft 365 Software as a Service platform. Note that the functionality described here is all about traditional on-premise Exchange instances, *not* about Microsoft 365 Software-as-a-Service email functionality.

Bacula Enterprise Exchange EWS Plugin is a tool to provide backup and restore operations at item level of elements managed by a Microsoft Exchange platform, meaning getting, downloading, cataloging and restoring individual emails, attachments, calendar events, tasks or contacts. For information about Exchange EWS Plugin, refer to:

- *Exchange EWS Plugin*

Bacula Enterprise also provides a traditional plugin to backup and restore Exchange instances through a different plugin, based on Windows VSS technology that is capable of working at a database level. That option is recommended for disaster recovery strategies, while the plugin discussed in the current document is intended to be used for item level capabilities. For information about Exchange Plugin based on VSS, refer to:

- *Exchange VSS Plugin*

In order to have the best protection level for a Microsoft Exchange on-premise instance, Bacula Systems recommends using a combined strategy of the two plugins. Exchange EWS Plugin can be used to restore small pieces of lost information at a user level. However, in case of suffering a disaster where the whole Exchange deployment has been impacted, the recommended protection will be given by the Exchange Plugin, based on VSS, where the speed to recover a full database will be superior.

Exchange EWS Plugin

The following article aims at presenting the reader with information about the **Bacula Enterprise Exchange EWS Plugin** (Exchange Plugin based on EWS - Exchange Web Services). The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features.

Scope

Bacula Enterprise Exchange Plugin currently supports the following platforms:

- Exchange Server 2019
- Exchange Server 2016
- Exchange Server 2013 with Service Pack 1 or later.

The underlying version of the Windows operative system can be any of the supported ones associated to each Exchange Server product version according to the official Microsoft documentation. As an example, at the time of writing this document, this information can be found here: https://learn.microsoft.com/en-us/exchange/plan-and-deploy/system-requirements?view=exchserver-2019 .

This plugin is available since **Bacula Enterprise 16.2**, and needs to be deployed in a Linux host.

Features

The main feature of **Bacula Enterprise Exchange EWS Plugin** is to offer backup and restore of Exchange Server environments at item level, which is the major possible granularity for Exchange services. This includes: emails, attachments, calendar appointments, tasks, contacts and folder structures.

In addition to the main goal, this plugin permits the user to adjust the overall functions to fit his environment offering large flexibility to select the target information to protect, to filter it because of privacy reasons or to do it efficiently through different strategies involving the parallelization of the implied operations.

General Features

Below, there is a list of general features this plugin offers:

- Backup and restore Exchange Server items
- Microsoft EWS API based backups
- Multi-service backup in the same job (email, calendar, contact and/or task)
- Multi-service parallelization capabilities
- Multi-thread single service processes
- Automatic parallelization of fetching processes
- Generation of user-friendly report for restore operations
- Network resiliency mechanisms
- Mailbox discovery capabilities
- List/query and auto-generation capabilities if combined with ScanPlugin

- Restore objects to Exchange Server
  - To original mailbox
  - To any other mailbox
  - To a different Exchange Server (cross-server restore)
- Restore any object to filesystem
- Full, Incremental & Differential backups
  - Advanced delta function for improved performance
- Mail folder, messages, appointments, contacts, tasks and attachments granularity for backup and restore
- Email addresses and folders selection capabilities for backup
- Backup and restore MIME objects for migration purposes
- Emails indexed at item level into Bacula Catalog
- Advanced search capabilities for restore operations
- Generation of user-friendly restore report into the destination mailbox
- Privacy filters for emails:
  - Ability to exclude email message fields from the catalog
  - Exclude private or spam messages through powerful filtering capability by rules.

Protected Items

Below, there is a list of all the items that can be backed up and restored using this plugin:

- User mailboxes
- Shared mailboxes
- Mailbox folder structure
- In-place archiving
- Emails and associated attachments
- Calendar appointments and associated attachments
- Contacts and associated attachments
- User tasks and associated attachments.

Architecture

Bacula Enterprise Exchange EWS Plugin is a Bacula File Daemon plugin built over the **Exchange EWS (Exchange Web Services) API** to perform all of its operations to retrieve from and feed to the target Exchange service. The plugin runs a Java Daemon which uses a custom extension of the EWS Managed API SDK originally built by Microsoft.

All the information is obtained using secure and encrypted HTTPS queries to Exchange Server from the File Daemon (and through the mentioned Java Daemon), where the plugin is installed. All the requests are performed over the following endpoint: https://exchange.hostname/EWS/EWS/Exchange.asmx

To get more information about EWS, visit: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/start-using-web-services-in-exchange

The metadata of every backed up item is stored in Bacula using JSON format. If MIME option is enabled, the information is also stored with that format (RFC 2077 for emails). Any attachment associated to a given item is downloaded and stored as it is. The download process is done locally to the host, and then sent to the Bacula Storage Daemon.

Backup and restore processes use different parallelization techniques in order to maximize performance, and overcome latency times when doing each needed request to EWS. Parallelization of several backup jobs is also supported.

Below, there is a simplified vision of the architecture of this plugin within a generic **Bacula Enterprise** deployment:

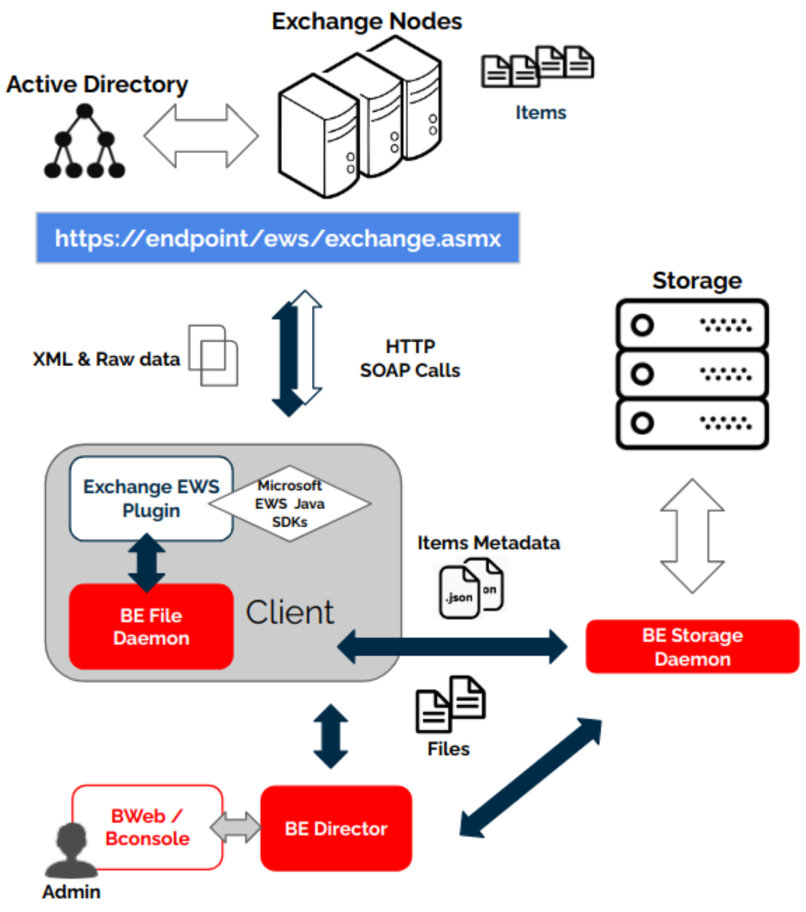


Fig. 36: Exchange EWS Plugin Architecture

Installation

This article describes how to install Bacula Enterprise Exchange EWS Plugin.

Prerequisites

- The Bacula File Daemon and the Exchange EWS Plugin need to be installed on the host that is going to connect to the Exchange Server.
- The plugin is implemented over a Java layer, and even if it backs up a Windows product, it must be deployed in a host running Linux. It is possible to use any of the supported **Linux** distributions of Bacula Enterprise, including RHEL, Debian, Ubuntu or Suse Linux Enterprise Server as some examples.
- The plugin works through a Java daemon, therefore Java needs to be installed into the host through a JRE or JDK package (openjdk-8-jre for example). Installed Java environment needs to be in version 8 or above and the Java binary must be available in the system PATH.
- Memory and computation requirements completely depend on the plugin configuration and usege (parallelization, size of data to backup, etc.). However, it is expected to have a minimum of **4GB RAM** in the server where the File Daemon is running. By default, every job could end using up to 512Mb of RAM in demanding scenarios (usually it will be much less). In some situations this could be higher. Memory limits can be adjusted (see Out of memory).
- Exchange EWS Service is used to perform all plugin operations. Therefore, it must be up and accessible through HTTPS from the host where Bacula FD and the plugin are going to be deployed. To do this, Outlook service needs to be installed on the host where the Bacula Enterprise Exchange Plugin is going to connect to fetch and protect the data.
- In order to fetch the data, the connection to EWS is done using Basic Authentication with username and password. An administrator user (a user belonging to the 'Organization Management' group) properly configured to access the mailboxes of any target user is needed. Details about how to configure such user are given in the next sections.
- EWS endpoint is usually served through HTTPS protocol. The certificate needs to be valid and the included CN (example: myhost.com) needs to match the endpoint configured in the plugin parameters.

Installation Methods

- EWSInstallationWithBIM (recommended)
- EWSInstallationPackageManagers
- EWSInstallationManual

Result

The package installs the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-exchange-ews-plugin-x.x.x.jar and bacula-exchange-ews-plugin-libs-x.x.x.jar). Note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.
- Plugin connection file (e2ws-fd.so) in the plugins directory (usually /opt/bacula/plugins). Note that e2ws acronym means Exchange EWS.
- Backend file (e2ws_backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-exchange-ews-plugin-x.x.x.jar file in order to launch it, even thought usually we should have only one file.

Once the plugin is installed, it should be possible to see it loaded through a status client command in bconsole ('Plugin:' line must contain 'e2ws'):

Listing 62: **Status client**

```
*st client
Automatically selected Client: 127.0.0.1-fd
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102

127.0.0.1-fd Version: 16.0.5 (05 April 2023)  x86_64-pc-linux-gnu ubuntu 22.04
Daemon started 14-abr-23 10:14. Jobs: run=2 running=0 max=100.
Ulimits: nofile=1024 memlock=2026356736 status=ok
Heap: heap=827,392 smbytes=436,939 max_bytes=5,100,087 bufs=153 max_bufs=248
Sizes: boffset_t=8 size_t=8 debug=600 trace=1 mode=1,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL 3.0.2 15 Mar 2022
APIs: !GPFS
Plugin: bpipe(2) e2ws(1.0.0)
```

Exchange EWS Plugin Installation with BIM

The recommended way to install any Bacula component, including any Daemon and any plugin is using the Bacula Installation Manager (BIM).

Steps

1. Install File Daemon.
2. Select the exchange-ews key in the plugin selection step.

Result

Exchange EWS Plugin is installed. Click here for more details.

For more information, visit Linux: Bacula Enterprise Installation with BIM.

Exchange EWS Plugin Installation with Package Managers

Another way to install this plugin is using the Package Manager of the used distribution.

Here, Debian Bullseye is taken as an example base operative system. The process will be very similar in any version of Debian, or any other Debian-based distribution (e.g. such as Ubuntu). In case of using RPM based distributions, like RHEL, Oracle Linux or Suse Linux, the main difference is to switch to the proper package manager of that distribution, usually *yum*.

Steps

1. Add the repository file suitable for the existing customer subscription, and the Debian version utilized. An example would be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 63: **APT repositories**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪bullseye-64/ bullseye main
deb https://www.baculasystems.com/dl/@customer-string@/debs/exchange_ews/
↪@version@/bullseye-64/ bullseye exchange_ews
```

2. Run of apt update:

Listing 64: **APT update**

```
apt update
```

3. Install the plugin using:

Listing 65: **APT install**

```
apt install bacula-enterprise-exchange-ews-plugin
```

The plugin has two different packages implied that should be installed automatically with the command shown:

- bacula-enterprise-exchange-ews-plugin
- bacula-enterprise-exchange-ews-plugin-libs

Result

Exchange EWS Plugin is installed. Click here for more details.

Exchange EWS Plugin Manual Installation

Alternately, manual installation of the packages may be done after downloading the packages, and then using the package manager to install them.

1. Download the packages from your Bacula Systems download area.

There are the packages you need to download for this plugin:

- bacula-enterprise-exchange-ews-plugin
- bacula-enterprise-exchange-ews-plugin-libs

2. After downloading, install them with the command:

Listing 66: **Manual dpkg install**

```
dpkg -i bacula-enterprise-*
```

Result

Exchange EWS Plugin is installed. Click here for more details.

Configuration

The following chapter presents the information on how to configure admin user, and fileset.

Admin User Configuration

Bacula Enterprise Exchange EWS Plugin needs an administrator user to access to the server and to retrieve the information to back up.

This admin user needs to be able to: - Impersonate other users with Full rights (and therefore access their mailboxes) - Have mailbox discovery abilities.

To configure impersonation, it is necessary to run the following command in Powershell:

Listing 67: **Impersonation**

```
New-ManagementRoleAssignment -name:impersonationAssignmentName -
↪Role:ApplicationImpersonation -User:<AdminUserName>
```

For more information about impersonation, visit: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/how-to-configure-impersonation

In order to provide full permissions to the admin user for the impersonated mailboxes, this command needs to be run in Powershell for every user:

Listing 68: **Mailbox permissions**

```
Add-MailboxPermission -Identity <user-to-impersonate@mydomain.com> -User
↪<admin-user@mydomain.com>   -AccessRights fullaccess
```

Mailbox discover capabilities are enabled with the following command in Powershell:

Listing 69: **Discovery Management**

```
Add-RoleGroupMember -Identity "Discovery Management" -Member <AdminUserName>
```

Note: Be aware that the effect of all these commands can take time. Especially the Mailbox discovery capabilities may need more than 30 minutes to be activated.

Important: In addition to running this command, the Admin user needs to have his mailbox activated. Otherwise, discovery capabilities won't work.

Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to invoke the Exchange EWS Plugin:

Listing 70: **Fileset E2WS**

```
Fileset {
   Name = FS_E2WS
   Include {
      Options {
        signature = MD5
        ...
      }
      Plugin = "e2ws: <e2ws-parameter-1>=<e2ws-value-1> <e2ws-parameter-2>=
↪<e2ws-value-2> ..."
   }
}
```

It is **strongly recommended** to use only one 'Plugin' line in every fileset. The plugin offers the needed flexibility to combine different modules backup inside the same plugin line. Different exchange servers, in case of existing, should be using different filesets and different jobs.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ','.

Below, in the subsections, there are lists that present all the parameters you can use to control Exchange EWS Plugin behavior.

Fileset Connection Parameters

The following parameters control the connection of the Exchange EWS Plugin to the Exchange Server.

Option	Require	Default	Values	Example	Description
endpoint	**Yes**		A hostname or IP address	win19-cl1-exch	Hostname or IP address that matches the DN of the SSL Certificate of the Exchange service
admin_	No		A domain name	MYEXCHAN DOMAIN	The users domain name. If admin_user is including already the domain, this parameter must not be set
admin_	**Yes**		Email address or username (with or without the domain prefix)	myadmin@r	An email address, or the username of the admin user that has permissions to impersonate all the other users. The format can be an email address, a single username (then admin_domain needs to be filled in) or domainusername. For simplicity, it is recommended to use the email address
admin_	**Yes**		A password string	G3934	The password associated to the admin user

Note: The admin user must have his mailbox enabled and working. Otherwise, discovery operations will fail.

Fileset Backup Parameters

The following list of parameters control what is going to be included into the associated backup:

Option	Require	Default	Values	Example	Description
service	No		email, contact, calendar, task, (list parameter: it can contain 0, 1 or more elements separated by ‘,’)	email	Establish the service or services that will be backed up. If this is not set, the plugin will try to backup all supported services.
user	No		Valid email addresses of existing users on the selected exchange service separated by ‘,’	AlexW@yo LeeY@you	Backup mailboxes associated to this list of users. If no user is provided the plugin will use discovery mechanism and include any user with an active mailbox
user_e	No		Valid email addresses of existing users on the selected exchange service separated by ‘,’	LauraG@yo AmandaT@yourd	Exclude selected mailboxes. If this is the only parameter found for selection, all elements will be included and this list will be excluded
user_r	No		Valid regex	.*@management\.mydo	Backup matching user mailboxes. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them
user_r	No		Valid regex	.*@guests\.	Exclude matching user mailboxes. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for user selection, all users will be included and matching users will be excluded
folder	No		Folder names separated by ‘,’	inbox,compan	Backup only the list of folders of this parameter from the mailboxes of the selected users. If no folder is provided, all folders will be included
folder_	No		Folder names separated by ‘,’	travel,perso	Exclude selected folders from the mailboxes of the selected users. If this is the only parameter found for folder selection, all folders will be included and this list will be excluded
folder_	No		Valid regex	.*mycompany	Backup matching folders by name. Please, only provide list parameters or regex ones. But do not try to combine them
folder_	No		Valid regex	.*private	Exclude matching user folders by name. Please, only provide list parameters or regex ones. But do not try to combine them. If this is the only parameter found for folder selection, all folders will be included and matching folders will be excluded
exclude_	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude any attachment from backup
mime	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Backup raw MIME file of items, in addition to the item objects themselves (regular objects are backed up as json formatted objects)
archiv	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include In-place archiving tree folders and items of the selected mailboxes

Fileset Common Parameters

These parameters are controlling generic aspects of the behavior of the Exchange EWS Plugin, it is possible to find also these parameters in other Bacula Enterprise Plugins with similar effects, so you may be familiar with them.

Option	Require	Default	Values	Example	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/e	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset
log	No	/opt/bacu debug.lo	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/e2ws.lo	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/e2ws/e2ws-debug.log* files containing debut information which is more complete with a greater debug number
path	No	/opt/bacu	An existing path with enough permissions for File Daemon to create any internal (and usually temporary) plugin file	/mnt/my-vol/	Uses this path to store metadata and temporary files

Fileset Tuning Parameters

These set of parameters can be used to fine-tune the behavior of the plugin to be more flexible in cases of bad network environments, or when significant job concurrency is happening, etc. It is not necessary to modify them for the great majority of the cases:

Option	Require	Default	Values	Example	Description
back	No	100	0-500	1	Number of maximum enqueued internal operations between service static internal threads (there are 3 communicating through queues with the set size: service fetcher, service opener and general publisher to bacula core). This could potentially affect api concurrent requests and consequently, throttling and cpu/memory consumption for both, the FileDaemon and the Exchange server. It is only needed to modify this parameter, in general, if you are need a ver precise control of your concurrency levels
concurrent_	No	10	0-100	1	Number of maximum concurrent backup threads running in parallel in order to open data for running download actions. If you want to have a precise control of your parallelization through different jobs, please set up this value to 1. Please be careful also with the memory requirements, multi-threaded increases very significantly memory consumption per job
concurrent_	No	2	0-20	1	Number of maximum concurrent backup page listing threads running in parallel in order to fetch sets of data. This parameter will also affect api concurrent requests
api_l	No	100	10-500	1	Number of items got in each page for multi-page requests to EWS API
general_	No	5	Positive integer (number of retries)	10	Number of retries for failed requests to the EWS API
general_	No	50	Positive integer (seconds)	100	Delay between retries to the EWS API

Fileset Advanced Parameters

The following parameters are advanced ones, and they should not be modified in the great majority of cases:

Option	Re-quire	Default	Values	Ex-am-ple	Description
stream_sl	No	1	Positive integer (1/10 sec-conds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_m	No	120	Positive integer (sec-onds)	360	Max wait time for FD to answer packet requests
time_max	No	86400	Positive integer (sec-onds)	4320	Maximum time to wait to ovewrite a debug log that was marked as being used by other process
log-ging_max	No	50MB	String size	300M	Maximum size of a single debug log fileGenerates the working/e2ws/e2ws-debug.log* files containing debut information which is more complete with a greater debug number
log-ging_max	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rollin	No	e2ws.log.% MMM}.lo	No, Yes	Yes	Log patter for rotated log files
split_confi	No	=	Charac-ter	:	Character to be used in config_file parameter as separator for keys and values
opener_qu	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object opener queue is full
pub-lisher_que	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The ".log" files are rotated automatically. Currently, each file can be 50Mb at maximum and the plugin will keep 25 files.
  - This behavior can be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The ".err" file can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like 'logrotate'.

- Backups in parallel and also failed backups will generate several log files. For example: e2ws-debug-0.log, e2ws-debug-1.log…

Fileset Examples

In this section, some fileset examples are presented:

Listing 71: **Fileset: for all data belonging to a user**

```
Fileset {
   Name = fs-e2ws-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: endpoint=myexchange.myorg.com admin_user=ex-admin@myorg.
↪com admin_password=xxxxxxx user=adelev@myorg.com"
   }
}
```

Listing 72: **Fileset: using a config file**

```
Fileset {
   Name = fs-e2ws-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings␣
↪user=adelev@myorg.com"
   }
}

Config file contents in stored in the same File Daemon host in /opt/bacula/
↪etc/e2ws.settings:
endpoint=myexchange.myorg.com
admin_user=ex-admin@myorg.com
admin_password=xxxxxxx
```

Listing 73: **Fileset: Backup only emails**

```
Fileset {
   Name = fs-e2ws-adelev-email
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email␣
↪user=adelev@myorg.com"
   }
}
```

Listing 74: **Fileset: Backup emails and appointments of all users**

```
Fileset {
   Name = fs-e2ws-email-calendar
   Include {
      Options { signature = MD5 }
```

(continues on next page)

(continued from previous page)

```
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email,
↪calendar"
   }
}
```

Listing 75: **Fileset: Backup only email folders: inbox and important custom folder**

```
Fileset {
   Name = fs-e2ws-all-inbox-important
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email␣
↪folder=inbox,important"
   }
}
```

Listing 76: **Fileset: Backup emails in mime format for two users**

```
Fileset {
   Name = fs-e2ws-mime-u1-u2
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email␣
↪mime=true user=user1@myorg.com,user2@myorg.com"
   }
}
```

Listing 77: **Fileset: Backup emails and contact in mime format for two users, but exclude attachments**

```
Fileset {
   Name = fs-e2ws-user1-user2-no-attach
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email,
↪contact mime=true exclude_attachments=true user=user1@myorg.com,user2@myorg.
↪com"
   }
}
```

Listing 78: **Fileset: Backup all services from all users starting with 'org'**

```
Fileset {
   Name = fs-e2ws-org-users
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings user_regex_
↪include=\"org.*\""
   }
```

(continues on next page)

(continued from previous page)

```
}
```

Listing 79: **Fileset: Backup one user reducing the concurrency configuration**

```
Fileset {
   Name = fs-e2ws-user1-min
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings␣
↪user=user1@myorg.com concurrent_threads=1 concurrent_listing_threads=1"
   }
}
```

Listing 80: **Fileset: Backup one user maximizing the concurrency configuration**

```
# Warning: This configuration could provoke throttling issues
Fileset {
   Name = fs-e2ws-user1-max
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings␣
↪user=user1@myorg.com concurrent_threads=50 concurrent_listing_threads=10␣
↪backup_queue_size=500 api_list_page_size=500"
   }
}
```

Listing 81: **Fileset: Backup all services, all users, but exclude emails where the subject contains 'private'**

```
Fileset {
   Name = fs-e2ws-exclude-private
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings email_exclude_
↪index_expr=\"emailSubject.includes('private')\""
   }
}
```

Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise Exchange EWS Plugin**.

Backup in Exchange EWS Plugin

Backup jobs in Exchange EWS plugin behave as any other backup job in Bacula Enterprise once the fileset has been created, as described in the configuration section. Below, some special features of the plugin that happen at backup time are described, as well as the file structure that a backup creates.

Backup File Structure

Items are formatted in the backup catalog in order to not include sensitive information. They are included in a path in the following format:

`/@e2ws/domain.name/users/user@domain.name/foldername/shortId_itemDate.itemExtension`

Depending on the type of the item, `itemExtension` wil be:

- Message: .msg
- Appointment: .pp
- Task: .task
- Contact: .con
- Contact group: .con.gr

Mime files will have the extra word 'mime' in their extension. For example:

`/@e2ws/testlab.local/users/ex-admin@testlab.local/regress_20230417125041/AAPfLdTrAAA=_r20230417-125325.mime.msg`

Attachments will be stored together with item objects:

- They include their original name (file name)
- They have a special extension ".att"
- They include the attachment type (file or item)
- The first part of the attachment name is the name of the parent message.

Here is an example of an attachment :

`/@e2ws/testlab.local/users/ex-admin@testlab.local/inbox/AAPfLdTuAAA=_r20230417-125329.msg.Prompta.gen.file.att`

`/@e2ws/testlab.local/users/ex-admin@testlab.local/inbox/AAPfLdTuAAA=_r20230417-125329.msg` → *Parent message*

MIME Objects Backup

Based on the fileset parameter **mime**, it is possible to get mime formatted items as well as regular objects (which are in json format). This kind of objects can be useful to get if there is any plan of using the information outside the Exchange service (e.g. for migration purposes).

Note: Activating this option has a performance penalty, and the backup time will be significantly higher, as for every email the information will be requested twice (one for the regular format, so .json message file plus attachments; another one for mime format, so to get a unified .mime.msg file containing the message and attachments).

At restore time, if the restore operation is done via EWS services and not to any local filesystem, selected mime objects are automatically ignored. It means if only those .mime.msg files were manually selected during a restore session over Exchange, the restore won't restore any file. While doing the same over a Local filesystem, destination will end up with those .mime.msg files restored. Usually, the selection would include both kind of files (a folder, a whole backup). In that situation, the restore will be simply successful, while those .mime.msg won't be used.

Email Privacy Filters

Bacula Systems is aware of one of many privacy concerns that may arise when tools like the Exchange EWS Plugin enables the possibility to backup and restore data coming from different users, so the backup administrator can restore potentially private data at his will. Moreover, emails are usually one of the most critical items in terms of privacy.

One of many strategies the plugin offers in order to deal with that problem is the possibility to exclude messages. This is a very powerful feature where it is possible to use quite flexible expressions that allow to select a subset of messages and simply exclude them from the backup:

- `email_exclude_expr` fileset parameter will exclude completely the selected messages
- `email_exclude_index_expr` fileset parameter will exclude the selected messages from the index (MetaEmail catalog table).

Not only messages can be excluded, but also select only a subset of email fields to be included in the indexed information using `email_fields_exclude_index` fileset parameter.

All three discussed expressions are based on an internal structure of fields to work with. Below, you can see the entire list of fields that you can use:

- emailTags
- emailSubject
- emailFolderName
- emailFrom
- emailTo
- emailCc
- emailBodyPreview
- emailImportance
- emailTime
- emailIsRead

- emailIsDraft

Note: It is very important to write the fields exactly as written above.

These fields can be used in a comma separated list in the `email_fields_exclude_index` parameter.

Then, for `email_exclude_index_expr` and `email_exclude_expr`, use them in a valid boolean expression in **Javascript** language syntax. Some examples are provided below:

Listing 82: **Expression to exclude messages where subject includes the word 'private'**

```
emailSubject.includes('private')
```

Listing 83: **Complex expression to exclude messages that are not read and are Draft or their folder name is named Private**

```
!emailIsRead && (emailIsDraft || emailFolderName == 'Private')
```

Listing 84: **Expression to exclude messages based on the received or sent date**

```
!emailTime < Date.parse('2012-11-01')
```

Listing 85: **Expression to exclude messages using a regex based on emailFrom**

```
/.*private.com/.test(emailFrom)
```

Note: This feature is available since Bacula Enterprise version 14.0

Expression Tester

This expression mechanism can sometimes be uncertain for end users as they can have doubts about the correct behavior of their prepared expressions. In order to help with that, Exchange EWS Plugin presents a query method that allows to test those expressions against a static preloaded set of data.

There are two commands available:

- Show command
- Test command

The show command will show the static data in json format, so it is possible to see the contents to adapt the expressions to test command - it will apply the expression parameters to the preloaded static data.

The test command has the following format:

Listing 86: **Expression tester Show command**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_
↪user=<username> admin_password=<password>" parameter=email-expr-show
```

The show command has the following format:

Listing 87: **Expression tester Test command**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_
↪user=<username> admin_password=<password> email_exclude_expr = \"<your-js-
↪expression>\"" parameter=json|email-expr-test
// Or
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_
↪user=<username> admin_password=<password> email_exclude_index_expr = \"
↪<your-js-expression>\"" parameter=json|email-expr-test
```

Note: You need to provide a valid endpoint and user credentials, even if it's not really used to process any data.

The test command produces JSON output with objects with the exact format that is received from Microsoft and, consequently, the same format that is stored in backup. Note that 'total' value at the end, where the value of 12 total preloaded messages is shown.

Listing 88: **Expression tester Show command output**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_
↪user=<username> admin_password=<password>" parameter=json|email-expr-show
....
    "email-12": {
      "body": {
        "content": "These are the contents in text format of the 12 email of␣
↪test data. It has the following categories:orange, black, white, purple.␣
↪You can try to filter this body using any JS method like /.*12.*/.
↪test(emailBody) or emailBody.includes(12)",
        "contentType": "TEXT"
      },
      "ccRecipients": [
        {
          "emailAddress": {
            "address": "danny@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "lucas@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "terese@other.com"
```

(continues on next page)

(continued from previous page)

```
          }
        }
      ],
      "from": {
        "emailAddress": {
          "address": "elon@other.com"
        }
      },
      "hasAttachments": false,
      "isDraft": false,
      "isRead": false,
      "replyTo": [
        {
          "emailAddress": {
            "address": "elon@other.com"
          }
        }
      ],
      "sentDateTime": {
        "dateTime": {
          "date": {
            "year": 2021,
            "month": 12,
            "day": 5
          },
          "time": {
            "hour": 11,
            "minute": 30,
            "second": 0,
            "nano": 0
          }
        },
        "offset": {
          "totalSeconds": 0
        }
      },
      "subject": "This is private subject 12",
      "toRecipients": [
        {
          "emailAddress": {
            "address": "laura@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "jack@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "john@other.com"
          }
```

(continues on next page)

(continued from previous page)

```
        }
      ],
      "categories": [
        "orange",
        "black",
        "white",
        "purple"
      ]
    }
  },
  {
    "total": "12"
  }
```

The test command on its side will produce two different outputs. The first part presents the same format as the show format, and those are the messages that would be included in the backup. The second part presents a different format, so an output like:

Listing 89: **Expression tester Test command, index part output**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_
↪user=<username> admin_password=<password>" parameter=json|email-expr-show
....
      {
    "meta-email-12": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "ews.test",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is private subject 12",
      "EmailFolderName": "/",
      "EmailFrom": "elon@other.com",
      "EmailTo": "laura@other.com,jack@other.com,john@other.com",
      "EmailCc": "danny@other.com,lucas@other.com,terese@other.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 235,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "e2ws"
    }
  },
  {
    "total-backup": "12"
  },
  {
    "total-index": "12"
```

(continues on next page)

(continued from previous page)

```
  }
```

That part represents the information that would be indexed in the backup (included into the Catalog). You can also see the total entries at the end, which are very useful to quickly compare with the original 12 value and so, knowing if our expression is filtering the expected data or not. Below, we provide an example where some filtering to the backup is applied, but also it is applied to the index:

Listing 90: **Expression tester Test command, index part output**

```
.query client=127.0.0.1-fd plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password> email_exclude_expr=\"emailFrom ==
↪'elon@other.com'\" email_exclude_index_expr=\"emailSubject.includes('private
↪')\"" parameter=json|email-expr-test
...
     {
    "meta-email-4": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "ews.test",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is orange subject 8",
      "EmailFolderName": "/",
      "EmailFrom": "bob@company.com",
      "EmailTo": "laura@company.com,jack@company.com,john@company.com",
      "EmailCc": "danny@company.com,lucas@company.com,terese@company.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 232,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "e2ws"
    }
  },
  {
    "total-backup": "6"
  },
  {
    "total-index": "4"
  }
```

In case your expression is not valid, the plugin will also inform about that with the following message:

``error=Error listing elements. Cause: Predicate test error!! Review your query .....

Delta Backup

The Microsoft EWS API provides a Delta function to track changes of some objects. Bacula Enterprise Exchange EWS Plugin uses this function in order to speed up Incremental/Differential processes.

Delta function has the following important characteristics:

- Delta tokens can expire at some point, or even become invalid due to internal Microsoft issues. If that happens, the plugin will try to start a new Delta cycle.
- There are two delta types of tokens implied: one for the folder structure, another for every folder that has changes inside.
- Any situation where the Delta function cannot be used will trigger a regular Full/Inc/Diff, where every element is listed and selected or discarded according to the item dates.

The Delta backup cycle is described below:

- Full backup: All entity elements are backed up. A token (token_1) is generated and the token is stored locally by the FD.
- Incremental 1 backup: token_1 is used to retrieve changes since token_1's generation, so every change is backed up. A new token is generated and stored locally by the FD.
- Incremental 2 backup: token_2 is used to retrieve changes since token_2's generation, so every change is backed up. A new token is generated and stored locally by the FD.
- And so on…

Tokens are stored in a file placed in a path defined by the **path** parameter of the plugin. The name is: `jobname.deltaLink`.

The file stores tokens required for every execution, and it is renewed (emptied) during every Full backup execution.

This file is also backed up in the backup itself, so it can be restored manually, before an Incremental/Differential execution in case it was lost and in case you don't want to run a Full backup again.

Here, we can see an example of the contents of the file, with one execution and one user entity involved. The structure is tree-based, so it is easy to understand what would be generated in case of backing up other folders or users:

Listing 91: **deltaLink**

```
{
   "deltaServices" : {
      "e2ws" : {
         "entities" : {
            "ex-admin@testlab.local" : {
               "containers" : {

↪"AAMkADkwMWYyMWQwLWZjZmMtNDU3NS1iMmM3LWVmMTRkNTQ0MjVjYQAuAAAAAAD2ghxkrOnXTJN4mEgvv12nAQAtUQmYk+IgR
↪AAPfLblCAAA=" : {
                     "deltaEntries" : [
                        {
                           "date" : "Apr 14, 2023, 10:14:42 AM",
                           "delta" :
↪"H4sIAAAAAAAAEAGNgYGcAAotqE0tHE2NTA0ddZ3NHC10TR2djXSdnJ2ddNyMnC2cnczdLU1OD2vBgveDKvOTgksSSVOfEvMSiS
↪j8GaaK3+QMuKS4JSk1Mzy1JTQjJzU0nwrU9icYlnXnFJYl5yqncqKb71zS9K9SxJzS32zwtOLSpLLSLByXDfhgNxUW5iUTYklr
```

(continues on next page)

(continued from previous page)

```
↪sckYG3UDOGZMfKbjuPaV2Yc5p/
↪f0MzPd1dzoxMDAy8DEwg7RwM9jViBcesLntwSAEFOUFYqB1rIwMDL6OAZ6+jn4gRQxupm5hYOVooB2I5ZD4S9H4MHAT
↪Xj0sKYLgXwr5brZ7Dd7L+v6q5Sj/
↪2soE1YfomCCoO0paMRRbo2b1aYNIYr94knDYyMTA0AAAccVKZJwMAAA==",
                          "job" : "pluginTest.2023-04-14_10.14.42_03"
                       }
                    ],
                    "description" : "regress_20230414101303",
                    "id" :
↪"AAMkADkwMWYyMWQwLWZjZmMtNDU3NS1iMmM3LWVmMTRkNTQ0MjVjYQAuAAAAAAD2ghxkrOnXTJN4mEgvv12nAQAtU
↪AAPfLblCAAA="
                 },
                 "MailboxFolders" : {
                    "deltaEntries" : [
                       {
                          "date" : "Apr 14, 2023, 10:14:42 AM",
                          "delta" :
↪"H4sIAAAAAAAEAK2be1ST5x3HX6HeQJCgwqzIXaVqJEhCiEBpLsSkBpRytG6oECBcLAaXAkqFeQFXT6TUiU5AgWCpt
↪h83t/veZ/
↪3uSMIowX7Fb1BqlBKI2USpVgtV0aLpUp1pFilVqnF2vmqaLVKrlXIZJKyV1PmpZSYM1MKjYUmtdFstJQIC/
↪hJbUF+lsmizxIU/
↪Owyk+X1vAKzIFHn55nMhU9+lqaWSzQRURKpWKJIkIul0YpssSoiWiXWqCIj5SqFMlIr0QpR6iJLislSbLIkGs152abl
↪8/7kn5uAmCjz2MHyLX5ZksRktmbokjKkFk/7XE/pnn+Lt/
↪bfHLOnSzz1C9vkYX3rny4ChBnDy+pvpqYMInF2Z277sY3un4q1Hj7B/B0/6xX/bvQqXjC5fFtU/
↪fM0UQXPgpH4jyhSg/
↪QXDlp6ZDlD9EBUBUIEQFQVQwRIVAVChEzYComRA1C6LCIOoFiJoNUXO43y9HO+IBUQboXZ4I3cuTuzQcFH+NclD8NU
↪rR1CE/
↪DXKEeEUiJoMRYi1AJMgij9CBzUVop6HqGlQGYY4enduyo2bcu0aMgVDFBbhRIji780dVCx0rwkQ5SUIY/
↪gpoOTtNWoVRKVDlBGisiEqByp5+9h5ND/
↪lDtX5IIgKhagZEDUTomZBVBhEvQBRsyFqDkTN5ad6XWzfQ9QwQjUJwPulHVoEUQaISoRaAC+IEkGUN0QBdcNOiSFqH
↪gKidEFUNUU0QtR+iDkMl38p/
↪ryvij1VQhBkQlQlRWRCVC1F5ELUaoswQVQBRayGqEKKKIKoYouohqgGiGqGVivEQ5Q2Nvjz5KfvMF5in2GejoyDKBS
↪tGXg+IfETko/jGbg3oyZuPfjBWebun62L/9j0e6KFpUWj18nr8mTGFKfaR91wr4pT5M6TRRW/
↪sW4vQN6UP/HGjij9SPKTW+W77nDrU0Q9T1HWeZPjP9VVnLVxdkUEtNB1ed30YtzTY/
↪fNTMX6ZBTGnOnbo/fUYtzT3R/
↪fa31OnnfSwXfsIfaShTurrs7cV66khfC72oXE8szbCpA3qXEEvXhLSM+VUctTT04dW014ilFkum3Pcjaun6B2GH91NI
↪0e33uOXerCkXjebZz88RCm1pz+r8drPjzbQlunLj2/8UXf/
↪U36pJyt991c81j43SC1dGXNkFNCbMqSBbbrTva31LZRS+4OKdb+aqwDeKHb61pzSTi11+rve9MrfTS2t6GxP+j11+tc
↪STqWsl9uf28P23ycWDp5IOXS1+uo36iQ0bb3DNRl2hDTOW4mdeX3LdnWkkabvuHH1/
↪9Q7tiAJJUaH4tNtcP8DUo6S5oh3Drp0UgcqXlX3e7fcg7QRg70OJXWJsR4Ax0fu0o1+ja8JaOW1lpT2/
↪xJy7Tr3o6yfxQZf0cstf307spbocTS5rr3C9bc5W+kZzCljeaCSs41lGdG2rJp++V1JmJp6/
↪tuwYGf86cfxpS2Hq8+Qjvh7bp3rPxbzUvU9fTEId+BA5yD3pFDVE6l7eM8vJeRRtrreqMs/
↪Xq+B7F08EBs/
↪tI3udepfzih5VT6wVd7z5A2fa59OlncwqkXgFk0UxqXlDMhj3Oxa+T4l1NpyqDKnTjSRXOaRB/
↪UWYml4Rdqd1sX83d8Iqa069wS8zjiSEs11tZdHxJLN61ZXnX5KH/
↪6YqbU0lJxu5840h1bj+0fWEos3dXfIJsPrPYwpbWRYS6/OUMsrU/LmnvsKrG0adtR246z/E8/
↪kil9643Hnb7EkXZ4nQqqCqGWin79TSD1flSHT9JSaSqx9JR+xW1JPLH0nObmh7HbiaWffuL97rIOYuml8tXNw7OJpV
↪5Y9nSttl8X+mrlLfNOVPfMS5w/
↪ssqcG18sSXa4lHKIaxe1LE5YX8ZZrAlO4bqlLTTiT6DO5HldP1xBsyhoknNwSuAA5l6JjSs5+P7RURR+p7Xap5MJk/
↪0kVM6WBaevzPiCMNzUi//eox/kiXMKU5m385uZM40uQTy9qLaac8fYaW/
↪G75ke+IpSfbB2fsBHbNK1jSM4cTo+4Q91GGPjdVcm0Ff6RWlnRgzJWK0gXEkd4V2S5t1BFL7/
↪tZ4zZxbnCPnHp1Kg2apu6roo00MahM69IDPKgmpnSLvi0A2ONjRpp4auHKawf4Iz3MkiZ3HDzdSjvmvxK+avHy4neA
```

(continues on next page)

(continued from previous page)

```
↪kjzWVK/zbUhKSfx5TeaOsXUQ/QNq79S8YmzlZq5CCyU2nRQDCSfgFT+obfcNlm4vQrx5+v7K/
↪nj7SQKfWs8ZAA6RcxpZNSQ8+WEjd9U+c2zM2O5Y+0nimNGKs9+G9+aQNTGhWwfKeNepdnT+9p/
↪xJqqe2vzQHAv/h7s2Z8ITv9ai5uIF7nP9WifBBFu23c63rn/
↪rSvPuPdOnp6Ntyp9NFH1bIVxA9qwt7c76uB2YmGKe144PUQ2I5jSw+UJ00CljvY0nZznf9WYbzwX9d/
↪AO8yNqGhTQAA",
                          "job" : "pluginTest.2023-04-14_10.14.42_03"
                       }
                    ],
                    "description" : "mailboxfolders",
                    "id" : "MailboxFolders"
                 }
              },
              "id" : "ex-admin@testlab.local",
              "name" : "ex-admin@testlab.local"
           }
        }
     }
  },
  "jobName" : "pluginTest"
}
```

Restore in Exchange EWS Plugin

Exchange EWS plugin is able to restore to any local filesystem mounted over the host where the File Daemon is running, or to the Exchange environment. The restore method is selected based on the value of the `where` parameter at restore time:

- Empty or '/' (example: `where=/`) → Exchange EWS restore will be triggered
- Any other path for where (example: `where=/tmp`) → Local file system restore will be triggered.

When using the Exchange EWS restore method, the following parameters are available to control the restore behavior under 'Plugin Options' menu during a bconsole restore session:

Option	Require	Default	Values	Example	Description
destination_u	No		Existing email address on the target Exchange service	AlexV	Destination User where restore data will be uploaded. If no user is set, every selected file will be restored in the original account
destination_a	No		0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	yes	Restore using the in-place archving tree instead the regular mailbox tree
send_	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Send an email to the destination user with a report containing the result of all restored (or failed) items
foreign_c	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Generate a general folder to put inside restored items coming from different mailboxes. For example, if we restore emails from user a@domain.com into Mailbox of user b@domain.com, with this option enabled the plugin will generate an automatic folder a@domain.com inside the destination restore folder used over destination user b@domain.com
endpoint	No	Original backu value	A hostname or IP address	win19 cl1-exch	Cross-server restore: Hostname or IP address that matches the DN of the SSL Certificate of the Destination Exchange service
admin_d	No	Original backu value	A domain name	MYE CHAN DOMAIN	Cross-server restore: The users domain name. If admin_user is including already the domain, this parameter must not be set
admin_u	No	Original backu value	Email address or username (with or without the domain prefix)	myadmin@	Cross-server restore: An email address, or the username of the admin user that has permissions to impersonate all the other users. The format can be an email address, a single username (then admin_domain needs to be filled in) or domainusername. For simplicity, it is recommended to use the email address
admin_p	No	Original backu value	A password string	G393	Cross-server restore: The password associated to the admin user
debug	No	0	0, 1, 2 ,3, 4, 5, 6, 7, 8, 9	3	Change debug level

Restore Use Cases

The following restore scenarios are supported:

- Restore whole directories, specific items (email, task, contact, appointments or attachments) to original user or to a different user:
  - Restore parameters implied: `destination_user`.
- Restore directories, emails, or attachments to original path or to a different path:
  - Restore parameters implied: `destination_path`.
- Restore to the local filesystem (general restore `where` parameter must be set to a path).
- It is possible to control replacement behavior (items are compared using exchange id) with the generic replace option of Bacula.
- Restore to a different exchange server:
  - Restore parameters implied: `endpoint`, `admin_domain`, `admin_user`, `admin_password`.

Some particularities to remark:

- If no `destination_user` is set, every message will be restored into its original mailbox.
- If no `destination_path` is set, every message will be restored into its original path.
- If the selection contains messages from several users:
  - Original user messages will be restored in their original location
  - For other users, a special folder will be created with the email address of each of them, containing the full path and messages of the restored objects, unless the parameter `foreign_container_generation` is disabled
  - Example: Restore of emails from 2 different users over a third mailbox without destination_path result in auto-generated Restore_date folder containing those 2 foreign users with the restored folder inside of them.

Restore Example Session

In the following restore example session, we restore into the original mailbox all the emails of the backup, inside the 'restored' folder.

Note: It is also possible to run backup or restore operations from any of the Bacula Graphical User Interfaces.

Listing 92: **Restore bconsole session**

```
Connecting to Director 127.0.0.1:8101
1000 OK: 10002 127.0.0.1-dir Version: 16.0.5 (05 April 2023)
Enter a period to cancel a command.
*restore
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"

First you select one or more JobIds that contain files
```

(continues on next page)

(continued from previous page)

```
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
   1: List last 20 Jobs run
   2: List Jobs where a given File is saved
   3: Enter list of comma separated JobIds to select
   4: Enter SQL list command
   5: Select the most recent backup for a client
   6: Select backup for a client before a specified time
   7: Enter a list of files to restore
   8: Enter a list of files to restore before a specified time
   9: Find the JobIds of the most recent backup for a client
   10: Find the JobIds for a backup for a client before a specified time
   11: Enter a list of directories to restore for found JobIds
   12: Select full restore to a specified Job date
   13: Select object to restore
   14: Cancel
Select item:  (1-14): 5
Automatically selected Client: 127.0.0.1-fd
Automatically selected Fileset: FS_E2WS
+-------+-------+----------+------------+---------------------+---------------
↪----+
| jobid | level | jobfiles | jobbytes   | starttime           | volumename    ␣
↪    |
+-------+-------+----------+------------+---------------------+---------------
↪----+
|     1 | F     |       32 | 10,632,924 | 2023-04-17 12:52:41 | TEST-2023-04-
↪17:0 |
+-------+-------+----------+------------+---------------------+---------------
↪----+
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
31 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ mark *
31 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================
```

(continues on next page)

(continued from previous page)

```
   TEST-2023-04-17:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


31 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /tmp/regress/tmp/bacula-restores
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: Fileset
   5: Restore Client
   6: When
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 9
Please enter the full path prefix for restore (/ for none): /
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
```

(continues on next page)

(continued from previous page)

```
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: Fileset
   5: Restore Client
   6: When
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : e2ws: service=email endpoint=w16-cl02-exch admin_
↪user=ex-admin@testlab.local admin_password=Bacula18 debug=4 user="ex-
↪admin@testlab.local" folder="REGRESS_20230417125041"
Plugin Restore Options
Option                         Current Value        Default Value
destination_user:              *None*               (*None*)
destination_path:              *None*               (*None*)
send_report:                   *None*               (1)
foreign_container_generation:  *None*               (1)
send_invitations_mode:         *None*               (AllCopy)
endpoint:                      *None*               (*None*)
admin_domain:                  *None*               (*None*)
admin_user:                    *None*               (*None*)
admin_password:                *None*               (*None*)
debug:                         *None*               (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: destination_user (Destination User)
   2: destination_path (Destination Path in Exchange)
   3: send_report (Send report of the restore operation to the affected user)
   4: foreign_container_generation (Generate a general container (usually a␣
↪folder) to put inside restored objects coming from different entities)
   5: endpoint (Destination Exchange endpoint)
   6: admin_domain (Destination Exchange endpoint admin user domain)
   7: admin_user (Destination Exchange endpoint admin user)
   8: admin_password (Destination Exchange endpoint admin password)
   9: debug (Change debug level)
Select parameter to modify (1-10): 2
Please enter a value for destination_path: restore
Plugin Restore Options
Option                         Current Value        Default Value
destination_user:              *None*               (*None*)
destination_path:              restore              (*None*)
send_report:                   *None*               (1)
foreign_container_generation:  *None*               (1)
endpoint:                      *None*               (*None*)
admin_domain:                  *None*               (*None*)
```

(continues on next page)

(continued from previous page)

```
admin_user:                    *None*              (*None*)
admin_password:                *None*              (*None*)
debug:                         *None*              (*None*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
Job queued. JobId=3
```

Listing 93: **Restore job result**

```
*llist joblog jobid=3
time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Start Restore Job RestoreFiles.2023-04-17_13.
↪16.54_11

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Restoring files from JobId(s) 1

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Connected to Storage "File" at 127.0.0.1:8103␣
↪with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Using Device "FileStorage" to read.

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Connected to Client "127.0.0.1-fd" at 127.0.0.
↪1:8102 with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: Connected to Storage at 127.0.0.1:8103 with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-sd JobId 3: Ready to read from volume "TEST-2023-04-17:0"␣
↪on File device "FileStorage" (/tmp/regress/tmp).

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-sd JobId 3: Forward spacing Volume "TEST-2023-04-17:0" to␣
↪addr=260
```

(continues on next page)

(continued from previous page)

```
time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Plugin log of this job available in: /
↪tmp/regress/working/e2ws/e2ws-debug-0.log

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Backend connection to testlab.local␣
↪stablished

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Jar Version: 1.0.0 | Java version: 11.0.
↪18

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Starting backend restore process

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Restore to Microsoft Exchange Service

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: No destination entity provided. Trying␣
↪to restore each item into its original owner entity

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Destination Path: restore

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Generate report: enabled

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Foreign container generation: enabled

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Send invitations mode for appointments:␣
↪AllCopy

time: 2023-04-17 13:17:03
logtext: 127.0.0.1-sd JobId 3: End of Volume "TEST-2023-04-17:0" at␣
↪addr=10694381 on device "FileStorage" (/tmp/regress/tmp).

time: 2023-04-17 13:17:03
logtext: 127.0.0.1-sd JobId 3: Elapsed time=00:00:06, Transfer rate=1.780 M␣
↪Bytes/second

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-fd JobId 3: e2ws: Report sent to:ex-admin@testlab.local

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-fd JobId 3: e2ws: No more items to restore. Restore ended

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Bacula 127.0.0.1-dir 16.0.5 (05Apr23):
```

(continues on next page)

(continued from previous page)

```
   Build OS:               x86_64-pc-linux-gnu ubuntu 22.04
   JobId:                  3
   Job:                    RestoreFiles.2023-04-17_13.16.54_11
   Restore Client:         "127.0.0.1-fd" 16.0.5 (05Apr23) x86_64-pc-linux-
↪gnu,ubuntu,22.04
   Where:
   Replace:                Always
   Start time:             17-abr-2023 13:16:57
   End time:               17-abr-2023 13:17:19
   Elapsed time:           22 secs
   Files Expected:         31
   Files Restored:         31
   Bytes Restored:         10,669,371 (10.66 MB)
   Rate:                   485.0 KB/s
   FD Errors:              0
   FD termination status:  OK
   SD termination status:  OK
   Termination:            Restore OK

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Begin pruning Jobs older than 6 months .

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: No Jobs found to prune.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Begin pruning Files.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: No Files found to prune.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: End auto prune.

            jobid: 3
              job: RestoreFiles.2023-04-17_13.16.54_11
             name: RestoreFiles
      purgedfiles: 0
             type: R
            level: F
         clientid: 1
       clientname: 127.0.0.1-fd
        jobstatus: T
    jobstatuslong: Completed successfully
        schedtime: 2023-04-17 13:16:33
        starttime: 2023-04-17 13:16:57
          endtime: 2023-04-17 13:17:19
      realendtime: 2023-04-17 13:17:19
    realstarttime: 2023-04-17 13:16:57
         jobtdate: 1,681,730,239
     volsessionid: 3
   volsessiontime: 1,681,728,758
```

(continues on next page)

(continued from previous page)

```
        jobfiles: 31
        jobbytes: 10,669,371
       readbytes: 10,629,087
       joberrors: 0
 jobmissingfiles: 0
          poolid: 0
        poolname:
      priorjobid: 0
        priorjob:
       filesetid: 0
         fileset:
        hascache: 0
         comment:
        reviewed: 0
   isvirtualfull: 0
            rate: 485
   compressratio: 0
      statusinfo:
    writestorage:
     writedevice:
 lastreadstorage: File
  lastreaddevice: FileStorage
```

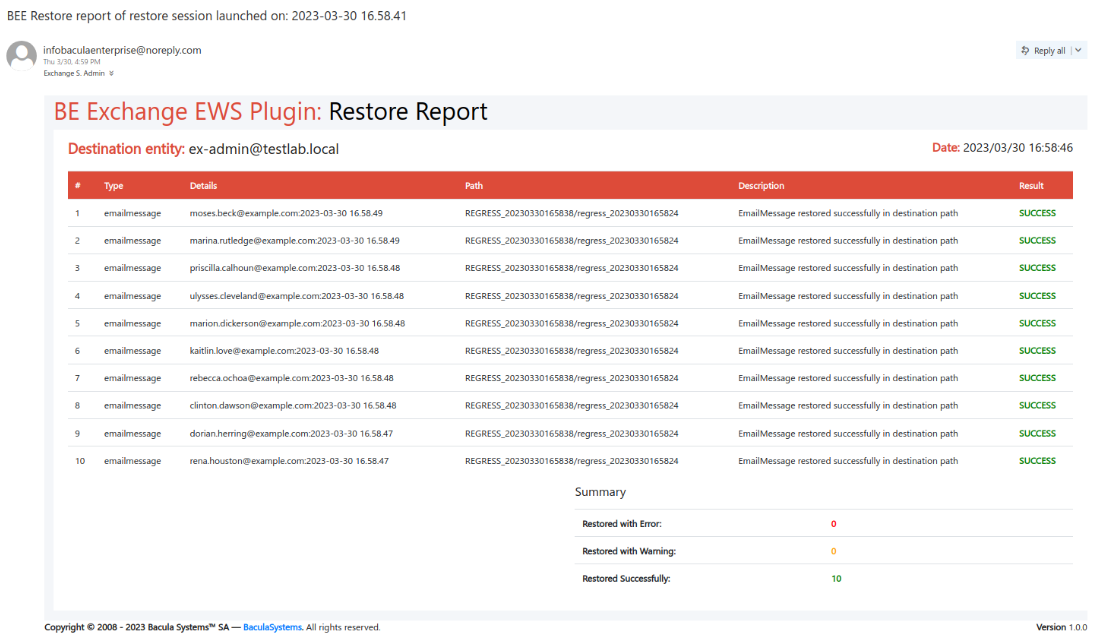
User Restore Report

Files and emails can represent very sensitive information for end-users. For that reason, information included in backup/restore logs is not exhaustive by default. For example, email restores do not include information such as the subject or sender when they are displayed in the backup log. However, for reporting and controlling purposes, the information of what has been exactly restored, what permissions have been applied, and other information can be useful and necessary for the affected user.

Bacula Enterprise Exchange EWS Plugin includes an option to generate a restore report in the user mailbox destination. The restore report contains detailed information about the items that have been restored successfully, if any of them had any trouble during the restore, and it also reports the date when the action was performed.

The generation of the report can be enabled/disabled in the bconsole restore session. If enabled, depending on the service, the report can generate an HTML file or an email in the Inbox of the affected user.

The image below shows an example report from an Email restore session:



BEE Restore report of restore session launched on: 2023-03-30 16.58.41

infobaculaenterprise@noreply.com
Thu 3/30, 4:59 PM
Exchange S. Admin

BE Exchange EWS Plugin: Restore Report

Destination entity: ex-admin@testlab.local

Date: 2023/03/30 16:58:46

#	Type	Details	Path	Description	Result
1	emailmessage	moses.beck@example.com:2023-03-30 16.58.49	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
2	emailmessage	marina.rutledge@example.com:2023-03-30 16.58.49	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
3	emailmessage	priscilla.calhoun@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
4	emailmessage	ulysses.cleveland@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
5	emailmessage	marion.dickerson@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
6	emailmessage	kaitlin.love@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
7	emailmessage	rebecca.ochoa@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
8	emailmessage	clinton.dawson@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
9	emailmessage	dorian.herring@example.com:2023-03-30 16.58.47	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
10	emailmessage	rena.houston@example.com:2023-03-30 16.58.47	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS

Summary

Restored with Error:	0
Restored with Warning:	0
Restored Successfully:	10

Copyright © 2008 - 2023 Bacula Systems™ SA — BaculaSystems. All rights reserved.

Version 1.0.0

Fig. 37: Restore Email Example Report

List & Query

It is possible to list information using the bconsole `.ls` command and providing a path. In general, we need to provide a path representing a folder inside the user mailbox. In addition, it is also possible to list the users from a given exchange endpoint through a `.query` command.

Below, there are some examples:

List users:

Listing 94: **Query example: Users**

```
*.query client=127.0.0.1-fd plugin="e2ws: endpoint=xxxx admin_user=xxxxx@my.
↪domain admin_password=xxxxx" parameter=user
user=ex-admin@testlab.local
displayName=Exchange S. Admin
guid=82190796-3221-4f26-8efb-c52ffdc2c4d2
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=df20c005a2b64a3a82d431e97f12ce78-Exchange S
user=support@testlab.local
displayName=First S. User
guid=48b2b400-44ab-40e5-995c-fede3b4453c3
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=8efecafa2b724e85990b1047fca7cba3-First S. U
user=beuser@testlab.local
displayName=Backup Exec
guid=ca497f2d-5b03-4959-8dc8-78f475905a10
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=3f1b5ef6f9ef4a00bd5ea92206d7675c-Backup Exe
user=support2@testlab.local
```

(continues on next page)

(continued from previous page)

```
displayName=Second D. User
guid=527e07d5-a961-4d4b-bc7a-ec00ebc09c32
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=ccc701ae568643329c87dbcaa8f0acfd-Second D.
user=CompanyMeeting@testlab.local
displayName=Bsys room
guid=8b357da2-1249-4fbb-ad27-070272ef7524
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=93b1df2aff8a4fadbf15d58b73c69999-Bsys room
```

List inbox emails:

Listing 95: **List example: Inbox emails**

```
*.ls client=127.0.0.1-fd plugin="e2ws: endpoint=xxxxxxxx admin_
↪user=xxxx@mydomain.com admin_password=xxxx user=ex-admin@testlab.local␣
↪service=email" path=Inbox
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
drwxr-xr-x   1 nobody   nogroup                  -1 1970-01-01 00:59:59  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/
-rw-r-----   1 nobody   nogroup              386774 2023-04-14 10:16:15  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPfLckyAAA=_
↪r20230414-101615.msg
-rw-r-----   1 nobody   nogroup              381535 2023-03-30 17:04:26  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqFAAA=_
↪r20230330-170426.msg
-rw-r-----   1 nobody   nogroup              381535 2023-03-30 17:04:09  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqEAAA=_
↪r20230330-170409.msg
-rw-r-----   1 nobody   nogroup              385825 2023-03-30 17:03:22  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqDAAA=_
↪r20230330-170322.msg
-rw-r-----   1 nobody   nogroup              385598 2023-03-30 17:02:10  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqCAAA=_
↪r20230330-170210.msg
-rw-r-----   1 nobody   nogroup              386081 2023-03-30 17:00:59  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqBAAA=_
↪r20230330-170059.msg
-rw-r-----   1 nobody   nogroup              382465 2023-03-30 16:59:48  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgqAAAA=_
↪r20230330-165948.msg
-rw-r-----   1 nobody   nogroup              381644 2023-03-30 16:59:05  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgp%2FAAA=_
↪r20230330-165905.msg
-rw-r-----   1 nobody   nogroup              384487 2023-03-30 16:58:09  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgp+AAA=_
↪r20230330-165809.msg
-rw-r-----   1 nobody   nogroup              381568 2023-03-30 16:57:44  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgp9AAA=_
↪r20230330-165744.msg
-rw-r-----   1 nobody   nogroup              381978 2023-03-30 16:57:18  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgp8AAA=_
```

(continues on next page)

(continued from previous page)

```
↪r20230330-165718.msg
-rw-r-----   1 nobody   nogroup            381402 2023-03-30 16:55:51  /
↪@e2ws/testlab.local/users/ex-admin@testlab.local/Inbox/AAPXSgp7AAA=_
↪r20230330-165551.msg
...
```

Best Practices

The following article presents best practices regarding jobs distribution, concurrency and performance.

Jobs Distribution

It is recommended to split the target backup between different users, or even having one job per user. This way errors in one job will not invalidate a whole backup cycle, where some users have been successful, and others experienced errors. This also makes it easier to identify the cause of the error.

Concurrency

When using Exchange EWS APIs, it is possible to find a variety of boundaries that need to be considered. We highlight some of them below:

- Exchange EWS Throttling: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/ews-throttling-in-exchange
- Capabilities of the host serving the Exchange Service
- Usage of the service during the backup window
- Internet Information Server (IIS) limits: https://learn.microsoft.com/en-us/exchange/architecture/client-access/client-message-size-limits?view=exchserver-2019

If a boundary is crossed, the corresponding request will usually fail. Bacula Exchange EWS Plugin is prepared to wait some amount of time and retry it, so it has a certain level of resiliency. However, it is crucial to plan an adequate strategy to backup all the elements without needing to reach any boundary on a regular basis. This means to control how many concurrent requests are done during the backup window.

A single job implements some parallelism which can be reduced until a point, if necessary, using the following parameters:

- `backup_queue_size` - this variable controls the size of internal queues communicating internal threads, that are designed to fetch, open and send every item to Bacula core. Reducing its size will produce, ultimately (with a value of 1 for example), an execution very similar to a single threaded process.
- `concurrent_threads` which controls the number of simultaneous processes fetching and downloading data. This can be reduced or increased to directly affect the concurrency level of a single job.
- `concurrent_listing_threads` controls a different pool of threads intended only to fetch information from the API. It can be reduced to 1, but increasing it over the default values won't change significantly the behavior of the plugin.

The recommended strategy to backup a new environment is to plan a step-by-step testing scenario before putting it into production, where the number of users and the concurrency of the jobs are increased progressively. Other important point is the timing schedule as some boundaries are related to time-frames (number of request per amount of time). If you detect you reach boundaries when running all your backups during a single day of the week, try to increase the time window, and spread the load through it in order to achieve better performance results.

Performance

The performance of this plugin is highly dependent on many external factors:

- Exchange latency and bandwidth
- Network infrastructure
- FD Host hardware
- FD Load
- Ratio number of elements/size
- And many more.

In summary, it is not possible to establish an exact reference about how much time a backup will need to complete.

As a reference and regarding the number of elements and their size:

- Many little objects to protect: More objects per second, but smaller speed (MB/s).
- Big files to protect: Fewer objects per second, but greater speed (MB/s).

It is recommended to benchmark your own environment in base to your requirements and needs.

The automatic parallelization mechanism (using `concurrent_threads=x`) should work well for most scenarios, however, fine-tune is possible if we define one job per user, and we control how many of them run in parallel, together to decrease the `concurrent_threads` value in order to avoid throttling or Exchange server capacity problems.

There are many possible strategies to use this plugin, so it is recommended to study what suits your needs best before deploying the jobs in your entire environment, so you can get the best possible results:

- You can have a job per user and all services.
- You can have multiple entities and only some services inside a job.
- You can split your workload through a schedule, or try to run all your jobs together.
- You can run jobs in parallel or take advantage of `concurrent_threads` and so, run less jobs in parallel.
- You can select what services to backup or backup them all.
- You can backup whole data to backup or select precisely what elements you really need inside each service (folders).
- And more.

Limitations

The following article presents limitations of Exchange EWS Plugin.

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Protection Scope

Only the items that are listed in the features' section of this document are backed up. This means that backup with this plugin **does not include** elements such as the mailbox configuration, mailbox rules or any other database element outside the items the users can directly work with.

If you are interested in including also those elements into your backup strategy, consider the combination of this plugin with the Bacula Enterprise Exchange VSS Plugin, that works at database level.

Legacy public folders, when not connected to a shared and accessible mailbox are not supported. They will be supported in future versions of this plugin.

Backup of Attachments and Files

In general, this plugin backups two types of information:

- Objects
- Files.

Objects are elements representing some item in Exchange such as a calendar event, a contact, an email, etc., while files are attachments of those items.

While objects are directly streamed from memory to the backup engine, files need to be downloaded to the FD host before being sent. This is done in order to perform metadata checks and to improve overall performance, as this the way operations can be parallelized. Every file is removed just after being completely downloaded and sent to the backup engine.

The path used for this purpose is established by the `path` plugin variable, that usually is set up in the backend script (e2ws_backend) with the value: `/opt/bacula/working`.

Inside the `path` variable, a `spool` directory will be created and used for those temporary download processes.

Therefore, it is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs, or through the same job (by default this is the case through the `concurrent_threads=5` parameter), you would need at least that size for the largest file multiplied by the number of operations you run in parallel.

Empty Files

In general, empty files (files with 0 byte contents) are simply not backed up by the Exchange EWS plugin. In particular, item attachments will show a message in the joblog to inform about empty files detected and so, not processed.

Troubleshooting

In this article, there are suggested solutions to common situations that can cause trouble during the usage of the Exchange EWS plugin.

Certificate Problem

The certificate associated to the configured endpoint should be a valid one, or the plugin will reject to connect to it for security reasons.

The Common Name (CN) of the certificate should match the hostname used in the endpoint variable. Otherwise, the plugin will raise errors like: "The request failed. The request failed. Host name '10.10.10.99' does not match the certificate subject provided by the peer".

If the certificate CN uses a hostname, we need to use that hostname from the plugin configuration, instead of the IP address. If needed, because the DNS Server used by the File Daemon host cannot resolve that hostname, the IP-hostname association should be added to the local /etc/hostname file (or equivalent local hostname configuration, depending on the Operative System).

On the other hand, if the certificate is not valid but it is needed to tell the plugin to trust it, the procedure is to add it to the local keystore of the Java Virtual Machine running on the File Daemon. An example of how to do it is given below:

Listing 96: **Throttling unlimited**

```
keytool -cacerts -storepass changeit -importcert -alias ewscert -file␣
↪certificate.cer
```

Note that:

- keytool should be available in the PATH of the system. If it's not, you should look for it inside the Java JRE installation path (bin directory).
- changeit is the default password of the Java keystore. You should change it.
- certificate.cer is the file containing the Exchange certificate to import. You should download it, for instance, using a browser, and connect to Outlook.

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (you will find them in the e2ws-debug.err file), you are very likely using a high level of concurrency through internal `concurrent_threads` parameter and/or parallel jobs.

To overcome this situation you can:

a) Reduce `concurrent_threads` parameter.

b) Reduce the number of jobs running in parallel.

c) If you cannot do that, you should increase JVM memory.

To increase JVM memory, you will need to:

Create this file: '/opt/bacula/etc/e2ws_backend.conf'.

Below, an example of the contents: E2WS_JVM_MIN=2G E2WS_JVM_MAX=8G

Those values will define the MIN (E2WS_JVM_MIN) and MAX (E2WS_JVM_MAX) memory values assigned to the JVM Heap size. In this example, we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough. Be careful if you are running jobs in parallel, as very big values and several jobs at a time could quickly eat all the memory of your host.

The '/opt/bacula/etc/e2ws_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

Throttling

It is possible to manage Exchange throttling policies, and increase them if it detected a high number of requests rejected while doing backup jobs. Below, there is an example of how to configure an unlimited throttling policy for a given account:

Listing 97: **Throttling unlimited**

```
New-ThrottlingPolicy BaculaNoThrottling
Set-ThrottlingPolicy BaculaNoThrottling -RCAMaxConcurrency unlimited -
↪RcaMaxBurst unlimited -RcaRechargeRate unlimited -RcaCutoffBalance unlimited
Set-Mailbox <user or service account> -ThrottlingPolicy BaculaNoThrottling
```

Exchange VSS Plugin

Introduction to Bacula

This chapter is mostly intended for readers without any familiarity with Bacula. We expect that many Windows administrators might decide to research how to back up their Exchange, as the required procedures have changed significantly over time.

Note that the functionality described here is all about traditional Exchange instances, *not* about Microsoft 365 Software-as-a-Service email functionality. For information about backing up and restoring that, refer to *Microsoft 365 Plugin*.

Bacula Enterprise provides commercial, enterprise-level support options and additional features compared to the community-supported open source Bacula software and can be one solution considered by many Exchange administrators. In particular, on top of the "normal" open source advantages provided by the community version, it provides a plugin to back up and recover Microsoft Exchange.

High Level Overview

Bacula, like most other enterprise-ready backup software, consists of several parts which work together through a network:

Director (Dir)
: is the component that controls all operations.

Storage daemon (SD)
: is the component that manages the final targets for backed up data.

File daemon (FD)
: is the *agent* that is run on all systems backed up.

Console
: refers to one of the various user interface programs. Terminal, graphical, and web interface programs are available.

There are more Bacula components, but we don't need to discuss them here. For now, it's sufficient to understand that Bacula File Daemons (**FD** s) are deployed to all your clients to back up and require minimal configuration and maintenance, Bacula Storage Daemons (**SD** s) require a bit more configuration, but normally few configuration changes over time, and all the important configuration exists in one single location, the Bacula Director program (**Dir**)'s configuration.

Configuration

All **Bacula** configuration is traditionally done in simple text files, using a text editor. This may seem a bit inconvenient to many Windows administrators used to graphical user interfaces, but it allows configuration under almost all circumstances, even from a handheld computer and through a low-bandwidth network connection.

A graphical, web-based user interface to operate and configure Bacula Enterprise is available for **Bacula Systems** customers: BWeb Management Suite. This Web GUI allows to configure and execute backing up of Exchange data, as well as most other specific application data.

Platforms

While **FD**s are available for a wide range of operating systems including Windows, the server components of **Bacula Enterprise** require a Linux or Solaris operating system. This may cause some slight culture shock in Windows-only organizations, but experience shows that for systems dedicated to a single purpose, which can essentially be viewed as a black box, this often poses no serious problems. (Actually, even seemingly Windows-only organizations often run a number of different operating systems, often even Linux, on network infrastructure and storage appliances.)

Bacula Systems still recommends that a **Bacula Enterprise** administrator should have some fundamental understanding of the operating system chosen as the platform, especially if disaster recovery is one of the reasons to create backups.

Exchange: An Overview

Microsoft Exchange is Microsoft's server application to provide email, groupware, and unified communication services. Echange runs on Microsoft Windows Servers and consists of a number of services. The actual list of features is quite flexible, as not all organizations will need all the features Exchange can offer.

Backup and restore of Exchange 2003 is not supported by the VSS plugins, so is not discussed in this white paper.

Exchange integrates tightly into Microsoft's Active Directory (AD) technology and can not be run outside of an AD domain. Both AD and Exchange can be run in clusters, providing better performance and higher reliability than single instance installations. All instances making up an Exchange cluster can be managed centrally.

Both the configuration and the data of an Exchange installation can be spread over a number of computers, but will usually be kept consistent between the involved Exchange instances.

However, several distinct Exchange sites may be closely linked to each other, even sharing some configuration information. For purposes of backup and recovery, these need to be handled independently while an Exchange cluster itself could be considered only one backup data source.

Microsoft provides tools and procedures to allow deployment and management of Exchange installations by non-trained administrators, but to get the best out of Exchange, a trained, dedicated administrator is required. Also, to design, implement, test, and maintain backup and recovery procedures an experienced Exchange administrator is helpful. However, **Bacula Systems** believes that, with the knowl edge presented in this White Paper, robust backup and recovery scenarios can be implemented even without that level of in-depth knowledge.

As for any application of a certain complexity, the set of data needed to re-create the full functionality of the application requires restoring both the configuration and the actual (user-) data. As Exchange keeps most of its data in a dedicated database, backing up and restoring that data poses the typical challenges of any database backup:

- Consistency of disk files is not guaranteed while the database engine is running normally, because many changes may still be in the computer RAM not yet written to disk;
- ensuring consistency of disk files requires database-specific tools and knowledge;
- and feeding restored data into the database requires database-specific procedures and knowledge.

The **Bacula Enterprise** VSS plugin handles most of those tasks automatically, only requiring minimal Exchange administrator intervention.

Regarding the configuration of an Exchange installation, backing up the configuration is a bit complicated, because the configuration information is stored in the Windows' Registry, and can be managed and distributed by AD. Accordingly, to allow the full recovery of an Exchange installation, one has to ensure availability or recovery of the base system's configuration including the Registry. The **Bacula Enterprise** VSS Plugin can handle that, as well as the AD information. In an environment with several AD servers, this requires Disaster Recovery (DR) procedures to ensure recovery of at least one AD server.

Naturally, the programs making up Exchange are also required to be running when restoring Exchange data. Ensuring this is in many cases best done by preparing for Bare-Metal Recovery of server machines hosting Exchange.

Microsoft Windows Disaster Recovery is described in more detail in the **Bacula Systems** White Paper "Windows Bare Metal Recovery", so we will not go into much detail here.

The Bacula Enterprise VSS Plugin

Volume Shadow Copy Service (VSS) is the name for Microsoft's snapshot technology, which not only creates snapshots of file systems, but also interfaces to applications to make on-disk data consistent (*freezing, quiescing*) to allow proper backups of application data.

In the simplest case, a vss-aware application's data can simply be backed up from disk, reading from the snapshot, not the original file system. This functionality is part of **Bacula** for a number of years.

However, in order to do online restores (i. e. restores while Windows is running rather than a Bare Metal Recovery) some applications require additional data processing, so using vss alone is not sufficient to back up and restore their data. In particular, understanding of database log files may be required, or, on restore, files that are always locked when Windows is running need to be replaced during a subsequent system boot.

The **Bacula Enterprise** VSS plugin handles the additional data processing needed to be able to do such online restores of many applications, among them Microsoft Exchange.

Please note, in this White Paper, we discuss only the **Bacula Enterprise** version 6.x VSS plugin and not the older Exchange plugin (exchange-fd.dll) that was previously supported by the Bacula project. To avoid all possible interference with the VSS plugin, we recommend that the older exchange-fd.dll should not exist in the plugin directory defined in the **FD** 's configuration file. If it does, Exchange backups and restores could possibly fail.

The VSS plugin needs to be explicitly included in a Fileset used for backups (the plugin takes care of files that would be backed up, making sure no unnecessary or even harmful files get stored).

In addition, the VSS plugin will also create and store some data that needs to be restored or is needed to control the restore process prior to the actual data files being restored. This data is not stored on the backup media, but in **Bacula**'s catalog. This allows restoring data in a different order than it is written, but it requires that all **the Bacula components: Bacula Director program, Bacula Storage Daemon and Bacula File Daemon must be the exact same versions** to ensure that all components know how to handle that additional job data. It also means that you can not restore Exchange data with only the volumes available, after losing the catalog.[1]

As backing up through the vss plugin uses the functionality provided in Microsoft application specific writers, some limitations exist when using this approach to backup. Essentially, **Bacula** can not do things the Microsoft VSS application writers do not provide.

Important Points

For Exchange, some important points are:

- In Exchange 2005, the Recovery Storage Groups (RSGs) exist, so in principle, the Exchange Administrator can enable restore to the RSG, and whe Bacula does a full restore it will go into the RSG rather than restore to the active Exchange database. Once the RSG is restored, the Windows Exchange Administrator can do individual mailbox restores. We have not yet tested this.
- In Exchange 2010 Microsoft eliminated the Recovery Storage Group code that was in previous versions of ex. The Recovery Storage Group allowed the Exchange Administrator to automatically direct a restore to the Recovery Storage Group. Thus other techniques such as use of the Recovery Database are necessary for individual mail box recovery.
- Exchange 2010 permits a single Recovery Database (RDB), which works somewhat similarly to the Recovery Storage Groups in prior versions. The difference is to restore to a Recovery Database,

[1] Regularly saving the catalog database dumps is a practice strongly recommended by anyway, so this should not be an issue in any production system.

the user must explicitly specify the restore location. This feature is not implemented in the initial version 6.0 release of the VSS plugin but available as of **Bacula Enterprise** version 6.0.4.

- The Exchange VSS plugin is supplied in a separate **Bacula Enterprise** VSS plugin installer that must be executed after installing the File daemon.
- The Exchange VSS plugin described in this white paper may not work with the old **exchange-fd.dll** plugin installed in the PluginDirectory. Please make sure it is not present. How to check which plugins are loaded is described in the general VSS Plugin White paper provided by **Bacula Systems**.
- We have not fully tested the Differential backup feature, and recommend that you not use it, since Microsoft's concept of Differential backups is different from Bacula's. Doing a Differential backup after several Incremental backups, could possibly result in lost log files and thusunrecoverable databases.
- We recommend that you use Exchange 2010 SP2 since it corrects many of the problems in previous Exchange versions.
- When you do a restore, you must always do a full restore (item 5 on the restore prompt) that includes the Full backup as well as all the Incremental backups. If you attempt to select Jobs to restore by individual JobIds, the restore will fail. This is because in order for the Microsoft VSS writer to work correctly, it needs all the information about all the backup jobs that were made.
- If you unmount (dismount in Microsoft terminology) a database, its time and date stamp will be updated, and thus it and its log files will be backed up on the next Bacula backup. The same is true if you reboot your system. Backing up active Exchange data files without the VSS plugin and not using Accurate mode will cause incomplete data to be stored.
- Bacula's Accurate mode **must** be enabled in your Job resource. If it is not, you will get a warning message, but much worse, the files being backed up will not be the set of files that needs to be backed up (files may be missed or erroneously be backed up). Bacula needs Accurate mode enabled in order to optimize the backup.
- Do not use Bacula's **Accurate** mode to check MD5 (accurate flag '5') or SHA1 (accurate flag '1') signatures as they will produce a large number of error messages during backup operations. In fact, the default criteria for Accurate Mode, "mcs", are sufficient. Using a Signature directive with an MD5 or SHA1 is OK.
- We do not support and do not guarantee that the VSS plugin can properly backup and restore Exchange in a clustered environment. Databases that are part of A Database Availability Group, however, can usually be backed up and restored correctly.
- Due to the tight coupling between Exchange and Active Directory, we do not support backup of Exchange and restore to a machine that has a different Active Directory environment. It should work if Active Directory on both machines is identical, i. e. the machines are members of the same AD domain.

Fortunately, these are usually not serious restrictions, as Exchange by default provides settings to make partial restores unnecessary, so actually restoring Exchange data would only be needed in case of a disaster recovery, never during regular operations. This also implies that Exchange should always be run in a clustered environment (Database Availability Group or DAG), as Microsoft recommends.

Plugin Options

- **index** Creates additional internal parameters needed for Single Item Restore
- **cinclude=<glob>** Specifies the names of mailbox databases to backup. It is possible to specify the `cinclude` parameter multiple times on the plugin command line.
- **cexclude=<glob>** Specifies the names of mailbox databases to exclude from the backup. It is possible to specify the `cexclude` parameter multiple times on the plugin command line.

Backup Scenarios

Due to Microsoft Exchange's feature to not delete mailbox items, but keep them hidden from regular users when deleted, recovery operations to get individual items back should happen rarely, if at all.[2] Thus, backing up Exchange is mostly useful as part of a dr plan. Accordingly, **Bacula Systems**'s procedures focus on complete backup of Exchange and full restores to the original location.

Recovery operations are similarly focused on (Exchange-specific) DR procedures. Single object restores are discussed but should not be the main goal of deploying a backup and recovery tool for Exchange.

For the purposes of this White Paper, DR does not imply a full Bare Metal Restore of the underlying Operating System and all applications and data on it, but refers to the data `only`. In other words, we are concerned about Application-level Disaster Recovery. This has some consequences on the presented scenarios:

- We assume the version of remains the same, i. e. the backed up data was written by an exact identical version of Exchange as the one it is restored to,
- the operating version is the same for the backup source and backup target,
- and the target locations for the Exchange databases remain unchanged or are part of the same Exchange deployment.

To clarify the outlined backup and recovery scenario, the following table may be helpful:

Table 28: Backup and Recovery Scenario

Situation	Next step
Complete Mailbox Server lost	Set up Windows and Exchange with Mailbox Server role, or do Bare-Metal Recovery
All Mailbox Databases lost	Restore complete sequence of all backups to original location
One Mailbox Database lost	Restore complete sequence of this individual Mailbox Database to original location
One Mailbox Database corrupt	Restore complete sequence of this individual Mailbox Database to an alternate location and proceed as necessary
Single Mailbox or single Mailbox Item required	Restore to Recovery Database and pick out desired Items

[2] The Retention policy is defined in Exchange Management Console and can be applied per Database or per Mailbox. Read more about it at technet.microsoft.com/en-us/library/dd297955.aspx

Minimal Exchange

In our examples, we will initially use a single server running in an AD domain. The Exchange instance uses only one Mailbox storage database which is located on a disk drive with the drive letter `D:`, but the main ex installation is located on drive `C:`. Accordingly, to create a useful backup of the Exchange data, the disk drives lettered `C:` and `D:` need to be included in the backup[3]. The resulting Fileset is shown below.

```
File Set {
  Name = Exchange-CD
  Include {
    Options {
      Verify = pnugsi1
      Signature = SHA1
    }
    File = c:/backmeup
    File = d:/backmeup
    Plugin = "vss:/@EXCHANGE/"
  }
}
```

Note the dummy files included on each drive where data is located.

To allow the VSS plugin of **Bacula Enterprise** to work correctly, backup jobs must be done in accurate mode. Accordingly, the job definition we use in our examples includes that option; the job definition we use is shown below.

```
Job {
  Name = Exchange
  Type = Backup
  Level = Incremental
  Accurate = Yes
  File Set = Exchange-CD
  Client = wsb-exch10-fd
  Storage = File
  Messages = Standard
  Pool = Tier1
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%n.bsr"
}
```

Backing Up

To create a backup of a data set, one needs to specify a Fileset to use the Exchange functionality of **Bacula Enterprise**'s VSS plugin as outlined in chapter *The Bacula Enterprise VSS Plugin*. One essential consideration is that at least one file from each drive that is used by Exchange **has** to be included explicitly.

On the Exchange side, databases that are to be backed up with bsee's VSS plugin must be set to **not do circular logging**. This has to be done for each database and can be achieved with the Database Properties panel in Exchange Management Console (see figure *Mailbox Database Maintenance Properties need to Allow Overwriting*, the last line of the checkboxes, below the arrow) or through ps.

[3] Starting with version 12.5, specifying the volumes is not mandatory anymore

Backup Levels

Backups with **Bacula** can be of three levels: Full, Differential, and Incremental. A Full backup is simple to understand; it just backs up everything included in its Fileset then allows Exchange to remove the log files that are no longer needed. A Differential level backup backs up everything changed since the latest Full level backup, but does not trigger log file removal. Incremental backup takes everything since the last backup job run and allow Exchange to remove any log files that are no longer needed.

Transaction Log Truncation

A transaction log as written by many database engines stores all database transactions done so they can be replayed in a recovery situation. To ensure that disk space used by those logs can be reclaimed, those logs should be truncated from time to time. When using the VSS Exchange plugin, log truncation is automatically done by the Windows Exchange writer when a Full or an Incremental backup succeeds.

Single Database Backup

A single database backup is possible, by excluding other databases from the file set and including the wanted database. Code below presents a fileset structure where 4 databases are present on volume G[4]. MDB03 will be backuped while MDB01, MDB02 and MDB04 will be excluded.

```
File = "G:/backmeup.txt" ## volume G: will be part of the snapshot

Plugin = "vss:/@EXCHANGE/
   cexclude=*/*/*/MDB01/*   ## to cexclude MDB01/Log or MDB01/File
   cexclude=*/*/*/MDB01      ## to cexclude MDB01 folder
   cexclude=*/*/*/*/MDB01/* ## extra level to cexclude MDB01/Log or MDB01/
↪File when in a Replica
   cexclude=*/*/*/*/MDB01   ## extra level to cexclude MDB01 folder when it's␣
↪a Replica

## same for DB02
   cexclude=*/*/*/MDB02/* cexclude=*/*/*/MDB02 cexclude=*/*/*/*/MDB02/*␣
↪cexclude=*/*/*/*/MDB02
## same for DB04
   cexclude=*/*/*/MDB04/* cexclude=*/*/*/MDB04 cexclude=*/*/*/*/MDB04/*␣
↪cexclude=*/*/*/*/MDB04

## cinclude DB03 with the same logic
   cinclude=*/*/*/MDB03/* cinclude=*/*/*/MDB03 cinclude=*/*/*/*/MDB03/*␣
↪cinclude=*/*/*/*/MDB03
```

However, a VSS snapshot is always global to a single volume. In this example all databases present on volume G: will get their transaction logs truncated, although some are not backuped. To avoid this, consider separating the databases on different volumes or backup all databases present on the volume in 1 single job.

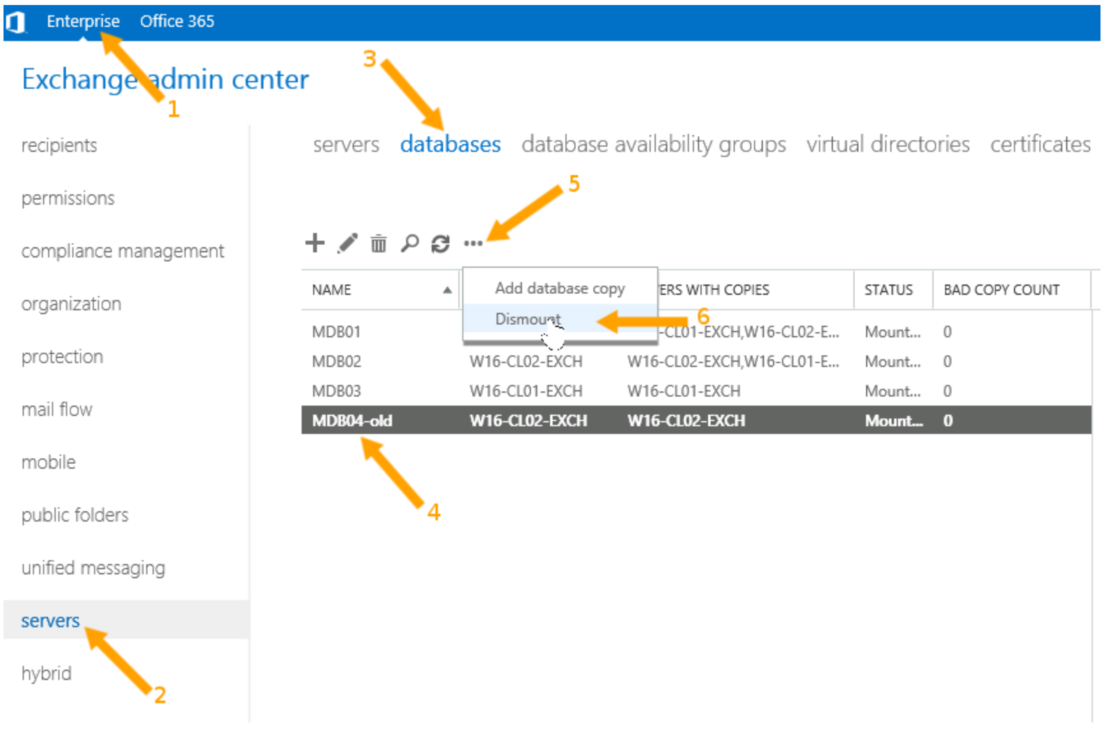
[4] Starting with version 12.5, specifying the volumes is not mandatory anymore

Differential Backups

We have not sufficiently tested Differential backups with Exchange, and suspect that due to a difference in concept of Differential backups between Exchange and Bacula, using Differential backups may result in lost log files. Thus we recommend that you do not use Differential backups with the Exchange plugin.

Recovery

The Mailbox Database needs to be unmounted and marked to allow overwriting to be restored. An example on how to unmount a Mailbox Database is shown below. In `Exchange Administrative Center`, navigate to the database to restore, select it, click on `...` and then `Dismount`:



NAME		ERS WITH COPIES	STATUS	BAD COPY COUNT
MDB01		-CL01-EXCH,W16-CL02-E...	Mount...	0
MDB02	W16-CL02-EXCH	W16-CL02-EXCH,W16-CL01-E...	Mount...	0
MDB03	W16-CL01-EXCH	W16-CL01-EXCH	Mount...	0
MDB04-old	W16-CL02-EXCH	W16-CL02-EXCH	Mount...	0

Fig. 38: Preparing an Exchange Database for Restore with the `Exchange Administrative Center`

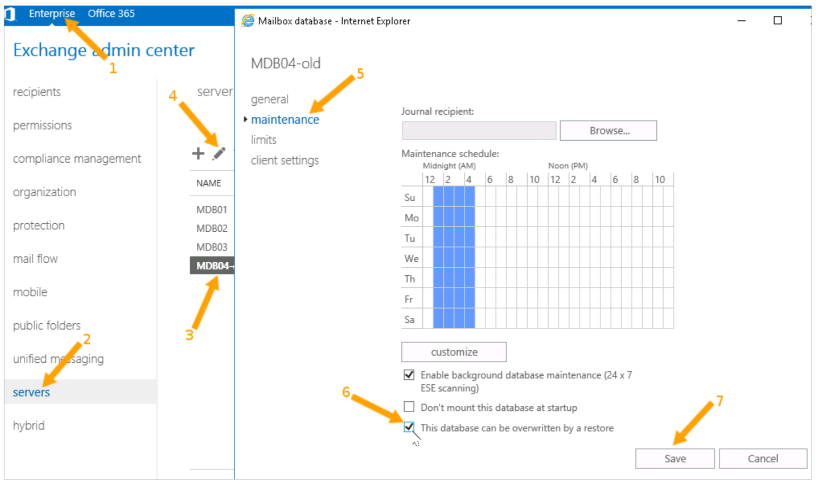
Then edit the Database Mailbox Maintenance properties by clicking on the pencil icon and checking the box `This Database can be Overwritten by a Restore`:

Alternatively, these steps can also be done with **PowerShell** commands. Please refer to the documentation available with the Microsoft Exchange Management Shell.

If the database to be restored exists, for example when testing, or if parts of its files still exist (which might be the case after a file system fault) it is best to remove the remaining parts before restoring.

To do this to just safely rename the whole database directory with Window's shell, `cmd`. The path can be found through `Exchange Administrative Center`, from the Database Properties panel, as shown in figure *Database Path in Exchange Administrative Center*, or with the **PowerShell**-based Exchange Management Shell. A way to do this, and the result, is shown in figure *Database Path in Exchange Management Shell*.

The full database path and name cannot be edited but can be selected and copied.



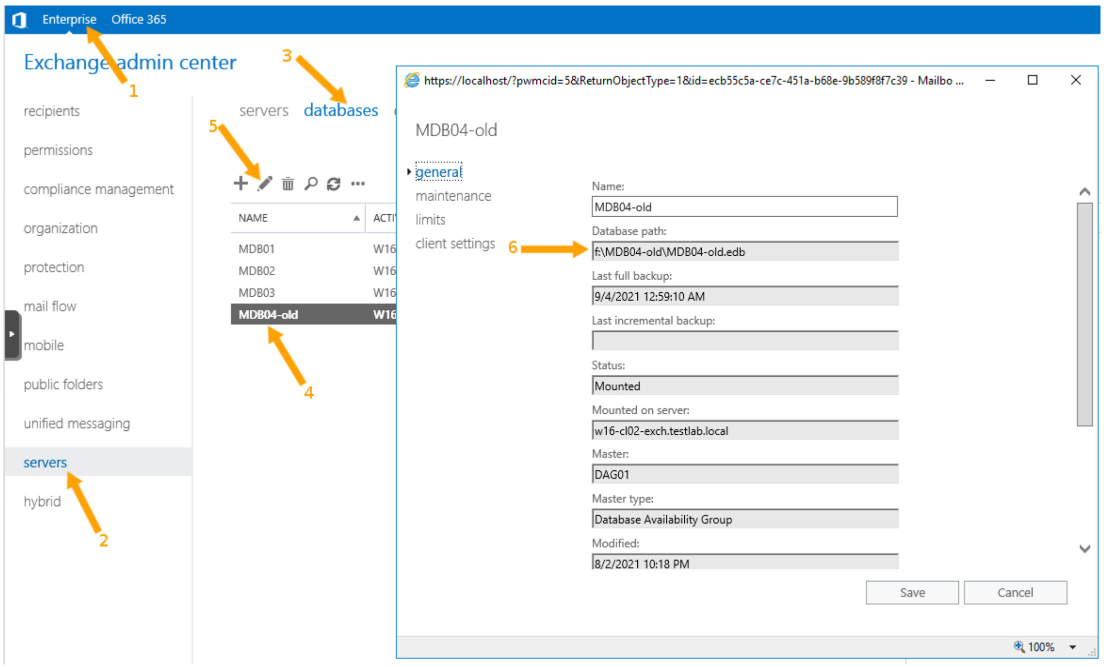
Journal recipient:

☑ Enable background database maintenance (24 x 7 ESE scanning)

☐ Don't mount this database at startup

☑ This database can be overwritten by a restore

Fig. 39: Mailbox Database Maintenance Properties need to Allow Overwriting



Name:
MDB04-old

Database path:
f:\MDB04-old\MDB04-old.edb

Last full backup:
9/4/2021 12:59:10 AM

Last incremental backup:

Status:
Mounted

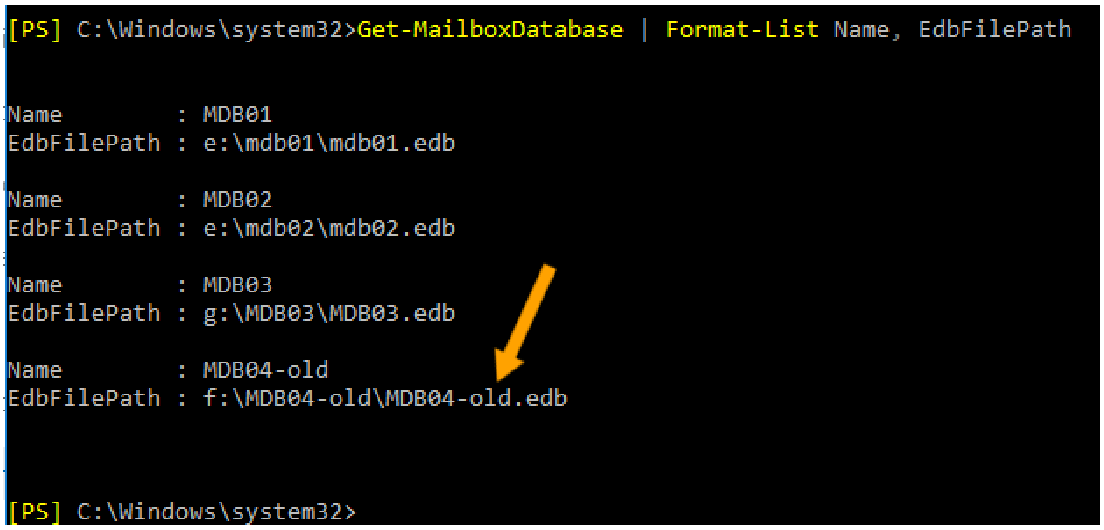
Mounted on server:
w16-cl02-exch.testlab.local

Master:
DAG01

Master type:
Database Availability Group

Modified:
8/2/2021 10:18 PM

Fig. 40: Database Path in `Exchange Administrative Center`



```
[PS] C:\Windows\system32>Get-MailboxDatabase | Format-List Name, EdbFilePath

Name        : MDB01
EdbFilePath : e:\mdb01\mdb01.edb

Name        : MDB02
EdbFilePath : e:\mdb02\mdb02.edb

Name        : MDB03
EdbFilePath : g:\MDB03\MDB03.edb

Name        : MDB04-old
EdbFilePath : f:\MDB04-old\MDB04-old.edb


[PS] C:\Windows\system32>
```

Fig. 41: Database Path in `Exchange Management Shell`

The path name is shown completely – remember that you need the whole directory, not only the `.edb` file.

The fault that can be observed when restoring with database files remaining is caused by out-of-sequence log files that are found by the Exchange-specific database recovery and is reported in the Job Report with lines like these:

```
Exchange VSS Writer failed restoring a backup with error code -515
when performing an integrity-check of the log files to be used for
database recovery after restore for 'Mailbox Database 1568811476'.
```

If you encounter those problems the safest way to proceed is to redo the whole restore process, this time removing the remaining database files before the actual restore job starts.

Restore procedure with Bat

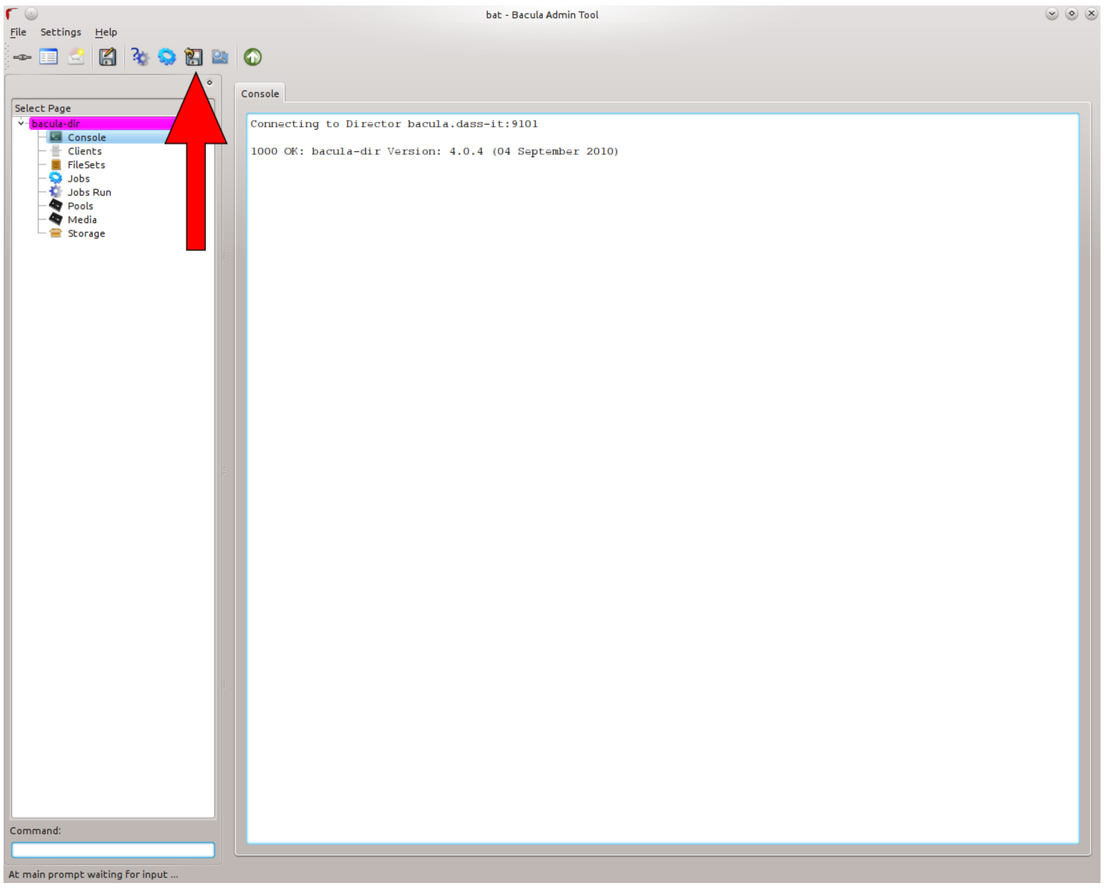
In `BAT` press the restore Button (see figure *Starting the Restore Procedure*).

Select the corresponding job for your exchange server and choose the "@EXCHANGE" folder presented in the directory overview (see figure *Restore Selection*)

It is important to remove the directive in the "Where" line as shown in figure *Restore Where*, otherwise the database may be restored to an alternate location, which is not supported without additional preparation. This is explained in chapter *Restoring to an Alternative Location*.

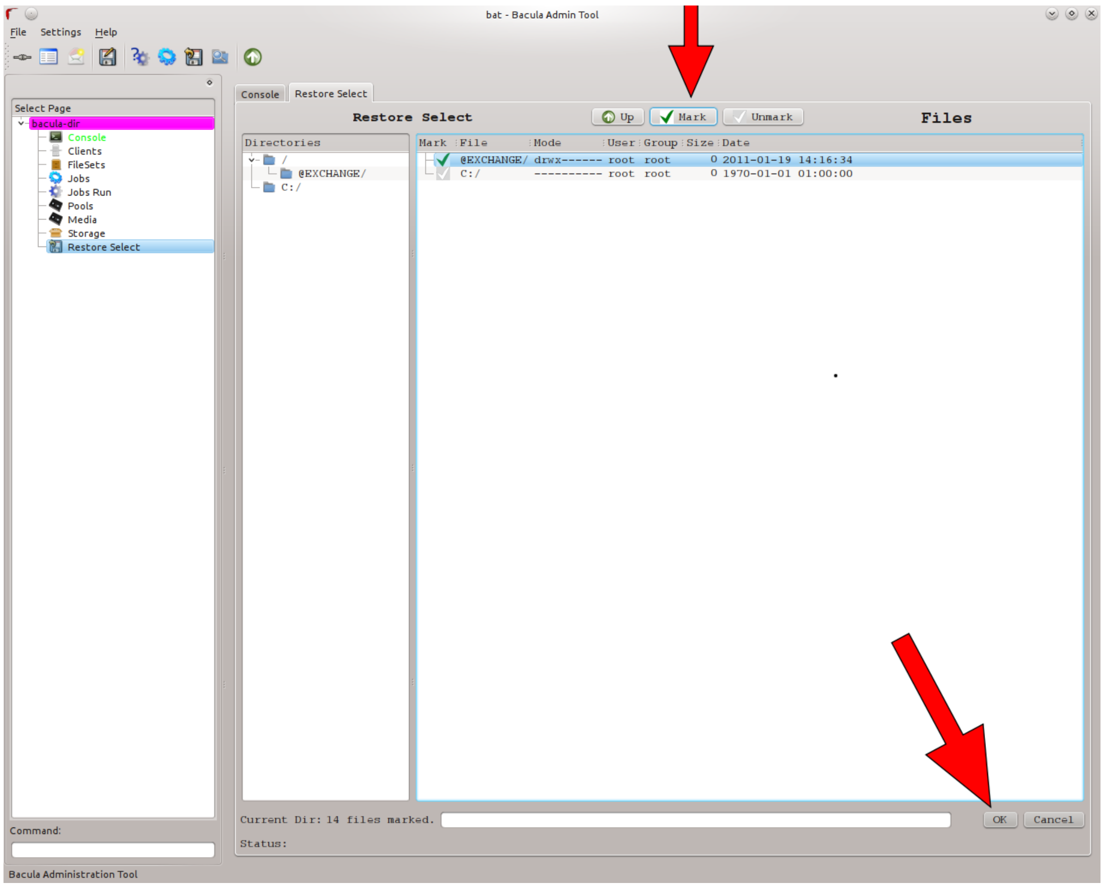
After the actual restore (which could also be initiated from a **Bacula** console) is finished the Exchange machine may need to be rebooted, during which some of the actual database files will be moved to their final locations. If the reboot is needed, Bacula will print a message to that effect in the Job report. After the reboot, or after the restore is done, the administrator will need to mount the restored database(s), which is done similar to dismounting, and the server will be functional again.

For reasons of safety, **Bacula Systems** recommends to verify that the setting to allow the database to be overwritten is turned off when mounting it after the server's restart.



```
Connecting to Director bacula.dass-it:9101

1000 OK: bacula-dir Version: 4.0.4 (04 September 2010)
```

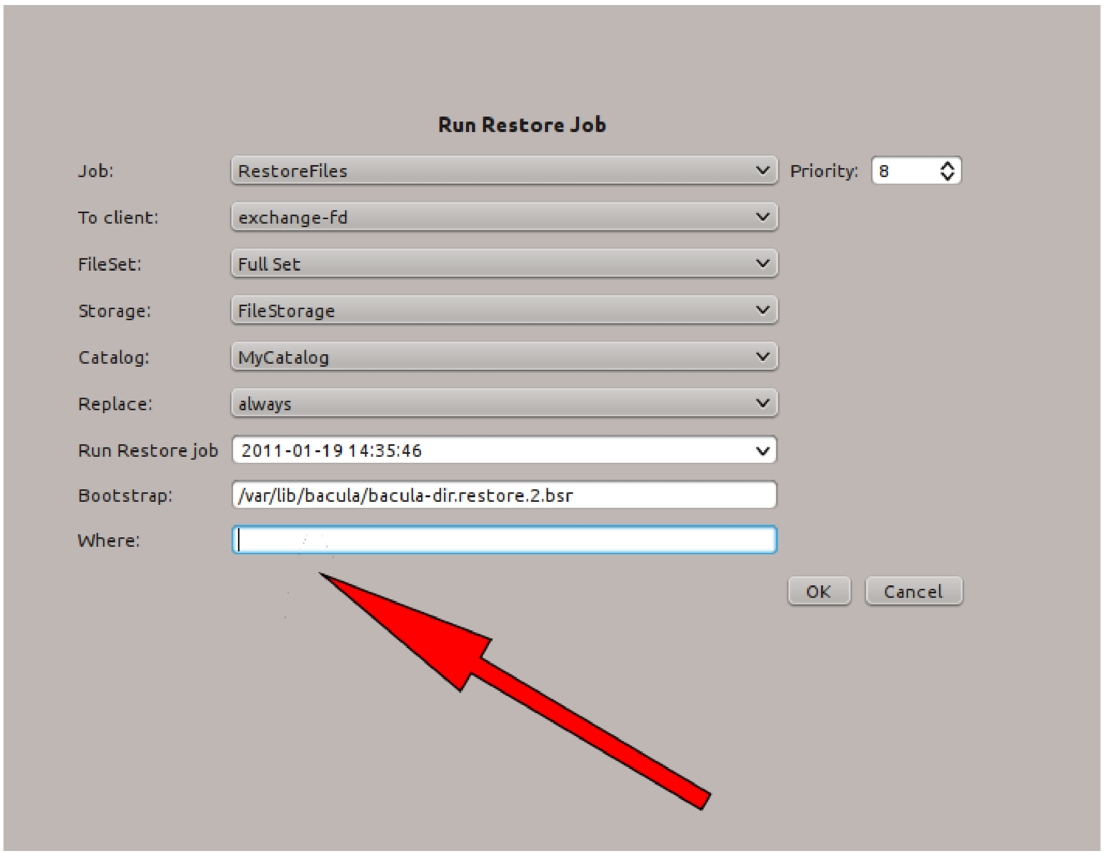
Fig. 42: Starting the Restore Procedure



```
Mark File        Mode        User Group Size Date
 ✓   @EXCHANGE/  drwx------  root root     0 2011-01-19 14:16:34
     C:/         ----------  root root     0 1970-01-01 01:00:00
```

```
Current Dir: 14 files marked.
Status:
```

Fig. 43: Restore Selection


Run Restore Job

Job:	RestoreFiles
Priority:	8
To client:	exchange-fd
FileSet:	Full Set
Storage:	FileStorage
Catalog:	MyCatalog
Replace:	always
Run Restore job	2011-01-19 14:35:46
Bootstrap:	/var/lib/bacula/bacula-dir.restore.2.bsr
Where:	

Fig. 44: Restore Where

Restoring a Single Database

By default, if you have multiple Exchange databases, they will all be restored when following the procedures given above. If you wish to restore only a single database, it is possible, but during the restore file selection procedure instead of marking the "@EXCHANGE" folder presented in the directory overview, you must descend several levels – usually four – into the "@EXCHANGE" directory, until you find the names of all the databases that you have. At that point you may mark only the database or databases that you want restored. An example of this, where only one database is selected for restore is given below.

```
$ pwd
cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/
↪Microsoft Information Store/WSB-EXCH10/
$ ls
Mailbox A/
Mailbox Database 1568811476/
mark "Mailbox Database 1568811476"
16 files marked.
$
```

Database Availability Groups

Database Availability Groups (DAG) is one of the main new features of Exchange 10. The DAG provides automatic, database-level recovery from a database, server, or network outage. In order to use the full potential of the DAG groups, two or more Exchange Servers are required. DAG is based on continuous replication, enabling the Exchange environment to offer high availability and site resilience. In a DAG the recovery mechanism is done automatically. Hence if a mailbox in the Exchange cluster has a failure another Exchange Server (within the cluster) will automatically fix the database failure. In a scenario of a Database Availability Group, the database is replicated to all involved Servers. In this situation, the Backup of only one instance is necessary because all databases are kept synchronized.

Other Ways to Cluster Exchange

There are several other ways to cluster environments. In order to give a brief but sophisticated overview it will be focused on mainly two scenarios in order to cover a simple and a more advanced solution. The first solution is the easiest one. To cluster an Exchange Server a shared storage is used and a Heartbeat in order to monitor the two nodes. If one of the Exchange Servers now realizes a fail-over situation, the running node takes over all connections and work in order to still provide availability. In this scenario no separation between the database and server is implied.

The second scenario involves in total four Exchange Servers where three actives nodes are productive and the fourth is set in a passive mode in order to cover the fail-over case. In the background redundant, shared mass storage is used to store logs and database files. With this environment an effective usage of the hardware is possible as well as a high level of security in terms of availability and consistency of the database is provided. It has to be mentioned that, if this environment is used, the database files have to be saved from just one Exchange server, since data is stored on a central, shared storage device. The other Exchange server instances will automatically pick up database changes.

Restoring to an Alternative Location

Note: This procedure is available for Exchange 2010 and later, not with older versions.

In order to do restores of single mailboxes or single mailbox items, it is necessary that the administrator restores to a secondary database and then moves the interesting items to their final destination.

For this purpose, Microsoft provides special *Recovery Databases* which are used to (temporarily) store the recovered objects before they are moved to their final location. These Recovery Databases are managed quite differently than regular Mailbox Databases, as they are only intended as a temporary storage during recovery operations. See http://technet.microsoft.com/en-us/library/dd876954.aspx for details.

Creating and operating on Recovery Databases is done exclusively through Exchange Management Shell; **Bacula Systems** provides scripts making these tasks easier.

The high-level overview of the steps required to restore individual items looks like this:

1. If necessary, create a Recovery Database.
2. Unmount this database and enable restores to it (this is the same procedure as described in *Recovery*)
3. Restore data (usually restoring an individual database, as in section *Restoring a Single Database*). Use a `where=` setting to direct the restored data to the Recovery Database.
4. Move the items of interest to their final location using Microsoft Exchange Management Shell.
5. After checking the success of the restore, remove the Recovery Database.

All those steps can be done through Powershell commands with Microsoft Exchange Management Shell, and some of them can only be done this way.

Most of those steps have to be done on a Windows machine and need to work with the ex server data will be restored to. For stability reasons, **Bacula Systems** recommends that you run the actual restore to the ex server that physically hosts the recovery mailbox, and in this case it's most convenient to work on that server itself.

A typical restore session done the way **Bacula Systems** recommends accordingly looks like this:

1. Log on to a ex mailbox database server with credentials allowing to manage Microsoft Exchange.
2. Open the Microsoft Exchange Management Shell. In the start menu, this will look similar to figure *Microsoft Exchange Management Shell in the Start Menu*.
3. A command windows opens which will look similar to the one in figure *Microsoft Exchange Management Shell Startup Window*. You should know the Powershell essentials if you're an Administrator; in-depth information can be found on Microsoft's web site, for example at http://technet.microsoft.com/de-de/library/bb123778
4. Use appropriate commands to set up a Recovery Database and enable restores to it. **Bacula Systems** provides an example script which is also attached to this white paper in its pdf version. This script should, if stored in `C:\Users\Administrator\Documents`, be sourced with the``Powershell`` dot command, i. e. `."\C:\Users\Administrator\Documents\file.ps1"`. An example can be seen in figure *Calling the Preparation Script*, and sample output is shown in figure *Preparation Script Output*. In *Script to Prepare Recovery Database* we provide the listing of this script. The commands that are essential are `New-MailboxDatabase` and `Set-MailboxDatabase`.
5. Do the actual restore with **Bacula Enterprise**

   1. Set up the restore job as usual, using any convenient console. Make sure you restore a full sequence of backups beginning with a full level backup, and all jobs based on this one up to the point in time required.
   2. Select the files to restore – normally, a specific database will be selected for restore. See above in **ch:singledbrestore** for details.
   3. Make sure no file name mangling is done by adding a prefix or suffix.
   4. Set the restore location to the Recovery Database name. Do not use the path but the plain name!
   5. Let the job execute.

   An example restore session using `bconsole` is shown in figure *Restore Job Relocating a Single Mailbox Database*.

6. Move the relevant items from the Recovery Database to their final location. The Recovery Database has to be mounted for that purpose. The following steps can be used:
   1. Execute `Mount-Database Bacula EnterpriseRecovery` (substitute the Recovery Database name if necessary) in .
   2. In , you can use the following command to get a listing of all the mailboxes that exist in the Recovery Database (again, substitute the Recovery Database Name if necessary):

      ```
      Get-MailboxStatistics -Database Bacula EnterpriseRecovery | Format-
      ↪List DisplayName
      ```

   3. The following Powershell command in restores a complete mailbox from the Recovery Database to the regular user's database (which must exist) into a dedicated folder:

      ```
      Restore-Mailbox -Identity "DisplayName" -RecoveryDatabase Bacula␣
      ↪EnterpriseRecovery -TargetFolder "Recovered 2010-07-10" -
      ↪RecoveryMailbox "DisplayName"
      ```

      The `DisplayName` must be the displayed name of an existing mailbox – the source, i. e. the mailbox inside the Recovery Database is identified by the `RecoveryMailbox` parameter.
   4. The user can now arrange (or delete) recovered items as needed. It should be noted that items are restored without any permissions set, so that the user may have to give herself permissions to manage the restored folders.

      An example of the process and results of this recovery procedure is given in figures *Restoring a Mailbox with Microsoft Exchange Management Shell* and *Restored Mailbox and Folder Permissions*.
7. To clean up after all data is moved to its final location, unmount the Recovery Database, remove it from ex, and delete the data files on disk. The exs commands `Dismount-Database`, `Remove-MailboxDatabase` and `Remove-Item -Recurse` are suitable to achieve that.

Listing 98: Restore Job Relocating a Single Mailbox Database

```
  *restore client=wsb-exch10-fd before="2012-07-09 23:59:00" where="Bacula␣
↪EnterpriseRecovery"
   restoreclient=wsb-exch10-fd

  First you select one or more JobIds that contain files
  to be restored. You will be presented several methods
```

(continues on next page)

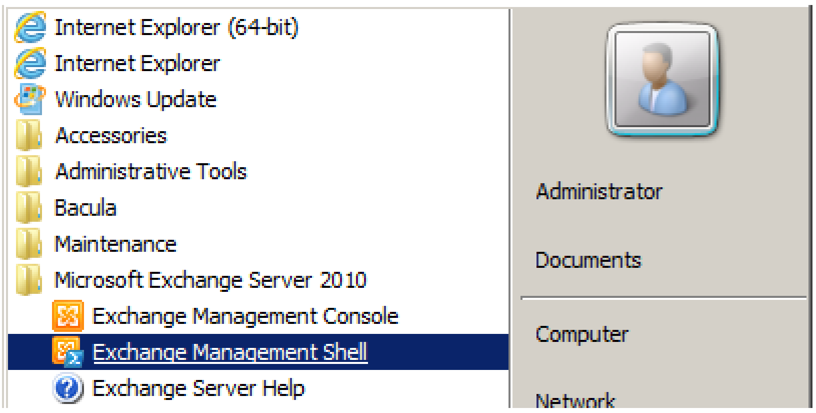
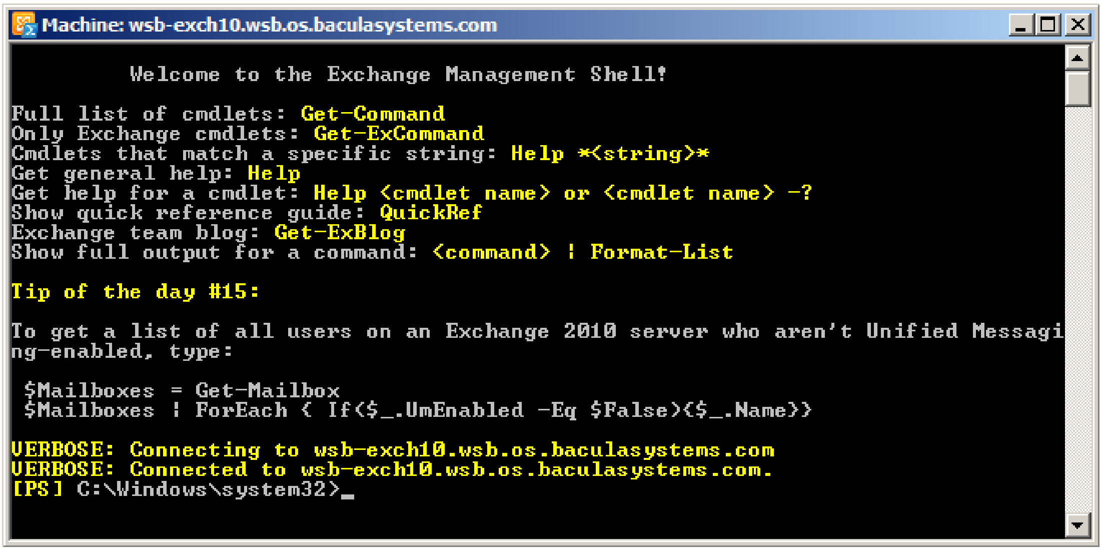
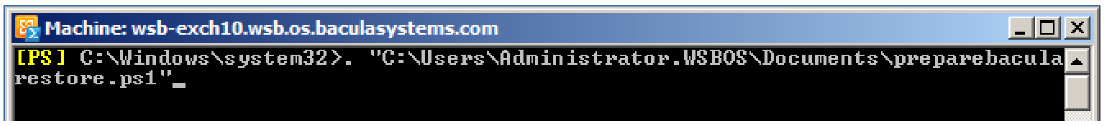


Fig. 45: Microsoft Exchange Management Shell in the Start Menu



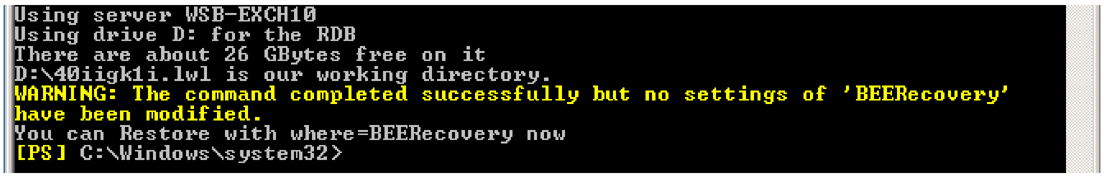
```
         Welcome to the Exchange Management Shell!

Full list of cmdlets: Get-Command
Only Exchange cmdlets: Get-ExCommand
Cmdlets that match a specific string: Help *<string>*
Get general help: Help
Get help for a cmdlet: Help <cmdlet name> or <cmdlet name> -?
Show quick reference guide: QuickRef
Exchange team blog: Get-ExBlog
Show full output for a command: <command> | Format-List

Tip of the day #15:

To get a list of all users on an Exchange 2010 server who aren't Unified Messagi
ng-enabled, type:

 $Mailboxes = Get-Mailbox
 $Mailboxes | ForEach { If($_.UmEnabled -Eq $False){$_.Name}}

VERBOSE: Connecting to wsb-exch10.wsb.os.baculasystems.com
VERBOSE: Connected to wsb-exch10.wsb.os.baculasystems.com.
[PS] C:\Windows\system32>_
```

Fig. 46: Microsoft Exchange Management Shell Startup Window



```
[PS] C:\Windows\system32>. "C:\Users\Administrator.WSBOS\Documents\preparebacula
restore.ps1"_
```

Fig. 47: Calling the Preparation Script



```
Using server WSB-EXCH10
Using drive D: for the RDB
There are about 26 GBytes free on it
D:\40iigk1i.lwl is our working directory.
WARNING: The command completed successfully but no settings of 'BEERecovery'
have been modified.
You can Restore with where=BEERecovery now
[PS] C:\Windows\system32>
```

Fig. 48: Preparation Script Output

(continued from previous page)

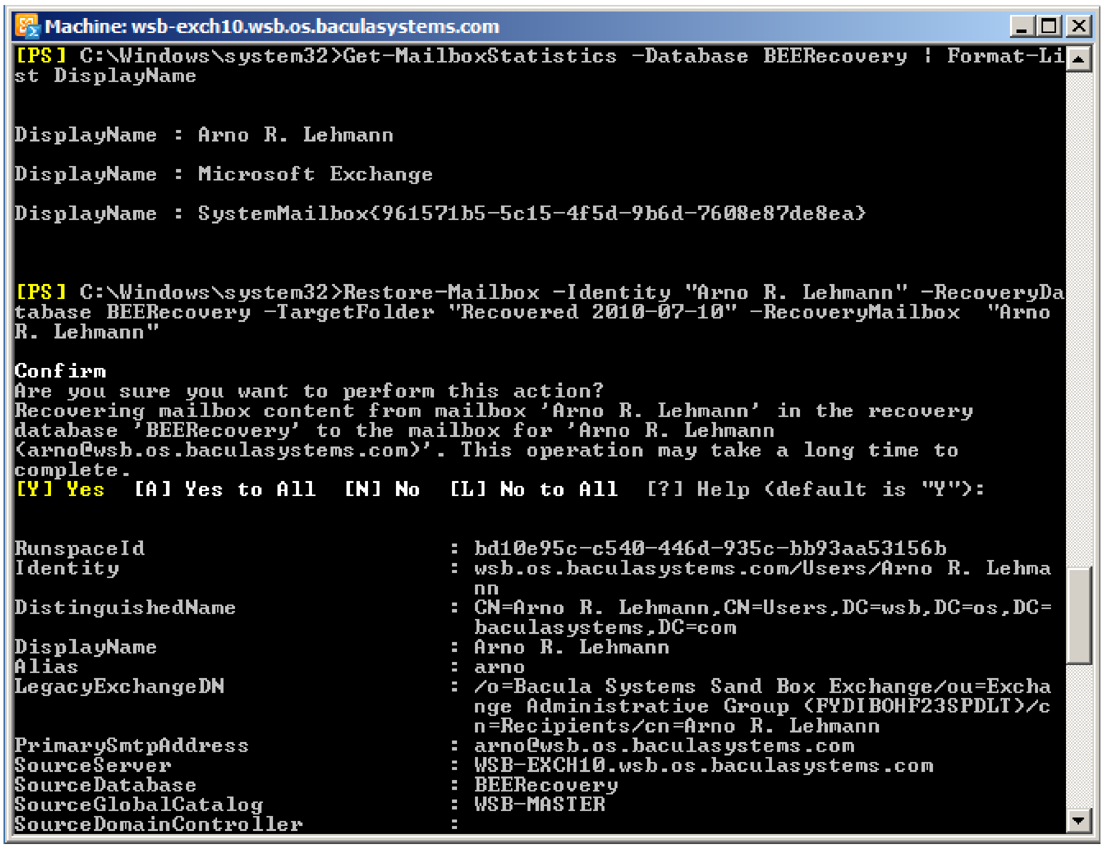
```
  of specifying the JobIds. Then you will be allowed to
  select which files from those JobIds are to be restored.

  To select the JobIds, you have the following choices:
  ...
       5: Select the most recent backup for a client
  ...
      13: Cancel
  Select item:  (1-13): 5
  The defined Fileset resources are:
       1: Exchange-CD
       2: Exchange-SS
       3: WindowsUserC
  Select Fileset resource (1-3): 1
  +-------+-------+----------+-------------+---------------------+-----------
↪-+
  | jobid | level | jobfiles | jobbytes    | starttime           |␣
↪volumename |
  +-------+-------+----------+-------------+---------------------+-----------
↪-+
  |   362 | F     |      258 | 397,610,415 | 2012-07-09 00:26:41 | F-0007   ␣
↪ |
  |   363 | I     |       23 |   6,294,004 | 2012-07-09 00:32:58 | F-0007   ␣
↪ |
  +-------+-------+----------+-------------+---------------------+-----------
↪-+
  You have selected the following JobIds: 362,363

  Building directory tree for JobId(s) 362,363 ... ␣
↪+++++++++++++++++++++++++++
  +++++++++++++++
  257 files inserted into the tree.

  You are now entering file selection mode where you add (mark) and
  remove (unmark) files to be restored. No files are initially added, unless
  you used the "all" keyword on the command line.
  Enter "done" to leave this mode.

  cwd is: /
  $ cd @EXCHANGE
  cwd is: /@EXCHANGE/
  $ cd "Microsoft Exchange Writer/"
  cwd is: /@EXCHANGE/Microsoft Exchange Writer/
  $ cd "Microsoft Exchange Server/"
  cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/
  $ cd "Microsoft Information Store/"
  cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/
↪Microsoft
   Information Store/
  $ cd WSB-EXCH10/
  cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/
↪Microsoft
```

(continues on next page)

(continued from previous page)

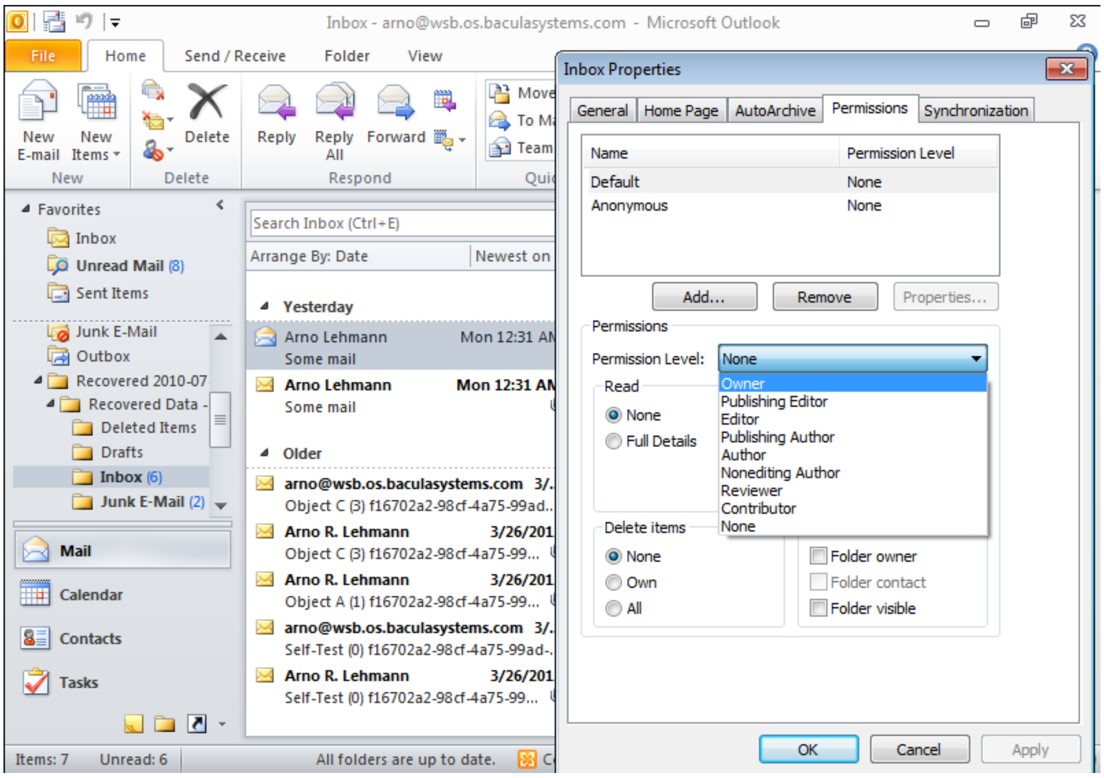
```
 Information Store/WSB-EXCH10/

$ ls
Mailbox A/
Mailbox Database 1568811476/
$ mark "Mailbox Database 1568811476"
131 files marked.
$ done

Bootstrap records written to /var/lib/bacula/s-1-dir.restore.10.bsr

The job will require the following
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    F-0007                    File                      FileStorage

Volumes marked with "*" are online.

133 files selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /var/lib/bacula/s-1-dir.restore.10.bsr
Where:           Bacula EnterpriseRecovery
Replace:         always
Fileset:         Test Set
Backup Client:   wsb-exch10-fd
Restore Client:  wsb-exch10-fd
Storage:         File
When:            2012-07-09 00:38:03
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=364
*
```

Permission level “Owner” allows the user to manage and delete restored folders.



```
[PS] C:\Windows\system32>Get-MailboxStatistics -Database BEERecovery | Format-Li
st DisplayName


DisplayName : Arno R. Lehmann

DisplayName : Microsoft Exchange

DisplayName : SystemMailbox{961571b5-5c15-4f5d-9b6d-7608e87de8ea}



[PS] C:\Windows\system32>Restore-Mailbox -Identity "Arno R. Lehmann" -RecoveryDa
tabase BEERecovery -TargetFolder "Recovered 2010-07-10" -RecoveryMailbox  "Arno
R. Lehmann"

Confirm
Are you sure you want to perform this action?
Recovering mailbox content from mailbox 'Arno R. Lehmann' in the recovery
database 'BEERecovery' to the mailbox for 'Arno R. Lehmann
(arno@wsb.os.baculasystems.com)'. This operation may take a long time to
complete.
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [?] Help (default is "Y"):


RunspaceId                        : bd10e95c-c540-446d-935c-bb93aa53156b
Identity                          : wsb.os.baculasystems.com/Users/Arno R. Lehma
                                    nn
DistinguishedName                 : CN=Arno R. Lehmann,CN=Users,DC=wsb,DC=os,DC=
                                    baculasystems,DC=com
DisplayName                       : Arno R. Lehmann
Alias                             : arno
LegacyExchangeDN                  : /o=Bacula Systems Sand Box Exchange/ou=Excha
                                    nge Administrative Group (FYDIBOHF23SPDLT)/c
                                    n=Recipients/cn=Arno R. Lehmann
PrimarySmtpAddress                : arno@wsb.os.baculasystems.com
SourceServer                      : WSB-EXCH10.wsb.os.baculasystems.com
SourceDatabase                    : BEERecovery
SourceGlobalCatalog               : WSB-MASTER
SourceDomainController            :
```

Fig. 49: Restoring a Mailbox with Microsoft Exchange Management Shell



Name	Permission Level
Default	None
Anonymous	None

Permissions

Permission Level: None

Owner
Publishing Editor
Editor
Publishing Author
Author
Nonediting Author
Reviewer
Contributor
None

Read: [✓ None] Full Details

Delete items: [✓ None] Own All

Folder owner, Folder contact, Folder visible

Arno Lehmann	Mon 12:31 AM	Some mail
Arno Lehmann	Mon 12:31 AM	Some mail
arno@wsb.os.baculasystems.com	3/.	Object C (3) f16702a2-98cf-4a75-99ad..
Arno R. Lehmann	3/26/201	Object C (3) f16702a2-98cf-4a75-99...
Arno R. Lehmann	3/26/201	Object A (1) f16702a2-98cf-4a75-99...
arno@wsb.os.baculasystems.com	3/.	Self-Test (0) f16702a2-98cf-4a75-99ad-.
Arno R. Lehmann	3/26/201	Self-Test (0) f16702a2-98cf-4a75-99...

Fig. 50: Restored Mailbox and Folder Permissions

Troubleshooting

As with any complex application, problems during backup or recovery may happen with Microsoft Exchange. In this chapter, we try to guide you through resolving some problems.

Backup

The most common problem during backup is that not all required file systems are included in the snapshot set; in these cases, the VSS writer responsible for handling ex will refuse the snapshot's creation. The solution is to explicitly include the required drive by adding a dummy file entry to the file set used.

If the backup itself works, but no files or not all required files are being backed up, this is often caused by some of the required infrastructure not being functional. For example, in a DAG configuration, if the *Microsoft Exchange Replication* service is not running because of a cluster environment issue, backups may be incomplete.

Restore

In most cases, when **Bacula Enterprise** reports a restore as successful, but actual Mailbox access shows no restore took place, the cause of the problem can be found in Windows' Event log. In most cases, detailed error messages will also be included in the Job report. For those Events that we encountered, we have a listing prepared below, explaining how to proceed.

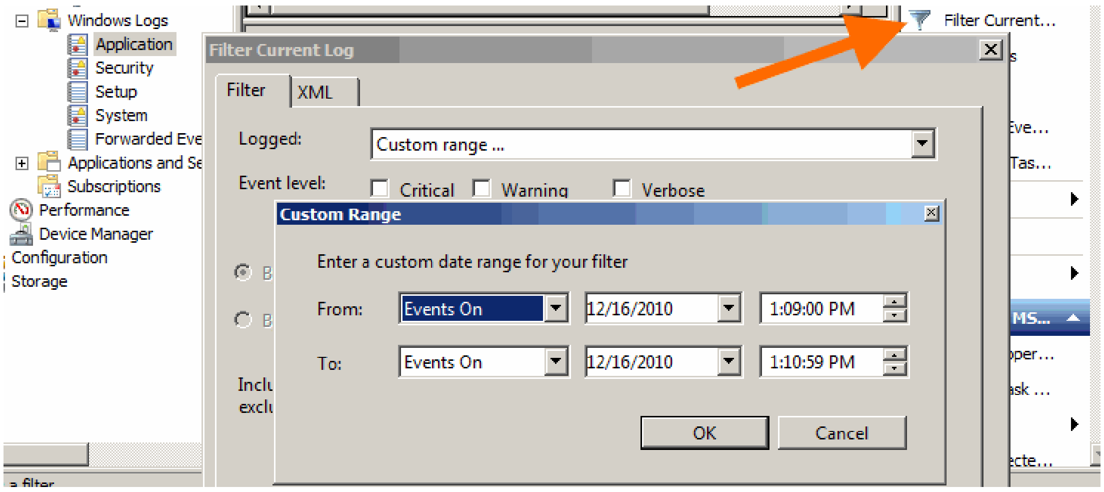


Fig. 51: Windows' Event Log Filtering

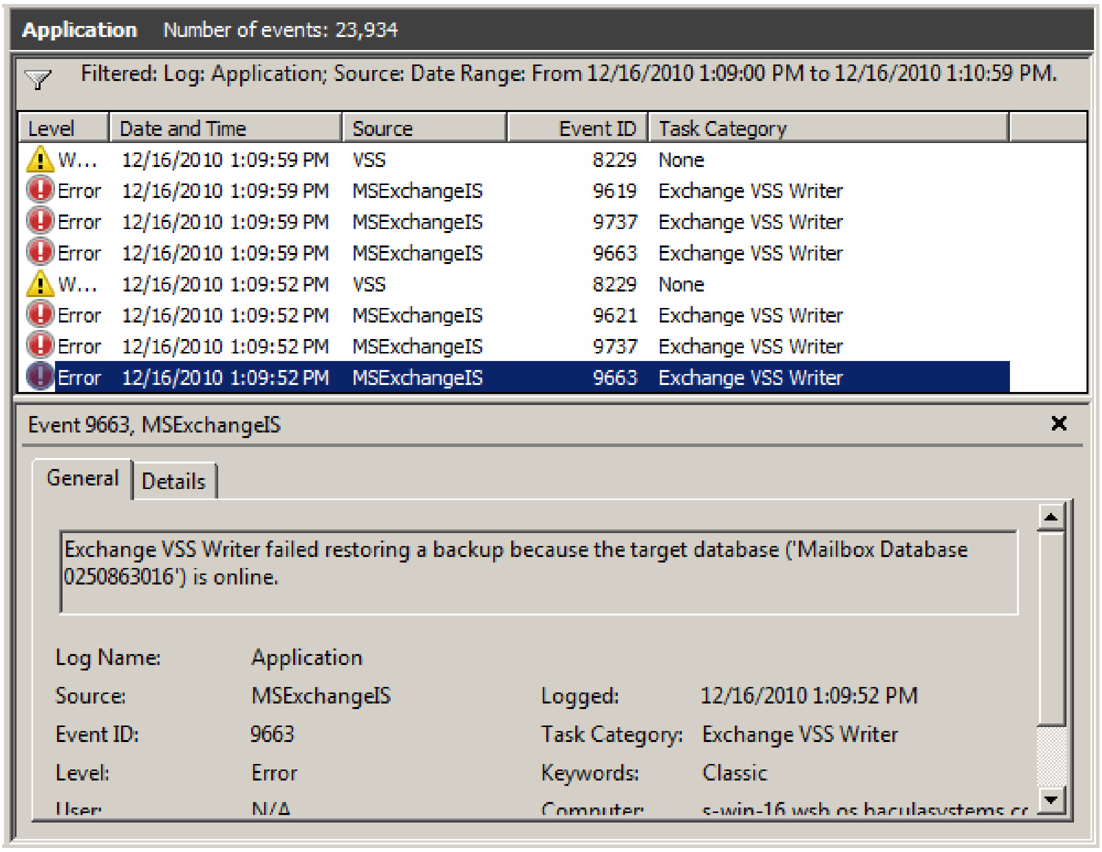
Windows Events

This is a list of Events that you may find in the Windows Application Event log after a restore operation, which is reported as successful by **Bacula Enterprise**. When looking for problem causes, always start with the `earliest` Event related to the current restore attempt (this is most easily done by filtering the Event log by date and time, using the start and end dates of the restore job). We provide an example in figures *Windows' Event Log Filtered, Showing Cause for Failed Restore* and *Job Report is OK, but actual Restore Failed*. In figure *Exchange Message after Database Mount Failure* the corresponding message from ex when trying to mount the database is shown, and figure *Application Log Messages after Database Mount Failure* shows an overview of the corresponding Windows Event Log messages. Note that, in most cases, there will be many failures reported in Windows' Event Log, but normally the first Error is the one you are interested in. With the examples we provide it should be easy to correlate the different reports and drill down to the underlying problem.

Listing 99: Job Report is OK, but actual Restore Failed

```
...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer successfully
 restored the backup set to database ...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer will perform
 database recovery on database ...
...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer failed restoring
 a backup with error code -528 when processing post-restore tasks ...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer failed with error
 code -528 when processing the post-restore event.

If any databases were restored, they are likely in a dirty-shutdown state.
20-Mar 00:31 wsb-exch10-fd JobId 20: A VSS writer has rejected an event with
 error 0x00000000, The operation completed successfully.
. Changes that the writer made to the writer components while handling the
 event will not be available to the requester.
Check the event log for related events from the application hosting the VSS␣
↪writer.
```

(continues on next page)



Application Number of events: 23,934

Filtered: Log: Application; Source: Date Range: From 12/16/2010 1:09:00 PM to 12/16/2010 1:10:59 PM.

Level	Date and Time	Source	Event ID	Task Category
W...	12/16/2010 1:09:59 PM	VSS	8229	None
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9619	Exchange VSS Writer
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9737	Exchange VSS Writer
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9663	Exchange VSS Writer
W...	12/16/2010 1:09:52 PM	VSS	8229	None
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9621	Exchange VSS Writer
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9737	Exchange VSS Writer
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9663	Exchange VSS Writer

Event 9663, MSExchangeIS

Exchange VSS Writer failed restoring a backup because the target database ('Mailbox Database 0250863016') is online.

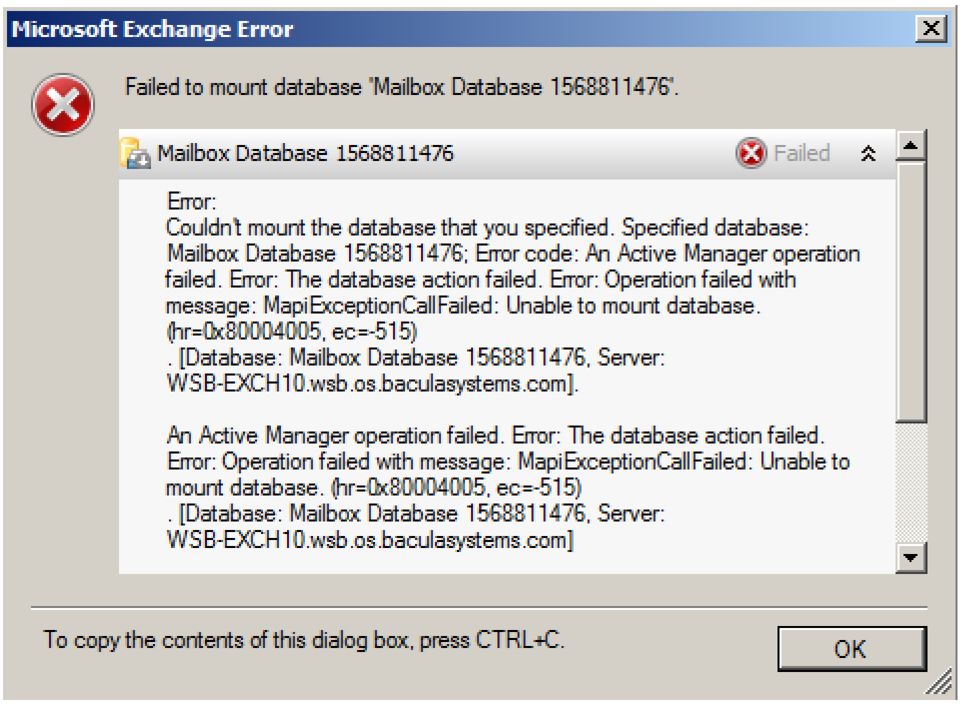
Log Name:	Application		
Source:	MSExchangeIS	Logged:	12/16/2010 1:09:52 PM
Event ID:	9663	Task Category:	Exchange VSS Writer
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	s-win-16.wsb.os.baculasystems.co

Fig. 52: Windows' Event Log Filtered, Showing Cause for Failed Restore

(continued from previous page)

```
...
20-Mar 00:31 s-1-dir JobId 20: Bacula s-1-dir 6.0.0.5 (06Mar12):
  Build OS:               x86_64-unknown-linux-gnu suse 12.1
  JobId:                  20
  Job:                    RestoreFiles.2012-03-20_00.30.23_01
  Restore Client:         wsb-exch10-fd
  Start time:             20-Mar-2012 00:30:25
  End time:               20-Mar-2012 00:31:58
  Files Expected:         17
  Files Restored:         17
  Bytes Restored:         150,022,642
  Rate:                   1613.1 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Note that the actual Job report contains more detailed information!



Failed to mount database 'Mailbox Database 1568811476'.

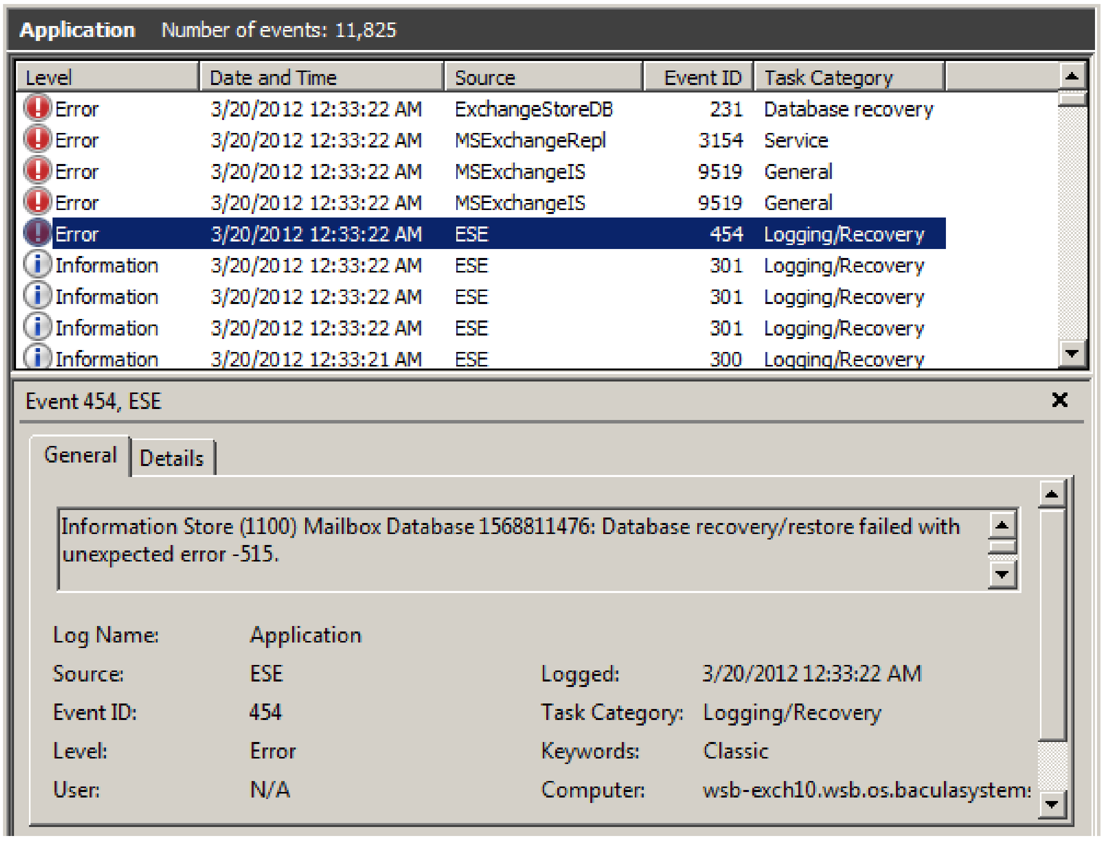
Mailbox Database 1568811476

Error:
Couldn't mount the database that you specified. Specified database: Mailbox Database 1568811476; Error code: An Active Manager operation failed. Error: The database action failed. Error: Operation failed with message: MapiExceptionCallFailed: Unable to mount database. (hr=0x80004005, ec=-515)
. [Database: Mailbox Database 1568811476, Server: WSB-EXCH10.wsb.os.baculasystems.com].

An Active Manager operation failed. Error: The database action failed. Error: Operation failed with message: MapiExceptionCallFailed: Unable to mount database. (hr=0x80004005, ec=-515)
. [Database: Mailbox Database 1568811476, Server: WSB-EXCH10.wsb.os.baculasystems.com]

To copy the contents of this dialog box, press CTRL+C.

Fig. 53: Exchange Message after Database Mount Failure


Application Number of events: 11,825

Level	Date and Time	Source	Event ID	Task Category
Error	3/20/2012 12:33:22 AM	ExchangeStoreDB	231	Database recovery
Error	3/20/2012 12:33:22 AM	MSExchangeRepl	3154	Service
Error	3/20/2012 12:33:22 AM	MSExchangeIS	9519	General
Error	3/20/2012 12:33:22 AM	MSExchangeIS	9519	General
Error	3/20/2012 12:33:22 AM	ESE	454	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:21 AM	ESE	300	Logging/Recovery

Event 454, ESE

General | Details

Information Store (1100) Mailbox Database 1568811476: Database recovery/restore failed with unexpected error -515.

Log Name:	Application		
Source:	ESE	Logged:	3/20/2012 12:33:22 AM
Event ID:	454	Task Category:	Logging/Recovery
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	wsb-exch10.wsb.os.baculasystem:

Fig. 54: Application Log Messages after Database Mount Failure

Table 29: Problem Resolution by Windows Event

Sourc	Ever ID	Task Category	Resolution
MSExchangeRepl	3154	Service	Mount the database through Exchange Management Console
MSExchange	9663	Exchange VSS Writer	Dismount the Database prior to restoring
MSExchange	9751	Exchange VSS Writer	Try the following: Move the `*.env`-file out of the Mailbox Database directory. Run `eseutil /r <Three-char logfile base name>` in the Mailbox Database directory. Re-run the restore as usual. The database should now be restored correctly.
ESE	440	Logging/Re	Remove the existing database log files from the data directory prior to restoring
ESE	518	Logging/Re	see above
ESE	454	Logging/Re	see above

Script to Prepare Recovery Database

```
# $Id: preparebacularestore-E10.ps1 1387 2014-01-26 13:57:57Z arno $
# (C) 2012 Bacula Systems SA
#
# This script is an EXAMPLE of how to prepare Exchange 2010 for an alternate␣
↪location
# restore of data backed up with the vss plugin's @EXCHANGE module
#
# Use at your own risk, make sure to audit before using!
#
# Configuration is done right at the top.
#
# NULL means detect server automatically. Fails if there's more than one
# To hard-code the server to use, try
#  $server = "<nodename>"  with a hard-coded nodename or
#  $server = hostname to use the local hostname as reported by the hostname␣
↪command
$server = $null


if ($server -eq $null) {
  $mbs = @(Get-ExchangeServer | where {$_.IsMailboxServer} | ForEach-Object {
↪$_.Name})
#  $mbs += @(Get-ExchangeServer | where {$_.IsMailboxServer} | ForEach-Object
↪{$_.Name})
```

(continues on next page)

(continued from previous page)

```
  if ($mbs.Count -gt 1) {
        "More than one Mailbox host. Define explicitly, please!"
        Exit
  } else {
        $server = $mbs.Get(0)
  }
}

$me = hostname
if ([string]::Compare($me, $server, $true)) {
  "The Mailbox server we would use is not the local machine... we do not␣
↪support that!"
  Exit
}
"Using server $server"

$mdbs = @(Get-MailboxDatabase | where { $_.Revovery -or $_.AllowFileRestore})

if ($mdbs.Count -gt 0) {
  "The following Mailbox Databases are Recovery Databases or set to allow␣
↪Recovery. This may cause problems, please fix and retry!"
  $mdbs | Format-Table -AutoSize Server,Identity,AllowFileRestore,Recovery
  Exit
}

$tgtdrive = (Get-WMIObject Win32_logicaldisk | Sort-Object -Property␣
↪FreeSpace -Descending).Get(0)

"Using drive $($tgtdrive.DeviceId) for the RDB"
"There are about $("{0:N0}" -f ($tgtdrive.FreeSpace / 1024 / 1024 /1024) )␣
↪GBytes free on it"


do {
  $tgtdir = $tgtdrive.DeviceId + "\" + [System.IO.Path]::GetRandomFileName()
} while (Test-Path $tgtdir)

"$tgtdir is our working directory."

New-Item -Path $tgtdir -Type "directory" | Out-Null

$rdbn = "Bacula EnterpriseRecovery"
$suff = $null

while (($rdb = Get-MailboxDatabase -Identity ($rdbn + $suff) -WarningAction␣
↪SilentlyContinue -ErrorAction SilentlyContinue) -and ($rdb.Count -gt 0)) {
  "MailboxDatabase " + $rdbn + $suff + "exists... trying next one"
  $suff++
}

$rdbn += $suff
```

(continues on next page)

(continued from previous page)

```
New-MailboxDatabase -Recovery -Name $rdbn -Server $server -EdbFilePath (
↪$tgtdir + "\RecoveryDB.edb") -LogFolderPath $tgtdir | Out-Null

Set-MailboxDatabase -Identity $rdbn -AllowFileRestore $true -WarningAction␣
↪SilentlyContinue -ErrorAction SilentlyContinue | Out-Null

"You can Restore with where=" + $rdbn + " now"
```

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Exchange Single Item Restore

Overview

This user's guide presents how to use the Exchange Single Item Restore feature with **Bacula Enterprise** and the VSS Exchange Plugin allowing to interactively restore selected Exchange Mailboxes.

Note: Since 16.0.7, Bacula Enterprise offers the Exchange EWS Plugin which uses the advanced technology from Microsoft to restore single item in a very easy way. Review *this documentation* for further information about it.

Features Summary

The **Bacula Enterprise** Exchange Single Item Restore provides the following main features:

- Console interface
- BWeb Management Suite interface
- Exchange interface

Scope

This document will present solutions for **Bacula Enterprise** 8.4 and later, which are not applicable to prior versions. The Exchange Single Item Restore has been tested and is supported on RHEL 6 and 7, Ubuntu 14.04, Debian 7 and 8, working with Microsoft Exchange Server 2010 and 2013.

Installation

Packages of the Exchange Single Item Restore plugin are available for supported platforms in the "single-item-restore" download area. The package delivered is usually a previous **Bacula Enterprise** version, therefore please search for previous versions in your download area or keep the version installed when upgrading your Storage Daemon. Please contact Bacula Systems to get access to them or if you would have any question.

Download the plugin package to your **Storage Daemon** server and then install using the package manager:

```
# rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the Exchange Single Item Restore plugin and will install dependencies. On RHEL, installing the `perl-JSON` package from **rpmforge** is required.

Note: On RHEL 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/DAG/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0

# cat /etc/yum.repos.d/baculasystems.repo
[singleitemrestore]
name = Bacula Enterprise
baseurl = https://www.baculasystems.com/dl/<id>/rpms/single-item-restore/
↪<version>/rhel6-64
```

(continues on next page)

(continued from previous page)

```
enabled = 1
protect = 0
gpgcheck = 0

[bacula]
name = Bacula Enterprise
baseurl = https://www.baculasystems.com/dl/<id>/rpms/bin/<version>/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0
```

Where <id> is your customer download area identifier and <version> is the current Bacula Enterprise version.

```
# yum install bacula-enterprise-single-item-restore
```

The Exchange Single Item Restore also contains a specific Bacula Enterprise service called "bee-exchange" that must be installed and running on each Exchange server that is used to run the restore process. This installation is done by executing the dedicated installer program `bacula-enterprise-win64ExchangeSingleItemRestore-VERSION.exe`, where `VERSION` represents the product version number like `8.4.0-1` or `8.6.5-1`.

Note: Upgrading the "bee-exchange" service may not work in all cases. **Bacula Systems** recommends to explicitly deinstall an older version before installing a newer one.

The service will start a REST / HTTP daemon that will listen on `localhost:8081` by default. It is possible to configure the service to listen on an external IP address and use HTTPS and / or http authentication. If you decide to keep the default values for the bind address (localhost), you will need to run the internet browser used for the restore operation on the Exchange Server itself. Assuming the Exchange Server to be properly protected against unauthorized access, this provides a reasonably safe default protection against misuse.

Notes about the "bacula" Account on RHEL

All shell commands in this document use the "bacula" Unix account to run.

On RHEL, the Unix "bacula" account is locked by default. This implies that it is not possible to execute a command such as `su - bacula` successfully.

It is possible to unlock the "bacula" account, or to use "sudo -u bacula" to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

This could be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using

```
root@storage# sudo -u bacula /bin/bash
```

Alternatively, unlock the "bacula" unix account and use `su` with a command such as:

```
root@storage# chsh -s /bin/bash bacula
root@storage# su - bacula
bacula@storage# whoami
bacula
```

Samba SMB Shares

The **Bacula Enterprise** Exchange Single Item Restore plugin will use Samba SMB shares automatically to provide the needed data to the Exchange system. It will set up those shares automatically.

To use Samba SMB network shares, installing and configuring the “samba” package is mandatory. To configure the `/etc/samba/smb.conf` file for use with Bacula the script `install-single-item-restore.sh` needs to be run.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

Need to set a password to "bacula" user
New SMB password: *******
Retype new SMB password: ******

Configuration done.

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
wide links = yes
follow symlinks = yes
unix extensions = no
include = /etc/samba/bacula.d/main.conf
```

At this point, it is possible to modify `/etc/samba/smb.conf` to add your own configuration directives. See below for some hints regarding security-related settings.

The network share configurations used by the Exchange Single Item Restore will be stored in the directory `/etc/samba/bacula.d`. Customizing the template used by Bacula to generate configuration files is possible by using a template file.

Depending on the version of the Samba server and its actual configuration, modifications to the template share will be required. In particular, the Exchange Single Item Restore will create symbolic links inside the created share which point to locations outside the shared directory tree. As this can pose a security risk, Samba by default may prevent clients access to such links. To allow these client accesses, which are reasonable as the target location is under control, and the “bacula” user account is assumed to be safe against misuse, a template such as the following should be put in place:

```
[root@storage ~]# cat /etc/samba/bacula.d/custom.tpl
[__share__]
    path = __path__
    browseable = No
    level2 oplocks = No
    oplocks = No
    posix locking = No
```

(continues on next page)

(continued from previous page)

```
    wide links = Yes
    read only = No
    valid users = bacula
    follow symlinks = Yes
```

Note the explicit setting of **follow symlinks = Yes** and **wide links = Yes**. These, along with the global Samba configuration setting of **allow insecure wide links = Yes**, will allow a client to access the symbolically linked data locations that are required by the Exchange Single Item Restore. We suggest to review the impact of those settings in the manual describing the `smb.conf` configuration file.

The Samba server can also be configured to join your AD domain, and it is possible to allow a group of users to mount and use the Bacula network share. However, configuring the Samba server to use an existing Active Directory server is not covered by this document.

The Samba share will be used by the Exchange server to access the database and log files. We advise testing the creation of a Samba share and accessing it from the Exchange server to make sure that everything is configured properly at the network level.

```
bacula@storage# /opt/bacula/bin/smbadd --add --share test1 --path /opt/bacula/
↪bin
bacula@storage# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[test1]"
Loaded services file OK.

# Global parameters
[global]
        workgroup = BACULA
        unix extensions = No
        idmap config * : backend = tdb
        include = /etc/samba/bacula.d/bacula-test1.conf
        wide links = Yes


[test1]
        path = /opt/bacula/bin
        valid users = bacula
        read only = No
        browseable = No

bacula@storage# smbclient //localhost/test1
Enter bacula's password:
Domain=[BACULA] OS=[Windows 6.1] Server=[Samba 4.2.3]
smb: \> ls
  .                                    D        0  Fri May  3 14:15:48 2013
  ..                                   D        0  Fri Jun 12 16:21:39 2015
  bacula                               A     1614  Fri May  3 14:15:46 2013
  bacula-dir                           A  2771768  Fri May  3 14:15:47 2013
  bacula-fd                            A   887111  Fri May  3 14:15:47 2013
  bacula-sd                            A  2169994  Fri May  3 14:15:48 2013
...
```

At this point, it should be possible to mount the network share "test1" from the Exchange server using the "bacula" account. This can be verified using Windows' Explorer. As the created shares are not browseable, they will not appear automatically in an Explorer window showing the Storage Daemon host; instead, the full path to the share needs to be entered into the address bar.

To delete the network share the following command can be used:

```
bacula@storage# /opt/bacula/bin/smbadd --del --share test1
```

Note on SMB Password

The password used in the SMB share should not contain the following characters: &, ' and ".

BWeb Management Suite GUI Notes

To use the BWeb Management Suite graphical GUI with the Exchange Single Item Restore option, it is currently necessary to install and configure BWeb Management Suite on the Storage Daemon where the relevant jobs are stored. If the Director is not on the same machine as the Storage Daemon, it should not be a problem, just remember that the administrator needs to connect to the correct BWeb Management Suite instance to use specific Exchange Single Item Restore screens.

Configuration

Storage Daemon Configuration

On the **Storage Daemon** host server, the `bconsole` program has to be configured properly to allow the "bacula" user to connect to the Director with the configuration file `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 8.4.0 (31 October 2015)
Enter a period to cancel a command.
* version
mydir-dir Version: 8.4.0 (31 October 2015) x86_64-redhat-linux-gnu
* quit
```

The Exchange Single Item Restore package contains a script that enables testing the connection with the Director and testing if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage#  /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you will not see any files.

bee-exchange Service

It is possible to configure the `bee-exchange` service on the Exchange server by creating a configuration file named `bee-exchange.conf` in the **Bacula Enterprise** installation directory on the Exchange server.

```
C:> type c:\Program Files\Bacula\bee-exchange.conf
###################################################################
# Bacula Enterprise Exchange Single Item Restore Configuration File
###################################################################
Port=8082
Hostname=*
SSL
```

The following parameters can be set:

Port
: Specify the TCP/IP port to use. Ex: *Port=8080*

Hostname
: Specify the TCP/IP interface to bind to. Ex: *Hostname=192.168.0.1*

SSL
: Use SSL. Ex: *SSL*

SSLName
: Specify SSL Certificate name to use. Ex: *SSLName=MyCert*

LogFile
: Specify a custom Log file.

DataDir
: Specify a custom Data directory.

Authentication
: Specify a Authentication scheme (Ntlm, Basic, IntegratedWindows, Anonymous). Ex: *Authentication=Anonymous*

The service must be restarted after creation or modification of the configuration file.

The service account requires `Read` and `Write` permissions to all files in the folder with the database and sufficient access rights including the “mailbox import export” role. These permissions can be granted using impersonation as described in https://msdn.microsoft.com/en-us/library/bb204095.aspx and https://technet.microsoft.com/en-us/library/ee633452%28v=exchg.141%29.aspx

For example, if the local “System” account is member of the “Server Management” security group

```
# New-ManagementRoleAssignment -Name "Import Export_Enterprise Support" -
↪SecurityGroup "Server Management" -Role "Mailbox Import Export"
```

Backup

This product uses backups that are created as described in the **Bacula Systems** documentation of the Exchange server functionality of the VSS plugin, which at this time is available as exchange-plugin.

In short, backups need to be done with a line like **Plugin = "vss:/@EXCHANGE/"** in the File Set used.

There are no additional considerations beyond what is detailed in the above mentioned white paper, which also implies that Exchange Single Item Restore can be added to an existing **Bacula Enterprise** infrastructure and can be used with all existing backups.

Restore Scenarios

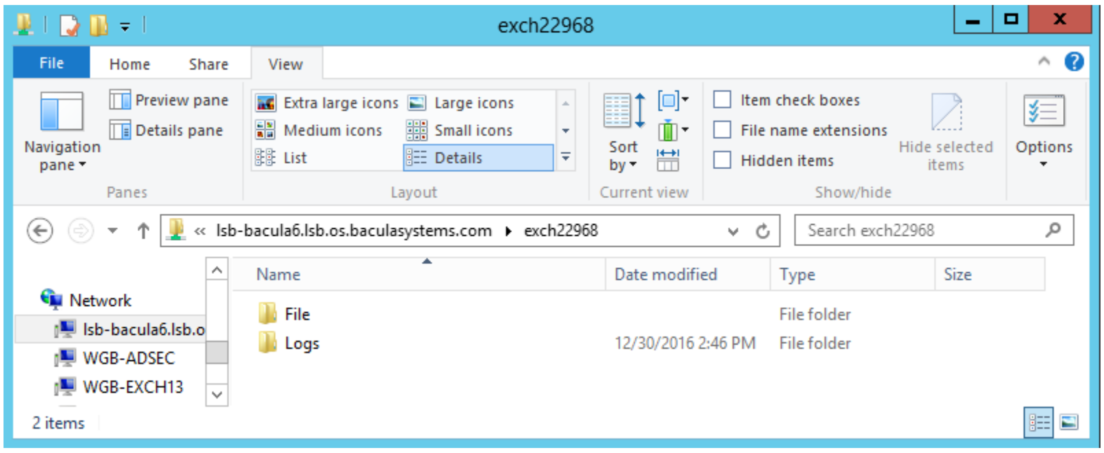
With Text Console Interface

The Exchange Single Item Restore provides a console program that allows you to initialize the Exchange restore process.

```
[root@storage ~]# su - bacula -c '/opt/bacula/bin/mount-exchange'
Automatically Selected Catalog: MyCatalog

Client list:
1: bacula6-fd
2: wgb-exch13-fd
Select a Client: 2
Selected wgb-exch13-fd

Job list:
1: wgb-exch13.2016-12-20_13.09.55_18
2: wgb-exch13.2016-12-20_14.28.11_20
3: wgb-exch13.2016-12-20_14.28.53_21
4: wgb-exch13.2016-12-21_17.37.15_31
5: wgb-exch13.2016-12-21_17.42.36_32
6: wgb-exch13.2016-12-21_23.45.18_35
7: wgb-exch13.2016-12-27_13.09.24_06
Select a Job: 7
Selected wgb-exch13.2016-12-27_13.09.24_06
Automatically Selected Exchange Server: WGB-EXCH13

Exchange Database list:
1: MailboxDBA
2: MailboxDBStandalone
Select an Exchange Database: 1
Selected MailboxDBA

I: From the Exchange server web browser, you need to access the bee-echange
    service that uses the default URL: http://localhost:8081/?Share=exch22968

I: The Network share name is "exch22968"

I: Once the restore is done and the database is dismounted,
   press enter to finish and cleanup the session.
```

At this step, the Exchange data files are available locally on the Storage Daemon server, and a samba network share is configured.

For a verification of basic functionality, the Windows' Explorer can be used (remember that, as the share should not be browseable, the complete path needs to be entered in the address line). This is shown in figure *Verifying Access to a Created SMB Share*. Note the "File" folder without time stamp – this is represented by a symbolic link, and unless it can be opened in the Explorer window without problems, security restrictions are set too tightly in the `/etc/samba/smb.conf` file and the dynamically created share. In that case, reviewing and modifying those settings will be necessary.



Name	Date modified	Type	Size
File		File folder	
Logs	12/30/2016 2:46 PM	File folder	

Fig. 55: Verifying Access to a Created SMB Share

Note that this access verification should only be useful during deployment and initial configuration of the Exchange Single Item Restore. Once things work out without further manual interaction, it is possible to directly use the actual restore functionality, described below.

To actually mount and browse the Exchange database, connect to the Exchange server and start a browser with the URL printed above.

The Bacula Enterprise Exchange service will handle all steps necessary to restore a mailbox (see chapter *Exchange Single Item Restore Screens*) through a Graphical User Interface.

To cleanup the restore session, just press "Enter" in the terminal session where the `mount-exchange` script was started.

With BWeb Management Suite Interface

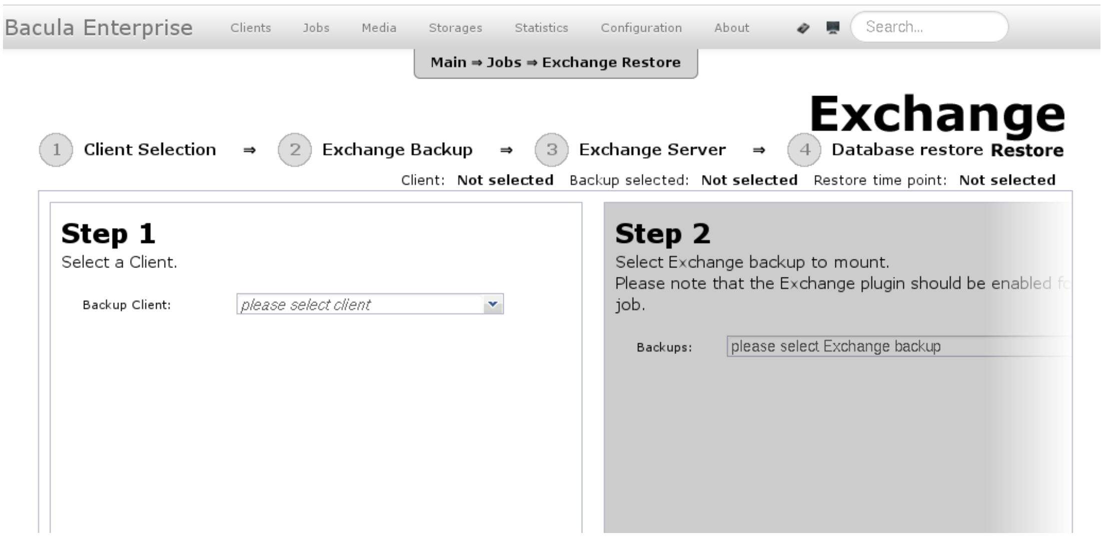
The Exchange Single Item Restore option in BWeb Management Suite is a wizard allowing easy restoration of items from an Exchange database.

The first step is to select the Client where the VSS backup job was done (see fig. *Client Selection*).

Once the Client is selected, the administrator needs to select the Job (a Restore Point) to restore. (Fig. *Restore Point Selection*).

If the selected Job is a valid VSS backup job, the third step will display a list of all Exchange databases included. (Fig. *Exchange Database Selection*).

At this point, Bacula will create a network share with Samba, and the administrator needs to connect to the Exchange server and open the URL displayed in the wizard with a web browser. The Exchange Server address field is set to the Bacula Client address (see fig. *Connection to Exchange Single Item Restore Service*) automatically and can be changed if desired. Note that by default, the Exchange Single


Bacula Enterprise
Clients
Jobs
Media
Storages
Statistics
Configuration
About
Search...
Main ⇒ Jobs ⇒ Exchange Restore
Exchange Restore
1 Client Selection ⇒ 2 Exchange Backup ⇒ 3 Exchange Server ⇒ 4 Database restore

Client: **Not selected** Backup selected: **Not selected** Restore time point: **Not selected**

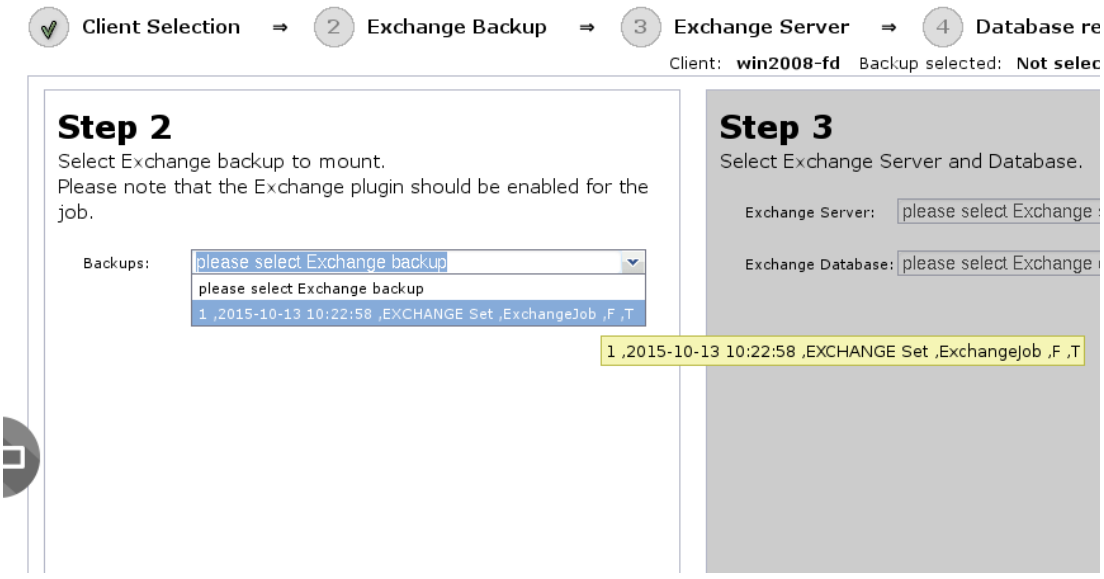
Step 1
Select a Client.

Backup Client: *please select client*

Step 2
Select Exchange backup to mount.
Please note that the Exchange plugin should be enabled f
job.

Backups: please select Exchange backup

Fig. 56: Client Selection


Client Selection ⇒ 2 Exchange Backup ⇒ 3 Exchange Server ⇒ 4 Database re

Client: **win2008-fd** Backup selected: **Not selec**

Step 2
Select Exchange backup to mount.
Please note that the Exchange plugin should be enabled for the job.

Backups: please select Exchange backup
please select Exchange backup
1 ,2015-10-13 10:22:58 ,EXCHANGE Set ,ExchangeJob ,F ,T

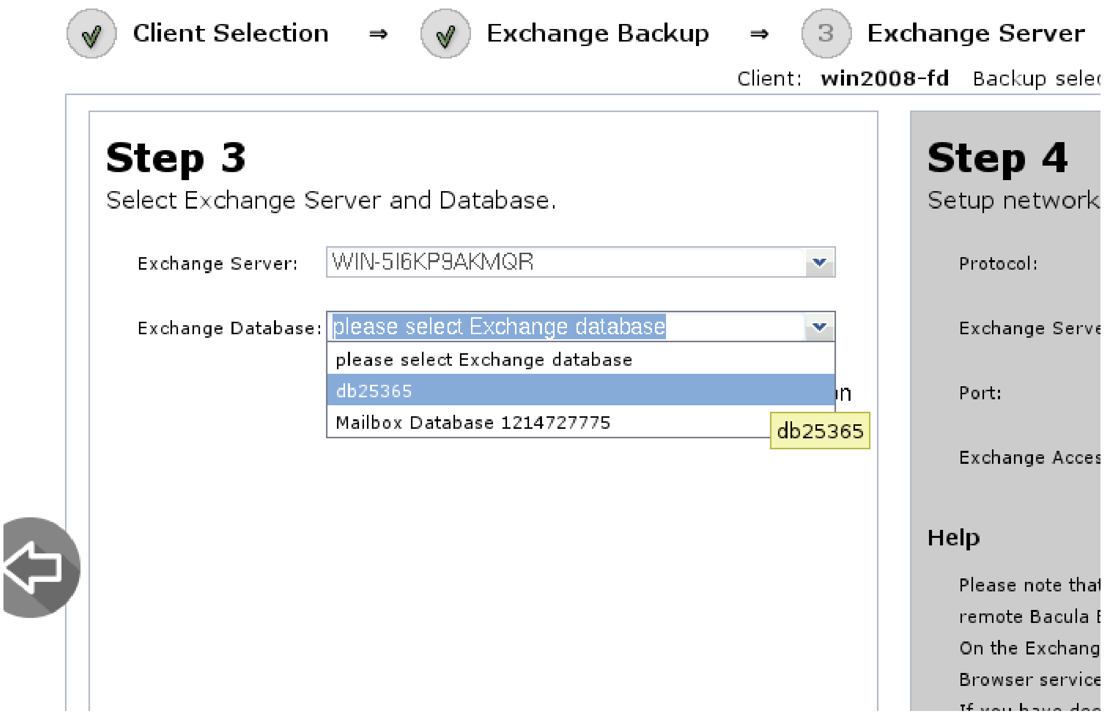
Step 3
Select Exchange Server and Database.

Exchange Server: please select Exchange

Exchange Database: please select Exchange

1 ,2015-10-13 10:22:58 ,EXCHANGE Set ,ExchangeJob ,F ,T

Fig. 57: Restore Point Selection


Client Selection ⇒ Exchange Backup ⇒ 3 Exchange Server

Client: win2008-fd Backup selec

Step 3

Select Exchange Server and Database.

Exchange Server: WIN-5I6KP9AKMQR

Exchange Database: please select Exchange database

please select Exchange database
db25365
Mailbox Database 1214727775

db25365

Step 4

Setup network

Protocol:

Exchange Serve

Port:

Exchange Acces

Help

Please note that
remote Bacula E
On the Exchang
Browser service

Fig. 58: Exchange Database Selection

Item Restore windows service does **not** accept remote connections, so accessing it requires using a web browser started **on the Exchange server** console. (See chapter *Exchange Single Item Restore Screens*)

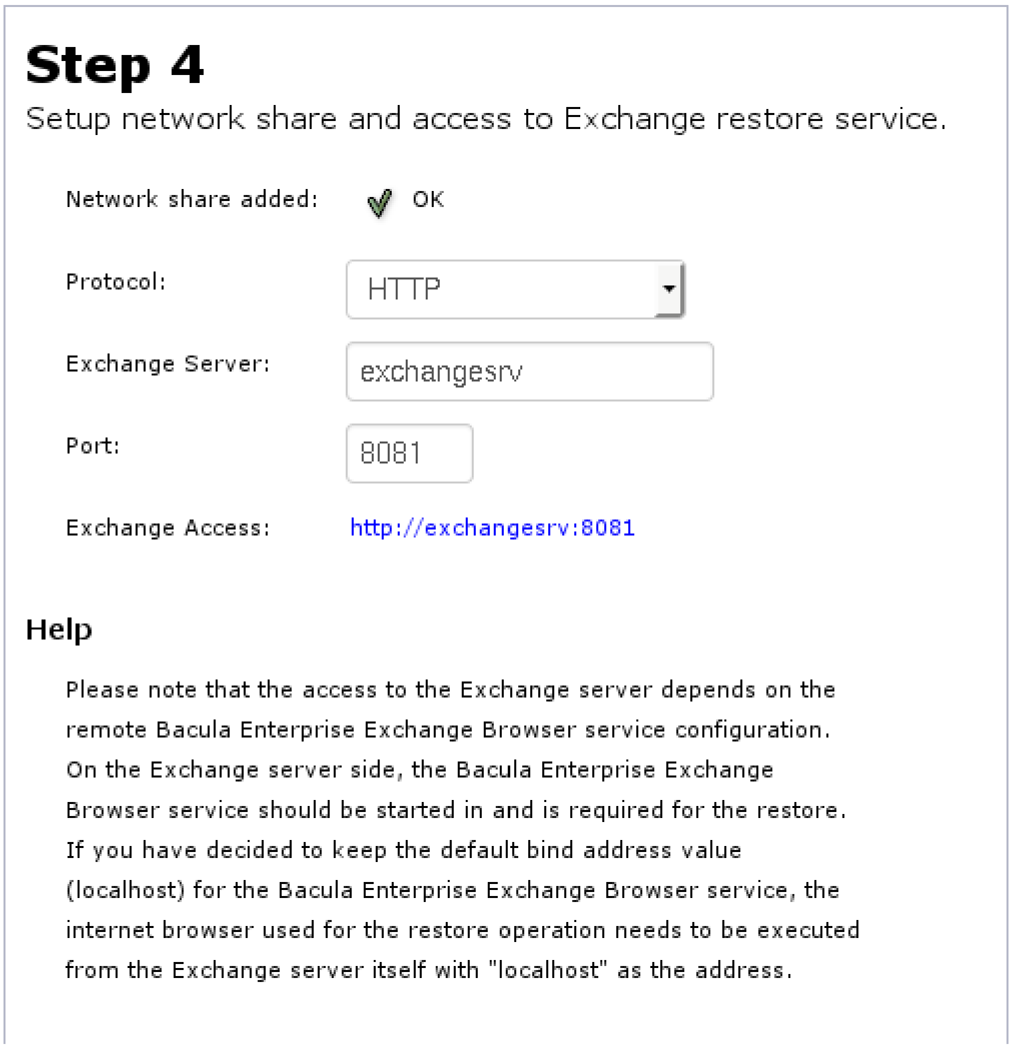
Once the restore is actually done, it is important to terminate the restore session to release resources (fig. *Terminate the Restore Session*).

Exchange Single Item Restore Screens

The "Restore Single Mailbox from Exchange Database" page should be displayed as shown in figure *Bacula Exchange Single Item Restore GUI*. At this point, operations will take place directly on the Exchange server and will accomplish the following operations:

1. Mount the network share
2. Check the Exchange database status
3. Perform recovery if needed
4. Create a Recovery database
5. Mount the Recovery database
6. Browse and restore mailboxes
7. Dismount and cleanup

In order to mount the network share where the Exchange database is available, "Bacula Share Parameters" must be specified. Then press the "Start Recovery" button. All steps will then be executed one after the other (fig. *Recovery Database Setup*). Information about a particular step is available by clicking on "Details" (fig. *Details of an Operation*).



Step 4

Setup network share and access to Exchange restore service.

Network share added:	✓ OK
Protocol:	HTTP
Exchange Server:	exchangesrv
Port:	8081
Exchange Access:	http://exchangesrv:8081

Help

Please note that the access to the Exchange server depends on the remote Bacula Enterprise Exchange Browser service configuration. On the Exchange server side, the Bacula Enterprise Exchange Browser service should be started in and is required for the restore. If you have decided to keep the default bind address value (localhost) for the Bacula Enterprise Exchange Browser service, the internet browser used for the restore operation needs to be executed from the Exchange server itself with "localhost" as the address.

Fig. 59: Connection to Exchange Single Item Restore Service

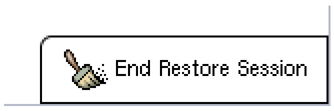
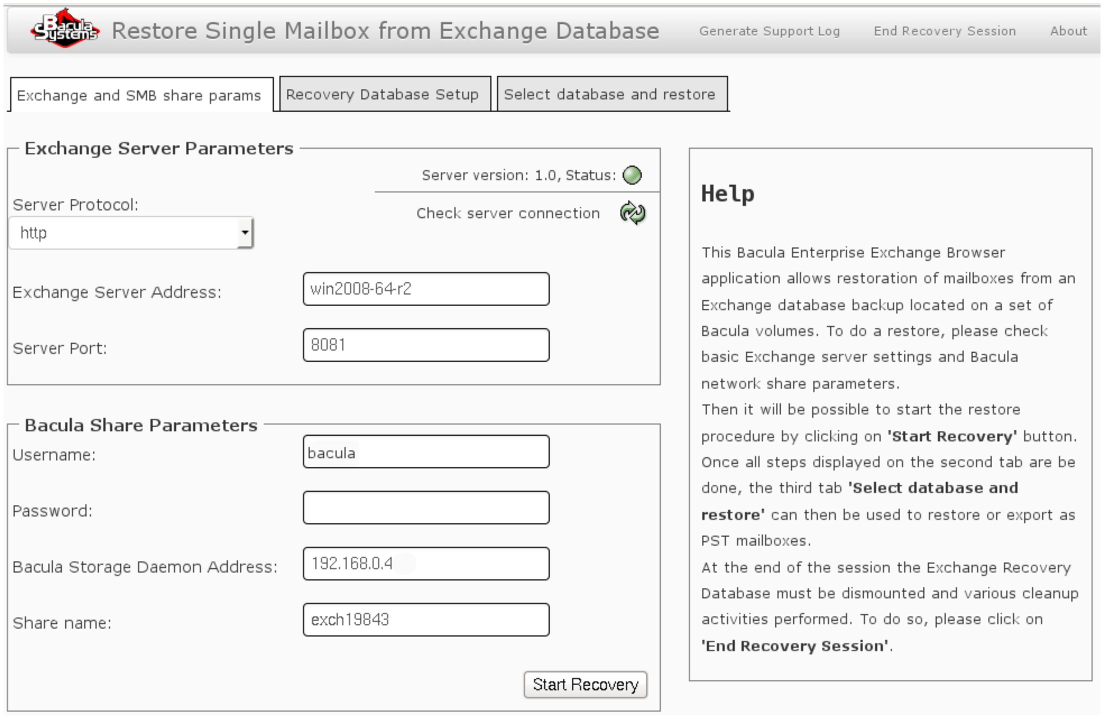


Fig. 60: Terminate the Restore Session



Restore Single Mailbox from Exchange Database
Generate Support Log
End Recovery Session
About
Exchange and SMB share params
Recovery Database Setup
Select database and restore

Exchange Server Parameters

Server version: 1.0, Status: ●

Check server connection

Server Protocol:	http
Exchange Server Address:	win2008-64-r2
Server Port:	8081

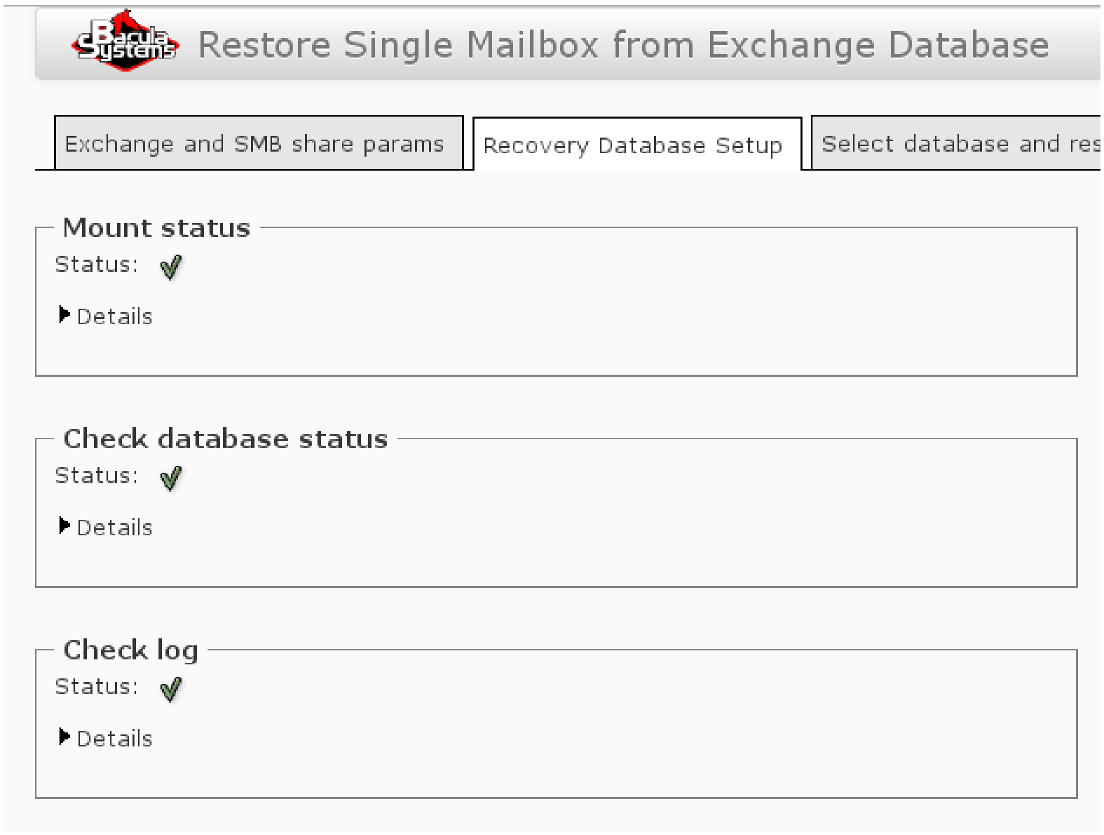
Bacula Share Parameters

Username:	bacula
Password:	
Bacula Storage Daemon Address:	192.168.0.4
Share name:	exch19843

Help

This Bacula Enterprise Exchange Browser application allows restoration of mailboxes from an Exchange database backup located on a set of Bacula volumes. To do a restore, please check basic Exchange server settings and Bacula network share parameters.
Then it will be possible to start the restore procedure by clicking on **'Start Recovery'** button. Once all steps displayed on the second tab are be done, the third tab **'Select database and restore'** can then be used to restore or export as PST mailboxes.
At the end of the session the Exchange Recovery Database must be dismounted and various cleanup activities performed. To do so, please click on **'End Recovery Session'**.

Fig. 61: Bacula Exchange Single Item Restore GUI



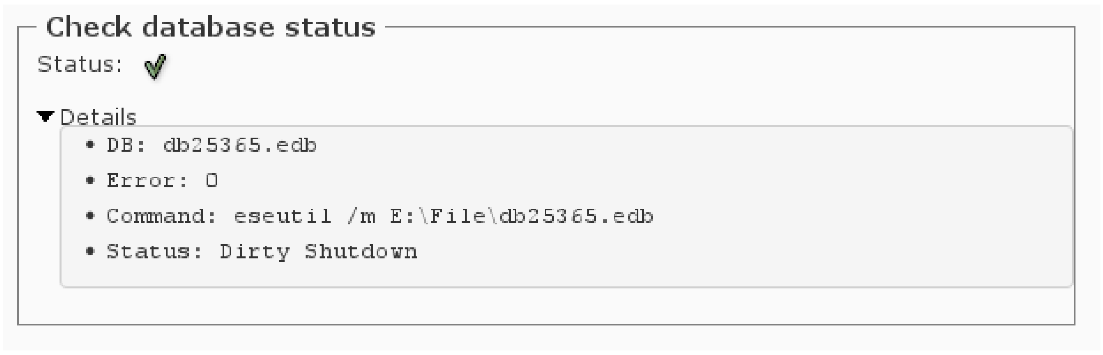
Restore Single Mailbox from Exchange Database
Exchange and SMB share params
Recovery Database Setup
Select database and res

Mount status
Status: ✓
Details

Check database status
Status: ✓
Details

Check log
Status: ✓
Details

Fig. 62: Recovery Database Setup



Check database status

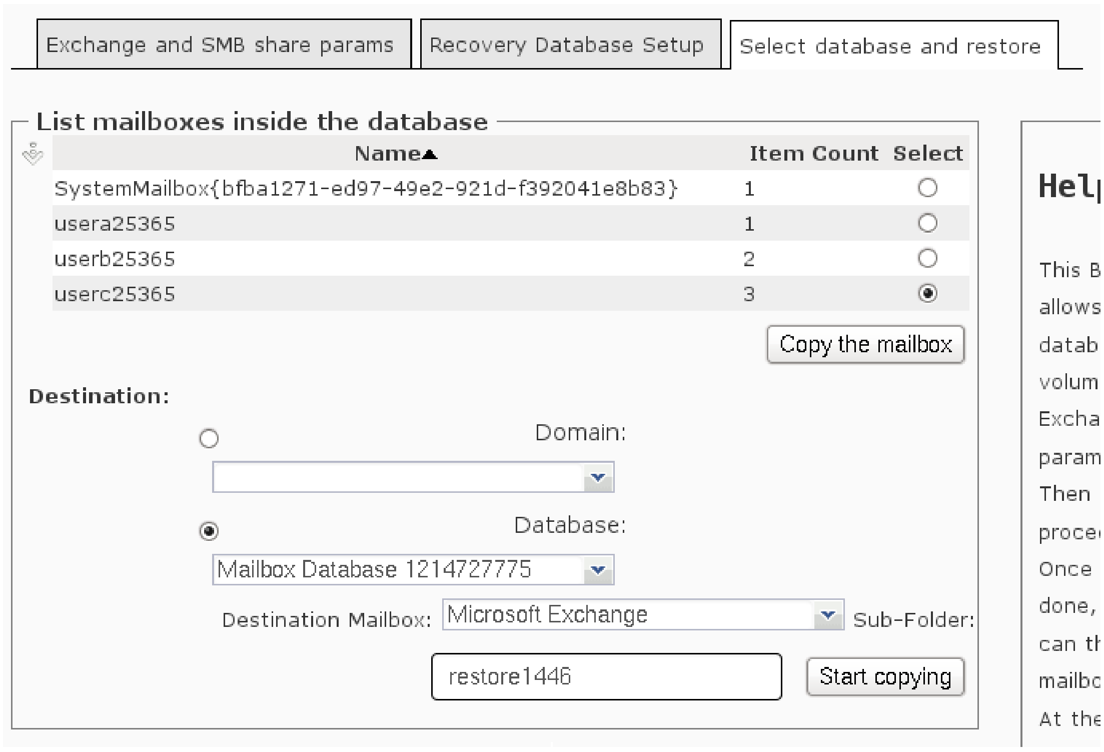
Status: ✓

Details

- DB: db25365.edb
- Error: 0
- Command: eseutil /m E:\File\db25365.edb
- Status: Dirty Shutdown

Fig. 63: Details of an Operation

Once in the last tab, the recovery database will be mounted and available to the Exchange console and the Powershell interface. The list of all mailboxes will be displayed like shown in figure *Mailbox List*. It is possible to select a Mailbox and restore it to another database.



List mailboxes inside the database

Name▲	Item Count	Select
SystemMailbox{bfba1271-ed97-49e2-921d-f392041e8b83}	1	○
usera25365	1	○
userb25365	2	○
userc25365	3	●

Destination:

○ Domain:

● Database: Mailbox Database 1214727775

Destination Mailbox: Microsoft Exchange Sub-Folder: restore1446

Fig. 64: Mailbox List

Once the restore is actually done, it is important to terminate the restore session to release resources (fig. *Dismount Recovery Database*) on the Exchange server, and then terminate the restore session in the "mount-exchange" terminal session or in the BWeb page.

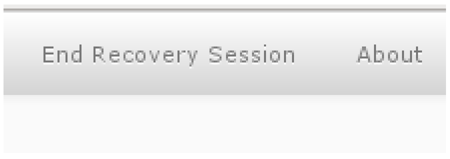


Fig. 65: Dismount Recovery Database

Notes

Cache Directory

To speed up future Exchange Single Item Restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
storage# ls /opt/bacula/working/mount-cache
MyCatalog-2.idx  MyCatalog-5.idx  MyCatalog-8.idx
MyCatalog-4.idx  MyCatalog-6.idx  MyCatalog-9.idx
```

It is possible to remove files in the cache after some time. They are re-generated if needed.

Limitations

- The PST export is not yet implemented in the Bacula Exchange Single Item Restore service.
- The Bacula Exchange Single Item Restore Service is not multi-threaded and can serve only one request at a time.
- The Exchange Single Item Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backends due to internal MySQL limitations with indexes on TEXT columns. For VMWare and Exchange Single Item Restore, it should not impact performance too much (the backup structure is usually quite small) but **Bacula Systems** advises using the PostgreSQL catalog backend for the best experience.
- The Exchange Single Item Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was written with a large number of concurrent jobs.
- The Exchange Single Item Restore is not compatible with FileDaemon data encryption.
- The password used in the network share between the Bacula Storage Daemon and the Exchange server should not use : &, ' and "
- The Exchange Single Item Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.

LDAP/MSAD Plugin

Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. This whitepaper presents solutions for object level backup and restore of Directory servers including Active Directory using the LDAP/MSAD Plugin with Bacula Enterprise version 10.

Overview

This document is intended to provide insight into the considerations and processes required to implement LDAP and MSAD Backup and Restore using **Bacula Enterprise**.

LDAP and MSAD

The LDAP Plugin was designed to perform a backup and restore of a single LDAP object. It uses the LDAP network protocol and the standard schema to search and fetch objects, so it should support a variety of different LDAP servers in addition to the OpenLDAP server which we have used for testing (for LDAP servers other than OpenLDAP, the plugin should be tested before using it in production).

The MSAD Plugin was designed to perform a backup and restore of a single MS Active Directory object. It uses the LDAP network protocol and Active Directory schema to search and restore objects. However, please be aware that even though it backs up Microsoft AD the plugin currently runs only on Linux and Unix machines. The MS Active Directory Plugin required the queries and code to be rewritten to correspond to the different Microsoft fields and structures. The MSAD Plugin was built and is run under Linux using a network connection to the Microsoft Active Directory server.

Features Summary

The **Bacula Enterprise** Directory Server Plugin provides the following main features for LDAP servers:

- support for backup levels: Full, Differential, Incremental
- support for Accurate mode (finds deleted objects)
- object size and modification time is properly saved in Bacula catalog
- optional object relocation during restore
- LDIF-like internal archive format
- supports replace options: always, never, ifnewer, ifolder
- connects to LDAP server using the LDAP network protocol
- support for the OpenLDAP server
- support for ldaps (SSL) communication with LDAP server
- support listing mode for browsing objects directly on Bacula console
- built and tested on Linux

In addition the MSAD Plugin provides the following additional features for Active Directory servers:

- support for Microsoft Active Directory server (from Windows 2003 up to the latest and future Windows versions)
- connects to MS Active Directory server using the LDAP network protocol
- allow automatic MS Active Directory objects tombstone recovery
- special Active Directory attributes handling

Scope

This paper presents solutions for **Bacula Enterprise** versions 10 and later, which are not applicable to prior versions.

Installation

The Bacula File Daemon and Directory Server Plugin can be installed on any Linux host which has access to the LDAP or MSAD server to be protected. Installation of the MSAD Plugin on a Windows Server is not currently supported.

Configuration

The **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `ldap-fd.so` or `msad-fd.so` plugin files are installed. The standard Bacula plugin directory is `/opt/bacula/plugins`

```
FileDaemon {
    Name = bacula-fd
    Plugin Directory = /opt/bacula/plugins
```

(continues on next page)

(continued from previous page)

```
...
}
```

Installation of the Plugin

Installation of the Bacula Enterprise Directory Plugin is most easily done by adding the repository file suitable for the existing subscription to the Linux package manager for your distribution of choice. For RHEL distributions an example would be `/etc/yum.respos.d/bee.repo` with the following content:

```
[bacula-enterprise]
name=Bacula Enterprise for RHEL $releasever - $basearch
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/10.0.1/
↪rhel7-64
enabled=1
gpgcheck=1
gpgkey=https://www.baculasystems.com/dl/@customer-string@/BaculaSystems-
↪Public-Signature.asc

[bacula-enterprise-ldap]
name=Bacula Enterprise LDAP for RHEL $releasever - $basearch
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/ldap/10.0.1/
↪rhel7-64
enabled=1
gpgcheck=1
gpgkey=https://www.baculasystems.com/dl/@customer-string@/BaculaSystems-
↪Public-Signature.asc
```

For Ubuntu/Debian distributions an example would be `/etc/apt/sources.list.d/bacula.list` with the following content:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪stretch-64/ stretch main
#Bacula Enterprise LDAP
deb https://www.baculasystems.com/dl/@customer-string@/debs/ldap/@version@/
↪stretch-64/ stretch ldap
```

After that, a run of `apt-get update` for any Ubuntu/Debian distribution is needed. Then, the Plugin can be installed using

`apt-get install bacula-enterprise-ldap-plugin` or

`yum install bacula-enterprise-ldap-plugin` at RHEL.

Manual installation of the packages, can be done after downloading the package files from the Bacula Systems provided download area, and then using the package manager to install.

Plugin Configuration

The plugin is configured using dedicated config files and/or **Plugin Parameters** defined in a Fileset **Include** section of the Bacula Enterprise Director configuration.

Note, even though you may be using the MSAD plugin to backup a Windows Active Directory, your File daemon will be a Linux or Unix machine. This limitation could change in the future.

Plugin Config file

The LDAP and MSAD plugins require a configuration file on the File Daemon machine. This configuration file contains parameters for LDAP or MSAD server connection. Each plugin uses its own configuration file.

The default configuration file is located at:

`/opt/bacula/etc/ldap.conf` for the LDAP plugin and

`/opt/bacula/etc/msad.conf` for the MSAD plugin.

You may specify a different config file (using `config=...` plugin parameter) in each Fileset definition to point your backup to different Directory servers.

You can see an example config file for LDAP Plugin:

```
#
# Sample config file for the ldap plugin /opt/bacula/etc/ldap.conf
#
LDAPURI = "ldap://192.168.0.100/"
BINDDN = "cn=backup,dc=acme,dc=com"
BINDPASS = "PASSWORD"
BASEDN = "dc=acme,dc=com"
```

You can see an example config file for MSAD Plugin:

```
#
# Sample config file for the msad plugin
#
LDAPURI = "ldap://10.0.101.1/"
BINDDN = "cn=Administrator,cn=Users,dc=domain,dc=com"
BINDPASS = "password"
BASEDN = "dc=domain,dc=com"
```

The definitions and default values for Plugin config file parameters are described at Table *LDAP plugin parameters*. The config file supports the following parameters: `LDAPURI`, `BINDDN`, `BINDPASS`, `BASEDN`, `TLS_CACERT`, `TLS_CERFILE`, `TLS_CACERTDIR`, `TLS_REQCERT`. All other plugin parameters are not supported in the config file.

Plugin Parameters

The following plugin parameters can be used to configure connection, backup or restore jobs with a plugin.

Table 30: LDAP plugin parameters

Options	Default	Description
attribs		It is a comma separated list of additional attributes which plugin should skip during restore.
config	LDAP: `/opt/bacula/etc/ldap.conf` Active Directory: `/opt/bacula/etc/msad.conf`	The LDAP Plugin configuration file.
ldapuri	ldap://localhost/	The LDAP URI parameter specifies how to connect to the ldap server. The ldap, ldaps and ldapi schemes are supported.
binddn	cn=admin,dc=example,dc=com	Backup user distinguish name.
bindpass	secret	Backup user password.
hbindpass		Backup user password obscured.
basedn	dc=example,dc=com	A base location (DN) for backup, it could be a LDAP server's root tree or some other subtree.
TLS_CACERT	•	Path to a file on disk with the TLS Certificate
TLS_CACERTDIR	•	Path to a directory on disk with the TLS Certificates
TLS_REQCERT	demand	never, allow, try, demand
schemaapply	Yes	A boolean option (yes/no) to toggle an automatic skipping of well known read-only attributes of the Active Directory server (msad-fd plugin only).

Using the **config=...** plugin parameter, a prepared configuration file providing the plugin parameters can be used instead of the plugin command line.

Obscure the LDAP Password

As of the 8.0.3 version of the LDAP plugin, it is possible to obscure the password. The obscured password field is called **hpassword**. The bconsole `@encode` command can be used to generate the obscured password. Note that if the string to be encoded contains "=", the `string=` keyword with its parameter value in quotes has to be used:

```
# /opt/bacula/bin/bconsole
* @encode apassword
```

(continues on next page)

(continued from previous page)

```
MTEyOjEyNzoGAwAYFQIVABEDAwcfAhQA

* @encode string="passwordwith="
NTMwOjU0Mzpic2FhZX1gdmV7ZnovAA
```

You can see an example config file for LDAP Plugin with user password obscured:

```
# cat /opt/bacula/etc/ldap.conf
#
# Sample config file for the ldap plugin /opt/bacula/etc/ldap.conf
#
LDAPURI = "ldap://192.168.0.100/"
BINDDN = "cn=backup,dc=acme,dc=com"
HBINDPASS = "NTMwOjU0Mzpic2FhZX1gdmV7ZnovAA"
BASEDN = "dc=acme,dc=com"
```

Fileset Examples

LDAP

In the example below, the plugin will use a default config file to make a backup:

```
Fileset {
    Name = FS_LDAPDefault
    Include {
        Plugin = "ldap:"
    }
}
```

In this example, the plugin will use a selected config file to make a backup:

```
Fileset {
    Name = FS_LDAPConfig
    Include {
        Plugin = "ldap: config=/opt/bacula/etc/openldap-server.conf"
    }
}
```

In the following example all required config parameters are provided at the Plugin=... command line. No config file is used:

```
Fileset {
    Name = FS_LDAPAll
    Include {
        Plugin = "ldap: ldapuri=ldap://192.168.0.200/ binddn=DOM/
↪Administrator \
            bindpass=secret basedn=dc=dom,dc=com"
    }
}
```

MSAD

In the example below, the plugin will use a default config file to make a backup:

```
Fileset {
    Name = FS_MSADDefault
    Include {
        Plugin = "msad:"
    }
}
```

In this example, the plugin will use a selected config file to make a backup:

```
Fileset {
    Name = FS_MSADConfig
    Include {
        Plugin = "msad: config=/opt/bacula/etc/msad-server.conf"
    }
}
```

In the following example all required config parameters are provided at the Plugin=... command line. No config file is used:

```
Fileset {
    Name = FS_MSADAll
    Include {
        Plugin = "msad: ldapuri=ldap://10.0.101.1/ binddn=cn=Administrator,
↪cn=Users,dc=domain,dc=com \
            bindpass=password basedn=dc=domain,dc=com"
    }
}
```

Preparation

You may specify a different config file in each Fileset definition. Note that the configuration file is loaded by the plugin, not the Director, thus it needs to be located on the host running the File Daemon. The plugin requires a LDAP/MSAD account with permissions to query and read objects for backup. This account can be an admin account or standard account with a Backup Operator role. LDAP/MSAD plugin creates a virtual namespace in Bacula catalog which consists of "ldap:" or "msad:" prefixes and the DIT is represented as a directory tree. The Bacula virtual namespace does not contain the ldap/msad server/instance name so the backup admin must distinguish the same tree between different servers on the same Bacula FD client.

Testing the Connection

To test the connection parameters for your LDAP or MSAD server, you can use the following command:

```
# /opt/bacula/bin/ldaptest
Usage: ./ldaptest <uri> <binddn> <bindpass>
```

where:

uri
: is the value set in `LDAPURI` config parameter

binddn
: is the value set in `BINDDN` config parameter

bindpass
: is the value set in `BINDPASS` config parameter

You can find an example of `ldaptest` utility execution below which display ldap debug information during connection initiation and server Root DSE information on successful connection.

```
# /opt/bacula/bin/ldaptest ldap://192.168.0.200/ 'DOM\Administrator' password
ldap_sasl_bind_s
ldap_sasl_bind
ldap_send_initial_request
ldap_new_connection 1 1 0
ldap_int_open_connection
ldap_connect_to_host: TCP 192.168.0.200:389
ldap_new_socket: 3
ldap_prepare_socket: 3
ldap_connect_to_host: Trying 192.168.0.200:389
ldap_pvt_connect: fd: 3 tm: -1 async: 0
attempting to connect:
connect success
ldap_open_defconn: successful
ldap_send_server_request
ldap_result ld 0x55cbc7b57630 msgid 1
wait4msg ld 0x55cbc7b57630 msgid 1 (infinite timeout)
wait4msg continue ld 0x55cbc7b57630 msgid 1 all 1
** ld 0x55cbc7b57630 Connections:
* host: 192.168.0.200  port: 389  (default)
refcnt: 2  status: Connected
last used: Mon May 21 08:48:59 2018
(...)
LDAP Server connection OK

RootDSE:
    currentTime: 20180521154906.0Z
    subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=dom,DC=com
    namingContexts: DC=dom,DC=com
    namingContexts: CN=Configuration,DC=dom,DC=com
    namingContexts: CN=Schema,CN=Configuration,DC=dom,DC=com
    namingContexts: DC=DomainDnsZones,DC=dom,DC=com
    namingContexts: DC=ForestDnsZones,DC=dom,DC=com
    defaultNamingContext: DC=dom,DC=com
```

(continues on next page)

(continued from previous page)

```
    schemaNamingContext: CN=Schema,CN=Configuration,DC=dom,DC=com
    configurationNamingContext: CN=Configuration,DC=dom,DC=com
    rootDomainNamingContext: DC=dom,DC=com
(...)
Test successful!
```

To find a BINDDN for your user for Active Directory you can use a PowerShell command below:

```
PS> dsquery user -name Administrator
"CN=Administrator,CN=Users,DC=dom,DC=com"
```

Backup

The plugin performs a single base query to find all objects to backup. The plugin will use a paged control response to get all available objects. It saves all standard and extended attributes returned from server. This includes any system or dynamic attributes.

When the plugin detects a referral object during backup, it will not descend to it and will log a message, i. e.:

```
(...) Referal found. Will not descend to ref: ldap://.../...
```

The backup will create a single file for every object found in Directory Server subtree started from `BASEDN` parameter.

Testing your Fileset

The `estimate` command can be used to test your `Fileset`, especially with the `listing` parameter.

```
* estimate listing job=pluginTest level=Full
Using Catalog "MyCatalog"
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
drwxr-xr-x  1 root  root   585 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/
-rw-r--r--  1 root  root   542 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/
↪cn=root
-rw-r--r--  1 root  root   535 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/
↪cn=test
2000 OK estimate files=3 bytes=1,077
```

Restore

The plugin supports restore operations to a working LDAP or MSAD server only. You cannot restore objects to the local filesystem. You can however restore objects to their original location or relocate them to different parts of the subtree.

To restore an object to its original location you have to use a **where=/** restore parameter.

The Plugin is not designed for Disaster Recovery procedures where the LDAP or MSAD servers may not be functional. For Disaster Recovery of an MS AD servers you can use the Bacula Enterprise VSS plugin.

LDAP objects are restored like regular files with the Bacula “restore” command.

```
cwd is: /
$ cd ldap:/dc=com/dc=bacula
cwd is: ldap:/dc=com/dc=bacula/
$ dir
-rw-r--r-- 1 root  root  568    ldap:/dc=com/dc=bacula/cn=admin
drwxr-xr-x 1 root  root  480    ldap:/dc=com/dc=bacula/ou=Accounting/
drwxr-xr-x 1 root  root  491    ldap:/dc=com/dc=bacula/ou=Administrative/
drwxr-xr-x 1 root  root  494    ldap:/dc=com/dc=bacula/ou=Human Resources/
drwxr-xr-x 1 root  root  479    ldap:/dc=com/dc=bacula/ou=Janitorial/
drwxr-xr-x 1 root  root  479    ldap:/dc=com/dc=bacula/ou=Management/
drwxr-xr-x 1 root  root  470    ldap:/dc=com/dc=bacula/ou=Payroll/
drwxr-xr-x 1 root  root  464    ldap:/dc=com/dc=bacula/ou=Peons/
drwxr-xr-x 1 root  root  506    ldap:/dc=com/dc=bacula/ou=Product Development/
drwxr-xr-x 1 root  root  494    ldap:/dc=com/dc=bacula/ou=Product Testing/
drwxr-xr-x 1 root  root  450    ldap:/dc=com/dc=bacula/ou=groups/
drwxr-xr-x 1 root  root  448    ldap:/dc=com/dc=bacula/ou=hosts/
drwxr-xr-x 1 root  root  450    ldap:/dc=com/dc=bacula/ou=people/
$ add "ou=Product Testing"
121 files marked.
$ cd ou=people
cwd is: ldap:/dc=com/dc=bacula/ou=people/
$ dir
-rw-r--r-- 1 root  root 1006618   ldap:/dc=com/dc=bacula/ou=people/uid=john
$ add *
1 file marked.
```

or for the MSAD plugin:

```
$ dir
drwxr-xr-x 1 root  root   924    msad:/DC=com/DC=bacula/DC=msad/CN=Builtin/
drwxr-xr-x 1 root  root   621    msad:/DC=com/DC=bacula/DC=msad/CN=Computers/
-rw-r--r-- 1 root  root   723    msad:/DC=com/DC=bacula/DC=msad/
↪CN=Infrastructure
-rw-r--r-- 1 root  root   630    msad:/DC=com/DC=bacula/DC=msad/
↪CN=LostAndFound
-rw-r--r-- 1 root  root   663    msad:/DC=com/DC=bacula/DC=msad/CN=NTDS Quotas
drwxr-xr-x 1 root  root   579    msad:/DC=com/DC=bacula/DC=msad/CN=Program␣
↪Data/
drwxr-xr-x 1 root  root   583    msad:/DC=com/DC=bacula/DC=msad/CN=System/
drwxr-xr-x 1 root  root   601    msad:/DC=com/DC=bacula/DC=msad/CN=Users/
-rw-r--r-- 1 root  root  1495    msad:/DC=com/DC=bacula/DC=msad/CN=backup
drwxr-xr-x 1 root  root   775    msad:/DC=com/DC=bacula/DC=msad/OU=Domain␣
↪Controllers/
$ add CN=backup
1 file marked.
```

You can change the restore subtree using a **where=...** parameter with restore command. It should contain a relocation DN, i.e:

```
* restore where = "dc=restore,dc=example,dc=com"
```

The LDAP or MSAD Plugin relocation restore works similar to a standard **where=/tmp/restores** does for regular files and directories recovering the whole object's subtree path. You may also change the replace mode during the restoration process. The supported modes are: *always*, *never*, *ifnewer*, *ifolder*. Replace mode can be changed during `bconsole restore` command execution.

The plugin saves all standard and extended attributes even if they are not directly recoverable or recoverable at all (read only attributes). Some of the saved attributes will be indirectly restored (like **memberOf** attribute), others will be simply skipped during restore. In this case the you will get appropriate information in the job log, similar to the following example below:

```
JobId 220: msad: Restore for AD Schema(69): Windows Server 2012 R2
JobId 220: Elapsed time=00:00:01, Transfer rate=1.021 K Bytes/second
JobId 220: msad: Skipping read-only attribute: DN: ... Attr: whenCreated
JobId 220: msad: Skipping read-only attribute: DN: ... Attr: whenChanged
(...)
```

Restore Options

Both plugins support a restore options interface where you can set required plugin variables. With restore options interface you can change the location of the restore server, bind user or specific restore mechanism.

To use the restore options interface you need to select option 13 of the modification interface during restore process:

```
(...)
13: Plugin Options
Select parameter to modify (1-13): 13
```

For LDAP Plugin you have a following options:

```
Plugin Restore Options
basedn:              *None*              ()
ldapuri:             *None*              ()
binddn:              *None*              ()
bindpass:            *None*              ()
hbindpass:           *None*              ()
config:              *None*              ()
attribs:             *None*              ()

You have the following choices:
1: basedn (Base location (DN) for restore, it could be a ldap server root␣
↪tree \
    or some other subtree. ex: dc=example,dc=com)
2: ldapuri (Universal Resource Identifier for MSAD server, connection string.␣
↪\
    ex: ldap://192.168.0.100/)
3: binddn (Restore user distinguish name. ex: cn=backup,dc=example,dc=com)
4: bindpass (Restore user password)
5: hbindpass (Restore user password obscured)
6: config (Path to alternate configuration file)
7: attribs (Additional attribute list to skip during the restore (comma␣
↪separated list))
```

And for MSAD Plugin you have the following options:

```
Plugin Restore Options
basedn:              *None*              ()
ldapuri:             *None*              ()
binddn:              *None*              ()
bindpass:            *None*              ()
hbindpass:           *None*              ()
config:              *None*              ()
schemaapply:         *None*              (yes)
attribs:             *None*              ()

You have the following choices:
1: basedn (Base location (DN) for restore, it could be a ldap server root␣
↪tree \
    or some other subtree. ex: dc=example,dc=com)
2: ldapuri (Universal Resource Identifier for MSAD server, connection string.␣
↪\
    ex: ldap://192.168.0.100/)
3: binddn (Restore user distinguish name. ex: cn=backup,dc=example,dc=com)
4: bindpass (Restore user password)
5: hbindpass (Restore user password obscured)
6: config (Path to alternate configuration file)
7: schemaapply (Skip automatically well known read-only attributes)
8: attribs (Additional attribute list to skip during the restore (comma␣
↪separated list))
```

- **ldapuri** Universal Resource Identifier for LDAP/MSAD server, connection string
- **binddn** backup user distinguish name
- **bindpass** backup user password
- **hbindpass** backup user encoded password
- **basedn** A base location (DN) for backup, it could be a ldap server root tree or some other subtree
- **config** A config file to use which has all required parameters
- **schemaapply** A boolean option (yes/no) to toggle an automatic skipping of well known read-only attributes of the Active Directory server
- **attribs** It is a comma separated list of additional attributes which plugin should skip during restore

When you restore LDAP or Active Directory objects sometimes you can get a following error:

```
ldap err: 53 errmsg: "0000209A: SvcErr: DSID-031A104A, problem 5003 (WILL_NOT_
↪PERFORM)
```

which means some of the restored attributes cannot be restored due to restrictions set at Directory server side. These attributes need to be skipped during restore to meet the restrictions. The plugin can automatically skip some well known system/read-only attributes when you toggle the restore parameter: **schemaapply** to yes (the default value). Then during restore a plugin will verify the Active Directory schema number and apply the correct restore filter to this attributes. You can extend this filter with your own list using **attribs** restore parameter. This parameter is available for LDAP and MSAD plugin. The **schemaapply** parameter is available for MSAD plugin only.

The restore job log will provide information about attributes skipped during restore, see *Restore*.

Active Directory restore

Active Directory is a special kind of Directory server which requires special treatment during restore operations, and this is handled by the dedicated MSAD plugin.

Tombstone recovery

Note: New in 10.0.2

As of version 10.0.2, the MSAD Plugin supports Active Directory tombstone objects recovery.During restore the MSAD Plugin automatically checks if a restored object has an available tombstone. If found the MSAD Plugin recovers it from tombstone and restores all other attributes from backup. This feature allows proper recovery for some system attributes like `SID` or `objectGUID` which aren't recoverable with other methods.

memberOf recovery

The plugin recognizes the `memberOf` object's attribute during restore and handles it correctly. After a successful object restore it adds restored object to all groups pointed by the `memberOf` attribute list. When a destination group does not exist on the AD server this group will be skipped.

userAccountControl recovery

The `userAccountControl` attribute in Active Directory requires special permissions for successful restore. It may be required to add `Enterprise Admins` security group to your `BINDDN` restore user or modify `BINDDN` user permissions directly.

The plugin will check if `userAccountControl` attribute restore is allowed with current permissions and inform the user in the job log if the restore was unsuccessful. In this case `userAccountControl` attribute will be restored with default flags, i. e. user account will be disabled.

sAMAccountName

The `sAMAccountName` attribute found in Active Directory user object is a special attribute which has to be unique on the whole directory context. This requires special handling during a relocation restore when an original user object already exist in its original location. In this case the object relocated during restore will be altered. The default `sAMAccountName` attribute will be suffixed with _$(JobID), the value of the restore job, i. e. when an original `sAMAccountName` attribute has a value: `testuser` then during restore it becomes: `testuser_230` where the restore job id is 230. This `sAMAccountName` attribute change is performed only when the following conditions are met:

- restore job executed with relocation - **where=...** parameter different from `/`
- the `sAMAccountName` attribute value already exists on directory context, i. e. an original user object still exists on directory server

Other

Common Problems

- Error: msad plugin: ldap err: 50 errmsg: "00000005: SecErr: DSID-031521E1, problem 4003 (INSUFF_ACCESS_RIGHTS) - Your backup and restore user has no rights to create or modify restored objects, please add a required permissions to the user.
- Error: msad plugin: ldap err: 53 errmsg: "0000209A: SvcErr: DSID-031A104A, problem 5003 (WILL_NOT_PERFORM), data 0 - Some of restored AD attributes cannot be modified and you need to exclude them using Restore Plugin Options as described in **restoreopt**.
- You have to use `ldap:` or `msad:` as a name of the plugin (please check the colon ':' character) even when no additional parameters were set. Without it the job won't backup any objects and finishes without an error.

Object listing

The Bacula Enterprise LDAP/MSAD Plugin supports the new Plugin Listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available LDAP/MSAD objects or server default name context.

The new feature uses the special **.ls** command with a new **plugin=<plugin>** parameter. The command requires the following parameters to be set:

client=<client>
A Bacula Client name with the LDAP/MSAD Plugin installed.

plugin=<plugin>
A Plugin name, which should be **ldap:** or **msad:** in this case, with optional plugin parameters as described in section **parameters**.

path=<path>
An object path to display.

If you do not specify a plugin parameters in command line then plugin will use parameters defined in default config file (i.e. `/opt/bacula/etc/ldap.conf`) or default parameters value which in most cases are very different from your environment. All examples below assumes you have a valid default config file available.

The supported values for a **path=<path>** parameter are:

/ to display ldap server default name context.

`DN` to display all child objects at a following **DN**.

To display ldap server default name context, use the following command example:

```
*.ls client=bacula-fd plugin=msad: path=/
Connecting to Client bacula-devel-fd at bacula-devel:9102
drwxr-x---   1 root     root               0 2018-05-21 09:31:39  DC=dom,
↪DC=com
2000 OK estimate files=1 bytes=0
```

To display the contents of a selected ldap subtree :

```
*.ls client=bacula-devel-fd plugin=msad: path=DC=dom,DC=com
Connecting to Client bacula-devel-fd at bacula-devel:9102
 root      root      137 2018-05-04 20:37:32  CN=Builtin,DC=dom,DC=com/
 root      root      135 2018-05-04 20:37:28  CN=Computers,DC=dom,DC=com/
 root      root      153 2018-05-04 20:37:29  OU=Domain Controllers,DC=dom,
↪DC=com/
 root      root      151 2018-05-04 20:37:29  CN=ForeignSecurityPrincipals,
↪DC=dom,DC=com/
 root      root      151 2018-05-04 20:37:29  CN=Infrastructure,DC=dom,DC=com
 root      root      141 2018-05-04 20:37:28  CN=LostAndFound,DC=dom,DC=com
 root      root      150 2018-05-04 20:37:29  CN=Managed Service Accounts,
↪DC=dom,DC=com/
 root      root      147 2018-05-04 20:37:29  CN=NTDS Quotas,DC=dom,DC=com
 root      root      138 2018-05-04 20:37:29  CN=Program Data,DC=dom,DC=com/
 root      root      142 2018-05-16 21:47:46  OU=restore,DC=dom,DC=com/
 root      root      132 2018-05-04 20:37:28  CN=System,DC=dom,DC=com/
 root      root      161 2018-05-04 20:37:29  CN=TPM Devices,DC=dom,DC=com
 root      root      131 2018-05-04 20:37:28  CN=Users,DC=dom,DC=com/
2000 OK estimate files=13 bytes=600
```

Limitations

- For a restore to work, the LDAP/MSAD server must be fully functional.
- It is not possible to restore LDAP objects as a regular files because LDAP nodes (bacula directories in catalog) contain attributes (data to restore) like LDAP leafs (bacula files in catalog). These limitations may be removed in the future.
- Active Directory tombstone recovery is performed automatically for the MSAD Plugin and cannot be disabled.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Backing up SAP

Overview

This document presents various techniques and strategies to backup SAP with **Bacula Enterprise** version 6.4 and later, which are not applicable to prior versions.

The sysadmin installing the Bacula Enterprise SAP plugin is assumed to have a good general knowledge of Bacula and in particular configuration of Bacula plugins. In addition, throughout this document, we assume that the sysadmin has a very good understanding of SAP and how it works.

The goal of document is to demonstrate how to configure Bacula backups of SAP databases, HANA and Oracle in particular.

Conventions Used In This Document

- <SAPSID> Anything between < and > should be adapted by the user, for example, <SAPSID> should be replaced by your current SAP SID. If your SAP SID is TEST a file written as init<SAPSID>.ora will become initTEST.ora.
- SAP1 is a valid <SAPSID> example used in this document.
- % means that the command should be run with a normal unix user such as SAP1.
- # means that the command should be run with the unix root account.
- <RMAN> means that the command should be run inside a `rman` session.
- <SQL> means that the command should be run inside a `sqlplus` session.

Presentation

The Bacula Enterprise SAP Plugin implements the official SAP Backint interface to simplify the backup and restore procedure using traditional SAP database tools.

The current Bacula Enterprise SAP Plugin implements the Backup and Restore Interface for:

- BC-BRI BACKINT Interface for Oracle databases
- SAP DB Systems (SAPDB/MAXDB) 7.x
- SAP HANA BACKINT 1.04

Bacula Systems has done thorough testing on the BACKINT interface for Oracle and HANA databases. The SAP DB Systems have not yet had significant testing.

The Bacula Enterprise SAP plugin can be combined with the Bacula Enterprise Oracle SBT Plugin to allow direct data transfer between Oracle RMAN and Bacula Enterprise (`backup_dev_type = rman_util`). This configuration permits incremental backups to be run.

This plugin is available on RedHat Linux 7-8, 64 bit and SUSE Linux Enterprise Server 12 and 15, 64 bit platforms. If you need this plugin on other platforms please contact Bacula Support. This plugin is available on Linux platforms 32/64bit supported by SAP.

Architecture

The figure below will help explain how the various pieces fit together. At the top with the light purple background, it shows the Oracle database with its various files and logs. In the middle with light green background, you see how the SAP BRTools interface with Oracle and the database, and connect to the BACKINT program furnished by Bacula Systems shown with a light gray background. The BACKINT program in turn interfaces to the Bacula SAP plugin (sap-fd).

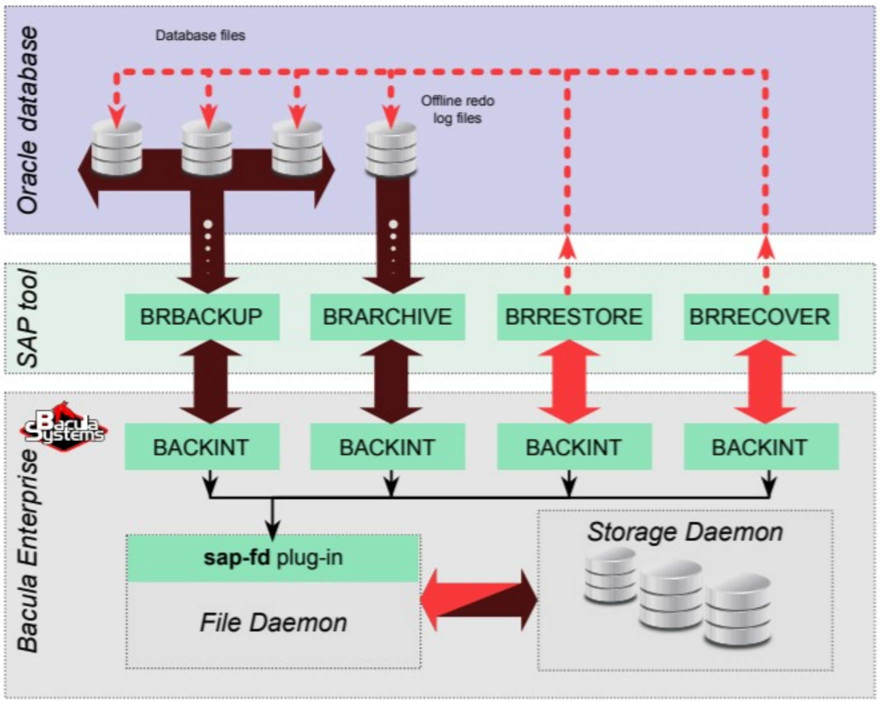


Fig. 66: Interaction Between SAP with Oracle, Backint and Bacula

For everything to work, each of the pieces must be installed in the proper directories and, in general, each component has a configuration file (or parameter file).

Installing the SAP Plugin

SAP HANA

```
bacula-enterprise-sap-hana-plugin-8.10.0-1.el6.x86_64.rpm
```

The first step is to install the Bacula SAP HANA plugin package, which also contains the `hdbbackint` program. This package, shown above, must be installed on the Client machine where your SAP HANA database resides. In addition, please be aware that you ensure that the Bacula Enterprise **bconsole** program is also installed on the Client machine where the SAP database and Bacula Enterprise SAP HANA plugin are installed.

The Bacula Enterprise SAP package contains the following files:

```
/opt/bacula/scripts/install-sap.sh
/opt/bacula/bin/hdbbackint
/opt/bacula/plugins/sap-fd.so
/opt/backint/bacula-enterprise/hdbbackint
```

SAP Oracle/MaxDB

```
bacula-enterprise-sap-8.10.0-1.el6.x86_64.rpm
```

The first step is to install the Bacula SAP plugin package, which also contains the BACKINT program. This package, shown above, must be installed on the Client machine where your SAP database resides. In addition, please be aware that you ensure that the Bacula Enterprise **bconsole** program is also installed on the Client machine where the SAP database and Bacula Enterprise SAP plugin are installed.

The Bacula Enterprise SAP package contains the following files:

```
/opt/bacula/scripts/install-sap.sh
/opt/bacula/bin/backint
/opt/bacula/plugins/sap-fd.so
```

Upgrade

To upgrade an existing installation of the SAP plugin to Bacula Enterprise 8.10, the following commands must be added to the restricted Console ACLs used by the backint process. The Director configuration must be reloaded.

- `.bvfs_delete_fileid`
- `.bvfs_version`
- `list`

Configuring the SAP Plugin

As with all Bacula plugins, you must define the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

In order to send commands to the SAP database, the Bacula Enterprise SAP Plugin must share files on disk with SAP. After installing packages provided by Bacula Systems, the shared files are located on `/opt/bacula/sap` where the directory permissions should be:

```
% ls -ld /opt/bacula/sap
drwxrwx--- 13 root sdba 4096 Mar 28 14:04 /opt/bacula/sap
```

As seen above, sbda is the main group of the unix user running SAP. This permission is automatically set when installing packages, however, if your SAP unix group is not sbda, you may need to manually set permissions and ownership on this directory yourself and make sure that your changes are still in effect after each upgrade of the Bacula Enterprise SAP Plugin package. You also might want to relocate these control files. If so, please see section *Restoring to a Different Tenant*

The Fileset that you normally would use to backup SAP is as follows:

```
# cat SAPFileset.cfg
Fileset {
  Name = SAPFileset
  Include {
    Options { Signature = MD5 }
    Plugin = "sap"
  }
```

Configuring HDBBackint/Backint

In this section, we describe how to install and configure the Bacula Enterprise Backint interface with SAP.

When running a backup or restore from BRTools, Backint will need to contact the Bacula Enterprise Director to get information about files and volumes, or run backup and restore jobs. This communication involves shared FIFO command files, and the bconsole program.

By default, the Bacula Director will not be able to start a backup from bconsole or from a Schedule. Only SAP Tools will be able to initiate the session and start a Backup. If you would prefer to use Bacula's scheduling feature to initiate backups, you can do so by using a Bacula RunScript to execute BRTools, which will in turn initiate a backup. For more details, please see section *Using a Bacula Schedule*.

Bacula Configuration

When using the Backint interface, Bacula's console program bconsole must be installed, and the console must be able to connect to your Director and have access to the local Client, the backup Job and other Pool specifications. This is normally done by using an Admin **bconsole**, which does not have access restrictions. If you want to use a restricted console, please see the *Using Restricted Consoles* section.

For a given <SAPID> SAP1, a backup of the SAP database files can be done using the following Job and Fileset:

```
# The SAP-Backup job will be run each time backint and brtools
# are exectuted. This Job should never be run by backup administrators.
Job {
  Name    = SAP-BRTOOLs
  Fileset = SAP-Fileset
  Client  = sap-fd
  Maximum Concurrent Jobs = 10
}

Fileset {
  Name = SAP-Fileset
  Include {
    Options {
```

(continues on next page)

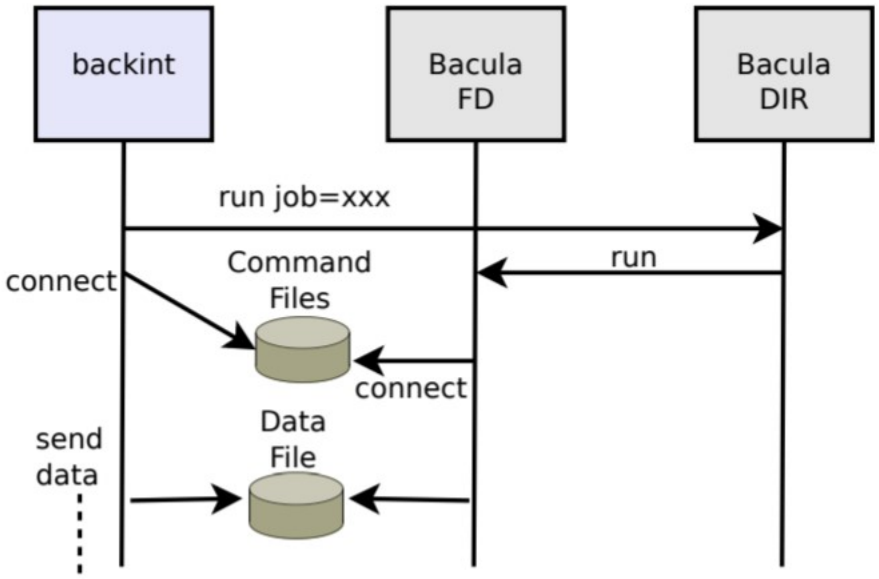


Fig. 67: Interaction Between HDBBackint/Backint and Bacula

(continued from previous page)

```
        Signature = MD5
     }
     Plugin = "sap"
  }
}
```

Then, the following regular backup job "SAP-Env-Backup" may be used to backup non essential SAP files. This regular backup job should not backup the SAP files that are handled by the SAP-BRTOOLs job show above. If such a scenario can't be avoided, it is advised to create a second Client that points to the same physical SAP File Daemon and run jobs with it.

```
Job {
  Name     = SAP-Env-Backup
  Fileset = SAP-Env-Fileset
  Client  = sap-fd              # or a 2nd client that points to sap-fd
  Maximum Concurrent Jobs = 10
}

Fileset {
  Name = SAP-Env-Fileset
  Include {
    Options {
       Signature = MD5
    }
    File=/usr/sap/SAP1/SYS
    File=/opt/oracle/11/       # /<ORACLE_HOME>/<ORACLE_VERSION>
  }
  Exclude {
```

(continues on next page)

(continued from previous page)

```
    File=/u01                    # files handled by SAP Plugin
  }
}
```

The unix user `SAP1` should be able to execute `bconsole` and read the associated configuration file `bconsole.conf`.

```
# cp /opt/bacula/etc/bconsole.conf /usr/sap/SAP1/SYS
# chown SAP1 /usr/sap/SAP1/SYS/bconsole.conf
# chmod go-rxw /usr/sap/SAP1/SYS/bconsole.conf
```

Backint Installation and Configuration

After the Bacula SAP plugin package is installed as shown above, the `backint` or the `hdbbackint` program will be present in `/opt/bacula/bin` directory. You must then ensure that it is either copied or linked into the directory where the SAP Tools programs reside and that it is properly configured:

```
# ls /opt/bacula/bin/backint
-r-xr-xr-x 1 root root 95654 2013-03-02 01:00 backint

# ls /opt/bacula/bin/hdbbackint
-r-xr-xr-x 1 root root 95654 2013-03-02 01:00 hdbbackint
```

The `backint/hdbbackint` program must be installed in the same directory where the SAP tools are installed. This is because SAP uses the backint program as an interface between its tools and Bacula. The SAP directory in question contains all the SAP BRTools (brachive, brbackup, brtools, . . .) is usually:

```
  /usr/sap/<SAPSID>/SYS/exe/run
or
  /usr/sap/<SAPSID>/SYS/global/hdb/opt
```

The final installation stage of the `backint` or `hdbbackint` program can be done automatically, in most cases, using the `install-sap.sh` script available under `/opt/bacula/scripts`

```
# /opt/bacula/scripts/install-sap.sh install
```

The script will create a backint configuration file (also called the Backint parameter file) and will install the backint binary inside the SAP instance file tree. A template of the configuration which needs to be added to the Director's bacula-dir.conf file will also be generated.

Configuring the Backint Parameter File

Backint can be configured with the `/opt/bacula/sap/backint.conf` file, which is referred to as the Backint Parameter File in SAP. It may also be configured using the `-p` option of the BRTools command. With an Oracle database, the default parameter file may be symlinked to its default location by using the following command:

```
# ln -s /opt/bacula/sap/backint.conf ${ORACLE_HOME}/dbs/init${ORACLE_SID}.utl
```

The keywords presented here are accepted in the backint.conf file.

Parameter	Example	Description	Required
client	`client=sap-fd`	Bacula Client name.	Yes
restoreclient	`restoreclient=s`	Bacula Client name used to restore data.	No
ClientCluster	`clientcluster=s sap2-fd, sap3-fd`	List of the Bacula Clients included in the SAP cluster.	No
job	`job=SAP-BRTOOLs`	Bacula Backup Job name.	Yes
bconsole	`bconsole="/ opt/bacula/ bin/bconsole -n -c /opt/ bacula/sap/ bconsole. conf"`	Bconsole command with all arguments.	Yes
RestoreJob	`restorejob=Rest`	Bacula Restore Job name. If multiple restore jobs are defined in your configuration and this option is not used, `backint` will automatically choose the first restore Job defined.	No
WaitJobCompletion	`waitjobcompleti`	Indicates to wait for Job completion at the end of the `backint` session. The default is to finish the `backint` session as soon as possible.	No
JobOpt	`jobopt="spoolda`	Allows you to specify additional Job options.	No
MaximumConcurrentJobs	`MaximumConcurre`	Specifies the number of concurrent job sessions allowed. All Bacula resources must be configured properly to allow jobs to run in parallel.	No
CtrlFile	`ctrlfile=/ opt/bacula/ sap/backint`	Specifies the base path of control files used to connect with the bacula-fd plugin. You must use the same location on the Plugin command line in the Fileset, and in the backint.conf configuration file.	No
wait_retry	`wait_retry=30`	Specifies the number of times that `backint` will try to reach the Bacula Enterprise SAP plugin (10s between each try). Starting with 8.6.4, it is also possible to use the keyword `retry`.	No
catalog	`catalog="MyCata 2"`	Specifies a Bacula Catalog name if your director is using multiple catalogs.	No
trace	`trace=/tmp/ log.txt`	Points to an optional trace file.	No
debug	`debug=50`	Debug level.	No
RawFileOffset	`rawfileoffset=4`	On some systems, raw devices use an offset. You can use the $ORACLE_HOME/bin/offset utility to determine the offset value.	No

A minimal parameter file will require the `client`, `job` and `bconsole` options to be set. Note that if the configuration option contains spaces (such as bconsole), you must use double quotes to enclose it. The keyword is case insensitive.

```
# cat ${ORACLE_HOME}/dbs/init<ORACLE_SID>.utl
client=sap-fd
job=SAP-BRTOOLs
bconsole="/opt/bacula/bin/bconsole -n -c /usr/sap/SAP1/SYS/exe/run/bconsole.
↪conf"
ctrlfile="/usr/sap/SAP1/bacula/base"
maximumconcurrentjobs=5
```

If the `ctrlfile` option is set in the `backint.conf` file, the same `ctrlfile` option should be specified in the Fileset Plugin command line.

For a SAP HANA cluster:

```
# cat /opt/bacula/sap/backint.conf
client=sap-fd
clientcluster=sap-fd,sap2-fd,sap3-fd
job=SAP-BRTOOLs
bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/sap/bconsole.conf"
maximumconcurrentjobs=5
waitjobcompletion=yes
```

TOOLOPTION SAP Configuration

Available since Bacula Enterprise 12.2.4

The `TOOLOPTION` parameter can be used to customize some backint parameter at the runtime. The following options can be modified:

- job
- pool
- level

```
hdbsql -i 00 -u SYSTEM -p X -d SYSTEMDB "BACKUP DATA INCREMENTAL USING␣
↪BACKINT ('Inc2') TOOLOPTION 'level=full'"
```

Running Parallel Jobs

In order to run Backup or Restore using multiple channels, you need to ensure that all required resources in Bacula are properly configured using appropriate `Maximum Concurrent Jobs` directive settings to allow concurrent jobs (see table below). By default, if `Maximum Concurrent Jobs` is not explicitly set, the default value is 1, so you must ensure that all resources are properly configured.

Table 31: Setting Maximum Concurrent Jobs

Component	Resource	Possible Value
Director		
	Director	MaximumConcurrentJobs=100
	Client	MaximumConcurrentJobs=10
	Job	MaximumConcurrentJobs=10
	Storage	MaximumConcurrentJobs=20
Storage		
	Storage	MaximumConcurrentJobs=20
	Device	MaximumConcurrentJobs=10
File Daemon		
	FileDaemon	MaximumConcurrentJobs=10
Backint (on Client)		
		MaximumConcurrentJobs=10

To allow concurrent Backup and Restore jobs using the same Director Storage resource, the configuration must use a Virtual Autochanger disk device. See the *Disk Backup* white paper about this specific configuration.

Getting Debug and Trace Outputs

To enable debug traces with the Bacula Enterprise SAP Backint Plugin, it is necessary to combine both File Daemon Plugin traces and Backint traces at level 50.

```
# bconsole
* setdebug level=50 trace=1 client=<clientname> options=t
```

Traces are stored in the Client's working directory

```
/opt/bacula/working/<host-fd>.trace
```

Do not forget to disable debug traces on the client

```
# bconsole
* setdebug level=0 trace=0 client=<clientname>
```

To enable traces for Backint, you must modify the parameter file and use `debug` and `trace` options.

```
# cat $ORACLE HOME/dbs/init<ORACLE_SID>.utl
trace=/tmp/backint.log
debug=50
```

Testing backint.conf/init<SID>.utl Configuration

To test the Bacula Enterprise SAP Backint Plugin configuration, the following command can be run as the `root` user:

```
# /opt/bacula/scripts/install-sap.sh test
Enter the unix SAP account name: sapuser
1000 OK: sap-dir Version: 6.4.6 (18 Aug 2013)
INFO: Connection to the Director OK
INFO: Connection from the Director to the Client OK
INFO: Plugin installed correctly
INFO: Job found on the Director
INFO: Fileset configured on the Director
```

If a connection error is detected, a message will be displayed. It is useless to run any SAP backup until the connection is properly configured.

SAP Configuration

To use the Bacula Enterprise Backint SAP Plugin, the following entries have to be added in the SAP parameter file `init<ORACLE_SID>.sap`. For example, in the case of ORACLE:

```
backup_dev_type=util_file             #  Backup Device Type
util_par_file=$ORACLE_HOME/dbs/init<ORACLE_SID>.utl$
```

The `util_par_file` should point to the Backint parameter file.

For installations with Oracle 11g (or higher) RAC, the parameter file is located in the directory `<SAPDATA.HOME>/sapprof`. For more information see the SAP Database Guide for Oracle documentation on the `util_par_file` parameter.

SAP Backup Retention

When using SAP plugin of the Bacula Enterprise, the backup retention defined in SAP should match the Bacula Volume or Job retention. When the SAP backup retention expires and SAP sends commands to delete backup files, Bacula will delete selected files but will not purge associated jobs. So in order to have the Bacula catalog entries for these files deleted, they must also have the same Bacula Volume or Job retention settings in the bacula-dir.conf file.

Backup Scenarios

Using SAP Oracle Tools

You can start a backup through the SAPDBA utility's menus or through the `brbackup` command line.

```
# brbackup -t online -m full -c force
BR0051I BRBACKUP 7.00 (40)
BR0055I Start of database backup: belxgvax.fnf 2013-08-20 10.46.03
BR0484I BRBACKUP log file: /u01/db_1/sapbackup/belxgvax.fnf
BR0477I Oracle pfile /u01/db_1/dbs/initEAR.ora created from spfile /u01/db_1/
```

(continues on next page)

(continued from previous page)

```
↪dbs/spfileEAR.ora

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.04
BR0057I Backup of database: EAR
BR0058I BRBACKUP action ID: belxgvax
BR0059I BRBACKUP function ID: fnf
BR0110I Backup mode: FULL
BR0077I Database file for backup: /u01/db_1/sapbackup/cntrlEAR.dbf
BR0061I 6 files found for backup, total size 861.773 MB
BR0143I Backup type: online
BR0130I Backup device type: util_file
BR0109I Files will be saved by backup utility
BR0134I Unattended mode with 'force' active - no operator confirmation allowed

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.04
BR0315I 'Alter database begin backup' successful

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.04
BR0229I Calling backup utility with function 'backup'...
BR0278I Command output of 'backint -u EAR -f backup -i /u01/db_1/sapbackup/.
↪belxgvax.lst
                             -t file -p /opt/bacula/sap/backint.conf -c':
Profile: /opt/bacula/sap/backint.conf
Input File: /u01/db_1/sapbackup/.belxgvax.lst
Output File: *None*
Client node: sap-fd

...

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.18
#FILE..... /u01/oradata/EAR/undotbs01.dbf
#SAVED.... EAR-51

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.18
#FILE..... /u01/oradata/EAR/users01.dbf
#SAVED.... EAR-51

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.18
BR0232I 5 of 5 files saved by backup utility
BR0230I Backup utility called successfully

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.20
BR0530I Cataloging backups of all database files...

BR0522I 5 of 5 files/save sets processed by RMAN

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.24
BR0531I Backups of all database files cataloged successfully

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.24
BR0317I 'Alter database end backup' successful
```

(continues on next page)

(continued from previous page)

```
BR0280I BRBACKUP time stamp: 2013-08-20 10.46.24
BR0340I Switching to next online redo log file for database instance EAR ...
BR0321I Switch to next online redo log file for database instance EAR␣
↪successful

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.24
BR0319I Control file copy created: /u01/db_1/sapbackup/cntrlEAR.dbf 7061504

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.24
BR0229I Calling backup utility with function 'backup'...
BR0278I Command output of 'backint -u EAR -f backup -i /u01/db_1/sapbackup/.
↪belxgvax.lst
                              -t file -p /opt/bacula/sap/backint.conf -c':
Profile: /opt/bacula/sap/backint.conf
Input File: /u01/db_1/sapbackup/.belxgvax.lst
Output File: *None*
Client node: sap-fd

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.27
#FILE..... /u01/db_1/sapbackup/cntrlEAR.dbf
#SAVED.... EAR-52

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.27
BR0232I 1 of 1 file saved by backup utility
BR0230I Backup utility called successfully

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.27
BR0229I Calling backup utility with function 'backup'...
BR0278I Command output of 'backint -u EAR -f backup -i /u01/db_1/sapbackup/.
↪belxgvax.lst
                              -t file -p /opt/bacula/sap/backint.conf -c':
Profile: /opt/bacula/sap/backint.conf
Input File: /u01/db_1/sapbackup/.belxgvax.lst
Output File: *None*
Client node: sap-fd

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.28
#PFLOG.... /u01/db_1/dbs/initEAR.ora
#SAVED.... EAR-53

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.28
#PFLOG.... /u01/db_1/dbs/spfileEAR.ora
#SAVED.... EAR-53

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.28
#PFLOG.... /u01/db_1/dbs/initEAR.sap
#SAVED.... EAR-53

...

BR0280I BRBACKUP time stamp: 2013-08-20 10.46.28
BR0232I 8 of 8 files saved by backup utility
```

(continues on next page)

(continued from previous page)

```
BR0230I Backup utility called successfully

BR0056I End of database backup: belxgvax.fnf 2013-08-20 10.46.28
BR0280I BRBACKUP time stamp: 2013-08-20 10.46.28
BR0052I BRBACKUP completed successfully
```

To use the interactive command line, use `brtools`.

```
% brtools
...
Backup and database copy

 1 = Database backup
 2 - Archivelog backup
 3 - Database copy
 4 - Non-database backup
 5 - Backup of database disk backup
 6 - Verification of database backup
 7 - Verification of archivelog backup
 8 - Additional functions
 9 - Reset program status

...

BR0280I BRARCHIVE time stamp: 2013-08-20 11.29.36
BR0229I Calling backup utility with function 'backup'...
BR0278I Command output of 'backint -u EAR -f backup
     -i /u01/db_1/saparch/.aelxgyxc.lst
     -t file -p /opt/bacula/sap/backint.conf':

Profile: /opt/bacula/sap/backint.conf
Input File: /u01/db_1/saparch/.aelxgyxc.lst
Output File: *None*
Client node: sap-fd


...

BR0232I 8 of 8 files saved by backup utility
BR0230I Backup utility called successfully
```

Using a Bacula Schedule

In order to control the database backup from the Director side and use Bacula's scheduler, a Job using a RunScript can be used as presented in the following example.

```
Job {
 Name = job.sap-fd.base
 Description = "Basic job for SAP server"
 ClientRunBeforeJob = "/opt/bacula/scripts/job.sap-fd.EAR %l"
 Schedule = sched.sap-fd.base
```

(continues on next page)

(continued from previous page)

```
 Fileset = fs.sap-fd.base
 ...
}
Fileset {
  Name = fs.sap-fd.base
  Include {
    Options { Signature = MD5 }
    File=/usr/sap/EAR/SYS
    File=/opt/oracle/11/       # /<ORACLE_HOME>/<ORACLE_VERSION>
...
    File = /etc
  }
}
Schedule {
  Name = sched.sap-fd.base
  ....
}

Job {
 Name = job.sap-fd.EAR
 Description = "Backint SAP Job for DB EAR"
 Fileset = fs.sap-fd.EAR
 # No Schedule for this job, will be triggered by job.sap-fd.base
 # No RunScript for this job
...
}

Fileset {
 Name = job.sap-fd.EAR
 Include {
   Options { Signature = MD5 }
   Plugin = sap
 }
}
```

The main goal here is to perform the backup of some essential files and spawn a SAP Backint backup session using a RunScript. The RunScript will start a new Job. The custom script might be something like:

```
% cat /opt/bacula/scripts/job.sap-fd.EAR
#!/bin/sh

level=incr

if [ "$1" = f ]; then
  level=full

elif [ "$1" = d ]; then
  level=diff
fi

su - sapuser -c "brbackup -t online -m $level -c force"
```

(continues on next page)

(continued from previous page)

```
exit $?
```

Note: The “brbackup” command should be adapted to your environment using brtools or sapdba tools.

To use this scripting technique, the configuration should allow running multiple concurrent Jobs on the SAP Client (See the MaximumConcurrentJobs directive in Table **tab:maximumconcurrentjobs**).

The backint parameter file will probably be something like:

```
client=sap-fd
job=job.sap-fd.EAR
bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/sap/bconsole.conf"
```

Restore Scenarios

Using BRRESTORE SAP Tools

```
# brrestore -m 0
BR0401I BRRESTORE 7.00 (40)
BR0405I Start of file restore: remllqsn.rsb 2013-11-03 10.29.45
BR0484I BRRESTORE log file: /oracle/product/10.2.0/db_1/sapbackup/remllqsn.rsb

BR0428W File /oracle/product/10.2.0/oradata/EAR/control01.ctl will be␣
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/control02.ctl will be␣
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/control03.ctl will be␣
↪overwritten

BR0280I BRRESTORE time stamp: 2013-11-03 10.29.45
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRRESTORE:
c
BR0280I BRRESTORE time stamp: 2013-11-03 10.29.46
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

BR0280I BRRESTORE time stamp: 2013-11-03 10.29.46
BR0407I Restore of database: EAR
BR0408I BRRESTORE action ID: remllqsn
BR0409I BRRESTORE function ID: rsb
BR0449I Restore mode: partial
BR0411I Database files for restore:
/oracle/product/10.2.0/oradata/EAR/control01.ctl
/oracle/product/10.2.0/oradata/EAR/control02.ctl
/oracle/product/10.2.0/oradata/EAR/control03.ctl
BR0419I Files will be restored from backup: bemllqeh.anf 2013-11-03 10.23.35
BR0416I 1 file found to restore, size 6.797 MB
BR0421I Restore device type: util_file
```

(continues on next page)

(continued from previous page)

```
BR0280I BRRESTORE time stamp: 2013-11-03 10.29.46
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRRESTORE:
c
BR0280I BRRESTORE time stamp: 2013-11-03 10.29.47
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

BR0280I BRRESTORE time stamp: 2013-11-03 10.29.47
BR0229I Calling backup utility with function 'restore'...
BR0278I Command output of 'backint -u EAR -f restore -i /oracle/product/10.2.
↪0/db_1/sapbackup/.remllqsn.lst -t file -p /opt/bacula/sap/backint.conf':
Profile: /opt/bacula/sap/backint.conf
Input File: /oracle/product/10.2.0/db_1/sapbackup/.remllqsn.lst
Output File: *None*
Client node: sap-fd

BR0280I BRRESTORE time stamp: 2013-11-03 10.29.52
#FILE..... /oracle/product/10.2.0/db_1/sapbackup/cntrlEAR.dbf
#RESTORED. EAR-5

BR0280I BRRESTORE time stamp: 2013-11-03 10.29.52
BR0374I 1 of 1 file restored by backup utility
BR0230I Backup utility called successfully

BR0351I Restoring /oracle/product/10.2.0/oradata/EAR/control01.ctl
BR0355I from /oracle/product/10.2.0/db_1/sapbackup/cntrlEAR.dbf ...

BR0351I Restoring /oracle/product/10.2.0/oradata/EAR/control02.ctl
BR0355I from /oracle/product/10.2.0/db_1/sapbackup/cntrlEAR.dbf ...

BR0351I Restoring /oracle/product/10.2.0/oradata/EAR/control03.ctl
BR0355I from /oracle/product/10.2.0/db_1/sapbackup/cntrlEAR.dbf ...

BR0406I End of file restore: remllqsn.rsb 2013-11-03 10.29.52
BR0280I BRRESTORE time stamp: 2013-11-03 10.29.52
BR0403I BRRESTORE completed successfully with warnings
```

```
# brrestore  -p initEAR.sap -b bemllsxd.anf -d util_file -r /opt/bacula/sap/
↪backint.conf -m all -l E -i 30

BR0401I BRRESTORE 7.00 (40)
BR0405I Start of file restore: remlltnz.rsb 2013-11-03 11.01.35
BR0484I BRRESTORE log file: /oracle/product/10.2.0/db_1/sapbackup/remlltnz.rsb

BR0428W File /oracle/product/10.2.0/oradata/EAR/example01.dbf will be␣
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/sysaux01.dbf will be␣
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/system01.dbf will be␣
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/undotbs01.dbf will be␣
```

(continues on next page)

(continued from previous page)

```
↪overwritten
BR0428W File /oracle/product/10.2.0/oradata/EAR/users01.dbf will be␣
↪overwritten

BR0280I BRRESTORE time stamp: 2013-11-03 11.01.35
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRRESTORE:
c
BR0280I BRRESTORE time stamp: 2013-11-03 11.01.38
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

BR0456I Probably the database must be recovered due to restore from online␣
↪backup

BR0280I BRRESTORE time stamp: 2013-11-03 11.01.38
BR0407I Restore of database: EAR
BR0408I BRRESTORE action ID: remlltnz
BR0409I BRRESTORE function ID: rsb
BR0449I Restore mode: ALL
BR0419I Files will be restored from backup: bemllsxd.anf 2013-11-03 10.54.17
BR0416I 5 files found to restore, total size 855.039 MB
BR0421I Restore device type: util_file

BR0280I BRRESTORE time stamp: 2013-11-03 11.01.38
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRRESTORE:
c
BR0280I BRRESTORE time stamp: 2013-11-03 11.01.42
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

BR0280I BRRESTORE time stamp: 2013-11-03 11.01.42
BR0229I Calling backup utility with function 'restore'...
BR0278I Command output of 'backint -u EAR -f restore -i /oracle/product/10.2.
↪0/db_1/sapbackup/.remlltnz.lst -t file -p /opt/bacula/sap/backint.conf':
Profile: /opt/bacula/sap/backint.conf
Input File: /oracle/product/10.2.0/db_1/sapbackup/.remlltnz.lst
Output File: *None*
Client node: sap-fd

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
#FILE..... /oracle/product/10.2.0/oradata/EAR/example01.dbf
#RESTORED. EAR-13

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
#FILE..... /oracle/product/10.2.0/oradata/EAR/sysaux01.dbf
#RESTORED. EAR-13

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
#FILE..... /oracle/product/10.2.0/oradata/EAR/system01.dbf
#RESTORED. EAR-13

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
```

(continues on next page)

(continued from previous page)

```
#FILE..... /oracle/product/10.2.0/oradata/EAR/undotbs01.dbf
#RESTORED. EAR-13

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
#FILE..... /oracle/product/10.2.0/oradata/EAR/users01.dbf
#RESTORED. EAR-13

BR0280I BRRESTORE time stamp: 2013-11-03 11.02.04
BR0374I 5 of 5 files restored by backup utility
BR0230I Backup utility called successfully

BR0406I End of file restore: remlltnz.rsb 2013-11-03 11.02.05
BR0280I BRRESTORE time stamp: 2013-11-03 11.02.05
BR0403I BRRESTORE completed successfully with warnings
```

Verifying Backup Using Brtools

```
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.22
BR0656I Choice menu 9 - please make a selection
---------------------------------------------------------------------------
Backup and database copy

 1 = Database backup
 2 - Archivelog backup
 3 - Database copy
 4 - Non-database backup
 5 - Backup of database disk backup
 6 + Verification of database backup
 7 - Verification of archivelog backup
 8 - Additional functions
 9 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
---------------------------------------------------------------------------
BR0662I Enter your choice:
6
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.26
BR0663I Your choice: '6'

BR0699I Reading log file /oracle/product/10.2.0/db_1/sapbackup/backEAR.log ...

BR0280I BRTOOLS time stamp: 2013-11-03 10.37.26
BR0658I List menu 20 - please select one entry
---------------------------------------------------------------------------
BRBACKUP database backups for verification

Pos.  Log           Start                Type        Files  Device     Rc

  1 = bemllqeh.anf  2013-11-03 10.23.35  onl_cons      5/6  util_file   0
  2 - bemllqad.anf  2013-11-03 10.21.47  onl_cons      0/6  util_file   5
```

(continues on next page)

(continued from previous page)

```
  3 - bemllpwy.anf  2013-11-03 10.20.24  onl_cons      0/6  util_file   5

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
---------------------------------------------------------------------------
BR0662I Enter your selection:
1
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.27
BR0663I Your selection: '1'

BR0280I BRTOOLS time stamp: 2013-11-03 10.37.27
BR0657I Input menu 21 - please enter/check input values
---------------------------------------------------------------------------
BRRESTORE main options for verification of database backup

 1 - BRRESTORE profile (profile) ....... [initEAR.sap]
 2 - BRBACKUP backup run (backup) ...... [bemllqeh.anf]
 3 - Verification device type (device) . [util_file]
 4 ~ BACKINT/Mount profile (parfile) ... [/opt/bacula/sap/backint.conf]
 5 # Database user/password (user) ..... [/]
 6 - Recovery interval (interval) ...... [30]
 7 - Verification mode (verify) ........ [yes]
 8 ~ Files for verification (mode) ..... [full]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
---------------------------------------------------------------------------
BR0662I Enter your choice:
c
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.34
BR0663I Your choice: 'c'
BR0259I Program execution will be continued...

BR0280I BRTOOLS time stamp: 2013-11-03 10.37.34
BR0657I Input menu 22 - please enter/check input values
---------------------------------------------------------------------------
Additional BRRESTORE options for verification of database backup

 1 - Confirmation mode (confirm) ...... [yes]
 2 - Query mode (query) ............... [no]
 3 # Compression mode (compress) ...... [no]
 4 # Parallel execution (execute) ..... [0]
 5 - Additional output (output) ....... [no]
 6 - Message language (language) ...... [E]
 7 - BRRESTORE command line (command) . [-p initEAR.sap -b bemllqeh.anf -d␣
↪util_file -r /opt/bacula/sap/backint.conf -i 30 -w -m full -k no -l E]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
---------------------------------------------------------------------------
BR0662I Enter your choice:
c
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.36
BR0663I Your choice: 'c'
BR0259I Program execution will be continued...
```

(continues on next page)

(continued from previous page)

```
BR0291I BRRESTORE will be started with options '-p initEAR.sap -b bemllqeh.
↪anf -d util_file -r /opt/bacula/sap/backint.conf -i 30 -w -m full -k no -l E
↪'

BR0280I BRTOOLS time stamp: 2013-11-03 10.37.36
BR0670I Enter 'c[ont]' to continue, 'b[ack]' to go back, 's[top]' to abort:
c
BR0280I BRTOOLS time stamp: 2013-11-03 10.37.37
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

#########################################################################

BR0401I BRRESTORE 7.00 (40)
BR0405I Start of file restore: remllrkr.rsb 2013-11-03 10.37.37
BR0484I BRRESTORE log file: /oracle/product/10.2.0/db_1/sapbackup/remllrkr.rsb

BR0280I BRRESTORE time stamp: 2013-11-03 10.37.37
BR0407I Restore of database: EAR
BR0408I BRRESTORE action ID: remllrkr
BR0409I BRRESTORE function ID: rsb
BR0449I Restore mode: FULL
BR0411I Database file for restore: /oracle/product/10.2.0/db_1/sapbackup/
↪cntrlEAR.dbf
BR0414I Offline redo log file for restore of database instance EAR: 28
BR0419I Files will be restored from backup: bemllqeh.anf 2013-11-03 10.23.35
BR0416I 7 files found to restore, total size 863.722 MB
BR0421I Restore device type: util_file
BR0147I Verify option set - verification of backup only, no restore

BR0280I BRRESTORE time stamp: 2013-11-03 10.37.37
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRRESTORE:
c
BR0280I BRRESTORE time stamp: 2013-11-03 10.37.39
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

BR0280I BRRESTORE time stamp: 2013-11-03 10.37.39
BR0229I Calling backup utility with function 'restore'...
BR0278I Command output of 'backint -u EAR -f restore -i /oracle/product/10.2.
↪0/db_1/sapbackup/.remllrkr.lst -t file -p /opt/bacula/sap/backint.conf':
Profile: /opt/bacula/sap/backint.conf
Input File: /oracle/product/10.2.0/db_1/sapbackup/.remllrkr.lst
Output File: *None*
Client node: sap-fd

BR0280I BRRESTORE time stamp: 2013-11-03 10.38.13
#RESTORED. /oracle/product/10.2.0/oradata/EAR/example01.dbf /oracle/product/
↪10.2.0/db_1/sapreorg/example01.dbf EAR-5

BR0280I BRRESTORE time stamp: 2013-11-03 10.38.13
```

(continues on next page)

(continued from previous page)

```
#RESTORED. /oracle/product/10.2.0/oradata/EAR/sysaux01.dbf /oracle/product/10.
↪2.0/db_1/sapreorg/sysaux01.dbf EAR-5

...

BR0280I BRRESTORE time stamp: 2013-11-03 10.38.14
BR0374I 7 of 7 files restored by backup utility
BR0230I Backup utility called successfully

BR0353I Verifying backup of /oracle/product/10.2.0/oradata/EAR/example01.dbf
BR0355I from /oracle/product/10.2.0/db_1/sapreorg/example01.dbf ...

BR0362I Verification of backup of /oracle/product/10.2.0/oradata/EAR/
↪example01.dbf successful

BR0280I BRRESTORE time stamp: 2013-11-03 10.38.23
BR0063I 1 of 7 files processed - 100.008 MB of 863.722 MB done
BR0204I Percentage done: 11.58%, estimated end time: 10:39
BR0001I ******_____________________________________________

...

BR0280I BRRESTORE time stamp: 2013-11-03 10.38.24
BR0063I 7 of 7 files processed - 863.722 MB of 863.722 MB done
BR0204I Percentage done: 100.00%, estimated end time: 10:38
BR0001I **************************************************

BR0406I End of file restore: remllrkr.rsb 2013-11-03 10.38.24
BR0280I BRRESTORE time stamp: 2013-11-03 10.38.24
BR0402I BRRESTORE completed successfully

###########################################################################

BR0292I Execution of BRRESTORE finished with return code 0
```

Restoring to a Different Client or Without SAP

To restore data to a specific directory without calling BRTools, you can use the bconsole `where=` parameter during the restore session. To restore to / without calling SAP tools, the Plugin option `restore_to_disk` can be used from the restore menu.

```
* restore where=/tmp
...

JobId 66: Start Restore Job RestoreFiles.2013-08-20_12.21.44_19
JobId 66: Using Device "FileStorage" to read.
JobId 66: Ready to read from volume "TestVolume001" on file device ...
JobId 66: Forward spacing Volume "TestVolume001" to file:block 0:38525.
JobId 66: Restoring SAP data to /tmp without backint.
JobId 66: Elapsed time=00:00:01, Transfer rate=806  Bytes/second
```

(continues on next page)

(continued from previous page)

```
JobId 66: Bacula 127.0.0.1-dir 6.4.6 (16Aug13):
  Build OS:               x86_64-unknown-linux-gnu archlinux
  JobId:                  66
  Job:                    RestoreFiles.2013-08-20_12.21.44_19
  Restore Client:         sap-fd
  Start time:             20-Aug-2013 12:21:46
  End time:               20-Aug-2013 12:21:47
  Files Expected:         2
  Files Restored:         2
  Bytes Restored:         916
  Rate:                   0.9 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

If you try to restore files without the SAP Plugin installed, a warning message will be printed for each file, and files will be restored properly.

In some cases you might want to use a custom directory for the communication between the backint process and the Bacula FD Plugin. For example, if multiple instances of SAP are running on the same host using different unix users. Using the `ctrlfile` parameter on the Fileset Plugin command and in the backint parameter file will allow you to specify a custom location where unix rights are properly set and will not change over time.

Restoring to a Different Tenant

In order to restore SAP HANA DB through backint to a different client and a different database SID or another tenant, please install and configure the File Daemon and the SAP HANA plugin on the target tenant and restore the database from this newly installed File Daemon. You will specify the `restoreclient` option in backint.conf in order to specify the client name of this restore target tenant.

The configuration file backint.conf on the restore target server would look like this:

```
# cat /opt/bacula/sap/backint.conf
client=sap-fd
job=SAP-BRTOOLs
bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/sap/bconsole.conf"
maximumconcurrentjobs=5
waitjobcompletion=yes
restoreclient=sap-restoretarget-fd
```

Relocating the Control Files

The control files can be relocated to a non-standard or unique directory, and if this is done, the Bacula SAP plugin must know the location of those files, which is done by specifying the **ctrlfile** parameter on the plugin line as follows:

```
# cat SAPFileSet.cfg
Fileset {
  Name = SAPFileSet
  Include {
    Options { Signature = MD5 }
    Plugin = "sap: ctrlfile=/usr/sap/<SAPSID>/bacula/base"
  }

# cat backint.conf
ctrlfile=/usr/sap/<SAPSID>/bacula/base
...
```

In this example, the `backint` process and the Bacula SAP Plugin will create a set of communication links using the `ctrlfile` as base name. For example:

```
-rw------- 1 <SAPID> sdba  0 Jul  2 12:29 /usr/sap/<SAPID>/bacula/base.50
-rw------- 1 <SAPID> sdba  0 Jul  2 12:29 /usr/sap/<SAPID>/bacula/base.50.1
-rw------- 1 <SAPID> sdba  0 Jul  2 12:29 /usr/sap/<SAPID>/bacula/base.50.0
```

The unix owner of the SAP database (in this example `<SAPID>`) must have a read / write access to the specified directory.

Using Restricted Consoles

If you want more granular control on the permissions that the SAP administrator will have when interfacing with Bacula, you might want to use a restricted console. To do so, you may use the following Console definition:

```
Client {
  Name = sap-fd
  Maximum Concurrent Jobs = 10
  ...
}
Console {
  Name = sap-fd
  Password = "pass"

  CommandACL = .bvfs_lsfiles, .bvfs_get_volumes, use, .bvfs_get_jobids, wait
  CommandACL = .bvfs_restore, .bvfs_cleanup, restore, run, gui, .jobs, quit
  CommandACL = .bvfs_version, .bvfs_delete_fileid, list

  # These commands are used only by the install-sap.sh test
  # procedure and can be commented out after the installation
  CommandACL = show, status

  ClientACL  = sap-fd
```

(continues on next page)

(continued from previous page)

```
  JobACL     = SAP-Backup, RestoreJob

  DirectoryAcl = *all*          # Available with 8.8
  UserIdAcl  = *all*            # Available with 8.8

  CatalogACL = *all*
  StorageACL = *all*
  FilesetACL = *all*
  PoolACL    = *all*
  WhereACL   = *all*
}
```

If you are upgrading from a previous version of the SAP plugin, you must check the `CommandACL` definition with what is defined above.

SAP HANA Cluster Installation Procedure

Before starting this procedure, you must collect the following information for each member of the cluster:

- Director IP address or DNS name (ex: 192.168.0.1, director.lan)
- Client IP address or DNS name (ex: 192.168.0.10, sap1.lan)
- Bacula Client name for each member of the cluster (ex: sap1-fd, sap1.lan)

Package Installation

The following packages are required on each member of the cluster:

- `bacula-enterprise-libs`
- `bacula-enterprise-client`
- `bacula-enterprise-sap-hana-plugin`

Packages may be installed via yum, zypper or rpm, depending on your Linux distribution.

By default, the directory `/opt/bacula/sap` should be accessible to the SAP programs that will communicate with Bacula via the `hdbbackint`. Usually, the unix account has the form `<SID>adm`. ex: if the instance name is HXE, the unix acount will be `hxeadm`.

```
hxehost:/tmp # id hxeadm
uid=1001(hxeadm) gid=79(sapsys) groups=79(sapsys),1000(hxeshm)

hxehost:/tmp # chown hxeadm:sapsys /opt/bacula/sap
```

Cluster Member Setup

The script /opt/bacula/scripts/intall-sap.sh helps to configure each member of the cluster and generates Bacula Director templates.

Program Installation

Specifying the install option of the install-sap.sh script will copy binaries inside the SAP file structure.

```
hxehost:/tmp # /opt/bacula/scripts/install-sap.sh install

Enter the unix SAP account name:
hxeadm

Enter the SAP instance name [HXE]:


Enter the SAP binary location [/usr/sap/HXE/SYS/global/hdb/opt]:

INFO: Link from /opt/bacula/bin/backint to /usr/sap/HXE/SYS/global/hdb/opt/
↪hdbbackint OK
```

Backint and Bacula Configuration

Specifying the configure option of the install-sap.sh script will configure the backint/hdbbackint parameter file, configure the Bacula Client configuration file and will generate a template for the Bacula Director.

```
hxehost:/tmp # /opt/bacula/scripts/install-sap.sh configure

Enter the unix sap account name [sap]:
hxeadm

Enter the Bacula Director Name [hxehost-dir]:
bacula-dir
INFO: Changing Director name from hxehost-dir to bacula-dir in bacula-fd.conf

Do you want to restart the bacula-fd service now? [y/N]:
y

Enter the Bacula Director Address [bacula-dir]:
192.168.0.46
Enter the Bacula Director Port [9101]:

Enter the Bacula FileDaemon name [hxehost-fd]:

Enter the Backup Job name [job.hxehost-fd.sap]:

Enter the Fileset name [fs.hxehost-fd.sap]:
```

(continues on next page)

(continued from previous page)

```
INFO: Creating configuration template for the Director
      /opt/bacula/sap/bacula-dir.conf.sample will help you to setup
      a Job with the Bacula Enterprise SAP Backint Plugin.

      The template can be included in your Director configuration and
      you need to review all items marked as "might need to be adjusted"
```

After this step, the Bacula Client configuration file should be properly configured to accept jobs from the Director. For example, if the Director is named “bacula-dir”:

```
hxehost:/tmp # cat /opt/bacula/etc/bacula-fd.conf
[...]
#
# List Directors who are permitted to contact this File daemon
#
Director {
  Name = bacula-dir
  Password = "zLyGZXcJLcKNKt7jvKCmHlUi08mhg/96prCBh0kavvyI"
}
[...]
```

The access to the Director via bconsole, a restricted console should be properly configured on the Client:

```
hxehost:/tmp # cat /opt/bacula/sap/bconsole.conf

# Bacula User Agent (or Console) Configuration File
#

Director {
  Name = "bacula-dir"
  DirPort = 9101
  Address = 192.168.0.46
  Password = "NotUsed"
}

Console {
  Name = "hxehost-fd"
  Password = "rTW383vwxUmsN8oQCOemwY7j4Yg5UQ/CExFUR5+fuxkH"
}
```

This restricted console can be shared between all members of the cluster.

The /opt/bacula/sap/backint.conf parameter file for hdbbackint should be configured with the following options:

```
hxehost:/tmp # cat /opt/bacula/sap/backint.conf

# Bacula SAP Backint configuration file
# This file can be linked to $ORACLE_HOME/dbs/init<ORACLE_SID>.utl
# or /usr/sap/<SID>/SYS/global/hdb/opt/hdbconfig/
```

(continues on next page)

(continued from previous page)

```
job="job.hxehost-fd.sap"
bconsole="/opt/bacula/bin/bconsole -n -c /opt/bacula/sap/bconsole.conf"
client="hxehost-fd"
waitjobcompletion=yes

# In cluster mode, all members of the cluster should be defined in
# the clientcluster variable separated with a comma
# clientcluster=hxehost-fd
```

You must edit the `/opt/bacula/sap/backint.conf` file to specify the Bacula Client name of the other cluster members via the `clientcluster` option:

```
clientcluster=hxehost-fd,sles12-fd
```

If one member of the cluster is not in the list, files that were backed up on this host will not be available to the cluster.

Director Configuration

The `install-sap.sh configure` script has created a Bacula Director template file with the following resources:

- Client
- Restricted Console
- Job
- Fileset

```
hxehost:/tmp # more /opt/bacula/sap/bacula-dir.conf.sample
#
# This template should help you to setup the SAP Backint
# plugin for hxehost-fd
#

Client {
  Name = "hxehost-fd"
  Description = "SAP Client"
  Address = "hxehost-fd"      # might need to be adjusted
  Password = "zLyGZXcJLcKNKt7jvKCmHlUi08mhg/96prCBh0kavvyI"

  Maximum Concurrent Jobs = 10

  File Retention = 5 years
  Job Retention  = 5 years
  Catalog = MyCatalog             # might need to be adjusted
}

Console {
  Name = "hxehost-fd"
  Description = "Console for SAP plugin"
```

(continues on next page)

(continued from previous page)

```
  Password = "rTW383vwxUmsN8oQCOemwY7j4Yg5UQ/CExFUR5+fuxkH"

  CommandACL = .bvfs_lsfiles, .bvfs_get_volumes, use, .bvfs_get_jobids, wait
  CommandACL = .bvfs_restore, .bvfs_cleanup, restore, run, gui, .jobs, quit
  CommandACL = .bvfs_version, list, .bvfs_delete_fileid

  CommandACL = status, show         # Can be removed after the installation

  ClientACL  = "hxehost-fd"
# ClientACL  = sap1-fd, sap2-fd     # Contain all members of the cluster

  FilesetACL = "fs.hxehost-fd.sap"

# FilesetACL = "Full Set"           # Should contain the RestoreFiles fileset

  JobACL     = "job.hxehost-fd.sap"
  JobACL     = RestoreFiles         # might need to be adjusted

  DirectoryAcl = *all*
  UserIdAcl  = *all*
  CatalogACL = *all*
  StorageACL = *all*
  PoolACL    = *all*
  WhereACL   = *all*
}

Fileset {
  Name = "fs.hxehost-fd.sap"
  Description = "SAP Backint Fileset"
  Include {
     Options {
        Signature = MD5
     }
     Plugin = "sap"                # might use ctrlfile=
  }
}

Job {
  Name = "job.hxehost-fd.sap"
  Description = "SAP Backint Job"
  Type = Backup
  JobDefs = "DefaultJob"          # might need to be adjusted

  Client = "hxehost-fd"
  Fileset = "fs.hxehost-fd.sap"

# Should be defined in DefaultJob
# Pool = Default                  # might need to be adjusted
# Storage = File
# Messages = Standard

  Maximum Concurrent Jobs = 10
```

(continues on next page)

(continued from previous page)

```
  Level = Incremental
}
```

The Bacula Director configuration template above must be adapted to your current Bacula setup. The following items must be reviewed to fit your setup:

- Client / Address
- Console / FilesetACL
- Console / ClientACL
- Job / JobDefs
- Job / Pool
- Job / Storage
- Job / Messages

The restricted Console can be used for all members of the cluster if the file `/opt/bacula/sap/bconsole.conf` is identical on all your systems.

The Bacula Director configuration must be copied into the `bacula-dir.conf` file, and the Director configuration must be reloaded.

Testing

As described in section *Testing backint.conf/init<SID>.utl Configuration*. The `install-sap.sh test` program must be executed on each cluster node.

Limitations

The Bacula Enterprise SAP Backint plugin version 6.4.6 and newer has the following limitations:

- no support for `volume` and `volume_online` for Backint option -t
- no support for `mount` and `dismount` for Backint option -f
- No support for database suspend and resume for Disk split
- No support for copying of snapshot or clone files
- To support RMAN incremental/differential levels, it is required to install the Bacula Enteprise Oracle SBT plugin.
- Empty jobs are not purged automatically after a backint delete command. The backint command "pruned" can list the jobs that can be deleted from the catalog.
- The estimate command is not supported by the SAP Backint Plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Problem Resolution

#NOTFOUND Error with HANA Cluster

If you have spurious `#NOTFOUND` errors, you must double check the `ClientCluster` backint configuration and the ClientACL list in the restricted Console. If one member of the cluster is missing, then the files backed up by the system will not be accessible to other member of the cluster.

Appendix

SAP Variable Limits

When configuring and operating Bacula Enterprise with SAP, some restrictions imposed by the SAP product need to be honored. The most relevant ones are collected in table below:

Table 32: SAP Variable Definition and Limitations

Entry	Description	Max Length Oracle/HANA
file	File, directory, raw device	255/512
dest dir	Directory used for restore	255/512
backupid	Backup ID, not containing any white spaces	16/32
userid	User ID	16/32

Sharepoint Plugin

Scope

Sharepoint plugin allows Bacula to backup Sharepoint site collections.

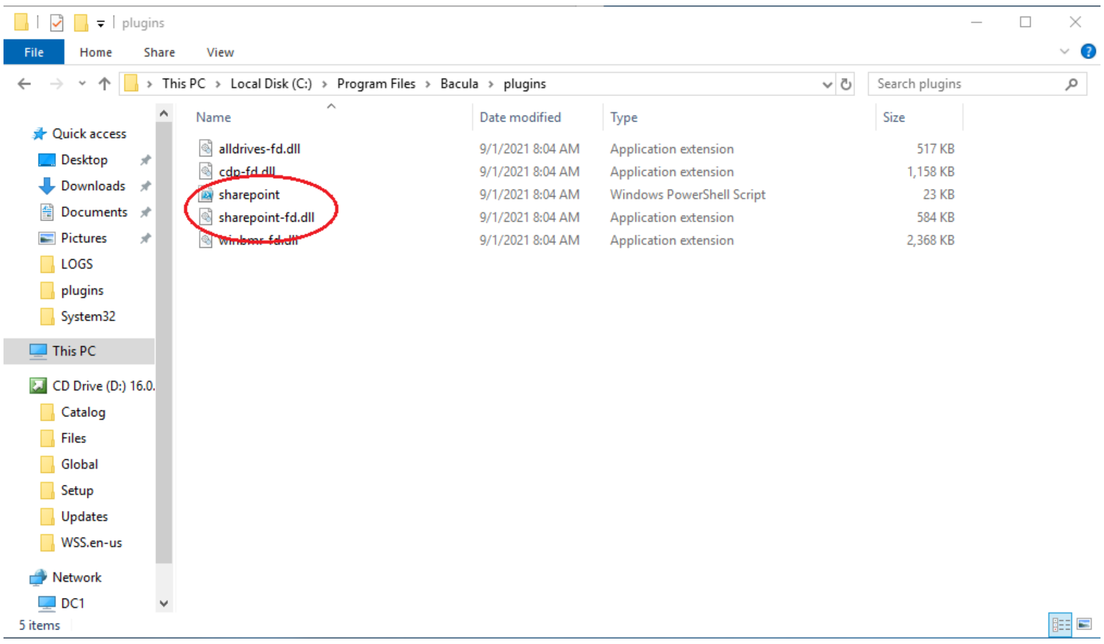
This documentation presents solutions for **Bacula Enterprise** 14.0.0 and higher, and is not applicable to prior versions of Bacula.

Features

- Backup site collection(s) based on url name
- Restore complete site collection(s) images

Installation

The Bacula File Daemon and the **Sharepoint plugin** need to be installed on the Sharepoint host server. The **Sharepoint plugin** windows installer will deploy required files within the Bacula File Daemon plugins directory.



Name	Date modified	Type	Size
alldrives-fd.dll	9/1/2021 8:04 AM	Application extension	517 KB
cdp-fd.dll	9/1/2021 8:04 AM	Application extension	1,158 KB
sharepoint	9/1/2021 8:04 AM	Windows PowerShell Script	23 KB
sharepoint-fd.dll	9/1/2021 8:04 AM	Application extension	584 KB
winbmr-fd.dll	9/1/2021 8:04 AM	Application extension	2,368 KB

To configure the Bacula File Daemon, refer to the general installation documentation.

Verify the correct installation of the FD and the **Sharepoint plugin** by running status client from the bconsole or from BWeb.

```
*status client=share1hyperv-fd
Connecting to Client share1hyperv-fd at 172.22.22.50:9102
share1hyperv-fd Version: 14.0.1 (21 January 2022)  VSS Linux Cross-compile␣
↪Win64
Daemon started 26-Jan-22 07:40. Jobs: run=1 running=0 max=20.
Microsoft Windows Server 2012 Standard Edition (build 9200), 64-bit
Heap: heap=4,984,832 smbytes=44,624 max_bytes=4,886,808 bufs=142 max_bufs=307
Sizes: boffset_t=8 size_t=8 debug=0 trace=1 mode=0,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL
Plugin: alldrives(1.2) cdp(0.1) sharepoint(0.1) winbmr(3.1.0)
```

Verify **sharepoint** is in the Plugin line.

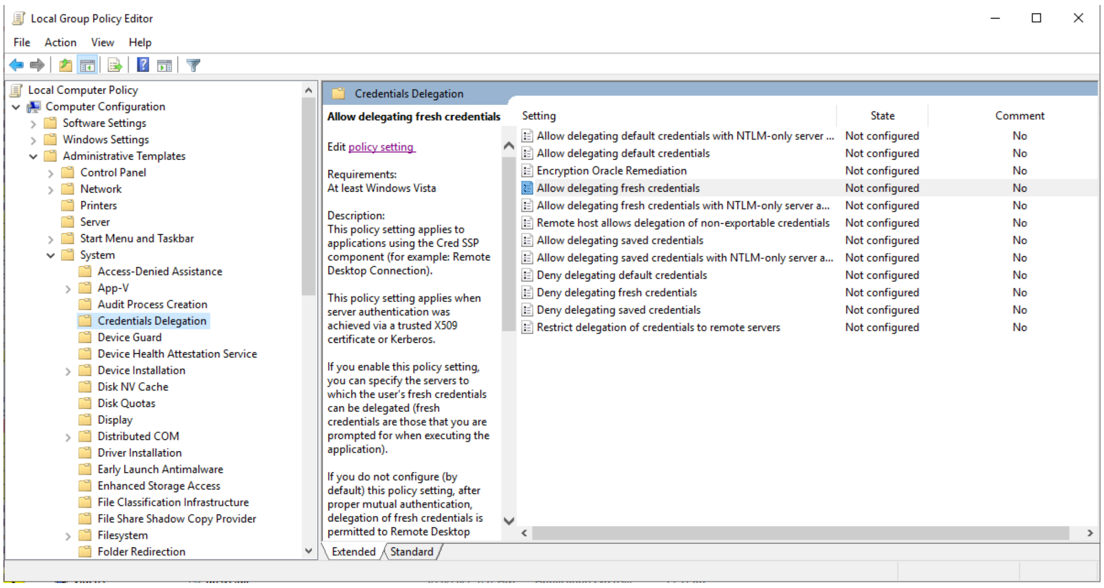
Configuration

Credentials Settings

Important: In order to access and backup Sharepoint site collections, the delegation of the User credentials must be enabled on the Sharepoint server and the Bacula file daemon must be logged as an authorized user within the Sharepoint server. Read access are sufficient to backup. Full read-write access is mandatory for restore.

Enable the delegation of the user credentials on the Hyper-V server

1. Run gpedit.msc (normally in C:\Windows\System32) on the Hyper-V server and look at the following policy: Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials.



Allow delegating fresh credentials

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop

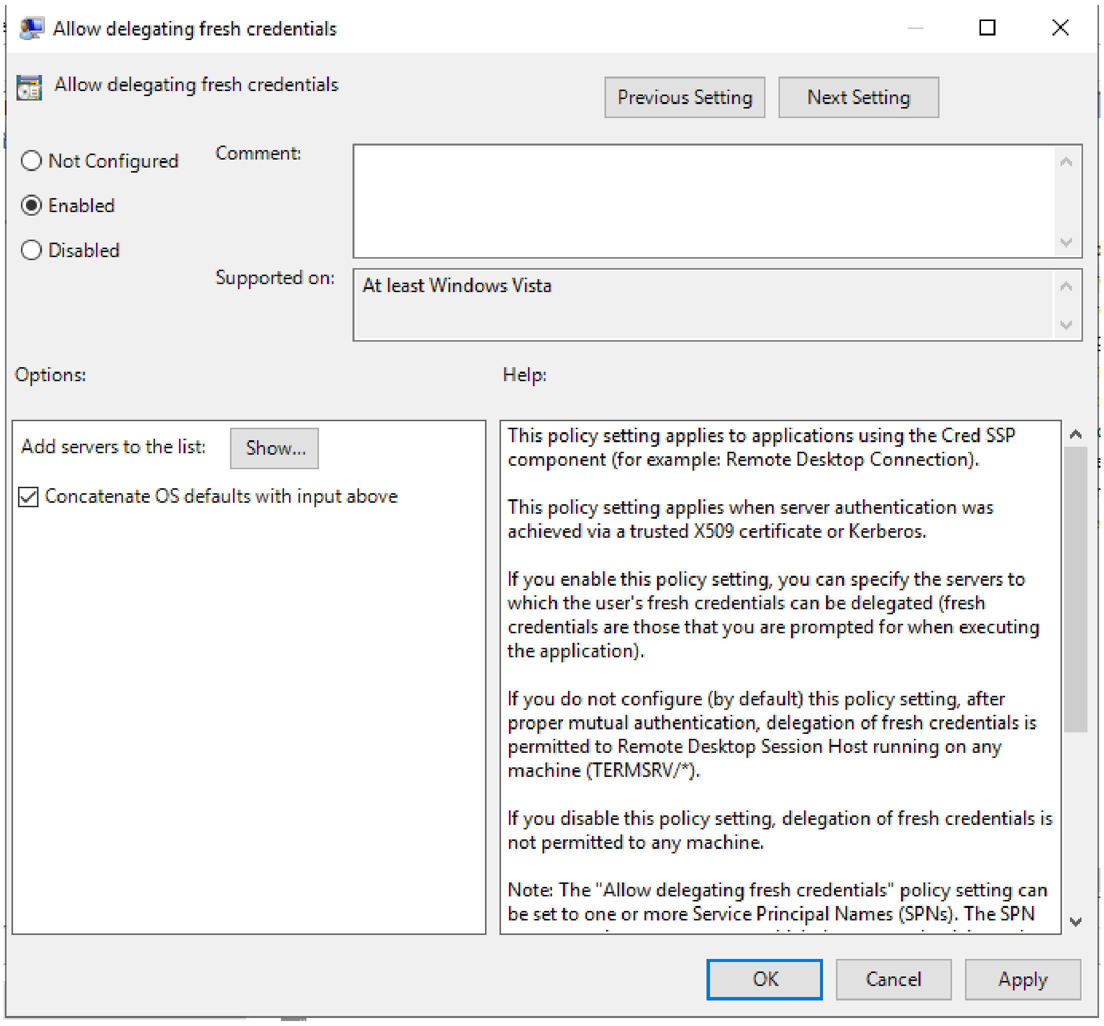
Setting	State	Comment
Allow delegating default credentials with NTLM-only server ...	Not configured	No
Allow delegating default credentials	Not configured	No
Encryption Oracle Remediation	Not configured	No
Allow delegating fresh credentials	Not configured	No
Allow delegating fresh credentials with NTLM-only server a...	Not configured	No
Remote host allows delegation of non-exportable credentials	Not configured	No
Allow delegating saved credentials	Not configured	No
Allow delegating saved credentials with NTLM-only server a...	Not configured	No
Deny delegating default credentials	Not configured	No
Deny delegating fresh credentials	Not configured	No
Deny delegating saved credentials	Not configured	No
Restrict delegation of credentials to remote servers	Not configured	No

2. Verify that it is enabled and configured with the WSMAN SPN appropriate for the target computer.

For example, for a target computer name "myserver.domain.com", the SPN can be one of the following: WSMAN//myserver.domain.com or WSMAN//*.domain.com. Introduce it in the "Add servers to the list": "Show" dialog box.

3. Finally run a powershell console on the Hyper-V server (normally in C:\Windows\Systeme32\WindowsPowerShellv1.0powershell.exe) and enter the following commands:

```
Enable-WSManCredSSP -Role Server -Force
Enable-WSManCredSSP -Role "Client" -DelegateComputer myserver.domain.com -
↪Force
```


Allow delegating fresh credentials

Allow delegating fresh credentials

Previous Setting | Next Setting

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Vista

Options:

Add servers to the list: Show...

☑ Concatenate OS defaults with input above

Help:

This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

This policy setting applies when server authentication was achieved via a trusted X509 certificate or Kerberos.

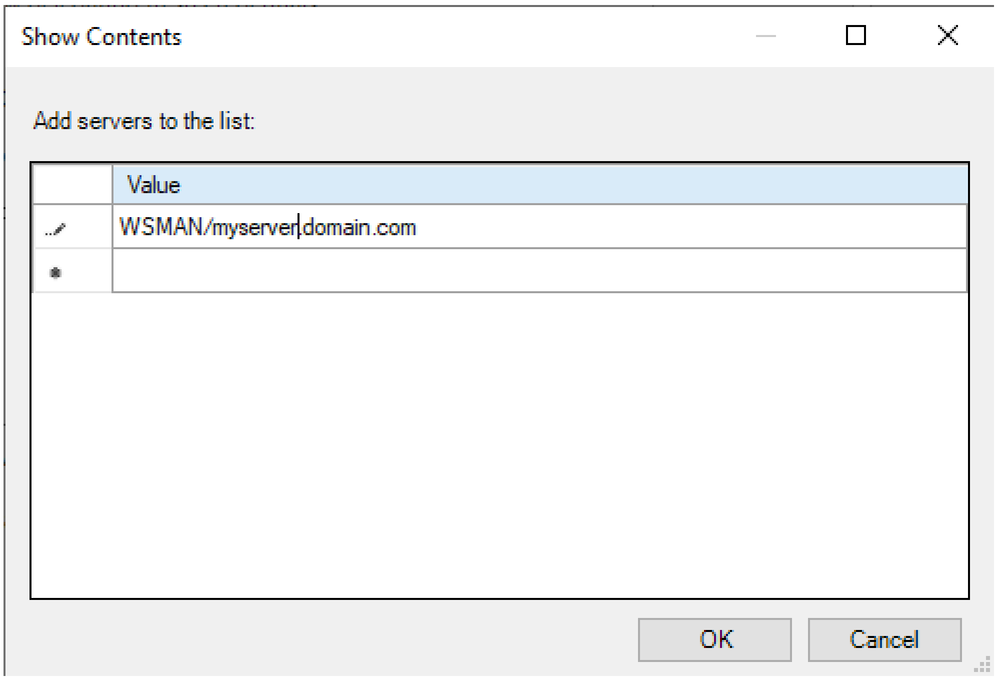
If you enable this policy setting, you can specify the servers to which the user's fresh credentials can be delegated (fresh credentials are those that you are prompted for when executing the application).

If you do not configure (by default) this policy setting, after proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop Session Host running on any machine (TERMSRV/*).

If you disable this policy setting, delegation of fresh credentials is not permitted to any machine.

Note: The "Allow delegating fresh credentials" policy setting can be set to one or more Service Principal Names (SPNs). The SPN

OK | Cancel | Apply


Show Contents

Add servers to the list:

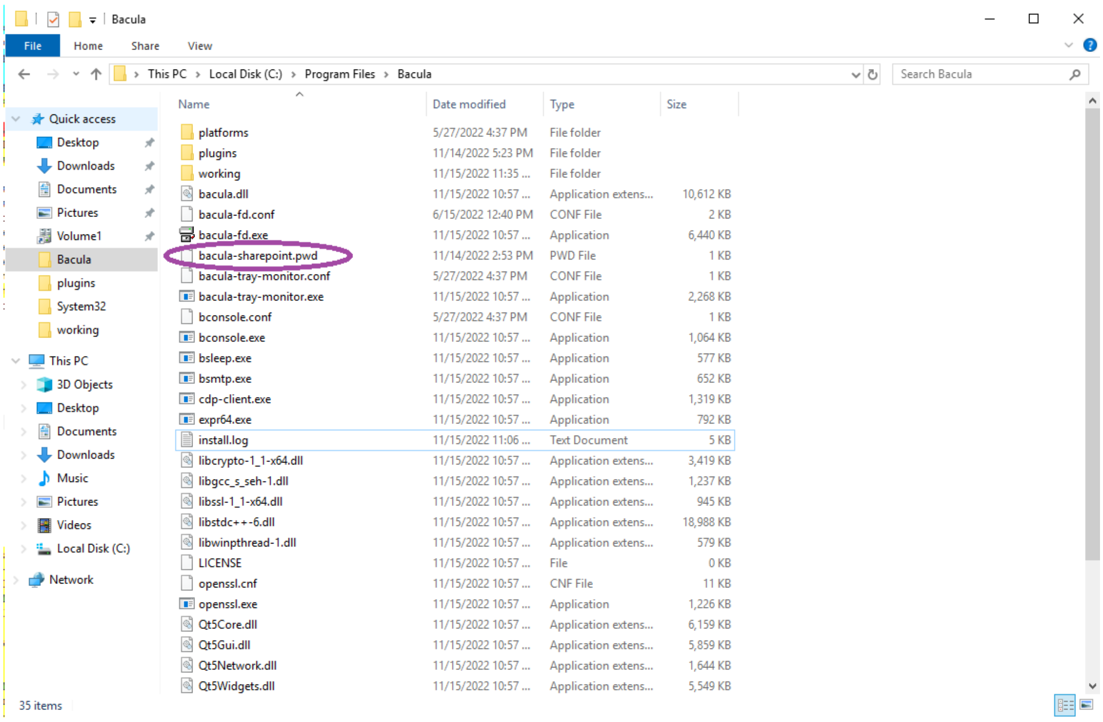
	Value
	WSMAN/myserver.domain.com
*	

OK | Cancel

Impersonification of the Sharepoint plugin

The impersonification of the **Sharepoint plugin** can be achieved in different ways.

1. Specify the user name and password locally on the sharepoint server. This is the **recommanded method**. In a `bacula-sharepoint.pwd` file, located by the `bacula-fd.conf` config file (typically C:\Program Files\Bacula).



Name	Date modified	Type	Size
platforms	5/27/2022 4:37 PM	File folder	
plugins	11/14/2022 5:23 PM	File folder	
working	11/15/2022 11:35 ...	File folder	
bacula.dll	11/15/2022 10:57 ...	Application extens...	10,612 KB
bacula-fd.conf	6/15/2022 12:40 PM	CONF File	2 KB
bacula-fd.exe	11/15/2022 10:57 ...	Application	6,440 KB
bacula-sharepoint.pwd	11/14/2022 2:53 PM	PWD File	1 KB
bacula-tray-monitor.conf	5/27/2022 4:37 PM	CONF File	1 KB
bacula-tray-monitor.exe	11/15/2022 10:57 ...	Application	2,268 KB
bconsole.conf	5/27/2022 4:37 PM	CONF File	1 KB
bconsole.exe	11/15/2022 10:57 ...	Application	1,064 KB
bsleep.exe	11/15/2022 10:57 ...	Application	577 KB
bsmtp.exe	11/15/2022 10:57 ...	Application	652 KB
cdp-client.exe	11/15/2022 10:57 ...	Application	1,319 KB
expr64.exe	11/15/2022 10:57 ...	Application	792 KB
install.log	11/15/2022 11:06 ...	Text Document	5 KB
libcrypto-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	3,419 KB
libgcc_s_seh-1.dll	11/15/2022 10:57 ...	Application extens...	1,237 KB
libssl-1_1-x64.dll	11/15/2022 10:57 ...	Application extens...	945 KB
libstdc++-6.dll	11/15/2022 10:57 ...	Application extens...	18,988 KB
libwinpthread-1.dll	11/15/2022 10:57 ...	Application extens...	579 KB
LICENSE	11/15/2022 10:57 ...	File	0 KB
openssl.cnf	11/15/2022 10:57 ...	CNF File	11 KB
openssl.exe	11/15/2022 10:57 ...	Application	1,226 KB
Qt5Core.dll	11/15/2022 10:57 ...	Application extens...	6,159 KB
Qt5Gui.dll	11/15/2022 10:57 ...	Application extens...	5,859 KB
Qt5Network.dll	11/15/2022 10:57 ...	Application extens...	1,644 KB
Qt5Widgets.dll	11/15/2022 10:57 ...	Application extens...	5,549 KB

`bacula-sharepoint.pwd` contains the user name followed by the user password, separated by a colon.

```
name@domain.com:mypassword
```

or

```
DOMAIN\name:mypassword
```

2. Impersonificate the **Sharepoint plugin** by passing user and password, as plugin options See *Job Configuration* for user_name and user_password options.

3. Manually change the Bacula File Daemon default login account.

   Log on to the Sharepoint server with administrative privileges.

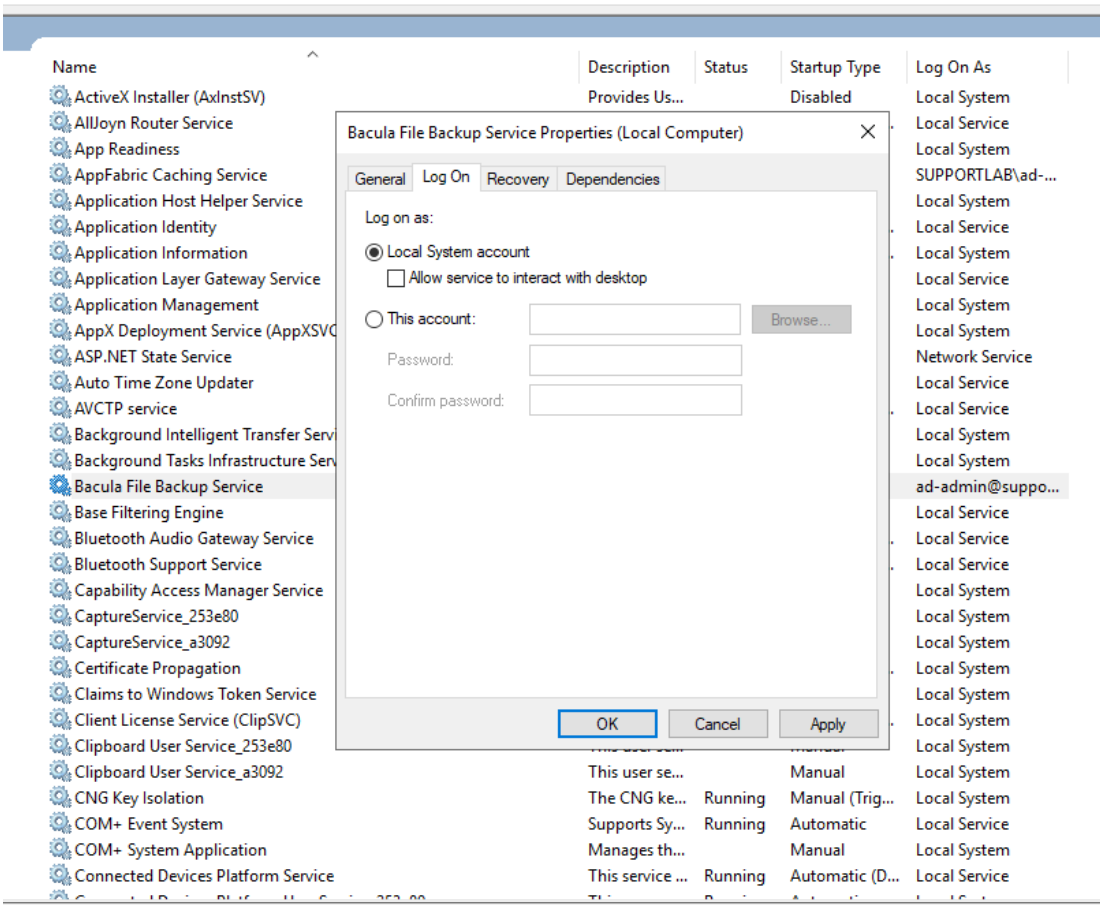
   From the Windows Start menu, type "Services" and press enter - this will show a list of the installed services.

   Find the Bacula File Backup Service and right-click on it. Choose Properties, select the Log On tab. By default it should look like this:

4. Add the default Bacula file daemon login as Sharepoint site Collection user

   By default the Bacula file daemon is logged as "NT Authority\System".

   With Sharepoint administrator privilege, do the following:



Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local System
AllJoyn Router Service				Local Service
App Readiness				Local System
AppFabric Caching Service				SUPPORTLAB\ad-...
Application Host Helper Service				Local System
Application Identity				Local Service
Application Information				Local System
Application Layer Gateway Service				Local Service
Application Management				Local System
AppX Deployment Service (AppXSVC				Local System
ASP.NET State Service				Network Service
Auto Time Zone Updater				Local Service
AVCTP service				Local Service
Background Intelligent Transfer Servi				Local System
Background Tasks Infrastructure Serv				Local System
Bacula File Backup Service				ad-admin@suppo...
Base Filtering Engine				Local Service
Bluetooth Audio Gateway Service				Local Service
Bluetooth Support Service				Local Service
Capability Access Manager Service				Local System
CaptureService_253e80				Local System
CaptureService_a3092				Local System
Certificate Propagation				Local System
Claims to Windows Token Service				Local System
Client License Service (ClipSVC)				Local System
Clipboard User Service_253e80				Local System
Clipboard User Service_a3092	This user se...		Manual	Local System
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local System
COM+ Event System	Supports Sy...	Running	Automatic	Local Service
COM+ System Application	Manages th...		Manual	Local System
Connected Devices Platform Service	This service ...	Running	Automatic (D...	Local Service

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

(●) Local System account

☐ Allow service to interact with desktop

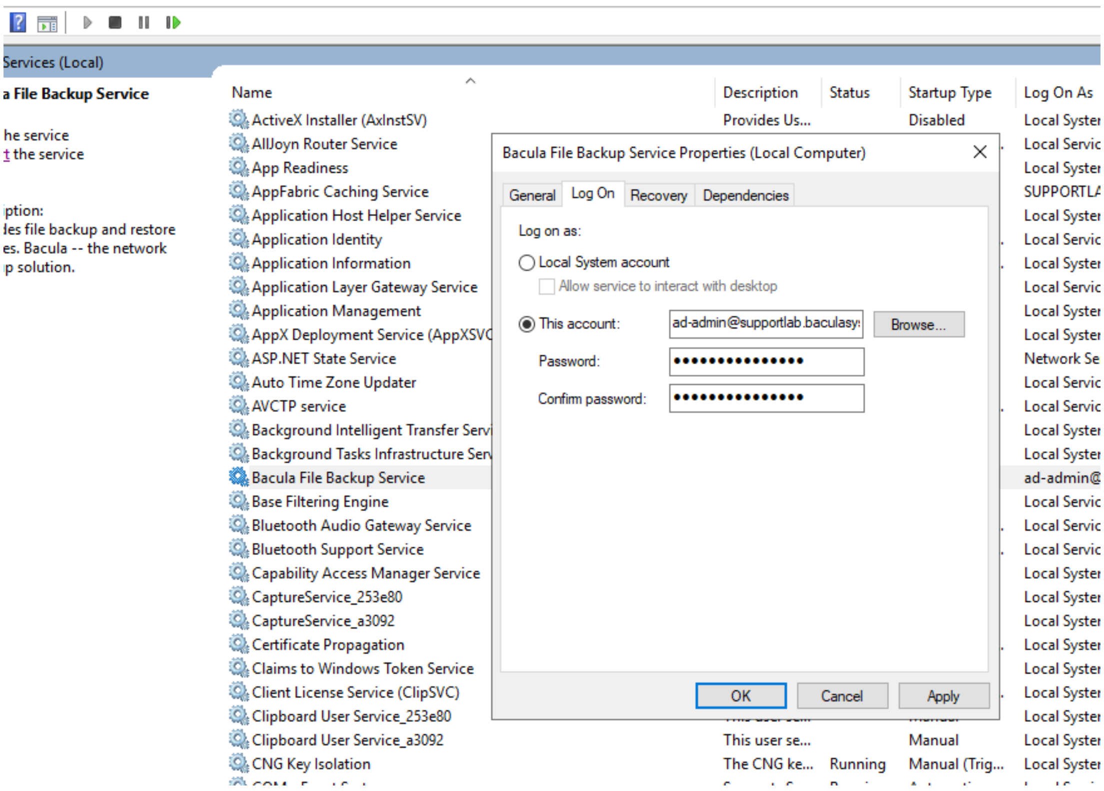
( ) This account: [ ] Browse...

Password: [ ]

Confirm password: [ ]

OK | Cancel | Apply

Fig. 68: Toggle the selection from “Local System account” to “This account”.
Enter the credentials of a Sharepoint administrator user (either read only or read-write). Click OK.



Services (Local)

a File Backup Service

he service
t the service

ption:
les file backup and restore
es. Bacula -- the network
p solution.

Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Disabled	Local Syster
AllJoyn Router Service				Local Servic
App Readiness				Local Syster
AppFabric Caching Service				SUPPORTLA
Application Host Helper Service				Local Syster
Application Identity				Local Servic
Application Information				Local Syster
Application Layer Gateway Service				Local Servic
Application Management				Local Syster
AppX Deployment Service (AppXSVC				Local Syster
ASP.NET State Service				Network Se
Auto Time Zone Updater				Local Servic
AVCTP service				Local Servic
Background Intelligent Transfer Servi				Local Syster
Background Tasks Infrastructure Serv				Local Syster
Bacula File Backup Service				ad-admin@
Base Filtering Engine				Local Servic
Bluetooth Audio Gateway Service				Local Servic
Bluetooth Support Service				Local Servic
Capability Access Manager Service				Local Syster
CaptureService_253e80				Local Syster
CaptureService_a3092				Local Syster
Certificate Propagation				Local Syster
Claims to Windows Token Service				Local Syster
Client License Service (ClipSVC)				Local Syster
Clipboard User Service_253e80				Local Syster
Clipboard User Service_a3092	This user se...		Manual	Local Syster
CNG Key Isolation	The CNG ke...	Running	Manual (Trig...	Local Syster

Bacula File Backup Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

( ) Local System account

[ ] Allow service to interact with desktop

(●) This account: ad-admin@supportlab.baculasy

Password: •••••••••••••••

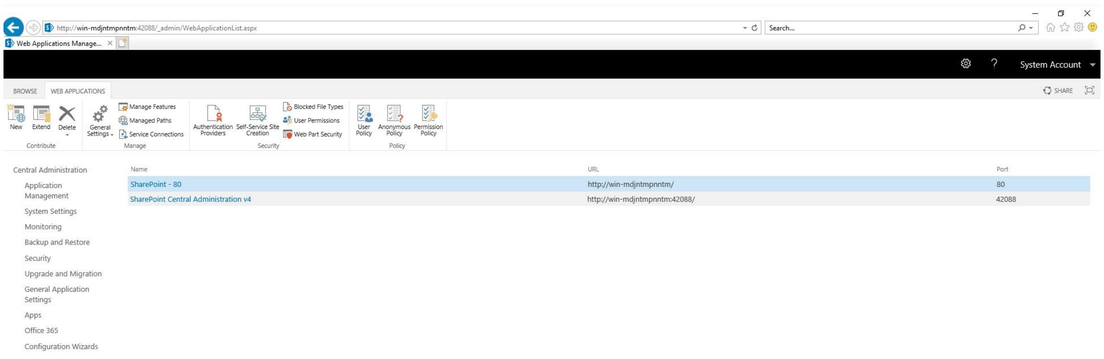
Confirm password: •••••••••••••••

Fig. 69: Right-click on the Bacula File Backup Service entry again, and click “Restart”, so the changes are applied.

From the Windows start menu, launch the SharePoint Central Administration.

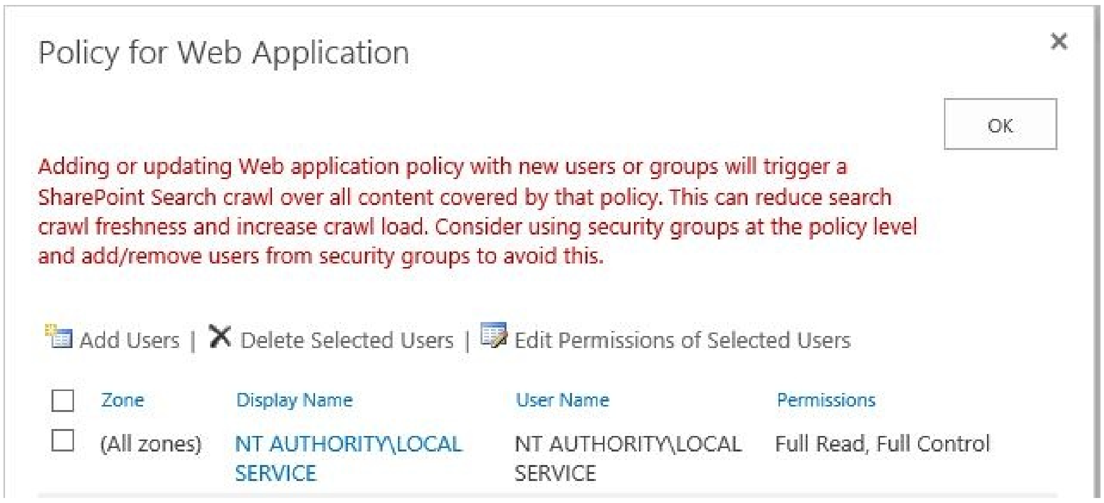
Under Application Management, select “Manage web applications” and you should see the list of available site collections.

For each backup target collection, select it, then choose User Policy from the ribbon menu.



Name	URL	Port
SharePoint - 80	http://win-mdjntmpnntm/	80
SharePoint Central Administration v4	http://win-mdjntmpnntm:42088/	42088

Fig. 70: The Policy for Web Application windows is displayed. Click “Add Users”.


Policy for Web Application
OK
Add Users | Delete Selected Users | Edit Permissions of Selected Users

Adding or updating Web application policy with new users or groups will trigger a SharePoint Search crawl over all content covered by that policy. This can reduce search crawl freshness and increase crawl load. Consider using security groups at the policy level and add/remove users from security groups to avoid this.

Zone	Display Name	User Name	Permissions
(All zones)	NT AUTHORITY\LOCAL SERVICE	NT AUTHORITY\LOCAL SERVICE	Full Read, Full Control

Fig. 71: Add “NT AUTHORITY\SYSTEM” in the Users box and check “Full Read” for backup only, or “Full Control” for backup and restore.

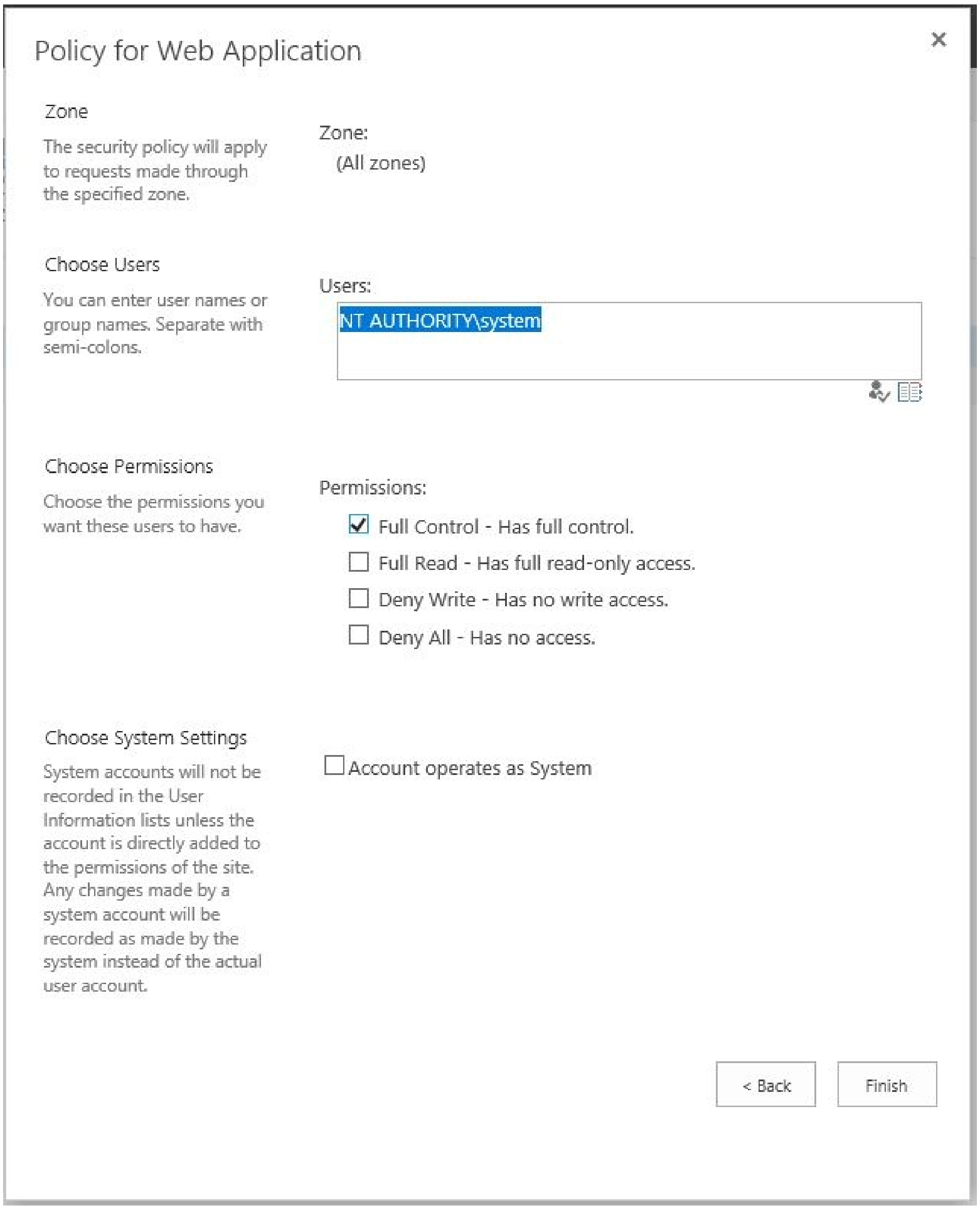
Press the finish button.

Job Configuration

Once the Bacula File Daemon and the **Sharepoint plugin** are correctly installed and configured, setting a backup job up is as simple as adding the job and the fileset within the Bacula Director configuration file.

Important: The `Enable VSS` parameter must be set to `no` in the Fileset (see examples below).

The following plugin options are supported :


Policy for Web Application

Zone

The security policy will apply to requests made through the specified zone.

Zone:
(All zones)

Choose Users

You can enter user names or group names. Separate with semi-colons.

Users:
NT AUTHORITY\system

Choose Permissions

Choose the permissions you want these users to have.

Permissions:

- [✓] Full Control - Has full control.
- [ ] Full Read - Has full read-only access.
- [ ] Deny Write - Has no write access.
- [ ] Deny All - Has no access.

Choose System Settings

System accounts will not be recorded in the User Information lists unless the account is directly added to the permissions of the site. Any changes made by a system account will be recorded as made by the system instead of the actual user account.

- [ ] Account operates as System

Fig. 72: In addition, you also need to grant access to the Sharepoint MSSQL databases to Bacula File Daemon.
From the Windows start menu, launch the SQL server Management Studio and log in with MSSQL administrator privileges.

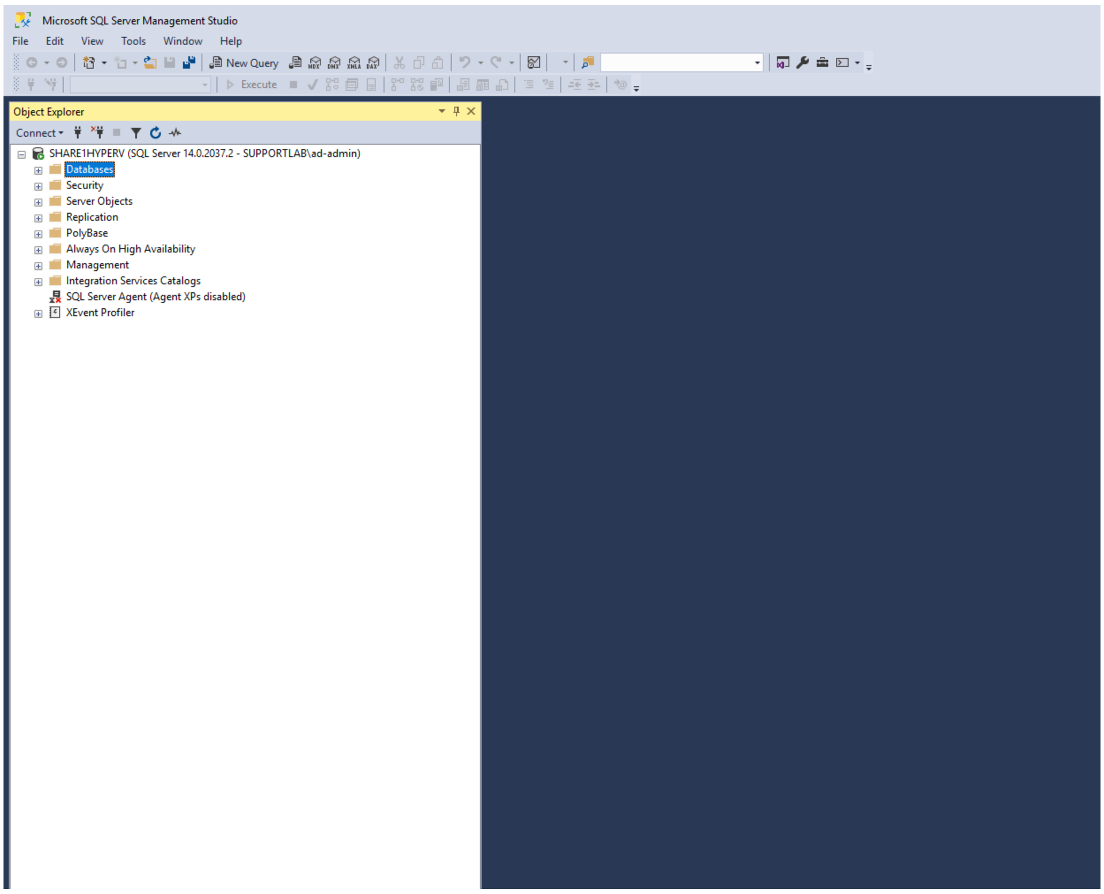
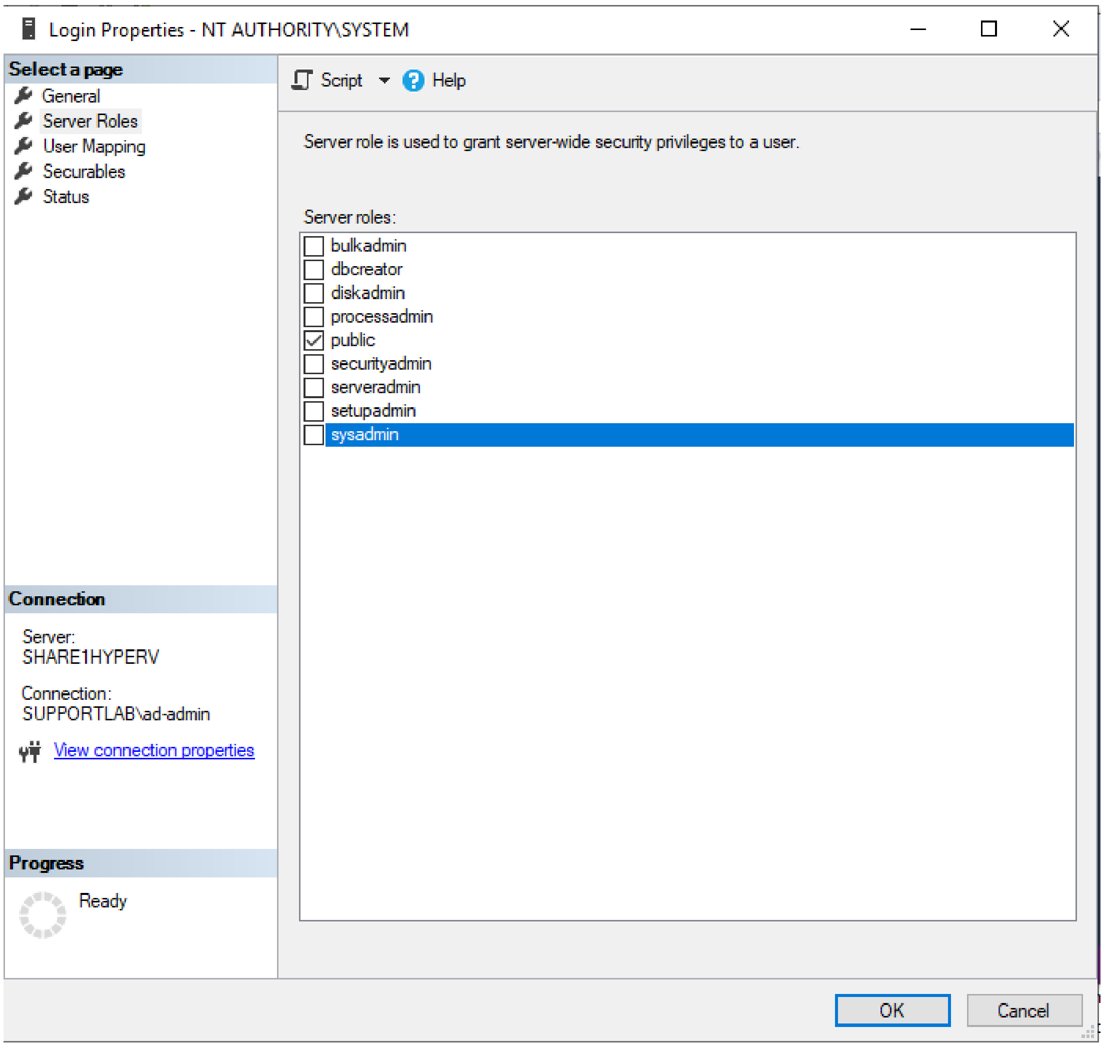

Microsoft SQL Server Management Studio
File
Edit
View
Tools
Window
Help
New Query
Execute
Object Explorer
Connect
SHARE1HYPERV (SQL Server 14.0.2037.2 - SUPPORTLAB\ad-admin)
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler

Fig. 73: Under Security/Logins, find “NT AUTHORITY\SYSTEM”, right click on it and select properties. Select the Server Roles page, check sysadmin.


Login Properties - NT AUTHORITY\SYSTEM

Select a page
- General
- Server Roles
- User Mapping
- Securables
- Status

Script ▾ Help

Server role is used to grant server-wide security privileges to a user.

Server roles:
- [ ] bulkadmin
- [ ] dbcreator
- [ ] diskadmin
- [ ] processadmin
- [✓] public
- [ ] securityadmin
- [ ] serveradmin
- [ ] setupadmin
- [ ] sysadmin

Connection

Server:
SHARE1HYPERV

Connection:
SUPPORTLAB\ad-admin

View connection properties

Progress

Ready

OK Cancel

Name	Status	Default	Description
include	Optiona	Include all (*)	a Unix shell-style wildcards pattern for including a site, bases on its URL
exclude	Optiona	Exclude none	a Unix shell-style wildcards pattern for excluding a site, bases on its URL
tmp_d	Optiona	Working	locate the working folder for the **Sharepoint plugin**. Make sure there's enough space in this location to create sites snapshots and exports.
user_n	Optiona	None	the user name that will run the backup/restore operation. This is **not the recommended method**. The user name can be specified locally on the hyper-v node in a `bacula-sp.usr` file located in the FD plugins folder.
user_p	Optiona	None	the user password that will run the backup/restore operation. This is **not the recommended method**. The user password can be specified locally on the hyper-v node in a `bacula-sp.pwd` file located in the fd plugins folder.

Examples

Example 1: Backup All Sites Using Bacula's Default Working Directory

```
Job {
  Name = "SharepointBackupAll"
  Type = Backup
  Client= w2019-sp01-fd
  Fileset="SimplestSharepointFileset"
  Storage = File
  Messages = Standard
  Pool = Default
}

Fileset {
  Name = "SimplestSharepointFileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "sharepoint:"
  }
}
```

Example 2: Backup Only Sites Which URLs Contain «Linux», Using Bacula's Default Working Directory

```
Job {
  Name = "SharepointBackupOnlyLinux"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="LinuxSharepointFileset"
  Storage = File
  Messages = Standard
  Pool = Default
}
Fileset {
  Name = "LinuxSharepointFileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "sharepoint: include=\"*Linux*\""
  }
}
```

Example 3: Backup Any Site Having «Windows» in Its Name, Using a Custom Working Directory

```
Job {
  Name = "SharepointBackupOnlyWindowsOnF"
  Type = Backup
  Client= w2019-hv01-fd
  Fileset="WindowsOnSiteFileset"
  Storage = File
  Messages = Standard
  Pool = Default
}
Fileset {
  Name = "WindowsOnSiteFileset"
  Enable VSS = no
  Include {
    Options {
      signature=MD5
    }
    Plugin = "sharepoint: include=\"*Windows*\" tmp_dir=\"F:/backup\""
  }
}
```

Operations

Backup

The Sharepoint server backed up files will appear in a **bconsole** or prefixed with `/@SHAREPOINT/` .

Typically, a Site Collection backup data is organized as follows:

```
/@SHAREPOINT/a4d05f41-e9e0-43f6-9478-d8adb337cf6c_sharepoint_backup.
2022-01-26_07.14.22_14.bak /@SHAREPOINT/a4d05f41-e9e0-43f6-9478-d8adb337cf6c_sharepoint_backup.
2022-01-26_07.14.22_14.xml
```

Incremental-Differential backups: the Site Collection backup doesn't support incremental-differential backup.

Restore

Although it's possible to cherry pick the backed up files, it's recommended to select the Site Collection UID folder for a single Site Collection in order.

When restoring a Site Collection on a server that already hosts a VM with the same name, the restored Site Collection name is post-fixed with "Restore.<date>_<time>".

Best practices

While it is technically possible to backup multiple Site Collections in one Bacula **Sharepoint plugin** backup job, this is not necessarily the best way to perform backup. It is strongly recommended that one backup Job is created for each Site Collection being backed up for the following reasons:

- If one of your Site Collection fails to backup in a "multi-SiteCollections" backup job, the main Bacula job will terminate **Backup OK – with warnings**. The catalog's "JobStatus" for jobs that terminate **Backup OK** and **Backup OK – with warnings** are not differentiated. They are both 'T' so this means that you will have to carefully monitor your backup jobs logs in case some Site Collections backups fail and pay attention to the "JobErrors" field.
- There is a **Sharepoint plugin** option `abort_on_error` but, if you use this option, and Site Collection number 11 in a list of 50 Site Collections fails, then the whole job will fail, and VMs 12-50 will not be backed up during that job's run.
- A 1:1 (one VM per job) configuration means that the `abort_on_error` option will make more sense to enable in each job, so you will immediately know when a VM fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the catalog for the job.
- In the case describing the 50 Site Collections, without a 1:1 configuration, there is no way to easily re-run a backup of just the 1 Site Collection that failed to backup.
- With a 1:1 configuration, re-running a specific Site Collections backup job is simple to do after the cause of the failure is investigated.
- With a 1:1 configuration, job metrics will have more meaning because each Site Collection will be one job, and you will know to expect a specific number of jobs each night with each one representing one machine, or Site Collection.

- With a "multi-Site Collections" configuration, each Site Collection will be backed up "serially", one at a time, disk by disk, Site Collection by Site Collection. A 1:1 configuration will allow several Site Collection backups to be run concurrently which will reduce the overall time to perform the Site Collections backups (paying close attention to SD and Sharepoint resources of course, and adjusting the number of concurrent jobs accordingly).

Limitations

- Sharepoint site collection(s) backup doesn't support incremental and differential.
- Single item restore is not supported.
- It's recommended to restore full site collection(s).
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

VSS Plugin

Backing up Windows Machines

In general, there are three distinct ways Windows machines can be backed up with **Bacula Enterprise**:

- Backup Windows as an image. This requires shutting down the Windows system, rebooting a Linux system (CDROM or USB) and then backing up the raw Windows partition or partitions using a Linux File daemon running on the Linux system. Since this method requires shutting down the Windows system, booting a Linux system, then backing up the raw partitions, it is rather labor intensive. Each backup is the full size of the Windows partition or partitions (disks). In addition, you cannot do file level restores; only the full image can be restored. The advantage is that a restore is quite fast. We will not explore this option any more in this white paper.
- Normal Bacula Windows backup without the VSS plugin. This is a standard backup where you back up everything on every disk with the exception of temporary files. This backup uses VSS (default Bacula option), but not the VSS plugin. Bacula has worked in this manner for many years, the code is stable and many users have restored user files and done many successful Bare Metal Recoveries. In fact, Bacula Systems has a separate WinBMR tool that automates this technique for Bare Metal Recovery.

  This normal backup must be done **without** the Bacula VSS plugin, although Bacula will use Windows VSS APIs (but not the VSS plugin) to create a snapshot of the filesystem that is then backed up. Normally such a backup includes all the operating system files.

  The disadvantage of this normal backup is that you cannot easily restore individual components of the System State as you can with the VSS plugin. However, you can do two kinds of restores:

1. A restore of standard user files or any system files that are not in use by the OS (most are in use and cannot be restored while the system is running).
2. A Bare Metal Recovery.

   During a Bare Metal Recovery, you can restore the whole system including all the System State files. The disadvantage compared to the VSS plugin restore is that you must restore the whole system rather than individual components. Note: it is also possible to restore any subset of the files rather than all files, so if you have expert knowledge of System State files, you could potentially restore only the System State or any other system component. In practice very few people would be inclined to do this.

- Backup the System State and / or several other system elements such as MSSQL, Exchange, etc. These are described in detail in this white paper. The difference between this kind of a backup and the previous item described above is that this approach uses the Bacula Enterprise VSS plugin.

  A backup that uses the Bacula Enterprise VSS plugin should be a separate Job from a normal user file backup (see above). The advantage of this kind of backup is that if your system state is damaged, you can restore the particular element that is broken or the full system state while the system is running then reboot to complete the process.

  The disadvantage of this kind of backup is that it cannot be used for a Bare Metal Recovery. This is because any backup containing data written by the VSS plugin requires VSS for its restores, and Microsoft does include neither the VSS subsystem in WinPE, which is used for bare metal recovery, nor can the recovery environment handle any other VSS-aware application like MSSQL Server or Exchange, since those can not be available and configured during a Bare Metal restore.
- The VSS plugin is compatible with Copy/Migration jobs. Please read the CopyMigrationJob-sReplication for more information.

Summary: Bacula provides several methods of backing up and restoring Windows filesystems. As mentioned above, the most common is simply to backup everything *without* the VSS plugin. Normal restores can then be done on any file that is not locked by the system (mostly user files) while the system is running, or you can shutdown the system and do a Bare Metal Recovery where normally, all files are restored.

The rest of this white paper will discuss the Bacula Enterprise VSS plugin for **Bacula Enterprise** version 6 and newer and how to use it.

Bacula Windows VSS Plugin

We provide a single plugin named **vss-fd.dll** that permits to back up a number of different components on Windows machines. Components that are supported currently include:

- **System State writers**
  - Registry
  - Event Logs and Performance Counters
  - COM+ REGDB (COM Registration Database)
  - System (Systems files – most of what is under c:/windows and more)
  - WMI (Windows Management and Instrumentation)
  - NTDS (Active Directory)
  - Task Scheduler
  - Dhcp Jet (DHCP status and leases)

  - FSRM (File Server Resource Manager) data

- **Exchange**. Please note, there is an older Bacula community Exchange plugin that works entirely differently (it uses an old Exchange API rather than VSS). This old community Exchange plugin (**exchange-fd.dll**) is not discussed here.
- **MSSQL databases** Backing up and restoring MSSQL databases works very well for Full and Differential backups. It is not possible to do Incremental backups because backing up an MSSQL database uses block differencing, which requires Differential backups.

  Note that experience shows that large and busy MSSQL Server instances may often not be able to be backed up using VSS. In those situations, the **Bacula Systems** SQL Server VDI Plugin will provide a more reliable solution.

Each of the above specified Microsoft components can be backed up by specifying a different plugin command line within the Bacula Fileset. All specifications must start with **vss:** and be followed with a keyword which indicates the component, such as **/@SYSTEMSTATE/** (see below).

To activate each component you use the following:

- **System State writers**

  ```
  Plugin = "vss:/@SYSTEMSTATE/"
  ```

  Note, exactly which subcomponents (writers) will be backed up depends on which ones you have enabled. For example, on a standard default system only COM+ REGDB, System State, and WMI are enabled. The plugin automatically finds all subcomponents (writers) that are enabled and will list them in the Job report.

  The Windows ASR Writer is automatically disabled by the plugin because Bacula Systems uses its own Bare Metal Recovery techniques.

- **Exchange**

  ```
  Plugin = "vss:/@EXCHANGE/"
  ```

  The Exchange writer supports Full, Differential, and Incremental backups. For more details on the Exchange component of the VSS plugin, please see our white paper for Exchange with Bacula Enterprise version 6.0.

- **MSSQL databases**

  ```
  Plugin = "vss:/@MSSQL/"
  ```

  Note, MSSQL backup works only for Full and Differential backups.

The plugin directives must be specified exactly as shown above. A Job may have one or more of the **vss** plugin components specified.

You must ensure that the **vss-fd.dll** plugin is in the plugins directory on the FD doing the backup (done by default with the installer), and that the **Plugin Directory** directive line is present and enabled in the FD's configuration file bacula-fd.conf.

If the plugin is loaded can be verified by checking the `status client` output with a debug level set, which shows the available plugins. An example session should look like this, where the line starting with "Plugin" shows which plugins are available, including the **vss-fd** one:

```
*setdebug level=1 client=wsb-exch10-fd
Connecting to Client wsb-exch10-fd at wsb-exch10:9102
2000 OK setdebug=1 trace=1 hangup=0
```

(continues on next page)

(continued from previous page)

```
*status client=wsb-exch10-fd
Connecting to Client wsb-exch10-fd at wsb-exch10:9102

wsb-exch10-fd Version: 6.0.0.5 (06 Mar 2012)  VSS Linux
Cross-compile Win64
Daemon started 19-Mar-12 11:44. Jobs: run=0 running=0.
Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1
(build 7601), 64-bit
VSS enabled, Priv 0x22f
APIs=OPT,ATP,LPV,CFA,CFW,
 WUL,WMKD,GFAA,GFAW,GFAEA,GFAEW,SFAA,SFAW,BR,BW,SPSP,
 WC2MB,MB2WC,FFFA,FFFW,FNFA,FNFW,SCDA,SCDW,
 GCDA,GCDW,GVPNW,GVNFVMPW
 Heap: heap=0 smbytes=23,678 max_bytes=23,678 bufs=75 max_bufs=75
 Sizeof: boffset_t=8 size_t=8 debug=1 trace=1 mode=0,2010 bwlimit=0kB/s
Plugin: alldrives-fd.dll delta-fd.dll vss-fd.dll

Running Jobs:
Director connected at: 19-Mar-12 11:45
No Jobs running.
====

Terminated Jobs:
====
*setdebug level=0 client=wsb-exch10-fd
Connecting to Client wsb-exch10-fd at wsb-exch10:9102
2000 OK setdebug=0 trace=1 hangup=0
```

Newer versions of **Bacula Enterprise** report the plugins loaded even without the debug level being set. With the debug level set, they will report the plugins internal version number, like this:

```
wsb-master-fd Version: 6.2.7 (08 July 2013)  VSS Linux Cross-compile Win64
Daemon started 12-Sep-13 13:09. Jobs: run=5 running=0.
Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1
(build 7601), 64-bit
VSS enabled, Priv 0x73f
APIs=OPT,ATP,LPV,CFA,CFW,
 WUL,WMKD,GFAA,GFAW,GFAEA,GFAEW,SFAA,SFAW,BR,BW,SPSP,
 WC2MB,MB2WC,FFFA,FFFW,FNFA,FNFW,SCDA,SCDW,
 GCDA,GCDW,GVPNW,GVNFVMPW,LZO
 Heap: heap=0 smbytes=154,611 max_bytes=40,440,895 bufs=112 max_bufs=59,514
 Sizes: boffset_t=8 size_t=8 debug=1 trace=1 mode=0,2010 bwlimit=0kB/s
 Plugin: alldrives-fd.dll(1.1) delta-fd.dll(1) vss-fd.dll(1)
winbmr-fd.dll(3.0.12)
```

The details of doing backups and restores with the **vss** Exchange component are discussed in a separate White Paper entitled **ExchangePlugin-6.0** (which covers any **Bacula Enterprise** version 6 and newer).

Please take note of the comments above concerning the difference between this VSS plugin (**vss-fd.dll**) with the Exchange component selected and the older community Exchange plugin (**exchange-fd.dll**).

BMR and VSS

The VSS plugin will not work correctly during a Bare Metal Recovery because Microsoft does not include the VSS subsystem in WinPE. As a consequence, any backup you do with the **vss** plugin cannot be used for a bare metal recovery. To have a good backup for bare metal recovery restore, you must run the Bacula FD **without** the **Plugin =** directive in your Fileset (i. e. the plugin must be turned off).

Backup

If everything is set up correctly as above then the backup will include the system state. The system state files backed up will appear in a **bconsole** or **bat** restore like:

```
/@SYSTEMSTATE/
/@SYSTEMSTATE/Registry Writer/
/@SYSTEMSTATE/COM+ REGDB Writer/
etc
```

Only a backup of all the system state subcomponents is supported. That is it is not possible to just back up only one subcomponent such as the Registry or NTDS alone. In almost all cases a complete backup is a good idea anyway as most of the components are interconnected in some way.

Both Differential and Incremental backups are supported.

Windows does not update the time and date of modification on all system files, so if you want correct Differential and Incremental backups, you **must** use the Bacula **Accurate** option. If you do not use this option, the plugin will print a warning message.

The Full backup size varies according to your installation. Backup sizes of a few GB under Windows Server 2003 and up to 20 GB under Windows 2008 are typical, mostly because of the "System" writer. The actual size depends on how many Windows services (writers) are enabled.

The system state component automatically respects all the excludes present in the Windows **FilesNotToBackup** registry key, which includes things such as %TEMP%, `pagefile.sys`, , etc. Each plugin component may automatically specify additional files to exclude, e. g. the VSS Registry Writer will tell Bacula to not back up the registry hives under `C: WINDOWS system32 config` because they are backed up as part of the system state backup.

Restore

In most cases a restore of all the backed up system state subcomponents is recommended. Individual writers can be selected for restore provided the whole writer is selected. To restore just the Registry, you would need to do **mark Registry*** to mark the Registry writer and everything under it.

Restoring anything less than a whole writer will cause the restore Job to fail with an error message.

When doing a restore, you must do a restore of the current system (i.e. option 5 on the restore menu) or a restore to a point in time. You should not attempt to select individual JobIds for restore, because to do a proper restore, one must choose all jobs that were previously run from the Full Job up to the point in time. If any JobIds or files are skipped as might happen if you choose the JobIds yourself, the restore may fail.

If you want to restore multiple subcomponents, we recommend that you do them all together or do a restore of the whole System State component.

If you are trying to restore a writer (component) to a different machine, please be aware that Bacula Systems does not support this, because as far as we can tell, it is not supported by Microsoft. This seems to be because most writers are integrated with or rely on Active Directory, and so if you do not have exactly the same Active Directory environment on the alternate machine, the restore is not likely to work correctly.

Bacula Systems do not yet support restoring any component that is configured in a Microsoft cluster. You are free to try it, but according to the Microsoft documentation, there are a good number of restrictions and requirement for doing restores in clustered environments. If you succeed in backing up and restoring in a clustered environment, please let us know as we will be very interested in your results.

Reboot Required After Restore

After the restore of certain System State components, you must reboot before the changes can be properly applied. Bacula will print a message in the Job report indicating when this is necessary. The reboot is required because some VSS components will restore files that are currently in use. However, since files that are in use cannot be deleted, modified, or replaced, a reboot will be required to complete the restore by moving those restored files in place. Once you have restored such System State component, you must reboot your machine.

Until the reboot, the system will be in an unstable state, so please do not forget this step. In particular, starting another restore of VSS data after a restore requiring a reboot without first rebooting will cause any subsequent VSS plugin restore Job to fail.

Restore to Alternate Locations

Restore to alternate locations is not implemented in the **vss** plugin, nor is file name mangling using regular expressions.

Restoring Active Directory

To restore Active Directory, the system will need to be booted into Directory Services Restore Mode, an option at Windows boot time. Consequently, restoring Active Directory is a bit complicated. You might want to read Microsoft's comments about doing restores (note: their comments about Backup tools do not, in general, apply to Bacula).

technet.microsoft.com/en-us/library/cc961934.aspx

Bacula Systems can provide more detailed help regarding restoration of NTDS data; please contact our support team if you require that advice.

You need to be aware of Directory Services Restore Mode, the difference between authoritative and non-autoritative AD restores.

One important thing to consider is that, if you've got more than one AD server in your domain, you probably never have to go through the recovery procedure in practice, as it's easier to just set up a fresh server rather than doing an AD recovery, make the new server an AD server, and let it pull its data through replication from the existing servers. In other words, using backed up AD data should be considered a last-resort, disaster-recovery only approach.

Example

Suppose you have the following backup Fileset:

```
@SYSTEMSTATE/
  System Writer/
    instance_{GUID}
    System Files/
  Registry Writer/
    instance_{GUID}
    Registry/
  COM+ REGDB Writer/
    instance_{GUID}
    COM+ REGDB/
  NTDS/
    instance_{GUID}
    ntds/
```

If only the Registry needs to be restored, then you could use the following commands in **bconsole**:

```
cd @SYSTEMSTATE
mark "Registry Writer"
```

Windows Plugin Items to Note

- Reboot required after a plugin restore In general after any VSS plugin is used to restore certain components, you may need to reboot the system. This is required because in-use files cannot be replaced during restore time, so they are noted in the registry and replaced when the system reboots.
- Longer boot time after a restore After a System State restore, a reboot will generally take longer than normal because the pre-boot process must delete the old files and move the newly restored files into their final place prior to actually starting the OS. For a large system, this can take quite a long time (we have seen up to 20 minutes).
- One file from each drive needed by the plugin must be backed up At least one file from each drive that will be needed by the plugin must have a regular file that is marked for backup. This is to ensure that the main Bacula code does a snapshot of all the required drives.

  Available as of version 8.0.1

  An alternative approach is to use the `alldrives` plugin, or, with **Bacula Enterprise** version 8.0.1 or newer, the **File=/** semantic in the file set.

  Available as of version 12.6.0

  Since version 12.6.0, the `alldrives` plugin or a file on every drive that will needed by the plugin are not necessary anymore.

- Bacula does not automatically backup mounted drives. Any drive that is mounted in the normal file structure using a mount point will not be automatically backed up by Bacula. If you want it backed up, you must explicitly mention it in a **File=** directive in your Fileset, or use the **File=/** semantic mentioned above.

- The VSS plugin does not work with WinBMR When doing a backup that is to be used as a Bare Metal Recovery, do **not** use the VSS plugin. The reason is that during a Bare Metal Recovery, VSS is not available nor are the writers from the various components that are needed to do the restore. You might do a full backup to be used with a Bare Metal Recovery once a month or once a week, and all other days, do a backup using the VSS plugin, but under a different Job name. Then to restore your system, use the last Full non-VSS backup during the bare metal restoration of your system, and after rebooting do a restore with the VSS plugin to get everything fully up to date.
- Restoring components to a different machine that is not identical (or, in some cases, nearly identical) to the machine on which the component was backed up is not supported.
- **Available as of version 12.6.0**

  Backup of clustered volumes (csvfs) is supported.
- The `estimate` command does not work with the VSS plugin. When estimating a job that uses plugins, an error message regarding the plugin will be displayed.
- Some programs such as Windows Defender are protected from external operations. It is not possible for the SYSTEM service account to restore Windows Defender files. Using WinBMR and the Live CD can restore files.
- On Windows 10, the Kernel file (ntoskrnl.exe) is protected and cannot be restored during a System Writer restore. Using WinBMR can and the Live CD can restore these files.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

4.2 SaaS

Microsoft 365 (M365) Plugin

This whitepaper presents how to protect the most relevant elements of Microsoft 365 services using **Bacula Enterprise**.

Overview

Requirements

Microsoft 365 Personal, Family, Microsoft Home & Student subscriptions **are not supported** for backup/restore purposes.

It is necessary to have **full administrative access to the target Tenant** to protect in order to provide the required permissions to the Azure Application linked to this Bacula Enterprise Microsoft 365 Plugin.

Currently the plugin must be installed on a Linux based OS (RH, Debian, Ubuntu, SLES ..) where a Bacula Enterprise File Daemon is installed.

The OS where the File Daemon is installed must have installed Java version 8 or above.

If the Sharepoint module is going to be used, the OS where File Daemon is installed must also have the following packages installed:

- PowerShell v7.2.1 or above
- PnP Powershell v1.9.0 or above

Memory and computation requirements completely depend on the usage of this plugin (parallelization, environment size, etc). However, it is expected to have a minimum of 4GB RAM in the server where the File Daemon is running. By default, every job could end up using up to 512Mb of RAM in demanding scenarios (usually it will be less). However, there can be particular situations where this could be higher. This memory limit can be adjusted internally (see out-of-memory). Refer to the Scope section below for any service specific requirements.

Why Protect Microsoft 365?

This is a common question that arises frequently among IT and Backup professionals, so it is important to have a clear picture of it. It is true that Microsoft offers some services intended to prevent data loss:

- As with any cloud data, Microsoft 365 data is geo-replicated using Azure cloud to several destinations automatically and transparently. Therefore, complete data loss because of hardware failures are very unlikely to happen.
- Data Loss Protection service: Policy based services capable of detecting filtered content and act upon it encrypting it or modifying it in order to protect it (remove headers, etc). This is not a backup tool, is a service to prevent undesired actions to the content stored in Microsoft 365 (for example sharing confidential information with the wrong people).
- Retention policies of Microsoft 365: Microsoft retains a maximum of 30 days of deleted information from active subscriptions. Therefore it is possible to recover accidental deleted items inside that period. For more information:
  - Subscription Retention
  - High availability and business continuity

There are no other protection mechanisms for data protection. Below is a listing of challenges not covered:

- No Ransomware protection: If data suffers an attack and becomes encrypted, data is lost.
- No malicious attacker protection: If data is deleted permanently, data is lost.
- No real point-in-time recovery for Exchange 365, and recoveries of partially deleted files are limited to 30 days.
- Point in time recovery for OneDrive/Sharepoint limited to 30 days from deletion.
- It is not possible to align data protection of Microsoft 365 services to general retention periods or policies longer than 30 days.
- No automated way to extract any data from the cloud to save it in external places (this could lead to eventual compliance problems)

Scope

Bacula Enterprise Microsoft 365 Plugin is applicable on environments using any enterprise plan where target services of this plugin are included:

- Office 365 platform
- Exchange Online
- OneDrive for Business
- Sharepoint Online

The lack of any of the services in your subscription could lead to authentication problems (See troubleshooting section app-reg-error).

For more details about Microsoft 365 plans, visit these links:

- https://www.microsoft.com/microsoft-365/enterprise/compare-office-365-plans
- https://docs.microsoft.com/en-us/office365/servicedescriptions/office-365-platform-service-description/office-365-plan-options

This paper presents solutions for **Bacula Enterprise** version 12.8 and later, and is not applicable to prior versions.

Note: Important considerations

Before using this plugin, please carefully read the elements discussed in this section.

Onedrive Recycle Bin

Onedrive Recycle Bin cannot be protected with this plugin. This is a Microsoft limitation coming from their exposed REST APIs, where it is not possible to access Recycle Bin information. If future versions of Microsoft REST APIs officially include this function, Bacula Systems will include it as a new feature of this plugin.

Empty files

In general, empty files (files with 0 byte contents) are simply not backed up by Microsoft 365 plugin. In particular, email attachments or onedrive files will show a message in the joblog to inform about empty files detected and so not processed.

Backup of Attachments and Files

In general, this plugin backups two types of information:

- Objects
- Files

Objects are elements representing some entity in Microsoft 365 such as a calendar event, a contact, an email, a team, etc.

Files are attachments, hosted contents or OneDrive files.

While objects are directly streamed from memory to the backup engine, files need to be downloaded to the FD host before being sent. This is done in order to make some metadata checks and to improve overall performance, as this way operations can be parallelized. Every file is removed just after being completely downloaded and sent to the backup engine.

The path used for this purpose is established by the 'path' plugin variable, that usually is set up in the m365_backend script with the value: /opt/bacula/working

Inside the path variable, a 'spool' directory will be created and used for those temporary download processes.

Therefore, it is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs or through the same job (by default this is the case through the concurrent_threads=5 parameter), you would need at least that size for the largest file multiplied by the number of operations in parallel you run.

Accurate Mode and Virtual Full Backups

Accurate mode is ignored and Virtual Full backups are not supported. These features will be addressed in future versions of this plugin.

Details on the Sharepoint backup/restore

The M365 plugin overcomes the current limitations of the MS Graph API on backup/restore of SharePoint sites by using the Powershell PnP opensource project:

- https://learn.microsoft.com/en-us/powershell/sharepoint/sharepoint-pnp/sharepoint-pnp-cmdlets

This project has been designed to abstract many complexities of the Sharepoint Online APIs and provides all sorts of automation processes. Among them, site template provisioning, which is the main part of the method **Bacula Enterprise** Microsoft 365 Plugin uses in order to provide backup/restore capabilities for Sharepoint Online.

Site Template Provisioning

Microsoft can modify at any time (and they do it frequently) the properties, structure or limitations of any element belonging to a site kind (team site, communication site, project site, etc) whicn can cause the restore provision process to fail. It is even common to see native structures of Sharepoint Online trigger errors with particular items/values and this error comes from the Sharepoint Online API.

As a result, is not possible to guarantee that a given template is going to work in a provision process. Many times, the template might need to be partially modified before being applied (removing some particular item, removing some particular column..). That part of the process is not possible to predict or automate, so we strongly recommend frequent testing of site restore processes to detect any need of manual restore processes before a real restore is required.

Particular List/ListItems Restores

It is possible to select and restore a particular list or listItems, without restoring an entire Sharepoint Site, and pointing the operation to an existing Site. This is done using the Graph API, but as we have previously exposed, this API has many limitations with the Sharepoint API. One of the limitations is that lists or list items with advanced elements (location, images, etc) are not supported. Therefore this feature should only be used for user created lists implying simple elements like texts, checks, dates, etc.

Site Backups In Parallel

Backing up sites in parallel from the same FD is not supported.

Email Limitations

The following items associated to Outlook elements (Exchange Online service) are not supported due to the fact they are currently not accessible using the APIs this plugin relies on (MS Graph API and PnP Powershell):

- **Sharing permissions** associated to **MailFolders**
- **Public Folders**

Onenote Limitations

The following items associated to Onenote elements are not supported due to the fact they are currently not accessible using the APIs this plugin relies on (MS Graph API and PnP Powershell):

- **Sharing permissions** associated to **Notebooks**
- **Shared elements** from one entity to another one. To protect them it is necessary to make it always from the source entity.

Notebook 5000 files limitation

Onenote notebooks can only store up to 5000 different items into their associated OneDrive Libraries, this a M365 limitation.

When they surpass that limit they become unaccessible, not only for backup but also for accessing them through any other Microsoft native interface.

When a job finds this kind of notebooks, an error message will be generated and the notebook won't be included in the backup.

You can get more information about this limit and how to solve it in the following Microsoft documentation link:

- https://learn.microsoft.com/en-us/microsoftteams/troubleshoot/teams-onenote-integration/issue-access-notebook

External Personal Calendars

External calendars (as Google Calendar) connected to Microsoft accounts are not supported. They are external elements not exposed through MS APIs, so M365 Plugin cannot protect that information.

Calendar Events icons

Little icons that appear on the left of a calendar event title (usually auto-generated by the Outlook service) cannot be backed up, as they are not exposed through the MS Graph API the plugin is relying on.

Chats types

For the chats module, only types 'group' or 'one_on_one' can be backed up. Other types like: 'meeting' or 'unknownFutureValue' will be ignored.

MS Cloud APIs General Disclaimer

MS Cloud APIs are Microsoft property and they can change or evolve at any time. In particular, the Graph API is actively developed, containing new features every week, even if the version number of the service (1.0) is not changed as a result of any of those additions:

- https://developer.microsoft.com/en-us/graph/changelog

This situation is significantly different from traditional on-premise software, where each update is clearly numbered and controlled for a given server, so applications consuming that software, can clearly state what is offered and what are the target supported versions.

Microsoft is committed to trying to not break any existing functionality that could affect external applications. However, this situation can happen and therefore, cause some occasional problems with this plugin. Bacula Systems controls this with an advanced automatic monitoring system which is always checking the correct behavior of existing features, and will react quickly to that hypothetical event, but please be aware of the nature and implications of this kind of cloud technologies.

Features

The **Bacula Enterprise Microsoft 365 Plugin** is a very easy to deploy and configure plugin supporting the following M365 services:

- OneDrive
- Emails
- Mailbox settings
- Sharepoint Online
- Calendars
- Contacts
- OneNote
- Tasks
- Teams

- Chats
- Activity reports

It is shipped with advanced parallelization, resiliency, and flexibility features in addition to covering most of the possible M365 use case scenarios. A full feature list is presented below:

- Common features
  - Microsoft Graph API based backups
  - Multi-service backup in the same task
  - Multi-service parallelization capabilities
  - Multi-thread single service processes
  - Automatic parallelization of fetching processes
  - Generation of user-friendly report for restore operations
  - Network resiliency mechanisms
  - Latest Microsoft Authentication mechanisms
  - Discovery/List/Query capabilities
  - Restore objects to Microsoft 365
    - To original entity
    - To any other entity
    - To a different tenant (cross-tenant restore)
  - Restore any object to filesystem
  - Owner data protection feature:
    - Notify data owner about restore actions of his data
    - Do not proceed further until he enters his M365 credentials
  - Incremental & Differential backups
    - Advanced delta function for improved performance (for selected services)
- Backup and Restore of Exchange Online Mailboxes
  - Mailfolder, message and attachment granularity for restore
  - Email addresses and mailfolders selection capabilities for backup
  - Mailbox settings protection
  - Folder rules protection
  - Restore objects to Microsoft 365 or to any file-system
  - Restore MIME messages to any filesystem
  - Incremental & Differential backup
  - Support for user mailboxes and shared mailboxes
  - User categories protection
  - Fully indexed information into Bacula Catalog
  - Advanced search capabilities for restore operations

    - Ability to filter input data by date
    - Ability to exclude message fields from backup or from index
    - Exclude private or spam messages through powerful filtering capability
    - Export to PST format
- Backup and Restore of OneDrive for Business & Sharepoint Document libraries
    - Backup and Restore of User drives
    - Backup and Restore of Group drives
    - Backup and Restore of Sharepoint document libraries
        - Include/Exclude system libraries
        - Include/Exclude hidden libraries
    - Backup main entity drive unit, but also any other unit
    - Advanced selection capabilities
        - Target entities (Users/Groups/Sites)
            - List Include/List Exclude/Regex include/Regex Exclude…
        - Folder selection capabilities for backup
            - List Include/List Exclude
        - File selection capabilities for backup
            - Regex include/Regex Exclude…
        - Drive unit selection capabilities
    - Folder and file granularity for restore
    - Computed hash check at backup and restore time
    - Backup and restore of permissions shares
    - Backup and restore of shared elements to each entity
    - Backup and restore of OneDrive file versions
- Backup and Restore of Sharepoint Sites
    - Backup of Site Objects (MS Graph Object)
        - Backup Site Sharing permissions
    - Backup of full Site Templates (PnP Powershell Provisioning):
        - Site metadata
        - Lists metadata
        - ListItems metadata
        - WebPages metadata
        - DocumentLibraries metadata
        - Support for Sub-Sites
    - Restore backed up sites as new sites using Site Template (PnP Powershell Provisioning)
    - Backup/Restore Lists Objects (MS Graph Object)

  - Backup/Restore ListItems (MS Graph Object)
  - Backup/Restore Document Libraries Drive Items
- Backup and Restore of Contacts/People
  - Backup/restore Contacts
  - Backup/restore groups of contacts
  - Backup Organizational contacts (Read-only)
- Backup and Restore of Tasks
  - Backup/restore User todo tasks
  - Backup/restore Teams Planner tasks
- Backup and Restore of Calendars
  - Backup/Restore user calendars
  - Backup/Restore user groups of calendars
  - Backup/Restore groups calendar
    - Calendar permissions
  - Backup/Restore Events
    - Support for Attachments
      - File Attachments
      - Reference Attachments
      - Item Attachments (including backup of MIME objects)
- Backup and Restore of Onenote
  - Backup/Restore of Notebooks
    - User notebooks
    - Site notebooks
    - Group notebooks
  - Backup/Restore of SectionGroups/Sections
  - Backup/Restore of Pages
    - Support for Page resources: Images and files
- Backup and Restore of Teams
  - Backup/Restore of Team
    - Public and Private Teams
    - Team entity
    - Team members and roles
    - Team installed apps
  - Backup/Restore of Team Channels
    - Public and Private Channels
    - Channel entity

    - Channel members
    - Channel tabs
    - Channel messages
    - Channel messages hosted contents

- Backup and Restore of Chats
  - Backup/Restore of Chat
    - Chat entity
    - Chat installed apps
    - Chat tabs
    - Chat messages
    - Chat messages hosted contents
- Backup and Restore of Activity Reports
  - Groups Activity Storage CSV Report
  - Groups Activity Detail CSV Report
  - Active User Detail CSV Report
  - Activations User Detail CSV Report
  - App User Detail CSV Report
  - Yammer Device Usage User Detail CSV Report
  - Yammer Activity User Detail CSV Report
  - Yammer Activity Groups Detail CSV Report
  - Teams Device Usage User Detail CSV Report
  - Teams Activity User Detail CSV Report
  - Teams Activity Group Detail CSV Report
  - Skype Device Usage User Detail CSV Report
  - Skype Activity User Detail CSV Report
  - Sharepoint Site Usage Storage CSV Report
  - Sharepoint Site Usage Detail CSV Report
  - Sharepoint Activity User Detail CSV Report
  - OneDrive Usage Storage CSV Report
  - OneDrive Usage Account Detail CSV Report
  - OneDrive Activity User Detail CSV Report
  - Mailbox Usage Storage CSV Report
  - Mailbox Usage Detail CSV Report
  - Email App Usage User Detail CSV Report
  - Email Activity User Detail CSV Report

Architecture

Bacula Enterprise Microsoft 365 Plugin is using the **Microsoft Graph API** to perform almost all of its operations. Therefore, the plugin is working at the maximum granularity that the service provides.

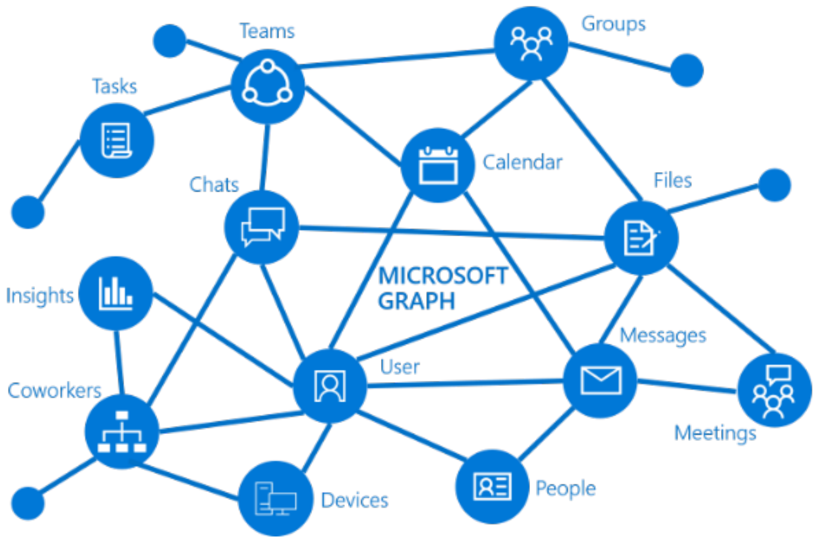


Fig. 74: Microsoft Graph

All the information is obtained using secure and encrypted HTTPS queries to Microsoft 365 from the File Daemon where the plugin is installed. All the requests are performed over the following endpoints:

- Graph API to manage all the services (except Sharepoint Online): https://graph.microsoft.com
- Login endpoint: https://login.microsoftonline.com/
- Sharepoint Online service endpoint: https://{tenantname}.sharepoint.com/

To get more information about Graph API, visit: https://learn.microsoft.com/en-us/graph/overview

The plugin will contact an Azure registered app named **bacula-m365-plugin** and will use it as a bridge to download the required data or objects during the time of a backup and send them to the Storage Daemon. Conversely, the plugin will receive them from an SD and perform uploads as needed during a restore.

The implementation is done through a Java Daemon, therefore Java is a requirement in the FD host. For more information about the bacula-m365-plugin, please, consult auth section.

Below is a simplified vision of the architecture of this plugin inside a generic **Bacula Enterprise** deployment:

Listed below is the information that can be protected using this plugin:

- Email
  - Common MailFolders (Inbox, Deleted Items, Drafts..)
  - User MailFolders & SubFolders
  - Messages & EventMessages
  - Attachments (ItemAttachments, FileAttachments and ReferenceAttachments)
  - Mailbox settings
  - Folder Rules

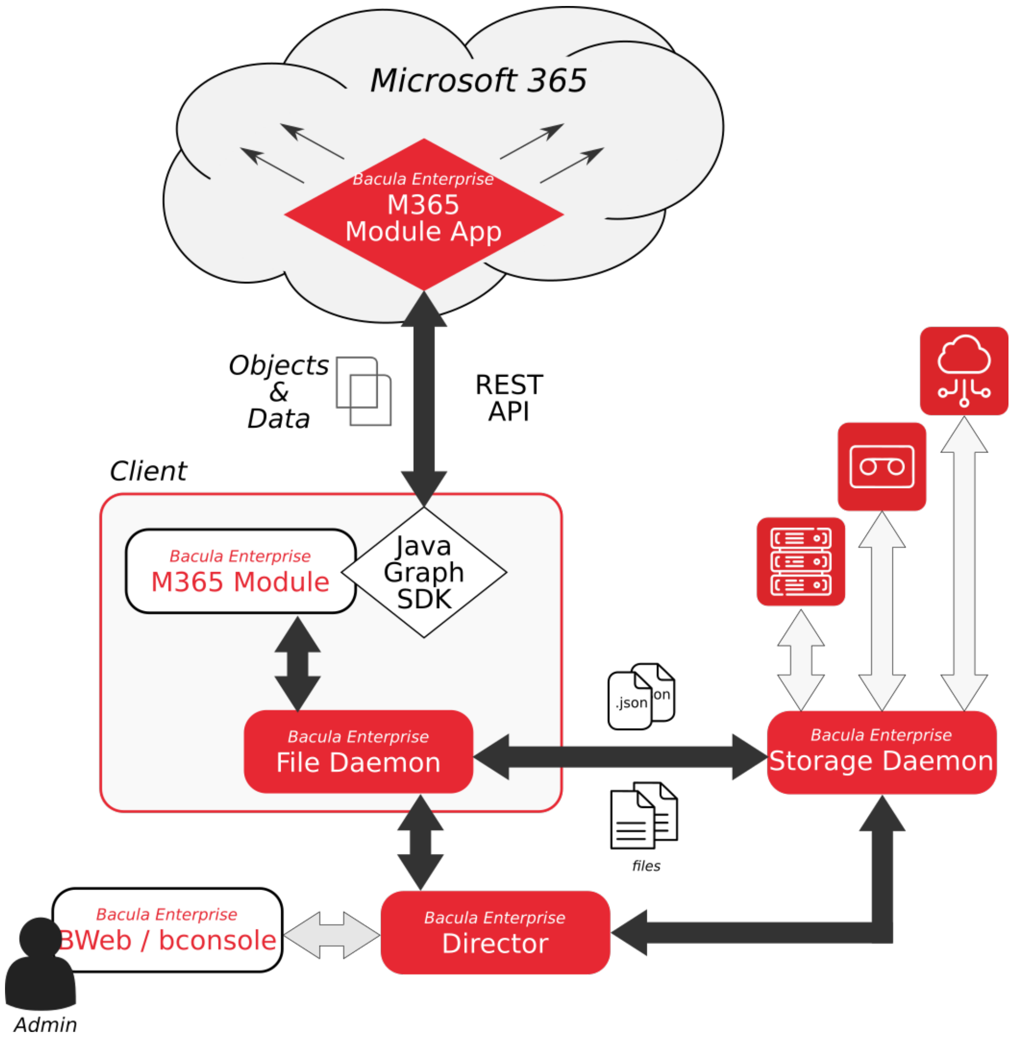


Fig. 75: M365 Plugin Architecture

- OneDrive
  - Onedrive for Business of Users, for each drive unit
    * Folders
    * Files
    * File Versions
  - Group libraries, for each drive unit
    * Folders
    * Files
    * File Versions
  - Sharepoint site libraries, for each drive unit
    * Folders
    * Files
    * File Versions
  - Shared permissions (direct access, share links, expiration times..)
  - SharedWithMe Objects
- Sharepoint
  - Pnp Site template
    * Site metadata
    * Lists metadata
    * ListItems metadata
    * WebPages metadata
  - Site Object
    * Site sharing permissions
  - Lists Objects
  - ListItem Objects
- Contacts/People
  - Contact object
  - Name of Groups of contacts
  - Organizational contact object
- Tasks
  - User Todo lists
  - User Todo tasks
- Calendars
  - Calendar objects
  - Calendar group objects
    * Calendar permissions

    - Events objects
    - Attachments (ItemAttachments, FileAttachments and ReferenceAttachments)
        - MIME objects where possible
- Notebooks
    - Notebook objects
    - Section objects
    - SectionGroup objects
    - Pages
        - Page contents (Html formatted) - Page resources
            - Page image files
            - Page object files (any other file apart from images)
- Teams
    - Team objects
    - Team settings
    - Team members and associated roles
    - Team installed apps
    - Channel objects
        - Channel tabs
        - Channel chat messages
            - Chat messsages hosted contents
- Chat
    - Chat objects
    - Chat installed apps
    - Chat tabs
    - Channel chat messages
        - Chat messages hosted contents
- Activity
    - Varied service reports in CSV format

All the information of each object is stored in JSON format (except for Pnp site template, which is stored in XML), preserving all their original values. When the plugin works with objects containing additional data (MIME files for messages, data for attachments and files of OneDrive, etc), that data is also backed up.

Services

In this section we will dig into how this plugin behaves for each particular service, describing special features and and behaviors that require an extended description.

OneDrive

Bacula Enterprise Microsoft 365 Plugin can protect OneDrive Business units associated to users, groups or sites. It is possible to utilize advanced selection methods to decide exactly what is backed up, as well as control precisely which items to restore and their destinations. The information protected with this service is:

- Files
- File versions
- Items M365 metadata (files and folders)
- Shared permissions

Files will keep their names in the catalog and will be included in a path like this:

- `/@m365/tenant.name/entitykind/ownerentityname/drives/unitname/root:/path/to/file/name-file.extension`

  (where `entitykind` can be users, sites or groups)

Version History

OneDrive and SharePoint can be configured to retain the history for files/items.

Onedrive hash check

Onedrive service stores a hash for every file hosted, using Microsoft algorithm QuickXOR. Bacula Enterprise Microsoft 365 Plugin calculates this hash and compares it to the Microsoft hashes at backup time, and also at restore time in order to ensure data integrity. Debug mode shows information about these hashes.

Onedrive shares

Bacula Enterprise Microsoft 365 Plugin is able to backup and restore shared elements. These kind of elements require a special treatment, as they are composed of two parts:

- In the source account, shared elements include special information about the permissions of the share (who and how the share must work)
- In the destination account, shared elements appear within an special category called 'SharedWithMe'.
  - SharedWithMe elements are not directly accessible from the destination account, as they are links to the source account. Therefore, Bacula Enterprise Microsoft 365 Plugin will query the source account in order to get these elements when they are part of the backup target. The source of those files must be located inside the same tenant the plugin is protecting, otherwise it will not be able to access them.

    - SharedWithMe elements need “delegated permissions” to be accessed. It means that an account having the elements will need to log in (during the backup or using the .query parameter=login method) in order to access those files. Therefore, it is advisable to carefully consider if you need to back up them with this mechanism or if it’s not necessary. By default, backing up these elements is disabled. Note that if you back up both accounts (source of share and destination of share), you don’t really need to enable this function. If you are backing up all users from your tenant, this is definitely not needed.

Shared permissions

Bacula Enterprise Microsoft 365 Plugin will query for the permissions of an item if this item has been shared directly. This means the plugin will not backup inherited permissions. In order to have inherited permissions in a backup, the top element where the original shared permissions were set needs to be included in the backup and in the restore. As an example, if a directory is shared but we restore only specific files contained in it, those files will not be shared as they were in originally. It is necessary to restore the whole directory in order to replicate the original inherited permissions as they were at the time of backup.

The method to store shared permissions is to include them as ‘extended_attributes’ of every file. This implies that permissions can only be restored directly to the Microsoft 365 service. A File Daemon restore to a local filesystem will only restore files, and shared permissions will not be restored.

Shared permissions can include links. Links are pre-generated URLs that can include, password, expiration dates and other configuration parameters as scopes, types, or affected identities. Shared permissions restores have special characteristics that must be considered and they are described below:

- Permission will be generated exactly as it was, but it will be a new permission object. This is similar to the situation with files. A restored file has the same contents as the original, but can include slightly different metadata because creation process was different.
    - If permission had a static link, a link will be generated, but the associated URL will be different from the original.
    - If a link had a password, the password will not be restored. Every link will be restored without a password (restore job log will warn about all links that had a password)
        * This is a limitation of OneDrive Business, as it does not allow passwords to be generated from APIs.
- Permissions can send a notification message to destination users.
    - This is configured at restore time with a parameter. Consequently, if different shares require different treatment, different restore sessions will be needed.
- Possible unexpected problems with permissions at restore time are treated as Warnings, not as Errors - as long as the file was restored successfully.

Shared permissions are not restored by default. You need to enable the option ‘drive_restore_shared_permissions’ during the restore session.

Shared with me

SharedWtihMe elements of each target account, if included in the Fileset, are backed up in a predefined directory called SharedWithMe inside the top folder of every selected account. For example, for a given account youraccout@yourdomain.com in a tenant called yourtenant:

Listing 100: **SharedWithMe**

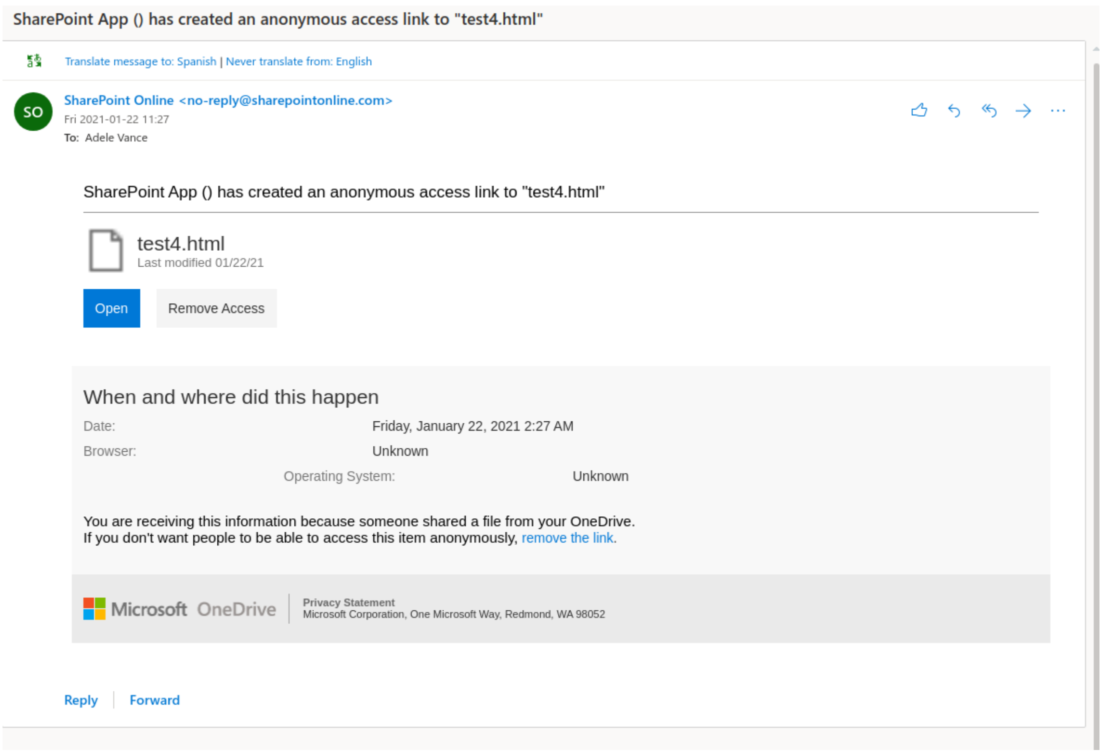
```
``/@m365/yourtenant/users/youraccout@yourdomain.com/drive/onedrive/root:/
↪sharedWithMe/``
```

At restore time, sharedWithMe elements are treated as any other regular file. However, it is important to note that sharedWithMe files, as we are in the receiver account, have no share permissions.

SharedWithMe elements included for backup are only the ones belonging to the current tenant. Backup of sharedWithMe elements from external tenants are not supported.

The plugin has an special parameter at restore time allowing it to skip sharedWithMe elements even if they are selected. This feature is intended to facilitate full restores where source and destination accounts are included. Please, note that a restore of a source account with share elements will present those elements to any receiver account if you enable the option to restore share permissions, as we have discussed in 3.2.1.

Please, be aware that restoring permissions from the plugin will generate automatic M365 reporting emails in the user Inbox as the following example shows:


SharePoint App () has created an anonymous access link to "test4.html"

Translate message to: Spanish | Never translate from: English

SharePoint Online <no-reply@sharepointonline.com>
Fri 2021-01-22 11:27
To: Adele Vance

SharePoint App () has created an anonymous access link to "test4.html"

test4.html
Last modified 01/22/21

Open | Remove Access

When and where did this happen

Date: Friday, January 22, 2021 2:27 AM
Browser: Unknown
Operating System: Unknown

You are receiving this information because someone shared a file from your OneDrive.
If you don't want people to be able to access this item anonymously, remove the link.

Microsoft OneDrive | Privacy Statement
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Reply | Forward

Fig. 76: Restored permissions warning message

Please, go to the Configuration section of this document to see how to set up the sharedWithMe skip option.

Note: If you enable the backup for file versions (calendars) and sharedwithme elements, sharedwithme elements will include all their versions in the backup.

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the drive module.

In order to select the drive module, the common *service* parameter must be equals or be containing the value *drive*.

Entities that can include one drive units are: users, groups or sites.

Option	Required	Default	Values	Example	Description
drive	No		Valid names or ids of existing drives on the selected tenant belonging to the selected entity (user, group or site) separated by ,	documents, images, b!12344	Will backup only selected drive units belonging to the specified entity (user, group or site)
drive	No		Valid names or ids of existing drives on the selected tenant belonging to the selected entity (user, group or site) separated by ,	onedrive	Will backup all drives except the excluded in this lists, belonging to the specified entity (user, group or site)
drive	No		Valid regex	*.pages	Backup matching drive units (based in the drive unit name)
drive	No		Valid regex	^site.*	Exclude matching drive units (based in the drive unit name)
drive	No		Strings representing existing **folders** for the given users separated by ,	Customers, Partners	Backup only specified **folders** belonging to the selected users
drive	No		Strings representing existing **folders** for the given users separated by ,	Personal	Exclude selected **folders** belonging to the selected users
drive	No		Valid regex	.*Company	Backup matching drive files or folders. If you provide list parameters (files + files_exclude) combined with regex ones, be aware that regex ones are applied after list parameters
drive	No		Valid regex	.*Plan	Exclude matching drive files or folders from the selection. If you provide list parameters (files + files_exclude) combined with regex ones, be aware that regex ones are applied after list parameters. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
drive	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include system documentLibraries of sharepoint (sitepages, siteassets, etc).
drive	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include SharedWithMe elements of every target entity in the backup process
drive	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include Onedrive former versions of every file into the backup process
drive	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include Onedrive files sharing permissions as part of the backup. If you need higher performance, it is recommended to disable this option
drive	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Disable the hash checking mechanism for each file downloaded. This is only recommended for document libraries where the hash stored at M365 level is detected to be incorrect for an important number of files (specially in sharepoint

- Note: In previous versions, instead of drive_files* parameters, it was possible to use files_* parameters. They are deprecated parameters now as we recommend to use module specific parameters in order to have better control and more possibilities inside a single fileset. However, jobs using files_* parameters will still work.
- Note: In previous versions, drive_shared_with_me was enabled by default. However, Microsoft changed at some point the permissions of these elements and now they are only accessible through delegated permissions. Delegated permissions require a more complex operation, while sharedWithMe elements are only needed in some particular

use cases (such as protecting only a sub-set of users). Therefore, this now disabled by default (since Bacula 18.0.4).

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of onedrive module restore operations:

- destination_user, destination_group, destination_site, destination_drive, destination_path, send_report, allow_duplicates
- drive_skip_sharedwitme, skip_versions, restore_share_permissions, drive_send_invitations, drive_invitations_message
- debug, foreign_container_generation

Use cases

The following restore scenarios are supported:

- Restore files, directories, or file versions to original entity drive or to a different entity drive
  - Restore parameters implied: **destination_user, destination_group, destination_site**
  - If the backup was from a user (using user=user@name), **destination_user** should be used if we want to restore to another user
  - If the backup was from a group (using group=groupName), **destination_group** should be used if we want to restore to another group
  - If the backup was from a Sharepoint site (using site=groupName), **destination_site** should be used if we want to restore to another site
- Restore file(s)/dir(s) or file version(s) to original path or to a different path
  - Restore parameters implied: **destination_path**
- Restore file(s)/dir(s) or file version(s) to original drive unit or to a different drive unit
  - Restore parameters implied: **destination_drive**
- Restore file(s)/dir(s) or file version(s) to local file system (general restore **where** parameter must be set to a path)
- It is possible to make general restore selections, but avoid restoring versions
  - Restore parameters implied: **skip_versions**
- It is possible to restore sharing permissions of implied files

    - Restore parameters implied: **restore_shared_permissions, drive_send_invitations, drive_invitations_message**

- It is possible to make general restore selections, but specify if backed up shared elements must be considered
    - Restore parameters implied: **drive_skip_sharedwitme**
- It is possible to control whether or not duplicate elements are allowed (based on file id):
    - Restore parameters implied: **allow_duplicates**

Particularities:

- If no **destination entity** is provided, the destination entity will be looked for inside the backed up path, so the destination user will be the same as the original one
- If no **destination_path** is provided, the destination path will be the same as the original one
    - If a destination entity was provided, but no destination_path was provided and the selected file did not belong to the destination entity:
        - ∗ A new folder will automatically be created inside the target entity
        - ∗ For each 'foreign' entity, a new folder will be created
        - ∗ Inside each 'foreign' entity folder, the original path structure will be preserved when restoring the files
            - · *Unless the parameter **foreign_container_generation** is disabled

For more details about the behavior of each parameter, please check the general section of restore parameters.

Fileset examples

Full OneDrive of one user:

Listing 101: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com"
   }
}
```

Folders of one user, but include sharedWithMe elements:

Listing 102: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-adelev-shared
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
```

(continues on next page)

(continued from previous page)

```
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com drive_
↪files=\"dir1,dir2\"
   drive_shared_with_me=yes"
   }
}
```

Full OneDrive of one group, include version history:

Listing 103: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-devteam-versions
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 group=\"Dev Team\" drive_version_history=true"
   }
}
```

Full OneDrive of one sharepoint site:

Listing 104: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-live
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 site=\"Live @ MyEnterprise\""
   }
}
```

Specific drives of a sharepoint site, including some system:

Listing 105: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-live-2-drives
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 site=\"Live @ MyEnterprise\" drive_system_
↪include=yes drives=b!7Tf3z7izSES7kjOHD-YM-
   0kScgNXL6lFnL4O2LWLMy4yH2tqunhTSpSwmPR5a0hq,b!7Tf3z7izSES7kjOHD-YM-
↪0kScgNXL6lFnL4O2LWLMy41XlWc2E0pTKfPzlKxhztE"
   }
}
```

Exclude directories of two users:

Listing 106: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-adjon-users-notemp
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=\"adelev@baculaenterprise.onmicrosoft.com,
↪jonis@baculaenterprise.onmicrosoft.com\"
   drive_files_exclude=\".*.temporary\""
   }
}
```

Exclude mp3 files and avi files:

Listing 107: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-exclude
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=\"adelev@baculaenterprise.onmicrosoft.com,
↪jonis@baculaenterprise.onmicrosoft.com\"
   drive_files_exclude=\".*.mp3|.*.avi\""
   }
}
```

Email/Mailboxes

Bacula Enterprise Microsoft 365 Plugin can protect M365 Mailboxes associated to users. It is possible to utilize advanced selection methods to decide exactly what is backed up (folders included/excluded, users included/excluded), as well as control precisely which messages or attachments to restore and where (original user's account or another user's account). The information protected with this service is:

- Folders (Inbox, Deleted Items, Sent…)
  - Messages (Metadata and contents)
  - Attachments
    - File Attachments
    - Item Attachments
    - Reference Attachments
- Folder Rules
- Mailbox Settings
- Outlook Categories

Public folders are not supported and **folder sharing permissions are not protected** (see email-limit). On the other hand, folders are backed up always from their original accounts:

- This means if a folder 'folderA' is shared from user 'U1' to user 'U2', if you only backup the entire mailbox of U2, contents of folderA will not be protected. In order to protect folderA you will need to add some additional job protecting U1 and folderA into the backup set or simply add the entire mailbox of U1.

User mailboxes and **shared mailboxes** are supported, it is only required to setup the correct address in user* parameters (shared mailboxes are treated as users). For **groups**, please note that they **have no real mailbox**, therefore it is not possible to backup the emails belonging to a given group. However, protecting emails of any of the users that is member of those groups will result in protecting all the received emails of that group.

Mailbox backup includes the following features:

- Incremental backup with Delta function:
  - Delta function is applied for each folder individually
- MIME object backup:
  - Based on the fileset parameter **email_mime** it is possible to get mime messages as well as the M365 objects. These kind of objects can be useful to have if there is any plan of using the emails outside the M365 service.
  - At restore time, if the restore operation is done via M365 services and not to a local FileSystem, mime_objects are automatically ignored.
- Multilevel-attachments:
  - M365 supports having messages or event objects as the attachment of a given message object. These attachments can have additional attachments. This situation is what M365 Plugin considers multilevel-attachments. Some particular objects such as Files, when present in this multilevel way, are got embedded in the metadata of the message attached. This implies that the plugin must get them in a single HTTP call and load them entirely in memory. If there is not enough memory in the system, the variable **multilevel_attach** can be disabled and those objects will be ignored.
  - Multilevel-attachments are backed up at the same level. This means that a message with this structure:
    - Base Message has 1 Message-Attachment, Message-Attachment has 1 more internal Message-Attachment 2, and Message-Attachment 2 has a File attachment
    - When restored, the result will be all Attachments being restored to the same level, not one inside the another as they were originally. For example:
    - Base Message
      - Message-Attachment 1
      - Message-Attachment 2
      - File Attachment

Messages will be formatted in the catalog in order to not include sensitive information and will be included in a path like this:

- `/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 09.58.52..J-123.10.msg`
  - Where the message name is composed as: messageDate..J-JobId.JobIndex.msg
    - messageDate corresponds with the receivedDate
    - JobIndex is an internal index relative to a backlup job execution

        - The extension .msg corresponds to a file containing an email
        - Mime messages will have the extra word 'mime' in their extension. For example:
            - `/@m365/tenant.name/users/user@tenant.com/email/foldername/ 2021-04-19 15.54.06..J-2021-04-19_15.54.45_03.10.mime.msg`
- `/@m365/tenant.name/users/user@tenant.com/email/mailbox\_settings.mail.set`
    - Mailbox Settings
- `/@m365/tenant.name/users/user@tenant.com/email/rules/nameRule.mail.rule`
    - Mailbox Rules
- `/@m365/tenantname/users/usermail/outlook\_category_name.olk.cat/`
- Mailbox Categories

Attachments will be stored together with message objects:

- They include their original name (file name)
- They have an extension about their type (.file.att, .ref.att or .item.att)
- The first part of the attachment name is the name of the parent message

Here are a few attachment examples:

`/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 15.54.06..J-123.10.Aliquid.gen.1.file.att`

`/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 15.54.06..J-123.10.Dolor.gen.4.ref.att`

`/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 15.54.06..J-123.10.Sapien Nostrum Aperiri Unum - t.3.item.att`

`/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 15.54.06..J-123.10.Veri.gen.2.file.att`

`/@m365/tenant.name/users/user@tenant.com/email/foldername/2021-04-19 15.54.06..J-123.10.msg` → *Parent message*

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the email module.

In order to select the email module, the common *service* parameter must be equals or be containing the value *email*.

Entities that can include mailboxes are: users.

Option	Required	Default	Values	Example	Description
email_	No		Strings representing existing **mailfolders** for the given users or groups separated by ‘,’	Inbox, Sent	Backup only specified **mailfolders** belonging to the selected users
email_	No		Strings representing existing **mailfoders** for the given users or groups separated by ‘,’	Archive Personal	Exclude selected **mailfolders** belonging to the selected users
email_	No		Valid regex	.*Company	Backup matching mailfolders. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
email_	No		Valid regex	.*Plan	Exclude matching mailfolders from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
email_	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include multilevel attachments in backup. A multilevel attachment is, for example, a message attached to a message containing more elements attached inside. Under certain circumstances, multilevel attachments could imply higher memory needs, as some of them are embedded in the metadata of the objects to backup, so they need to be caught in a single call and be loaded entirely in memory. This parameter is intended to control the behavior against that situation.
email_	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude any attachment from backup
email_	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Backup raw MIME file of every email, in addition to the Message object itself
email_	No	No	0, no, No, false	Yes	Backup mailbox settings of included users

Note: In previous versions, instead of email_files* parameters, it was possible to use files_* parameters. They are deprecated parameters now as we recommend to use module specific parameters in order to have better control and more possibilities inside a single fileset. However, jobs using files_* parameters will still work. The following fields: email_messages_exclude_expr, email_messages_exclude_index_expr, email_fields_exclude, email_fields_exclude_index, email_filter_received_from are available from Bacula Enterprise version 14.0.

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of email module restore operations:

- destination_user, destination_path, send_report, allow_duplicates, debug, foreign_container_generation

Use cases

The following restore scenarios are supported:

- Restore directories, emails, or attachments to original user or to a different user
  - Restore parameters implied: **destination_user**
- Restore directories, emails, or attachments to original path or to a different path
  - Restore parameters implied: **destination_path**
- Restore files, directories, or file versions to local file system (general restore **where** parameter must be set to a path)
- It is possible to control whether or not duplicate elements are allowed (based on file id):
  - Restore parameters implied: **allow_duplicates**

Particularities:

- If no **destination_user** is set, every message will be restored into its original mailbox
- If no **destination_path** is set, every message will be restored into its original path
  - If the selection contains messages from several users:
    - ∗ Original user messages will be restored in their original location
    - ∗ For other users, a special folder will be created with the email address of each of them, containing the full path and messages of the restored objects, unless the parameter **foreign_container_generation** is disabled
    - ∗ Example:
- Restored elements will be duplicated by default, unless **allow_duplicates** variable is disabled
  - Even when disabling that variable, messages will be checked by id. So if there is an element with the same information but different ID, it will not be considered to be a duplicate

For more details about the behavior of each parameter, please check the general section of restore parameters.

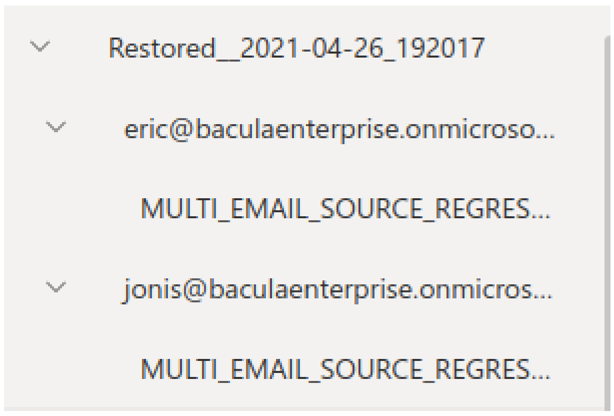

Restored__2021-04-26_192017
eric@baculaenterprise.onmicroso...
MULTI_EMAIL_SOURCE_REGRES...
jonis@baculaenterprise.onmicros...
MULTI_EMAIL_SOURCE_REGRES...

Fig. 77: Restore of emails from 2 different users over a third mailbox without destination_path result in auto-generated Restore_date folder containing those 2 foreign users with the restored folder inside of them

Messages exclude expressions

Bacula Systems is aware about one of many privacy concerns that may arise when tools like this M365 Plugin enables the possibility to backup and restore data coming from different users, so the backup administrator can restore potentially private data at his will. Moreover, emails are usually one of the most critical items in terms of privacy.

One of many strategies this plugin offers in order to deal with that problem is the possibility to exclude messages. This is a very powerfull feature where it is possible to use quite flexible expressions that allow to select a subset of messages and simply exclude them from the backup:

- email_messages_exclude_expr new fileset parameter

Or only from the index (from the catalog)

- email_messages_exclude_index_expr new fileset parameter

Not only messages can be excluded but also select only a subset of email fields to be included in the protected information. It is possible to exclude fields from the backup:

- email_fields_exclude new fileset parameter

Or only from the index (from the catalog):

- email_fields_exclude_index new fileset parameter

Please, be aware that if the fields are excluded from backup, the restore operation to M365 can fail easily, since for instance emails cannot exist in M365 without From field, Subject field, etc.

All four discussed expressions are based on an internal structure of fields to work with. Below you can see the entire list of fields that you can use:

- emailTags
- emailSubject
- emailFolderName
- emailFrom
- emailTo
- emailCc

- emailBodyPreview
- emailImportance
- emailTime
- emailIsRead
- emailIsDraft

Please note that it is very important to write the fields exactly as written above.

These fields can be used in a comma separated list in the 'email_fields_exclude' parameter and also 'email_fields_exclude_index' parameter.

Then, for 'email_messages_exclude_expr' and 'email_messages_exclude_index_expr' use them in a valid boolean expression in **Javascript** language syntax. Some examples are provided below:

Listing 108: **Expression to exclude messages where subject includes the word 'private'**

```
emailSubject.includes('private')
```

Listing 109: **Complex expression to exclude messages that are not read and are Draft or their folder name is named Private**

```
!emailIsRead && (emailIsDraft || emailFolderName == 'Private')
```

Listing 110: **Expression to exclude messages based on the received or sent date**

```
!emailTime < Date.parse('2012-11-01')
```

Listing 111: **Expression to exclude messages using a regex based on emailFrom**

```
/.*private.com/.test(emailFrom)
```

Note: This feature is available since Bacula Enterprise version 14.0

Expression tester

This expression mechanism can sometimes be uncertain for end users, where they can have doubts about the correct behavior of their prepared expressions. In order to help with that, M365 Plugin presents a query method that allows to test those expressions against a static pre-loaded set of data.

There are two commands available:

- Show command: It will show the static data in json format, so it is possible to see the contents to adapt the expressions to test
- Test command: It will apply the expression parameters to the pre-loaded static data

The test command has the following format:

Listing 112: **Expression tester Show command**

```
.query client=<your-fd-client> plugin="m365: tenant=<your-tenant-id>"␣
↪parameter=email-expr-show
```

The show command has the following fomat

Listing 113: **Expression tester Test command**

```
.query client=<your-fd-client> plugin="m365: tenant=<your-tenant-id> email_
↪messages_exclude_expr = \"<your-js-expression>\"" parameter=json|email-expr-
↪test
// Or
.query client=<your-fd-client> plugin="m365: tenant=<your-tenant-id> email_
↪messages_exclude_index_expr = \"<your-js-expression>\""␣
↪parameter=json|email-expr-test
```

Please, not that you need to provide a valid tenantId, even if it's not really used to process any data.

The test command produces some JSON output with objects with the exact format that is received from Microsoft and, consequently the same format that is stored in backup. Please not the 'total' value at the end, where the value of 12 total pre-loaded messages is shown

Listing 114: **Expression tester Show command output**

```
.query client=<your-fd-client> plugin="m365: tenant=<your-tenant-id>"␣
↪parameter=json|email-expr-show
....
    "email-12": {
      "body": {
        "content": "These are the contents in text format of the 12 email of␣
↪test data. It has the following categories:orange, black, white, purpleYou␣
↪can try to filter this body using any JS method like /.*12.*/.
↪test(emailBody) or emailBody.includes(12)",
        "contentType": "TEXT"
      },
      "ccRecipients": [
        {
          "emailAddress": {
            "address": "danny@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "lucas@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "terese@other.com"
          }
        }
      ],
```

(continues on next page)

(continued from previous page)

```
      "from": {
        "emailAddress": {
          "address": "elon@other.com"
        }
      },
      "hasAttachments": false,
      "isDraft": false,
      "isRead": false,
      "replyTo": [
        {
          "emailAddress": {
            "address": "elon@other.com"
          }
        }
      ],
      "sentDateTime": {
        "dateTime": {
          "date": {
            "year": 2021,
            "month": 12,
            "day": 5
          },
          "time": {
            "hour": 11,
            "minute": 30,
            "second": 0,
            "nano": 0
          }
        },
        "offset": {
          "totalSeconds": 0
        }
      },
      "subject": "This is private subject 12",
      "toRecipients": [
        {
          "emailAddress": {
            "address": "laura@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "jack@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "john@other.com"
          }
        }
      ],
      "categories": [
```

(continues on next page)

(continued from previous page)

```
        "orange",
        "black",
        "white",
        "purple"
      ]
    }
  },
  {
    "total": "12"
  }
```

The test command, on its side will produce two different outputs. The first part presents the same format than the show format, and those are the messages that would be included in the backup. The second part presents a different format, so an output like:

Listing 115: **Expression tester Test command, index part output**

```
.query client=<your-fd-client> plugin="m365: tenant=<your-tenant-id>"␣
↪parameter=json|email-expr-show
....
      {
    "meta-email-12": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "johndoe.onmicrosoft.com",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is private subject 12",
      "EmailFolderName": "/",
      "EmailFrom": "elon@other.com",
      "EmailTo": "laura@other.com,jack@other.com,john@other.com",
      "EmailCc": "danny@other.com,lucas@other.com,terese@other.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 235,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "m365"
    }
  },
  {
    "total-backup": "12"
  },
  {
    "total-index": "12"
  }
```

That part represents the information that would be indexed in the backup (included into the catalog). You can also see the total entries at the end, that are very useful to quickly compare with the original 12 value and so, knowing if our expression is filtering the expected data or not. Below we provide an example where some filtering is applied to the backup, but also to the index:

Listing 116: **Expression tester Test command, index part output**

```
.query client=127.0.0.1-fd plugin="m365: tenant=xxxx-xx-xxxx-xxx-xxxx email_
↪messages_exclude_expr=\"emailFrom == 'elon@other.com'\" email_messages_
↪exclude_index_expr=\"emailSubject.includes('private')\""␣
↪parameter=json|email-expr-test
...
     {
    "meta-email-4": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "johndoe.onmicrosoft.com",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is orange subject 8",
      "EmailFolderName": "/",
      "EmailFrom": "bob@company.com",
      "EmailTo": "laura@company.com,jack@company.com,john@company.com",
      "EmailCc": "danny@company.com,lucas@company.com,terese@company.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 232,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "m365"
    }
  },
  {
    "total-backup": "6"
  },
  {
    "total-index": "4"
  }
```

In case your expression is not valid, the plugin will also inform about that with the following message:

- error=Error listing elements. Cause: Predicate test error!! Review your query ....

Fileset examples

Backup Full MailBox of some users, but excluding some folders:

Listing 117: **Fileset Example**

```
Fileset {
   Name = fs-m365-drive-adjon-users-notemp
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=\"adelev@baculaenterprise.onmicrosoft.com,
↪jonis@baculaenterprise.onmicrosoft.com\"
   email_files_exclude=\"*.temporary\""
   }
}
```

Backup all MailBoxes:

Listing 118: **Fileset Example**

```
Fileset {
   Name = fs-m365-email-all
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5"
   }
}
```

Backup only the Inbox folder of some users:

Listing 119: **Fileset Example**

```
Fileset {
   Name = fs-m365-email-2user-inbox
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user="peter@mycompany.com,john@mycompany.com" email_
↪files=inbox"
   }
}
```

Backup some users and include MIME messages:

Listing 120: **Fileset Example**

```
Fileset {
   Name = fs-m365-email-2user-mime
   Include {
```

(continues on next page)

(continued from previous page)

```
      Options { signature = MD5 }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user="peter@mycompany.com,miriam@mycompany.com"␣
↪email_mime=yes"
   }
}
```

Disable backup of multi-level attachments:

Listing 121: **Fileset Example**

```
Fileset {
   Name = fs-m365-email-multilevel-attach
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user="peter@mycompany.com,miriam@mycompany.com"␣
↪multilevel_attach=no"
   }
}
```

Well known folders

Microsoft 365 can present the folders information in local languages to the user.

In general, there is no 'multilanguage' support, in the sense that folders must be included with their original name. For example, if you create a folder named 'books', you cannot expect it to be backed up if you use something like 'livres' or 'libros' from other languages. You need to use the real name that was used to create such folder.

There is one very important special case though, which is 'well known folders'. Well known folders are folders like 'inbox', 'outbox', 'archive'… A full list can be found here: https://docs.microsoft.com/en-us/graph/api/resources/mailfolder?view=graph-rest-1.0

That kind of folders can be 'found' by the plugin using their 'well known name', instead their internal id, as it's the general case. Therefore, for them it is possible to get the folder using their English well known word even if the user sees the folder with a translated word.

For example, to backup inbox it is needed to use 'inbox' even if for some user it is 'Posteingang' or 'boîte de réception'. Microsoft 365 Plugin will recognize this special words and will query the information through them.

To summarize:

- Well known folders -> Use English word
- Other user folders -> Use original name

Calendars

Bacula Enterprise Microsoft 365 Plugin can protect M365 Calendars associated to users or groups:

- Groups on M365 have only one calendar assigned, without the possibility to add more
- Users can utilize calendar selection features using the files* parameters. Target users can also be customized using user* parameters.

It is important to note that the Calendar of a group can only be accessed using delegated permissions. Delegated permissions mean the plugin will impersonate a logged in user in order to access the required information. Therefore, to backup a calendar of a group, the logged in user needs to belong to that group.

The information protected with this module is detailed below:

- For Users:
  - Calendar Groups
  - Calendars
    - Events (Metadata and contents)
      - Master events are backed up only once
      - Restoring master event results in the application of all event instances
    - Attachments
      - File Attachments
      - Item Attachments
      - Reference Attachments
- For Groups
  - Calendar
    - Events (Metadata and contents)
      - Master events are backed up only once
      - Restoring master event results in the application of all event instances
    - Attachments * File Attachments * Item Attachments * Reference Attachments
    - Calendar for groups uses delegated-permissions

Calendar module includes the following features:

- Incremental backup (it is done without Delta function)
- Multilevel-attachments:
  - M365 supports having messages or event objects as the attachment of a given message object. Those attachments can present additional attachments. This situation is what the M365 Plugin considers multilevel-attachments. Some particular objects like Files, when present in this multilevel way, are embedded in the metadata of the message attached. This implies that the plugin must get them in a single HTTP call and load them entirely in memory. If there is not enough memory available on the system, the variable **multilevel_attach** can be disabled and those objects will be ignored.
  - Multilevel attachmets are backed up at the same level. This means that a message with this structure:

   - Base Event have 1 Event-Attachment, Event Attachment 1 has a Message Attachment 2, and Message Attachment 2 has a File attachment
   - When restored, the result will be all Attachments being restored to the same level, not one inside the another as they were originally. For example:
   - Base Event
     - Event-Attachment 1
     - Message-Attachment 2
     - File Attachment

Catalog structure for Calendar and Events presents the same structure as the Email module (changing 'email' for 'calendar' in the path ; event extensions are .evt in comparison to the .msg extension of messages). For more information, please check section 4.2 Email/Mailboxes

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the calendar module.

In order to select the calendar module, the common *service* parameter must be equals or be containing the value *calendar*.

Entities that can include calendars are: users (calendar groups, calendars and events) and groups (one calendar and events).

Op-tion	Re-quire	De-fault	Values	Ex-am-ple	Description
cal-en-dar_f	No		Strings repre-senting existing **calendars** for the given users or groups separated by ‘,’	Hol-i-days, Meet ings	Backup only specified **calendars** belonging to the selected entities (Users, Groups, Sites..)
cal-en-dar_f	No		Strings repre-senting existing **calendars** for the given users or groups separated by ‘,’	Train ings, Team Beer	Exclude selected **calendars** belonging to the selected enti-ties
cal-en-dar_f	No		Valid regex	.*Co pany	Backup matching calendars. Please, only provide list pa-rameters (files + files_exclude) or regex ones. But do not try to combine them.
cal-en-dar_f	No		Valid regex	.*Pla	Exclude matching calendars from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
cal-en-dar_ı	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include multilevel attachments in backup. A multilevel at-tachment is, for example, a message attached to a message containing more elements attached inside. Under certain circumstances, multilevel attachments could imply higher memory needs, as some of them are embedded in the meta-data of the objects to backup, so they need to be caught in a single call and be loaded entirely in memory. This parame-ter is intended to control the behavior against that situation.
cal-en-dar_e	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude any attachment from backup

Note: In previous versions it was possible to use files_* parameters instead of calendar_files* parameters,. These parameters are now deprecated. We recommend to use module specific parameters in order to have better control and more possibilities inside a single fileset. However, jobs using files_* parameters will still work.

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of calendar module restore operations:

- destination_user, destination_group, destination_path, send_report, allow_duplicates
- restore_share_permissions, calendar_name, debug, foreign_container_generation

Use cases

The following restore scenarios are supported:

- Restore calendars, events, or attachments to the original user or to a different user
  - Restore parameters implied: **destination_user**
- Restore calendars, events, or attachments to the original calendar group or to a different calendar group
  - Restore parameters implied: **destination_path**
- Restore calendars, events, or attachments to the original calendar or to a different calendar
  - Restore parameters implied: **calendar_name**
- Restore calendars, events, or attachments to local file system (general restore **where** parameter must be set to a path)
- It is possible to control whether or not duplicate elements are allowed (based on file id):
  - Restore parameters implied: **allow_duplicates**

Particularities:

- If no **destination_user** is set, every event will be restored into its original calendar
- If no **calendar_name** is set, every event will be restored into its original calendar
  - If the selection contains events from several users:
    - Original user messages will be restored in their original location
    - For other users, a special calendar will be created with the email address of each of them in the name, unless the parameter **foreign_container_generation** is disabled
      - If **foreign_container_generation** is disabled, a calendar with the same name present in the original user will be looked for in the destination user, if it does not exist, an error will be triggered
  - Restored elements will be duplicated by default, unless the **allow_duplicates** variable is disabled
    - Even disabling that variable, events will be checked by id. So if there is an element with the same information but a different ID, it will not be considered to be a duplicate

For more details about the behavior of each parameter, please check the general section of restore parameters.

Fileset examples

Backup all calendars of one user:

Listing 122: **Fileset Example**

```
Fileset {
   Name = fs-m365-calendar-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=calendar tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com"
   }
}
```

Backup all calendars from all users or groups:

Listing 123: **Fileset Example**

```
Fileset {
   Name = fs-m365-calendar-all
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=calendar tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5"
   }
}
```

Backup all calendars from a group:

Listing 124: **Fileset Example**

```
Fileset {
   Name = fs-m365-calendars-dev-team
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=calendar tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 group=\"Dev Team\" "
   }
}
```

Backup only one calendar of some users:

Listing 125: **Fileset Example**

```
Fileset {
   Name = fs-m365-calendar-2user-inbox
```

(continues on next page)

(continued from previous page)

```
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=calendar tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 group="Dev Team,Marketing team" calendar_
↪files=CompanyCal"
   }
}
```

Activate backup of multi-level attachments:

Listing 126: **Fileset Example**

```
Fileset {
   Name = fs-m365-calendar-multilevel-attach
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=calendar tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 user="peter@mycompany.com,miriam@mycompany.com
↪" calendar_multilevel_attach=yes"
   }
}
```

Contacts/People

Bacula Enterprise Microsoft 365 Plugin can protect M365 Contacts associated to users, but also organizational contacts that are common to a given tenant. It is possible to select what users to backup (user* parameters) and also the specific contact folders to include/exclude (files* parameters).

The information protected by this module is detailed below:

- Users:
  - Contact Folders (Main contact folders and any others)
  - Contacts
- Tenant
  - Organization contacts

Contact module includes the following features:

- Incremental backup with delta function for user contacts
  - Delta function is applied to each Contact Folder
  - Organization contacts do not support incremental backup
- **Organization contacts are read-only** in M365 Graph
  - Only local restore is supported

Organization contacts can be located in Microsoft 365 Admin Center > Users > Contacts:

Catalog structure for contacts is presented below:

- User contacts are stored in:

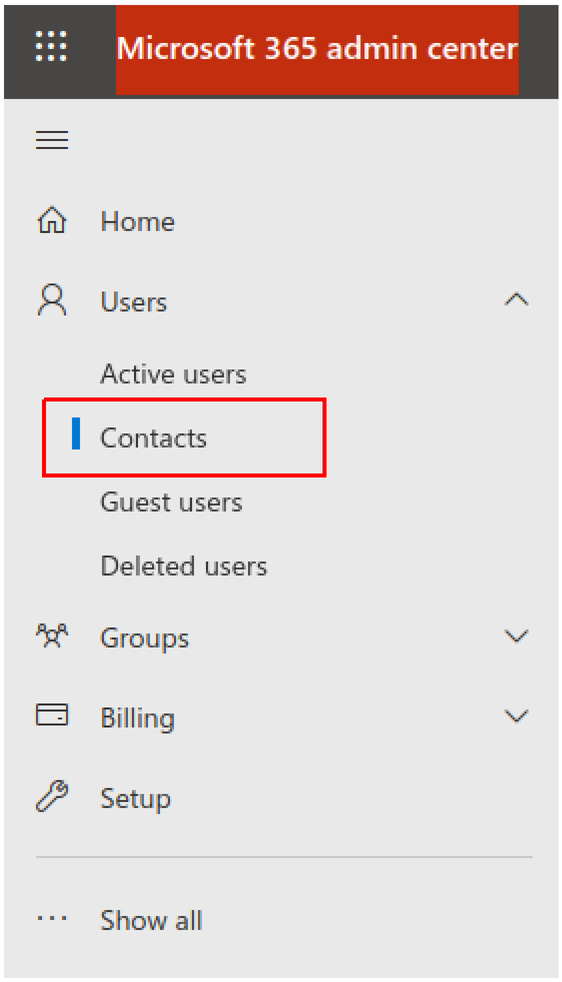


Fig. 78: Organization contacts location

- `/@m365/tenantname/users/usermail/contact/folderName/`
  - Default folder name is ‘Contact’
  - Each element has the special extension ‘.con’
- Organization contacts are stored in:
  - `/@m365/tenantname/users/organization/contact/Contacts/`
    - Each element has the special extension ‘.orgcon’

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the contact module.

In order to select the contact module, the common *service* parameter must be or contain the value *contact*.

Entities that can include mailboxes are: users.

Option	Require	Default	Values	Example	Description
contact_fil	No		Strings representing existing **contactFolders** for the given users separated by ‘,’	Customer Partners	Backup only specified **contactFolders** belonging to the selected users
contact_fil	No		Strings representing existing **contactFolders** for the given users separated by ‘,’	Personal	Exclude selected **contactFolders** belonging to the selected users
contact_fil	No		Valid regex	.*Company	Backup matching contact folders. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
contact_fil	No		Valid regex	.*Pla	Exclude matching contact folders from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
contact_or	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include Organizational contacts (common to all the people in the tenant). It is recommended to include organizational contacts only in a single backup, as part of the backup strategy, but not for every user.

- Note: In previous versions, instead of contact_files* parameters, it was possible to use files_* parameters. They are deprecated parameters now as we recommend to use module specific parameters in order to have better control and more possibilities inside a single fileset. However, jobs using files_* parameters will still work.

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of contact module restore operations:

- destination_user, destination_path, send_report, allow_duplicates, debug, foreign_container_generation

Use cases

The following restore scenarios are supported:

- Restore user contacts to the original user or to a different user
  - Restore parameters implied: **destination_user**
- Restore user contacts to the original contact folder group or to a different contact folder
  - Restore parameters implied: **destination_path**
- Restore user contacts to local file system (general restore **where** parameter must be set to a path)

Particularities are the same as email module.

For more details about the behavior of each parameter, please check the general section of restore parameters.

Fileset examples

Backup all contacts of one user:

Listing 127: **Fileset Example**

```
Fileset {
   Name = fs-m365-contacts-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=contact tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com"
   }
}
```

Backup all contacts from all users and include organization contacts:

Listing 128: **Fileset Example**

```
Fileset {
   Name = fs-m365-contacts-all
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=contact tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 contact_organization=yes"
   }
}
```

Sharepoint

Sharepoint is a special module in the Bacula Enterprise Microsoft 365 Plugin. While all the other modules rely only on the M365 Graph API to perform all of their operations, this module relies on the PnP PowerShell project, that simplifies many automation operations over Sharepoint Online APIs. More information about this project can be found here: https://github.com/pnp/powershell

Pnp Framework Usage Background

The current recommendation from Microsoft to handle any Azure Cloud data is to use the MS Graph API, and that is exactly the strategy that the M365 Plugin is following. However, the Graph Sharepoint API presents several limitations which do not allow the plugin to perform a full backup/restore operation at the time of the writing this document:

- It is not possible to create Sites
- It is not possible to get or upload web pages
- It is not possible to create some types of columns for Sharepoint lists
- It is not possible to create or update items including many types of allowed Sharepoint fields

That makes a full process of backup/restore not possible today using just the MS Graph API. On the other hand, Microsoft has native Sharepoint Online APIs. These APIs are expected to be entirely replaced in the future with MS Graph APIs, but today they are supporting more advanced functions for Sharepoint Online than MS Graph does.

Sharepoint Online APIs are fully featured and significantly complex. They may be used for many different purposes. In order to simplify many automation operations over Sharepoint Online using these APIs, as well as others over other M365 services, the PnP PowerShell project was created, and the Bacula Enterprise Microsoft 365 Plugin relies on it for doing its export/import procedures involving the parts not supported by Graph APIs.

In order to connect to Sharepoint using this framework, Bacula Enterprise M365 Plugin uses a connection through a certificate and a password (while for other modules, the connection is based in the secret). As a best practice, this certificate should be managed automatically and transparently through the add-app command and the config_file fileset parameter, as described in the authentication section of this document.

Sharepoint Protected Information

The information protected by this module is detailed below, distinguishing what is protected using each method:

- Sites Metadata (MS Graph)
  - Sites Sharing permissions (MS Graph)
- Site Templates (PnP Powershell) which includes:
  - Site metadata
  - Lists metadata
  - ListItems metadata
  - WebPages metadata
  - DocumentLibraries metadata

- Lists (MS Graph Object)
- ListItems (MS Graph Object)
  - ListItem versions
- Document Libraries Files (MS Graph Drive Items)
  - Files versions

The catalog stores the information with the following structure:

- `/@m365/tenantName/sites/siteName/`
  - siteName.site → Graph Object
  - siteName.site.template.xml → PnP template
  - drives/ → document libraries containing files
    * documents
    * sitePages
    * …
  - lists/ → Lists containing metadata
    * documents
    * sitePages
    * …

Version History

OneDrive and SharePoint can be configured to retain the history for files/items.

Sharepoint Backup General Guidelines

As detailed in the previous section, Sharepoint backups are composed by several elements. The most relevant element is the Sharepoint Site Template generated by the PnP framework. That xml file contains all the information needed to re-generate a complete Sharepoint site from scratch, including the different lists and their elements, but without the files included in the document libraries. Those files are backed up individually with their original name inside the 'drive' folder when the parameter 'sharepoint_include_drive_items' is activated.

For large sites and also when a large number of sites will be protected, it us a good practice to separate the backup of the sites in individual jobs, as well as separate each of them in two parts: - The first part is a Sharepoint site backup with 'sharepoint_include_drive_items' set to 'No' - The other part will be Drive module backup (setting the same 'site' parameter, with the same name).

The mentioned separation will have the site template in the Sharepoint backup, while document libraries files will be kept in the drive module backup.

Sharepoint backup can protect site images. It is important to note the images are usually stored inside the 'Site Assets' list, which is a system list. In order to backup system lists it is necessary to activate the 'include_system' backup parameter.

Sharepoint backup can also protect site pages. Howerver, similarly to site images, these ones are store in another system list, called 'Site Pages'. In order to include its contents, 'include_system' backup parameter is required. Note that Site Pages make up the SharePoint site, so the information in them is part of the site template. The purpose of having them separately is to version those files, allowing access to each one individually.

In the previous section, other files corresponding to lists and listitems, outside the sharepoint site template, are mentioned. The purpose of these files is to allow potential indinvidual operations of a single list or a single listitem. However, this information is already included in the site template, when it comes to full sharepoint site restore operations.

Sharepoint supports Full, Incremental and Differential backup levels. However, please note that Incremental and Differential levels will not download lists metadata (list items mentioned in the previous paragraph). Those elements are only included in Full backups, as they do not present modification dates to be properly handled.

Backup Parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the sharepoint module.

In order to select the sharepoint module, the common *service* parameter must be equals or be containing the value *sharepoint*.

Option	Require	Default	Values	Example	Description
lists	No		Valid names of existing lists from the selected sharepoint site separated by ‘,’	siteassets, documents	Backup only selected lists of the included sharepoint sites
lists_	No		Valid names of existing lists from the selected sharepoint site separated by ‘,’	sitepages	Backup all lists of the included sharepoint sites except the ones listed in this parameter
lists_	No		Valid regex	*.pages	Backup matching lists
lists_	No		Valid regex	^site.*	Exclude matching lists
share point	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include subsites in selected sites for sharepoint backup
share point	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include hidden sharepoint lists of selected sites for sharepoint backup
share point	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include system documentLibraries (sitepages, siteassets, etc).
share point	No	No	0, no, No, false, FALSE,	Yes	Include item former versions of every list item into the backup process

Note: sharepoint_certificate_path and sharepoint_certificate_password are available since Bacula Enterprise 18.0.8.

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of sharepoint module restore operations:

- send_report, local_path_json_objects, skip_versions, restore_share_permissions, debug -> Common restore parameters
- sharepoint_newsite_name, sharepoint_newsite_owner -> To be used for full site restore operations
- sharepoint_skip_system -> To determine whether to attempt restoring system lists. Many system lists cannot be overridden through the API. They exist to provide information for extraction and to create new sites with the previous two restore variables on this list. However, when restoring a full site and selecting all files,it's often best to skip the individual files in the backup that represent these lists, as they are already part of the site template
- destination_site, destination_path, sharepoint_list_name -> To be used for list item restore operations
- sharepoint_local_template_path -> To be used for full site restore operations where we need to apply a local filesystem template, previously restored or manually built

Use Cases

For restoring over Microsoft 365 purposes, this plugin can: - Restore an entire Sharepoint site to a new one - Restore regular files over existing document libraries - Restore listItems over an existing list (source list must be the same list kind than the destination list) - Restore a list over an existing site - Restore any element locally to the File Daemon

Bellow more details on each use case:

- Restore an entire site or a particular sub-site using a new site name (combines PnP and Graph actions)
  - Restore parameters implied: **sharepoint_newsite_name,sharepoint_newsite_owner**
  - ***Note:** The template provisioning operation can show some errors in the joblog because of the reasons explained in the general limitations section. Even if you see those messages, please check the restore result on Sharepoint Online as it will usually be correct as the joblog itself will suggest. If you still experience problems, please check the troubleshooting section.
  - ***Note:** This restore method will restore locally Graph objects like listItems or any other not applicable in the path pointed by: **local_path_json_objects**
- Restore files from document libraries to their original location or to a different document library (Graph based)
  - Refer to section restore-parameters-drive, as document libraries are managed as drive objects
- Restore listItems to their original list or to a list with the same structure in the same or different site (Graph based)
  - listItems can only contain simple fields (text, numbers..)

- Restore parameters implied: **sharepoint_list_name**

- Restore a list to its original site or to a different site (Graph based)
  - lists can only contain simple columns (text, numbers..)
  - Restore parameters implied: **destination_site**
- Restore any information to a local filesystem (general restore **where** parameter must be set to a path)

Please note that the following scenarios are not currently supported:

- To simultaneously restore a site and its subsites:
  - It is necessary to specifically select the parent site and unmark its subsites in order to restore the parent site
  - It is necessary to select only one subsite for each restore job in order to restore them
- Restoring a site template over an existing site is not supported. It is required to always assign **sharepoint_newsite_name,** and use with a non-existing name

For more details about the behavior of each parameter, please check the general section of restore parameters.

Restoring Large Site Templates from Incremental Backups

A backup of a SharePoint Site with Bacula Enterprise M365 Plugin includes the site template as the most important element. This XML file contains all the structure and all the data that is necessary to restore the site. M365 Plugin includes additional information to facilitate certain granular restore operations, such details are generally unnecessary when restoring a complete site, as outlined in the previous Use Cases section.

This site template is protected as a file, so it can be restored locally as any other file in Bacula. However, for a restore with the M365 service, this file is used to recreate the original site. As a result, the file needs to be created at the destination prior to restoring anything else.

When restoring a site from an Incremental backup, the file may not be the first element processed. Microsoft 365 Plugin includes the site template in the Catalog (as an internal `restore object`) to ensure it is the first element to restore. However, if the template is big enough (greater than `sharepoint_site_size_catalog_limit` parameter), this prioritization is not possible, and the job log will report it with a message as follows:

Listing 129: **Warning about site template size**

```
Site [site-name] template size is [size bytes] which is greater than the␣
↪limit established by the sharepoint_site_size_catalog_limit parameter:␣
↪[limit bytes]. The site template will be kept in the local file system. If␣
↪you remove it, to restore this site, you may need to perform a 2-step␣
↪restore process, restoring first the site template to the local filesystem.
```

As the message indicates, the template won't be sent to the Catalog (although it will be protected regularly as a file), instead it will be sent locally to the filesystem. If a restore operation from an Incremental backup is performed after the filesystem template is not found, it would be necessary to:

1. Run a local filesystem restore of the .xml site template.

If a restore attempt fails due to the absence of the template, the process will indicate a path where the XML can be placed for automatic inclusion in the next attempt. However, it is also possible to restore in any other place and then use the restore parameter: `sharepoint_local_template_path`.

2. Run a regular SharePoint site template restore.

Depending on the choice made in step 1, the restore will be exactly as described in the use cases, or it will require the use of `sharepoint_local_template_path` to specify the location where the template was restored.

Fileset Examples

Backup sharepoint site:

Listing 130: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-events
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"Company Events\" "
   }
}
```

A site can also be set up by its id. However, ids in Sharepoint can include several elements separated by ‘,’. As that character is already used to separate different elements (different sites for the site parameter), the plugin will expect the separation inside the site Id to be specified with ‘;’. Example:

Listing 131: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-events
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"mysite.sharepoint.com;55068291-a2cb-
↪144y-a401-03928474881141;acj948fb-9417-4bgf4-b31a-0erc6bd39ed6\" "
   }
}
```

Backup sharepoint site including hidden lists:

Listing 132: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-events-hidden
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"Company Events\" system_include=yes␣
```

(continues on next page)

(continued from previous page)

```
↪sharepoint_hidden_lists=yes"
   }
}
```

Backup sharepoint site and its sub-sites:

Listing 133: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-events-sub
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"Company Developments\" sharepoint_
↪subsites=yes "
   }
}
```

Backup a particular sharepoint sub-site:

Listing 134: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-devs
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"Company Developments/Doc Site\" "
   }
}
```

Backup specific sharepoint site lists:

Listing 135: **Fileset Example**

```
Fileset {
   Name = fs-m365-site-company-events-docs-travel
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 site=\"Company Events\" lists=\"documents,␣
↪travel lists\" "
   }
}
```

Onenote

The M365 Plugin Onenote module is able to protect any notebook belonging to users, groups, or sites of the base tenant. The pages of the notebook and any reference file are extracted in HTML format.

The information protected with this module is detailed below:

- For Users, groups or sites:
  - Notebook metadata
  - SectionGroup metadata
  - Section metadata
  - Page Html contents
  - Page Resources
    - Object files
    - Image files

Onenote module includes the following features:

- Incremental backup (it is done without the Delta function)
- It is possible to exclude images or file resources associated to pages from the backup
- It is possible to control the destination notebook, section, or section group at restore time (in addition to the general destination entity: user, group or site)

The catalog stores the information with the following structure:

- `/@m365/tenantName/entityKind/entityName/onenote/`
  - notebooks/notebookName
    - notebokName.note.book
      - sectionGroupName1
      - sectionGroupName2
      - ...
    - sectionA
      - page1.note.page.html

- resource1.note.res.png
- resource1.note.res.pdf
- …

Please, note that Onenote is probably the most sensitive module to throttling limits in Microsoft 365 APIs: https://docs.microsoft.com/es-es/graph/throttling

It is recommended to select the information to be protected of it and spread the jobs during time in order to avoid rejections for this reason.

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the onenote module.

In order to select the onenote module, the common *service* parameter must be or contain the value *onenote*.

Option	Re-quire	De-fault	Values	Exam-ple	Description
notebooks	No		Valid names of existing notebooks from the selected entities to backup separated by','	My notebook, John @ Work	Backup only selected notebooks of the included entities (users, groups, or sites)
note-books_exclu	No		Valid names of existing notebooks from the selected entities to backup separated by','	Quicknotes	Backup all notebooks of the included entities except the ones listed in this parameter
note-books_rege	No		Valid regex	*.buylist	Backup matching notebooks
note-books_rege	No		Valid regex	^Work.*	Exclude matching notebooks
onenote_exc	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Do not backup images from pages of selected notebooks
onenote_exc	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Do not backup objects (files that are not images) from pages of selected notebooks

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of onenote module restore operations:

- destination_user, destination_site, destination_group, send_report, local_path_json_objects, debug -> Common restore parameters
- notebook_name, notesection_name, notesection_group_name -> Control onenote destination element. Usually you only need to set up a non existing notebook_name to restore one entire notebook into a new one.

Use cases

The following restore scenarios are supported:

- Restore user, group, or site notebooks to their original entity or to a different entity
  - Restore parameters implied: **destination_user**, **destination_group**, **destination_site**
- To restore information to its original location, do not set the following parameters:
  - Restore over a new notebook: **notebook_name**
  - Restore over a new section: **notesection_name**
  - Restore over a new section group: **notesectiongroup_name**
- Restore to local file system (general restore **where** parameter must be set to a path)

Particularities about **foreign_container_generation** present the same behavior as the email module.

For more details about the behavior of each parameter, please check the general section of restore parameters.

Fileset examples

Backup all notebooks of a user:

Listing 136: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-onenote
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=onenote tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 user=\
↪"peter@mycompany.com\" "
        }
}
```

Backup specific notebooks of a site:

Listing 137: **Fileset Example**

```
Fileset {
        Name = fs-m365-site-notebooks-plan
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=onenote tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 site=\"Company␣
↪Events\" notebooks=places,planning"
        }
}
```

Do not backup files or images:

Listing 138: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-onenote-noimages
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=onenote tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 user=\
↪"peter@mycompany.com\" onenote_exclude_page_images=yes onenote_exclude_page_
↪files=yes"
        }
}
```

Teams

The M365 Plugin Teams module is able to protect one or more teams of the Microsoft 365 Teams service. Public and private teams are supported, as well as public or private channels.

The information protected with this module is detailed below:

- For groups:
  - Team metadata
  - Team settings metadata
  - Team members metadata
  - Team installed applications
  - Team channels (Public channels and private channels)
    - Channel metadata
    - Channel members for private channels
    - Channel tabs metadata
    - Channel messages - Chat messages - Hosted contents

The Teams module includes the following features:

- Incremental/Differential backup
- It is possible to exclude tabs or applications from the backup
- It is possible to select what channels to include into the backup
- It is possible to restore the data as a new Team in M365

Incremental backup of teams do not download Channel metadata of existing channels, installed apps or channel tabs. Those elements are only included in Full backups, as they do not present modification dates to handle.

The catalog stores the information with the following structure:

- `/@m365/tenantName/group/groupName/teams/`
  - team/teamName
    - teamName.team

· apps/

· app1.teamapp

· app2.teamapp

· …

· channels/

· channel1Name/ (tabs/tab1.tab, tabs/tab2.tab)

· 2021-04-19 10.58.52..J-123.1.channel.msg

· 2021-04-19 09.58.52..J-123.2.channel.msg

· 2021-04-19 09.58.52..J-123.2.file.1.host.cnt

· 2021-04-19 09.58.52..J-123.2.file.2.host.cnt

· channel2Name/

· …

It is important to note how the information is distributed in M365, since many modules are related to each other. When talking about MS Teams this is very important to understand, as some information accessible from Teams is stored in other M365 locations that are already managed through other parts of the M365 Plugin.

For example, it is possible to include attachments (which is a different concept than hosted contents) in chat messages. These files are stored into the drive unit associated to the group associated to the team, so that information should be backed up simply including the drive module in the fileset.

Some of the most important relations to have in mind are:

- A team is also directly attached to a group. So we need to use group parameters in the fileset in order to control 'what team' we want to backup.
- A team can use a notebook from onenote: This is the group onenote
- A team can store files: This is the onedrive unit for the group.
- A team can have a calendar: This is the calendar of the group.
- A team has a Team Site in Sharepoint: This can also be protected using the sharepoint module (and site parameters).

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the teams module.

In order to select the teams module, the common *service* parameter must be equals or be containing the value *teams*.

Option	Re-quire	De-fault	Values	Ex-ample	Description
teams_chan	No		Valid names of existing channels from the selected teams to backup separated by','	Gen-eral, De-velop-ment	Backup only selected channels of the included teams (from groups parameters)
teams_chan	No		Valid names of existing channels from the selected teams to backup separated by','	Fun Chan-nel	Backup all channels of the in-cluded teams (from groups param-eters) except the ones listed in this parameter
teams_chan	No		Valid regex	*.corp	Backup matching channels
teams_chan	No		Valid regex	^Pri-vate.*	Exclude matching channels
teams_apps	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Backup team apps from selected teams
teams_chan	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Backup team channel tabs from selected teams

Restore parameters

The list below shows the subset of restore parameters that can be used to control the behavior of teams module restore operations:

- send_report, debug -> Common restore parameters
- team_name -> Control destination team: If a non existing one is provided, a new team will be created. If an existing team name is provided, it will be used to restore channels/apps/tabs inside. If this parameter is not set, original team name will be considered and the same logic will be applied.
- team_channel_name -> Control destination channel name: If a non existing one is provided, a new channel will be created. If an existing channel name is provided, it will be used to restore messages inside. If this parameter is not set, original channel name will be considered and the same logic will be applied.
- team_private_channels_mode -> Decide what to do about restoring private channels:
  - DELEGATED: Default value. Will ask for delegated permissions of the private channel owner
  - PUBLIC: Will convert private channels into public ones
  - SKIP: Won't restore the private channel and will simply skip it
- team_guest_members_enable -> Enable guest members (asking for interaction if needed)

Use cases

The following restore scenarios are supported:

- Restore teams data into a newly generated Team: Set team_name with a new name.
- Restore channels into an existing Team with their original name: Set team_name to the existing team.
- Restore a channel with a new name into an existing Team: Set team_name to the existing team name and team_channel_name to a new name.
- Restore messages into an existing Team/Channel: Set team_name and team_channel_name to both existing objects.

If you restore teams data into a new generated team, users and dates will be preserved as they originally were. However, if you restore data of a team inside an existing team, please be aware that messages won't keep their original users or sent dates. This is only possible for new generated teams, existent ones do not allow to create such messages, so M365 Plugin adds a text prefix with that information, but all messages are actually sent with team owner user or channel owner (private channels case).

For private channels, if delegated mode is selected, they will be restored using the same strategy than messages into an existing team (previx with user and date). If they are converted to public, users and dates will be preserved, but the mode of the channel will be converted to public (this is due to Graph API limitations).

Please note that you can select particular channels or messages and restore them into the local filesystem (as json files) or into a new team/existing (it is not mandatory to select everything to have the restore working).

Microsoft Teams and Groups are associated entities. Therefore, the restore process will generate also a group with the same name of the Team.

If you want to restore also onedrive data, onenote data, etc, you could restore first teams data and then, restore any other data using as destination_group the name of the generated team (which is the same name the new generated group gets).

Fileset examples

Backup all teams from the tenant:

Listing 139: **Fileset Example**

```
Fileset {
        Name = fs-m365-all-teams
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=teams tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5"
        }
}
```

Backup specific team:

Listing 140: **Fileset Example**

```
Fileset {
        Name = fs-m365-site-notebooks-plan
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=teams tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 group=MyTeam"
        }
}
```

Backup specific team including drive files and onenote:

Listing 141: **Fileset Example**

```
Fileset {
        Name = fs-m365-site-notebooks-plan
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=teams,drive,onenote tenant=57uia43-
↪d107-17a2-a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5␣
↪group=MyTeam"
        }
}
```

Backup 'General' channel and do not backup tabs:

Listing 142: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-teams-noimages
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=teams tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 group=MyTeam␣
↪teams_channels=General teams_channel_tabs=no"
        }
}
```

Chat

The M365 Plugin Chat module is able to protect any chat where a given user is participating.

Please, note that this module needs Delegated permissions to work, so you will need to login using the Device Auth Flow
in order to have enough permissions to get the data: delegated-permissions.

The information protected with this module is detailed below:

- For Users:
  - Chat metadata
  - Chat tabs metadata
  - Chat installed apps metadata

- **Chat messages**
  - Hosted contents

Chat module includes the following features:

- Incremental/Differential backup
- It is possible to exclude tabs or applications from the backup

The catalog stores the information with the following structure:

- `/@m365/tenantName/user/userName/chats/`
  - chat1/ - chatName.chat
    - **apps/**
      - app1.chatapp
      - app2.chatapp
      - …
    - **tabs/**
      - tab1.chat.tab
      - tab2.chat.tab
      - …
      - 2021-04-19 10.58.52..J-123.1.chat.msg
      - 2021-04-19 09.58.52..J-123.2.chat.msg
      - 2021-04-19 09.58.52..J-123.2.file.1.host.cnt
      - 2021-04-19 09.58.52..J-123.2.file.2.host.cnt
      - …
  - chat2/ - …

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the chat module.

In order to select the chat module, the common *service* parameter must be or contain the value *chat*.

Option	Re-quire	De-fault	Values	Ex-ample	Description
chat_topics	No		Valid topics associated to existing chats from the selected user(s) to backup separated by','	ProjectA, Important	Backup only selected chats of the included users
chat_topics_	No		Valid topics associated to existing chats from the selected user(s) to backup separated by','	Fun Channel	Backup all chats of the included users except the ones listed in this parameter
chat_topics_	No		Valid regex	*.corp	Backup matching chats (by topic)
chat_topics_	No		Valid regex	^Private.*	Exclude matching chats (by topic)
chat_apps	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Backup chat apps from selected user chats
chat_tabs	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Backup team chat tabs from selected user chats
chat_exclude	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude any chat that has no messages inside
chat_exclude	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude chats of type 'group'
chat_exclude	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude chats of type 'one_on_one'

Restore parameters

The list below shows the subset of restore parameters that can be used to control the behavior of teams module restore operations:

- destination_user -> User that will be used to restore chats to it. Usually, it should be the same owner user of the messages to restore
- send_report, debug -> Common restore parameters
- chat_topic -> Control destination chat (add a new one)

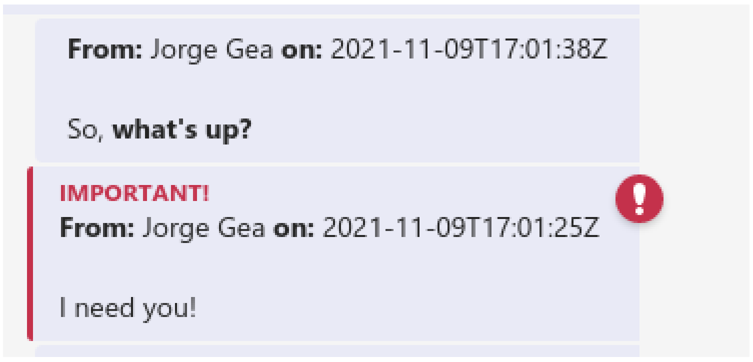
Use cases

Depending on the chat type, it is possible to:

- Group chats:
  - Restore chats as new ones, specifying a new topic with the restore variable 'chat_topic' (group chats)
  - Restore chat messages inside original chats (if they where deleted)
- One on one chats:
  - One on one chats can have only one instance per pair of users. Therefore, the restore can only create chat messages with the implied user in case they were deleted before

As with any other plugin, local filesystem restore is possible too and we would get all objects (chat, chatMessages. . .) as json files, except hosted contents that will be restored as the original files.

Please note that restore to M365 service function is not able to restore messages in the exact form that they were generated because of Microsoft security policy around this service. It is not possible to create a chat message with a different date or user than the real ones that are being used for the current connection. Therefore when the restore is performed what Bacula M365 Plugin does is to add a little header for each message containing that information, but the messages will be all generated using the 'destination_user' or the original user in case that variable was not used. Example:



From: Jorge Gea **on:** 2021-11-09T17:01:38Z

So, **what's up?**

IMPORTANT!
From: Jorge Gea **on:** 2021-11-09T17:01:25Z

I need you!

Fileset examples

Backup all chats of a user:

Listing 143: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-chat
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=chat tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 user=\
↪"peter@mycompany.com\" "
        }
}
```

Backup chats of all users, but do not backup tabs or installed apps:

Listing 144: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-chat-noimages
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=chat tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 chat_apps=no␣
↪chat_tabs=no"
        }
}
```

Tasks

Bacula Enterprise Microsoft 365 Plugin can protect M365 Todo Tasklists associated to users and M365 Planner tasks associated to Teams. Todo Taskslists will be considered in case user* parameters are defined, while Planner tasks will be considered in case group* parameters are defined.

For Todo Tasklists:

- It is possible to select what users to backup (user* parameters) and also the specific tasklists to include/exclude (tasklists* parameters).

The information protected by this module is detailed below:

- Users:
  - Task Lists
  - Tasks
- Groups:
  - Planner plan
  - Planner buckets
  - Planner tasks

This module includes the following features:

- Incremental/Differential backup

It is important to note here that Incremental or Differential backups won't take modified tasks. The reason is task objects do not store the information of last modification date, so we cannot determine if they were modified after creation. Therefore the recommendation is to run regularly Full backups to retrieve any modified task.

Please note the following limitatons:

- Todo Tasks LinkedResources are not supported. As of Bacula version 14.0, these objects are returned as empty entities from Graph API. This limitation is expected to be addressed in the future.
- Planner tasks won't keep their original order once they are restored and visualized through the Teams App or using the internet browser.

Catalog structure for tasks is presented below:

- User tasks are stored in:
  - `/@m365/tenantname/users/usermail/tasks/tasklist_name`
    - Default folder name is 'Tasks'
    - Each element has the special extension '.todo.task'
- Group planner tasks are stored in:
  - `/@m365/tenantname/group/groupdisplayname/tasks/plan_name/plan_bucket/planner_task_name`
    - Tasks have the special extension '.planner.task'. However other task related objects are stored also with different extensions. These objects are: task details, assignedto task board task format, progress task board task format and bucket task board task format

Please note that this module needs **Delegated permissions** to work, so it is necessary to login using the Device Auth Flow in order to have enough permissions to get the data: delegated-permissions

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the tasks module.

In order to select the tasks module, the common *service* parameter must be equals or be containing the value *tasks*.

Entities that can include mailboxes are: users.

Option	Require	Default	Values	Example	Description
tasklist	No		Strings representing existing todo **tasklists** for the given users separated by ‘,’	ProjectA ProjectB	Backup only specified **tasklists** belonging to the selected users
tasklist	No		Strings representing existing todo **tasklists** for the given users separated by ‘,’	Personal	Exclude selected **tasklists** belonging to the selected users
tasklist	No		Valid regex	.*Company	Backup matching tasklists. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
tasklist	No		Valid regex	.*Pla	Exclude matching tasklists from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
plannerplans	No		Strings representing existing planner **plans** for the given groups separated by ‘,’	ProjectA ProjectB	Backup only specified **plannerplans** belonging to the selected groups
plannerplans_	No		Strings representing existing planner **plans** for the given groups separated by ‘,’	Personal	Exclude selected planner **plans** belonging to the selected groups
plannerplans_	No		Valid regex	.*Company	Backup matching planner plans. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
plannerplans_	No		Valid regex	.*Pla	Exclude matching planner plans from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of tasks module restore operations:

- destination_user, destination_group, send_report, debug, tasklist_name, tasklist_skip_sharedwithme

Use cases

The following restore scenarios are supported:

- Restore task lists and their tasks into to their original user or to a different user
  - Restore parameters implied: **destination_user**
  - Destination tasklist (new or existent one): tasklist_name
  - Skip or not shared tasklists: tasklist_skip_sharedwithme
- Restore planner plans and their tasks into to their original groups (teams) or to a different group (team)
  - Restore parameters implied: **destination_group**
  - Destination plan (new or existent one): plan_name
  - Create or not associated tab: plan_create_tab

For more details about the behavior of each parameter, please check the general section of restore parameters.

Fileset examples

Backup all todo tasks of one user:

Listing 145: **Fileset Example**

```
Fileset {
   Name = fs-m365-tasks-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com"
   }
}
```

Backup todo tasklist 'work' of one user:

Listing 146: **Fileset Example**

```
Fileset {
   Name = fs-m365-tasks-adelev
   Include {
      Options { signature = MD5 }
```

(continues on next page)

(continued from previous page)

```
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com␣
↪tasklists=work"
   }
}
```

Backup all todo tasks from all users

Listing 147: **Fileset Example**

```
Fileset {
   Name = fs-m365-tasks-all
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5"
   }
}
```

Backup all planner plans of one group:

Listing 148: **Fileset Example**

```
Fileset {
   Name = fs-m365-planner-dev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 group=DevTeam"
   }
}
```

Backup all planner plans from all groups

Listing 149: **Fileset Example**

```
Fileset {
   Name = fs-m365-planner-all
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5"
   }
}
```

Backup planner plan ‘product’ of one user:

Listing 150: **Fileset Example**

```
Fileset {
   Name = fs-m365-tasks-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=tasks tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-
   eb5d-42nf-bac7-7b019fd284g5 group=DevTeam plannerplans=product"
   }
}
```

Activity

The M365 Plugin Activity module protects various reports concerning user activity and usage within m365.

The information protected with this module is detailed below:

- m365GroupsActivityStorage
- m365GroupsActivityDetail
- m365ActiveUserDetail
- m365ActivationsUserDetail
- m365AppUserDetail
- yammerDeviceUsageUserDetail
- yammerActivityUserDetail
- yammerActivityGroupsDetail
- teamsDeviceUsageUserDetail
- teamsActivityUserDetail
- teamsActivityGroupDetail
- skypeDeviceUsageUserDetail
- skypeActivityUserDetail
- sharepointSiteUsageStorage
- sharepointSiteUsageDetail
- sharepointActivityUserDetail
- oneDriveUsageStorage
- oneDriveUsageAccountDetail
- oneDriveActivityUserDetail
- mailboxUsageStorage
- mailboxUsageDetail
- emailAppUsageUserDetail

- emailActivityUserDetail

The catalog stores the information with the following structure:

- `/@m365/tenantName/activity/`
  - `reportName.csv`

Note: The activity module is available since Bacula 18.0.1.

Backup Parameters

The following list shows the specific backup parameters that can be set up in order to control the behavior of the activity module.

In order to select the chat module, the common *service* parameter must be or contain the value *activity*.

Option	Require	Default	Values	Example	Description
activity_excha	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include exchange/mailbox-related reports
activity_onedr	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include onedrive-related reports
activity_share	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include sharepoints-related reports
activity_teams	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include teams-related reports
activity_yamm	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include yammer-related reports
activity_skype	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include skype-related reports
activity_gener	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include general m365 service-related reports
activity_perio	No	180	7, 30, 90, 180	90	Period of days 'from' used to generate the report. Only documented values are allowed
activity_from	No		date formatted like: yyyy-MM-dd_HH:mm:ss	2024-01-01_00:00:	Date 'from' used to generate the reports. This method only permits values from the last 30 days

One recommended strategy to protect the information of these reports is to define a supported period, such as 1 week. Then, activity_period_days can be set to 7 and run weekly. This way, the reports will contain only updated information each time, avoiding duplicate entries.

Restore Parameters

Restore of reports can only be performed to a local filesystem to the File Daemon. Therefore, 'where' restore parameter should point to the destination directory where the information needs to be restored.

Fileset Examples

Backup all reports:

Listing 151: **Fileset Example**

```
Fileset {
        Name = fs-m365-reports
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=activity tenant=57uia43-d107-17a2-
↪a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5"
        }
}
```

Backup reports of the last week:

Listing 152: **Fileset Example**

```
Fileset {
        Name = fs-m365-peter-chat-noimages
        Include {
                Options { signature = MD5 }
                Plugin = "m365: service=activity tenant=57uia43-d107-17a2-
↪a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5 activity_
↪period_days=7"
        }
}
```

Special Features

In the following section, special features and behaviors are detailed.

Item Versions

OneDrive and SharePoint can be configured to retain the history for files/items. Depending on the service and configuration, a new version can be created for each edit, each time the file is saved, manually, or never. Previous versions of a document may be retained for a finite period of time depending on admin settings which may be unique per user or location. This feature is usually enabled and the information may be accessed through the 'Version history' option available once a file has been selected.

Bacula Enterprise Microsoft 365 Plugin is able to backup this information if a special **drive_version_history.** backup parameter is activated.

Versions are backed up in a special way:

- They are backed up as a regular file. This means a version has its own full metadata as the parent file itself has. All the metadata is the same as the file contains, except for size, dates and name.
- The name of the file is modified, so at restore time you can see the version number and the version date in the filename. Example:
  - Parent file: myDoc.doc
  - Versions:
    * myDoc###v25.0_2021-01-19_234537.doc
    * myDoc###v24.0_2021-01-17_212537.doc
    * myDoc###v23.0_2021-01-12_104537.doc
    * ...
    * *Notice that the extension of the file is kept in order to easily identify a possible name modification in OneDrive, once the file is restored
- Versions are not restored by default. You need to disable the special restore parameter 'drive_skip_versions', setting it to 0.

File versions are backed up in all backup levels (Full, Incremental, Differential), this means you can track all the changes of the files in your backup. For example, every Incremental run is going to backup only the new modified versions since the last Full or Incremental execution.

Here is an example of a file backed up, listed in a restore session:

Listing 153: **deltaLink in restore session**

```
Automatically selected Fileset: FS_M365_DRIVE
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
| jobid | level | jobfiles | jobbytes    | starttime           | volumename ␣
↪     |
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
|    11 | F     |      190 | 332,978,505 | 2021-01-22 10:39:34 | TEST-2021-01-
↪22:0 |
|    12 | I     |        1 |         550 | 2021-01-22 10:43:05 | TEST-2021-01-
↪22:0 |
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
You have selected the following JobIds: 11,12

Building directory tree for JobId(s) 11,12 ...
+++++++++++++++++++++++++++++++++++++++++++++++
188 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @m365/baculaenterprise/users/adelev@baculaenterprise.onmicrosoft.com/
↪drive/root:
```

(continues on next page)

(continued from previous page)

```
cwd is: /@m365/baculaenterprise/users/adelev@baculaenterprise.onmicrosoft.com/
↪drive/root:/
$ ls
Docs/
pluginTest.drive.deltaLink
sharedWithMe/
versionedSampleWeb###v1.0_2020-11-25_153507.html
versionedSampleWeb###v2.0_2020-11-25_153507.html
versionedSampleWeb###v3.0_2020-12-02_180612.html
versionedSampleWeb###v4.0_2020-12-02_180612.html
versionedSampleWeb###v5.0_2020-12-03_134422.html
versionedSampleWeb###v6.0_2020-12-02_180612.html
versionedSampleWeb.html
docVersioned###v1.0_2020-11-26_123244.txt
docVersioned###v2.0_2020-11-26_123244.txt
docVersioned###v3.0_2020-12-02_180613.txt
docVersioned###v4.0_2020-12-02_180613.txt
docVersioned.txt
$
```

Note: It is important to keep in mind that versions have no Delta function and cannot be filtered by date. Therefore, the process to decide if a version needs to be backed up or not requires the plugin to walk through all existent versions of a modified item. In some situations this could have some undesired impact on backup performance.

Delta Backup

The Microsoft Graph API provides a Delta function to track changes of some objects. Bacula Enterprise Microsoft 365 Plugin uses this function in order to speed up Incremental/Differential processes. Please note that Delta function is not a mandatory requirement and that Incremental or Differential backups will function also with the services that currently do not support it.

Delta function has some important characteristics:

- In OneDrive entities backup it can only be used for full entities (full user, full group, or full site). This means that selecting specific paths to backup will not trigger the Delta function.
- Delta tokens can expire at some point, or even become invalid due to internal Microsoft issues. If this situation happens, the plugin will try to start a new Delta cycle
- Any situation where the Delta function cannot be used will trigger a regular Full/Inc/Diff where every element is listed and selected or discarded according to the item dates.

The Delta backup cycle is described below:

- Full backup: All entity elements are backed up. A token (token_1) is generated and this token is stored locally by the FD.
- Incremental 1 backup: token_1 is used to retrieve changes since token_1's generation so every change is backed up. A new token is generated and stored locally by the FD.
- Incremental 2 backup: token_2 is used to retrieve changes since token_2's generation so every change is backed up. A new token is generated and stored locally by the FD.

- And so on…

Tokens are stored in an file placed in a path defined by the **path** parameter of the plugin. The name is: jobname.deltaLink

The file stores tokens required for every execution and it is renewed (emptied) during every Full backup execution.

This file is also backed up in the backup itself, so it can be restored manually, before an Incremental/Differential execution in case it was lost and in case you don't want to run a Full backup again.

Services supporting Delta backup are: **email**, **onedrive** and **contacts**.

Here we can see an example of the contents of the file, with 3 executions and one user entity involved. The structure is tree-based, so it is easy to understand what would be generated in case of backing up other services or entities:

Listing 154: **deltaLink**

```
{
"jobName": "M365-DRIVE-ADELE-BACK",
"deltaServices": {
      "DRIVE": {
         "entities": {
            "4bfec6ba-6e0c-455f-a86d-1dbfdd1c5754": {
               "id": "4bfec6ba-6e0c-455f-a86d-1dbfdd1c5754",
               "name": "adelev@baculaenterprise.onmicrosoft.com",
               "containers": {
                  "b!7Tf3z7izSES7kjOHD-YM-
↪0kScgNXL6lFnL4O2LWLMy41XlWc2E0pTKfPzlKxhztE": {
                  "id": "b!7Tf3z7izSES7kjOHD-YM-
↪0kScgNXL6lFnL4O2LWLMy41XlWc2E0pTKfPzlKxhztE",
                  "description": "onedrive",
                  "deltaEntries": [
                     {
                     "job": "M365-DRIVE-ADELE-BACK.2021-04-13_13.42.42_21",
                     "date": "Apr 13, 2021 1:42:42 PM",
                     "delta":

↪"MzslMjM0OyUyMzE7Mzs5YzU1NWUzNS00ZGQ4LTRjMjktYTdjZi1jZTUyYjE4NzNiNDQ7NjM3NTM5MTA5NjczMzAwM
7JTIzaXRCQmNGclZjVGpya0x1YmZhdEpWWHpWdlolMjUyRkxPN1FJYmM4OXNGJTI1MkZZZkxZJTI1M0Q7JTIzMA
↪"
                     },
                     {
                     "job": "M365-DRIVE-ADELE-BACK.2021-04-13_13.43.47_22",
                     "date": "Apr 13, 2021 1:43:47 PM",
                     "delta":

↪"MzslMjM0OyUyMzE7MTtjZmY3MzdlZC1iM2I4LTQ0NDgtYmI5Mi0zMzg3MGZlNjBjZmI7NjM3NTM5MTEwMzExNDAwM
                                             ␣
↪7JTIzaXRCQmNGclZjVGpya0x1YmZhdEpWWHpWdlolMjUyRkxPN1FJYmM4OXNGJTI1MkZZZkxZJTI1M0Q7JTIzMA
↪"
                     },
                     {
                     "job": "M365-DRIVE-ADELE-BACK.2021-04-13_13.44.39_23",
                     "date": "Apr 13, 2021 1:44:39 PM",
```

(continues on next page)

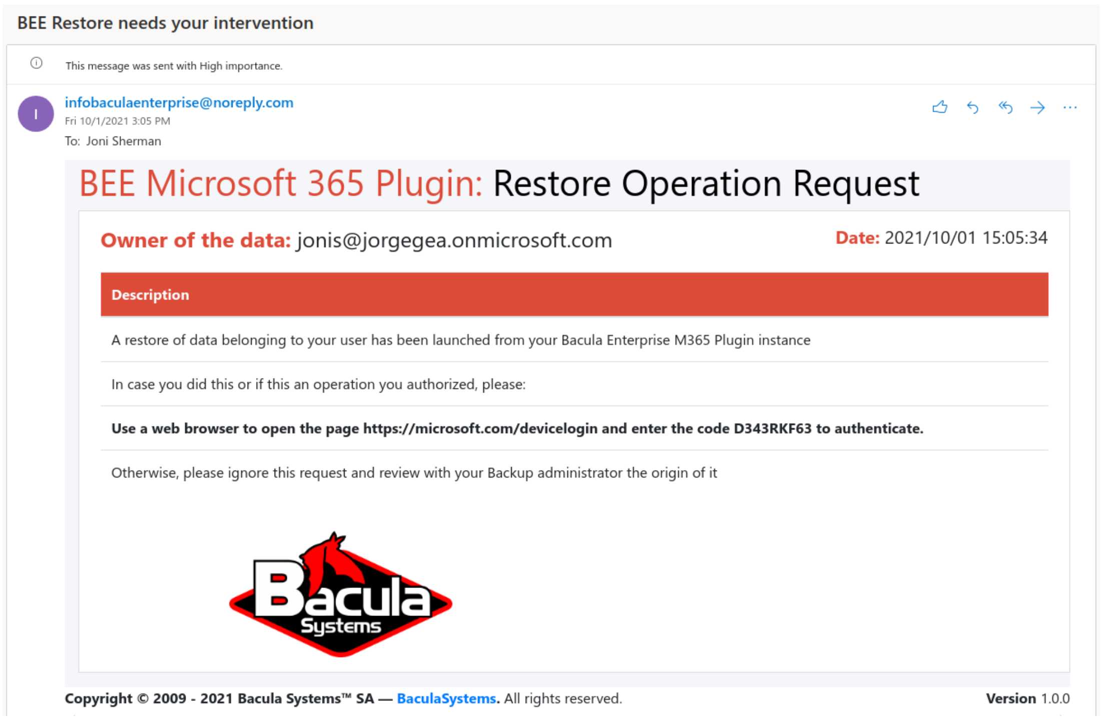
(continued from previous page)

```
                        "delta":
↪"MzslMjM0OyUyMzE7MTtjZmY3MzdlZC1iM2I4LTQ0NDgtYmI5Mi0zMzg3MGZlNjBjZmI7NjM3NTM5MTEwODMwMTAwMDAwOzI4N
7JTIzaXRCQmNGclZjVGpya0x1YmZhdEpWWHpWdlolMjUyRkxPN1FJYmM4OXNGJTI1MkZZZkxZJTI1M0Q7JTIzMA
↪"
                        }
                    ]
                }
            }
        }
    }
}
```

Data Owner Restore Protection

Bacula Systems is aware about one of many privacy concerns that may arise when tools like this M365 Plugin enables the possibility to backup and restore data coming from different users, so the backup administrator can restore potentially private data at will. To address this concern, **Bacula Enterprise** Microsoft 365 Plugin includes the **Owner restore protection** feature.

This feature is enabled at configuration time with the parameter 'owner_restore_protection' (please check the Configuration section of this document for further information). Once it is enabled, any restore operation to a different user than the original owner will trigger the intervention of the owner of the data. The restore job will be paused and a message asking to enter a Microsoft 365 page and enter a code will be logged in the joblog. A similar message will also be sent via email to the affected user:



BEE Restore needs your intervention

This message was sent with High importance.

infobaculaenterprise@noreply.com
Fri 10/1/2021 3:05 PM
To: Joni Sherman

BEE Microsoft 365 Plugin: Restore Operation Request

Owner of the data: jonis@jorgegea.onmicrosoft.com **Date:** 2021/10/01 15:05:34

Description

A restore of data belonging to your user has been launched from your Bacula Enterprise M365 Plugin instance

In case you did this or if this an operation you authorized, please:

Use a web browser to open the page https://microsoft.com/devicelogin and enter the code D343RKF63 to authenticate.

Otherwise, please ignore this request and review with your Backup administrator the origin of it

Copyright © 2009 - 2021 Bacula Systems™ SA — BaculaSystems. All rights reserved. **Version** 1.0.0

Fig. 79: Restore Operation Request example

If the user does not confirm the operation within 15 minutes, the restore will fail and no data will be restored. However, if the user knows that operation and wants to approve it, as soon as the owner credentials are confirmed, the restore will resume and the data will be processed as usual and restored to any configured destination.

Note: This feature is available starting with version 14.0 of Bacula Enterprise

---

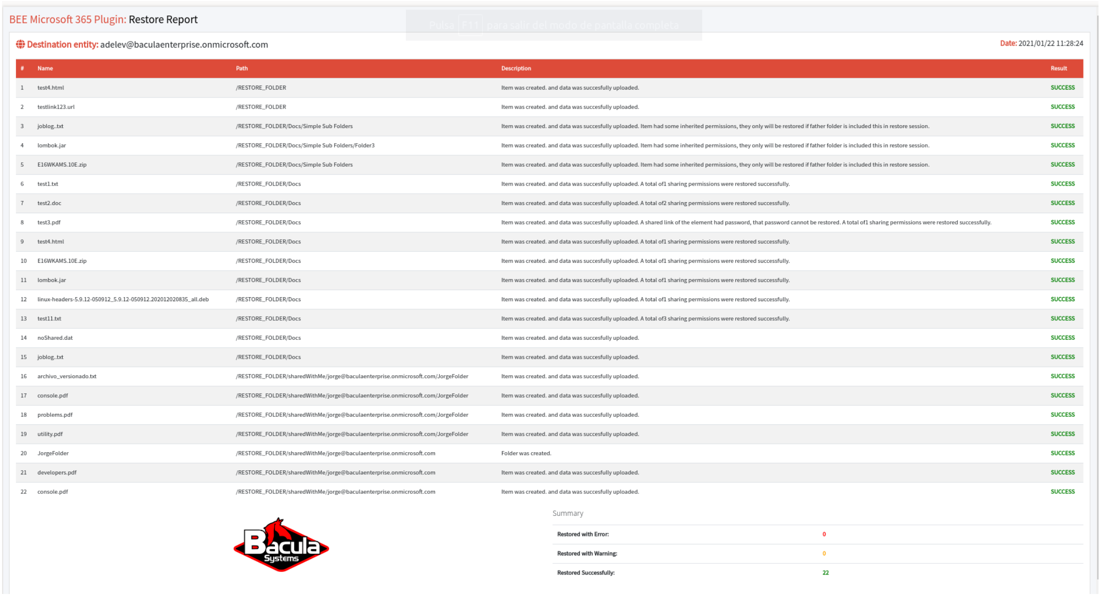
User Restore Report

Files and emails can represent very sensitive information for end-users. For that reason, information included in backup/restore logs is not exhaustive by default. For example, email restores do not include information such as the subject or sender when they are displayed in the backup log. However, for reporting and controlling purposes, the information of what has been exactly restored, what permissions have been applied, and other information can be useful and necessary for the affected user.

Bacula Enterprise Microsoft 365 Plugin includes an option to generate a restore report in the affected user's service (Drive or Mailbox). The restore report contains detailed information about the items that have been restored successfully, if any of them had any trouble during the restore, and it also reports the date when the action was performed.

The generation of the report can be enabled/disabled in the bconsole restore session. If enabled, depending on the service, the report can generate an HTML file or an email in the Inbox of the affected user.

The image below shows an example report from a Drive restore session:



BEE Microsoft 365 Plugin: Restore Report

Pulsa F11 para salir del modo de pantalla completa

Destination entity: adelev@baculaenterprise.onmicrosoft.com

Date: 2021/01/22 11:28:24

#	Name	Path	Description	Result
1	test4.html	/RESTORE_FOLDER	Item was created. and data was succesfully uploaded.	SUCCESS
2	testlink123.url	/RESTORE_FOLDER	Item was created. and data was succesfully uploaded.	SUCCESS
3	joblog..txt	/RESTORE_FOLDER/Docs/Simple Sub Folders	Item was created. and data was succesfully uploaded. Item had some inherited permissions, they only will be restored if father folder is included this in restore session.	SUCCESS
4	lombok.jar	/RESTORE_FOLDER/Docs/Simple Sub Folders/Folder3	Item was created. and data was succesfully uploaded. Item had some inherited permissions, they only will be restored if father folder is included this in restore session.	SUCCESS
5	E16WKAMS.10E.zip	/RESTORE_FOLDER/Docs/Simple Sub Folders	Item was created. and data was succesfully uploaded. Item had some inherited permissions, they only will be restored if father folder is included this in restore session.	SUCCESS
6	test1.txt	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of1 sharing permissions were restored successfully.	SUCCESS
7	test2.doc	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of2 sharing permissions were restored successfully.	SUCCESS
8	test3.pdf	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A shared link of the element had password, that password cannot be restored. A total of1 sharing permissions were restored successfully.	SUCCESS
9	test4.html	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of1 sharing permissions were restored successfully.	SUCCESS
10	E16WKAMS.10E.zip	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of1 sharing permissions were restored successfully.	SUCCESS
11	lombok.jar	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of1 sharing permissions were restored successfully.	SUCCESS
12	linux-headers-5.9.12-050912_5.9.12-050912.202012020835_all.deb	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of1 sharing permissions were restored successfully.	SUCCESS
13	test11.txt	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded. A total of3 sharing permissions were restored successfully.	SUCCESS
14	noShared.dat	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded.	SUCCESS
15	joblog..txt	/RESTORE_FOLDER/Docs	Item was created. and data was succesfully uploaded.	SUCCESS
16	archivo_versionado.txt	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com/JorgeFolder	Item was created. and data was succesfully uploaded.	SUCCESS
17	console.pdf	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com/JorgeFolder	Item was created. and data was succesfully uploaded.	SUCCESS
18	problems.pdf	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com/JorgeFolder	Item was created. and data was succesfully uploaded.	SUCCESS
19	utility.pdf	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com/JorgeFolder	Item was created. and data was succesfully uploaded.	SUCCESS
20	JorgeFolder	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com	Folder was created.	SUCCESS
21	developers.pdf	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com	Item was created. and data was succesfully uploaded.	SUCCESS
22	console.pdf	/RESTORE_FOLDER/sharedWithMe/jorge@baculaenterprise.onmicrosoft.com	Item was created. and data was succesfully uploaded.	SUCCESS

Summary

Restored with Error:	0
Restored with Warning:	0
Restored Successfully:	22

Fig. 80: Restore Drive Example Report

Export to PST

PST - Outlook Personal Folders File Format - is a representation of an Exchange Object store. It consists of a hierarchical structure of Folder objects that can contain Contacts, Tasks, Events, or Message objects. These Message objects can further contain Attachment objects. The properties of Folder objects, Message objects, and Attachment objects store all the relevant information about each item.

Bacula Enterprise Microsoft 365 Plugin has the ability to backup and restore Exchange emails and their attachments in a cross-platform format through the email_mime option, as it is discussed in the chapter dedicated to the email module. However, some users find PST format more handy to perform export/import operations. Microsoft 365 Plugin also provides an extension that is capable to perform a transformation from restored Exchange objects to a single PST file.

This extension is served through the extra m365 package named 'bacula-m365-pst-plugin', available in the same platforms as the m365 Plugin. This extension is built over an specific technology that requires an extra license. Contact our sales or support team in order to inquire about gaining access and obtaining all the necessary information.

PST Export Installation

Once you have the package, use the appropriate package manager to install it depending on your platform.

Some examples:

```
rpm -ivh bacula-enterprise-m365-pst-plugin_18.0.1.rpm
```

```
yum localinstall bacula-enterprise-m365-pst-plugin_18.0.1.rpm
```

```
yum install bacula-enterprise-m365-pst-plugin
```

```
apt install bacula-enterprise-m365-pst-plugin
```

```
dpkg -i bacula-enterprise-m365-pst-plugin_18.0.1.deb
```

The package will install the following elements:

- Two extra jar libraries in /opt/bacula/lib
- Export binary (m365pst) that invokes the jar files in /opt/bacula/bin. This is the binary to run in order to run the export function

PST Export Usage

Bacula Enterprise Microsoft 365 Plugin PST Export tool is designed to work only with data restored from backups done by M365 Plugin.

The general procedure to have a PST file with the desired contents is:

1. Perform M365 backup of the desired contents (compatible modules are: email, calendar, contact and tasks)
2. Run a restore in a local folder (where=/my/local/folder)
3. Use the m365pst export tool with the path of the locally restored files

This tool accepts 2 main parameters: - src: It must point to the folder containing the restored M365 files. - out: It must point to a path/name of the resulting PST file with the desired name

Below there is an example of the usage of this tool:

Listing 155: **m365pst running**

```
/opt/bacula/bin/m365pst -src /my/local/folder -out mypstfile.pst
```

Please, note the parameters use single '-' and the value of each need to be placed after a space.

An example of successful output:

Listing 156: **m365pst running**

```
Will read from source path: /home/bacula/pstdocs/mytenant/
Will write in: /home/bacula/pstdocs/mypstfile.pst
Initializing PST File in /home/bacula/pstdocs/mypstfile.pst...
Initialization of /home/bacula/pstdocs/mypstfile.pst completed
Organizing file tree...
File tree successfully read
Linking attachments to father objects...
Attachments successfully linked
Adding messsages with attachments...
Messsages with attachments successfully added
Adding rest rest of the messages...
Rest of the messages successfully added
Adding contacts...
Contacts successfully added
Adding messsages with attachments...
Events with attachments successfully added
Adding events...
Events successfully added
Adding TodoTasks...
TodoTasks successfully added
PST successfully generated in /home/bacula/pstdocs/mypstfile.pst with: 57␣
↪folders - 508 messages - 93 message attachments - 20 contacts - 952␣
↪calendar events - 10 event attachments - 133 todo tasks
```

Installation

The Bacula File Daemon and the Microsoft 365 Plugin need to be installed on the host that is going to connect to the cloud based services. The plugin is implemented over a Java layer, therefore it can be deployed on the platform better suited for your needs among any of the officially supported platforms of **Bacula Enterprise** (RHEL, SLES, Debian, Ubuntu, Windows, etc).

Please, note that you may want to deploy your File Daemon and the plugin on a virtual machine directly deployed in Azure Cloud in order to reduce the latency between it and the Microsoft Graph API and experience modest performance gains. However, this option is **only recommended in case of having a very stable connection between the File Daemon and the Storage Daemon**, which means a special, dedicated connection with Azure or when the Storage Daemon is deployed in the cloud as well. This is not usually the case - thus the data needs to traverse the Internet with standard and shared connections from the File Daemon to the Storage Daemon. Disconnections while transmitting data between these two daemons may make jobs fail and cause large timeouts that are difficult to manage and stabilize. The best strategy strategy for this kind of scenarios is to deploy the File Daemon and the Plugin to the same host as the destination Storage Daemon is installed. This way, disconnections between the two daemons will not happen, while disconnections between the FD and M365 will be transparently recovered (when possible), so jobs will finish successfully.

The system must have Java >= 8 installed (openjdk-8-jre for example) and the Java executable should be available in the system PATH.

The Sharepoint module depends on the Powershell and the PnP Powershell modules. Therefore, they also need to be installed before installing the Bacula packages (see section *PnP.Powershell* below).

PnP.Powershell

Install PowerShell

In order to install PowerShell it is necessary to follow the instructions for the particular OS involved which may be found in the github site of the project:

- https://github.com/PowerShell/PowerShell

For example, if using Debian, these are the instructions:

- https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-linux?view=powershell-7.1#debian-9

The procedure is shown below, some dependencies are installed, a repository is added, and then the apt package manager is used to install the tool:

Listing 157: **Install PowerShell**

```
# Install system components
sudo apt-get update
sudo apt-get install -y curl gnupg apt-transport-https

# Import the public repository GPG keys
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -

# Register the Microsoft Product feed
sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/
↪microsoft-debian-stretch-prod stretch
main" > /etc/apt/sources.list.d/microsoft.list'

# Update the list of products
sudo apt-get update

# Install PowerShell
sudo apt-get install -y powershell

# Start PowerShell
pwsh
```

Install PnP.Powershell

In order to install the PnP.Powershell module, once Powershell is already installed, we simply need to run the command:

- Install-Module -Name “PnP.PowerShell”

Inside a Powershell session. Below we provide an example:

Listing 158: **Install PnP.PowerShell**

```
yourworkstation:~$ pwsh
PowerShell 7.2.0
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS /home/john> Install-Module -Name "PnP.PowerShell"
```

Bacula Packages

We are taking Debian Buster as the example base system to proceed with the installation of the **Bacula Enterprise** Microsoft 365 Plugin. In this system, the installation is most easily done by adding the repository file suitable for the existing subscription and the Debian version utilized. An example would be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 159: **APT**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪buster-64/ buster main
deb https://www.baculasystems.com/dl/@customer-string@/debs/m365/@version@/
↪buster-64/ buster m365
```

After that, a run of apt update is needed:

Listing 160: **APT install**

```
apt update
```

Then, the plugin may be installed using:

Listing 161: **APT install**

```
apt install bacula-enterprise-m365-plugin
```

The plugin has two different packages implied that should be installed automatically with the command shown:

- bacula-enterprise-m365-plugin
- bacula-enterprise-m365-plugin-libs

Alternately, manual installation of the packages may be done after downloading the poackages from your Bacula Systems provided download area, and then using the package manager to install. An example:

Listing 162: **APT install**

```
dpkg -i bacula-enterprise-*
```

The package will install the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-m365-plugin-x.x.x.jar and bacula-m365-plugin-libs-x.x.x.jar). Please note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.

Note: Version in Jar Name

Version is included in the name of .jar files from Bacula Enterprise version 14.0.4. Before that, libraries were composed by: bacula-m365-plugin.jar, bacula-meta-plugin-1.0.0.jar and bacula-m365-plugin-libs-1.0.0.jar

---

- Plugin connection file (m365-fd.so) in the plugins directory (usually /opt/bacula/plugins)
- Backend file (m365_backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-m365-plugin-x.x.x.jar file in order to launch it, even thought usually we should have only one file.
- A collection of powershell files used in the Sharepoint module in /opt/bacula/bin.

Configuration

Authorization

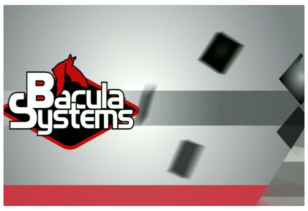
The first step in order to use the **Bacula Enterprise** Microsoft 365 Plugin is to authorize it to handle data of the target tenant to backup.

There are two possible strategies in order to allow the communication between the Bacula Enterprise Plugin and your tenant:

- **Method A (DEPRECATED): Common app model**
    - Register the pre-existing Bacula Systems **bacula-m365-plugin** Azure AD app into your tenant.
    - The communication will happen through this multi-tenant application.
    - Application Id and associated secrets are internal to the plugin.
    - Microsoft Graph limits associated to an application are common for everyone using this application (multi-customer).
    - For future new permissions you just need to click on 'Grant permissions' from Azure AD enterprise apps section
- **Method B (RECOMMENDED): Standalone app model**
    - Register the pre-existing Bacula Systems **bacula-m365-registratror** Azure AD app into your tenant. Then add a standalone application in your tenant.
    - Then add your own standalone application in your tenant calling the appropriate automatic command from bconsole

- A **bacula-m365-plugin-standalone** Azure AD application will be created specifically for your tenant. The communication will happen through it.
- Application Id and associated secrets need to be correctly set and they can be managed by you.
- Microsoft Graph limits associated to an application are specific to your standalone application
- For future new permissions you need to re-run the bconsole add-app command (which uses bacula-m365-registrator) or do it yourself manually

To walk through the process described in Method B with the help of a video, click on the image below:



Note: Authentication Method B is only available from Bacula Enterprise 12.8.2

The first method is simpler and faster to setup, however it is only advised for **testing purposes**. The second method needs a few extra variables to manage and it is recommended for medium or large environments. It is more secure and it can offer better performance.

Backup and restore operations will be using in general the 'Application permissions' model, where the application has enough privileges to perform all the operations without impersonating any user. However, some specific modules need to employ 'Delegated permissions'. To know more about them, please go to delegated-permissions

---

Note: Starting from Bacula Enterprise version 14.0 you can also perform these authorization tasks directly using BWeb, to see more details, please go to section bweb-console

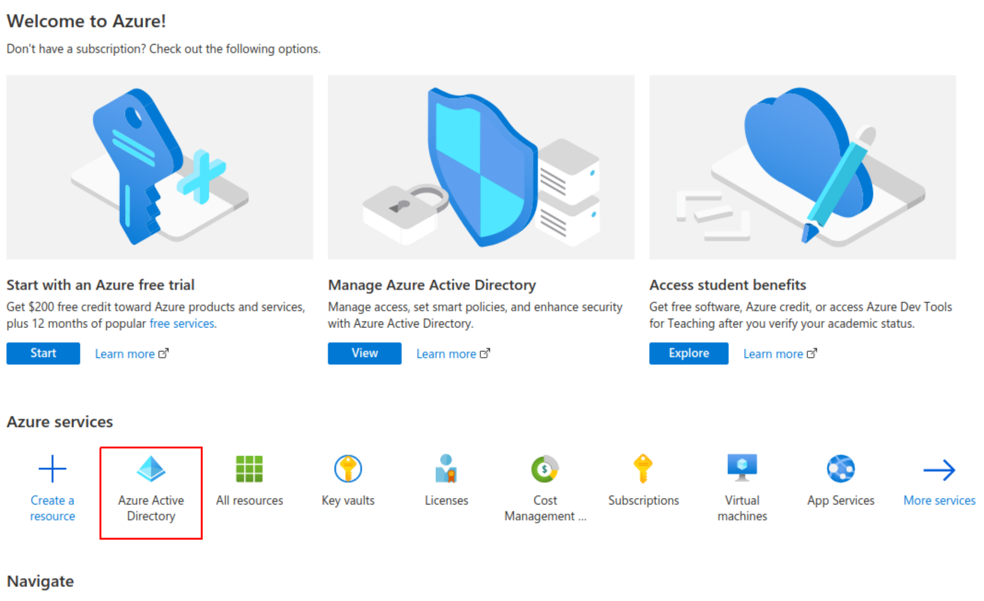
---

The sections below will show how to use both methods. For any of them, the first step is to find your Tenant ID:

How to find the Tenant ID

In order to find the Tenant ID you only need to login to the Azure portal (portal.azure.com) and take a look at the 'Overview' page of the service Azure Active Directory. Just click in the service as the below image shows:

Once there, you will find the Tenant ID in the box highlighted in the below image:


Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

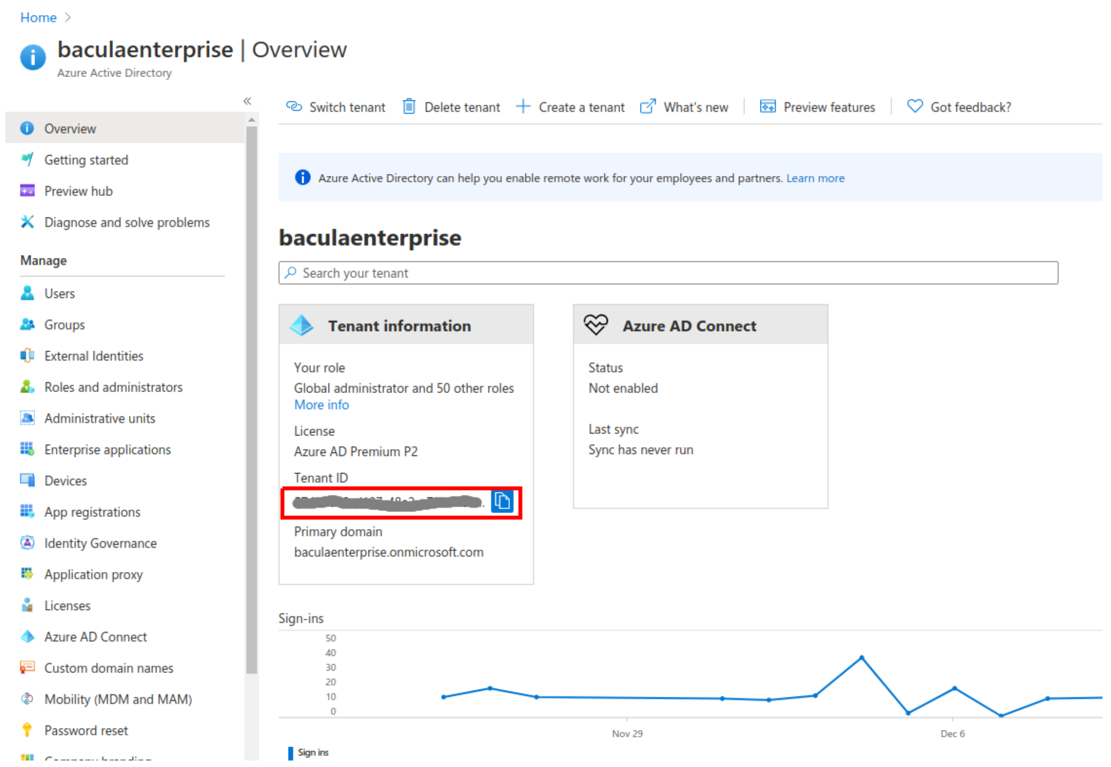
Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

Manage Azure Active Directory

Manage access, set smart policies, and enhance security with Azure Active Directory.

Access student benefits

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.


Home >

baculaenterprise | Overview

Azure Active Directory

Azure Active Directory can help you enable remote work for your employees and partners. Learn more

baculaenterprise

Tenant information

Your role: Global administrator and 50 other roles (More info)

License: Azure AD Premium P2

Tenant ID: [redacted]

Primary domain: baculaenterprise.onmicrosoft.com

Azure AD Connect

Status: Not enabled

Last sync: Sync has never run

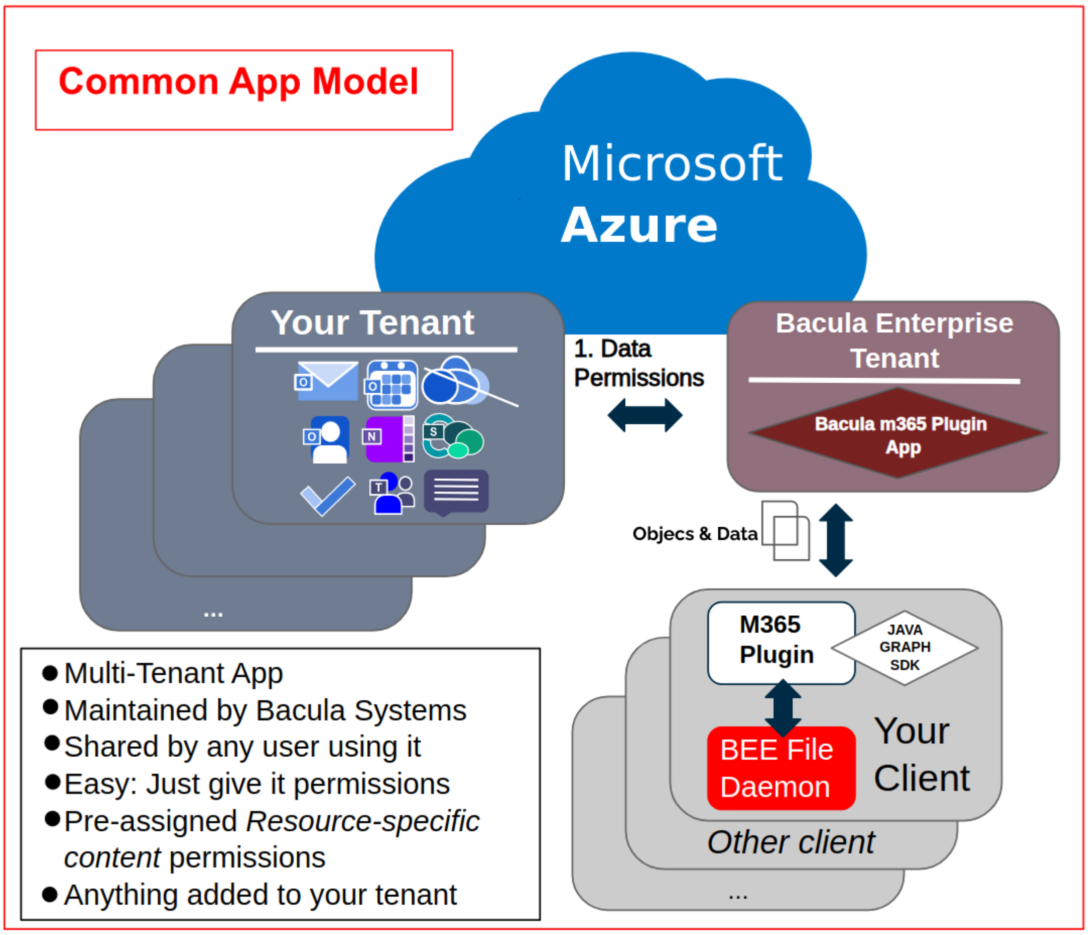
Authorization Method A: Common app model with 'bacula-m365-plugin' (DEPRECATED)

Bacula Systems has a registered application in Azure AD named **bacula-m365-plugin**. This section will show how to authorize it to perform backup and restore operations over your target tenant. All required steps to complete this authorization process are presented below.

Please, **use this connection method for testing purposes as it is deprecated**. For production systems, please go ahead with **Authorization method B**: *Authorization Method B: Standalone app model with 'bacula-m365-registrator' (RECOMMENDED)*.

Most of the procedures described in this section must be done by a **tenant administrator**. A tenant administrator is a user who has been assigned the Azure AD role **Global administrator**.

Below we show a schema of how this authorization method works:



A-1. Authorize bacula-m365-plugin

A **tenant administrator** must run the following query in a web browser, replacing **{tenantId}** with the value obtained in the first step:

Listing 163: **Authorization URL**

```
https://login.microsoftonline.com/{tenantId}/adminconsent?client_id=14a0b71a-
↪d9ca-496c-b4c0-76a3cbb5dc33&state=12345&redirect_uri=https://www.
↪baculasystems.com/m365-plugin-auth/common
```

You can get the exact same URL from the plugin command line itself, once you have installed it in a client named {your_client_name} using the following special Query command.

Listing 164: **Authorization URL**

```
*.query plugin="m365:" client={your_client_name} parameter=register:{tenantId}
```

Here is an execution example where the instructions are displayed and you need to open the provided URI:

Listing 165: **Query command for tenant URL**

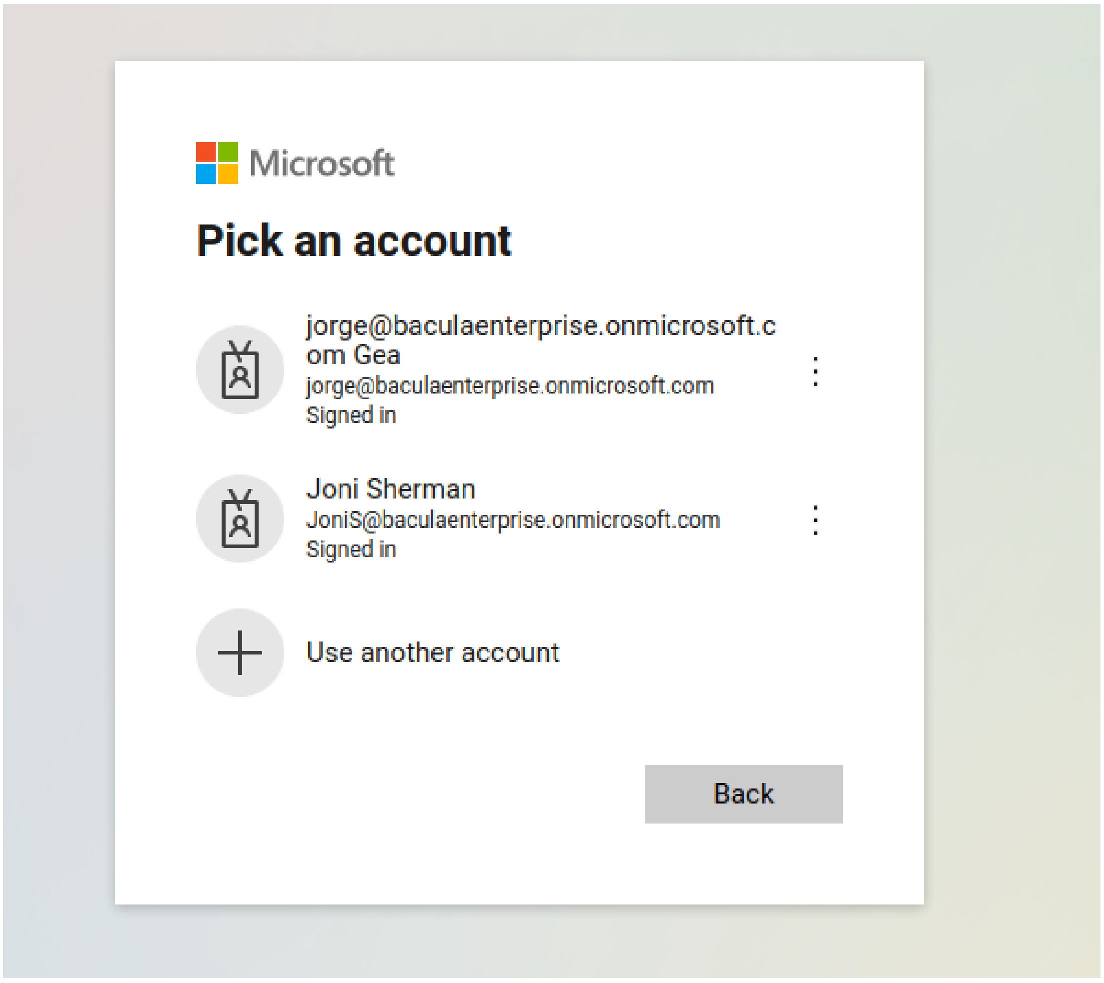
```
*.query plugin="m365:" client=127.0.0.1-fd parameter=register:57uia43-d107-
↪17a2-a2g2-aa53c10tdahc
console=---- M365 PLUGIN REGISTER COMMAND ----
console=------------------------------------
console=------------------------------------
info=Open the following URI in a browser with your tenant admin credentials
console=------------------------------------
console=------------------------------------
uri=https://login.microsoftonline.com/57uia43-d107-17a2-a2g2-aa53c10tdahc/
↪adminconsent?client_id=14a0b71a-d9ca-496c-b4c0-76a3cbb5dc33&state=12345&
↪redirect_uri=https://www.baculasystems.com/m365-plugin-auth/common
console=------------------------------------
console=------------------------------------
info=Once you have accepted the provided permissions to the bacula-m365-
↪plugin app..
info=You can get your ObjectId using the command below
console=------------------------------------
console=------------------------------------
command=.query plugin="m365: tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc"␣
↪client={your_client_name} parameter=objectid
console=------------------------------------
```

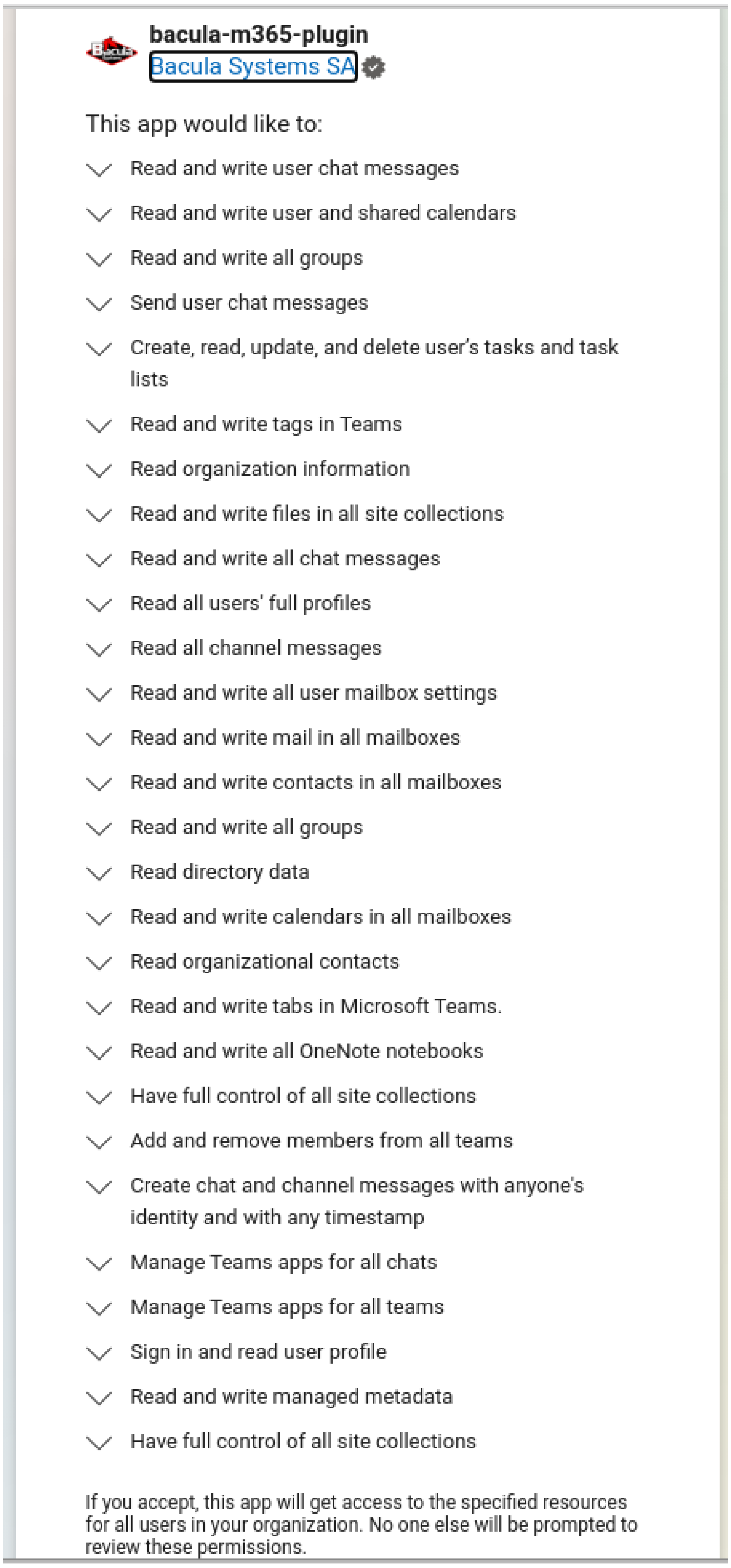
When opening the URI, Microsoft 365 will ask for credentials. You need to authenticate as a **tenant admin**:

. . . and once they are correctly provided, the app will ask for all the required permissions to backup and restore all of the supported elements:

The application needs **all of the listed permissions to be able to work**. If any of them is missing, backup or restore operationms will fail.

The image and list shown here are illustrative and some additional permissions may be needed as the plugin evolves. However, we provide also here a text list of current permissions needed:


Microsoft
Pick an account
jorge@baculaenterprise.onmicrosoft.com Gea
jorge@baculaenterprise.onmicrosoft.com
Signed in
Joni Sherman
JoniS@baculaenterprise.onmicrosoft.com
Signed in
Use another account
Back



bacula-m365-plugin

Bacula Systems SA

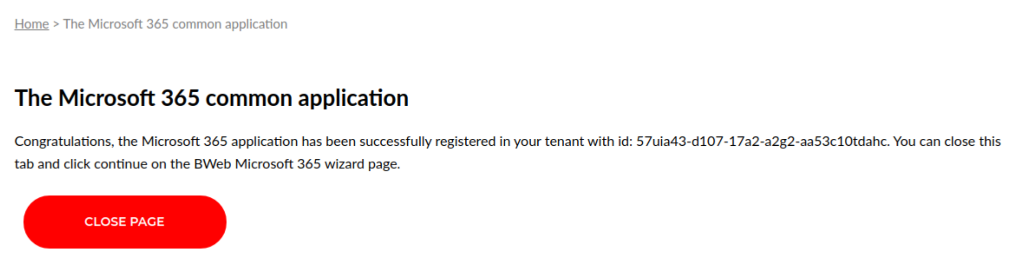
This app would like to:

- Read and write user chat messages
- Read and write user and shared calendars
- Read and write all groups
- Send user chat messages
- Create, read, update, and delete user's tasks and task lists
- Read and write tags in Teams
- Read organization information
- Read and write files in all site collections
- Read and write all chat messages
- Read all users' full profiles
- Read all channel messages
- Read and write all user mailbox settings
- Read and write mail in all mailboxes
- Read and write contacts in all mailboxes
- Read and write all groups
- Read directory data
- Read and write calendars in all mailboxes
- Read organizational contacts
- Read and write tabs in Microsoft Teams.
- Read and write all OneNote notebooks
- Have full control of all site collections
- Add and remove members from all teams
- Create chat and channel messages with anyone's identity and with any timestamp
- Manage Teams apps for all chats
- Manage Teams apps for all teams
- Sign in and read user profile
- Read and write managed metadata
- Have full control of all site collections

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

- Graph
  - Read and write user chat messages → Backup/Restore of Team Channels and Chats
  - Read and write user and shared calendars → Backup/Restore of group Calendars
  - Read and write all groups → Creation of Teams
  - Send user chat messages → Restore of Chats
  - Create, read, update and delete user's tasks and task lists → Backup/Restore of Tasks
  - Read and write tags in Teams → For future support of Team tags
  - Read organization information → Find tenant name
  - Read and write files in all sites collections → Backup/Restore OneDrive and Sharepoint
  - Read and write all chat messages → Backup/Restore of Team Channels
  - Read all users full profiles → Find users to Backup/Restore
  - Read all channel messages → Backup of Team Channels
  - Read and write all user mailbox settings → Outlook categories and more future mailbox information
  - Read and write contacts in all mailboxes → Backup/Restore contacts
  - Read directory data → Security checks of objectid, list groups, etc
  - Read and write calendars in all mailboxes → Backup/Restore User Calendars
  - Read organizational contacts → Backup organizational contacts
  - Read and write tabs in Microsoft Teams → Backup of Microsoft Teams tabs
  - Read and write all OneNote notebooks → Backup/Restore OneNote service*
  - Have full control of all site collections → Backup/Restore Sharepoint
  - Add and remove members from all Teams → Backup/Restore of Microsoft Teams members
  - Create chat and channel messages with anyone's identity and with any timestamp
  - Manage Teams apps for all chats → Backup of Microsoft Teams apps
  - Manage Teams apps for all teams → Backup of Chats apps
  - Read and write the names, descriptions, and settings of all channels of all Teams → Backup of Microsoft Teams Channels
  - Add and remove members from all channels → Backup of Microsoft Teams Channels
  - Read and write managed metadata → Security checks of objectid, list groups, etc
  - Sign in and read user profile → Ability to connect to the tennant
  - etc.
- Sharepoint Online
  - Full management of all site collections → Backup/Restore Sharepoint (PnP)
  - Full management of Term Store → Backup/Restore Sharepoint (PnP)

Once it is confirmed, the browser will use the 'redirect_uri', which is a page on baculasystems.com that will confirm the result of the registration process.



Home > The Microsoft 365 common application

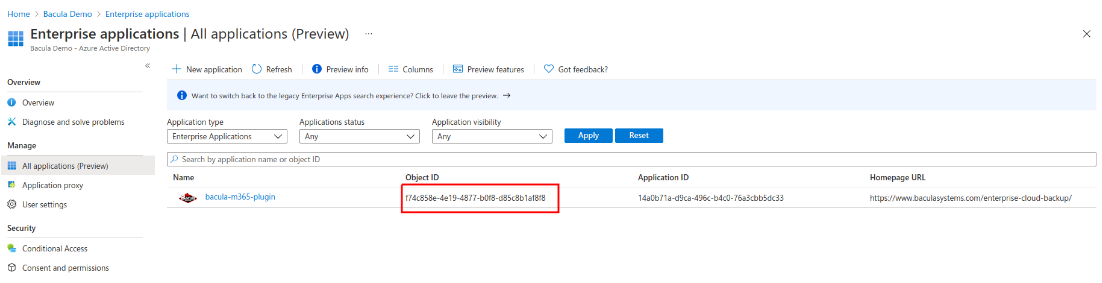
The Microsoft 365 common application

Congratulations, the Microsoft 365 application has been successfully registered in your tenant with id: 57uia43-d107-17a2-a2g2-aa53c10tdahc. You can close this tab and click continue on the BWeb Microsoft 365 wizard page.

CLOSE PAGE

The generated URI contains the parameter **admin_consent=True** if the action was successful, and you will see a confirmation message in that case. Otherwise, the operation may have not been successful for some reason and you will see an error message.

Once that action is done, the tenant where our app now has permissions will show the plugin in the Enterprise Apps section:



Home > Bacula Demo > Enterprise applications

Enterprise applications | All applications (Preview)

Bacula Demo - Azure Active Directory

Overview: Overview; Diagnose and solve problems
Manage: All applications (Preview); Application proxy; User settings
Security: Conditional Access; Consent and permissions

New application | Refresh | Preview info | Columns | Preview features | Got feedback?

Want to switch back to the legacy Enterprise Apps search experience? Click to leave the preview. →

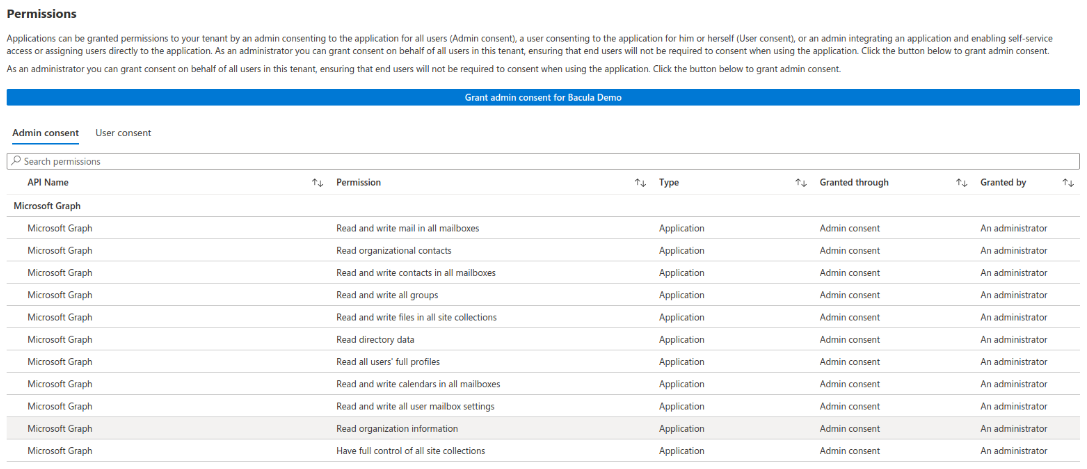
Application type: Enterprise Applications
Applications status: Any
Application visibility: Any

Apply | Reset

Search by application name or object ID

Name	Object ID	Application ID	Homepage URL
bacula-m365-plugin	f74c858e-4e19-4877-b0f8-d85c8b1af8f8	14a0b71a-d9ca-496c-b4c0-76a3cbb5dc33	https://www.baculasystems.com/enterprise-cloud-backup/

Clicking on the app, the tenant admin can always see the permissions assigned:



Permissions

Applications can be granted permissions to your tenant by an admin consenting to the application for all users (Admin consent), a user consenting to the application for him or herself (User consent), or an admin integrating an application and enabling self-service access or assigning users directly to the application. As an administrator you can grant consent on behalf of all users in this tenant, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

As an administrator you can grant consent on behalf of all users in this tenant, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

Grant admin consent for Bacula Demo

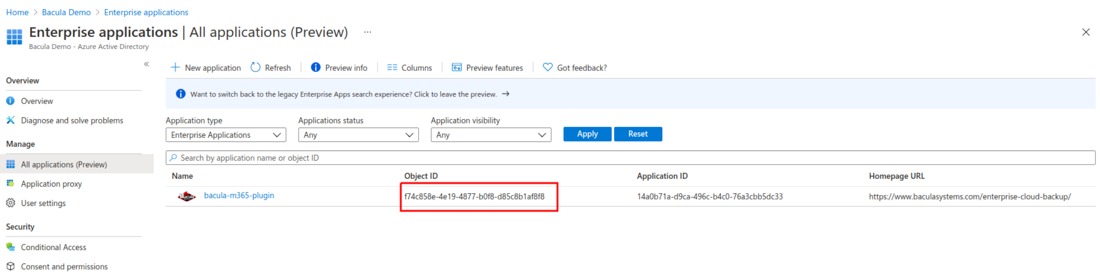
Admin consent | User consent

Search permissions

API Name	Permission	Type	Granted through	Granted by
Microsoft Graph				
Microsoft Graph	Read and write mail in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read organizational contacts	Application	Admin consent	An administrator
Microsoft Graph	Read and write contacts in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read and write all groups	Application	Admin consent	An administrator
Microsoft Graph	Read and write files in all site collections	Application	Admin consent	An administrator
Microsoft Graph	Read directory data	Application	Admin consent	An administrator
Microsoft Graph	Read all users' full profiles	Application	Admin consent	An administrator
Microsoft Graph	Read and write calendars in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read and write all user mailbox settings	Application	Admin consent	An administrator
Microsoft Graph	Read organization information	Application	Admin consent	An administrator
Microsoft Graph	Have full control of all site collections	Application	Admin consent	An administrator

A-2. Grab Object ID

The plugin needs a final parameter that is unique to each tenant and the plugin app. This is **ObjectID**, and may be obtained from the Overview app page, once step 2 has been completed:



Home > Bacula Demo > Enterprise applications

Enterprise applications | All applications (Preview)

Bacula Demo - Azure Active Directory

Want to switch back to the legacy Enterprise Apps search experience? Click to leave the preview. →

Application type: Enterprise Applications
Applications status: Any
Application visibility: Any

Name	Object ID	Application ID	Homepage URL
bacula-m365-plugin	f74c858e-4e19-4877-b0f8-d85c8b1af8f8	14a0b71a-d9ca-496c-b4c0-76a3cbb5dc33	https://www.baculasystems.com/enterprise-cloud-backup/

The plugin can also obtain it from the command line using another special Query command. You can see that this exact command is also suggested in the command that shows the register URL:

Listing 166: **Authorization URL**

```
*.query plugin="m365: tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc" client=
↪{your_client_name} parameter=objectid
```

Here is an execution example:

Listing 167: **Authorization URL**

```
*.query plugin="m365: tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc" client=127.
↪0.0.1-fd parameter=objectid
console=---- M365 PLUGIN OBJECTID COMMAND ----
console=-------------------------------------
objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5
console=-------------------------------------
```

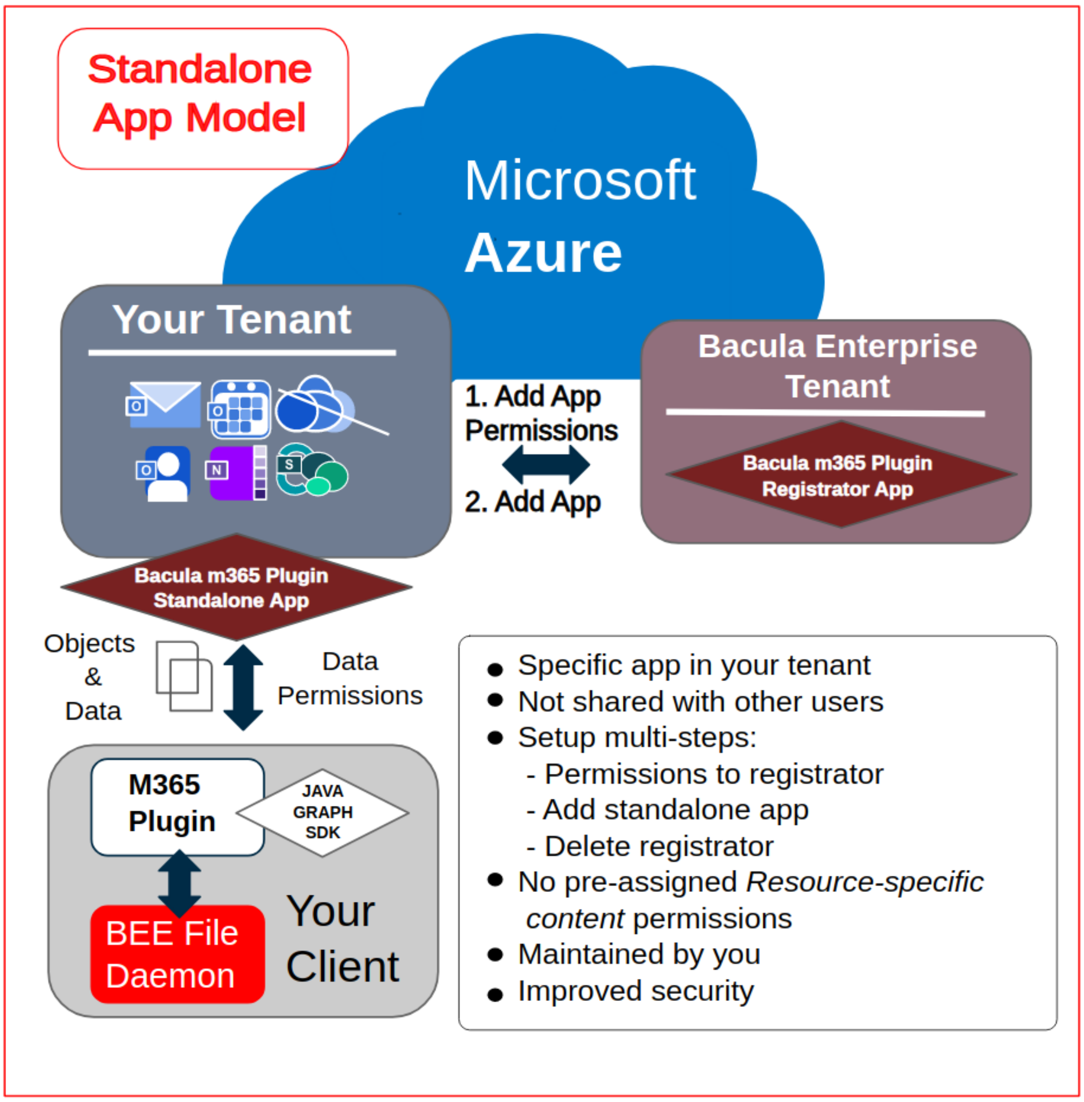
Authorization Method B: Standalone app model with 'bacula-m365-registrator' (RECOMMENDED)

Bacula Systems has a registered application in Azure AD named **bacula-m365-registrator**. This section will show how to authorize it to perform application management operations over your target tenant.

Once you authorize it, you will be able to add a new **bacula-m365-plugin-standalone** specifically for your tenant that you can directly control and that will be used only in your environment.

This is the **recommended authorization method** for any production environment.

Below we show a schema of how this authorization method works:


Standalone App Model
Microsoft Azure
Your Tenant
Bacula Enterprise Tenant
1. Add App Permissions
2. Add App
Bacula m365 Plugin Registrator App
Bacula m365 Plugin Standalone App
Objects & Data
Data Permissions
M365 Plugin
JAVA GRAPH SDK
BEE File Daemon
Your Client
Specific app in your tenant
Not shared with other users
Setup multi-steps:
- Permissions to registrator
- Add standalone app
- Delete registrator
No pre-assigned Resource-specific content permissions
Maintained by you
Improved security

B-1. Authorize bacula-m365-registrator

A **tenant admin** must run the following query in a web browser, replacing **{tenantId}** with the value obtained in the first step:

Listing 168: **Authorization URL**

```
https://login.microsoftonline.com/{your_tenant_id}/adminconsent?client_
↪id=8ed4fad3-3f63-44fb-b7e0-e324895d7df9&state=12345&redirect_uri=https://
↪www.baculasystems.com/m365-plugin-auth/standalone
```

You can get the exact same URL from the plugin command line itself, once you have installed it in a client named {your_client_name} using the following special Query command.

Listing 169: **Authorization URL**

```
*.query plugin="m365:" client={your_client_name} parameter=register-
↪registrator:{tenantId}
```

Here is an execution example where the instructions are displayed and you need to open the provided URI:

Listing 170: **Query command for tenant URL**

```
*.query client=127.0.0.1-fd plugin="m365:" parameter=register-
↪registrator:57uia43-d107-17a2-a2g2-aa53c10tdahc
console=---- M365 PLUGIN REGISTER COMMAND ----
console=------------------------------------
console=------------------------------------
info=Open the following URI in a browser with your tenant admin credentials
console=------------------------------------
console=------------------------------------
uri=https://login.microsoftonline.com/57uia43-d107-17a2-a2g2-aa53c10tdahc/
↪adminconsent?client_id=8ed4fad3-3f63-44fb-b7e0-e324895d7df9&state=12345&
↪redirect_uri=https://www.baculasystems.com/m365-plugin-auth/standalone
console=------------------------------------
```

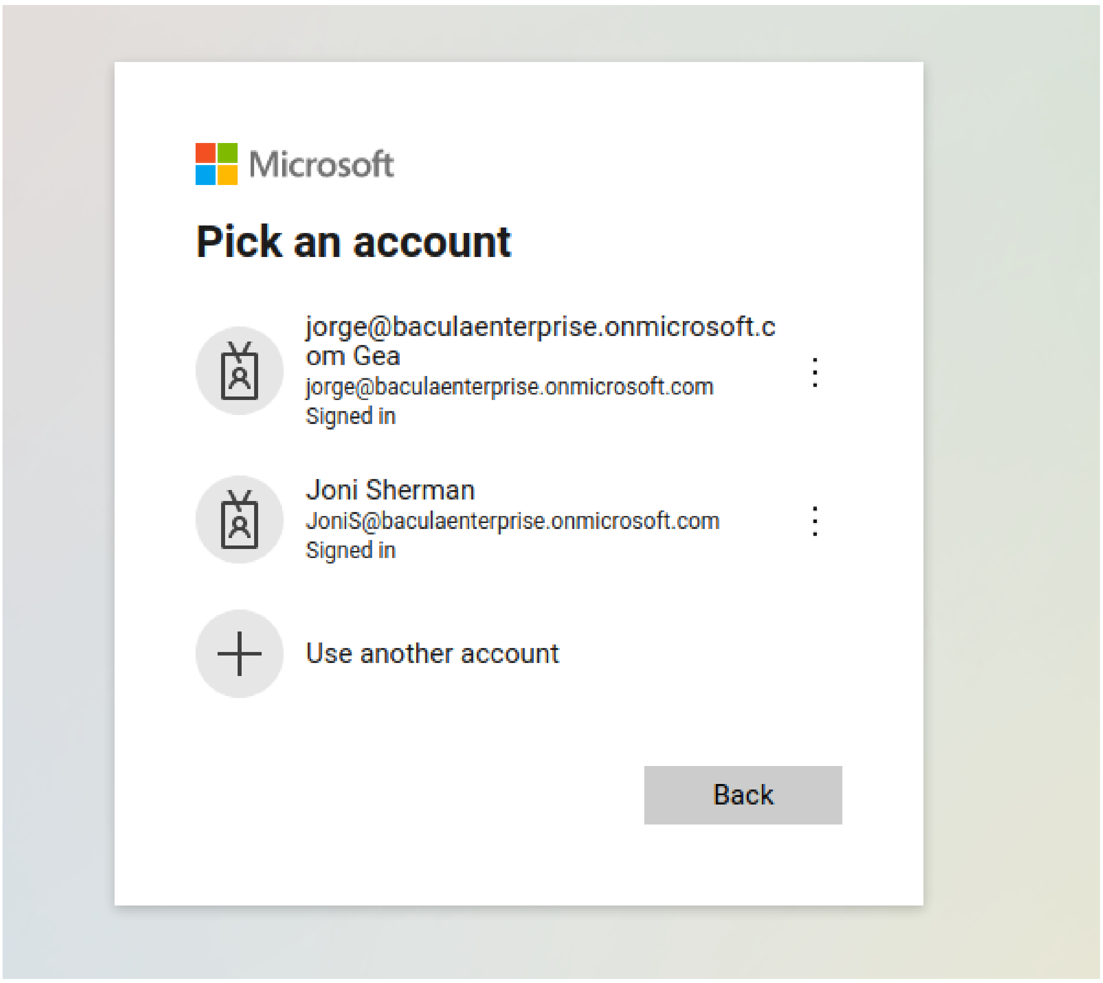
When opening the URI, Microsoft 365 will ask for credentials. You need to authenticate as a **tenant admin**:

…and once they are correctly provided, the app will ask for all the required permissions to backup and restore all of the supported elements:

The application needs **all of the listed permissions to be able to work**. If any of them is missing, the final command to add the application will fail.

The image shown here is illustrative and some additional permission may be needed as the plugin evolves. To facilitate that information, we are listing the permissions in text alongside the reason for their need.

- Graph - Sign in and read user profile: Connect to the tenant - Read and write all groups: Necessary to manage some permission grants (delegated form) - Read directory data: Find your tenant information, check service principals (delegated form) - Manage all delegated permission grants: Generate delegated permissions grants for the end app - Read and write all applications: Create/Update the new application (client credentials) - Read organization information: Find your tenant information and show user-friendly progress - Read and write all applications: Create/Update the new application (delegated form) - Read and write all groups: Necessary to manage some permis-



Microsoft

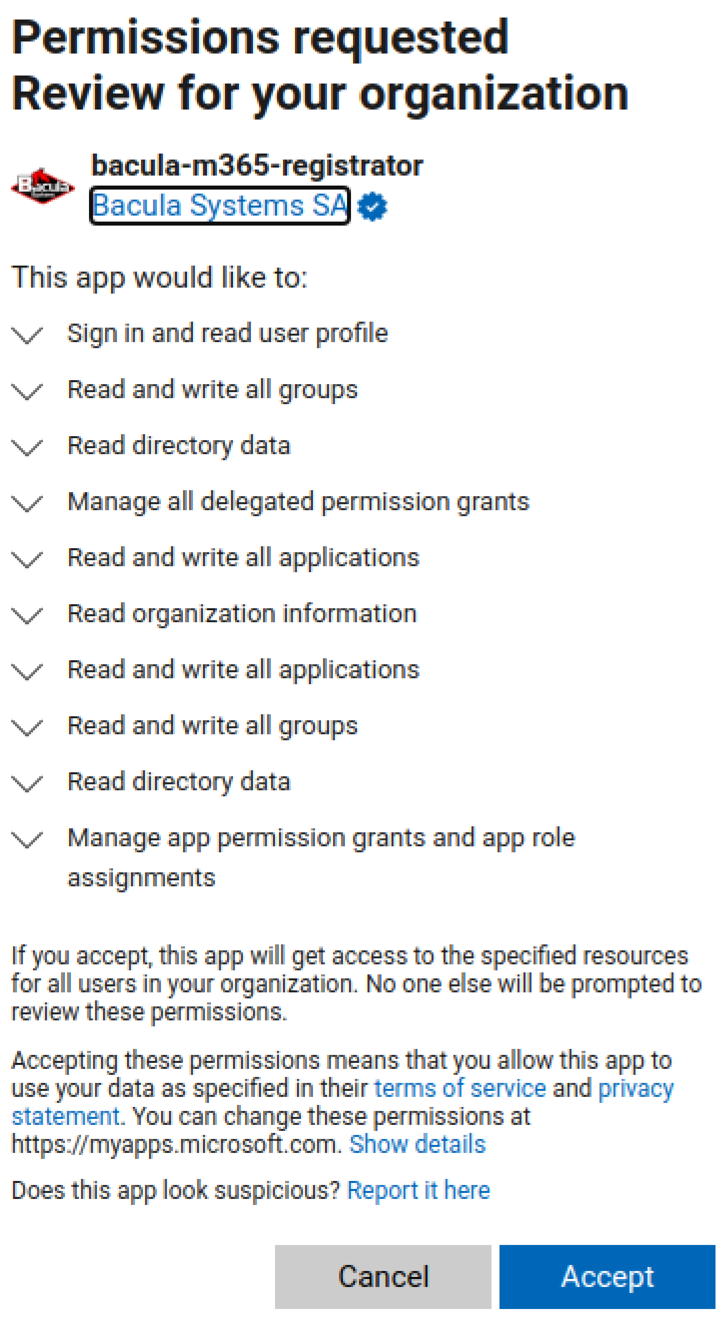
Pick an account

jorge@baculaenterprise.onmicrosoft.com Gea
jorge@baculaenterprise.onmicrosoft.com
Signed in

Joni Sherman
JoniS@baculaenterprise.onmicrosoft.com
Signed in

Use another account

Back


Permissions requested
Review for your organization

bacula-m365-registrator
Bacula Systems SA

This app would like to:

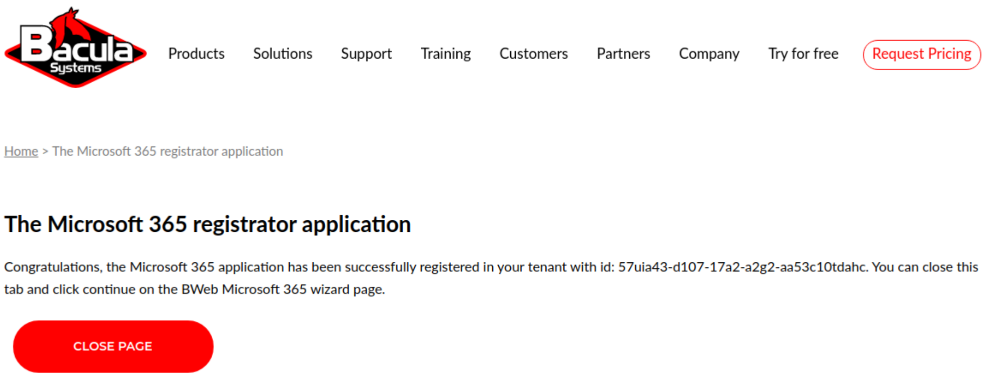
- Sign in and read user profile
- Read and write all groups
- Read directory data
- Manage all delegated permission grants
- Read and write all applications
- Read organization information
- Read and write all applications
- Read and write all groups
- Read directory data
- Manage app permission grants and app role assignments

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here sion grants (client credentials) - Read directory data: Find your tenant information, check service principals (client credentials) - Manage app permission grants and app role assignments: Add necessary plugin permissions

Once it is confirmed, the browser will use the 'redirect_uri', which is a page on baculasystems.com that will confirm the result of the registration process.



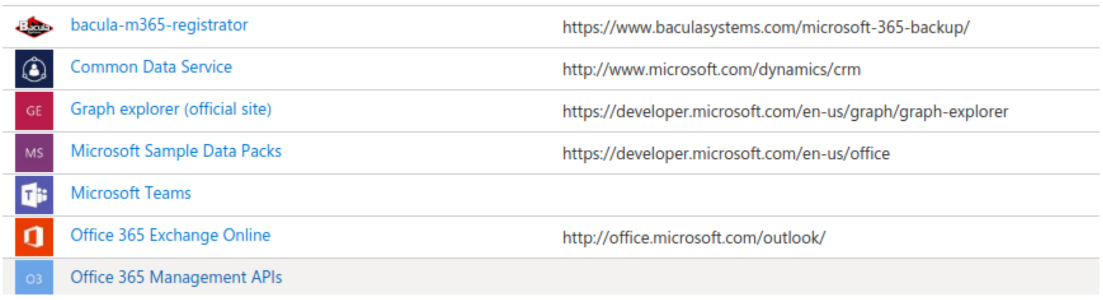
Home > The Microsoft 365 registrator application

The Microsoft 365 registrator application

Congratulations, the Microsoft 365 application has been successfully registered in your tenant with id: 57uia43-d107-17a2-a2g2-aa53c10tdahc. You can close this tab and click continue on the BWeb Microsoft 365 wizard page.

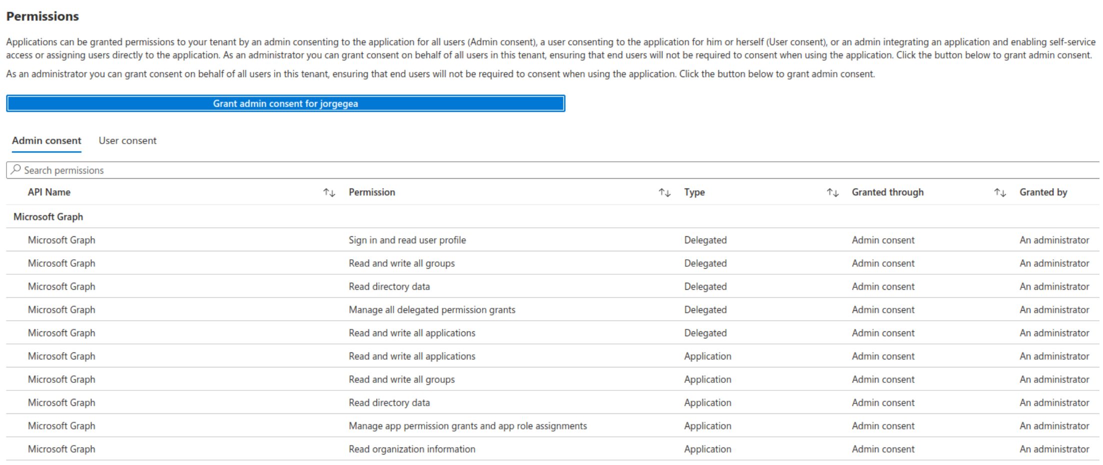
The generated URI contains the parameter **admin_consent=True** if the action was successful, so you will see a confirmation message in that case. Otherwise, the operation may have not been successful for some reason and you will see an error message.

Once that action is done, the tenant where our app now has permissions will show the plugin in the Enterprise Apps section:



	bacula-m365-registrator	https://www.baculasystems.com/microsoft-365-backup/
	Common Data Service	http://www.microsoft.com/dynamics/crm
GE	Graph explorer (official site)	https://developer.microsoft.com/en-us/graph/graph-explorer
MS	Microsoft Sample Data Packs	https://developer.microsoft.com/en-us/office
	Microsoft Teams	
	Office 365 Exchange Online	http://office.microsoft.com/outlook/
O3	Office 365 Management APIs	

Clicking on the app, the tenant admin can always see the permissions assigned:



Permissions

Applications can be granted permissions to your tenant by an admin consenting to the application for all users (Admin consent), a user consenting to the application for him or herself (User consent), or an admin integrating an application and enabling self-service access or assigning users directly to the application. As an administrator you can grant consent on behalf of all users in this tenant, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

As an administrator you can grant consent on behalf of all users in this tenant, ensuring that end users will not be required to consent when using the application. Click the button below to grant admin consent.

API Name ↑↓	Permission ↑↓	Type ↑↓	Granted through ↑↓	Granted by
Microsoft Graph				
Microsoft Graph	Sign in and read user profile	Delegated	Admin consent	An administrator
Microsoft Graph	Read and write all groups	Delegated	Admin consent	An administrator
Microsoft Graph	Read directory data	Delegated	Admin consent	An administrator
Microsoft Graph	Manage all delegated permission grants	Delegated	Admin consent	An administrator
Microsoft Graph	Read and write all applications	Delegated	Admin consent	An administrator
Microsoft Graph	Read and write all applications	Application	Admin consent	An administrator
Microsoft Graph	Read and write all groups	Application	Admin consent	An administrator
Microsoft Graph	Read directory data	Application	Admin consent	An administrator
Microsoft Graph	Manage app permission grants and app role assignments	Application	Admin consent	An administrator
Microsoft Graph	Read organization information	Application	Admin consent	An administrator

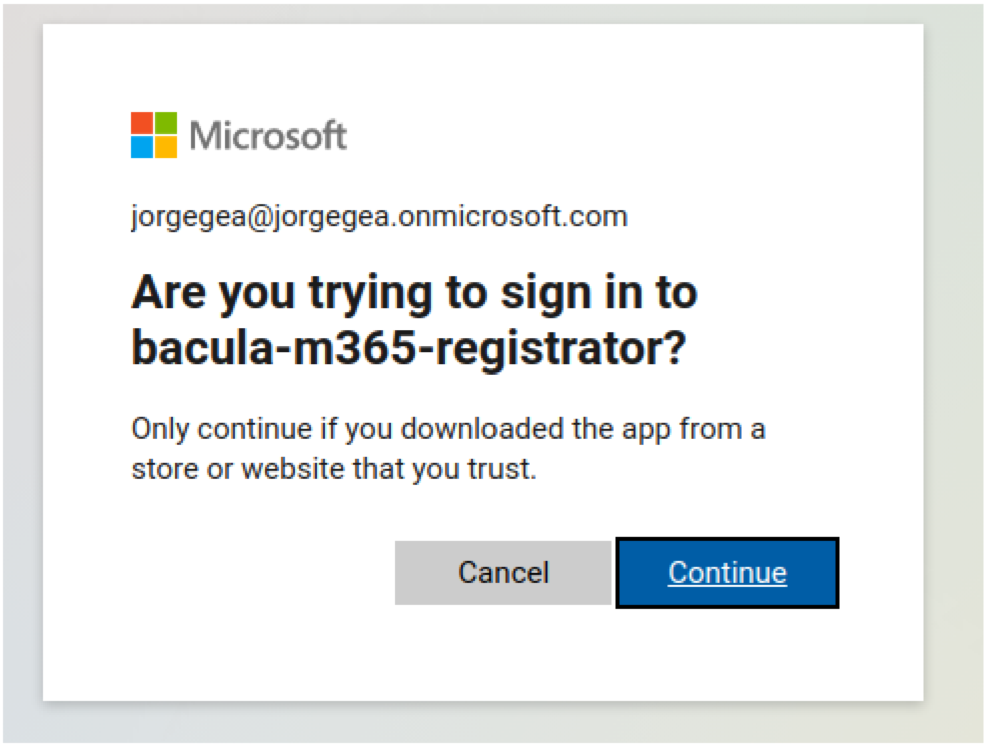
B-2. Add bacula-m365-plugin-standalone to your tenant

Once the registrator app is successfully registered into the target tenant, it is possible to add the bacula-m365-plugin-standalone application as a 'registered app' using the following bconsole command:

Listing 171: **Add application command**

```
*.query plugin="m365: tenant={tenantId}" client={your_client_name}␣
↪parameter=add-app
```

This command needs to run operations with 'client credential' permissions, but also with 'delegated permissions' where an admin user must be implied. Therefore, at some point during its execution it will ask for the user interaction in order to complete the authentication process using an special URL and a given code. Once introduced, we will be asked about confirming the sign-in of bacula-m365-registrator:



Microsoft

jorgegea@jorgegea.onmicrosoft.com

Are you trying to sign in to bacula-m365-registrator?

Only continue if you downloaded the app from a store or website that you trust.

The process must be performed by a tenant administrator. On the other hand, Microsoft Azure service needs some time in order to present newly created objects. Therefore, the process will be paused a couple of times for about one minute.

You need to wait **2-3 minutes** (minimum!) after the first registration command has been completed. If it is executed too soon it could fail, as the registrator may not be completely available yet. If this happens, please, just run the command again.

Please find an example for this procedure below:

Listing 172: **Add application example**

```
*.query client=127.0.0.1-fd plugin="m365: tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc" parameter=add-app
console=---- M365 ADD APP ----
```

(continues on next page)

(continued from previous page)

```
info=Creating destination config file in: /opt/bacula/working/m365/
↪baculaenterprise/bacula_m365_config.conf
info=Creating application: bacula-m365-plugin-standalone in tenant:␣
↪baculaenterprise ...
info=Waiting a bit so MS Azure cache has enough time to show our new␣
↪Application...
info=Adding certificate key credential ...
info=Adding password credential ...
info=Application successfully created
info=We don't have a valid token. Please:
info=To sign in: use a web browser to open the page https://microsoft.com/
↪devicelogin and enter the code CG2QYSK8Y to authenticate. You need to do it␣
↪with an administrator user
info=Adding owner: johndoe@johndoe.onmicrosoft.com to the application ...
info=Owner successfully added
info=Creating service principal for application ...
info=Service principal successfully created
info=Waiting a bit so MS Azure cache has enough time to show our new␣
↪ServicePrincipal...
info=Granting admin consent for permissions ...
info=Admin consent granted
info=Storing access configuration to the new app in file: /tmp/regress/
↪working/m365/johndoe/bacula_m365_config.conf
info=Access configuration successfully stored
console=---- ADD APP COMPLETED SUCCESSFULLY ----
console=--------------------------------------
info=Use the generated configuration file in your m365 filesets. Just add the␣
↪property config_file=/your/path/bacula_m365_config.conf to the line Plugin=
↪"m365: ..."
info=Sample: Plugin = "m365: config_file=/opt/bacula/etc/m365/mytenant/bacula_
↪m365_config.conf service=drive"
console=--------------------------------------
console=--------------------------------------
```

In case the add-app command fails for any reason, please, wait a bit and just run it again.

B-3. Utilize your generated config file

The result of the add-app command is the generation of a configuration file that is intended to be referenced in any future fileset, as the output of the command is indicating.

We recomend to copy that file to a convenience common location (for instance /opt/bacula/etc/filesets) and set the same diretive in any M365 fileset:

Listing 173: **Fileset example**

```
Fileset {
     Name = FS_M365
     Include {
             Options {
                     signature = MD5
```

(continues on next page)

(continued from previous page)

```
                    compression = LZO
                }
                Plugin = "m365: config_file=/opt/bacula/etc/m365/mytenant/bacula_
↪m365_config.conf service=drive"
        }
}
```

The contents of the configuration file are described below:

Listing 174: **Autogenerated config file contents**

```
#Thu Jul 01 12:30:34 CEST 2021                                                  ␣
↪                      -> Generation Date
secret=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx                                          ␣
↪                      -> Password secret auto-generated
appid=1587ba33-8eer-4512-bgb1-6d65fd393f51                                       ␣
↪                      -> Azure AD AppID of your bacula-m365-plugin-standalone
objectid=8jki8705-010p-4ee6-b2aa-c3klmn1b5b9                                     ␣
↪                      -> Service principal ID of your Application
tenant=e4wsdf55-de8a-4a83-945a-bfaabb6fa166                                      ␣
↪                      -> Your tenant id
sharepoint_certificate_path=/opt/bacula/etc/m365/example/bacula-m365-plugin-
↪standalone-cert.pfx    -> Certificate generated to use PnP in Sharepoint␣
↪module
sharepoint_certificate_password=fAG@205PoOSSU<j2u65zqEKIN<Dh~ya3AWKqK]vv         ␣
↪                      -> Sharepoint certificate password
```

Please, note that once this application is generated in your tenant, you have full control of it. As a result, you can manually generate a new secret and set the value yourself at your convenience whenever this is needed, but we recommend to use the add-app command to do this, as we explain in the following section.

It is recommended to use the add-app commands a best practice, allowing you to specify the path of the configuration file (config_file=path) within your fileset configurations. By following this approach, you will not have to alter your backup configuration each time you perform an update.

B-4. Optional: Renew your secrets

The add-app command discussed in section B-3 is prepared to automatically regenerate a new password secret if it detects that the current ones are expired or close to expire (less than 3 months).

Bacula automatically generated keys feature an extended timestamp, however, they can be regenerated at any time or on a regular basis by simply executing the add-app command again, incorporating the q_addapp_renew=true parameter within the plugin string, as demonstrated below:

Listing 175: **Regenerate secret and certificate**

```
*.query client=example-client-fd plugin="m365: tenant=********_****_****_****_
↪************ q_addapp_renew=true" parameter=add-app
console=---- M365 ADD APP ----
info=File /opt/bacula/etc/m365/example/bacula_m365_config.conf already exists.
↪ Contents will be just updated
```

(continues on next page)

(continued from previous page)

```
info=Application named: bacula-m365-plugin-standalone already exists in␣
↪tenant: example ...
info=Updating its permissions ...
info=Permissions updated successfully
info=Generating key credential ...
info=Key credential generated successfully
info=Generating new password credential ...
info=New password credential generated successfully
info=Service principal already exists
info=Granting admin consent for permissions ...
info=Admin consent granted
info=We don't have any valid token. It's needed to sign in
info=To sign in; use a web browser to open the page https://microsoft.com/
↪devicelogin and enter the code GJL5LGXC8 to authenticate. Please do it with␣
↪a tenant administrator user
info=Storing access configuration to the new app in file: /opt/bacula/etc/
↪m365/example/bacula_m365_config.conf
info=Access configuration successfully stored
console=---- ADD APP COMPLETED SUCCESSFULLY ----
console=----
info=Use the generated configuration file in your m365 filesets. Just add the␣
↪property config_file=/your/path/bacula_m365_config.conf to the line Plugin=
↪"m365: ..."
info=Sample: Plugin = "m365: config_file=/opt/bacula/etc/m365/example/bacula_
↪m365_config.conf service=drive"
console=----
console=----
```

Note: q_addapp_renew is available since Bacula Enterprise 18.0.8.

Please, note that this command will also regenerate the certificate and associated password that Sharepoint module needs to function. In general, it is recommended to simply use this add-app command (or its equivalent through BWeb) to manage the authentication and access M365 and so, using the config_file that contains the details. However, if you are using different variables manually, note that you will need to update them once you run any new add-app action with the renewal behavior.

Any add-app execution requires the proper configuration of the bacula-m365-registrator app. If you have uninstalled it following the initial add-app procedure as outlined in section B-5 during the plugin installation, you must repeat the B-1 step.

It is important to recognize that this process (excluding the registration part) could be automated through the use of a suitably configured Admin Job.

B-5. Optional: Remove bacula-m365-registrator app

The registrator application has elevated permissions, as without them it would not be possible to manage applications and assoicated permissions. It is handy to have it if you plan to run automatically the add-app command at some point, as suggested in B-4.

However, if you want to remove it from your tenant after completing the add-app process you could do it manually or you could call the following bconsole command:

Listing 176: **Delete registrator**

```
.query client={your-client} plugin="m365: tenant={yourTenantId}"␣
↪parameter=delete-registrator
```

Please, be aware that once you do that, you won't be able to run the add-app command. If you need to run it again after deletion, you will need to call again the register-registrator command.

Check permissions

It is possible to check if the service principal of the target tenant (this is shown under 'enterprise applications' in Azure AD) has all the needed permissions to perform plugin operations.

You just need to run the following command:

Listing 177: **Check permissions**

```
.query client={your-client} plugin="m365: tenant={yourTenantId} objectid=
↪{yourobjectid}" parameter=permissions
```

The ouput will report if everything is ok around permissions, or if there is any missing one:

Listing 178: **Check permissions: ok output**

```
.query client={your-client} plugin="m365: tenant={yourTenantId} objectid=
↪{yourobjectid}" parameter=permissions
info=All permissions are correctly set
```

Listing 179: **Check permissions: some is missing**

```
.query client={your-client} plugin="m365: tenant={yourTenantId} objectid=
↪{yourobjectid}" parameter=permissions
error=Delegated permission Chat.ReadWrite was not found in your Azure service␣
↪principal 8cee7605-0d5e-42b6-b269-c39f3411b5b9
error=Please review your app permissions
```

In case you are using the standalone application, you will need to run the command with appid and secret too:

Listing 180: **Check permissions**

```
.query client={your-client} plugin="m365: tenant={yourTenantId} objectid=
↪{yourobjectid} appid={yourappid} secret={yoursecret}" parameter=permissions
```

Please, run this command at any time you have doubts about the correct app permissions.

Please note though, that the command is unable to check if you have activated or not ‘Protected APIs’ for Teams and Chats. It will check the presence of those permissions, but it is up to Microsoft to enable those particular ones and they do not show (at the time of writing) any flag indicating if they are enabled or not.

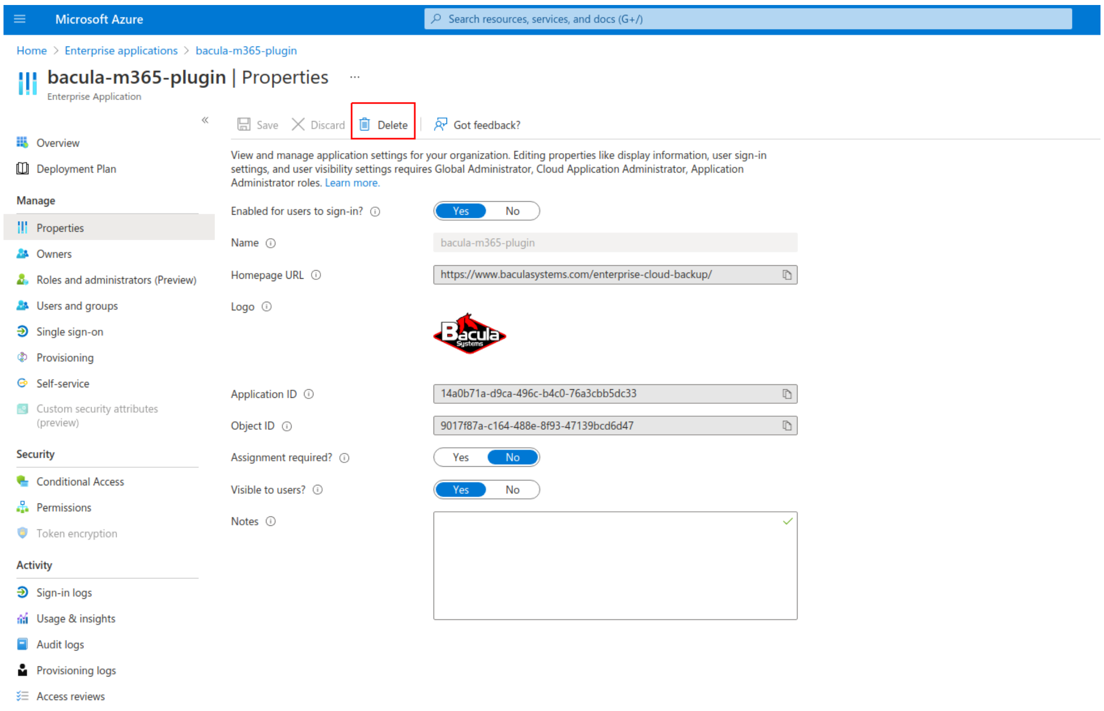
Migration from Authenticaton Method A to Authentication Method B

Perhaps you started to use this plugin using Method A and now you want to switch to Method B in order to enjoy its advantadges. This section will show the steps needed in order to complete such migration.

Migration - Step 1: Remove bacula-m365-plugin from your tenant

The first step is to remove the common app ‘bacula-m365-plugin’ from your tenant. In order to do it, you need to go to your Azure portal and open the ‘Azure Active Directory’ application as usual.

From there, please go to ‘Enterprise applications’ and locate bacula-m365-plugin app. You need to select the application and enter into ‘Properties’ section. From there, you need to click on the ‘Delete’ button:



View and manage application settings for your organization. Editing properties like display information, user sign-in settings, and user visibility settings requires Global Administrator, Cloud Application Administrator, Application Administrator roles. Learn more.

Enabled for users to sign-in? [✓ Yes] No

Name: bacula-m365-plugin

Homepage URL: https://www.baculasystems.com/enterprise-cloud-backup/

Logo

Application ID: 14a0b71a-d9ca-496c-b4c0-76a3cbb5dc33

Object ID: 9017f87a-c164-488e-8f93-47139bcd6d47

Assignment required? Yes [✓ No]

Visible to users? [✓ Yes] No

Notes

Migration - Step 2: Perform Authentication Method B

Once you have removed the common application, you simply need to follow all the steps explained in this documment for authentication method B. In case you are using BWeb and you have already registered you tenant, it is recommended to remove it first, and then add it also using method B. Otherwise, for manual configuration, simply follow the steps mentioned in this document: *Authorization Method B: Standalone app model with 'bacula-m365-registrator' (RECOMMENDED)*.

Migration - Step 3: Update your filesets

Once you have your new app in your tenant, you need to update all your filesets and: - Replace objectid with the new value associated to your app in your tenant - Add appid and secret associated to your app in your tenant

Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to call the M365 Plugin:

Listing 181: **Fileset M365**

```
Fileset {
   Name = FS_M365
   Include {
      Options {
        signature = MD5
        ...
      }
      Plugin = "m365: <m365-parameter-1>=<m365-value-1> <m365-parameter-2>=
↪<m365-value-2> ..."
   }
}
```

It is **strongly recommended** to use only one 'Plugin' line in every fileset. The plugin offers the needed flexibility to combine different modules backup inside the same plugin line. Different tenants, in case of existing, should be using different filesets and different jobs.

Below sub-sections list all the parameters you can use to control M365 Plugin behavior.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ','.

Note: This activity module is available since Bacula 18.0.1.

Common parameters

These parameters are common and applicable to all the modules of the M365 Plugin.

Op-tion	Re-quire	De-fault	Values	Exam-ple	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
con-fig_fi	No		Path pointing to a file containing any combination of plugin parameters	/opt/bacul	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset
log	No	/opt/bac debug.l	Existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/m36	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
de-bug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/m365/m365-debug.log* files containing debut information which is more complete with a greater debug number
path	No	/opt/bac	Existing path with enough permissions for File Daemon to create any internal plugin file	/mnt/my-vol/	Uses this path to store metadata and temporary files
ten-ant	**Yes**		Valid tenant id string	57uia43-d107-17a2-a2g2-aa53c10tc	The **tenant ID** where the plugin will connect to in order to run backups or restores. Please, check section 5.1 of this paper for more information
ob-jec-tid	**Yes**		String representing the objectid related to bacula-m365-plugin Azure app and the provided tenant id	56ddf1h9-eb5d-42nf-bac7-7b019fd2	The **object ID** of the plugin app of the plugin in Azure, once this is registered in the target tenant. Please, check section 5.1 of this paper for more information
ap-pid	No		String representing the appid associated to bacula-m365-plugin-standalone Azure app registered in the configured tenant id	89tt4hu7-r4h7-kied-56gu-0895kf94	A valid appid string associated to bacula-m365-plugin-standalone Azure app registered in the configured tenant id. Please, check section 5.1 of this paper for more information
se-cret	No		String representing the associated appid secret	Jn8.lU-B.3P5gIR	The secret associated to the m365 Azure app corresponding the configured appid string. Please, check section 5.1 of this paper for more information
to-ken_	No	to-ken_ca	File name located in a valid existing path (it's possible to put only the name of the file and 'working' dir will be used)	my_cache	The path that will be used to store the login cache for the device code flow authenticated users, wich is relative to the working tenant folder (working/tenant-name/token_cache.json)

Note: The path option is adjustable while M365 jobs are in progress. The running jobs will continue to use the current spool area, and new jobs will use the new spool area defined by the path option. If specifying more than one service in the fileset, note that they will run in parallel. Because of that, we suggest reducing the concurrency slightly.

Below some multi-services fileset examples:

Listing 182: **Fileset for all data belonging to a user**

```
Fileset {
   Name = fs-m365-adelev-user
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive,email,calendar,contact,onenote,tasks,chat␣
↪tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com␣
↪concurrent_threads=2"
   }
}
```

Listing 183: **Fileset for all data belonging to a group**

```
Fileset {
   Name = fs-m365-devteam-group
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=drive,calendar,onenote,tasks,teams␣
↪tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 group=DevTeam concurrent_threads=2"
   }
}
```

Listing 184: **Fileset for all data belonging to a site**

```
Fileset {
   Name = fs-m365-mysite-site
   Include {
      Options { signature = MD5 }
      Plugin = "m365: service=sharepoint,drive,onenote tenant=57uia43-d107-
↪17a2-a2g2-aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 site=MySite sharepoint_include_drive_items=no␣
↪sharepoint_system_include=yes drive_system_include=yes concurrent_threads=2"
   }
}
```

Note: Activating proxy mody will route all the requests through the proxy. However, it is needed DNS resolution to be working separately. Hence, the client where the FD is running will need to have a proper and working DNS Server configured. Other option is to setup the /etc/hosts file manually with the IP addresses of Microsoft authentication servers, but they could change over time. At the time of writing they are: 40.126.31.2 login.microsoftonline.com 20.190.160.131 login.microsoft.com

Advanced common parameters

Following parameters are common to all M365 modules (and even with some other plugins), but are advanced ones. They should not be modified in most common use cases.

Option	Re-quire	Default	Values	Ex-am-ple	Description
stream_sl	No	1	Positive integer (1/10 sec-conds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_m	No	120	Positive integer (sec-onds)	360	Max wait time for FD to answer packet requests
time_max	No	86400	Positive integer (sec-onds)	4320	Maximum time to wait to ovewrite a debug log that was marked as being used by other process
log-ging_max	No	50MB	String size	300M	Maximum size of a single debug log fileGenerates the working/m365/m365-debug.log* files containing debut information which is more complete with a greater debug number
log-ging_max	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rollin	No	m365.log.% MMM}.log	No, Yes	Yes	Log patter for rotated log files
split_confi	No	=	Charac-ter	:	Character to be used in config_file parameter as separator for keys and values
opener_qu	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object opener queue is full
pub-lisher_que	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The ".log" files are rotated automatically. Currently each file can be 50Mb at maximum and the plugin will keep 25 files.
  - This behavior can be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The ".err" file can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like 'logrotate'.

- Backups in paralel and also failed backups will generate several log files. For example: m365-debug-0.log, m365-debug-1.log…

Tuning parameters

These set of parameters are common to all modules and they are advanced ones. They should not be modified in general. They can be used to tune the behavior of the plugin to be more flexible in particular bad network environments or when significant job concurrency is happening, etc.

Option	Require	Default	Values	Example	Description
back	No	100	0-500	1	Number of maximum enqueued internal operations between service static internal threads (there are 3 communicating through queues with the set size: service fetcher, service opener and general publisher to bacula core). This could potentially affect graph api concurrent requests and consequently, Graph throttling. It is only needed to modify this parameter, in general, if you are going to run different jobs in parallel
concurrent_	No	10	0-100	1	Number of maximum concurrent backup threads running in parallel in order to fetch or open data for running download actions. This means every service fetcher and service opener will open this number of child concurrent threads. This will affect graph api concurrent requests. Graph API can throttle requests depending on a variety of circumstances, but this parameter impacts it directly. It is only needed to modify this parameter, in general, if you are going to run different jobs in parallel. If you want to have a precise control of your parallelization through different jobs, please set up this value to 1. Please be careful also with the memory requirements, multi-threaded increases very significantly memory consumption per job
concurrent_	No	5	0-20	1	Number of maximum concurrent backup page listing threads running in parallel in order to fetch sets of data for some modules. Currently it's only used in the email module. This parameter will also affect graph api concurrent requests. Graph API can throttle requests depending on a variety of circumstances, but this parameter impacts it directly. It is only needed to modify this parameter, in general, if you are going to run different jobs in parallel. If you want to have a precise control of your parallelization through different jobs, please set up this value to 1. Please be careful also with the memory requirements, multi-threaded increases very significantly memory consumption per job
grap	No	200	1-500	350	Number of maximum elements got from Graph API for each page of objects. Higher number implies less requests, but more memory and more time for each request
grap	No	9000	Positive integer (milliseconds)	6000	Graph call timeout inside HttpClient
grap	No	300	Positive integer (milliseconds)	3000	Graph read timeout inside HttpClient
grap	No	5	Positive integer (num	10	Graph number of retries for retry-candidate requestsInclude some stats information in the joblog. Useful to measure task times

Note: graph_list_page_size had default value 500 before BEE 14.0.7. A higher value for this parameter can improve the performance at it reduces the number of API calls done to M365 service. However, the service can also be overloaded and return more HTTP 503 errors (Bad Gateway), especially for the email module. Starting from version 16.0.3, default values for backup_queue_size and concurrent_threads have been increased, also the allowed ranges.

Entity parameters

The following list of parameters are commonly shared through any module used into the same fileset line and are intended to select the target entities to backup. Every module subsection mentions what entities are supported too.

Option	Required	Default	Values	Example	Services	Description
user	No		Valid email addresses of existing users on the selected tenant separated by ‘,’	AlexW@ LeeY@y	email, drive, contact, calendar, onenote chat, tasks	Backup mailboxes, drive unit spaces, categories, or whatever service is selected of this list of users. If no user is provided: - The backup will be done for all accessible users of the given tenant if no other entry has value either (group or site parameters)
user_	No		Valid email addresses of existing users on the selected tenant separated by ‘,’	LauraG Amanda T@yo	email, drive, contact, calendar, onenote chat, tasks	Exclude selected mailboxes or onedrive spaces. If this is the only parameter found for selection, all elements will be included and this list will be excluded
user_	No		Valid regex	.*@management\.my	email, drive, contact, calendar, onenote chat, tasks	Backup matching user mailboxes or onedrive spaces. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them
user_	No		Valid regex	.*@gues	email, drive, contact, calendar, onenote chat, tasks	Exclude matching user mailboxes or onedrive spaces from the selection. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded
site	No		Valid names of existing user sharepoint sites on the selected tenant separated by ‘,’	Communication site, Dev site	drive, sharepoint, onenote	Backup onedrive site library space (*OneDrive service*) belonging to this list of sharepoint sites or the site itself (*Sharepoint service*). If no site name is provided: - **OneDrive**: user and site variables will be checked. If no one has value either, backup will be done for all accessible users, groups or sites of the given tenant - **Sharepoint**: all accessible sites of the tenant will be backed up
site_	No		Valid names of existing user sharepoint sites on the selected	Test site	drive, sharepoint, onenote	Exclude listed sites from backup. If this is the only parameter found for selection, all elements will be included and this list will be excluded

Backup parameters

Please, check the specific module pages in order to see backup parameters that are applicable only to each of them:

- OneDrive
- Emails/Mailboxes
- Calendars
- Contacts
- Sharepoint
- OneNote
- Tasks
- Teams
- Chat
- Activity

Deprecated parameters

The following parameters have been deprecated since version 12.8.3, in order to provide more flexibility. General shared parameters have been transformed in specific ones for each service

- files -> email_files, drive_files, calendar_files, contact_files
- files_exclude -> email_files_exclude, drive_files_exclude, calendar_files_exclude, contact_files_exclude
- files_regex_include -> email_files_regex_include, drive_files_regex_include, calendar_files_regex_include, contact_files_regex_include
- files_regex_exclude -> email_files_regex_exclude, drive_files_regex_exclude, calendar_files_regex_exclude, contact_files_regex_exclude
- multilevel_attach -> email_multilevel_attach, calendar_multilevel_attach
- exclude_attachments -> email_exclude_attachments, calendar_exclude_attachments
- system_include -> drive_system_include, sharepoint_system_include
- version_history -> drive_version_history, sharepoint_version_history
- backup_threads -> backup_queue_size

Restore parameters

The plugin is able to restore to the local file system on the server where the File Daemon is running or to the Microsoft 365 environment. The method is selected based on the value of the *where* parameter at restore time:

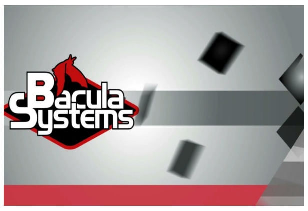
- Empty or '/' (example: where=/) → Microsoft 365 restore will be triggered
- Any other path for where (example: where=/tmp) → Local file system restore will be triggered

When using the Microsoft 365 restore option, the following parameters may be modified by selecting ‘Plugin Options’ during the bconsole restore session:

Option	Required	Default	Values
destination_user	No		Existing email address on the target Azure A
destination_group	No		Existing group name address on the target A
destination_site	No		Existing site name/display name on the targ
destination_path	No		Existing path on the selected user (mailfold
destination_drive	No		Existing drive name or drive id in the destin
send_report	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Y
allow_duplicates	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
drive_skip_sharedwitme	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Y
skip_versions	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
restore_share_permissions	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Y
drive_send_invitations	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Y
drive_invitations_message	No		String
calendar_name	No		String
sharepoint_list_name	No		String
sharepoint_newsite_name	No		String
sharepoint_newsite_owner	No		String
sharepoint_skip_system	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
sharepoint_local_template_path	No		String
notebook_name	No		String
notesection_name	No		String
notesection_group_name	No		String
team_name	No		String
team_channel_name	No		String
team_private_channels_mode	No	DELEGATED	DELEGATED, PUBLIC, SKIP
team_guest_members_enable	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Y
chat_topic	No		String
tasklist_name	No		String
tasklist_skip_sharedwithme	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
plan_name	No		String
plan_create_tab	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
local_path_json_objects	No	{path}/bacula-restores	String
foreign_container_generation	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Y
tenant	No		A valid tenant id string where some bacula
appid	No		Appid of bacula-m365-plugin-standalone ap
objectid	No		Object id associated to set up appid
secret	No		Valid secret associated to appid
debug	No	0	0, 1, 2 ,3, 4, 5, 6, 7, 8, 9

Operations

To learn about backup and restore from a video, click on the image below:



Backup

Microsoft 365 plugin backup configurations currently have one specific requirement in the Job resource. Below we show some examples.

Job Example

The only special requirement with M365 jobs is that Accurate mode backups must be disabled, as this feature is not supported at this time.

Listing 185: **Job Example**

```
Job {
Name = m365-mytenant-backup
Fileset = fs-m365-email-all
Accurate = no
...
}
```

Fileset Examples

The plugin supports enough flexibility to configure almost any type of desired backup. Multiple *Plugin=* lines should not be spoecified in he Include section of a Fileset for the M365 Plugin.

Fileset examples for every supported service are linked below. For common purposes, the following two examples show how to configure an external config file or configure the number of threads:

Setup external config file:

Listing 186: **Fileset Example**

```
Fileset {
   Name = FS_M365_DRIVE
   Include {
      Options {
        signature = MD5
      }
      Plugin = "m365: config_file=/opt/bacula/etc/m365.settings service=drive"
   }
}
```

Listing 187: **Settings file**

```
$ cat /opt/bacula/etc/m365.settings
tenant=57uia43-d107-17a2-a2g2-aa53c10tdahc
objectid=56ddf1h9-eb5d-42nf-bac7-7b019fd284g5
```

Increase number of threads:

Listing 188: **Fileset Example**

```
Fileset {
   Name = fs-m365-email-adelev
   Include {
      Options {
        signature = MD5
      }
      Plugin = "m365: service=email tenant=57uia43-d107-17a2-a2g2-
↪aa53c10tdahc objectid=56ddf1h9-eb5d-
   42nf-bac7-7b019fd284g5 user=adelev@baculaenterprise.onmicrosoft.com backup_
↪threads=10"
   }
}
```

More fileset examples for:

- OneDrive
- Emails/Mailboxes
- Calendars
- Contacts
- Sharepoint
- OneNote
- Tasks
- Teams
- Chat
- Activity

Restore

Restore operations are done using standard **Bacula Enterprise** bconsole commands.

The *where* parameter controls if the restore will be done locally to the File Daemon's file system or to the Microsoft 365 service:

- where=/ or empty value → Restore will be done over M365
- where=/any/other/path → Restore will be done locally to the File Daemon file system

Restore options are described in the restore-params section of this document, so here we are going to simply show an example restore session, particularly this example is about OneDrive service:

Listing 189: **Restore Drive Bconsole Session**

```
*restore where=/

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
1: List last 20 Jobs run
2: List Jobs where a given File is saved
3: Enter list of comma separated JobIds to select
4: Enter SQL list command
5: Select the most recent backup for a client
6: Select backup for a client before a specified time
7: Enter a list of files to restore
8: Enter a list of files to restore before a specified time
9: Find the JobIds of the most recent backup for a client
10: Find the JobIds for a backup for a client before a specified time
11: Enter a list of directories to restore for found JobIds
12: Select full restore to a specified Job date
13: Select object to restore
14: Cancel
Select item: (1-14): 5
Automatically selected Client: 127.0.0.1-fd
Automatically selected Fileset: FS_M365_DRIVE
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
| jobid | level | jobfiles | jobbytes    | starttime           | volumename   ␣
↪     |
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
| 11    | F     | 190      | 332,978,505 | 2021-01-22 10:39:34 | TEST-2021-01-
↪22:0 |
| 12    | I     | 1        | 550         | 2021-01-22 10:43:05 | TEST-2021-01-
↪22:0 |
+-------+-------+----------+-------------+---------------------+--------------
↪-----+
You have selected the following JobIds: 11,12

Building directory tree for JobId(s) 11,12 ...␣
↪++++++++++++++++++++++++++++++++++++++++++++++++
188 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd /@m365/baculaenterprise/adelev@baculaenterprise.onmicrosoft.com/drive/
```

(continues on next page)

(continued from previous page)

```
↪root:
cwd is: /@m365/baculaenterprise/adelev@baculaenterprise.onmicrosoft.com/drive/
↪root:/
$ ls
Docs/
pluginTest.drive.deltaLink
sharedWithMe/
test4###v1.0_2020-11-25_153507.html
test4###v2.0_2020-11-25_153507.html
test4###v3.0_2020-12-02_180612.html
test4###v4.0_2020-12-02_180612.html
test4###v5.0_2020-12-03_134422.html
test4###v6.0_2020-12-02_180612.html
test4.html
testlink123###v1.0_2020-11-26_123244.url
testlink123###v2.0_2020-11-26_123244.url
testlink123###v3.0_2020-12-02_180613.url
testlink123###v4.0_2020-12-02_180613.url
testlink123.url
$ mark *
189 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.3.bsr

The Job will require the following (*=>InChanger):
Volume(s) Storage(s) SD Device(s)
===========================================================================

TEST-2021-01-22:0 File FileStorage

Volumes marked with "*" are in the Autochanger.
189 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName: RestoreFiles
Bootstrap: /tmp/regress/working/127.0.0.1-dir.restore.3.bsr
Where: /
Replace: Always
Fileset: Full Set
Backup Client: 127.0.0.1-fd
Restore Client: 127.0.0.1-fd
Storage: File
When: 2021-01-22 11:54:55
Catalog: MyCatalog
Priority: 10
Plugin Options: *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
1: Level
2: Storage
3: Job
```

(continues on next page)

(continued from previous page)

```
4: Fileset
5: Restore Client
6: When
7: Priority
8: Bootstrap
9: Where
10: File Relocation
11: Replace
12: JobId
13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : m365: service=drive tenant="574dda43-d107-48e2-a7f2-
↪aa51c10bdaec" objectid="31ddf4b1-
ed5d-432f-bac7-7b946fd23394" user="adelev@baculaenterprise.onmicrosoft.com"␣
↪drive_version_history=yes debug=3
Plugin Restore Options
Option Current Value Default Value
destination_user: *None* (*None*)
destination_path: *None* (*None*)
send_report: *None* (0)
email_allow_duplicates: *None* (1)
drive_skip_sharedwithme: *None* (0)
drive_skip_versions: *None* (1)
drive_restore_share_permissions: *None* (0)
drive_send_invitations: *None* (0)
drive_invitations_message: *None* (*None*)
debug: *None* (*None*)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
1: destination_user (Destination User)
2: destination_path (Destination Path in M365)
3: send_report (Send report of the restore operation to the affected user)
4: email_allow_duplicates (Allow Duplicate Emails)
5: drive_skip_sharedwithme (Skip restoring shared with me elements even if␣
↪they are selected)
6: drive_skip_versions (Skip restoring file former versions (tagged with '##
↪#date') even if they are selected)
7: drive_restore_share_permissions (Restore share permissions of items so␣
↪they are shared if they originally
were)
8: drive_send_invitations (Send email invitations for restored OneDrive␣
↪shares)
9: drive_invitations_message (Set invitations message for restored OneDrive␣
↪shares)
10: debug (Change debug level)
Select parameter to modify (1-10): 2
Please enter a value for destination_path: MY_RESTORE_PATH
Plugin Restore Options
Option Current Value Default Value
destination_user: *None* (*None*)
destination_path: MY_RESTORE_PATH (*None*)
send_report: *None* (0)
```

(continues on next page)

(continued from previous page)

```
email_allow_duplicates: *None* (1)
drive_skip_sharedwithme: *None* (0)
drive_skip_versions: *None* (1)
drive_restore_share_permissions: *None* (0)drive_send_invitations: *None* (0)
drive_invitations_message: *None* (*None*)
debug: *None* (*None*)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
JobName: RestoreFiles
Bootstrap: /tmp/regress/working/127.0.0.1-dir.restore.3.bsr
Where: /
Replace: Always
Fileset: Full Set
Backup Client: 127.0.0.1-fd
Restore Client: 127.0.0.1-fd
Storage: File
When: 2021-01-22 11:54:55
Catalog: MyCatalog
Priority: 10
Plugin Options: User specified
OK to run? (yes/mod/no): yes
Job queued. JobId=14
```

Restore by Service

In this section some example restore configurations will be shown:

- OneDrive
- Email
- Calendar
- Contacts
- Sharepoint
- Onenote
- Tasks
- Teams
- Chat
- Activity

Cross-tenant Restore

You can perform cross-tenant restores using the restore variables: - tenantid - appid - objectid - secret

Obviously, it is needed to set up the destination tenant values, where one of the authentication methods should be applied first.

If you are using the standalone app authentication model, you will need to use the four values in your restore session.

If you want to use the common app authentication model, you won't need to put the secret. On the other hand, please, ask the bacula enterprise edition support team in order to get the correct value for appid.

List

It is possible to list information using the bconsole *.ls* command and providing a path. In general, we need to provide the service parameter and a path representing a folder or object (like calendars).

There are some general commands (user, groups, sites), the rest of the commands need to point to the correct service.

Below some examples:

List general info: Users, groups, sites

Here we are showing these 3 commands using the bconsoel *.ls* command, but notice you may also use them with the query interface (keep your variable values, but apply something like: *.query plugin="... " client=xxxx parameter=xxx*). Note that the referenced 'config_file' should contain the connection parameters (tenant, objectid, appid and secret).

Listing 190: **List example: General information**

```
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"␣
↪client=127.0.0.1-fd path=/user
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
ptcomm: Starting Plugin Job
ptcomm: Finished reading Plugin Params
ptcomm: Backend connection to M365 stablished
ptcomm: Connected to tenant: baculaenterprise
ptcomm: Starting ListingStart
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪AdeleV@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪AlexW@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪DiegoS@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪jane@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪GradyA@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
↪HenriettaM@baculaenterprise.onmicrosoft.com
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /
```

(continues on next page)

(continued from previous page)

```
↪IsaiahL@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪JohannaL@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪JoniS@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪john@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪LeeG@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪LidiaH@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪LynneR@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪MeganB@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪MiriamG@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪NestorW@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪PattiF@baculaenterprise.onmicrosoft.com
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /
↪PradeepG@baculaenterprise.onmicrosoft.com
2000 OK estimate files=18 bytes=0

*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"
client=127.0.0.1-fd path=/group
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
ptcomm: Starting Plugin Job
ptcomm: Finished reading Plugin Params
ptcomm: Backend connection to M365 stablished
ptcomm: Connected to tenant: baculaenterprise
ptcomm: Starting ListingStart
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Contoso Team
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Dev Team
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /baculaenterprise
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /All Company
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Bacula Systems␣
↪Team Site
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Dev people Group
2000 OK estimate files=6 bytes=0

*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"
client=127.0.0.1-fd path=/site
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
ptcomm: Starting Plugin Job
ptcomm: Finished reading Plugin Params
ptcomm: Backend connection to M365 stablished
ptcomm: Connected to tenant: baculaenterprise
ptcomm: Starting ListingStart
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Communication site
-rw-r--r--     1 root     root         -1 2106-02-07 07:28:15 /Adele Vance
```

(continues on next page)

(continued from previous page)

```
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Alex Wilber
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Diego Siciliani
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Grady Archie
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Henrietta Mueller
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Isaiah Langer
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Johanna Lorenz
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Joni Sherman
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Lidia Holloway
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Lynne Robbins
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Megan Bowen
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Pradeep Gupta
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Bacula Systems␣
↪Team Site
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Work @ Contoso
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Contoso Team
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Live @ Contoso
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Give @ Contoso
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Dev Team
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Dev people Group
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Team Site
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /John Doe
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Jane Doe
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Patti Fernandez
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Lee Gu
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Miriam Graham
-rw-r--r--    1 root    root        -1 2106-02-07 07:28:15 /Nestor Wilke
2000 OK estimate files=28 bytes=0

# Query equivalents follow
*.query plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"␣
↪client=127.0.0.1-fd parameter=user
*.query plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"␣
↪client=127.0.0.1-fd parameter=group
*.query plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"␣
↪client=127.0.0.1-fd parameter=site

# Other similar query function
*.query plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf"␣
↪client=127.0.0.1-fd parameter=team
```

The .query function to list users has an additional parameter that provides the function to filter the resulting user list by group. The parameter is called 'q_users_group' and it must be used inside the plugin line of the .query call. See the example below:

Listing 191: **List example: General information**

```
*.query client=example-fd plugin="m365: config_file=m365.conf q_users_
↪group=groupName" parameter=user
```

Note: q_users_group is available since Bacula Enterprise 18.0.8.

List Onedrive contents

Listing 192: **List example: General information**

```
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=drive user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.
↪1-fd path=/
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
m365: Starting Plugin Job
m365: Finished reading Plugin Params
m365: Connected to tenant: baculaenterprise
m365: Backend connection to M365 stablished
m365: Starting ListingStart
drwxr-xr-x   1 nobody    nogroup           26800469 2020-11-08 01:34:35 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/root/
drwxr-xr-x   1 nobody    nogroup           9876293 2020-12-03 15:28:31 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/
-rw-r-----   1 nobody    nogroup           217873 2021-04-13 13:29:56 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Hats.JPG
-rw-r-----   1 nobody    nogroup   1914266 2021-02-01 17:45:34 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/lombok.jar
-rw-r-----   1 nobody    nogroup           785043 2021-02-01 17:45:56 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/noShared.dat
-rw-r-----   1 nobody    nogroup           4 2021-02-01 17:45:35 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/test2.doc
-rw-r-----   1 nobody    nogroup           4 2020-11-25 15:35:07 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/test4.html
-rw-r-----   1 nobody    nogroup           42 2020-11-26 12:32:44 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/testlink123.url
drwxr-xr-x   1 nobody    nogroup           -1 2106-02-07 07:28:15 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/sharedWithMe/
2000 OK estimate files=23 bytes=53,600,938
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf
```

(continues on next page)

(continued from previous page)

```
service=drive user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.1-
↪fd path=/sharedWithMe
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
m365: Starting Plugin Job
m365: Finished reading Plugin Params
m365: Connected to tenant: baculaenterprise
m365: Backend connection to M365 stablished
m365: Starting ListingStart
-rw-r-----   1 nobody    nogroup              11722 2021-04-13 17:42:06 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Aeque.ppt
-rw-r-----   1 nobody    nogroup              4039 2021-04-13 17:42:02 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Voluptatibus.txt
-rw-r-----   1 nobody    nogroup              7697 2021-04-13 17:41:59 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Nullam.jpeg
-rw-r-----   1 nobody    nogroup              21031 2021-04-13 17:41:55 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Graeci.ppt
-rw-r-----   1 nobody    nogroup              21417 2021-04-13 17:41:52 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Habeo.txt
-rw-r-----   1 nobody    nogroup              15354 2021-04-13 17:41:47 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Atomorum.jpeg
-rw-r-----   1 nobody    nogroup              22217 2021-04-13 17:41:43 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Faucibus.ppt
-rw-r-----   1 nobody    nogroup              15714 2021-04-12 11:13:52 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Sodales.ppt
-rw-r-----   1 nobody    nogroup              14371 2021-04-12 11:13:48 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Cum.txt
-rw-r-----   1 nobody    nogroup              11059 2021-04-12 11:13:44 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Sapien.jpeg
-rw-r-----   1 nobody    nogroup              14144 2021-04-12 11:13:41 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Habitasse.ppt
```

(continues on next page)

(continued from previous page)

```
-rw-r-----   1 nobody    nogroup              12594 2021-04-12 11:13:37 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Deserunt.txt
-rw-r-----   1 nobody    nogroup              14923 2021-04-12 11:13:32 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Postulant.jpeg
-rw-r-----   1 nobody    nogroup               21307 2021-04-12 11:13:29 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪SharePoint App/Tation.ppt
drwxr-xr-x   1 nobody    nogroup               0 2021-01-08 14:32:20 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪john@baculaenterprise.onmicrosoft.
com/JohnFolder/
-rw-r-----   1 nobody    nogroup               1442312 2021-01-08 11:57:09 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪john@baculaenterprise.onmicrosoft.
com/developers.pdf
-rw-r-----   1 nobody    nogroup               630174 2021-01-08 11:57:08 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/sharedWithMe/
↪john@baculaenterprise.onmicrosoft.
com/console.pdf
2000 OK estimate files=17 bytes=2,280,075*.ls plugin="m365: config_file=/opt/
↪bacula/etc/m365/example.tenant.conf
service=drive user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.1-
↪fd path=/Docs
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
m365: Starting Plugin Job
m365: Finished reading Plugin Params
m365: Connected to tenant: baculaenterprise
m365: Backend connection to M365 stablished
m365: Starting ListingStart
drwxr-xr-x   1 nobody    nogroup               9876293 2020-12-03 15:28:31 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/
drwxr-xr-x   1 nobody    nogroup               793475 2020-11-25 15:34:49 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/Simple␣
↪Sub Folders/
-rw-r-----   1 nobody    nogroup               6383460 2020-12-08 10:13:11 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/E16WKAMS.
↪10E.zip
-rw-r-----   1 nobody    nogroup               1914266 2020-12-08 10:13:32 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/lombok.jar
-rw-r-----   1 nobody    nogroup               785043 2021-01-12 10:12:04 /
```

(continues on next page)

(continued from previous page)

```
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/noShared.
↪dat
-rw-r-----   1 nobody    nogroup
24 2020-11-25 15:35:05 /baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/test1.txt
-rw-r-----   1 nobody    nogroup                13 2021-01-11 15:39:57 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/test11.txt
-rw-r-----   1 nobody    nogroup                4 2020-11-25 15:35:05 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/test2.doc
-rw-r-----   1 nobody    nogroup                4 2020-11-25 15:35:05 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/test3.pdf
-rw-r-----   1 nobody    nogroup                4 2020-12-03 16:52:32 /
↪baculaenterprise/users
/adelev@baculaenterprise.onmicrosoft.com/drives/onedrive/root:/Docs/test4.html
2000 OK estimate files=10 bytes=19,752,586
```

Other list examples

Please note that the following examples are not available in .query mode, they are only for .ls command.

Listing 193: **List example: General information**

```
# List contents of Inbox folder of user adelev@baculaenterprise.onmicrosoft.
↪com
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=email user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.
↪1-fd path=/Inbox

# List contents of Sent folder of user adelev@baculaenterprise.onmicrosoft.com
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=email user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.
↪1-fd path=/Sent

# List all contacts of user adelev@baculaenterprise.onmicrosoft.com
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=contacts user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.
↪0.1-fd path=/

# List contacts of user adelev@baculaenterprise.onmicrosoft.com and folder␣
↪MyContactsDir
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=contacts user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.
↪0.1-fd path=/MyContactsDir

# List calendar events of user adelev@baculaenterprise.onmicrosoft.com and␣
↪her calendar MyCalendar
```

(continues on next page)

(continued from previous page)

```
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=calendar user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.
↪0.1-fd path=/MyCalendar

# List onenote pages of user adelev@baculaenterprise.onmicrosoft.com and her␣
↪section secA included in section Group groupA
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=onenote user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.
↪0.1-fd path=/MyNotebook/sectionGroups/groupA/sections/secA

# List contents of OneDrive folder 'folderA' of user adelev@baculaenterprise.
↪onmicrosoft.com
*.ls plugin="m365: config_file=/opt/bacula/etc/m365/example.tenant.conf␣
↪service=drive user=adelev@baculaenterprise.onmicrosoft.com" client=127.0.0.
↪1-fd path=/folderA
```

Performance command

This plugin provides a query command allowing to check the performance of a given setup. The process consist on uploading some randomly generated files of a customizable size (and a customizable number of them) to a target tenant, then download them using the same method the backup itself employs. Upload and download times are measured. An average is calculated and all the information is presented to the user.

Please, note that this command is currently working in single-threaded mode, while the plugin is running with multiple threads (configurable through concurrent_threads variable), so you could have better results with the actual backup process.

The usage of this command is:

Listing 194: **Query perf**

```
*.query client=<your-client> plugin="m365: tenant=<tenantId> objectid=
↪<objectId> user=<your-user> service=<service>" parameter=perf
```

Currently, theh plugin supports two types of performance commands: perf for onedrive (default) and perf for email. So the value service should be: drive or email.

The behavior of this service from performance perspective can differ significantly. Below we detail some important aspects:

- On Onedrive and other services implying user files, it is needed to request a ‘download session’ for each file.
  - Once we have it, the download starts. This requires significantly more time than just retrieving simple objects.
- In general, objects are not retrieved one by one. We request for lists of objects, that later on are processed.
  - For objects that are not pointing to any other file (as emails without attachments), the time needed per object is very low.

Drive perf command

For Onedrive service, in addition to user, you can also use group or site (site name or siteId) for this command. The final output will be similar to the following:

Listing 195: **Query perf drive output**

```
...
date=2021-09-06 17:25:56
tenant_name=johndoe.onmicrosoft.com
upload-speed-average=3.3 MiB/s
download-speed-average=32.2 MiB/s
latency-average=482.00 ms
console=---- Test END ----
```

You can control the size and the number of files used using the following query parameters: - q_perf_size (in bytes) - q_perf_files (number of files)

By default, their values are: - q_perf_size = 209715200 - q_perf_files = 3

Below we provide a usage example:

Listing 196: **Query perf drive sample**

```
*.query client=127.0.0.1-fd plugin="m365:
tenant=f421f256-748a-4a80-90da-bf47y66fa166 objectid=90173rea-c144-46ye-8f93-
↪4uh87bcd6d47 site=\"johndoe.sharepoint.com,fda66478-aab5-4896-a6ff-
↪e0fb02af2a40,c024ee54-dcc0-4f1e-ab76-671f6eb87fda\" service=drive q_perf_
↪files=5 q_perf_size=314572800" parameter=perf
```

Email perf command

For Email service, specifying a user is mandatory. The command can generate simple emails, but also emails with one attachment. The final output can be similar to the following:

Listing 197: **Query perf drive output**

```
...
console=---- Summary ----
date=2021-11-11 17:05:18
tenant_name=johndoe.onmicrosoft.com
upload-speed-average=44.6 KiB/s
download-speed-average=220.5 KiB/s
latency-average-per-list-call=510.83 ms
latency-average-per-message=36.18 ms
latency-average-attachments=617.64 ms
console=---- Test END ----
```

The behavior of the email perf command is a little different from the Onedrive one. Instead of uploading/downloading one message at a time, here everything is uploaded first. After that everything is downloaded. The reason is here the backup process uses lists of objects and, unless there are attachments implied, that's all, the objects are simply stored from that list.

Similarly to the onedrive perf command, you can control the size and the number of emails. However, there is an extra parameter to control the size of the attachments:

- q_perf_size (in bytes)
- q_perf_size_attachments (in bytes)
  - If not specified, no attachment will be generated
  - If specified, one simple text attachment will be generated together with each email
- q_perf_files (number of files)

By default, their values are:

- q_perf_size = 10240
- q_perf_size_attachments = 0
- q_perf_files = 50

Below we provide a usage example:

Listing 198: **Query perf email sample**

```
*.query client=127.0.0.1-fd plugin="m365:
tenant=f421f256-748a-4a80-90da-bf47y66fa166 objectid=90173rea-c144-46ye-8f93-
↪4uh87bcd6d47 site=\"johndoe.sharepoint.com,fda66478-aab5-4896-a6ff-
↪e0fb02af2a40,c024ee54-dcc0-4f1e-ab76-671f6eb87fda\" service=email q_perf_
↪size_attachment=204800" parameter=perf
```

Other query commands

Sometimes there is an element that causes some error while fetching it in the email service and it's difficult to identify it by its reported debug id. For this situation, there is a decoding command:

As the example shows, the query parameter is 'query=decode|{url}'. Url should be the URL until the element we are interested into (so the attachment here).

Another useful command shows all the options the plugin can accept as parameters, categorized by section:

The output of this command will return a json structure with all the available options.

Update App Permissions

Bacula Enterprise Microsoft 365 Plugin is an application that is under continuous evolution. In consequence it may happen that new features require new permissions to function properly.

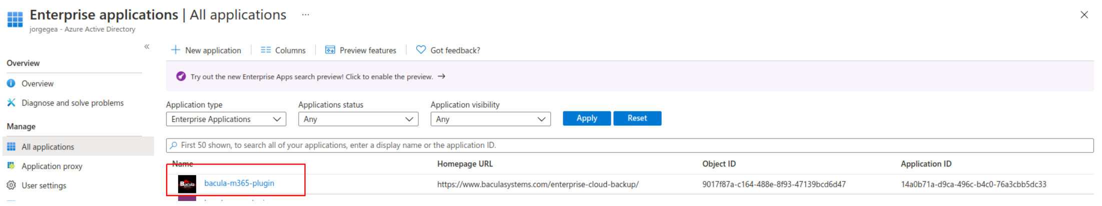
If you are using Authorization Method A: Common app model with bacula-m365-plugin (DEPRECATED)

Bacula Systems will add the required new permissions to the central bacula-m365-plugin app registered in Azure AD (or Microsoft Entra ID).

If you are already using the plugin through this central shared application (not recommended, as we recommend to use the Authentication “Standalone app model”), you will need to update the permissions of your own app (usually bacula-m365-plugin app itself, available under ‘enterprise applications’ option in your Azure Active Directory application).

To refresh and allow any new permission to your app, apply the following procedure:

1. Access the Azure Portal with a Tenant admin user
2. Locate your bacula-m365-plugin app under ‘Enterprise applications’ and click on it.



Enterprise applications | All applications

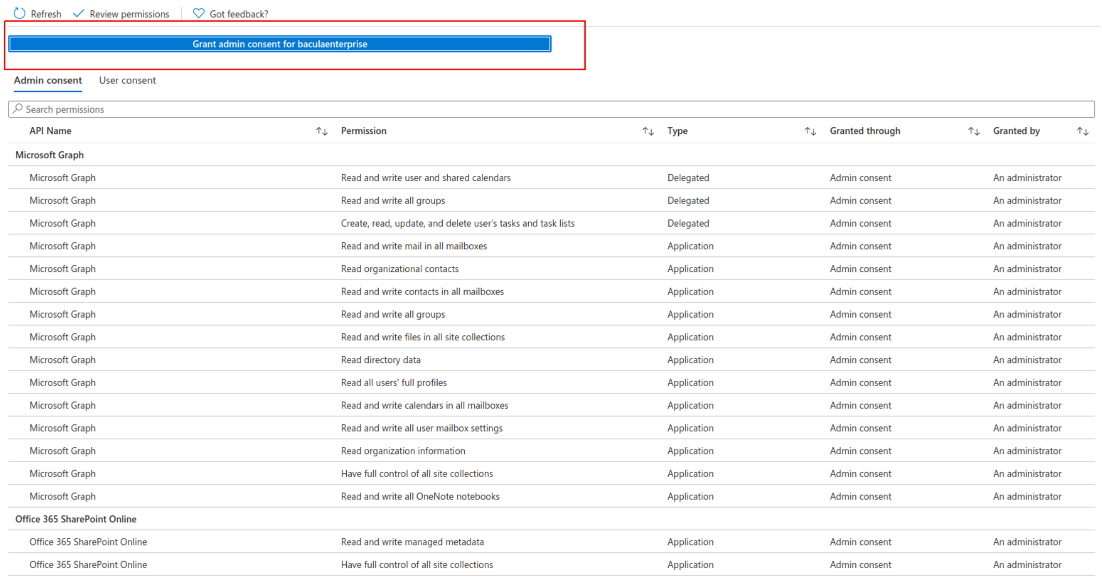
jorgegea - Azure Active Directory

Application type: Enterprise Applications
Applications status: Any
Application visibility: Any

First 50 shown, to search all of your applications, enter a display name or the application ID.

Name	Homepage URL	Object ID	Application ID
bacula-m365-plugin	https://www.baculasystems.com/enterprise-cloud-backup/	9017f87a-c164-488e-8f93-47139bcd6d47	14a0b71a-d9ca-496c-b4c0-76a3cbb5dc33

3. In the left menu, click on ‘Permissions’ and then click the big blue button showing ‘Grant admin consent for <yourtenantname>’:



API Name	Permission	Type	Granted through	Granted by
Microsoft Graph				
Microsoft Graph	Read and write user and shared calendars	Delegated	Admin consent	An administrator
Microsoft Graph	Read and write all groups	Delegated	Admin consent	An administrator
Microsoft Graph	Create, read, update, and delete user's tasks and task lists	Delegated	Admin consent	An administrator
Microsoft Graph	Read and write mail in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read organizational contacts	Application	Admin consent	An administrator
Microsoft Graph	Read and write contacts in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read and write all groups	Application	Admin consent	An administrator
Microsoft Graph	Read and write files in all site collections	Application	Admin consent	An administrator
Microsoft Graph	Read directory data	Application	Admin consent	An administrator
Microsoft Graph	Read all users' full profiles	Application	Admin consent	An administrator
Microsoft Graph	Read and write calendars in all mailboxes	Application	Admin consent	An administrator
Microsoft Graph	Read and write all user mailbox settings	Application	Admin consent	An administrator
Microsoft Graph	Read organization information	Application	Admin consent	An administrator
Microsoft Graph	Have full control of all site collections	Application	Admin consent	An administrator
Microsoft Graph	Read and write all OneNote notebooks	Application	Admin consent	An administrator
Office 365 SharePoint Online				
Office 365 SharePoint Online	Read and write managed metadata	Application	Admin consent	An administrator
Office 365 SharePoint Online	Have full control of all site collections	Application	Admin consent	An administrator

4. Wait for a few minutes
5. Refresh the page
6. See the new permissions in the list

New releases needing any new permissions will properly reflect that information with the specific permissions list.

If you are using Authorization Method B: Standalone app model with bacula-m365-registrator (RECOMMENDED)

You need to rerun the add-app command as explained in the Authentication section B-2.

Example:

Listing 199: **Update permissions through add-app command**

```
*.query client=example-client-fd plugin="m365: tenant=********_****_****_****_
↪************" parameter=add-app
console=---- M365 ADD APP ----
info=File /opt/bacula/etc/m365/example/bacula_m365_config.conf already exists.
↪ Contents will be just updated
info=Application named: bacula-m365-plugin-standalone already exists in␣
↪tenant: example ...
info=Updating its permissions ...
info=Permissions updated successfully
info=Service principal already exists
info=Granting admin consent for permissions ...
info=Admin consent granted
info=We don't have any valid token. It's needed to sign in
info=To sign in; use a web browser to open the page https://microsoft.com/
↪devicelogin and enter the code GJL5LGXC8 to authenticate. Please do it with␣
↪a tenant administrator user
info=Storing access configuration to the new app in file: /opt/bacula/etc/
↪m365/example/bacula_m365_config.conf
info=Access configuration successfully stored
console=---- ADD APP COMPLETED SUCCESSFULLY ----
console=----
info=Use the generated configuration file in your m365 filesets. Just add the␣
↪property config_file=/your/path/bacula_m365_config.conf to the line Plugin=
↪"m365: ..."
info=Sample: Plugin = "m365: config_file=/opt/bacula/etc/m365/example/bacula_
↪m365_config.conf service=drive"
console=----
console=----
```

Delegated Permissions

The great majority of services of the Bacula Enterprise M365 plugin are using the model of 'Application permissions'. With this model, the plugin application can directly access the implied services once a tenant admin has aproved it.

On the contrary, some other services can require the model of 'Delegated permissions' where the plugin needs to impersonate a given user in order to be capable of accessing and restoring the data inside the service. Although we will support more in the future, currently, the only service requesting delegated permissions is:

- Calendar: If any group is included as an entity to protect
- Chat: In order to be able to backup all chat messages belonging to a certain user the user in question will need to delegate the permissions as described below.

Delegated permissions or application permissions are automatically applied when they are needed. However, delegated permissions need user interaction the first time they are invoked, or if the local information about access tokens has been lost for some reason.

The user interaction can be manually triggered with a special query command that may be run as follows:

Listing 200: **query-login**

```
.query plugin="m365: tenant=e421f055-748a-4a80-90da-bf48b66fa166␣
↪objectid=9017f87a-c164-488e-8f93-47139bcd6d47" client=127.0.0.1-fd␣
↪parameter=login:peter@bacula.onmicrosoft.com
info=Connected through: bacula-m365-plugin-standalone
info=We don't have a valid token for peter@bacula.onmicrosoft.com. It's␣
↪needed to sign in
info=To sign in; use a web browser to open the page https://microsoft.com/
↪devicelogin and enter the code LXSZGLY3F to authenticate. Please do it with␣
↪peter@bacula.onmicrosoft.com
```

Note the format of the parameter, that is `parameter=login:[usermail]`.

Additionally, it is possible to log in using a group with the format: `parameter=login:[groupname]` or a site, using exactly the same format and the site name.

Listing 201: **query-login**

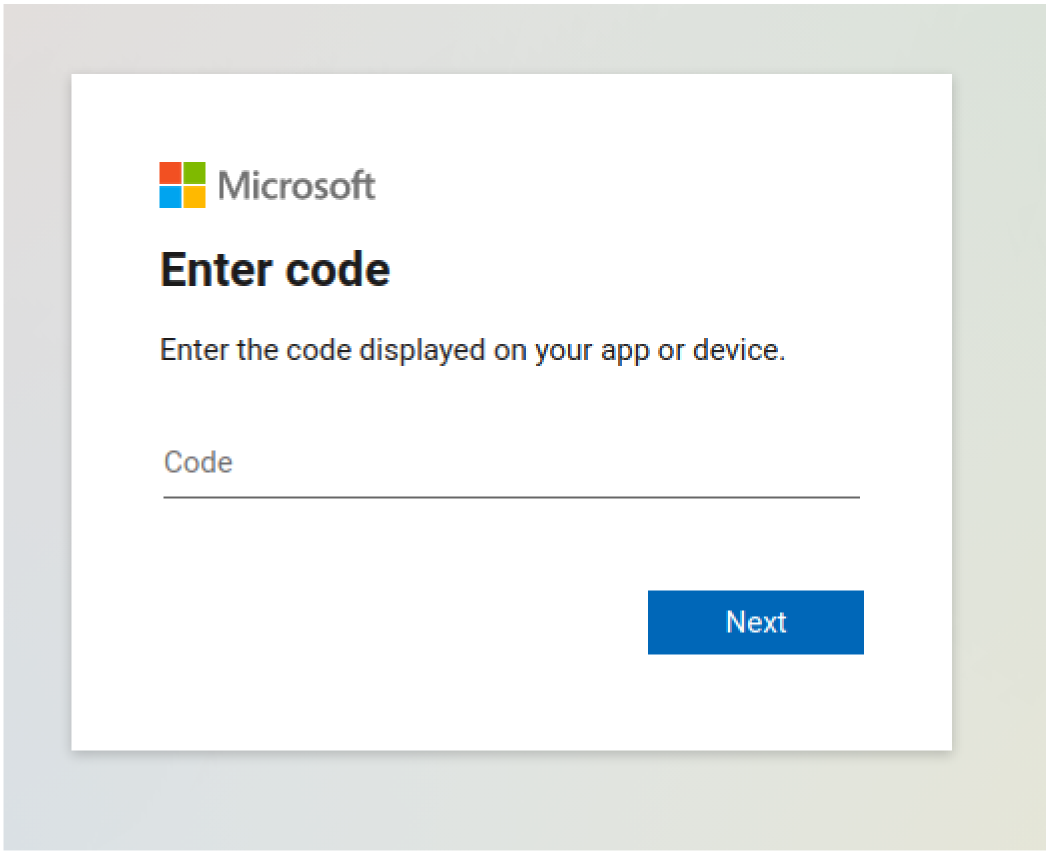
```
.query plugin="m365: tenant=e421f055-748a-4a80-90da-bf48b66fa166␣
↪objectid=9017f87a-c164-488e-8f93-47139bcd6d47" client=127.0.0.1-fd␣
↪parameter=login:Dev/Team/1
info=Connected through: bacula-m365-plugin-standalone
info=We don't have a valid token for any group owner of Dev/Team/1. It's␣
↪needed to sign in
info=To sign in; use a web browser to open the page https://microsoft.com/
↪devicelogin and enter the code LYZF9EC6Q to authenticate. Please do it with␣
↪some owner of the group Dev/Team/1
info=Owners found: jorge@bacula.onmicrosoft.com;AdeleV@bacula.onmicrosoft.com
info=We have a valid token got from account:
user=jorge@bacula.onmicrosoft.com
info=It will be automatically refreshed when needed. But the current␣
↪expiration date is: 2025-06-27T12:33:49Z
info=Backups using services with delegated permissions around jorge@bacula.
↪onmicrosoft.com won't ask for intervention
```

We only need to follow the instructions, open a browser, enter the code and use the correct user to login (which means a user with access to the resource we want to backup).

For example, if we are working with the calendar of a group, we will need a user with access to that calendar (in general, to protect calendars associated to groups, it will be necessary to create a special backup user with access to all of them and perform the login with, which will avoid the need of performing the login process for every different group).

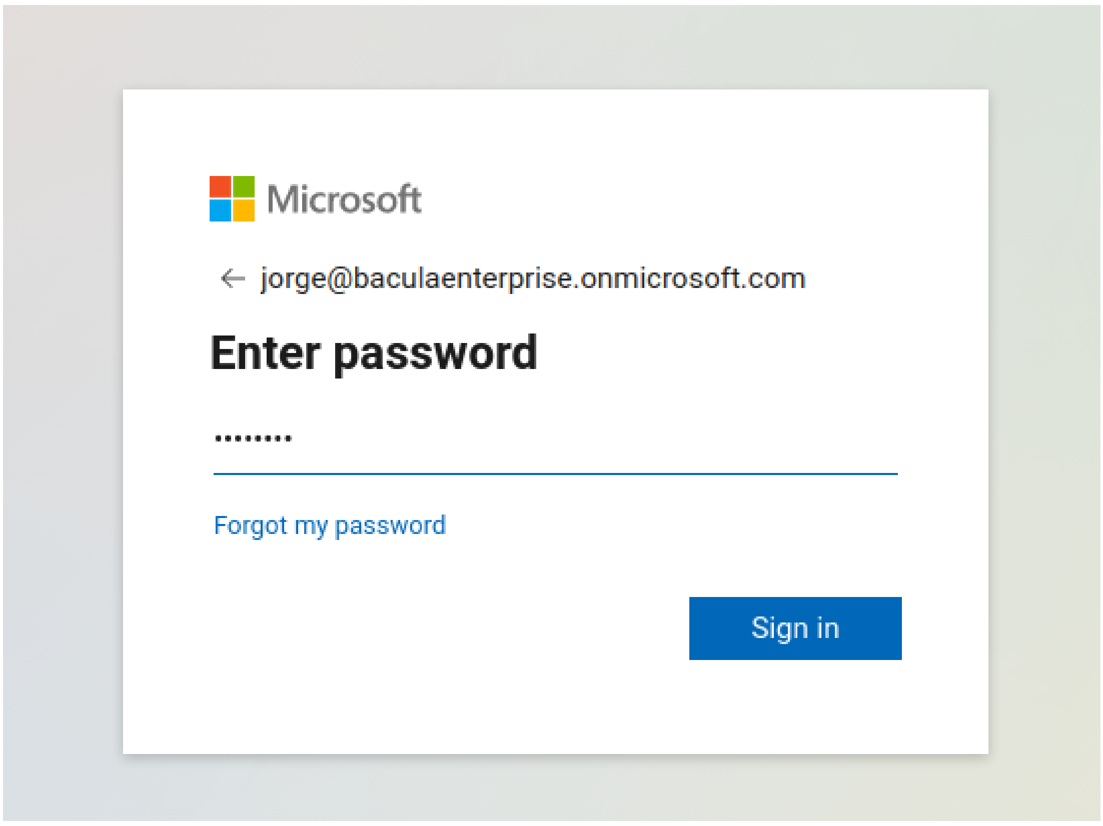
Here, if any MFA mechanism is enabled, it will be normally required and the user must complete the associated process. Finally Microsoft will ask to confirm if we want the plugin app to proceed with the operation, and we need to confirm:

Once the login is complete, the plugin will automatically end the operation and show something like:


Microsoft

Enter code

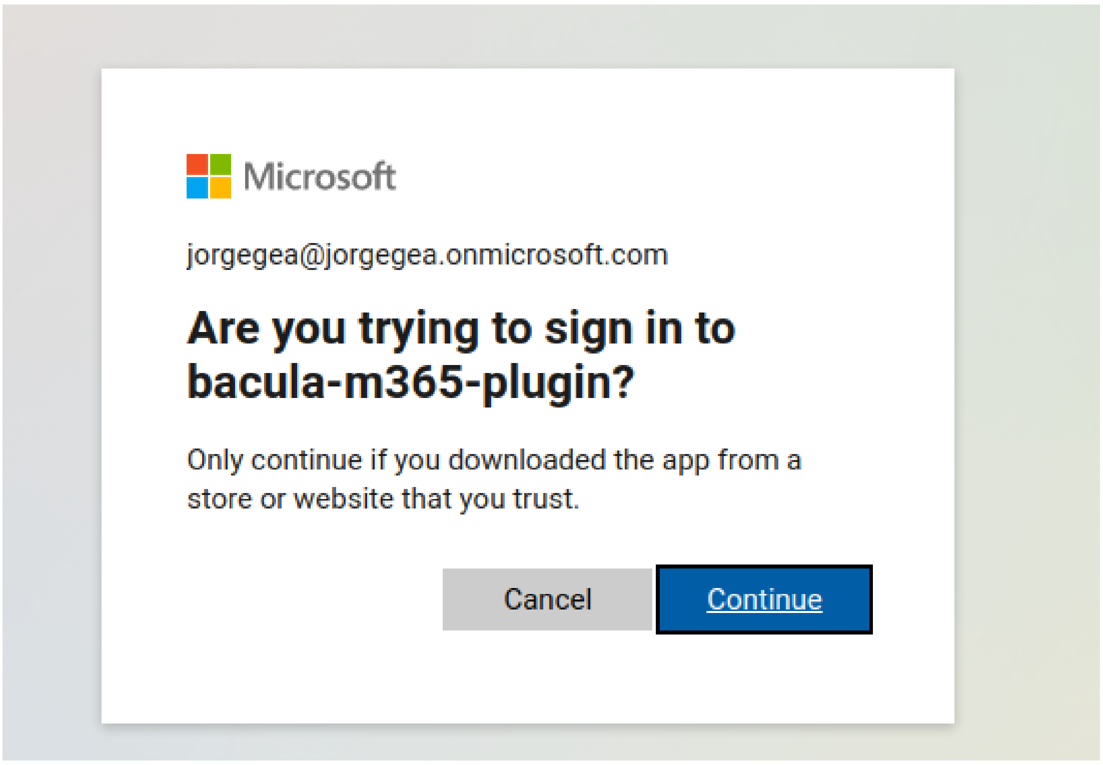
Enter the code displayed on your app or device.


Microsoft

jorge@baculaenterprise.onmicrosoft.com

Enter password

••••••••


Microsoft

jorgegea@jorgegea.onmicrosoft.com

Are you trying to sign in to bacula-m365-plugin?

Only continue if you downloaded the app from a store or website that you trust.

Listing 202: **query-login-2**

```
info=We have a valid token got from account: johndoe@johndoe.onmicrosoft.com
info=It will be automatically refreshed when needed. But the current␣
↪expiration date is: 2021-06-16T16:18:16Z
info=Backups using services with delegated permissions won't ask for␣
↪intervention
```

It will also store the token information locally in a special file that it is stored in the working directory inside a folder with the tenant name. For example: `/opt/bacula/working/m365/tennantname/[appid]-token_cache.json`

This file is also backed up in the backup operation itself, so it can be restored manually, in case it was lost and in case you don't want to run login operations again

In case the query method is not invoked prior to any job needing it, the job itself will show the message in the joblog and it will be blocked until a user performs the needed interaction described above. Once this happens, the job will automatically continue.

To verify the users currently logged into the system, we can use another query call using the `logged-users` parameter:

Listing 203: **query-logged-users**

```
.query plugin="m365: tenant=e421f055-748a-4a80-90da-bf48b66fa166␣
↪objectid=9017f87a-c164-488e-8f93-47139bcd6d47" client=127.0.0.1-fd␣
↪parameter=login:logged-users
info=Connected through: bacula-m365-plugin-standalone
user=AlexW@bacula.onmicrosoft.com
```

(continues on next page)

(continued from previous page)

```
user=jorge@bacula.onmicrosoft.com
```

Note: Starting with version 14.0 of Bacula Enterprise you can also see logged in users directly using BWeb, where you can also manually add logged-in users. In order to see more details, please go to section: BWeb Console<./m365-bweb-console>.

BWeb Management Console

Bacula Enterprise Microsoft 365 Plugin can be managed from a dedicated user friendly Web console, specifically designed to facilitate tasks as the authentication process for new tenants or the fileset configuration process.

Note: This feature is available starting with version 14.0 of Bacula Enterprise.

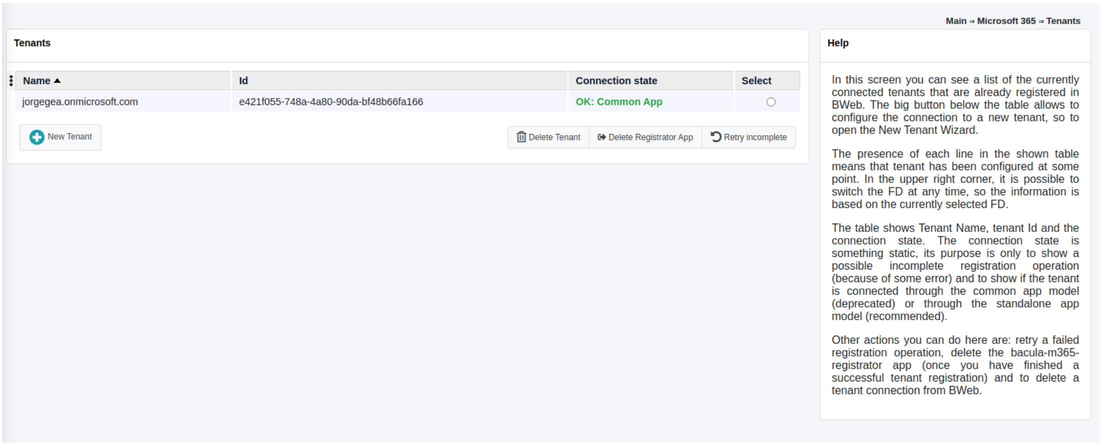
Menu options

The menu for M365 Plugin is composed by the following entries:

- Microsoft 365
  - Tenants
  - Users
  - Filesets

Connection

The purpose of this screen is to show a list of the currently connected tenants that are already registered. The presence of each line in the table means that a tenant has been configured at some point.



Name ▲	Id	Connection state	Select
jorgegea.onmicrosoft.com	e421f055-748a-4a80-90da-bf48b66fa166	OK: Common App	○

In this screen you can see a list of the currently connected tenants that are already registered in BWeb. The big button below the table allows to configure the connection to a new tenant, so to open the New Tenant Wizard.

The presence of each line in the shown table means that tenant has been configured at some point. In the upper right corner, it is possible to switch the FD at any time, so the information is based on the currently selected FD.

The table shows Tenant Name, tenant Id and the connection state. The connection state is something static, its purpose is only to show a possible incomplete registration operation (because of some error) and to show if the tenant is connected through the common app model (deprecated) or through the standalone app model (recommended).

Other actions you can do here are: retry a failed registration operation, delete the bacula-m365-registrator app (once you have finished a successful tenant registration) and to delete a tenant connection from BWeb.

In the upper right corner, it is possible to switch the FD at any time, so the information is based on the currently selected FD. After the first time a FD is selected, the information will be remembered using cookies for the next time and for any other M365 Console windows.

The table shows tenant name, tenant id and the connection state. The connection state can indicate a possible incomplete registration operation (because of some error), otherwise it will indicate a successful connection and the type of connection: through the common app or through the standalone app.

The big button placed at the bottom of the table allows us to configure the connection to a new tenant, so it will open the 'New tenant wizard' that is described in the section below.

Other actions that can be performed from this tenant list window are:

- Retry operation: It allows to retry a possible incomplete registration, so re-launch the proper internal query command (add-app or the objectid command) using a selected tenant.
- Delete registrator: It allows to delete the registrator app inside the selected tenant. It is not possible to delete a registrator if registration is incomplete . In this case, you must retry operation or delete tenant. This is a possibility only offered for security concerns, as the registrator app has some elevated permissions.
- Delete tenant: It removes the tenant from the internal BWeb configuration.

New tenant wizard

When opening new tenant wizard, we can select the registration model to be used: Standalone or Common.

Details of each model are shown in the images and you can also read more information in this whitepaper under the auth section.

It is needed to provide the tenant id, which can be found following the whitepaper instructions placed under the same auth section.

It is recommended to use the Standalone model (first one to appear in the web interface) and be using the common model only for testing purposes.

Once we proceed with one of the models, an URI of Microsoft website will be suggested and you need to provide the needed permissions, then, Microsoft will redirect you to a Bacula Systems page (same as shown in the auth section).

BWeb will automatically guide you until the end of the process where you will see your tenant correctly added into the tenant list screen.

Logged in users

In this screen a list of logged in users is shown.

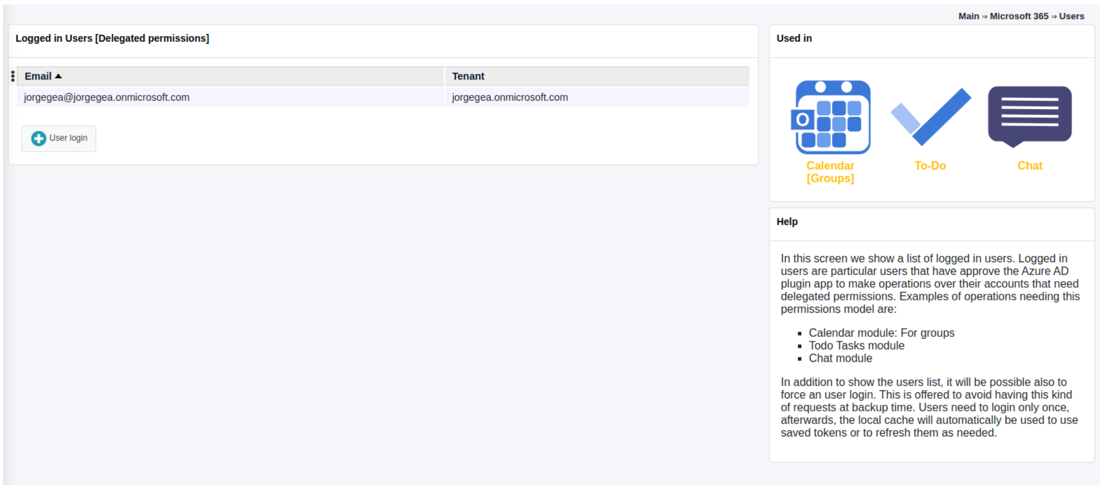
Logged in users are particular users that have approved the Azure AD plugin app to make operations over their accounts that need delegated permissions.

Examples of operations needing this permissions model are:

- Backup of calendar of groups
- Todo Tasks module
- Chats module

In addition to show the users list, it is also possible to force an user login. This function is offered to avoid having this kind of requests at backup time.

Users need to login only once, afterwards, the local cache will automatically be used to use saved tokens or to refresh them as needed.


Main → Microsoft 365 → Users
Logged in Users [Delegated permissions]
Used in
Calendar [Groups]
To-Do
Chat
User login

Email ▲	Tenant
jorgegea@jorgegea.onmicrosoft.com	jorgegea.onmicrosoft.com

Help

In this screen we show a list of logged in users. Logged in users are particular users that have approve the Azure AD plugin app to make operations over their accounts that need delegated permissions. Examples of operations needing this permissions model are:

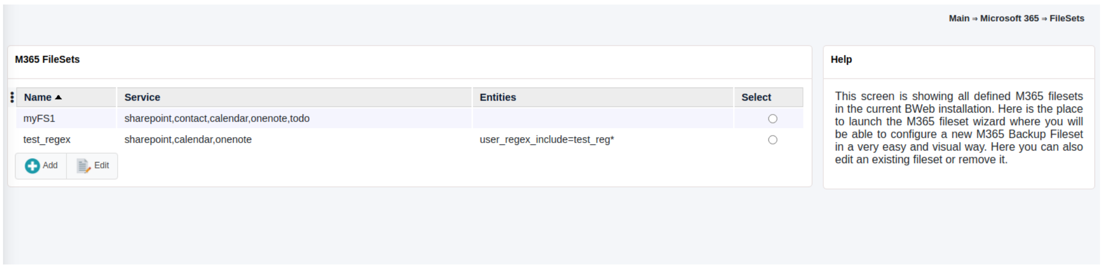
- Calendar module: For groups
- Todo Tasks module
- Chat module

In addition to show the users list, it will be possible also to force an user login. This is offered to avoid having this kind of requests at backup time. Users need to login only once, afterwards, the local cache will automatically be used to use saved tokens or to refresh them as needed.

For more information about delegated permissions, please take a look to <./m365-delegated-permissions>

Filesets

The purpose of the Filesets page is to control all the filesets of Microsoft 365 Plugin, so we will see them listed in the main screen:



Name ▲	Service	Entities	Select
myFS1	sharepoint,contact,calendar,onenote,todo		○
test_regex	sharepoint,calendar,onenote	user_regex_include=test_reg*	○

Help

This screen is showing all defined M365 filesets in the current BWeb installation. Here is the place to launch the M365 fileset wizard where you will be able to configure a new M365 Backup Fileset in a very easy and visual way. Here you can also edit an existing fileset or remove it.

From there it is possible to create a new M365 Fileset in an user friendly way:

We need to select the tenant to work with as a first step:

The second step is to select the entities to protect: they can be users, groups or sites. All the services of M365 are 'owned' by any of those entities:

It is possible to dinamically add or remove them, the wizard will store your final selection.

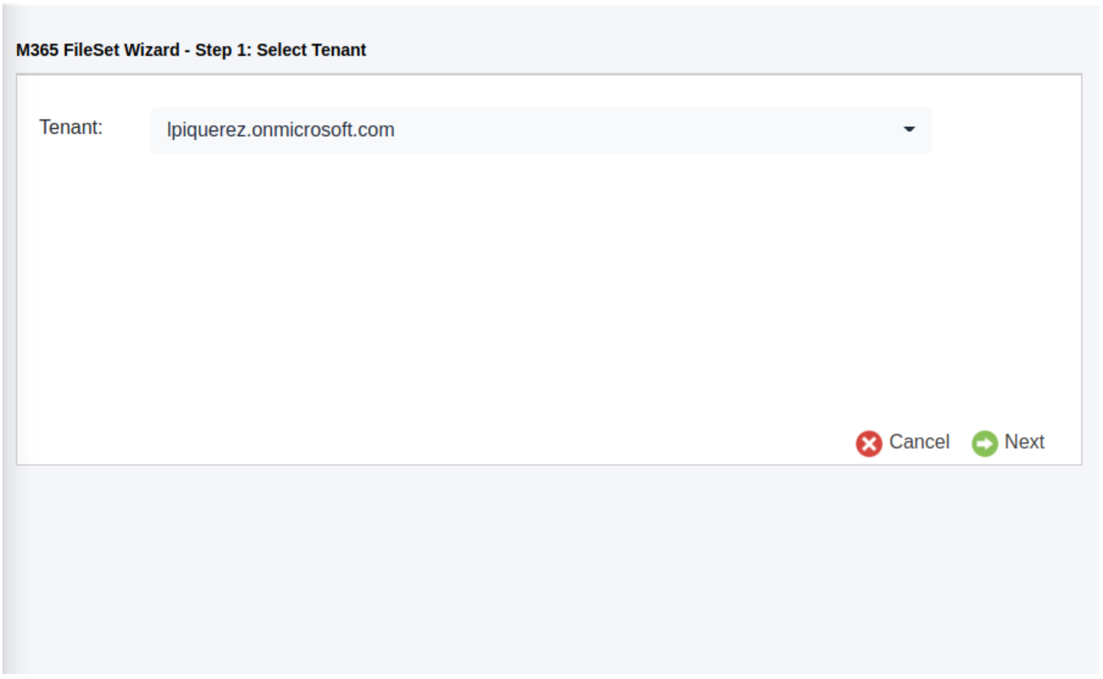
The third step allows us to select the services to protect in a similar way than the entities step, with multiselection allowed, and where we can remove or add services at will:

After selecting the services, we will have one independent screen for each selected service where we will enter the particular configuration for each of the services implied. In the screenshot below, we can see the Sharepoint services:

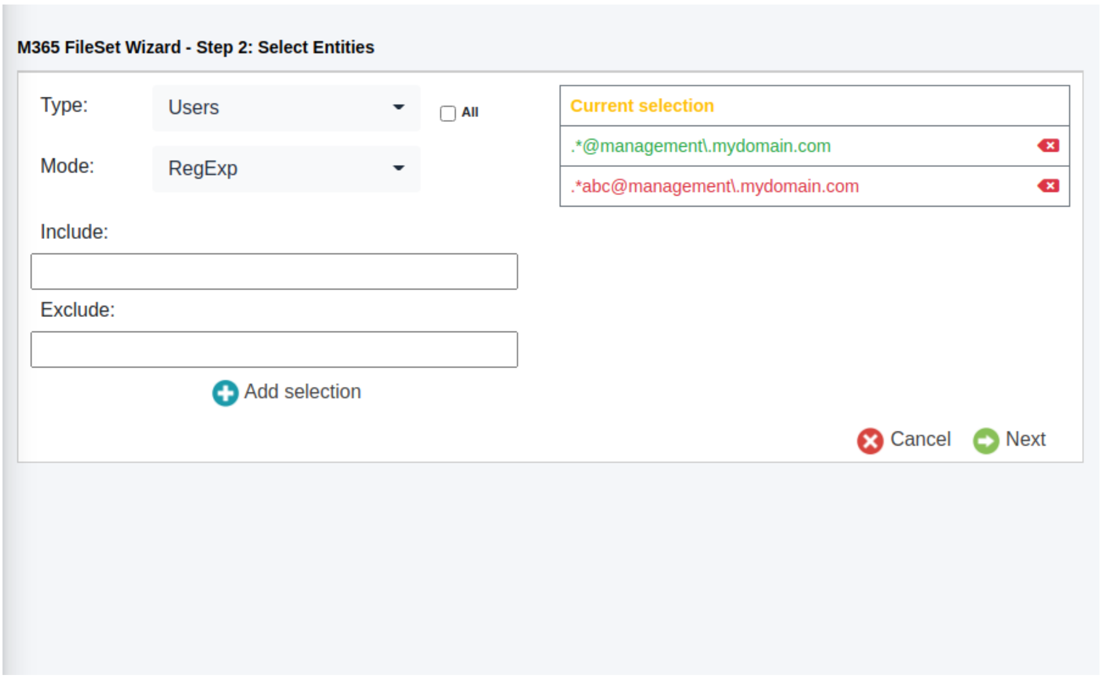
As a final step, it is possible to setup some advanced parameters and then finish the configuration:

From this final screen is possible to go directly to the job configuration wizard and finish a backup configuration or to return to the previous fileset list page.

Please, note that this Wizard contains the most common parameters and it is intended to help to configure the most common use cases. In case you need to setup more advanced parameters or to edit existing



Tenant: lpiquerez.onmicrosoft.com



Type: Users

All

Mode: RegExp

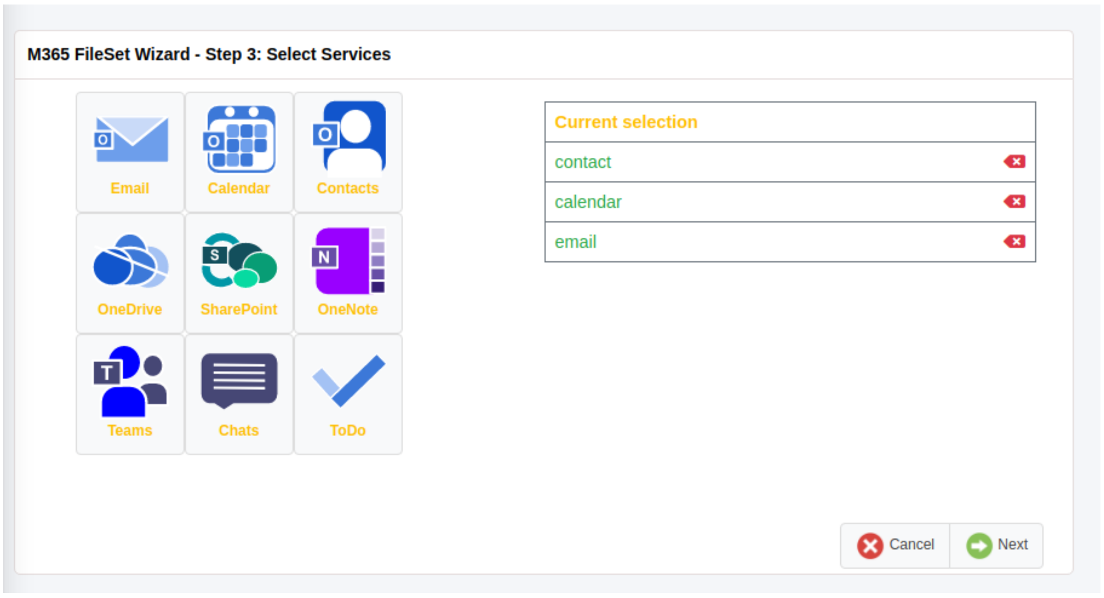
Include:

Exclude:

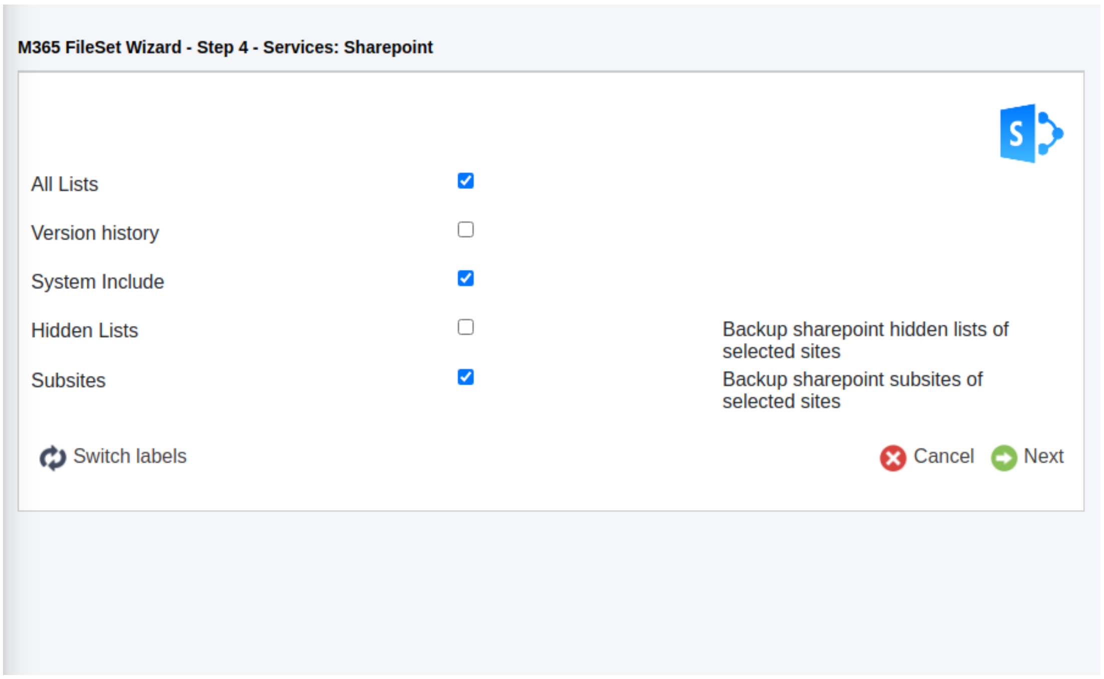
Current selection

.*@management\.mydomain.com

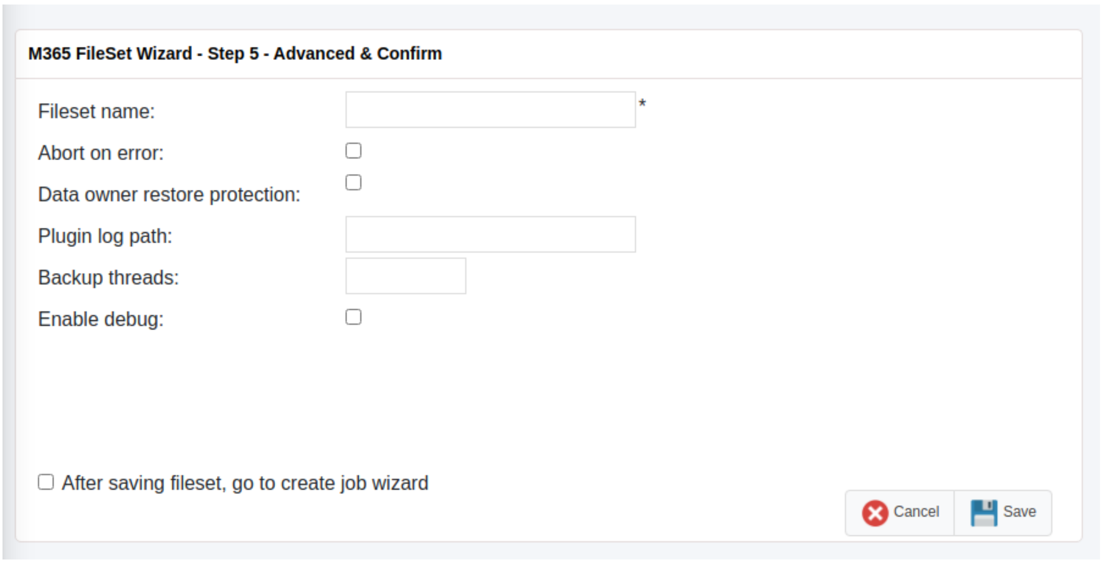
.*abc@management\.mydomain.com


M365 FileSet Wizard - Step 3: Select Services

Current selection
- contact
- calendar
- email


M365 FileSet Wizard - Step 4 - Services: Sharepoint

All Lists	✓	
Version history	☐	
System Include	✓	
Hidden Lists	☐	Backup sharepoint hidden lists of selected sites
Subsites	✓	Backup sharepoint subsites of selected sites


M365 FileSet Wizard - Step 5 - Advanced & Confirm

Fileset name:	*
Abort on error:	☐
Data owner restore protection:	☐
Plugin log path:	
Backup threads:	
Enable debug:	☐

☐ After saving fileset, go to create job wizard

filesets, BWeb will allow you to do it using the regular Fileset editor where you will find all plugin parameters.

Best Practices

Jobs Distribution

It is recommended, to split the target backup between different groups of entities or even having one job per entity (user, group, site, etc). This way errors in one job will not invalidate a whole backup cycle where some entities have been successful and some others had errors. This also makes easier to identify the cause of the error.

Concurrency

Microsoft public APIs impose a variety of boundaries that need to be considered. If a boundary is crossed, the corresponding API call will fail and the application will need to wait some amount of time to retry, which is different depending on the boundary crossed.

It is crucial to plan an adequate strategy to backup all the elements without reaching API boundaries. A single job implements some parallelism which can be reduced until a point, if necessary, using the variable **backup_queue_size** (default value is 30)**.** This variable controls the size of the internal queues communicating the internal threads, that are designed to fetch, open and send every item to Bacula core. Reducing its size will produce, ultimately (with a value of 1 for example), an execution very similar to a single threaded process. On the othere hand the plugin has **concurrent_threads** which controls the number of simultaneous processes fetching and downloading data (default value is 5).

If you are going to launch different jobs in parallel it is recommended to configure different services for each of them (protect in parallel email, drive, contacts…). However, be careful with the concurrency over the same service (in general, it is recommended a maximum of 4-5 jobs working with the same service) and plan a step-by-step testing scenario before putting it into production. Other important point is the timing schedule, as some boundaries are related to time-frames (number of request per 10 minutes or 1 hour, for example). If you detect you reach boundaries when running all your backups during a single day of the week, please try to use 2 or 3 days and spread the load through them in order to achieve better performance results.

More information about Microsoft 365 Graph API boundaries may be found here:

https://docs.microsoft.com/graph/throttling

Disk Space

It is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs or through the same job (by default this is the case through the concurrent_threads=5 parameter), you would need at least that size for the largest file multiplied by the number of operations in parallel you run.

Read more details in the Backup of Attachments and Files section.

Performance

The performance of this plugin is highly dependent on many external factors:

- ISP latency and bandwidth
- Network infrastructure
- FD Host hardware
- FD Load
- ...

In summary, it is not possible to establish an exact reference about how much time a backup will need to complete.

As reference values, in testing environments and using **sequential calls** we had the following results:

- 3000 emails in the same folder of a single mailbox, with emails ranging from 500 to 3000 words - about 10K to 30K in size
- The total time required to back them up was: 88243 ms.
- That implies a total time per email of about **30ms**

Concurrency has also been tested. For example, when using single-threaded mode (concurrent_threads=1) with emails, good values where found for 4-6 concurrent jobs that allowed us to backup around 200 emails per second (sizes between 20K-30K) when no attachments were implied. As more attachments were implied, the number of emails will decrease very significantly, while the overall size speed will be increased. This can be generalized:

- Many little objects to protect -> More objects per second, but less speed (MB/s)
- Big files to protect -> Less objects per second, but greater speed (MB/s)

It is recommended to benchmark your own environment in base to your requirements and needs.

The automatic parallelization mechanism (using concurrent_threads=x, default is 5) should work well for most scenarios, however, fine tune is possible if we define one job per entity and we control how many of them run in pararllel, together to decrease the concurrent_threads value in order to avoid throttling from MS Graph API.

There are many different possible strategies to use this plugin, so please, study what is best suiting for your needs before deploying the jobs for your entire environment, so you can get best possible results:

- You can have a job per entity (users, groups, sites. . .) and all services
- You can have inside a job multiple entities and only some services
- You can split your workload through an schedule, or try to run all your jobs together.
- You can run jobs in parallel or take advantadge of concurrent_threads and so run less jobs in parallel
- You can select what services to backup or backup them all
- You can backup whole services to backup or select precisely what elements you really need inside each service (folders, paths, exclussions. . .)
- For delegated permissions services, you can proceed with the login previously to backups (recommended) or you can wait for the job to ask for what it needs
- etc.

Best Practices by Service

Each Microsoft 365 service presents its own particularties. Moreover each deployment and each environment can be protected through different strategies depending on the size, number of entities and other parameters. However, there are some general considerations that can be helpful to consider for a selection of services:

Sharepoint

Sharepoint Online is the service where the target data varies the most among the M365 services. Sites can be structured very differently and there exists many different kinds of Sharepoint lists. In addition to that, some of them could have very large document libraries, while others could be very sparse.

For these reasons, it is recommended to split Sharepoint jobs to be as atomic as possible.

Ideally jobs should be split up to a single site backup job. This will help to monitor the health of the backups, to keep good backups for all sites presenting no issues and but to quickly troubleshoot any site presenting any kind of inconvenience. Usually, this is especially needed when Sharepoint service are intensively used.

If sites are created and removed often, another option to avoid a frequent configuration change is to employ the site_regex_include plugin variable and use the naming patterns to set up the jobs, so they will also include future created sites (possible examples: .*Team, .*Web, Dev.* …).

In order to facilitate the job split, it is possible to use the listing or query commands described in list-commands to have the site names list.

Another best practice is to split Sharepoint sites into two parts with two different jobs: - Pure Sharepoint job: Use sharepoint service (service=sharepoint site=SiteName) and the parameter sharepoint_include_drive_items=no. - Drive part job: Use drive service (service=drive site=SiteName)

Sharepoint jobs, when using the associated powershell commands, cannot be run in parallel. One will wait for any other session to finish, before executing the corresponding commands. Using the above strategy makes Sharepoint jobs to be much more quick and light, so it's a lot easier to control such concurrency. Other ways to control the concurrency in Bacula Enterprise are:

- Designing separated time windows through scheduling

- Sending all Sharepoint jobs to the same pool and the associated storage resource (both of them dedicated to Sharepoint backups) and limiting ConcurrentJobs to the desired value in that particular storage
- Having 1 specific client resource for Sharepoint and use ConcurrentJobs parameter there.

In general, it's not needed to enable system lists, hidden lists or even the versioning of the elements. All those options will impact the performance of the jobs directly, so you should plan carefully the usage of such parameters.

Some Sharepoint sites present an issue with the hashcheck of the files contained in some of its document libraries. If the hash is not correct at M365 side, it does not match the one that is calculated by the plugin locally, while the file is correct. This situation causes a lot of error messages around the hash and also many extra calls to the service to re-check this hash. When this situation happens, it is recommended to disable the hashcheck mechanism for the affected site using the parameter 'drive_disable_hashcheck=true'.

Onenote

Onenote is the most sensitive module to throttling limits in Microsoft 365 APIs: https://learn.microsoft.com/en-us/graph/throttling

It is recommended to be selective with the information to be protected and make one job for each selected entity: that is one job per user, group or site.

In case throttling problems appears with this strategy, jobs will need to be spread into larger time windows.

It is not recommended to include onenote for entity focused strategies where there is one job per user including all services (email, drive, contact, calendar ..). It is recomended to have specific onenote jobs.

Chats and Todo

This service uses delegated permissions. They can be protected together, but we recomend to run the 'login' query command for each individual user before running the actual jobs.

Email

Email service allows you to protect not only email and attachments, but also settings, user categories and an additional MIME copy for each email with its attachments. MIME files are very useful if you are planning to migrate your data from M365 to any other Email tool, however, the cost of getting each MIME file is very high and it duplicates backup time. MIME files are not needed to perform a local restore or a restore over M365. Therefore, it is recommended to disable MIME backup for a general basis and use it only in case of planning a complete migration of the data.

Similarly to other services, it is recommended to split the backup set into small sets of users (ideally one backup per mailbox, but it's also fine to group them by 5 or 10), specially if there is a large number of mailboxes and they contain thousands of emails. To make this easier, the plugin offers regular expression selection parameters (user_regex_include), as well as listing and query commands to analyze the backup target (list-commands).

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

Listed in this section are some scenarios that are known to cause issues.

Pwsh: The specified program requires a newer version of windows

If a Sharepoint backup shows the message referenced in the title, it means the powershell session of the job has conflicted with another open Powershell session.

Solution: Be sure no other Powershell sessions are open and try again

M365 DataPack Onedrive Hashes

If you test the plugin with the developer Datapack of M365 you may find problems with hash checking of OneDrive autogenerated files. You could see hash failures.

The reason is those auto-generated files have incorrect hashes.

Solution: Use your own uploaded files to test the plugin.

Sharepoint PnP Template restore problems

As stated in section sp-module-limit, templates of Sharepoint can present different problems when they are applied over new generated sites. To troubleshoot these issues, the following steps can be taken:

- Restore the site template locally
- Detect the exact part of the template that is causing problems and remove that section from the xml
- Manually apply the site template using PnP.Powershell

Suppose we are experiencing this problem and we have already restored our template in **/tmp/mytemplate.xml** and it is ready to be tested. Below we show how to run manually PnP actions:

1. The first thing to do is run Powershell. Just run the following command in your terminal:

Listing 204: **pwsh**

```
pwsh
```

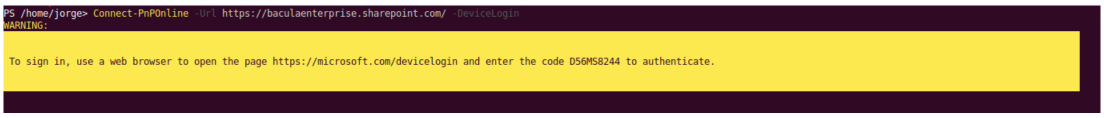
2. Import the Pnp Module:

Listing 205: **pwsh**

```
Import-Module -Name PnP.PowerShell
```

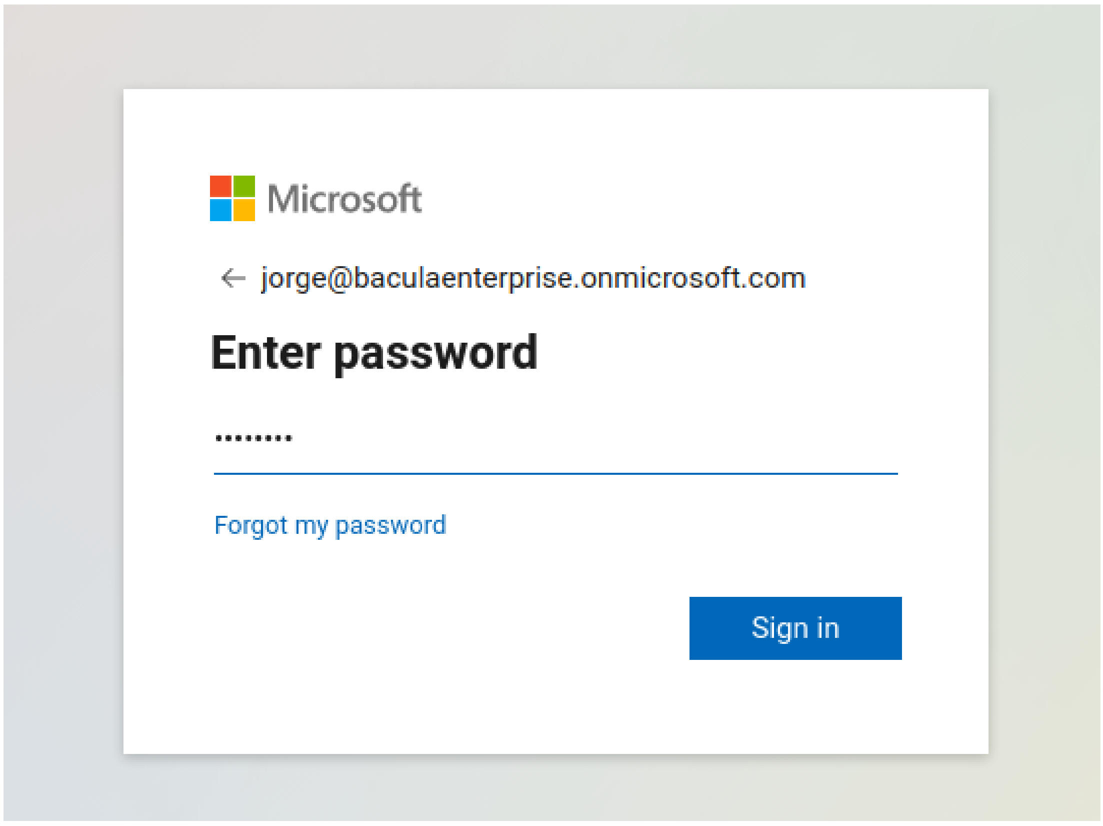
3. Connect to your main site, which is https://tenantname.sharepoint.com

   a. We will use the DeviceLogin method, as this method allows Multifactor Authentication. Just run the command shown with your main site:



```
PS /home/jorge> Connect-PnPOnline -Url https://baculaenterprise.sharepoint.com/ -DeviceLogin
WARNING:

 To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code D56MS8244 to authenticate.
```

   b. Follow the instructions and go to a browser pointing to the provided URL. You will be asked for credentials and you need to login with a tenant administrator user:



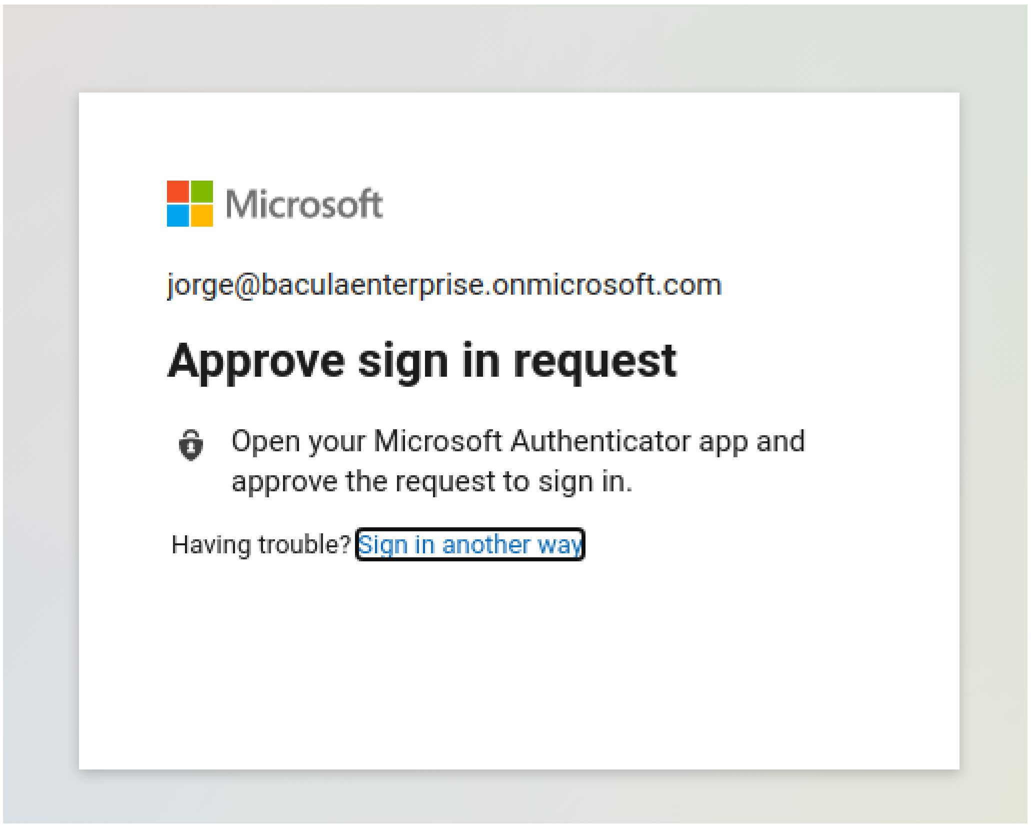
   c. If MFA is enabled, complete the process:

   d. Then put the code shown in the terminal in the first step and click Next (in this example code would be *D56MS8244*):

   e. Approve the required permissions for PnP.Powershell:

   f. Get back to the terminal and check that you are connected:

   g. You can check the correct result of the command Get-PnPSite:

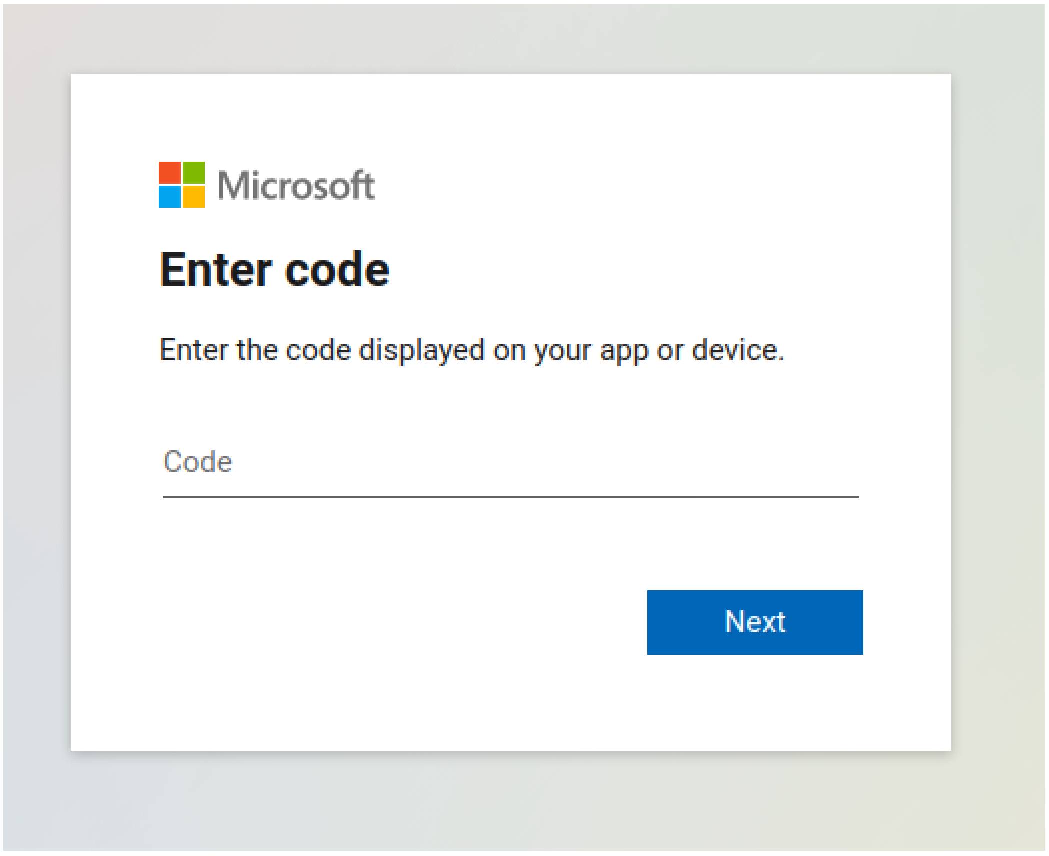

Microsoft

jorge@baculaenterprise.onmicrosoft.com

Approve sign in request

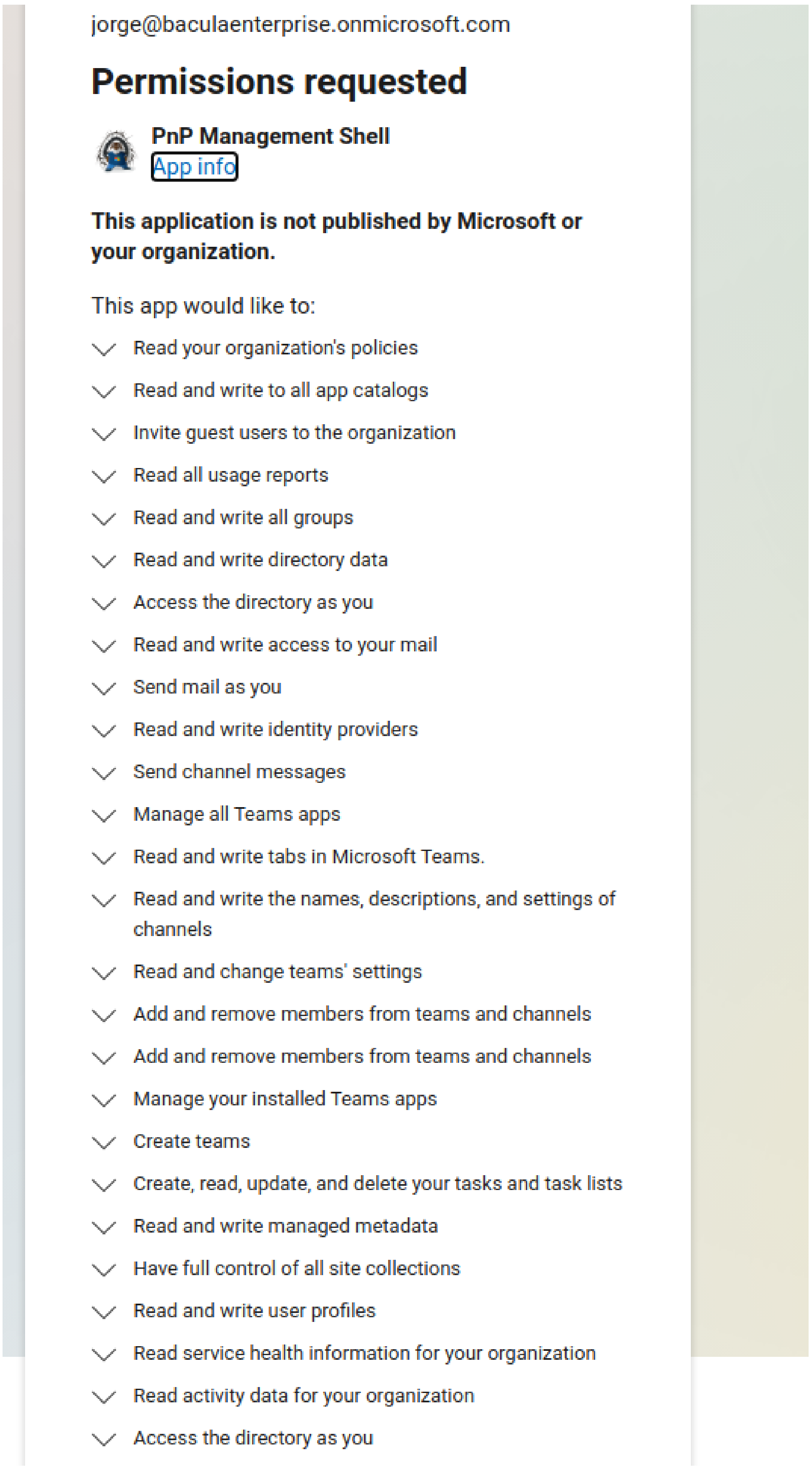
Open your Microsoft Authenticator app and approve the request to sign in.

Having trouble? Sign in another way


Microsoft

Enter code

Enter the code displayed on your app or device.


jorge@baculaenterprise.onmicrosoft.com

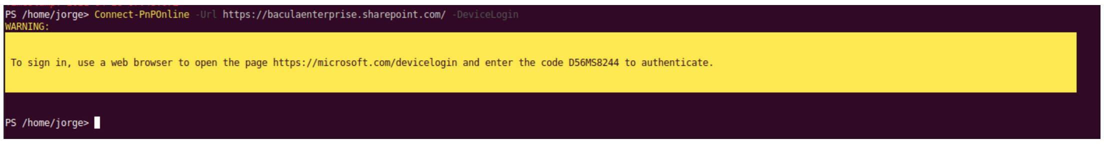
Permissions requested

PnP Management Shell
App info

This application is not published by Microsoft or your organization.

This app would like to:

- Read your organization's policies
- Read and write to all app catalogs
- Invite guest users to the organization
- Read all usage reports
- Read and write all groups
- Read and write directory data
- Access the directory as you
- Read and write access to your mail
- Send mail as you
- Read and write identity providers
- Send channel messages
- Manage all Teams apps
- Read and write tabs in Microsoft Teams.
- Read and write the names, descriptions, and settings of channels
- Read and change teams' settings
- Add and remove members from teams and channels
- Add and remove members from teams and channels
- Manage your installed Teams apps
- Create teams
- Create, read, update, and delete your tasks and task lists
- Read and write managed metadata
- Have full control of all site collections
- Read and write user profiles
- Read service health information for your organization
- Read activity data for your organization
- Access the directory as you



```
PS /home/jorge> Connect-PnPOnline -Url https://baculaenterprise.sharepoint.com/ -DeviceLogin
WARNING:

 To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code D56MS8244 to authenticate.


PS /home/jorge> 
```

Listing 206: **pwsh**

```
Get-PnPSite
Url CompatibilityLevel
--- ------------------            https://baculaenterprise.
↪sharepoint.com 15
```

4. Create your new site using the following commands. Depending on the site kind, you will need to utilize one command or another.

   a. Site kind is determined by @BaseSiteTemplate value which you can find in your template

   b. Replace $var with appropriate values:

      1. newSiteName → NonExistingSiteName in your tenant
      2. newSiteUrl → https://tenantname.sharepoint.com/sites/newSiteName
      3. newSiteOwner → anyadminuser@yourtenant.com
      4. newSiteTimeZone → RegionalSettings > TimeZone attribute value from your template
      5. newSiteLocale → RegionalSettings > LocaleId attribute value from your template
      6. newSiteKind → Commented BaseSiteTemplate attribute value from your template

   c. TeamSite

Listing 207: **pwsh**

```
New-PnPSite -Type TeamSite -Title $newSiteName -Alias $newSiteName -
↪Description "-" -Owners
$newSiteOwner -Wait
```

   d. CommunicationSite

Listing 208: **pwsh**

```
New-PnPSite -Type CommunicationSite -Title $newSiteName -Url $newSiteUrl␣
↪-Owner $newSiteOwner
```

   e. Other:

Listing 209: **pwsh**

```
New-PnPTenantSite -Title $newSiteName -Url $newSiteUrl -Owner
↪$newSiteOwner -TimeZone
$newSiteTimeZone -Lcid $newSiteLocale -Template $newSiteKind
```

5. Now you need to connect to your newly created site. Therefore, repeat step 1, but using your new site URL: https://tenantname.sharepoint.com/sites/newSiteName

6. Once connected to your target tenant, apply the template with these 2 commands:

Listing 210: **pwsh**

```
Set-PnPSite -NoScriptSite:$false
Invoke-PnPSiteTemplate -Path /tmp/mytemplate.xml
```

7. Once the process ends, if everything was OK, you can disconnect with:

Listing 211: **pwsh**

```
Disconnect-PnPOnline
exit
```

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (you will find them in the m365-debug.err file),
you are very likely dealing with sub-attachments containing big files or you are using a high level of concurrency through internal concurrent_threads parameter and/or parallel jobs. To overcome this situation you can:

a) Disable the backup of multi-level attachments setting either email_multilevel_attach=no, or calendar_multilevel_attach=no, depending on the service configured to back up.

b) Reduce concurrent_threads parameter

c) Reduce the number of jobs running in parallel

d) If you cannot do that or you are not using multi-level attachments you should increase JVM memory.

To increase JVM memory that you will need to:

Create a this file: '/opt/bacula/etc/m365_backend.conf'.

Below, an example of the contents:

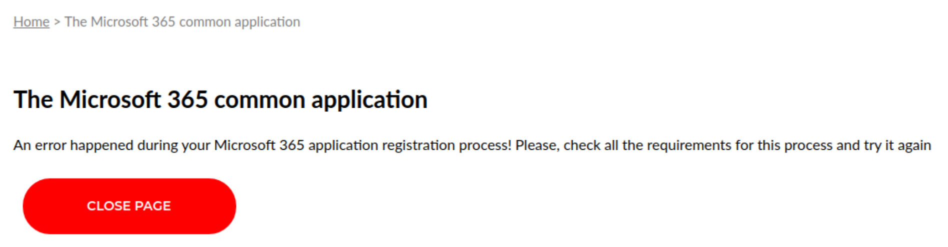
|M365_JVM_MIN=2G |M365_JVM_MAX=8G

Those values will define the MIN (M365_JVM_MIN) and MAX (M365_JVM_MAX) memory values assigned to the JVM Heap size. In this example we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough. Please, be careful if you are running jobs in parallel, as very big values and several jobs at a time could quickly eat all the memory of your host.

The '/opt/bacula/etc/m365_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

App Registration Error

When trying to register the bacula-m365-plugin app in a given tenant, the following error can appear in the URL of a resulting page placed on baculasystems.com:



The Microsoft 365 common application

An error happened during your Microsoft 365 application registration process! Please, check all the requirements for this process and try it again

Listing 212: **Registration error**

```
https://www.baculasystems.com/m365-plugin-auth/common?error=access_denied&
↪error_description=AADSTS650052%
3a+The+app+needs+access+to+a+service+(%27https%3a%2f%2f*.dps.mil%2f
↪%27)+that+your+organization+%27cd77bbc0-999e-
44c5-af9d-c11bea178407%27+has+not+subscribed+to+or+enabled.
+Contact+your+IT+Admin+to+review+the+configuration+of+your+service+subscriptions.
↪%0d%0aTrace+ID%3a+afdc0b36-
bad6-4d1f-8d24-9213439c0a01%0d%0aCorrelation+ID%3a+c73a3631-5573-41ee-8220-
↪cc281ea4882f%0d%0aTimestamp%3a+2021-
04-27+09%3a21%3a55Z&error_uri=https%3a%2f%2flogin.microsoftonline.com%2ferror
↪%3fcode%
3d650052&admin_consent=True&state=12345
```

If we look carefully into the error contents we will notice the following message:

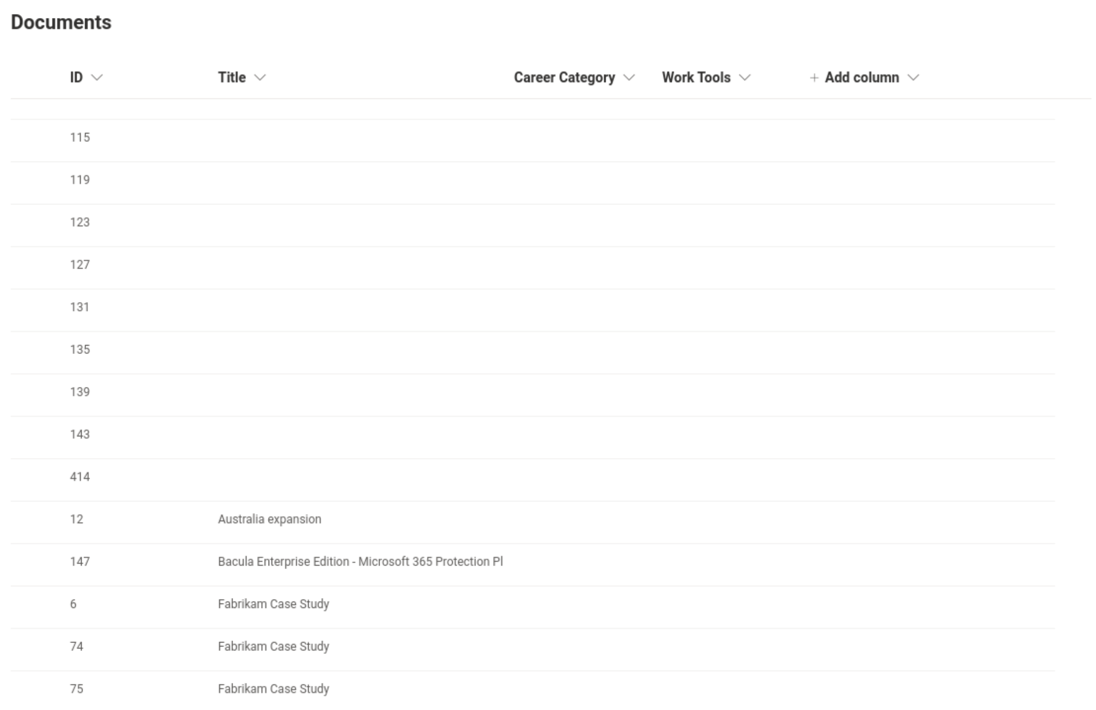
- *AADSTS650052*: The app needs access to a service (https://*.dps.mil.com) that your organization cd77bbc0-999e-44c5-af9d-c11bea178407 has not subscribed to or enabled. Contact your IT Admin to review the configuration of your service subscriptions.

Even if it is not a very clear message, the error is pointing us to the fact that our tenant is missing some kind of access or subscription in order to utilize bacula-m365-plugin app. The contents of the error can vary depending on the specific situation of the target tenant.

However, in general, and particularly for the example shown, the problem comes from the fact that **target tenant has no license to utilize Sharepoint Online service**. Without this license/permission, the plugin cannot be registered.

Empty Items in Group/Team Site Libraries

When working with the OneDrive module and Group entities you may experience a view of a list of apparently empty elements if navigating through them using Sharepoint views:


Documents

ID	Title	Career Category	Work Tools	+ Add column
115				
119				
123				
127				
131				
135				
139				
143				
414				
12	Australia expansion			
147	Bacula Enterprise Edition - Microsoft 365 Protection Pl			
6	Fabrikam Case Study			
74	Fabrikam Case Study			
75	Fabrikam Case Study			

This is not an error of this plugin. This is a problem with some kind of views available in the Sharepoint Online service, where list items corresponding to drive items are shown. As here we are dealing with drive items, those list items have no title attached.

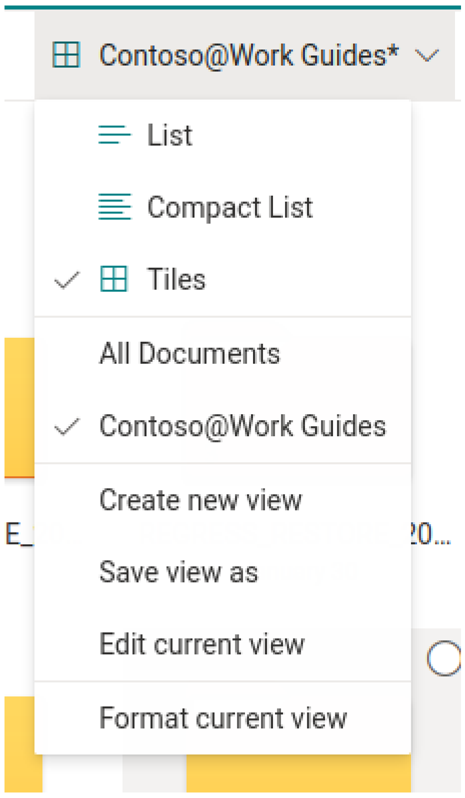
In the particular example shown above, a 'list view' of a document library is selected. However if we select any other view as 'Titles', 'All Documents', etc:

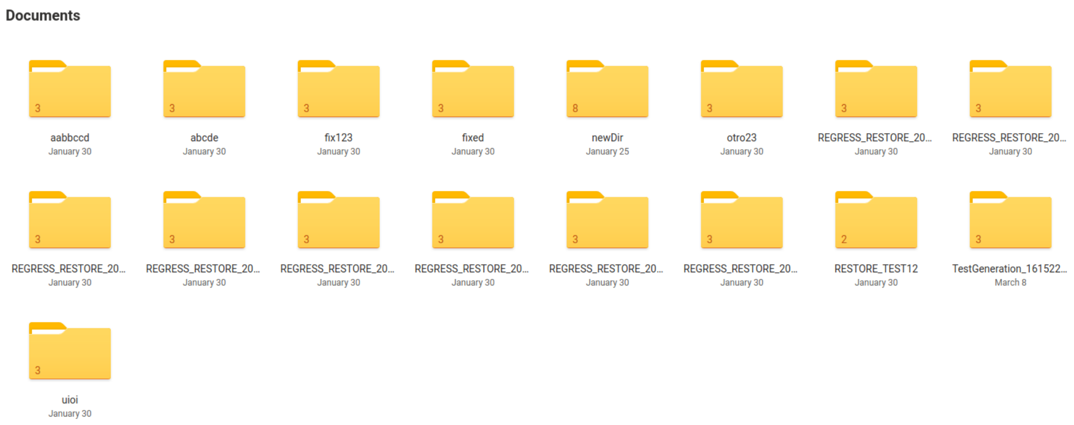
We will be able to see folders and files with their corresponding names:

Empty Chat messages after restoring Teams

After a successful restore operation with the Teams module, it can happen that accessing the data using the web interface of the M365 services will show the associated channels with empty contents. This is usually associated to some bad functions of implied cookies with the web browsers.

If you experience this situation we recommend to open the information using a 'private window/tab' or removing previous cookies, as the information is actually there. In case this strategy does not work, please use the Teams 'desktop app' for your OS, where the behavior is much more solid.


Contoso@Work Guides*
List
Compact List
Tiles
All Documents
Contoso@Work Guides
Create new view
Save view as
Edit current view
Format current view


Documents
aabbccd
January 30
abcde
January 30
fix123
January 30
fixed
January 30
newDir
January 25
otro23
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
REGRESS_RESTORE_20...
January 30
RESTORE_TEST12
January 30
TestGeneration_161522...
March 8
uioi
January 30

Chat messages with not displayed contents in Teams

There are several kind of auto-generated messages that rely on other Microsoft Cloud internal data structure. Examples of this are:

- Scheduled meetings through chat actions, not associated to calendar (they belong to 'onlineMeetings' module, which is not included today in Teams backup)
- Messages showing that a Tab was added
- Messages showing contents of external apps

This kind of messages will be restored exactly with the same form that they were got during backup and text contents will be visible. However, as the information linked is attached in the external applications to the old team, that visual content may not be visible as it was in the original message.

Error restoring teams app

Teams apps are external entities controlled by an undetermined number of external provides. When adding an app to a team, a process belonging to that provider takes place. That process is invoked through the plugin restore process when restoring teams with apps. Sometimes that external process can return unexpected results.

The M365 Plugin will show those situations with a propper message in the joblog that is not treated as error, and the restore will succeed in most cases.

As this is not managed by Bacula Systems or MS Graph API, the approach is just to show the result as it is, when it is not successful. However, please note that the app itself,
in most cases, will be also successfully restore even if that situation happens.

Error with Teams App after restoring a Planner plan

Teams desktop applications use cache data that can interfere sometimes with the real data. Restoring a planner plan can be an example. When you try to open a restored planner tab you may see an error message. However, please just click on the reload button and you should see your data without issue.

Orphaned Java Processes After Job Failure

In some circumstances, java processes belonging to the plugin can be left behind. They may be using noticeable CPU and memory resources. It is safe to remove them by sending an appropriate signal.

Such orphaned processes should not occur with Bacula Jobs finishing without problems.

A way to identify such processes would be to look for java processes running the plugin .jar file either when no such job is actually active, or when no such job of a similar age exists. For example

Listing 213: **Identify Old Java Processes**

```
[root@bsys-demo-m365] ~ # ps -o pid,stime,etime,args -wwC java | grep -E
↪'bacula-m365-plugin-.*'
 370270 Jul04 50-04:48:36 java -Xms512M -Xmx4G -jar /opt/bacula/lib/bacula-
↪m365-plugin-4.4.9.jar --mode=bacula --path=/opt/bacula/working --log=/opt/
↪bacula/working/m365/m365-debug.log
 370403 Jul04 50-04:42:10 java -Xms512M -Xmx4G -jar /opt/bacula/lib/bacula-
```

(continues on next page)

(continued from previous page)

```
↪m365-plugin-4.4.9.jar --mode=bacula --path=/opt/bacula/working --log=/opt/
↪bacula/working/m365/m365-debug.log
```

If no m365 job is running since July 04, it would be safe to *kill -9 370270 370403*.

In newer versions of the plugin, the actual Bacula Job running one of the java processes will be indicated in a command line option of the java process, and the Job report will indicate the process id of the java process started.

Google Workspace Plugin

This white-paper presents how to protect the most relevant elements of Google Workspace services using **Bacula Enterprise**.

Overview

This white-paper presents how to protect the most relevant elements of Google Workspace services using **Bacula Enterprise**.

Requirements

Bacula Google Workspace Plugin supports free Gmail accounts and Workspace accounts.

In order to protect Workspace accounts it is needed to have a Google Workspace active subscription: https://workspace.google.com/intl/es-419/pricing.html

On the other hand, it is necessary to have **full administrative access to the target associated Organization** to protect in order to generate a Google Application with all the needed permissions that will be used to communicate with this plugin.

In order to protect free accounts it is just needed to prepare some configurations in Google Cloud Platform, logging in with the user to protect, before using the plugin. Please refer to the authentication section of this document to have further details.

Currently, the plugin must be installed on a Linux based OS (RH, Debian, Ubuntu, SLES ..) where a Bacula Enterprise File Daemon is installed. *Bacula Systems may address support for running this plugin on a Windows platform in a future version.*

The OS where the File Daemon is installed must have installed Java version 11 or above.

Memory and computation requirements completely depend on the usage of this plugin (concurrency, environment size, etc). However, it is expected to have a minimum of 4GB RAM in the server where the File Daemon is running. By default, every job could end up using up to 512Mb of RAM in demanding scenarios (usually it will be less). However, there can be particular situations where this could be higher. This memory limit can be adjusted internally (see gw-out-of-memory). Refer to the Scope section below for any service specific requirements.

Why protecting Google Workspace?

This is a common question that arises frequently among IT and Backup professionals when it comes to SaaS or Cloud services, so it is important to clearly understand it.

It is a fact that Google or any cloud provider offers some capabilities intended to prevent data loss such us:

- Usually, all data stored in cloud services is geo-replicated using the underlying cloud infrastructure to have the information stored into several destinations automatically and transparently. Therefore, complete data loss because of hardware failures are very unlikely to happen.
- Google Data Loss Prevention service: This is a policy based service capable of detecting filtered content and act upon it encrypting it or modifying it in order to protect it (remove headers, etc). This is not a backup tool, it is a service to prevent undesired actions to the content stored in Google Workspace (for example sharing confidential information with the wrong people).
- Retention policies of Google Workspace: Google retains a maximum of 30 days of deleted information from active subscriptions. Therefore it is possible to recover accidental deleted items inside that period.

There is no other data protection mechanism. Below we show a list of challenges that are not covered by cloud services:

- No Ransomware protection: If data suffers an attack and becomes encrypted, data is lost.
- No malicious attacker protection: If data is deleted permanently, data is lost.
- No real point-in-time recovery, and recoveries of partially deleted files are limited to 30 days.
- It is not possible to align data protection of Google Workspace services to general retention periods or policies longer than 30 days.
- No automated way to extract any data from the cloud to save it in external places (this could lead to eventual compliance problems)

Scope

Bacula Enterprise Google Workspace Plugin is applicable on environments using any Workspace subscription.

This paper presents solutions for **Bacula Enterprise** version 14.1 and later, and is not applicable to prior versions.

Note: Important considerations

Before using this plugin, please carefully read the elements discussed in this section.

Empty Files

In general, empty files (files with 0 byte contents) are simply not backed up by Google Workspace plugin. In particular, Google Drive files will show a message in the joblog to inform about empty files detected and so not processed.

Files and Objects Spooling

In general, this plugin backups two types of information:

- Objects
- Files

Objects are elements representing some entity in Google Workspace such as a files metadata.

While objects are directly streamed from memory to the backup engine, files need to be downloaded to the FD host before being sent. This is done in order to make some checks and to improve overall performance, as this way operations can be paralleled. Every file is removed just after being completely downloaded and sent to the backup engine.

The path used for this purpose is established by the ‘path’ plugin variable, that usually is set up in the gw_backend script with the value: /opt/bacula/working

Inside the path variable, a ‘spool’ directory will be created and used for those temporary download processes.

Therefore, it is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs or through the same job (by default this is the case through the concurrent_threads=5 parameter), you would need at least that size for the largest file multiplied by the number of operations in parallel you run.

For emails it is important to note that download operations are done in one step because of some API requirements. This means the jvm should have enough memory to load those downloaded files inside RAM. In case you suffer any memory issue, please refer to the troubleshooting section to find out how to increase it.

Accurate Mode and Virtual Full Backups

Accurate mode is not supported. The feature will be addressed in future versions of this plugin.

Virtual Full backups are supported.

Google Workspace APIs General Disclaimer

Google Workspace APIs are owned by Google and they can change or evolve at any time. Almost all service APIs are actively developed, containing new features every week, even if the version number of the service is not changed as a result of any of those additions. Just as an example, Google Drive API now is tagged as v3 (and this plugin is using that version to work).

This situation is significantly different from traditional on-premise software, where each update is clearly numbered and controlled for a given server, so applications consuming that software, can clearly state what is offered and what are the target supported versions.

Google is committed to try not to break any existing functionality that could affect external applications. However, this situation can actually happen and therefore, cause some occasional problems with this plugin. Bacula Systems controls this with an advanced automatic monitoring system which is always checking the correct behavior of existing features, and will react quickly to that hypothetical event, but please be aware of the nature and implications of this kind of cloud technologies.

Features

The **Bacula Enterprise Google Workspace Plugin** is a very easy to deploy and configure plugin supporting the following services:

- Google Drive
- Google Mail

It is shipped with advanced concurrency, resiliency, and flexibility features in addition to covering the most relevant Google Workspace backup use cases. A full feature list is presented below:

- Common features
  - Google Workspace APIs based backups
  - Support for free Gmail accounts
  - Support for accounts under a Google Workspace subscription
  - Multi-service concurrency capabilities
  - Multi-threaded processes
  - Advanced tuning configurations
  - Automatic concurrency of fetching processes
  - Generation of user-friendly report for restore operations
  - Network resiliency mechanisms
  - Latest Google Authentication mechanisms
  - Discovery/List/Query capabilities
  - Restore objects to Google Workspace
    - To original entity
    - To any other entity
  - Restore any object to file-system
  - Restore HTML report to user mailbox or user drive
- Backup and Restore of Google Drive
  - Backup and Restore of Users My Drive
  - Backup and Restore of Shared Drive Units
  - Hash check during backup and restore to ensure data integrity
  - Incremental & Differential backup
    - Includes advanced delta function for improved performance
  - Advanced selection capabilities
    - Include/exclude by name
    - Automatic discovery to backup everything

    - Include/exclude by RegEx
    - Folder selection capabilities for backup
      - Include/exclude by name
      - Automatic discovery to backup everything
      - Include/exclude by regular expressions
  - Support for regular files and also native Google Workspace files (export)
  - Folder and file granularity for restore
  - Computed hash check at backup and restore time
  - Backup and restore of permissions shares
  - Backup and restore of shared elements with users
  - Backup and restore of Google Drive file versions
  - Backup and restore of file comments
  - Backup and restore of trash
- Backup and Restore of Google Mail (GMail)
  - Backup and Restore of email messages
    - Messages metadata
    - Messages content
  - Backup and Restore of attachments
  - Backup and Restore of mailbox settings
    - Auto-Forwarding, Imap, Language and Pop settings
    - Delegates
    - Filters
    - SendAs addresses
    - Forwarding addresses
  - Incremental & Differential backup with Delta function
    - Includes advanced delta function for improved performance
  - Advanced selection capabilities
    - Include/exclude users by name
    - Automatic discovery to backup all Workspace users
    - Include/exclude users by RegEx
    - Label selection capabilities for backup
      - Include/exclude by name
      - Automatic discovery to backup all of them
      - Include/exclude by regular expressions
  - Export mail messages to mime RFC 822 local files
  - Export attachments to local files

- Restore to original GMail mailbox
- Restore to a different user GMail mailbox
- Restore to the original labels
- Restore to a specific label
- Fully indexed information into Bacula Catalog
- Advanced search capabilities for restore operations
- Privacy excluding features:
  - Ability to exclude message fields from the index
  - Exclude private or spam messages through powerful filtering capabilities

---

Note: Future modules

Bacula Google Workspace Plugin will include more modules in the future, like Google Calendar among others.

Architecture

Bacula Enterprise Google Workspace Plugin is using several Google Workspace APIs to perform almost all of its operations. Therefore, the plugin is working at the maximum granularity that the service provides.

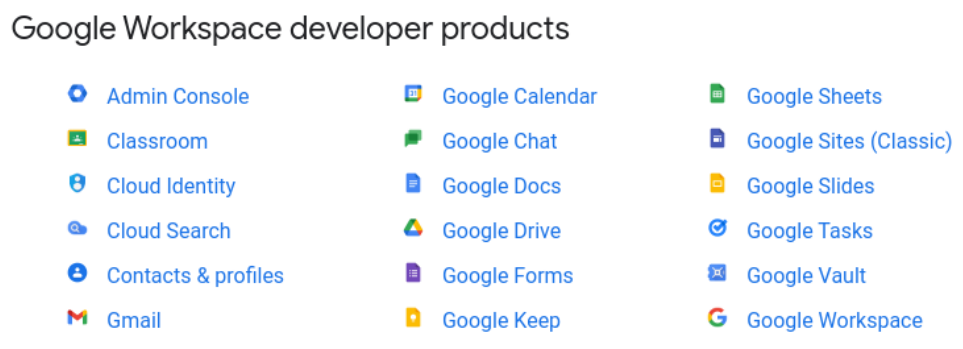


Fig. 81: Google Workspace APIs

All the information is gotten using HTTP requests to Google Cloud from the FD where the plugin is installed.

The plugin will contact a **Google Cloud Platform application** that needs to be **manually created and configured** before using the plugin. It will serve as a bridge to download the required data or objects during backup time and send them to the Storage Daemon. Conversely, the plugin will receive them from an SD and perform uploads as needed during a restore operation.

The implementation is done through a Java Daemon, therefore Java is a requirement in the FD host. For more information about how to create the application in GCP, please, consult auth section.

Below is a simplified vision of the architecture of this plugin inside a generic **Bacula Enterprise** deployment:

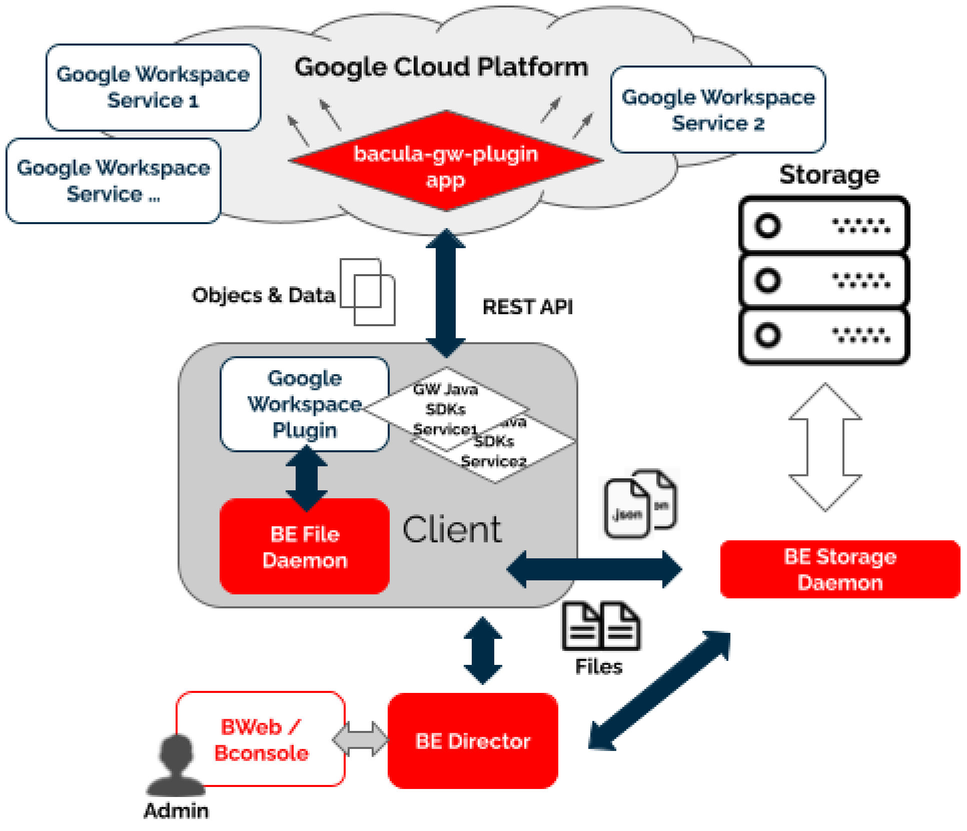


Fig. 82: Google Workspace Plugin Architecture

Listed below is the information that can be protected using this plugin:

- Google Drive
  - My Drive of users
    - Folders
    - Native Google services files (gdocs, gslides, gpresentation.. Export and download)
    - All other files (regular download)
    - File Versions
    - Trash bin
  - Shared drives
    - Folders
    - Native Google services files (gdocs, gslides, gpresentation.. Export and download)
    - All other files (regular download)
    - File Versions
    - Trash bin
  - Shared permissions (direct access, share links, expiration times..)
  - SharedWithMe User files
  - Files comments
- Google Mail
  - Mailbox user Labels
    - System labels: Inbox, Sent, Draft, Spam …
    - User labels
  - Mailbox user Mails
    - Metadata
    - Contents
  - Mail Attachments
  - Mailbox user Settings
    - Auto-Forwarding settings
    - Imap settings
    - Language settings
    - Pop settings settings
  - Delegates addresses
  - Filters
  - SendAs addresses
  - Forwarding addresses

All the metadata information of each object is stored in JSON format preserving all their original values.

Services

In this section we will dig into how this plugin behaves for each particular service, describing special features and and behaviors that require an extended description.

Google Drive

Bacula Enterprise Google Workspace Plugin can protect My Drive units associated to users from a workspace, My Drive units of free accounts, as well as Shared Drive units.

It is possible to utilize advanced selection methods to decide exactly what is backed up, as well as control precisely which items to restore and their destinations.

The detailed list of the information protected with this service is:

- My Drive of users
  - Folders
  - Native Google services files (gdocs, gslides, gpresentation.. Export and download)
  - All other files (regular download)
  - File Versions
  - Trash bin
- Shared drives
  - Folders
  - Native Google services files (gdocs, gslides, gpresentation.. Export and download)
  - All other files (regular download)
  - File Versions
  - Trash bin
- Shared permissions (direct access, share links, expiration times..)
- SharedWithMe User files
- Files comments

Files will keep their names in the catalog and will be included in a path like this:

- `/@gw/customerId/entitykind/entityname/drives/unitname/path/to/file/name-file.extension`

  (where `entitykind` can be users or shared_drives)

Version History

Google Drive can be configured to retain the history for files/items.

Google Drive hash check

Google Drive service stores a hash for every file hosted, using MD5 algorithm. Bacula Enterprise Google Workspace Plugin calculates this hash and compares it to ones stored in the cloud at backup time, and also at restore time in order to ensure data integrity. Debug mode shows information about these hashes. Please note that this is true only for non native Google files

Google Drive duplicated files

Google Drive stores its information in a different way compared to traditional filesystems. Instead of a tree structure, everything in google drive are pairs of keys and values (data maps). This makes some internal differences regarding the data structure and, for example, it is possible to have the same folder with the same name inside the exact same path. Similarly it is possible to have the same file with the same name several times in the same path.

Bacula Enterprise Google Drive plugin will combine the data inside folders with the same name. This is:

- **From Google Drive:**

  mypath/

  DirA/
  f1 f2

  DirA/
  f3 f4

- **To Bacula Catalog:**

  mypath/DirA/
  f1 f2 f3 f4

For files with the same name, bconsole will only show one and will restore the last one when using common options to find the most recent backup or a specific job id where the same file is present more than once. However, for example when using BWeb it is possible to see the different versions of each file and get the desired one using the restore Wizard.

Google Apps files

Files that come from Google online services like Google Docs, Google Spreadsheets or Google Slides are exported transparently during the backup process to open formats.

These kind of files do not expose versioning through Google APIs, so drive_version_history will not take any effect on them. The specific list of files affected by this behavior is as follows:

- Google Docs to odt
- Google Draw to jpeg
- Google Photo to jpeg
- Google Sheets to ods
- Google Slides to odp
- Google Scripts to json

Those are default values, but they can be configured with the proper fileset variables:

- drive_export_format_document, drive_export_format_drawing, drive_export_format_spreadsheet, drive_export_format_presentation, drive_export_format_appscript, drive_export_format_video

For more information about export formats, check the following guide: https://developers.google.com/workspace/drive/api/guides/ref-export-formats

Google Photos and Google Sites

Historically, Google Photos and Google Drive have been very close modules. However today they work with separated APIs. Bacula Workspace Plugin is not supporting Google Photos, even if it is planned to support it in the future as a different module of this plugin.

Google Sites do not support export functions so the plugin cannot protect them. Associated files are simply ignored.

Google Drive shares

Bacula Enterprise Google Drive Plugin is able to backup and restore shared elements. These kind of elements require a special treatment, as they are composed of two parts:

- In the source account, shared elements include special information about the permissions of the share (who and how the share must work)
- In the destination account, shared elements appear within an special category called 'SharedWithMe'.

Shared permissions

Bacula Enterprise Google Workspace Plugin will query for the permissions of an item if this item has been shared directly. This means the plugin will not backup inherited permissions. In order to have inherited permissions in a backup, the top element where the original shared permissions were set needs to be included in the backup and in the restore. As an example, if a directory is shared, but we restore only specific files contained in it, those files will not be shared as they were originally. It is necessary to restore the whole directory in order to replicate the original inherited permissions as they were at the time of backup.

The method to store shared permissions is to include them as 'metadata' of every file. This implies that permissions can only be restored directly to the Google Workspace service. A File Daemon restore to a local filesystem will only restore files, and shared permissions will not be restored.

Shared permissions can include links. Links are pre-generated URLs that can include expiration dates and other configuration parameters such as scopes, types, or affected identities. Shared permissions restores have special characteristics that must be considered and they are described below:

- Permission will be generated exactly as it was, but it will be a new permission object. This is similar to the situation with files. A restored file has the same contents as the original, but can include slightly different metadata because creation process was different.
  - If the permission had a static link, a link will be generated, but the associated URL will be different from the original.

Shared permissions are not restored by default. You need to enable the option 'drive_restore_shared_permissions' during the restore session.

Shared with me

SharedWtihMe elements of each target account, if included in the Fileset, are backed up in a predefined directory called SharedWithMe inside the top folder of every selected account. For example, for a given account youraccout@yourdomain.com in a workspace called customer_id:

```
:caption: **SharedWithMe**

``/@gw/customer_id/users/youraccout@yourdomain.com/drive/my drive/
↪sharedWithMe/``
```

At restore time, sharedWithMe elements are treated as any other regular file. However, it is important to note that sharedWithMe files, as we are in the receiver account, have no sharing permissions.

The plugin has a special parameter at restore time allowing it to skip sharedWithMe elements even if they are selected. This feature is intended to facilitate full restores where source and destination accounts are included. Please, note that a restore of a source account with share elements will present those elements to any receiver account if you enable the option to restore share permissions, as we have discussed in the upper section.

Please, go to the Configuration section of this document to see how to set up the sharedWithMe skip option.

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the drive module.

In order to select the Google Drive module, the common *service* parameter must be equals or be containing the value *drive*.

Entities that can include one drive units are: users, groups or sites.

Op-tion	Re-quire	De-fault	Values	Ex-am-ple	Description
drive_sl	No		Valid names of existing shared drives on the selected workspace separated by ‘,’	im-agesSha unit-My-Compa	Will backup only selected drive units belonging to the specified entity (user, group or site)
drive_sl	No		Valid names of existing shared drives on the selected workspace separated by ‘,’	we-bas-sets	Will backup all drives except the excluded ones in thelist(s), belonging to the specified entity (user, group or site)
drive_sl	No		Valid regex	*.pages	Backup matching drive units (based in the drive unit name)
drive_sl	No		Valid regex	^site.*	Exclude matching drive units (based in the drive unit name)
drive_fi	No		Strings representing existing **folders** for the given users or shared units separated by ‘,’	Cus-tomers, Part-ners	Backup only specified **folders** belonging to the selected users
drive_fi	No		Strings representing existing **folders** for the given users or shared unitsseparated by ‘,’	Per-sonal	Exclude selected **folders** belonging to the selected users
drive_fi	No		Valid regex	.*Com-pany	Backup matching drive folders. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
drive_fi	No		Valid regex	.*Plan	Exclude matching drive folders from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded
drive_ir	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include trashed elements from the user or shared drive selected to backup
drive_ir	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include comments of every file. Please, notice that performance is lower when this option is enabled as extra requests are needed for every single file to backup
drive_sl	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Include SharedWithMe elements of every target entity in the backup process
drive_v	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include Google Drive former versions of every file into the backup process
drive_e	No	odt	odt ; docx ; rft ; pdf ; txt ; zip ; epub	docx	Extension associated to the MIME Type format that Google Documents will be exported to during backup operations. Note that the availability of some of them depends completely on Google

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of Google Drive module restore operations:

- destination_user, drive_destination_shared_unit, destination_path, send_report, allow_duplicates
- drive_skip_sharedwitme, drive_skip_versions, drive_skip_comments, drive_restore_share_permissions,
- debug, foreign_container_generation

Use cases

The following restore scenarios are supported:

- Restore files, directories, or file versions to original drive or to a different drive
  - Restore parameters implied: **destination_user, drive_destination_shared_unit**
- Restore file(s)/dir(s) or file version(s) to original path or to a different path
  - Restore parameters implied: **destination_path**
- Restore file(s)/dir(s) or file version(s) to local file system (general restore **where** parameter must be set to a path)
- It is possible to make general restore selections, but avoid restoring versions
  - Restore parameters implied: **drive_skip_versions**
- It is possible to restore sharing permissions of implied files
  - Restore parameters implied: **drive_restore_share_permissions**
- It is possible to make general restore selections, but specify if backed up shared elements must be considered
  - Restore parameters implied: **drive_skip_sharedwitme**
- It is possible to make general restore selections, but specify if backed up file comments must be considered
  - Restore parameters implied: **drive_skip_comments**
- It is possible to control whether or not duplicate elements are allowed (based on file id):
  - Restore parameters implied: **allow_duplicates**

Particularities:

- If no **destination user** and no **destination shared unit** are provided, the destination user or unit will be looked for inside the backed up path, so the destination entity will be the same as the original one
- If no **destination_path** is provided, the destination path will be the same as the original one
  - If a destination entity was provided, but no destination_path was provided and the selected file did not belong to the destination entity:
    - ∗ A new folder will automatically be created inside the target entity
    - ∗ For each 'foreign' entity, a new folder will be created

        * Inside each 'foreign' entity folder, the original path structure will be preserved when restoring the files

            · *Unless the parameter **foreign_container_generation** is disabled

For more details about the behavior of each restore parameter, please check the general section of restore parameters.

Fileset examples

Please note that Google Drive Plugin works with two kind of entities:

- User Drives
- Shared Drives

By default, if not specifying anything on any parameters, the plugin backups everything. Therefore, in order to only backup users, we need to exclude all shared drives ; in order to backup only shared drives, we need to exclude all users. Below examples should show this more clearly.

Full Google Drive of only one user:

```
:caption: **Fileset Example**

Fileset {
   Name = fs-gw-drive-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=drive credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystmes.com
       user=adelev@baculasystems.com drive_shared_units_regex_exclude=\".*\""
   }
}
```

Folders of one user and include sharedWithMe elements:

```
:caption: **Fileset Example**

Fileset {
   Name = fs-gw-drive-adelev-shared
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=drive credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystmes.com
   user=adelev@baculasystems.com drive_shared_units_regex_exclude=\".*\"␣
↪drive_files=\"dir1,dir2\" drive_shared_with_me=yes"
   }
}
```

Backup of some specific shared drives:

```
:caption: **Fileset Example**

```

(continues on next page)

(continued from previous page)

```
Fileset {
   Name = fs-gw-drive-live-2-drives
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=drive credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystmes.com
   drive_shared_units=myunit1,myunit2 user_regex_exclude=\".*\""
   }
}
```

Exclude directories of two users:

```
:caption: **Fileset Example**

Fileset {
   Name = fs-gw-drive-adjon-users-notemp
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=drive credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystmes.com
      user=\"adelev@baculasystems.com,jonis@baculasystems.com\" drive_shared_
↪units_regex_exclude=\".*\" drive_files_exclude=temp"
   }
}
```

Google Mail

Bacula Enterprise Google Workspace Plugin can protect Google Mailboxes associated to users. It is possible to utilize advanced selection methods to decide exactly what is backed up (labels included/excluded, users included/excluded), as well as control precisely which messages to restore and where (original user's account or another user's account). The information protected with this service is:

- Labels
  - System labels: Inbox, Sent, Draft, Spam ...
  - User labels
- Mails
  - Metadata
  - Contents
- Attachments
- Settings
  - AutoForwarding settings
  - Imap settings
  - Language settings
  - Pop settings settings

- Delegates
- Filters
- SendAs addresses
- Forwarding addresses

Mailbox backup includes the following features:

- Incremental/Differential backup with Delta function:
  - Delta function is applied always, independently of what is the labels selection in the fileset (email_files* parameters)
- MIME object (RFC 822) export:
  - It is possible to restore to a local label a selection of emails in the RFC 822 format. This way emails could be read or imported in any other tool at will
- Attachments:
  - This plugin backs up the message with its attachments in a single file. This means everything will be restored when selecting a given file.
  - For export purposes there is a restore option to provoke the extraction of the attachments as separate files in addition to have the full MIME object.

Messages will be formatted in the catalog in order to not include sensitive information and will be included in a path like this:

- `/@gw/customer_id/users/user@customerdomain.com/email/labelname/AAAAXXXXDDDDDIIIII.msg`
  - Where the message name corresponds to the message Id provided by Google GMail API

Other objects will look as follows:

- `/@gw/customer_id/users/user@customerdomain.com/email/settings/`
  - `addr1@customerdomain.com.mailbox.sasad` -> Send ass address
  - `addr2@customerdomain.com.mailbox.fwad` -> Forwarding address
  - `addr2@customerdomain.com.mailbox.del` -> Delegates address
  - `3838383aaadffdfdf.mailbox.fil` -> Filter
  - `settings.mailbox.autfw` -> Autoforwarding settings
  - `settings.mailbox.imap` -> Imap settings
  - `settings.mailbox.lan` -> Autoforwarding settings
  - `settings.mailbox.pop` -> Autoforwarding settings
  - `settings.mailbox.vac` -> Autoforwarding settings

Backup parameters

The list below shows the specific backup parameters that can be set up in order to control the behavior of the email module.

In order to select the email module, the common *service* parameter must be equals or be containing the value *email*.

Entities that can include mailboxes are: users.

Option	Require	Default	Values	Example	Description
email_fi	No		Strings representing existing **labels** for the given users separated by ‘,’	Inbox, Sent	Backup only specified **labels** belonging to the selected users
email_fi	No		Strings representing existing **labels** for the given users separated by ‘,’	Archive, Personal	Exclude selected **labels** belonging to the selected users
email_fi	No		Valid regex	.*Company	Backup matching labels. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them.
email_fi	No		Valid regex	.*Plan	Exclude matching labels from the selection. Please, only provide list parameters (files + files_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for selection, all elements will be included and this list will be excluded.
email_s	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Backup mailbox settings of included users
email_s	No	Yes	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	No	Backup spam and trashed messages
email_n	No		String representing a valid Boolean Javascript expression regarding email message fields	emailSubject.includ && !emailIsRead	Exclude from backup all messages that match the provided expression
email_n	No		String representing a valid Boolean Javascript expression regarding email message fields	/.*private.com/.	Exclude only from indexing (catalog email tables) messages matching the provided expression
email_fi	No		String representing a list of email message fields	emailFrom, emailSubject	Do not store into the index (catalog email tables) the provided list of message fields

Restore

The list below shows the subset of restore parameters that can be used to control the behavior of email module restore operations:

- destination_user, destination_path, send_report, allow_duplicates, debug, foreign_container_generation
- email_export, email_export_attachments_extract

Use cases

The following restore scenarios are supported:

- Restore labels, emails (with their attachments) to original user or to a different user mailbox
  - Restore parameters implied: **destination_user**
- Restore emails to a specific label of a user:
  - Restore parameters implied: **destination_path**
- Export emails to local file system in export mode :
  - Restore parameters implied: **destination_path**, **email_export**
  - Extract attachments during the export: **email_export_attachments_extract**
- It is possible to control whether or not duplicate elements are allowed (based on file id):
  - Restore parameters implied: **allow_duplicates**

Particularities:

- If no **destination_user** is set, every message will be restored into its original mailbox
- If no **destination_path** is set, every message will be restored into its original path
  - If the selection contains messages from several users:
    - Original user messages will be restored in their original location
    - For other users, a special label will be created with the email address of each of them, containing the full path and messages of the restored objects, unless the parameter **foreign_container_generation** is disabled

  Restore of emails from 2 different users over a third mailbox without destination_path result in auto-generated Restore_date label containing those 2 foreign users with the restored label inside of them
- Restored elements will be duplicated by default, unless **allow_duplicates** variable is disabled
  - Even when disabling that variable, messages will be checked by id. So if there is an element with the same information but different ID, it will not be considered to be a duplicate

For more details about the behavior of each parameter, please check the general section of restore parameters.

Messages exclude expressions

Bacula Systems is aware about one of many privacy concerns that may arise when tools like this Google Workspace Plugin enables the possibility to backup and restore data coming from different users, so the backup administrator can restore potentially private data at his will. Moreover, emails are usually one of the most critical items in terms of privacy.

One of many strategies this plugin offers in order to deal with that problem is the possibility to exclude messages. This is a very powerful feature where it is possible to use quite flexible expressions that allow to select a subset of messages and simply exclude them from the backup:

- email_messages_exclude_expr new fileset parameter

Or only from the index (from the catalog)

- email_messages_exclude_index_expr new fileset parameter

Not only messages can be excluded but also select only a subset of email fields to be included in the protected information. It is possible to exclude fields from the backup index (the catalog):

- email_fields_exclude_index new fileset parameter

All four discussed expressions are based on an internal structure of fields to work with. Below you can see the entire list of fields that you can use:

- emailTags
- emailSubject
- emailFolderName
- emailFrom
- emailTo
- emailCc
- emailBodyPreview
- emailImportance
- emailTime
- emailIsRead
- emailIsDraft

Please note that it is very important to write the fields exactly as written above.

These fields can be used in a comma separated list in the 'email_fields_exclude' parameter and also 'email_fields_exclude_index' parameter.

Then, for 'email_messages_exclude_expr' and 'email_messages_exclude_index_expr' use them in a valid boolean expression in **Javascript** language syntax. Some examples are provided below:

Listing 214: **Expression to exclude messages where subject includes the word 'private'**

```
emailSubject.includes('private')
```

Listing 215: **Complex expression to exclude messages that are not read and are Draft or their label name is named Private**

```
!emailIsRead && (emailIsDraft || emailFolderName == 'Private')
```

Listing 216: **Expression to exclude messages based on the received or sent date**

```
!emailTime < Date.parse('2012-11-01')
```

Listing 217: **Expression to exclude messages using a regex based on emailFrom**

```
/.*private.com/.test(emailFrom)
```

Note: This feature is available since Bacula Enterprise version 14.0

Expression tester

This expression mechanism can sometimes be uncertain for end users, where they can have doubts about the correct behavior of their prepared expressions. In order to help with that, Google Workspace Plugin presents a query method that allows to test those expressions against a static pre-loaded set of data.

There are two commands available:

- Show command: It will show the static data in json format, so it is possible to see the contents to adapt the expressions to test
- Test command: It will apply the expression parameters to the pre-loaded static data

The test command has the following format:

Listing 218: **Expression tester Show command**

```
.query client=<your-fd-client> plugin="gw: credentials_file=/opt/bacula/etc/
↪gw-credentials.json customer_id=xxxxxx admin_user_email=admin@company.com"␣
↪parameter=email-expr-show
```

The show command has the following fomat

Listing 219: **Expression tester Test command**

```
.query client=<your-fd-client> plugin="gw: credentials_file=/opt/bacula/etc/
↪gw-credentials.json customer_id=xxxxxx admin_user_email=admin@company.com␣
↪email_messages_exclude_expr = \"<your-js-expression>\""␣
↪parameter=json|email-expr-test
// Or
.query client=<your-fd-client> plugin="gw: credentials_file=/opt/bacula/etc/
↪gw-credentials.json customer_id=xxxxxx admin_user_email=admin@company.com␣
↪email_messages_exclude_index_expr = \"<your-js-expression>\""␣
↪parameter=json|email-expr-test
```

The test command produces some JSON output with objects with the same format that the plugin uses to store data into the catalog. Please note the 'total' value at the end, where the value of 12 total pre-loaded messages is shown

Listing 220: **Expression tester Show command output**

```
.query client=<your-fd-client> plugin="gw: credentials_file=/opt/bacula/etc/
↪gw-credentials.json customer_id=xxxxx admin_user_email=admin@company.com"␣
↪parameter=json|email-expr-show
....
    "email-12": {
      "body": {
        "content": "These are the contents in text format of the 12 email of␣
↪test data. It has the following categories:orange, black, white, purpleYou␣
↪can try to filter this body using any JS method like /.*12.*/.
↪test(emailBody) or emailBody.includes(12)",
        "contentType": "TEXT"
      },
      "ccRecipients": [
        {
          "emailAddress": {
            "address": "danny@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "lucas@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "terese@other.com"
          }
        }
      ],
      "from": {
        "emailAddress": {
          "address": "elon@other.com"
        }
      },
      "hasAttachments": false,
      "isDraft": false,
      "isRead": false,
      "replyTo": [
        {
          "emailAddress": {
            "address": "elon@other.com"
          }
        }
      ],
      "sentDateTime": {
        "dateTime": {
          "date": {
```

(continues on next page)

(continued from previous page)

```
            "year": 2021,
            "month": 12,
            "day": 5
          },
          "time": {
            "hour": 11,
            "minute": 30,
            "second": 0,
            "nano": 0
          }
        },
        "offset": {
          "totalSeconds": 0
        }
      },
      "subject": "This is private subject 12",
      "toRecipients": [
        {
          "emailAddress": {
            "address": "laura@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "jack@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "john@other.com"
          }
        }
      ],
      "categories": [
        "orange",
        "black",
        "white",
        "purple"
      ]
    }
  },
  {
    "total": "12"
  }
```

The test command, on its side will produce two different outputs. The first part presents the same format as the show format, and those are the messages that would be included in the backup. The second part presents a different format, so an output like:

Listing 221: **Expression tester Test command, index part output**

```
.query client=<your-fd-client> plugin="gw: credentials_file=/opt/bacula/etc/
↪gw-credentials.json customer_id=xxxxxx admin_user_email=admin@company.com"␣
↪parameter=json|email-expr-show
....
      {
    "meta-email-12": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "johndoe.onmicrosoft.com",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is private subject 12",
      "EmailFolderName": "/",
      "EmailFrom": "elon@other.com",
      "EmailTo": "laura@other.com,jack@other.com,john@other.com",
      "EmailCc": "danny@other.com,lucas@other.com,terese@other.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 235,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "gw"
    }
  },
  {
    "total-backup": "12"
  },
  {
    "total-index": "12"
  }
```

That part represents the information that would be indexed in the backup (included into the catalog). You can also see the total entries at the end, this is very useful to quickly compare with the original 12 value and knowing if our expression is filtering the expected data or not. Below we provide an example where some filtering is applied to the backup, but also to the index:

Listing 222: **Expression tester Test command, index part output**

```
.query client=127.0.0.1-fd plugin="gw: credentials_file=/opt/bacula/etc/gw-
↪credentials.json customer_id=xxxxyyyy admin_user_email=jorge@company.com␣
↪email_messages_exclude_expr=\"emailFrom == 'elon@other.co ==
↪'elon@otessages_exclude_index_expr=\"emailSubject.includes('private')\""␣
↪parameter=email-expr-test
...
    meta-email-4={
     "EmailId": "";
     "EmailOwner": "jorge@company.com";
```

(continues on next page)

(continued from previous page)

```
     "EmailTenant": "xxxxyyyy";
     "EmailTime": "2021-08-05 12:30:00";
     "EmailTags": "SENT;UNREAD;SENT;orange;black;white;purple";
     "EmailSubject": "This is orange subject 8";
     "EmailFolderName": "sent";
     "EmailFrom": "bob@company.com";
     "EmailTo": "john@company.com";
     "EmailCc": "terese@company.com";
     "EmailInternetMessageId": "1533123860.7.1655130748637@jorge-Bravo-15-Bac
↪";
     "EmailBodyPreview": "These are the contents in text format of the 8␣
↪email of test data. It has the following categories:orange; black; white;␣
↪purpleYou can try to filter this body using any JS method like /.*8.*/.
↪test(emailBody) or emailBody.includes(8)";
     "EmailImportance": "";
     "EmailConversationId": "";
     "EmailIsRead": 1;
     "EmailIsDraft": 0;
     "EmailHasAttachment": 0;
     "Type": "EMAIL";
     "Version": 1;
     "Plugin": "gw"
   }
   total-backup=6
   total-index=4
```

In case your expression is not valid, the plugin will also inform about that with the following message:

- error=Error listing elements. Cause: Predicate test error!! Review your query ....

Fileset examples

Backup Full MailBox of some users, but excluding some labels:

Listing 223: **Fileset Example**

```
Fileset {
   Name = fs-gw-drive-adjon-users-notemp
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=email credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystems.com
   user=\"adelev@baculasystems.com,jonis@baculasystems.com\" email_files_
↪exclude=\"*.temporary\""
   }
}
```

Backup all MailBoxes:

Listing 224: **Fileset Example**

```
Fileset {
   Name = fs-gw-email-all
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=email credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystems.com"
   }
}
```

Backup only the Inbox label of some users:

Listing 225: **Fileset Example**

```
Fileset {
   Name = fs-gw-email-2user-inbox
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=email credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystems.com
   user="peter@baculasystems.com,john@baculasystems.com" email_files=inbox"
   }
}
```

Backup some users and include settings:

Listing 226: **Fileset Example**

```
Fileset {
   Name = fs-gw-email-2user-mime
   Include {
      Options { signature = MD5 }
      Plugin = "gw: service=email credentials_file=/opt/bacula/etc/bacula-gw-
↪plugin-credentials.json customer_id=G39add31l1 admin_user_
↪email=super@baculasystems.com
      user="peter@baculasystems.com,miriam@baculasystems.com" email_
↪settings=yes"
   }
}
```

System labels

Google Mail can present the labels information in local languages to the user.

In general, there is no 'multilanguage' support, in the sense that labels must be included with their original name. For example, if you create a label named 'books', you cannot expect it to be backed up if you use something like 'livres' or 'libros' from other languages. You need to use the real name that was used to create such label.

There is one very important special case though, which is 'system labels'. System labels are labels like 'inbox', 'sent' … A full list can be found here: https://developers.google.com/gmail/api/guides/labels

These kind of labels can be 'found' by the plugin using their standard name, instead of their internal id, as it's the general case. Therefore, for them it is possible to get the label using their English well known word even if the user sees the label with a translated word.

For example, to backup inbox it is needed to use 'inbox' even if for some users it is 'Posteingang' or 'boîte de réception'. Google Workspace Plugin will recognize these special words and will query the information through them.

To summarize:

- System labels -> Use English word
- Other user labels -> Use original name

Special Features

In the following section, special features and behaviors are detailed.

File Revisions

Google Drive service can be configured to retain the history for files/items (this is versions or revisions of the same file).

Depending on the service and configuration, a new version can be created for each edit, each time the file is saved, manually, or never. Previous versions of a document may be retained for a finite period of time depending on admin settings which may be unique per user or location. By default, this feature is enabled and Google will store up to 100 revisions for each item and/or revisions younger than 30 days (they are deleted after 30 days).

This feature is usually enabled and the information may be accessed through the 'Manage versions' option available once a file has been selected.

Bacula Enterprise Google Workspace Plugin is able to backup this information if the special **drive_version_history** backup parameter is activated.

File revisions have some particularities compared to normal files:

- They are backed up as a regular file. This means a revision has its own full metadata as the parent file itself. All the metadata is the same as the file contains, except for size, dates and name.
- The name of the file is modified, so at restore time you can see the version number and the version date in the filename. Example:
  - Parent file: myDoc.doc
  - Versions:

- myDoc###v25.0_2021-01-19_234537.doc
    - myDoc###v24.0_2021-01-17_212537.doc
    - myDoc###v23.0_2021-01-12_104537.doc
    - ...
    - *Notice that the extension of the file is kept in order to easily identify a possible name modification in GDrive, once the file is restored

- Versions are not restored by default. You need to disable the special restore parameter 'drive_skip_versions', setting it to 0.

File versions are backed up in all backup levels (Full, Incremental, Differential), this means you can track all the changes of the files in your backup. For example, every Incremental run is going to backup only the new modified versions since the last Full or Incremental execution.

Here is an example of some files backed up with revisions included, listed in a restore session:

Listing 227: **versions in a job**

```
cwd is: /@gw/XXXXX/users/jorge@baculasystems.com/drive/my drive/SOURCE_
↪REGRESS_20220512104335/
$ ls
Contentiones/
Dolores###v_2022-05-12_104436729.doc
Dolores###v_2022-05-12_104444796.doc
Dolores###v_2022-05-12_104448264.doc
Dolores.doc
Dolores.doc__comments/
Legimus###v_2022-05-12_104518541.mp4
Legimus###v_2022-05-12_104527444.mp4
Legimus###v_2022-05-12_104530638.mp4
Legimus.mp4
Legimus.mp4__comments/
Netus.ppt
Posse###v_2022-05-12_104456414.docx
Posse###v_2022-05-12_104506748.docx
Posse###v_2022-05-12_104510261.docx
Posse.docx
Posse.docx__comments/
Ridiculus.jpeg
```

Note: It is important to keep in mind that versions have no Delta or Changes in special function on the API side and cannot be filtered by date. Therefore, the process to decide if a version needs to be backed up or not requires the plugin to walk through all existent versions of a modified item. In some situations this could have some undesired impact on backup performance.

Delta Backup

Some modules of the Google Workspace Plugin implement a specific mechanism to handle incremental or differential backups in order to optimize them and offer good performance. In general, the plugin will store information about the current state of the running backup. It will use that checkpoint information in the next backup, so Google APIs return only the new elements or modified elements since then.

The Google Workspace Drive API provides a ‘Changes’ function to track changes of some objects in an efficient way. Bacula Enterprise Google Workspace Plugin uses this function for its Delta mechanism in order to speed up Incremental/Differential processes.

For Google Email API there is a ‘History’ function where each email stores a sequential number which is tied to the date when it was created. Bacula Enterprise Google Workspace Plugin uses this function for its Delta mechanism in order to speed up Incremental/Differential processes.

Please note that using this function is not a mandatory requirement, therefore Incremental or Differential backups will function also with the services that may not support it by navigating through the information and comparing the date with the last valid backup of the current chain.

Delta function has some important characteristics:

- In Google Drive, the function can only be used for full entities (full user or full shared drive). This means that selecting specific paths to backup will not trigger the Delta function.
- In Google Email it is possible to select specific labels and still get the benefits of the delta function.
- Google Drive delta tokens can expire at some point, or even become invalid due to internal Google issues. If this situation happens, the plugin will try to start a new Delta cycle
- Any situation where the Delta function cannot be used will trigger a regular Full/Inc/Diff where every element is listed and selected or discarded according to the item dates.

The Delta backup cycle is described below:

- Full backup: All entity elements are backed up. The current token (token_1) is received from the API (or the most recent historyId for Email module). This token is stored locally by the FD.
- Incremental 1 backup: token_1 is used to retrieve changes since token_1’s generation so every change is backed up. A new token is generated and stored locally by the FD.
- Incremental 2 backup: token_2 is used to retrieve changes since token_2’s generation so every change is backed up. A new token is generated and stored locally by the FD.
- And so on…

Tokens are stored in an file placed in a path defined by the **path** parameter of the plugin. The name is: jobname.deltaLink

The file stores tokens required for every execution and it is renewed (emptied) during every Full backup execution.

This file is also backed up in the backup itself, so it can be restored manually, before an Incremental/Differential execution in case it was lost and in case you don’t want to run a Full backup again.

Here we can see an example of the contents of the file, with 3 executions and one user entity involved. The structure is tree-based, so it is easy to understand what would be generated in the case of backing up other entities:

Listing 228: **deltaLink**

```
{
  "jobName": "GW_DEMO_JOB",
  "deltaServices": {
    "DRIVE": {
      "entities": {
        "JorgeShared1": {
          "id": "JorgeShared1",
          "name": "jorgeshared1",
          "containers": {
            "jorgeshared1": {
              "id": "jorgeshared1",
              "description": "jorgeshared1",
              "deltaEntries": [
                {
                  "job": "pluginTest.2022-05-12_10.54.30_15",
                  "date": "May 12, 2022, 10:54:30 AM",
                  "delta": "5677"
                },
                {
                  "job": "pluginTest.2022-05-12_11.00.59_18",
                  "date": "May 12, 2022, 11:00:59 AM",
                  "delta": "5677"
                },
                {
                  "job": "pluginTest.2022-05-12_11.03.35_21",
                  "date": "May 12, 2022, 11:03:35 AM",
                  "delta": "5683"
                }
              ]
            }
          }
        },
        "kara@baculasystems.com": {
          "id": "kara@baculasystems.com",
          "name": "kara@baculasystems.com",
          "containers": {
            "my drive": {
              "id": "my drive",
              "description": "my drive",
              "deltaEntries": [
                {
                  "job": "pluginTest.2022-05-12_10.54.30_15",
                  "date": "May 12, 2022, 10:54:30 AM",
                  "delta": "490"
                },
                {
                  "job": "pluginTest.2022-05-12_11.00.59_18",
                  "date": "May 12, 2022, 11:00:59 AM",
                  "delta": "490"
                },
```

(continues on next page)

(continued from previous page)

```
                {
                  "job": "pluginTest.2022-05-12_11.03.35_21",
                  "date": "May 12, 2022, 11:03:35 AM",
                  "delta": "493"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

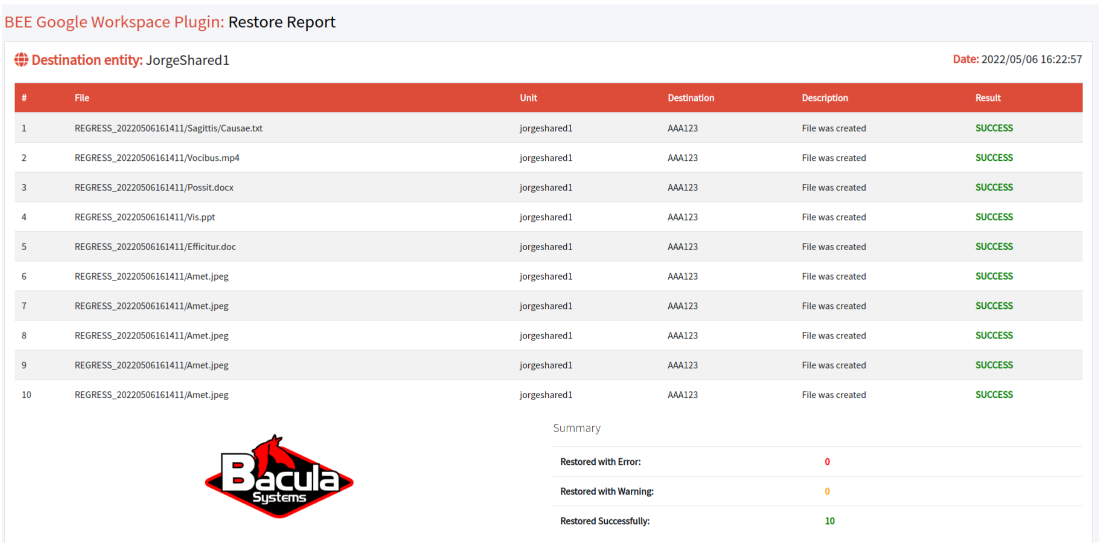
User Restore Report

Information stored in Google Workspace services can represent very sensitive information for end-users. For that reason, information included in backup/restore logs is not exhaustive by default. However, for reporting and controlling purposes, the information of what has been exactly restored, what permissions have been applied, and other information can be useful and necessary for the affected user.

Bacula Enterprise Google Workspace Plugin includes an option to generate a restore report in its Drive unit. The restore report contains detailed information about the items that have been restored successfully, if any of them had any trouble during the restore, and it also reports the date when the action was performed.

The generation of the report can be enabled/disabled in the bconsole restore session. If enabled the report will generate an HTML file that will be stored inside the drive unit of the user (Drive restore) or it will be generated as an email inside the user mailbox (Email restore).

The image below shows an example report from a Drive restore session:



BEE Google Workspace Plugin: Restore Report

Destination entity: JorgeShared1

Date: 2022/05/06 16:22:57

#	File	Unit	Destination	Description	Result
1	REGRESS_20220506161411/Sagittis/Causae.txt	jorgeshared1	AAA123	File was created	SUCCESS
2	REGRESS_20220506161411/Vocibus.mp4	jorgeshared1	AAA123	File was created	SUCCESS
3	REGRESS_20220506161411/Possit.docx	jorgeshared1	AAA123	File was created	SUCCESS
4	REGRESS_20220506161411/Vis.ppt	jorgeshared1	AAA123	File was created	SUCCESS
5	REGRESS_20220506161411/Efficitur.doc	jorgeshared1	AAA123	File was created	SUCCESS
6	REGRESS_20220506161411/Amet.jpeg	jorgeshared1	AAA123	File was created	SUCCESS
7	REGRESS_20220506161411/Amet.jpeg	jorgeshared1	AAA123	File was created	SUCCESS
8	REGRESS_20220506161411/Amet.jpeg	jorgeshared1	AAA123	File was created	SUCCESS
9	REGRESS_20220506161411/Amet.jpeg	jorgeshared1	AAA123	File was created	SUCCESS
10	REGRESS_20220506161411/Amet.jpeg	jorgeshared1	AAA123	File was created	SUCCESS

Summary

Restored with Error:	0
Restored with Warning:	0
Restored Successfully:	10

Fig. 83: Restore Example Drive Report

Installation

The Bacula File Daemon and the Google Workspace Plugin need to be installed on the host that is going to connect to for cloud based services. The plugin is implemented over a Java layer, therefore it can be deployed on the platform better suited for your needs among any of the officially supported platforms of **Bacula Enterprise** (RHEL, SLES, Debian, Ubuntu, etc). Please, note that you may want to deploy your File Daemon and the plugin on a virtual machine directly deployed in Google Cloud Platform in order to reduce the latency between it and the Google Workspace APIs.

The system must have Java >= 11 installed (openjdk-11-jre for example) and the Java executable should be available in the system PATH.

Bacula Packages

We are taking Debian Buster as the example base system to proceed with the installation of the **Bacula Enterprise** Google Workspace Plugin. In this system, the installation is most easily done by adding the repository file suitable for the existing subscription and the Debian version utilized. An example would be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 229: **APT**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪buster-64/ buster main
deb https://www.baculasystems.com/dl/@customer-string@/debs/gw/@version@/
↪buster-64/ buster gw
```

After that, a run of apt update is needed:

Listing 230: **APT install**

```
apt update
```

Then, the plugin may be installed using:

Listing 231: **APT install**

```
apt install bacula-enterprise-google-workspace-plugin
```

The plugin has two different packages implied that should be installed automatically with the command shown:

- bacula-enterprise-google-workspace-plugin
- bacula-enterprise-google-workspace-plugin-libs

Alternately, manual installation of the packages may be done after downloading the packages from your Bacula Systems provided download area, and then using the package manager to install. An example:

Listing 232: **APT install**

```
dpkg -i bacula-enterprise-*
```

The package will install the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-google-workspace-plugin-x.x.x.jar and bacula-google-workspace-plugin-libs-x.x.x.jar). Please note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.
- Plugin connection file (gw-fd.so) in the plugins directory (usually /opt/bacula/plugins)
- Backend file (gw_backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-google-workspace-plugin-x.x.x.jar file in order to launch it, even though usually we should have only one file.

Configuration

Authorization

The first step in order to use the **Bacula Enterprise** Google Workspace Plugin is to authorize it to handle data of the target workspace to backup.

The way of doing this is to:

- Define a Project in Google Cloud Platform
- Activate the proper APIs
- Generate a service account on that project with permissions
- Generate and get the credentials for that service account
- Connect the project and service credentials to the target Google Workspace to protect using domain wide permissions

Once those steps are completed, we also need to find our customer_id as well as an admin user email.

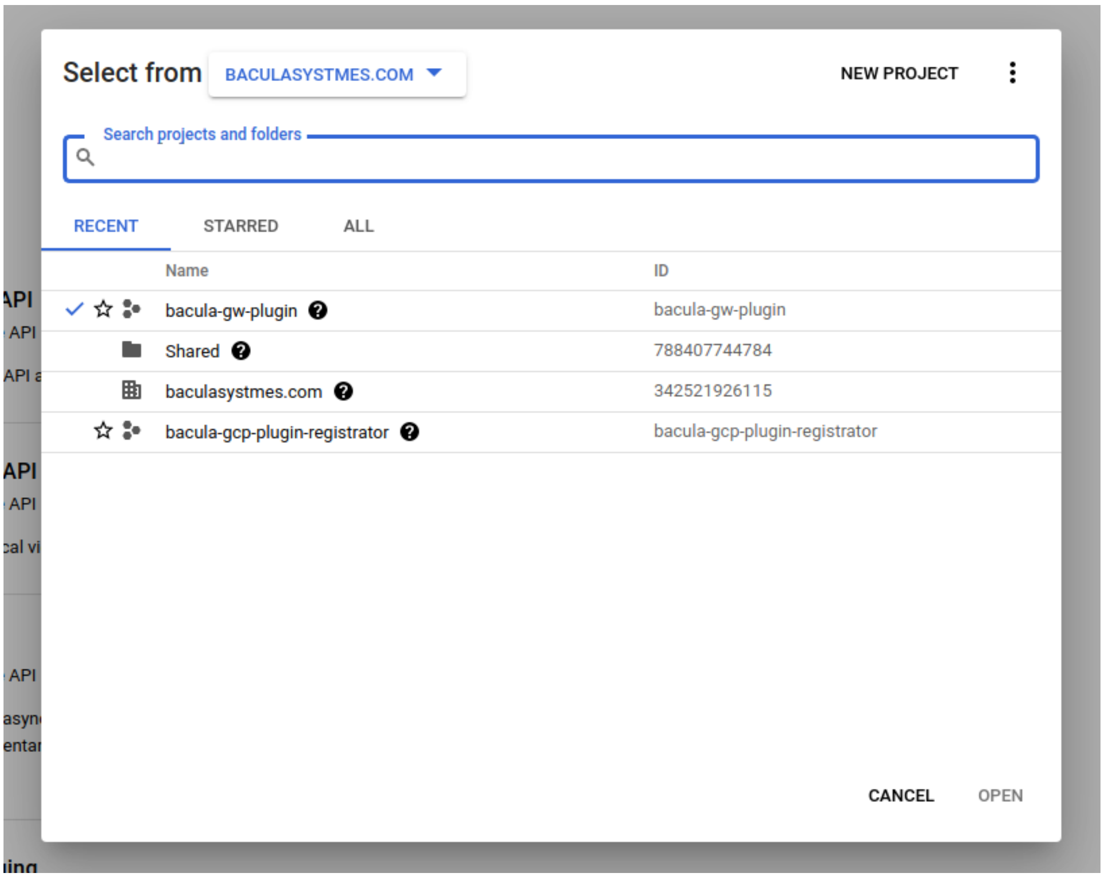
For protecting free users instead of Workspace users, the steps are very similar:

- Define a Project in Google Cloud Platform
- Activate the proper APIs
- Generate and get key credentials
- Add the target addresses to the allowed list of addresses

Google Cloud Platform Project Selection

We need to login with an administrator user to the Google Cloud Platform Console (https://console.cloud.google.com/).

Once there, we need to create a project inside our organization or select an existing one, using the combo-box located close to the Google Cloud Platform logo in the header.



Name	ID
bacula-gw-plugin	bacula-gw-plugin
Shared	788407744784
baculasystmes.com	342521926115
bacula-gcp-plugin-registrator	bacula-gcp-plugin-registrator

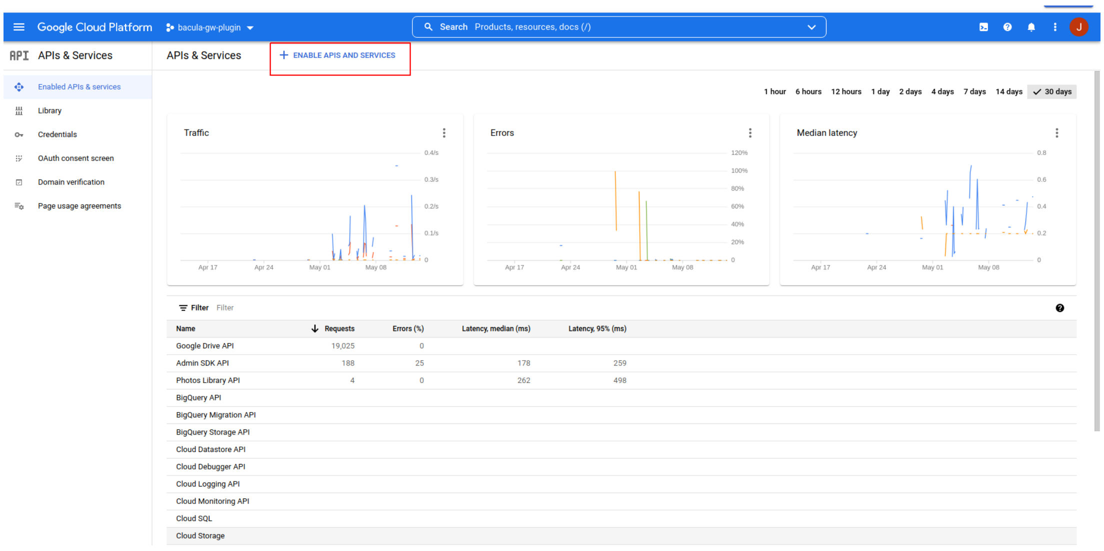
Fig. 84: Google Workspace Projects

Note: Workspace And Free users

This step is exactly the same for Google Workspace environment or free users.

Activate APIs

Once the project is selected, we need to go to API & Services > Enabled API & services.

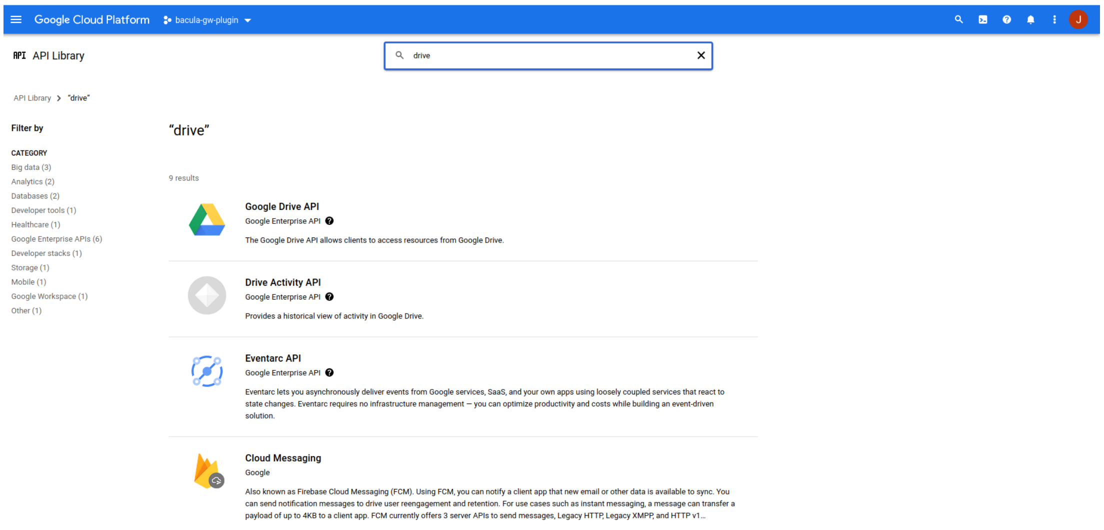


Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Google Drive API	19,025	0		
Admin SDK API	188	25	178	259
Photos Library API	4	0	262	498
BigQuery API				
BigQuery Migration API				
BigQuery Storage API				
Cloud Datastore API				
Cloud Debugger API				
Cloud Logging API				
Cloud Monitoring API				
Cloud SQL				
Cloud Storage				

Fig. 85: Google Workspace Project APIs

From there, it is needed to click on 'Enable APIs and Services' button. Once there, we can search and activate the required APIs.

We can search for the name of each API in order to activate it. In the sample, we look for the 'Google Drive API'.



"drive"

9 results

Google Drive API
Google Enterprise API
The Google Drive API allows clients to access resources from Google Drive.

Drive Activity API
Google Enterprise API
Provides a historical view of activity in Google Drive.

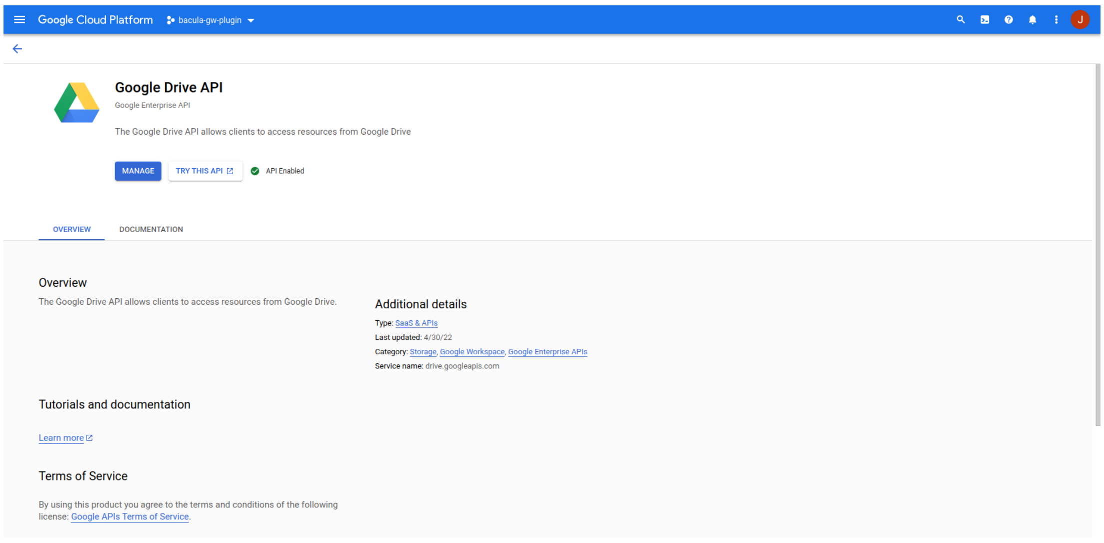
Eventarc API
Google Enterprise API
Eventarc lets you asynchronously deliver events from Google services, SaaS, and your own apps using loosely coupled services that react to state changes. Eventarc requires no infrastructure management — you can optimize productivity and costs while building an event-driven solution.

Cloud Messaging
Google
Also known as Firebase Cloud Messaging (FCM). Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user reengagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4KB to a client app. FCM currently offers 3 server APIs to send messages, Legacy HTTP, Legacy XMPP, and HTTP v1...

Fig. 86: Google Workspace Project Search Drive

We select it and we need to enable it. Once activated, the activation button will change to show 'Manage' as the image below.

The APIs that we need to be enabled are: **Google Drive API**, **Gmail API** and **Admin SDK API**.


Google Drive API

Google Enterprise API

The Google Drive API allows clients to access resources from Google Drive

Overview

The Google Drive API allows clients to access resources from Google Drive.

Additional details

Type: SaaS & APIs
Last updated: 4/30/22
Category: Storage, Google Workspace, Google Enterprise APIs
Service name: drive.googleapis.com

Tutorials and documentation

Learn more

Terms of Service

By using this product you agree to the terms and conditions of the following license: Google APIs Terms of Service.

Fig. 87: Google Workspace Project Enabled APIs

Note: Workspace And Free users

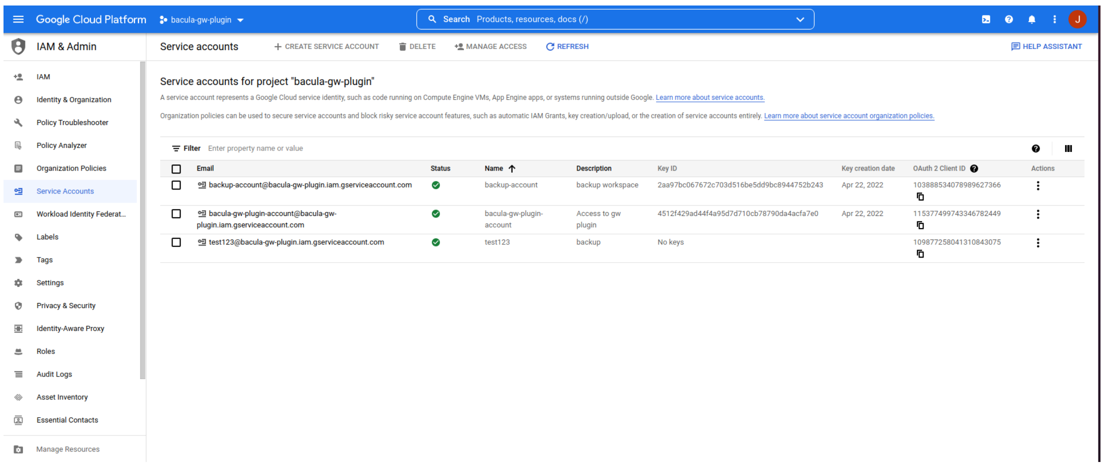
This step is exactly the same for Google Workspace environment or free users.

Service Account

Note: Workspace only

This step is only needed for Workspace environments.

From the same project, now it is needed to go to 'IAM & Admin' > Service Accounts.


Service accounts for project "bacula-gw-plugin"

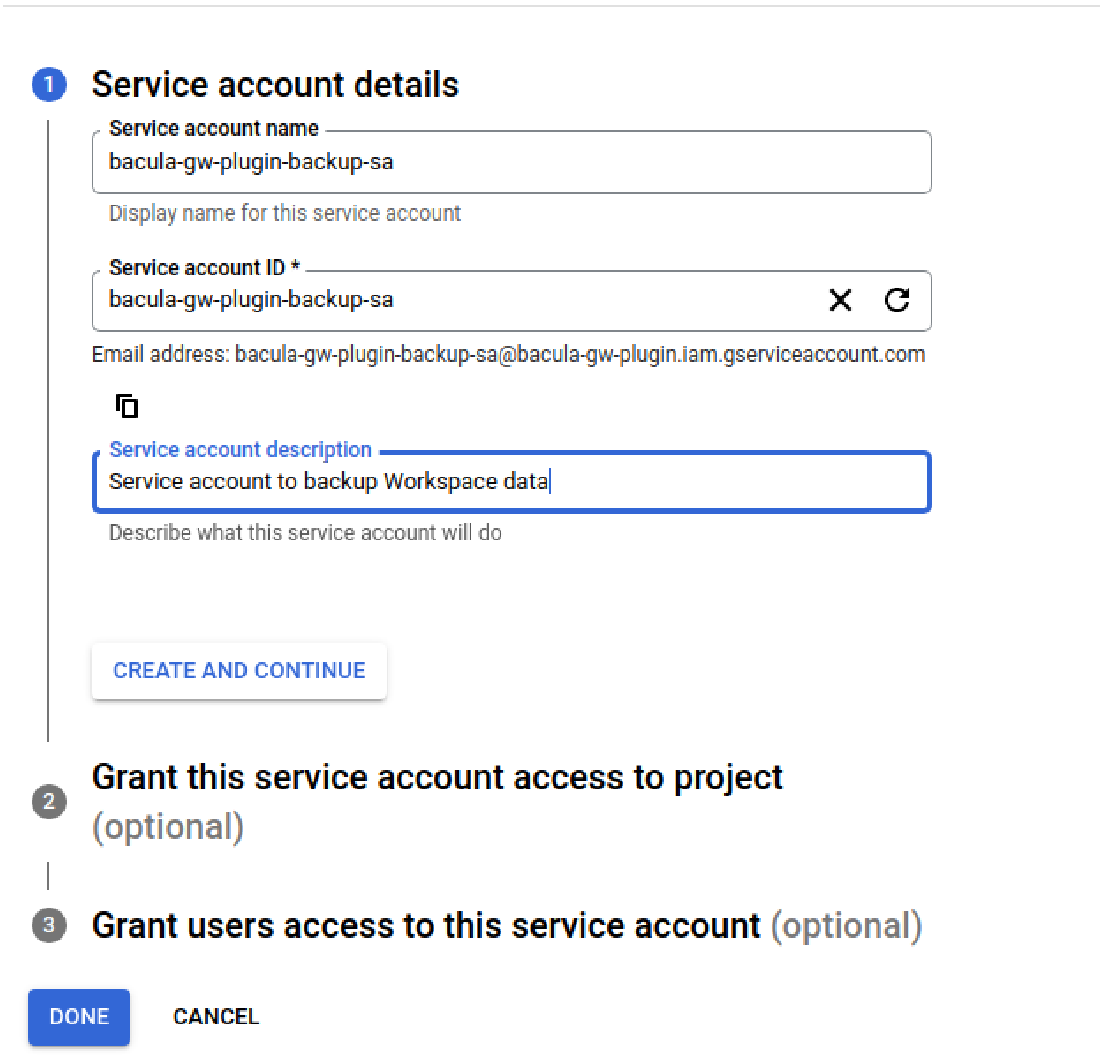
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
backup-account@bacula-gw-plugin.iam.gserviceaccount.com	✓	backup-account	backup workspace	2aa97bc067672c703d516be5dd9bc8944752b243	Apr 22, 2022	103888534078989627366	⋮
bacula-gw-plugin-account@bacula-gw-plugin.iam.gserviceaccount.com	✓	bacula-gw-plugin-account	Access to gw plugin	4512f429ad44f4a95d7d710cb78790da4acfa7e0	Apr 22, 2022	115377499743346782449	⋮
test123@bacula-gw-plugin.iam.gserviceaccount.com	✓	test123	backup	No keys		109877258041310843075	⋮

Fig. 88: Google Workspace Project Service accounts

We click on 'Create Service Account' and fill the form with values as the ones shown here:


1 Service account details

Service account name: bacula-gw-plugin-backup-sa

Display name for this service account

Service account ID *: bacula-gw-plugin-backup-sa

Email address: bacula-gw-plugin-backup-sa@bacula-gw-plugin.iam.gserviceaccount.com

Service account description: Service account to backup Workspace data

Describe what this service account will do

CREATE AND CONTINUE
2 Grant this service account access to project (optional)
3 Grant users access to this service account (optional)
DONE
CANCEL

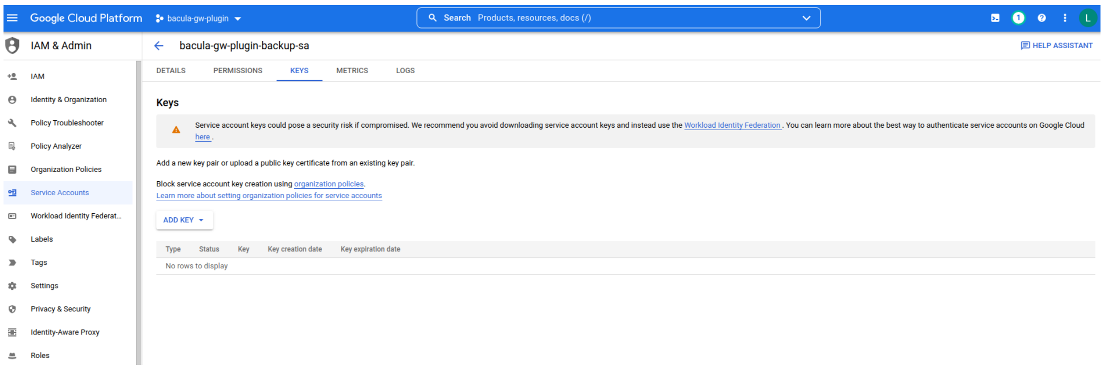
Fig. 89: Google Workspace Project Service accounts Step 1

In the second step, no special role is needed for the service account, as the permissions will be defined just inside Google Workspace after in the 'Connect Project to Google Workspace' section. There is no need to add anything in the third step either.

Service Account Key

This step is only needed for Workspace environments.

Now the service account is created and we will see it in the list. We need to select it now in order to generate a key, which will be the 'credentials_file' to use in any fileset of this plugin:



bacula-gw-plugin-backup-sa

Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .

Add a new key pair or upload a public key certificate from an existing key pair.

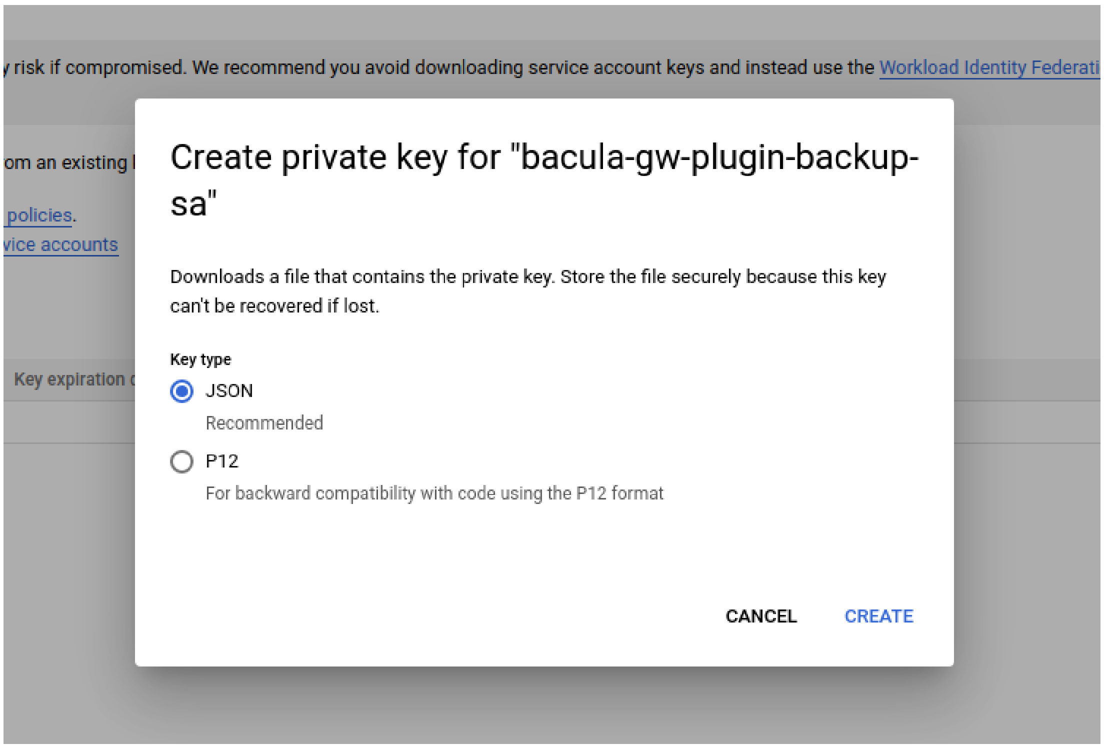
Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

Type	Status	Key	Key creation date	Key expiration date

No rows to display

Fig. 90: Google Workspace Service account keys

We need to generate a new one using the 'ADD KEY > Create new key' button. We select JSON format:



y risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federati
om an existing k
policies.
vice accounts
Key expiration d
CANCEL
CREATE

Create private key for "bacula-gw-plugin-backup-sa"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

[✓ JSON]
Recommended

P12
For backward compatibility with code using the P12 format

Fig. 91: Google Workspace Project Service account key creation

Once we accept, a json file containing all the information we need to use to connect to the project will be downloaded. We need to securely store that file and be referencing it in any fileset through the 'credentials_file' parameter.

It is recommended to rotate the key from time to time for security reasons. As a result, some expiration policy should be used: https://cloud.google.com/blog/products/identity-security/introducing-time-bound-key-authentication-for-service-accounts

When a key expires, a new one must be generated and replace the outdated .json file. This can be scripted using google cli commands: https://cloud.google.com/iam/docs/keys-create-delete

However, the exact procedure is out of scope as it implies extra privileges and a key lifecycle management that should be handled outside the plugin from a secure place in a transparent way for this plugin operation.

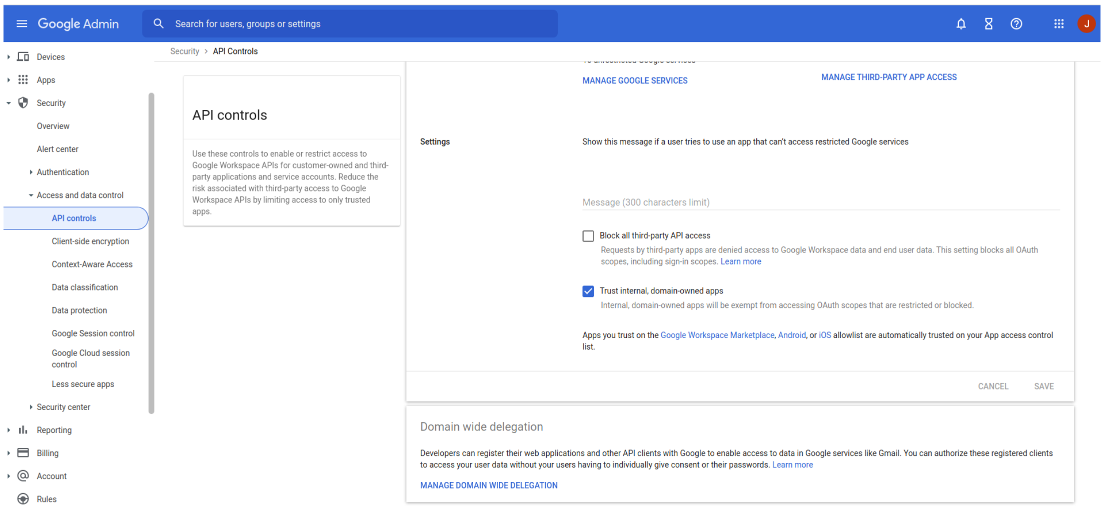
Connect Project to Google Workspace

Note: Workspace only

This step is only needed for Workspace environments.

Without closing the Google Cloud Platform window, we need now to open a new tab and login to the Google Workspace Admin console: https://admin.google.com/

Once there, it is needed to open the 'Security > Access and data control > API controls' option:



API controls

Use these controls to enable or restrict access to Google Workspace APIs for customer-owned and third-party applications and service accounts. Reduce the risk associated with third-party access to Google Workspace APIs by limiting access to only trusted apps.

Settings

Show this message if a user tries to use an app that can't access restricted Google services

Message (300 characters limit)

- [ ] Block all third-party API access
  Requests by third-party apps are denied access to Google Workspace data and end user data. This setting blocks all OAuth scopes, including sign-in scopes. Learn more
- [x] Trust internal, domain-owned apps
  Internal, domain-owned apps will be exempt from accessing OAuth scopes that are restricted or blocked.

Apps you trust on the Google Workspace Marketplace, Android, or iOS allowlist are automatically trusted on your App access control list.

Domain wide delegation

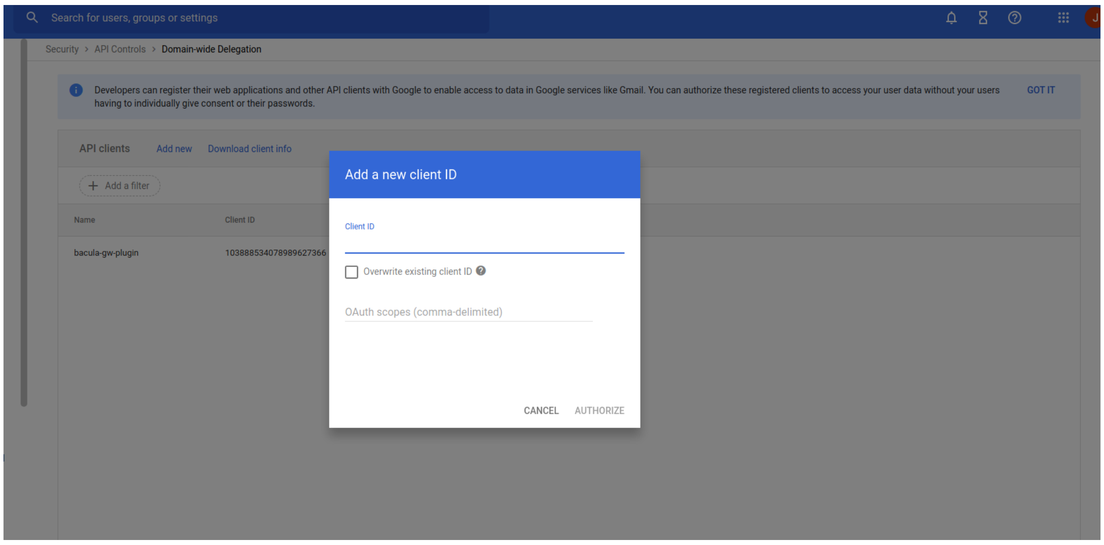
Developers can register their web applications and other API clients with Google to enable access to data in Google services like Gmail. You can authorize these registered clients to access your user data without your users having to individually give consent or their passwords. Learn more

Fig. 92: Google Workspace Admin API controls

We need to make sure we have enabled the check 'Trust internal, domain-owned apps'. Then, click on 'Manage Domain Wide delegation' option located at the bottom of the screen. Then we click the 'Add new' button to see:

The value we need there is the ID of the Service account we created in the previous step of this authentication guide. So we go back to the tab where we had services accounts and click on the 'Details' tab of our created service account. There we will find the required ID:

We copy that value into the Client ID field. For OAuth scopes, we need to put all the following ones:



Search for users, groups or settings

Security > API Controls > Domain-wide Delegation

Developers can register their web applications and other API clients with Google to enable access to data in Google services like Gmail. You can authorize these registered clients to access your user data without your users having to individually give consent or their passwords.

GOT IT

API clients   Add new   Download client info

+ Add a filter

Name	Client ID
bacula-gw-plugin	103888534078989627366

Add a new client ID

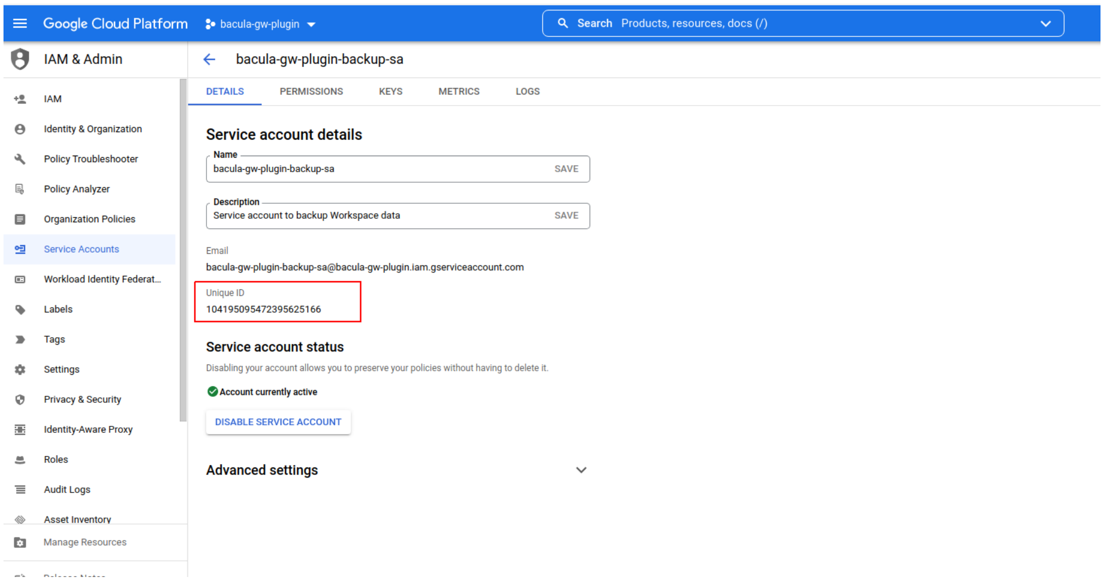
Client ID

☐ Overwrite existing client ID

OAuth scopes (comma-delimited)

CANCEL   AUTHORIZE

Fig. 93: Google Workspace Admin new domain wide client



Google Cloud Platform   bacula-gw-plugin   Search   Products, resources, docs (/)

IAM & Admin

IAM
Identity & Organization
Policy Troubleshooter
Policy Analyzer
Organization Policies
Service Accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy & Security
Identity-Aware Proxy
Roles
Audit Logs
Asset Inventory
Manage Resources

bacula-gw-plugin-backup-sa

DETAILS   PERMISSIONS   KEYS   METRICS   LOGS

Service account details

Name: bacula-gw-plugin-backup-sa   SAVE

Description: Service account to backup Workspace data   SAVE

Email
bacula-gw-plugin-backup-sa@bacula-gw-plugin.iam.gserviceaccount.com

Unique ID
104195095472395625166

Service account status

Disabling your account allows you to preserve your policies without having to delete it.

✓ Account currently active

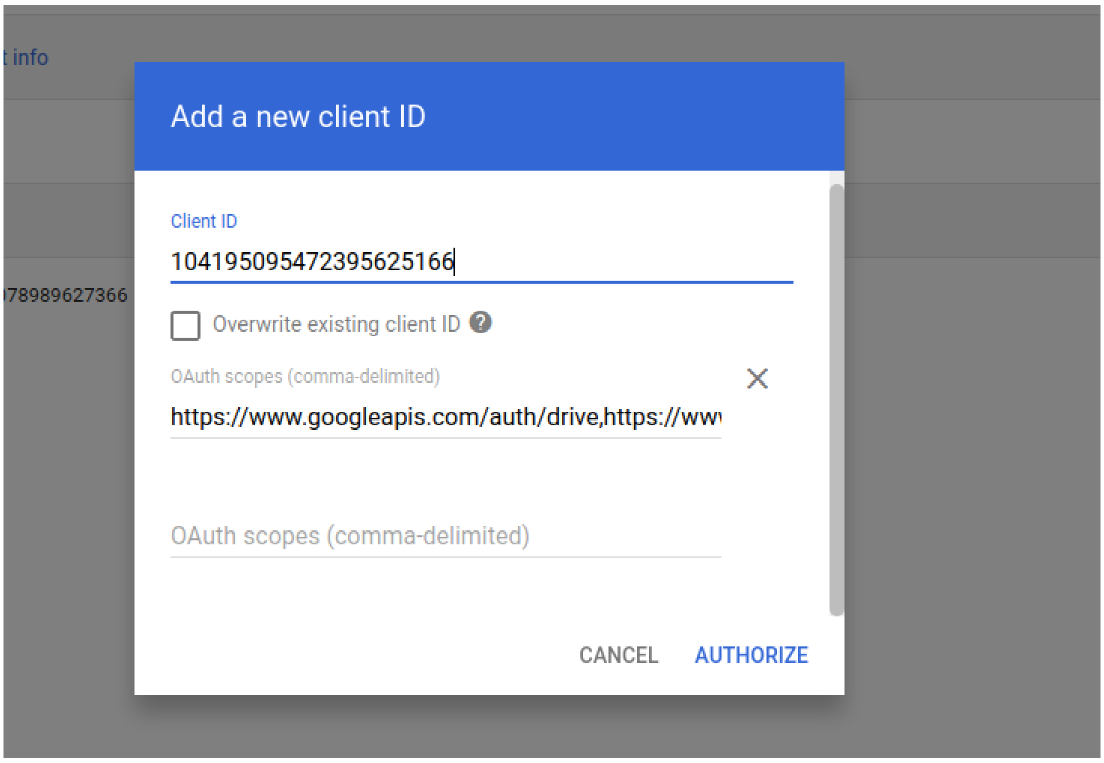
DISABLE SERVICE ACCOUNT

Advanced settings

Fig. 94: Google Workspace Service account id

- https://www.googleapis.com/auth/drive
- https://www.googleapis.com/auth/admin.directory.user.security
- https://www.googleapis.com/auth/admin.directory.user
- https://www.googleapis.com/auth/drive.appdata
- https://mail.google.com/
- https://www.googleapis.com/auth/gmail.settings.basic
- https://www.googleapis.com/auth/gmail.settings.sharing

Once everything is put in the form:


Add a new client ID

Client ID: 104195095472395625166

Overwrite existing client ID

OAuth scopes (comma-delimited): https://www.googleapis.com/auth/drive,https://ww

Fig. 95: Google Workspace Service account id completed

We click on authorize and congratulations! You should have successfully prepared your environment to use Bacula Enterprise Google Workspace Plugin.

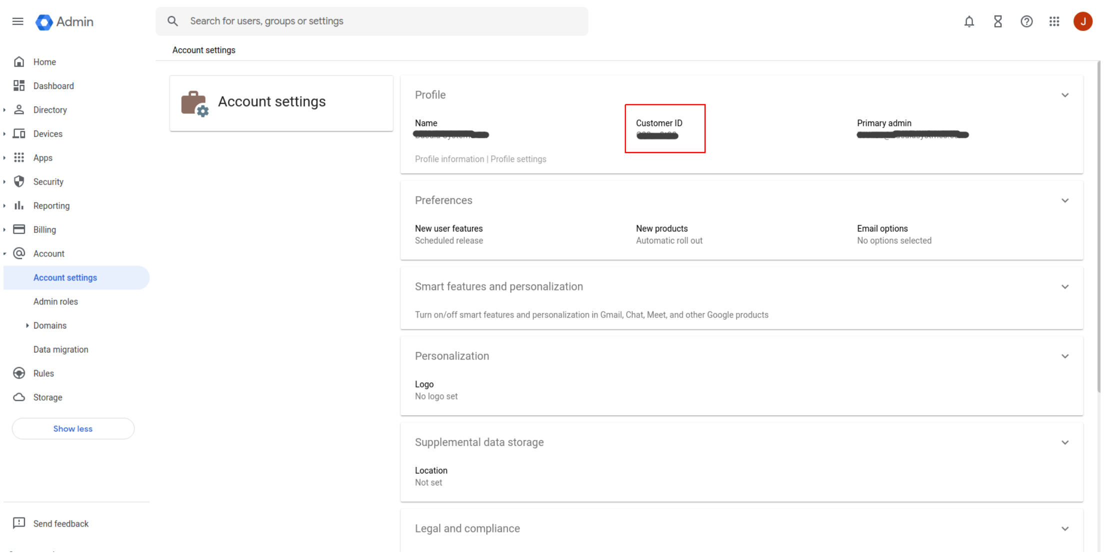
Customer Id and Admin User Email

Note: Workspace only

This step is only needed for Workspace environments.

In order to use this plugin, in addition to a credentials file pointing to a properly configured project and workspace, it is needed to specify the customer id, as well as an admin user email.

We can find customer id in the 'Google Admin Workspace' console. Just go to 'Account > Account settings'.


Admin

Search for users, groups or settings

Account settings

Home
Dashboard
Directory
Devices
Apps
Security
Reporting
Billing
Account
Account settings
Admin roles
Domains
Data migration
Rules
Storage
Show less

Send feedback

Account settings

Profile

Name [redacted]

Customer ID [redacted]

Primary admin [redacted]

Profile information | Profile settings

Preferences

New user features
Scheduled release

New products
Automatic roll out

Email options
No options selected

Smart features and personalization

Turn on/off smart features and personalization in Gmail, Chat, Meet, and other Google products

Personalization

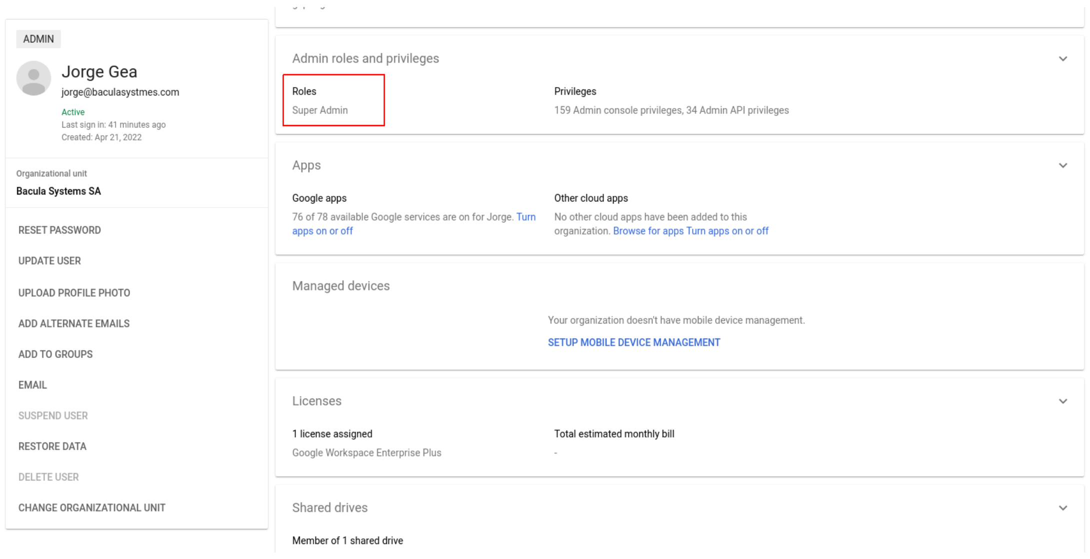
Logo
No logo set

Supplemental data storage

Location
Not set

Legal and compliance

For the email address of an admin user, we can use the same console, but going to 'Directory > Users'. Clicking on users, we can see the roles and privileges they have. We need to get the email of a user having the role of 'Super Admin' as the image below is showing:


ADMIN

Jorge Gea
jorge@baculasystmes.com
Active
Last sign in: 41 minutes ago
Created: Apr 21, 2022

Organizational unit
Bacula Systems SA

RESET PASSWORD
UPDATE USER
UPLOAD PROFILE PHOTO
ADD ALTERNATE EMAILS
ADD TO GROUPS
EMAIL
SUSPEND USER
RESTORE DATA
DELETE USER
CHANGE ORGANIZATIONAL UNIT

Admin roles and privileges

Roles
Super Admin

Privileges
159 Admin console privileges, 34 Admin API privileges

Apps

Google apps
76 of 78 available Google services are on for Jorge. Turn apps on or off

Other cloud apps
No other cloud apps have been added to this organization. Browse for apps Turn apps on or off

Managed devices

Your organization doesn't have mobile device management.
SETUP MOBILE DEVICE MANAGEMENT

Licenses

1 license assigned
Google Workspace Enterprise Plus

Total estimated monthly bill
-

Shared drives

Member of 1 shared drive

Fig. 96: Google Workspace Super Admin User

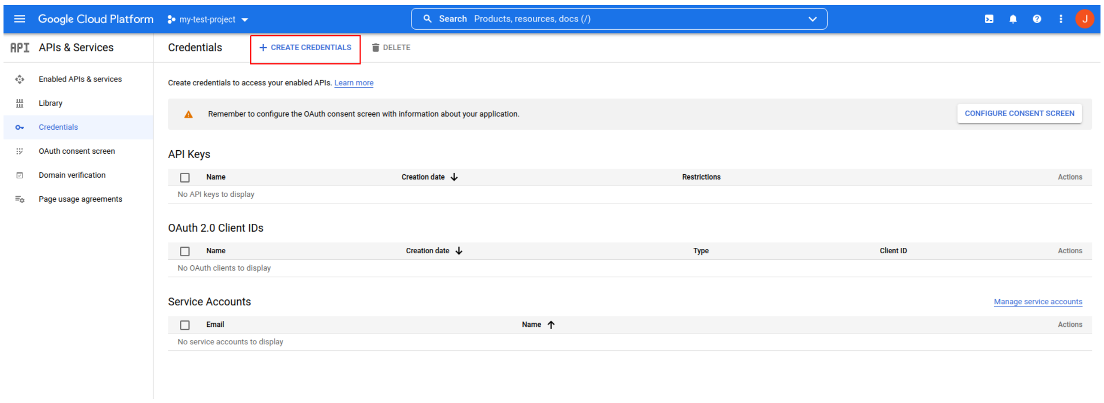
Then we need to be use the email associated to that user that is shown in the same screen just below the user name.

Key Credentials for Free Accounts

Note: Free users only

This step is only needed for Free Gmail users

We need to go to API & Services > credentials. From this section we will create a OAuth type credentials. We need to click in 'Create credentials' as the image shows:



Create credentials to access your enabled APIs. Learn more

Remember to configure the OAuth consent screen with information about your application.

API Keys

Name	Creation date ↓	Restrictions	Actions
No API keys to display			

OAuth 2.0 Client IDs

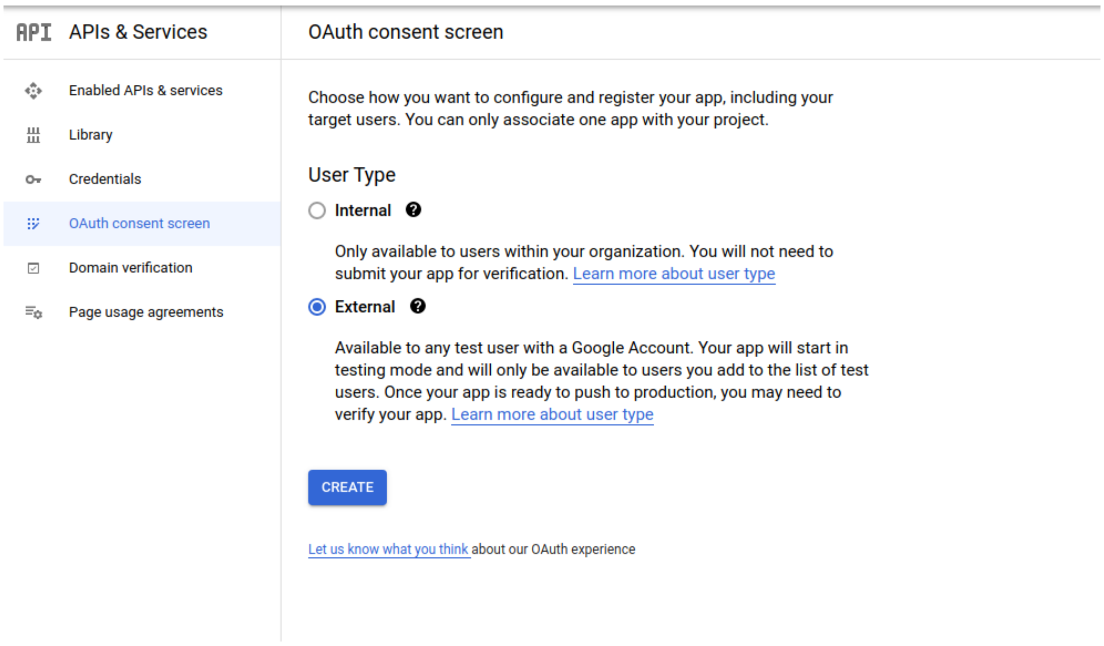
Name	Creation date ↓	Type	Client ID	Actions
No OAuth clients to display				

Service Accounts

Email	Name ↑	Actions
No service accounts to display		

As this is the first time, we will be redirected to configure the OAuth consent screen.

We just need to select 'External' and then put a name and our email as the following images show:



OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

Internal

Only available to users within your organization. You will not need to submit your app for verification. Learn more about user type

[✓ External]

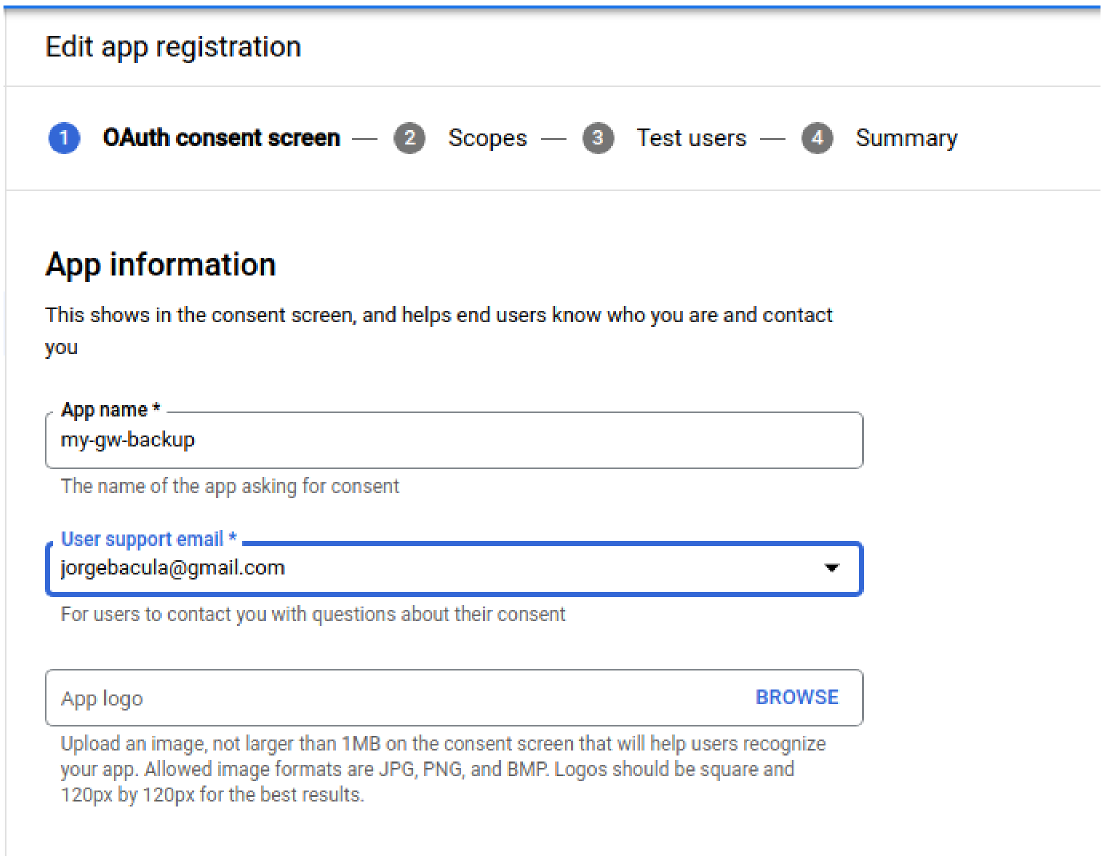
Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. Learn more about user type

Let us know what you think about our OAuth experience

In the next step we need to add any user that we want to be allowed to be backed up:

Now we are ready to continue with the OAuth type credentials, so we go again with the 'Create credentials' button we saw in the first step, where we need to select 'Desktop application' and put a name:

The OAuth Id will be generated and we need to click on the download json button the image shows:


Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

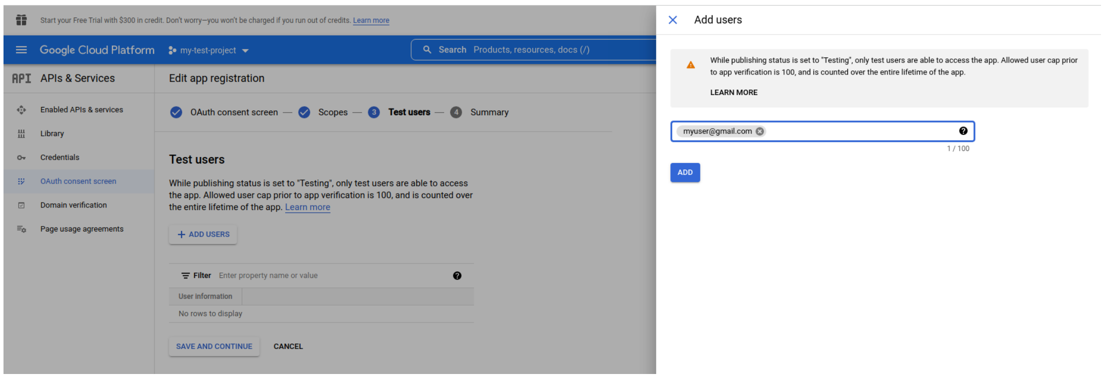
App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *
my-gw-backup
The name of the app asking for consent

User support email *
jorgebacula@gmail.com
For users to contact you with questions about their consent

App logo BROWSE
Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.


Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

Google Cloud Platform · my-test-project · Search Products, resources, docs (/)

APIs & Services: Enabled APIs & services, Library, Credentials, OAuth consent screen, Domain verification, Page usage agreements

Edit app registration

OAuth consent screen — Scopes — 3 Test users — 4 Summary

Test users

While publishing status is set to "Testing", only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more

+ ADD USERS

Filter Enter property name or value

User information
No rows to display

SAVE AND CONTINUE · CANCEL

Add users

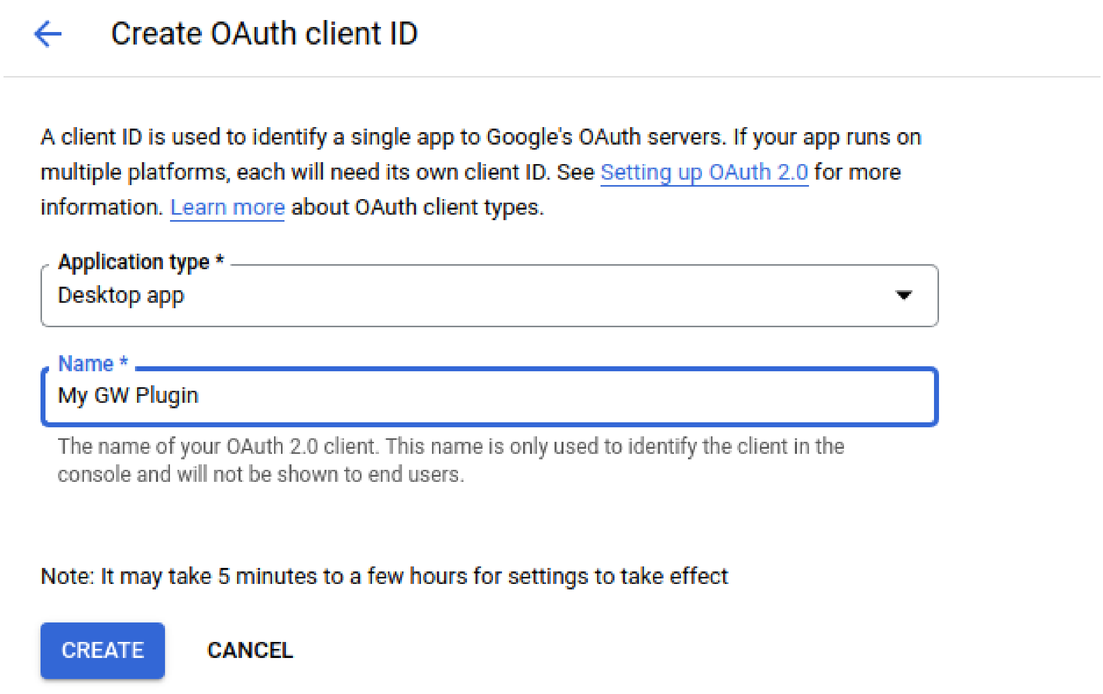
While publishing status is set to "Testing", only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app.

LEARN MORE

myuser@gmail.com

1 / 100

ADD



Create OAuth client ID

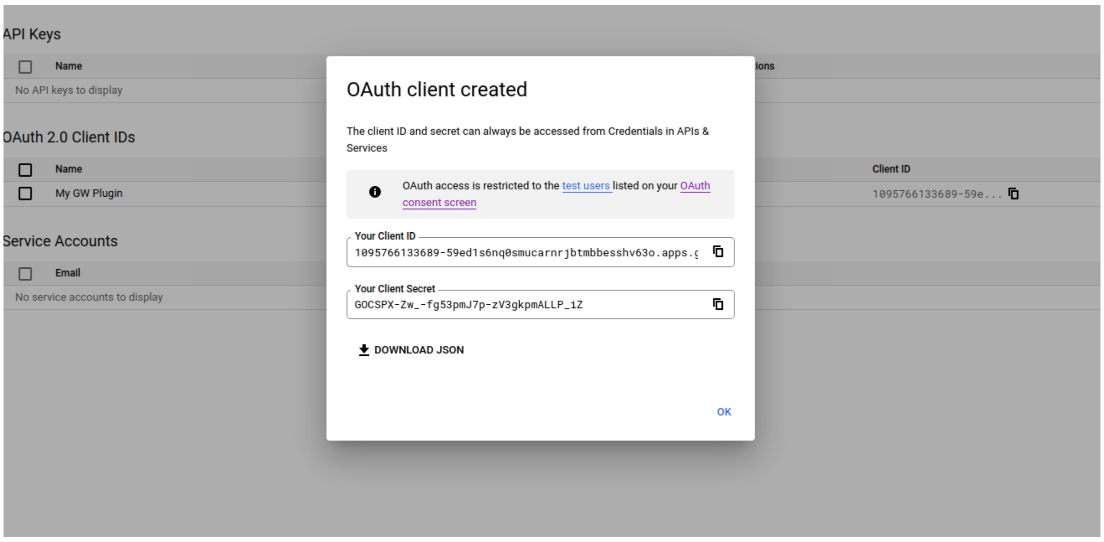
A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

Application type *: Desktop app

Name *: My GW Plugin

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Note: It may take 5 minutes to a few hours for settings to take effect



API Keys

	Name
	No API keys to display

OAuth 2.0 Client IDs

	Name	Client ID
	My GW Plugin	1095766133689-59e...

Service Accounts

	Email
	No service accounts to display

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to the test users listed on your OAuth consent screen

Your Client ID: 1095766133689-59ed1s6nq0smucarnrjbtmbbesshv63o.apps.ç

Your Client Secret: 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

That JSON downloaded file is the file we need to refer to in the ‘credentials_file’ fileset parameter in order for the plugin to authenticate our user. The first time we use the plugin with it it will ask to open a URL so we can confirm the permissions the plugin needs to perform the backup in our account, so it is needed to open that URL, to login with the proper user, select all the permissions shown and accept them.

Log example:

> time: 2022-06-13 17:44:14 logtext: 127.0.0.1-fd JobId 4: gw: Please, open the following address in your browser and login with ‘jorgebacula@gmail.com’
>
> time: 2022-06-13 17:44:14 logtext: 127.0.0.1-fd JobId 4: gw: https://accounts.google.com/o/oauth2/auth?access_type=offline&client_id=105945873553-e4bu9nipqljhbh1edgjtgueleqe4do7r.apps.googleusercontent.com&redirect_uri=http://localhost:8888/Callback&response_type=code&scope=https://www.googleapis.com/auth/drive%20https://www.googleapis.com/auth/drive.appdata%200https://mail.google.com/%20https://www.googleapis.com/auth/gmail.settings.basic

Once we open the URL and login with the proper user, we need to select all the items as the image shows and click on ‘Continue’:

The job will automatically continue its execution after that.

It is important to note that the service expecting to receive the result of this interaction will automatically listen on port 8888, this port can be adjusted using the plugin parameter ‘auth_port’ if needed.

The credentials will be stored by default into the ‘tokens’ path inside the ‘path’ directory. This can be changed using the ‘tokens_path’ variable. Those persistent credentials will avoid performing the authentication for every job execution.

Permissions for Shared Drives

The shared drives backup feature needs proper permissions configured. The user configured in the admin_user_email parameter, that needs to point to a user with ‘Super Admin’ role, needs also to be added as a member of the Shared Drives included in the backup.

If only backup access is required, the ‘Viewer’ role is enough. If restore ability is desired over the target Shared Drive, then ‘Content Manager’ or ‘Manager’ roles are necessary.

In order to add users to an existing Shared Drive, a ‘Manager’ user of it needs to access to its drive unit through the web browser and right-click on the target unit. There, the ‘Manage members’ option allows to add new users, so it is necessary to add the admin_user_email used for backup there.

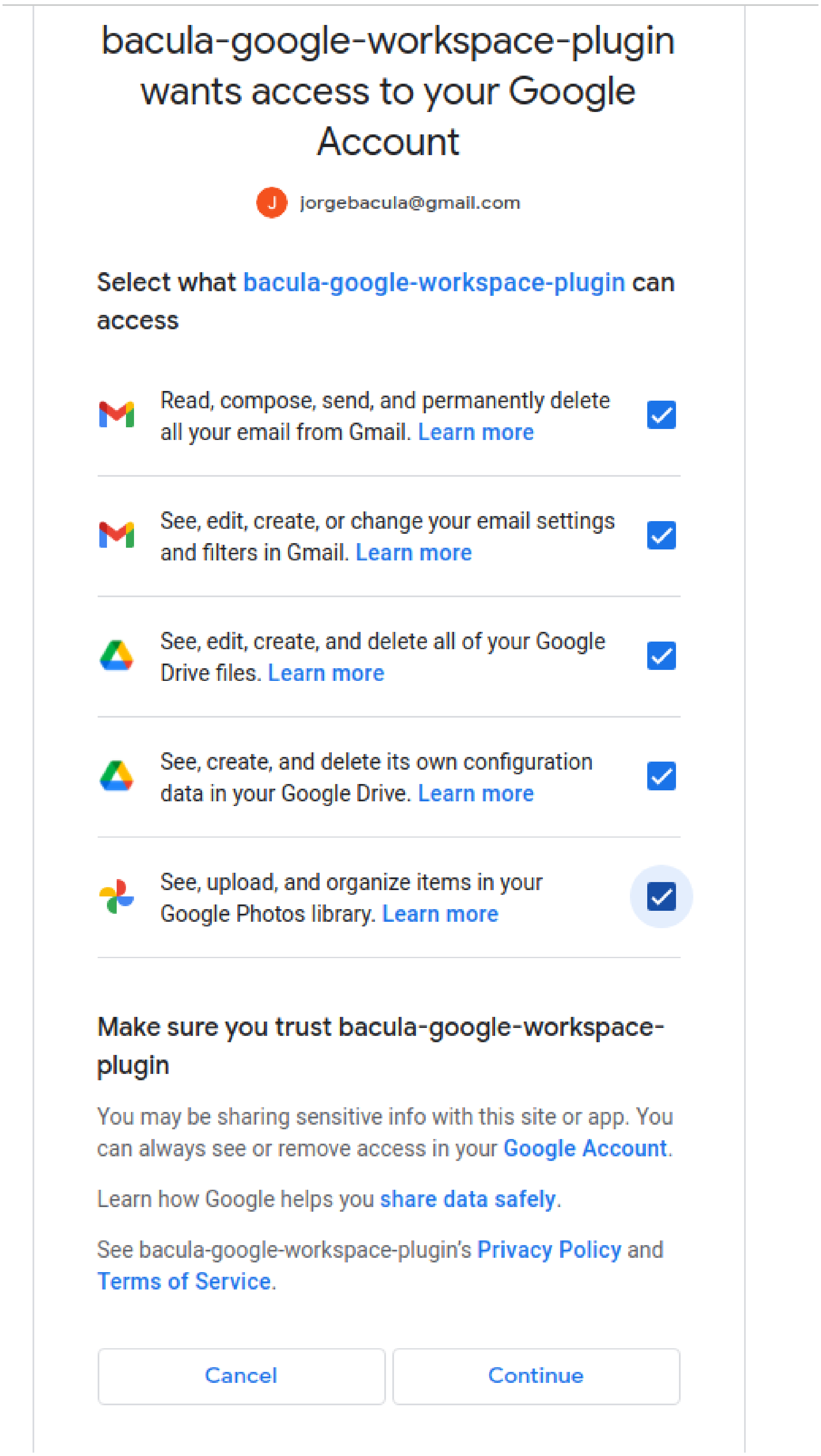
Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to call the Google Workspace Plugin:

Listing 233: **Fileset GW**

```
Fileset {
   Name = FS_GW
   Include {
      Options {
        signature = MD5
```

(continues on next page)



bacula-google-workspace-plugin wants access to your Google Account

J jorgebacula@gmail.com

Select what bacula-google-workspace-plugin can access

Read, compose, send, and permanently delete all your email from Gmail. Learn more ✓

See, edit, create, or change your email settings and filters in Gmail. Learn more ✓

See, edit, create, and delete all of your Google Drive files. Learn more ✓

See, create, and delete its own configuration data in your Google Drive. Learn more ✓

See, upload, and organize items in your Google Photos library. Learn more ✓

Make sure you trust bacula-google-workspace-plugin

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

Learn how Google helps you share data safely.

See bacula-google-workspace-plugin's Privacy Policy and Terms of Service.

(continued from previous page)

```
        ...
      }
      Plugin = "gw: <gw-parameter-1>=<gw-value-1> <gw-parameter-2>=<gw-value-
↪2> ..."
   }
}
```

It is **strongly recommended** to use only one 'Plugin' line in every fileset. The plugin offers the needed flexibility to combine different modules or entities to backup inside the same plugin line. Different workspaces, in case of existing, should be using different filesets and different jobs.

Below sub-sections list all the parameters you can use to control GW Plugin behavior.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ','.

Common Parameters

These parameters are common and applicable to all the modules of the Google Workspace Plugin.

Option	Require	Default	Values	Example	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacul	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset. This is specially useful for shared data between filesets and/or sensitive data as customer_id.
log	No	/opt/ba debug.	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/gw.lo	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/gw/gw-debug.log* files containing debug information which is more verbose with a greater debug number
path	No	/opt/ba	An existing path with enough permissions for File Daemon to create any internal plugin file	/mnt/myvol/	Uses this path to store metadata, plugin internal information and temporary files
custome	No		String representing the customer id associated to the Google Workspace subscription	Cbdi2930	The **customer id** associated to the Google Workspace subscription to be backed up. Please, check the authentication section of this document for more detailed information. Note that this is mandatory if you want to protect a workspace environment, but not needed to protect gmail free accounts.
admin_	No		A valid email address of one admin user of the Google Workspace subscription	rafael@cu	The email address of an admin user of the Google Workspace subscription to be protected. Please, check the authentication section of this document for more detailed information. Note that this is mandatory if you want to protect a workspace environment, but not needed to protect gmail free accounts.
credentials_	**Yes**		The path of the file where credentials are stored	/opt/bacul	The path of the file downloaded from the configured Google Cloud application that will act as a bridge in order to allow the communication between this plugin and Google Workspace. Please, check the authentication section of this document for more detailed information.
tokens_	No*	tokens	A path with enough permissions so File Daemon can write in it	/home/use	The path that will be used to store the login cache for the device code flow authenticated users, which is relative to the path folder folder (usually working/gw/customer_id/tokens_path/). This is

The plugin supports two different kind of users: Workspace users and free Gmail users.

For Workspace users, in addition to ‘credentials_file’, the following parameters are mandatory: ‘customer_id’ and ‘admin_user_email’.

For free Gmail users those parameters are not used, but it is possible to customize ‘tokens_path’ and ‘auth_port’.

Advanced Common Parameters

Following parameters are common to all Google Workspace modules (and even with some other plugins), but are advanced ones. They should not be modified in most common use cases.

Option	Require	Default	Values	Example	Description
stream_sleep	No	1	Positive integer (1/10 seconds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_max_	No	120	Positive integer (seconds)	360	Max wait time for FD to answer packet requests
time_max_las	No	86400	Positive integer (seconds)	4320(	Maximum time to wait to overwrite a debug log that was marked as being used by other process
logging_max_file	No	50MB	String size	300M	Maximum size of a single debug log file
logging_max_ba	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rolling_fil	No	gw.log.%d{dd MMM}.log.g:	No, Yes	Yes	Log patter for rotated log files
split_config_fi	No	=	Character	:	Character to be used in config_file parameter as separator for keys and values
opener_queue	No	1200	Positive integer (seconds)	3600	Timeout when internal object opener queue is full
publisher_queue_	No	1200	Positive integer (seconds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The “.log” files are rotated automatically. Currently each file can be 50Mb at maximum and the plugin will keep 25 files.
  - This behavior can be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The “.err” file can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like ‘logrotate’.
- Backups in parallel and also failed backups will generate several log files. For example: gw-debug-0.log, gw-debug-1.log…

Tuning Parameters

These set of parameters are common to all modules and they are advanced ones. They should not be modified in general. They can be used to tune the behavior of the plugin to be more flexible in particular bad network environments or when significant job concurrency is happening, etc.

Option	Require	Default	Values	Example	Description
back	No	30	0-50	1	Number of maximum en-queued internal operations between service static internal threads (there are 3 communicating through queues with the set size: service fetcher, service opener and general publisher to bacula core). This could potentially affect google api concurrent requests and consequently, Google throttling. It is only needed to modify this parameter, in general, if you are going to run different jobs in parallel
concurrent_	No	4	0-10	1	Number of maximum concurrent backup threads running in parallel in order to fetch or open data for running download actions. This means every service fetcher and service opener will open this number of child concurrent threads. This will affect google api concurrent requests. Google API can throttle requests depending on a variety of circumstances, but it is directly attached . It is only needed to modify this parameter, in general, if you are going to run different jobs in parallel. If you want to have a precise control of your concurrency through different jobs, please set up this value to 1. Please be careful also with the memory requirements, multi-threaded increases very significantly memory consumption per job
api_l	No	500	1-500	350	Number of maximum elements got from Google API for each page of objects. Higher number implies less requests, but more memory and more time for each request
api_t	No	9000	Positive integer (millisec-onds)	6000	Google call timeout inside HttpClient
api_r	No	300	Positive integer (millisec-onds)	3000	Google read timeout inside HttpClient
api_r	No	5	Positive integer (number of retries)	10	Google number of retries for retry-candidate requests
api_r	No	5	Positive integer (seconds)	10	Google API delay between retries
general	No	5	Positive integer (number of re-	10	Number of retries for the general external retry mechanism

Entity Parameters

The following list of parameters are commonly shared through any module used in the same fileset line and are intended to select the target entities to backup. Every module subsection mentions what entities are supported too.

Option	Require	Default	Values	Example	Services	Description
user	No		Valid email addresses of existing users on the selected workspace separated by ','	AlexW@you LeeY@your	drive email	Backup selected services of this list of users. If no user is provided, and no other user parameter is set, all users will be discovered and included in the backup
user_	No		Valid email addresses of existing users on the selected workspace separated by ','	LauraG@you Amanda T@yourdc	drive email	Exclude selected services of selected users If this is the only parameter found for selection, all elements will be included and this list will be excluded
user_	No		Valid regex	.*@management\.mydor	drive email	Backup selected services of matching users.
user_	No		Valid regex	.*@guests\.n	drive email	Exclude selected services of matching users. If this is the only parameter found for selection, all elements will be included and this list will be excluded
user_	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	drive email	Includ or exclude users that have been suspended and, therefore, are not active. By default, these are excluded from any backup, as the contents cannot be regularly accessed unless the user is reactivated

Backup Parameters

Please, check the specific module pages in order to see backup parameters that are applicable only to each of them:

- Google Drive
- Google Email

Restore Parameters

The plugin is able to restore to the local file system on the server where the File Daemon is running or to the Google Workspace environment. The method is selected based on the value of the *where* parameter at restore time:

- Empty or '/' (example: where=/) $\rightarrow$ Google Workspace restore will be triggered
- Any other path for where (example: where=/tmp) $\rightarrow$ Local file system restore will be triggered

When using Google Workspace restore option, the following parameters may be modified by selecting 'Plugin Options' during the bconsole restore session:

Option	Require	Default	Values	Example	Services	Description
destination_	No		Existing email address on the target Google Workspace	Alex\	drive email	Destination User where restore data will be uploaded. If no user is set, every selected file will be restored in the original account
destination_	No		Destination path to be created (or existing) into the selected user (drive folder path)	Re-store-Folde	drive email	Destination folder where all selected files to restore will be restored. If no path is set: - If no user is set either, every element will go to its original location - If a user is set using the variable **destination_user**: - Elements belonging to destination_user will be restored in their original location - Elements belonging to different users than destination_user will be restored in a new folder using the email address of the original user of the element
send_	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	1	drive email	Send a report to the user where every restore action is listed. - In drive service this will generate a new text file in the top restore folder
allow_	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	drive email	Set if we allow to have several files with the same name in the same path or not (if not, we can overwrite the file using the 'Replace' general restore variable)
drive	No		Existing shared drive name	MySl Drive	drive	Destination drive shared unit where restored data will be uploaded. If no drive is set, every selected file will be restored in the original shared drive
drive	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	drive	Skip restoring former file versions (tagged with '###date') even if they are selected. **Important**: Notice that this parameter is **enabled by default**, as we consider not restoring file versions the most common case. You need to disable it in order to have this kind of files restored
drive	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	drive	Skip restoring file comments (located inside the 'filename_comments' folder) even if they are selected. **Important**: Notice that this parameter is **enabled by default**, as we consider not restoring file comments the most common case. You need to disable it in order to have this kind of information restored
drive	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	1	drive	Skip restoring shared with me elements even if they are selected.
drive	No	0	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	1	drive	Restore share permissions of every element in order to regenerate sharing information as allowed identities, shared links, etc. **Important**: Notice that this parameter is **disabled by default**, as we consider not restoring sharing permissions the most common case. You need to enable it in order to have shared permissions restored
email	No	0	0, no, No, false, FALSE, false, off ;	1	email	Export selected emails to MIME format in local filesystem (RFC 822)

Operations

Backup

Google Workspace plugin backup configurations currently have just one specific requirement in the Job resource. Below we show some examples.

Job Example

The only special requirement with Google Workspace jobs is that Accurate mode backups must be disabled, as this feature is not supported at this time.

Listing 234: **Job Example**

```
Job {
Name = gw-myworkspace-backup
Fileset = fs-gw-drive-all
Accurate = no
...
}
```

Fileset Examples

The plugin supports enough flexibility to configure almost any type of desired backup. Multiple *Plugin=* lines should not be specified in the Include section of a Fileset for the Google Workspace Plugin.

Fileset examples for every supported service are linked below. For common purposes, the following two examples show how to configure an external config file or configure the number of threads:

Setup external config file:

Listing 235: **Fileset Example**

```
Fileset {
   Name = FS_GW_DRIVE
   Include {
      Options {
        signature = MD5
      }
      Plugin = "gw: config_file=/opt/bacula/etc/gw.settings service=drive"
   }
}
```

Listing 236: **Settings file**

```
$ cat /opt/bacula/etc/gw.settings
```

Increase number of threads:

Listing 237: **Fileset Example**

```
Fileset {
   Name = fs-gw-drive-kara
   Include {
      Options {
        signature = MD5
      }
      Plugin = "gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-sa-
↪credentials.json
      customer_id=\"B01ua5i29\" admin_user_email=\"peter@baculasystems.com\
↪"service=drive
       user=kara@baculasystems.com concurrent_threads=10"
   }
}
```

More fileset examples for:

- Google Drive
- Google Mail

Restore

Restore operations are done using standard **Bacula Enterprise** bconsole commands.

The *where* parameter controls if the restore will be done locally to the File Daemon's file system or to the Google Workspace service:

- where=/ or empty value → Restore will be done over Google Workspace
- where=/any/other/path → Restore will be done locally to the File Daemon file system

Restore options are described in the restore-params section of this document, so here we are going to simply show an example restore session:

Listing 238: **Restore Drive Bconsole Session**

```
*restore where=/

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
```

(continues on next page)

(continued from previous page)

```
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Select object to restore
    14: Cancel
Select item:  (1-14): 5
Automatically selected Client: 127.0.0.1-fd
Automatically selected Fileset: FS_GW
+-------+-------+----------+----------+---------------------+-----------------
↪--+
| jobid | level | jobfiles | jobbytes | starttime           | volumename     ␣
↪  |
+-------+-------+----------+----------+---------------------+-----------------
↪--+
|     1 | F     |       29 |  125,994 | 2022-05-12 17:49:27 | TEST-2022-05-
↪12:0 |
+-------+-------+----------+----------+---------------------+-----------------
↪--+
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
27 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd "/@gw/C02uv9t30/users/jorge@baculasystmes.com/drive/my drive/"
cwd is: /@gw/C02uv9t30/users/jorge@baculasystmes.com/drive/my drive/
$ ls
REGRESS_20220512174729/
sharedWithMe/
$ cd REGRESS_20220512174729/
cwd is: /@gw/C02uv9t30/users/jorge@baculasystmes.com/drive/my drive/REGRESS_
↪20220512174729/
$ ls
Elitr.mp4
Elitr.mp4__comments/
Graeco.docx
Graeco.docx__comments/
Interpretaris/
Mnesarchum.ppt
Scelerisque.jpeg
Vivamus.doc
Vivamus.doc__comments/
$ mark *
20 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
```

(continues on next page)

(continued from previous page)

```
The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    TEST-2022-05-12:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


20 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2022-05-12 18:03:23
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : gw: credentials_file="/home/jorge/projects/bacula-gw-
↪plugin-sa-2.json" customer_id="C02uv9t30" admin_user_email=
↪"jorge@baculasystmes.com" service="drive" user="jorge@baculasystmes.com"␣
↪drive_files="REGRESS_20220512174729" drive_shared_units_regex_exclude=".*"␣
↪debug=6
Plugin Restore Options
Option                  Current Value        Default Value
destination_user:       *None*               (*None*)
destination_path:       *None*               (*None*)
send_report:            *None*               (0)
```

(continues on next page)

(continued from previous page)

```
allow_duplicates:    *None*              (1)
drive_destination_shared_unit: *None*               (*None*)
drive_skip_versions: *None*              (1)
drive_skip_comments: *None*              (1)
drive_skip_sharedwithme: *None*              (0)
drive_restore_share_permissions: *None*              (0)
customer_id:         *None*              (*None*)
admin_user_email:    *None*              (*None*)
credentials_file:    *None*              (*None*)
tokens_path:         *None*              (*None*)
auth_port:           *None*              (*None*)
foreign_container_generation: *None*              (1)
debug:               *None*              (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
     1: destination_user (Destination User)
     2: destination_path (Destination Path in google-workspace)
     3: send_report (Send report of the restore operation to the affected␣
↪user)
     4: allow_duplicates (Allow Duplicate Objects (Files with the same name␣
↪in the same folder, emails with same id..))
     5: drive_destination_shared_unit (Destination Shared Unit name)
     6: drive_skip_versions (Skip restoring file former versions (tagged with
↪'###date') even if they are selected)
     7: drive_skip_comments (Skip restoring file comments even if they are␣
↪selected)
     8: drive_skip_sharedwithme (Skip restoring shared with me elements even␣
↪if they are selected)
     9: drive_restore_share_permissions (Restore sharing permissions of the␣
↪files, so they get shared with the same people than original files)
    10: customer_id (Destination Workspace customer id)
    11: admin_user_email (Destination Workspace admin user email)
    12: credentials_file (Credentials file path to be used for authentication)
    13: tokens_path (Directory to store authorization tokens for delegated␣
↪permissions)
    14: auth_port (Port to receive response from delegated authentication␣
↪process)
    15: foreign_container_generation (Generate a general container (usually a␣
↪folder) to put inside restored objects coming from different entities)
    16: debug (Change debug level)
Select parameter to modify (1-16): 2
Please enter a value for destination_path: restored1
Plugin Restore Options
Option                 Current Value       Default Value
destination_user:    *None*              (*None*)
destination_path:    restored1           (*None*)
send_report:         *None*              (0)
allow_duplicates:    *None*              (1)
drive_destination_shared_unit: *None*               (*None*)
drive_skip_versions: *None*              (1)
drive_skip_comments: *None*              (1)
drive_skip_sharedwithme: *None*              (0)
```

(continues on next page)

(continued from previous page)

```
drive_restore_share_permissions: *None*              (0)
customer_id:          *None*              (*None*)
admin_user_email:     *None*              (*None*)
credentials_file:     *None*              (*None*)
tokens_path:          *None*              (*None*)
auth_port:            *None*              (*None*)
foreign_container_generation: *None*              (1)
debug:                *None*              (*None*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2022-05-12 18:03:23
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
Job queued. JobId=3
```

Restore Single Email and Read It using any Email App

1. Identify the restored file

- The restored emails will be named in the following format: `xxxxxx.msg` and `xxxxxx.msg.metadata_catalog`.
- The `.msg` file contains the email content encoded in Base64 format.

2. Extract the Email Content

- Go to the directory where the restored file is located: `cd /path/to/your/file`
- Extract the Base64 content from the “raw” field: `grep '"raw"' filename.msg | sed -E 's/.*"raw":"([^"]+)".*/\1/' > email_base64.txt`
- Decode the Base64 content to create an email file: `base64 -d email_base64.txt > decoded_email.eml`

3. Open the Email in Thunderbird

    - Launch Thunderbird.
    - Go to File > Open > Saved Message…
    - Select the `decoded_email.eml` file you have just created.

   The email will open just like any other message.

Restore by Service

In this section some example restore configurations will be shown:

- Google Drive
- Google Drive

Cross Workspace Restore

You can perform cross-workspace restores using the restore variables:

- customer_id
- admin_user_email
- credentials_file

Obviously, it is needed to set up the destination workspace values, where a connection application should have been also set up previously to allow the connection.

List

It is possible to list information using the bconsole *.ls* command and providing a path or the *.query* command and providing a parameter.

In general, we need to provide the service parameter, the implied entity and a path representing a folder. However, there are some general commands (like listing users or shared drive units), while the rest of the commands need to have the service set

Below some examples:

List General Info: Users and Shared Drives of a Workspace

Here we are showing some commands using the bconsole *.ls* command and the *.query* command.

Listing 239: **List/query examples: Users and shared drives**

```
*.ls plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com" client=127.0.0.1-fd path=users
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
-rw-r-----   1 nobody   nogroup                  -1 1970-01-01 00:59:59  /
↪jorge@baculasystems.com
-rw-r-----   1 nobody   nogroup                  -1 1970-01-01 00:59:59  /
↪kara@baculasystems.com
-rw-r-----   1 nobody   nogroup                  -1 1970-01-01 00:59:59  /
↪john@baculasystems.com
2000 OK estimate files=3 bytes=0

*.query plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com" client=127.0.0.1-fd parameter=users
user=jorge@baculasystems.com
```

(continues on next page)

(continued from previous page)

```
id=98984561467687684
suspended=false
mailboxSetup=true
user=lickaraense@baculasystems.com
id=81023135434315334
suspended=false
mailboxSetup=true

*.query plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com user_suspended=true" client=127.0.0.1-fd␣
↪parameter=users
user=jorge@baculasystems.com
id=98984561467687684
suspended=false
mailboxSetup=true
user=lickaraense@baculasystems.com
id=81023135434315334
suspended=false
mailboxSetup=true
user=olduser@baculasystems.com
id=19457845789489343
suspended=true
mailboxSetup=true

*.query client=127.0.0.1-fd plugin="gw: credentials_file=/opt/bacula/etc/
↪bacula-gw-plugin-credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com" parameter=drive_shared_units
drive_shared_units=CustomersSharedDrive
id=0AHv03jdf834idjUk9PVA
drive_shared_units=DocsShared
id=0Aolck348764bn9uUk9PVA
```

List Google Drive Contents

Listing 240: **List example: General information**

```
*.ls plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com user=jorge@baculasystems.com service=drive"␣
↪client=127.0.0.1-fd path=/
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
-rw-r-----   1 nobody   nogroup             373568 2022-05-13 11:24:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-13_11.24.33_06.html
-rw-r-----   1 nobody   nogroup             372115 2022-05-12 18:03:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-12_18.03.36_10.html
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-12 18:03:41  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/restored1/
```

(continues on next page)

(continued from previous page)

```
-rw-r-----   1 nobody   nogroup              373602 2022-05-12 17:49:52  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-12_17.49.34_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-12 17:49:39  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220512174934/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-12 17:48:25  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220512174729/
-rw-r-----   1 nobody   nogroup              372471 2022-05-12 10:47:16  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-12_10.47.03_10.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-12 10:47:08  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/RESTORED_SKIPVER_
↪REGRESS_20220512104703/
-rw-r-----   1 nobody   nogroup              376389 2022-05-12 10:46:50  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-12_10.46.32_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-12 10:44:35  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/SOURCE_REGRESS_
↪20220512104335/
-rw-r-----   1 nobody   nogroup              376609 2022-05-10 12:56:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.56.12_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:56:17  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510125612/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:55:31  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/SRC_INCLUDE_REGRESS_
↪20220510125529/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:54:57  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/trash/SRC_REMOVE_
↪REGRESS_20220510125403/
-rw-r-----   1 nobody   nogroup              373736 2022-05-10 12:49:34  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.49.18_10.html
-rw-r-----   1 nobody   nogroup              376810 2022-05-10 12:49:05  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.48.42_06.html
-rw-r-----   1 nobody   nogroup              374586 2022-05-10 12:31:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.31.28_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:31:33  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510123128/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:30:10  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/trash/SRC_REMOVE_
↪REGRESS_20220510123006/
-rw-r-----   1 nobody   nogroup              372465 2022-05-10 12:29:46  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.29.31_10.html
-rw-r-----   1 nobody   nogroup              376395 2022-05-10 12:29:19  /@gw/
```

(continues on next page)

(continued from previous page)

```
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.28.58_06.html
-rw-r-----   1 nobody   nogroup              372210 2022-05-10 12:25:23  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_12.25.04_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:25:09  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510122504/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 12:23:48  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510122254/
-rw-r-----   1 nobody   nogroup              372218 2022-05-10 11:38:20  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_11.38.04_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 11:38:09  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510113804/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-10 11:36:53  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220510113557/
-rw-r-----   1 nobody   nogroup              372210 2022-05-10 11:34:28  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_11.34.11_06.html
-rw-r-----   1 nobody   nogroup              372196 2022-05-10 11:27:03  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_11.26.47_06.html
-rw-r-----   1 nobody   nogroup              373412 2022-05-10 11:23:53  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_11.23.39_06.html
-rw-r-----   1 nobody   nogroup              373574 2022-05-10 11:21:10  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-10_11.20.53_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-09 18:26:39  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/DoComplicateMyLife/
-rw-r-----   1 nobody   nogroup              370347 2022-05-09 18:24:18  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-09_18.24.08_18.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-09 18:24:13  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/testingMyRestore/
-rw-r-----   1 nobody   nogroup              373586 2022-05-09 17:56:49  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-09_17.56.33_06.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-09 17:56:38  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220509175633/
-rw-r-----   1 nobody   nogroup              373701 2022-05-07 14:16:03  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_14.15.48_10.html
-rw-r-----   1 nobody   nogroup              376757 2022-05-07 14:15:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_14.15.15_06.html
-rw-r-----   1 nobody   nogroup              371504 2022-05-07 13:57:22  /@gw/
```

(continues on next page)

(continued from previous page)

```
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_13.56.39_13.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-07 13:56:44  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/ANADALRG14/
-rw-r-----   1 nobody   nogroup              371491 2022-05-07 13:21:14  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_13.20.31_12.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-07 13:20:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/ADjoker/
-rw-r-----   1 nobody   nogroup              382293 2022-05-07 12:35:15  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_12.34.32_11.html
-rw-r-----   1 nobody   nogroup               23352 2022-05-07 12:34:41  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/Brute.jpeg
-rw-r-----   1 nobody   nogroup               18394 2022-05-07 12:34:40  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/Ultricies.txt
-rw-r-----   1 nobody   nogroup                8693 2022-05-07 12:34:40  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/Suavitate.ppt
-rw-r-----   1 nobody   nogroup               14981 2022-05-07 12:34:40  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/Cetero.txt
-rw-r-----   1 nobody   nogroup               15752 2022-05-07 12:34:39  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/Fastidii.ppt
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-07 12:34:38  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AFullRestore1/
-rw-r-----   1 nobody   nogroup              370327 2022-05-07 12:15:58  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-07_12.15.48_10.html
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-07 12:15:54  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AverAlc/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-06 17:12:34  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220506171229/
drwxr-xr-x   1 nobody   nogroup                  -1 2022-05-06 16:58:13  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220506165808/
-rw-r-----   1 nobody   nogroup              372243 2022-05-06 13:49:21  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_13.49.06_06.html
-rw-r-----   1 nobody   nogroup              372215 2022-05-06 13:39:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_13.39.20_06.html
-rw-r-----   1 nobody   nogroup              372167 2022-05-06 13:13:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_13.13.33_06.html
-rw-r-----   1 nobody   nogroup              372444 2022-05-06 13:09:03  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_13.08.47_10.html
-rw-r-----   1 nobody   nogroup              376549 2022-05-06 13:08:34  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_13.08.11_06.html
-rw-r-----   1 nobody   nogroup              372396 2022-05-06 11:20:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
```

(continues on next page)

(continued from previous page)

```
↪DRIVE_2022-05-06_11.20.37_10.html
-rw-r-----   1 nobody   nogroup            376416 2022-05-06 11:20:24  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_11.20.06_06.html
-rw-r-----   1 nobody   nogroup            373532 2022-05-06 09:51:58  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-06_09.51.43_06.html
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:19:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AutoSimple 2022-05-
↪05 05.19.34/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:14:12  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AutoSimple 2022-05-
↪05 05.14.10/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:03:02  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:46  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:35  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:31  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:23  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:20  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:12  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:02:09  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 17:01:58  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505170105/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 16:59:34  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 16:59:22  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                -1 2022-05-05 16:59:18  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
```

(continues on next page)

(continued from previous page)

```
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:59:06  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:59:02  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:58:51  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:58:49  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:58:39  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:58:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:58:27  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220505165734/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:56:25  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AutoSimple 2022-05-
↪05 04.56.13/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-05 16:56:15  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/AutoSimple 2022-05-
↪05 04.56.13/
-rw-r-----   1 nobody   nogroup             373488 2022-05-05 13:55:46  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-05_13.55.30_06.html
-rw-r-----   1 nobody   nogroup             373499 2022-05-04 12:03:24  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-04_12.03.07_06.html
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-04 12:03:12  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220504120307/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-04 12:02:01  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220504120058/
-rw-r-----   1 nobody   nogroup             373510 2022-05-03 13:32:34  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-03_13.32.18_06.html
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-03 13:32:23  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220503133217/
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-03 13:31:04  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220503133010/
-rw-r-----   1 nobody   nogroup             371497 2022-05-03 13:26:48  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-03_13.26.36_06.html
drwxr-xr-x   1 nobody   nogroup                 -1 2022-05-03 13:26:41  /@gw/
```

(continues on next page)

(continued from previous page)

```
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/REGRESS_
↪20220503132636/
-rw-r-----    1 nobody    nogroup              372431 2022-05-02 17:53:43  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-02_17.53.29_10.html
-rw-r-----    1 nobody    nogroup              376394 2022-05-02 17:53:17  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-02_17.52.56_06.html
-rw-r-----    1 nobody    nogroup              373668 2022-05-02 13:59:36  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-02_13.59.19_10.html
-rw-r-----    1 nobody    nogroup              375148 2022-05-02 13:59:09  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-02_13.58.48_06.html
drwxr-xr-x    1 nobody    nogroup                  -1 2022-05-02 13:56:52  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/SOURCE_REGRESS_
↪20220502135650/
-rw-r-----    1 nobody    nogroup              372168 2022-05-02 13:39:07  /@gw/
↪C02uv9t30/users/jorge@baculasystems.com/drive/my drive/BEE_RestoreReport_
↪DRIVE_2022-05-02_13.38.51_06.html
2000 OK estimate files=100 bytes=15,775,509
```

Other Query/List Examples

Listing 241: **Query/List examples**

```
*.ls plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com user=jorge@baculasystems.com service=drive"␣
↪client=127.0.0.1-fd path=/folder1

List emails inside inbox
*.ls plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-
↪credentials.json customer_id=Alo9783c12 admin_user_
↪email=jorge@baculasystems.com user=jorge@baculasystems.com service=email"␣
↪client=127.0.0.1-fd path=/inbox

Show free users loggedin
*.query plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-free.
↪json" client=127.0.0.1-fd parameter=logged-users

Force login of a particular free user
*.query plugin="gw: credentials_file=/opt/bacula/etc/bacula-gw-plugin-free.
↪json" client=127.0.0.1-fd parameter=login:myuser@gmail.com
```

Best Practices

Jobs Distribution

It is recommended to split the target backup between different groups of entities or even having one job per entity (user, drive unit, etc). This way errors in one job will not invalidate a whole backup cycle where some entities have been successful and some others had errors. This also makes it easier to identify the cause of the error.

Concurrency

Google Workspace APIs impose a variety of boundaries that need to be considered. If a boundary is crossed, the corresponding API call will fail and the application will need to wait some amount of time to retry, which is different depending on the boundary crossed.

It is crucial to plan an adequate strategy to backup all the elements without reaching API boundaries. A single job implements some parallelism which can be reduced until a point, if necessary, using the variable **backup_queue_size** (default value is 30**).** This variable controls the size of the internal queues communicating the internal threads, that are designed to fetch, open and send every item to Bacula core. Reducing its size will produce, ultimately (with a value of 1 for example), an execution very similar to a single threaded process. On the other hand the plugin has **concurrent_threads** which controls the number of simultaneous processes fetching and downloading data (default value is 4).

Caution is recommended with the concurrency over the same service (in general, it is recommended a maximum of 4-5 jobs or threads working with the same service) and plan a step-by-step testing scenario before putting it into production. Other important point is the timing schedule, as some boundaries are related to time-frames (number of request per 10 minutes or 1 hour, for example). If you detect you reach boundaries when running all your backups during a single day of the week, please try to use 2 or 3 days and spread the load through them in order to achieve better performance results.

Specifically for the GMail module, in addition to concurrency the plugin uses batch requests that are processed in parallel as soon as it gets the answer. Therefore, throttling can be reached very easily and it's recommended to not use almost any concurrency with this module. By default, GMail uses 2 threads in parallel. Even with 2 it is expected to have some request throttled. Limits can be raised under request with Google, but if this is not a possibility and you experience throttling problems with parallelism we recommend to disable it completely (setting concurrent_threads = 1).

More information about Google Workspace API boundaries may be found here:

https://developers.google.com/drive/api/guides/limits https://developers.google.com/gmail/api/reference/quota

Performance

The performance of this plugin is highly dependent on many external factors:

- ISP latency and bandwidth
- Network infrastructure
- FD Host hardware
- FD Load
- ...

In summary, it is not possible to establish an exact reference about how much time a backup will need to complete.

Some general guidelines to understand the performance we can get:

- Many little objects to protect -> More objects per second, but less speed (MB/s)
- Big files to protect -> Less objects per second, but greater speed (MB/s)

It is recommended to benchmark your own environment in base to your requirements and needs.

The automatic concurrency mechanism (using concurrent_threads=x, default is 5) should work well for most scenarios, however, fine tune is possible if we define one job per entity and we control how many of them run in parallel, together to decrease the concurrent_threads value in order to avoid throttling from Google Cloud APIs.

There are many different possible strategies to use this plugin, so please, study what is best suiting for your needs before deploying the jobs for your entire environment, so you can get best possible results:

- You can have a job per entity (users, shared drives…) and all services
- You can split your workload through a schedule, or try to run all your jobs together.
- You can run jobs in parallel or take advantage of concurrent_threads and so run less jobs in parallel
- You can backup whole services to backup or select precisely what elements you really need inside each service (folders, paths, exclusions…)
- etc.

Specifically for Drive service, in order to maximize the performance we recommend additionally to: - Disable comments backup - Disable version history backup - Run one job per user and use the full Drive (no path selection) so the Delta function is applied. Exclude all shared units in user jobs (drive_shared_units_regex_exclude=.*) - Run one job per shared unit and use the full Drive (no path selection) so the Delta function is applied. Exclude all users in shared unit jobs (users_regex_exclude=.*)

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

Listed in this section are some scenarios that are known to cause issues.

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (you will find them in the gw-debug.err file),

you are likely using a high level of concurrency through internal concurrent_threads parameter and/or parallel jobs.
To overcome this situation you can:

a) Reduce concurrent_threads parameter
b) Reduce the number of jobs running in parallel

c) If you cannot do that you should increase JVM memory.

To increase JVM memory, you will need to:

Create a this file: '/opt/bacula/etc/gw_backend.conf'.

Below, an example of the contents: GW_JVM_MIN=2G GW_JVM_MAX=8G

Those values will define the MIN (GW_JVM_MIN) and MAX (GW_JVM_MAX) memory values assigned to the JVM Heap size. In this example we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough. Please, be careful if you are running jobs in parallel, as very big values and several jobs at a time could quickly eat all the memory of your host.

The '/opt/bacula/etc/gw_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

S3 Plugin

Overview

This document describes how to protect data stored in Simple Storage Service (S3) endpoints S3 using the **Bacula Enterprise S3 Plugin**. The S3 plugin provides the ability to download, catalog, and store the data from S3 into any other kind of storage supported by **Bacula Enterprise** directly, without using any intermediary service.

Features

The S3 plugin allows the information stored in any S3 endpoint to be backed up using a very efficient approach. It also provides a set of extra functions which allow the selection of information to be protected through different variables, as well as protecting versions of the objects, the associated ACLs, or controlling how to deal with the information stored in Glacier Storage tier of AWS.

A backup job will be able to direct the protected information to any other supported storage technology in Bacula Enterprise. This includes other S3 endpoints, other cloud endpoints of other providers such us Azure, Google or Oracle, tape, disk, block storage…

A full feature list is presented below:

- Automatic multi-threaded processes for backup and restore
- Network resiliency mechanisms
- Discovery/List/Query capabilities
- Restore objects to S3 endpoints
  - To the original S3 endpoint
  - To any other S3 endpoint
  - To the original bucket
  - To any other bucket
  - To the original path
  - To any other path
- Restore any object, version, or acl to local filesystem

- Full, Incremental & Differential backups
- Hash checks during backup and restore to ensure data integrity
- Advanced selection capabilities
  - Automatic discovery to backup all of the buckets
  - Include/Exclude buckets by name
  - Include/Exclude buckets by RegEx
  - Automatic discovery to backup all of the directories
  - Include/Exclude directories by name
  - Include/Exclude directories by RegEx
  - Include/Exclude objects having a specific AWS storage class
  - Include objects newer or older than a given date
  - Glacier objects control:
    - Skip them
    - Retrieve them but do not wait until the retrieval finishes
    - Retrieve them and wait for the retrieval to finish in order to include them into the backup
    - Specify the desired Glacier restoring tier and the retention days
- Backup/Restore of any S3 Object in any storageclass, including Glacier
- Backup/Restore of specific versions of stored S3 Objects
- Backup/Restore of S3 Objects Metadata
- Backup/Restore of ACLs of S3 buckets
- Backup/Restore of ACLs of S3 objects
- File granularity for restore
- Automatically maintain the same storage class present in backup at restore time
- Specify a new storage class at restore time
- Support for AWS S3 as well as any other generic S3 endpoints

Requirements

The Bacula S3 plugin supports AWS S3 endpoints as well as generic S3 endpoints. To be able to access S3 buckets, an authorized user with enough permissions for reading (also writing if you need to restore to an S3 bucket). This user then needs to be associated to access keys which the plugin will use to connect. More information about how to handle your access keys on AWS is available here:

- https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html

Currently the iS3 plugin must be installed on a Linux-based Operating System (OS) such as RedHat Linux, Debian, Ubuntu, Suse Linux Enterprise Server, etc where a **Bacula Enterprise** File Daemon (FD) is installed.

Bacula Systems may address support for running this plugin on a Windows platform in a future version.

The system where the Bacula File Daemon and the plugin will run must have Java version 1.8 or above installed.

Memory and CPU requirements completely depend on the usage of this plugin (concurrency, environment size, etc). However, it is expected to have a minimum of 4GB RAM in the server where the File Daemon is running. By default, every job could end up using up to 512Mb of RAM in demanding scenarios (usually it will be less). However, there can be particular situations where this could be higher. This memory limit can be adjusted internally (see *Out of Memory*).

Why Protect S3?

This is a common question that arises frequently among IT and backup professionals when it comes to any SaaS or Cloud service, so it is important to have clear understanding of the situation.

S3 is a very reliable storage solution, especially when AWS service is used, where we can find the common cloud provider capabilities intended to prevent any data loss. Usually, all data stored in any cloud service is geo-replicated using the underlying cloud infrastructure to have the information stored into several destinations automatically and transparently. Therefore, complete data loss because of hardware failures are very unlikely to happen.

The data is replicated, however there is no other data protection mechanism. Below is a list of challenges when using cloud services to store your data:

- No ransomware protection: If data suffers an attack and becomes encrypted, data is lost.
- No malicious attacker protection: If data is deleted permanently (all versions of it), data is lost.
- No real or global point-in-time recovery.
- No automated way to extract any data from the cloud to save it in external places (this could lead to eventual compliance problems).

In particular, backup needs of data stored in S3 depend highly on the usage of the S3 services. An S3 service can be used as a backup repository itself, usually as a second tier backup location. Bacula Enterprise provides its own plugin to cover that need (Cloud Storage Plugin for S3). In this type of scenario, backing up the information again is not really useful.

However, S3 is used today to store all kinds of data, for example for web servers that look for easy, quick and highly available places to access some information from different places of the world or data for analytics among many other use cases.

Usually this kind of processes are not properly controlled to navigate to different states of the data through the time and that can represent a very good reason to employ a backup tool to provide such layer of control and security.

Scope

The S3 plugin is applicable in environments using any S3 endpoint.

This document presents solutions for **Bacula Enterprise** version 16.0 and up. It is not applicable to prior versions.

Note: Important Considerations

Before using this plugin, please carefully read the elements discussed in this section.

File Spooling

In general, this plugin backs up two types of information:

- Metadata
- Files

Metadata is information associated to the objects, but also the information represented by S3 bucket and object ACLs.

While metadata is directly streamed from the cloud source to the backup engine, files need to be downloaded to the FD host before being stored. This is done in order to make some checks and to improve overall performance, as this way operations can be done in parallel. Each downloaded file is removed immediately after being completely downloaded and sent to the backup engine.

The path used for file spooling is configured with the 'path' plugin variable which, by default is set up in the s3_backend configuration file with the value: /opt/bacula/working. However it may be adjusted as needed.

Under the path directory, a 'spool' directory will be created and used for the temporary download processes.

Therefore, it is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs or through the same job (by default, this is the case, as the 'concurrent_threads' parameter is set to 5), you would need at least that size for the largest file multiplied by the number of operations in parallel you will run.

Accurate Mode and Virtual Full Backups

Accurate mode and Virtual Full backups are not supported. These features will be addressed in future versions of this plugin.

S3 APIs General Disclaimer

This plugin relies on standard S3 APIs for generic operations and in AWS S3 API in particular for specific AWS S3 services such us Storage Tiers or ACLs.

These types of Cloud or Provider APIs are owned by the provider and they could change or evolve at any time. This situation is significantly different from traditional on-premise software where each update is clearly numbered and controlled for a given server, so applications consuming that software, can clearly state what is offered and what are the target supported versions.

Amazon and anyone providing S3 APIs is usually committed to try to not break any existing functionality that could affect external applications. However, this situation can actually happen and therefore cause some occasional problems with this plugin. Bacula Systems tries to mitigate this issue with an advanced automatic monitoring system which is always checking the correct behavior of existing features, and will react quickly to that hypothetical event, but please be aware of the nature and implications of these types of cloud technologies.

Architecture

The S3 plugin uses the standard S3 API, so it is based on HTTP(s) requests invoked from the FD host where the plugin is installed. It is using the REST version of the API through the official AWS Java SDK version 2. For more information about S3 APIs please see:

- https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html

The plugin will contact the S3 endpoint to be protected during backups in order to get the needed metadata and files. Conversely, the plugin will receive them from an SD and will perform uploads as needed during a restore operation.

The implementation is done through a Java daemon, therefore Java is a requirement in the FD host.

Below is a simplified vision of the architecture of this plugin inside a generic **Bacula Enterprise** deployment:

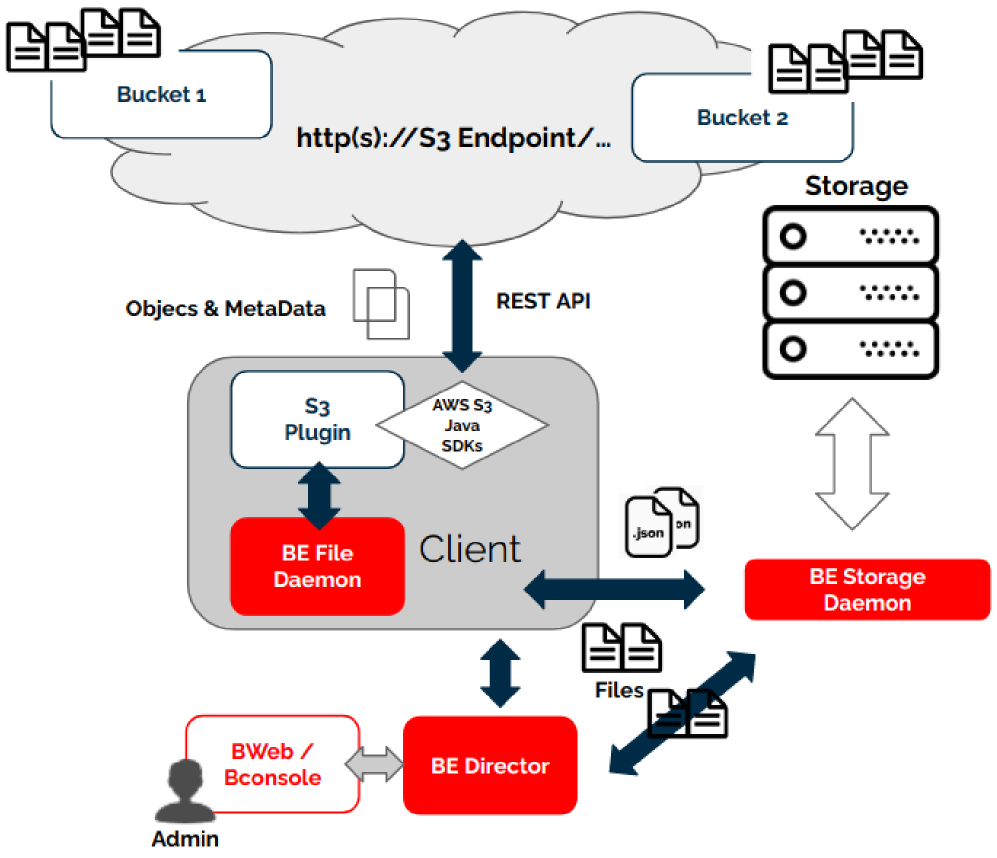


Fig. 97: S3 Plugin Architecture

ACLs are stored in JSON format preserving their original values, while files will present the key value of the S3 object as their name in the Bacula catalog.

Catalog Structure

Files will keep their names in the catalog and will be included in a path like the following one:

- `/@s3/bucketName/path/to/file/name-file.extension`

File Integrity and Checksums

When a file is uploaded to S3, the user can select to use a file integrity check using 4 different algorithms:

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/checking-object-integrity.html

The S3 plugin uses this information (it was used during the upload) in order to calculate the checksum of the downloaded data during the backup processes to validate the integrity of every file. In case there are any discrepancies, the plugin will warn about them with an error in the joblog.

When a file is restored to S3 buckets the S3 plugin will calculate an MD5 checksum and will inform the S3 service to calculate and compare the value once the data is completely uploaded.

Both checks may be disabled in case the target system does not support them or to save some computational resources by activating the fileset variable 'disable_hashcheck' (example: disable_hashcheck=true).

Versions History

AWS S3 service can be configured to retain the history for the stored objects (ie: versions or revisions of the same file):

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html

A new version of an object can be created each time the file is saved. Previous versions of an object may be retained for a finite period of time depending on specific settings associated to the bucket. By default, this feature is disabled.

The S3 plugin is able to backup this information if the special **version_history** backup parameter is activated.

File versions have some particularities compared to normal files:

- They are backed up as a regular file. This means a revision has its own full metadata as the parent file itself has. All the metadata is the same as the file contains, except for size, dates and name.
- The name of the file is modified, so at restore time you can see the version number and the version date in the filename. Example:
  - Parent file: myDoc.doc
  - Versions:
    - myDoc###v25.0_2021-01-19_234537.doc
    - myDoc###v24.0_2021-01-17_212537.doc
    - myDoc###v23.0_2021-01-12_104537.doc
    - …
    - Notice that the extension of the file is kept
- Versions are not restored by default. You need to disable the special restore parameter 'skip_versions', by setting it to 0.

File versions are backed up in all backup levels (Full, Incremental, Differential), this means you can track all the changes of the files in your backup. For example, every Incremental run is going to backup only the new modified versions since the last backup.

Here is an example of a some files backed up with revisions included, listed in a restore session:

Listing 242: **versions in a job**

```
cwd is: /@s3/bucketName/myDir/
$ ls
Contentiones/
Dolores###v_2022-09-12_104436729.doc
Dolores###v_2022-09-12_104444796.doc
Dolores###v_2022-09-12_104448264.doc
Dolores.doc
Legimus###v_2022-09-12_104518541.mp4
Legimus###v_2022-09-12_104527444.mp4
Legimus###v_2022-09-12_104530638.mp4
Legimus.mp4
Netus.ppt
Posse###v_2022-09-12_104456414.docx
Posse###v_2022-09-12_104506748.docx
Posse###v_2022-09-12_104510261.docx
Posse.docx
Ridiculus.jpeg
```

Installation

The Bacula File Daemon and the S3 Plugin need to be installed on the host that is going to connect to the S3 endpoint. The plugin is implemented over a Java layer, therefore it can be deployed on the platform better suited for your needs among any of the officially supported platforms of **Bacula Enterprise** (RHEL, SLES, Debian, Ubuntu, etc). Please, note that you may want to deploy your File Daemon and the plugin on a virtual machine directly deployed in Amazon Web Services, if your endpoint is under AWS, in order to reduce the latency between it and the S3 APIs.

The system must have Java >= 1.8 installed (openjdk-1.8-jre for example) and the Java executable should be available in the system PATH.

Bacula Packages

We are using Debian Buster as the example base system to proceed with the installation of the **Bacula Enterprise** S3 Plugin. In this system, the installation is most easily done by adding the repository file suitable for the existing subscription and the Debian version utilized. An example could be /etc/apt/sources.list.d/bacula.list with the following content:

Listing 243: **APT**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪buster-64/ buster main
deb https://www.baculasystems.com/dl/@customer-string@/debs/s3/@version@/
↪buster-64/ buster s3
```

Note: Replace @*customer-string*@ with your Bacula Enterprise download area string. This string is visible in the Customer Support portal.

After that, a run of apt update is needed:

Listing 244: **APT install**

```
apt update
```

Then, the plugin may be installed using:

Listing 245: **APT install**

```
apt install bacula-enterprise-s3-plugin
```

The plugin has two different packages which should be installed automatically with the command shown:

- bacula-enterprise-s3-plugin
- bacula-enterprise-s3-plugin-libs

Alternately, manual installation of the packages may be done after downloading the packages from your Bacula Systems provided download area, and then using the package manager to install. An example:

Listing 246: **APT install**

```
dpkg -i bacula-enterprise-*
```

The package will install the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-s3-plugin-x.x.x.jar and bacula-s3-plugin-libs-x.x.x.jar). Please note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.
- The S3 plugin file (s3-fd.so) in the plugins directory (usually /opt/bacula/plugins)
- Backend file (s3_backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-s3-plugin-x.x.x.jar file in order to launch it, even though usually there should only ever be one file.

Configuration

This plugin uses regular filesets to be used in backup jobs where it is necessary to include a 'Plugin =' line inside of an Include block. The structure of the *Plugin* = line is shown below:

Listing 247: **Fileset S3**

```
Fileset {
   Name = FS_S3
   Include {
      Options {
        signature = MD5
        ...
      }
      Plugin = "s3: <s3-parameter-1>=<s3-value-1> <s3-parameter-2>=<s3-value-
↪2> ..."
```

(continues on next page)

(continued from previous page)

```
    }
}
```

It is **strongly recommended** to use only one ‘Plugin’ line in a fileset. The plugin offers the flexibility to combine different modules or entities to backup inside the same plugin line. Different endpoints, should be using different filesets and different jobs.

TRhe sub-sections below list all of the parameters you can use to control the S3 Plugin’s behavior.

Parameters which allow a list of values can be assigned with a list of values separated by ‘,’.

Common parameters

These parameters are common to some other **Bacula Enterprise** plugins and they modify generic things not directly associated to the S3 plugin:

Option	Require	Default	Values	Example	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is encountered with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the rest and only show a warning
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/	Allows you to define a config file where you may configure any plugin parameter. Therefore you don’t need to put them directly in the Plugin line of the fileset. This is useful for shared data between filesets and/or sensitive data such as customer_id.
log	No	/opt/bac debug.l	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/s3.log	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/s3/s3-debug.log* files containing debug information which is more verbose with a greater debug number
path	No	/opt/bac	An existing path with enough permissions for File Daemon to create any internal plugin file	/mnt/myvol/	Uses this path to store metadata, plugin internal information, and temporary files

Advanced common parameters

The following are advanced parameters. They should not be modified in most common use cases:

Option	Require	Default	Values	Example	Description
stream_slee	No	1	Positive integer (1/10 seconds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_ma	No	120	Positive integer (seconds)	360	Max wait time for FD to answer packet requests
time_max_l	No	8640	Positive integer (seconds)	4320	Maximum time to wait to overwrite a debug log that was marked as being used by another process
logging_max_f	No	50M	String size	300M	Maximum size of a single debug log file (working/s3/s3-debug.log* files containing debug information which is more detailed with a greater debug number)
logging_max_l	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
split_config	No	=	Character	:	Character to be used in config_file parameter as separator for keys and values
opener_que	No	1200	Positive integer (seconds)	3600	Timeout when internal object opener queue is full
publisher_queu	No	1200	Positive integer (seconds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features:

- The ".log" files are rotated automatically. Currently each file can be 50Mb at maximum and the plugin will keep 25 files.
- This behavior may be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The rotated ".log" files are renamed like {path}/s3/s3.%d{yyyy-MM-ddHHmm}.log.gz, where `path` is taken from the value of the **path** parameter and `%d` is the date.
- The ".err" file may show contents even if no real error occurred in the jobs. It may also show contents even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general log rotating tool like 'logrotate'.
- Backups in parallel and also failed backups will generate several log files. For example: s3-debug-0.log, s3-debug-1.log…

Tuning Parameters

This set of parameters are again common to some other plugins and modify general things not directly associated to the S3 plugin. They are also advanced ones. They should not be modified in general.

They can be used to tune the behavior of the plugin to be more flexible in particularly bad network environments or when there is significant job concurrency, etc.

Option	Require	Default	Values	Example	Description
back	No	30	0-50	1	Number of maximum queued internal operations between service static internal threads (there are 3 communicating through queues with the set size: fetcher, opener and general publisher to Bacula core). This could potentially affect S3 API concurrent requests and consequently, Google throttling. It is only necessary to modify this parameter, in general, if you are going to run different jobs in parallel
concurrent_	No	5	0-10	1	Number of maximum concurrent backup threads running in parallel in order to fetch or open data for running download actions. This means every service fetcher and service opener will open this number of child concurrent threads. This will affect s3 api concurrent requests. S3 API could throttle requests depending on a variety of circumstances. It is only necessary to modify this parameter, in general, if you are going to run different jobs in parallel. If you want to have a precise control of your concurrency through different jobs, please set this value to 1. Also, please be careful with the memory requirements. Multi-threaded jobs can significantly increase job memory consumption
general_	No	5	Positive integer (number of retries)	10	Number of retries for the general external retry mechanism
general_	No	50	Positive integer (seconds)	100	General Plugin delay between retries

S3 has a very reasonable bandwidth for running concurrent request against any bucket and throttling should generally not be an issue. However, there still exist some limits, you can learn more about them in the following link:

- https://aws.amazon.com/premiumsupport/knowledge-center/s3-request-limit-avoid-throttling/

S3 Service Parameters

Parameters to connect and control the behavior of the plugin regarding the S3 service:

Option	Require	Default	Values	Example	Description
endpoint	No	https://s3.amaz com/	URL of a S3 endpoint	https://192.168.10.4:9000	Main URL where the S3 service is being served
access_	Yes		Valid S3 access key to read from or write to buckets	KMN02jCv5YpmirOa	Valid access key to read from or write to buckets
secret_	Yes		Valid S3 secret key associated to the provided access_key to read from or write to buckets	bTq6FzPbnU9x1jqka5	Valid S3 secret key associated to the provided access_key to read from or write to buckets
region	No	eu-west-1	AWS region code-name: eu-west-1, us-east-1, us-east-2, eu-west-1, eu-south-1…	us-east-2	AWS Region code name where the buckets to backup exist: https://docs.aws.amazon.com/directoryservice/latest/admin-guide/regions.html
force	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	true	Force requests to use PathStyle (http(s)://myS3host/bucketname) instead of HostStyle (http(s)://bucketname.myS3host)
thum	No		String representing a SHA-256 SSL Certificate thumbprint. It supports AA:AA:AA… format, but also aaaaaa format.	D8:B8:9D:B1:AD:3E: :62:0D:BA:EA:D6:12:21:5B or d8b89db1ad3e37fd72 5b7d9927630591127e3c	Specify a trusted certificat.e For a HTTPS connection with an specific endpoint, will get the certificate of the endpoint, compute the thumbprint and connect with it it matches, without considering if the certificate is valid
buck	No		Strings representing existing **buckets** for the given access information (endpoint, keys and region) separated by ','	my-bucket1,mybucket2	Backup only specified **buckets** existing into the provided endpoint (and accessible through the provided credentials). If no bucket is listed, all of them will be backed up
buck	No		Strings representing existing **buckets** for the given access information (endpoint, keys and region) separated by ','	Personal	Exclude selected **buckets** belonging to the configured endpoint (and accessible through the provided credentials)
buck	No		Valid regex	.*Company	Backup matching buckets. Please, only provide list parameters (bucket + bucket_exclude) or regex ones. But do not try to combine them.
objects_	No		Existing executable path by File Daemon User	/opt/bacula/scripts/s3-selector.sh	Run dynamic command that will output a list of strings representing S3 Objects of the configured bucket that need to be backed up
buck	No		Valid regex	.*Plan	Exclude matching buckets from the selection. Please, only provide list parameters

Note: `force_path_style` option is available since BE 16.0.7.

Note: `objects_from_script` option is available since BE 18.0.8.

The following example shows what is expected with the `objects_from_script` parameter:

Listing 248: **Script example for objects_from_script**

```
$ cat /tmp/s3-selector.sh
echo "FolderA/file1"
echo "file2"
echo "FolderB/"
echo "FolderC/file3.abc"
echo "FolderD/file4.def"

# The fileset would contain then
Fileset {
  Name = s3FromScript
  Include {
    ...
    Plugin = "s3: ... objects_from_script=\"/tmp/selector.sh\""
  }
}
```

Restore Parameters

Restore Parameters

The S3 plugin is able to restore to the local file system on the server where the File Daemon is running or to the S3 environment. The method selected is based on the value of the *where* parameter at restore time:

- Empty or '/' (example: where=/) → S3 restore method will be triggered
- Any other path for where (example: where=/tmp) → Local file system restore will be triggered

When using S3 restore method, the following parameters may be modified by selecting 'Plugin Options' during the bconsole restore session:

Option	Require	Default	Values	Example	Description
destination_	No		Destination bucket name	myrestore-bucket	Destination bucket where restore data will be uploaded. If no bucket is set, every selected file will be restored in the original bucket
destination_	No		Destination path to be created (or existing) into the selected bucket	RestoreFolder	Destination path where all selected files to restore will be placed. If no destination_path is provided, every selected file will be restored into their original path
destination_	No		STANDARD, REDUCED_REDUNDANCY, GLACIER, STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, DEEP_ARCHIVE, OUTPOSTS, GLACIER_IR	ONEZONE_	Destination storage class to be used for the restore. If none is provided, the original storage class of this object will be used
skip_	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Skip restoring former file versions (tagged with '###date') even if they are selected. **Important**: Note that this parameter is **enabled by default**, as we consider not restoring file versions the most common case. You need to disable it in order to have this kind of files restored
skip_	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Skip restoring ACLs even if they are selected. **Important**: Note that this parameter is **enabled by default**, as we consider not restoring file ACLs the most common case. You need to disable it in order to have this kind of information restored
disable_	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Disable hashcheck mechanism for file integrity, so you can avoid some computation resources or disable it if your S3 server does not support it
endpoint	No		URL of a S3 endpoint	https://192.168.10.4:9000	Cross-endpoint/bucket restore: Main URL where the S3 service is being served
access_	Yes		Valid S3 access key to read from or write to buckets to backup	KMN02	Cross-endpoint/bucket restore: Valid access key to write to the bucket to restore
secret_	Yes		Valid S3 secret key associated to the provided access_key to read from or write to buckets to backup	bTq6Fz	Cross-endpoint/bucket: Valid S3 secret key associated to the provided access_key to write to buckets to restore
region	No		AWS region code-name: eu-west-1, us-east-1, us-east-2, eu-west-1, eu-south-1…	us-east-2	Cross-endpoint/bucket: AWS Region code name where the buckets to write to exist: https://docs.aws.amazon.com/directoryservice/latest/admin-guide/regions.html
force	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	us-east-2	Cross-endpoint/bucket: Force requests to use PathStyle (http(s)://myS3host/bucketname) instead of HostStyle (http(s)://bucketname.myS3host)
debug	No		0, 1, 2 ,3, 4, 5, 6, 7, 8, 9	3	Change debug level

Operations

Backup

The S3 plugin backup configurations currently have one specific requirement in the Job resource. Below we show some examples.

Job Example

The only special requirement with S3 jobs is that Accurate mode backups must be disabled, as this feature is not supported at this time.

Listing 249: **Job Example**

```
Job {
   Name = s3-mybucket-backup
   Fileset = fs-s3-all
   Accurate = no
   ...
}
```

Fileset Examples

The S3 plugin is flexible enough to configure almost any type of desired backup. Multiple *Plugin=* lines should not be specified in the Include section of a Fileset for the S3 Plugin.

Fileset examples for different scenarios are shown below.

Setup external config file and backup 'mybucket' of AWS:

Listing 250: **Fileset Example**

```
Fileset {
   Name = FS_MYBUCKET
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: config_file=/opt/bacula/etc/s3.settings bucket=mybucket"
   }
}
```

Listing 251: **Settings file**

```
$ cat /opt/bacula/etc/s3.settings
access_key=XXXXXXXXXXXXXXXXXXXX
secret_key=YYYYYYYYYYYYYYYYYYYY
region=us-east-1
bucket=mybucket
```

Increase number of threads:

Listing 252: **Fileset Example**

```
Fileset {
   Name = fs-s3-concurrent
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪concurrent_threads=10"
   }
}
```

Backup all the buckets associated to the provided keys on AWS in region us-east-2:

Listing 253: **Fileset Example**

```
Fileset {
   Name = fs-s3-all-buckets
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪region=us-east-2"
   }
}
```

Backup folders A and B in the bucket ‘mybucket’ of region us-west-1 (default region):

Listing 254: **Fileset Example**

```
Fileset {
   Name = fs-s3-mybucket-A-B
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪bucket=mybucket folder=A,B"
   }
}
```

Backup folders starting with A in the bucket ‘mybucket’ of region us-west-1 (default region):

Listing 255: **Fileset Example**

```
Fileset {
   Name = fs-mybucket-startA
   Include {
      Options {
        signature = MD5
      }
```

(continues on next page)

(continued from previous page)

```
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪bucket=mybucket folder_regex_include=A.*"
   }
}
```

Backup bucket ‘mybucket’ and run retrievals for glacier objects, also wait for them so they are backed up. Use ‘expedited’ type with 30 days of retention after completing the retrievals:

Listing 256: **Fileset Example**

```
Fileset {
   Name = fs-s3-mybucket-glacier
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪bucket=mybucket glacier_mode=RETRIEVAL_WAIT_DOWNLOAD glacier_tier=EXPEDITED␣
↪glacier_days=30"
   }
}
```

Backup bucket ‘mybucket’, but do not get objects from GLACIER_IR, also get information only from 2022:

Listing 257: **Fileset Example**

```
Fileset {
   Name = fs-s3-mybucket-no-ir-2022
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪bucket=mybucket storageclass_exclude=GLACIER_IR date_from=\"2022-01-01␣
↪00:00:00\" "
   }
}
```

Restore

Restore operations are done using standard **Bacula Enterprise** bconsole commands.

The *where* parameter controls if the restore will be done locally to the File Daemon’s file system or to the S3 service:

- where=/ or empty value → Restore will be done over S3
- where=/any/other/path → Restore will be done locally to the File Daemon file system

Restore options are described in the *Restore Parameters* section of this document, so here we are going to simply show an example restore session:

Listing 258: **Restore Bconsole Session**

```
**restore
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Select object to restore
    14: Cancel
Select item:  (1-14): 5
Automatically selected Client: 127.0.0.1-fd
Automatically selected Fileset: FS_S3
+-------+-------+----------+----------+---------------------+-----------------
↪--+
| jobid | level | jobfiles | jobbytes | starttime           | volumename      ␣
↪  |
+-------+-------+----------+----------+---------------------+-----------------
↪--+
|     1 | F     |       14 |   35,463 | 2022-09-08 11:53:57 | TEST-2022-09-
↪08:0 |
+-------+-------+----------+----------+---------------------+-----------------
↪--+
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
12 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ mark *
12 files marked.
```

(continues on next page)

(continued from previous page)

```
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    TEST-2022-09-08:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


12 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /tmp/regress/tmp/bacula-restores
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2022-09-08 12:03:12
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 9
Please enter the full path prefix for restore (/ for none): /
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
```

(continues on next page)

(continued from previous page)

```
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2022-09-08 12:03:12
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : s3: region="US-EAST-1" access_key=
↪"XXXXXXXXXXXXXXXXXXXXXXXXXXX" secret_key=
↪"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY" bucket="bacbucket" folder="SRC_
↪REGRESS_20220908115256" storageclass="ONEZONE_IA" acl=1 version_history=1␣
↪debug=6
Plugin Restore Options
Option                         Current Value        Default Value
destination_bucket:            *None*               (*None*)
destination_path:              *None*               (*None*)
destination_storageclass:      *None*               (*None*)
skip_acls:                     *None*               (yes)
skip_versions:                 *None*               (yes)
disable_hashcheck:             *None*               (*None*)
endpoint:                      *None*               (*None*)
access_key:                    *None*               (*None*)
secret_key:                    *None*               (*None*)
region:                        *None*               (*None*)
debug:                         *None*               (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
     1: destination_bucket (Change destination bucket)
     2: destination_path (Set a destination path)
     3: destination_storageclass (Specify the storage class to be used for␣
↪restored objects)
     4: skip_acls (Skip ACLs object and do not restore them even if they are␣
↪selected)
     5: skip_versions (Skip versioned objects and do not restore them even if␣
↪they are selected)
     6: disable_hashcheck (Disable md5 file calculation and checking after␣
↪upload if computational resources are not enough for big files)
```

(continues on next page)

(continued from previous page)

```
     7: endpoint (Specify a different destination endpoint)
     8: access_key (Set a different access key to access to the destination)
     9: secret_key (Set a different secret key to access to the destination)
    10: region (Set the destination region)
    11: debug (Change debug level)
Select parameter to modify (1-11): 2
Please enter a value for destination_path: restored_data
Plugin Restore Options
Option                         Current Value        Default Value
destination_bucket:            *None*               (*None*)
destination_path:              restored_data        (*None*)
destination_storageclass:      *None*               (*None*)
skip_acls:                     *None*               (yes)
skip_versions:                 *None*               (yes)
disable_hashcheck:             *None*               (*None*)
endpoint:                      *None*               (*None*)
access_key:                    *None*               (*None*)
secret_key:                    *None*               (*None*)
region:                        *None*               (*None*)
debug:                         *None*               (*None*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:
Replace:         Always
Fileset:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2022-09-08 12:03:12
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
```

Restore options using S3 allow you to:

- Restore into the original bucket or in a different one (destination_bucket)
- Restore to the original endpoint or to a different one (see next ‘Cross endpoint restore’)
- Restore to the original path or to a different one (destination_path)
- Restore using the original storageclass or set up a new one for all the restored objects (destination_storageclass)
- Restore selected file versions (unset skip_versions)
- Restore selected ACLs (unset skip_acls)
- Restore without using MD5 hashcheck (set disable_hashcheck)

Cross Endpoint Restore

You can perform cross-endpoint restores and/or change the destination bucket using the restore variables:

- endpoint
- access_key
- secret_key
- region
- destination_bucket

Obviously, it is necessary to set up the destination endpoint values.

List

It is possible to list information using the bconsole *.ls* command and providing a path. In general, we need to provide the connection information and the path we are interested in.

Below we can see an example:

List S3 Contents

Listing 259: **List bucket contents**

```
*.ls plugin="s3:region=\"US-EAST-1\" access_key=\"XXXXXXXXXXXXXXXXXXXX\"␣
↪secret_key=\"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\"␣
↪bucket=jorgebacbucket" client=127.0.0.1-fd path=/
 Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
 -rw-r-----   1 nobody   nogroup              17553 2022-08-26 16:10:39  /
↪@s3/jorgebacbucket/dir1/Altera.doc
 -rw-r-----   1 nobody   nogroup               6183 2022-08-26 16:10:40  /
↪@s3/jorgebacbucket/dir1/Efficiantur.ppt
 -rw-r-----   1 nobody   nogroup              10336 2022-08-26 16:12:24  /
↪@s3/jorgebacbucket/dir2/Discere.ppt
 -rw-r-----   1 nobody   nogroup              17183 2022-08-26 16:13:24  /
↪@s3/jorgebacbucket/dir3/Tacimates.ppt
 -rw-r-----   1 nobody   nogroup              15062 2022-08-26 16:14:10  /
↪@s3/jorgebacbucket/dir3/Quas.doc
 -rw-r-----   1 nobody   nogroup              10646 2022-08-26 16:20:06  /
↪@s3/jorgebacbucket/dir3/Ligula.ppt
 -rw-r-----   1 nobody   nogroup               6958 2022-08-26 16:21:07  /
↪@s3/jorgebacbucket/dir3/Suscipiantur.ppt
 -rw-r-----   1 nobody   nogroup               4408 2022-08-26 16:21:06  /
↪@s3/jorgebacbucket/dir3/Vix.doc
 -rw-r-----   1 nobody   nogroup               6307 2022-08-26 16:27:05  /
↪@s3/jorgebacbucket/dir3/Cetero.doc
 -rw-r-----   1 nobody   nogroup               4078 2022-08-26 16:27:06  /
↪@s3/jorgebacbucket/dir3/Neglegentur.ppt
 -rw-r-----   1 nobody   nogroup              11607 2022-08-26 16:29:11  /
↪@s3/jorgebacbucket/dir3/Commodo.doc
 -rw-r-----   1 nobody   nogroup               5938 2022-08-26 16:29:18  /
```

(continues on next page)

(continued from previous page)

```
↪@s3/jorgebacbucket/dir3/Reque.ppt
 -rw-r-----    1 nobody   nogroup               13962 2022-08-31 17:12:11  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Bibendum.ppt
 -rw-r-----    1 nobody   nogroup               17716 2022-08-31 17:12:09  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Solum.doc
 -rw-r-----    1 nobody   nogroup               11254 2022-08-31 17:17:33  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Interdum.ppt.ONEZONE_IA
 -rw-r-----    1 nobody   nogroup               11254 2022-08-31 17:17:34  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Interdum.ppt.REDUCED_REDUNDANCY
 -rw-r-----    1 nobody   nogroup                5092 2022-08-31 17:17:32  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Tortor.doc.ONEZONE_IA
 -rw-r-----    1 nobody   nogroup                5092 2022-08-31 17:17:32  /
↪@s3/jorgebacbucket/AutoTiers 2022-08-31/Tortor.doc.REDUCED_REDUNDANCY
 -rw-r-----    1 nobody   nogroup                  12 2022-08-26 11:35:17  /
↪@s3/jorgebacbucket/IntelligentS31.txt
 -rw-r-----    1 nobody   nogroup                  12 2022-08-26 11:35:18  /
↪@s3/jorgebacbucket/IntelligentS32.txt
 ....
 2000 OK estimate files=279 bytes=3,059,174
```

We can also run the same command using the .query command.

Query S3 Contents

Listing 260: **Query bucket contents**

```
*.query client=127.0.0.1-fd plugin="s3: access_key=\"XXXXXXXXXXXXXXXXXXXX\"␣
↪secret_key=\"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\" region=us-east-1␣
↪bucket=jorgebacbucket" parameter=/
key=AutoSimple 2022-08-26 04.10.37/Altera.doc
storageclass=STANDARD
size=17553
lastModified=2022-08-26 14:10:39
key=AutoSimple 2022-08-26 04.10.37/Efficiantur.ppt
storageclass=STANDARD
size=6183
lastModified=2022-08-26 14:10:40
key=AutoSimple 2022-08-26 04.12.20/Discere.ppt
storageclass=STANDARD
size=10336
lastModified=2022-08-26 14:12:24
key=AutoSimple 2022-08-26 04.13.20/Tacimates.ppt
storageclass=STANDARD
size=17183
lastModified=2022-08-26 14:13:24
key=AutoSimple 2022-08-26 04.13.54/Quas.doc
storageclass=STANDARD
size=15062
lastModified=2022-08-26 14:14:10
key=AutoSimple 2022-08-26 04.18.51/Ligula.ppt
```

(continues on next page)

(continued from previous page)

```
storageclass=STANDARD
size=10646
lastModified=2022-08-26 14:20:06
...
```

Query command can also list buckets and show a remote server thumbprint we want to trust in a HTTPS context. Below some examples:

Query buckets

It is possible to list the buckets on a given endpoint or in AWS (permissions in the target server need to allow this list bucket function):

Listing 261: **List buckets**

```
*.query client=127.0.0.1-fd plugin="s3: access_key=\"XXXXXXXXXXXXXXXXXXXX\"␣
↪secret_key=\"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\"" parameter=bucket
bucket=bucket1
creationDate=2024-11-12 04:49:21
bucket=bucket2
creationDate=2024-09-17 15:51:42
bucket=test3
creationDate=2024-11-07 21:44:46
....
```

This command is convenient if there is many buckets to protect, as it can be used through scan plugin (Automation Center in BWeb).

Get Thumbprint

Listing 262: **Get thumbprint**

```
*.query client=127.0.0.1-fd plugin="s3: endpoint=\"https://mys3.endpoint:9000\
↪"" parameter=thumbprint
thumbprint=d8b89db1ai93k7fd721094a50fac04if0dbaead612215b7d992760039plm873c
```

Cloud Costs

As you will already know, storing data in the cloud will create additional costs. Please see the below information for the different cloud providers.

Data transfer needs to be considered as well. While upload of data is typically free or very low cost, the download is typically not free, and you will be charged per operation and per amount of data transerred.

Amazon has a pricing model for each of its storage tiers. Additionally, the costs will vary with the region you use. More information may be found here:

- https://aws.amazon.com/s3/pricing/

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Troubleshooting

This section lists some scenarios that are known to cause issues and how to solve them.

Out of Memory

If you ever face *OutOfMemory* errors from the Java daemon (you will find them in the s3-debug.err file),

you are likely using a high level of concurrency through the internal 'concurrent_threads' parameter and/or parallel jobs.
To overcome this situation you can:

a) Reduce concurrent_threads parameter

b) Reduce the number of jobs running in parallel

c) If you cannot do that you should increase JVM memory.

To increase JVM memory, you will need to:

Create the following file: '/opt/bacula/etc/s3_backend.conf'

Add the following parameters to the file:

|S3_JVM_MIN=2G |S3_JVM_MAX=8G

Those values will define the MIN (S3_JVM_MIN) and MAX (S3_JVM_MAX) memory values assigned to the JVM Heap size. In this example we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough. Please be careful if you are running jobs in parallel, as very big values and several concurrent jobs could quickly consume all of the memory of your host.

The '/opt/bacula/etc/s3_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

Swift Object Backup

Executive summary

This document is intended to provide insight into the considerations and processes required to successfully implement a Swift Object Storage backup technique using Bacula Enterprise.

Note: This functionality is available as of Bacula Enterprise version 8.10.

Swift Object Storage

Architecture

The Bacula Enterprise Swift plugin uses the python-swiftclient library to access the Swift Object Storage daemon.

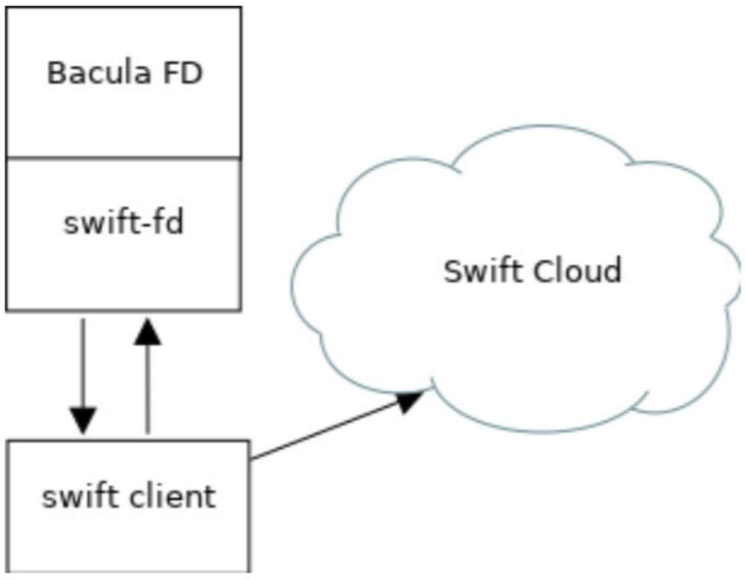


Fig. 98: Swift Plugin Architecture

Features

The configuration for Swift container backups is done in a Bacula Fileset configuration file using parameters to the Swift plugin command line.

During a backup, the Bacula plugin will contact the Swift system to retrieve Objects one by one. During an incremental or a differential backup session, the Bacula File Daemon will need to list all objects and retrieve their attributes to determine if an object must be included in the job.

If found, any container or object metadata are saved during backup.

At restore time, the Bacula Swift plugin will restore the Swift metadata, and the user will be able to restore Swift objects to:

- The original Swift container
- An alternate Swift container/system
- A local directory

Large objects will be restored automatically using the DLO technique.

Swift metadata for containers are automatically backed up with Objects.

Installation of the Plugin

On the Bacula File Daemon that you want to connect to your Swift Object Storage, extend the repository file for your package manager to contain a section for the Swift plugin. For example, in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseSwiftPlugin]
name=Bacula Enterprise Swift Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/swift/
↪@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

On Centos7, python3 and the Swift client library are not avaiable via the standard repositories. The EPEL repository provides python34 and the `python-swiftclient` package must be installed via the `python34-pip` package from this repository.

```
# yum install epel-release
# yum install python34 python34-pip
# pip3.4 install python-swiftclient
```

On RHEL 7, the package is available in the repository "rhel-7-server-openstack-11-tools-rpms".

On Debian Jessie, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪jessie-64/ jessie main
deb https://www.baculasystems.com/dl/@customer-string@/debs/swift/@version@/
↪jessie-64/ jessie swift
```

Once the repository is configured for your system, perform a `yum update` or `apt-get update`, then the package *bacula-enterprise-swift-plugin* can be installed with `yum install` or `apt-get install`.

```
  # yum install bacula-enterprise-swift-plugin
or
  # apt-get update
  # apt-get install bacula-enterprise-swift-plugin
```

If you prefer to manually install the packages, you may download them directly from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

Configuration

Plugin Parameters

The following parameters effect any type of Swift plugin Job (Backup, Estimation or Restore).

- `user=<string>` specifies the username to access the Swift system. This parameter is mandatory.
- `password=<string>` specifies the password to access the Swift system. The `password` or the `passfile` parameter is mandatory.
- `passfile=<string>` specifies a file local to the File Daemon that contains the password for the user. Only the first line of the file will be read. The `password` or the `passfile` parameter is mandatory.
- `url=<string>` specifies the URL of the Swift system. This parameter is mandatory.
- `abort_on_error=<0 or 1>`, or just `abort_on_error` specifies whether or not the Backend should abort it's execution if a fatal error occurs during Backup, Estimation or Restore. This parameter is optional. The default value is 0.
- `debug=<0 or 1>`, or just `debug` specifies how much information should be logged to the Backend File Log. This parameter is optional. The default value is 0.
- `insecure=<0 or 1>`, or just `insecure` specifies whether or not the SSL certificate should be verified. This parameter is optional. The default value is 0.

Plugin Estimation and Backup Parameters

- `include=<string>` specifies which containers and/or objects should be backed up from the Swift Storage System. This parameter is optional. There may be more than one `include` parameter.
- `regexinclude=<regex>` specifies, using a Regular Expression, which containers and/or objects should be backed up from the Swift Storage System. This parameter is optional. There may be more than one `regexinclude` parameter.
- `exclude=<string>` specifies which containers and/or objects should NOT be backed up from the Swift Storage System. This parameter is optional. There may be more than one `exclude` parameter.
- `regexexclude=<regex>` specifies, using a Regular Expression, which containers and/or objects should NOT be backed up from the Swift Storage System. This parameter is optional. There may be more than one `regexexclude` parameter.

If none of the paramaters `include`, `regexinclude`, `exclude` or `regexexclude` are specified, all containers and objects from the Swift Storage System for an account (`user` parameter) will be backed up.

Plugin Restore Parameters

- user=<string> specifies an account where restore will be performed. This parameter is optional. If not set, the user parameter from the backup Job will be used.
- password=<string> specifies the password to access the Swift system during restore. This parameter is optional. If not set, the password=<string> parameter from the backup job will be used.
- url=<string> specifies the URL of the Swift system during a restore. This parameter is optional. If not set, the url=<string> parameter from the backup Job will be used.
- be_object_segment_size=<size> specifies, in bytes, the size of the segments of a DLO object. The default is 5MB. This parameter is optional.
- restore_local=<yes or no> specifies that the files should be restored to a local directory based on the where= restore job parameter. This parameter is optional.

Fileset Examples

In the example below, all objects inside the container container1 will be backed up.

```
Fileset {
 Name = FS_Swift
 Include {
  Plugin = "swift: user=r1 password=t1 URL=http://swift:8080/auth/v1.0 \
           include=container1/*"
 }
}
```

In the example below, all objects that do not end with tmp inside the container container1 will be backed up.

```
Fileset {
 Name = FS_Swift_without_tmp
 Include {
   Plugin = "swift: user=root password=test1 URL=http://swift/auth/v1.0 \
            include=container1/* exclude=*tmp"
 }
}
```

This example is the same as the exclude one above, but using regexexclude instead:

```
Fileset {
 Name = FS_Swift_without_tmp
 Include {
  Plugin = "swift: include=container1/* regexclude=.*\\.tmp\\Z(?ms)"␣
↪URL=http://swift/auth/v1.0 user=r1 password=p1"
 }
}
```

In the example below, all objects that end with .pdf inside the container container1 will be backed up.

```
Fileset {
 Name = FS_Swift_without_tmp
 Include {
  Plugin = "swift: user=r1 password=t1 URL=http://swift/auth/v1.0 \
           include=container1/* regexinclude=.*\\.pdf\\Z(?ms)"
 }
}
```

In the example below, all objects of all containers will be backed up.

```
Fileset {
  Name = FS_Swift_everything
  Include {
    Options {
      Compression = LZO
    }
    Plugin = "swift: user=r1 password=t1 URL=http://swift:8080/auth/v1.0"
  }
}
```

In the example below, all objects of all containers will be backed up. The password of the swift user `root` will be taken from the file `/etc/swift.txt` file.

```
Fileset {
  Name = FS_Swift_passfile
  Include {
   Plugin = "swift: user=r1 passfile=/etc/swift.txt URL=http://swift/auth/v1.0
↪"
  }
}
```

Restore Examples

Restore to a Subdirectory Inside the Container

If the objects were selected inside `container1`, the following restore command will relocate objects under `/tmp`. For example: `/container1/home/file.png` will be restored as `/container1/tmp/home/file.png`.

```
* restore where=/container1/tmp
```

Restore to an Alternate Container

If the objects where selected inside `container1`, the following restore command will relocate them under `container2`. ex: `/container1/home/file.png` will be restored as `/container2/home/file.png`.

```
* restore where=container2
```

Restore to a Local Directory

You may restore the files to your local filesystem by setting the `restore_local` option to `yes` in the `Plugin Restore Options` menu. You must specify the `where=` option.

```
* restore where=/home/user/my_restored_files
```

After the selection of the files you want to restore, modify the plugin option, select option 5 `restore_local`, and set it to 'yes'

```
Run Restore job
JobName:         RestoreFiles
...
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
     ...
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : swift: user=test:tester password=testing
     URL=http://localhost:8080/auth/v1.0 include=data1/*

Plugin Restore Options
user:                  *None*               (*None*)
password:              *None*               (*None*)
url:                   *None*               (*None*)
be_object_segment_size: *None*                 (5 MiB)
restore_local:         *None*               (No)
insecure:              *None*               (No)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: user (Restore user name)
     2: password (Restore user password)
     3: url (Destination URL for restore)
     4: be_object_segment_size (DLO max segment size)
     5: restore_local (Restore as local file)
     6: insecure (Accept self-signed certificate)
Select parameter to modify (1-6): 5
Please enter a value for restore_local: yes
Plugin Restore Options
user:                  *None*               (*None*)
password:              *None*               (*None*)
url:                   *None*               (*None*)
be_object_segment_size: *None*                 (5 MiB)
restore_local:         yes                  (No)
insecure:              *None*               (No)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
...
Plugin Options:  User specified
OK to run? (yes/mod/no): yes
```

Useful Swift Commands

You may define environment variables to simplify Swift command calls.

```
# export ST_AUTH=http://swift.lan:8080/auth/v1.0
# export ST_KEY=testing
# export ST_USER=test:tester
```

Show information about a Swift account:

```
# swift stat
```

List containers

```
# swift list
```

List objects inside a container

```
# swift list container1
```

Upload a file

```
# swift upload container1 afile
```

Set a metadata for container or object

```
# swift post -m Bacula:Pass container1
# swift post -m Bacula:Pass container1 object1
```

Set an ACL data for container

```
# swift post --read-acl=www container1
# swift post --write-acl=www container1
```

For more information, please refer to: https://docs.openstack.org/python-swiftclient/latest/cli/index.html#swift-usage

Limitations

- Objects stored as SLO objects on the Swift System will be restored as DLO objects.
- The modification time of an object cannot be restored. This is a Swift limitation.
- The creation time of an object is always equal to the modification time. This is a Swift limitation.
- Bacula's Accurate backup mode is not supported. You will receive a warning message in this case.
- The regexwhere feature is not supported
- It is recommended to not set a high value for the Plugin Restore Parameter `be_object_segment_size` to limit the memory consumption.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

4.3 Big Data

Hadoop Distributed File System Plugin

The following article aims at presenting the reader with information about the **Bacula Enterprise Hadoop Distributed File System (HDFS) Plugin**. The document briefly describes the target technology of the plugin, and presents its main features.

Scope

The HDFS Plugin supports any Bacula Supported Operative System based on Linux.

This plugin is available since **Bacula Enterprise 12.4**, and needs to be deployed in a Linux host.

Features

The main feature of **Bacula Enterprise HDFS Plugin** is to offer backup and restore of any file contained in HDFS Clusters in an efficient way. The technology supports Full, Incremental, and Differential backups, and is able to perform backups with automatic snapshot management. Using the HDFS Plugin ensures protection of the information stored in Hadoop environments.

A unique characteristic of the Plugin is the ability to filter information based on date, which may be quite useful for very large systems, where old information may not be of somebody's interest, and/or where having a backup of everything could be problematic.

In order to increase user comfort, a wide range of backup filters have been incorporated. Moreover, a very useful feature of the Plugin is the ability to restore inside the original or a different HDFS filesystem, as well as to any other non-HDFS filesystem.

Also, the Plugin is integrated with Bweb, which guarantees ease of use.

See the detailed list of HDFS Plugin features:

Backup Features

- Full/Incremental/Differential backups
- Automatic snapshot management
- Backup filters:
  - Exclude directories with a specific name
  - Exclude files with a pattern
  - Include files with a pattern
  - Include files created/modified after a given time

The configuration for HDFS backups is done in a Bacula Fileset configuration file.

During a backup, the Bacula plugin will contact the Hadoop File System to generate a system Snapshot and retrieve Files one by one. During an Incremental or a Differential backup session, the Bacula File Daemon will need to read the difference between two Snapshots to determine which files should be backed up.

Restore Features

- Restore to local disk
- Restore to the same HDFS instance
- Restore to a different HDFS instance

Architecture

The Bacula Enterprise HDFS Plugin uses the Hadoop Java API to access its Distributed File System.

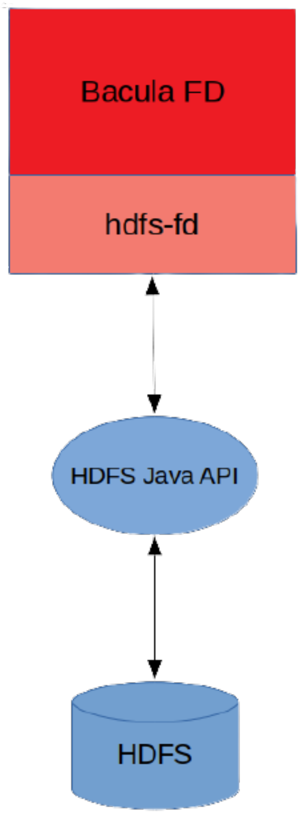


Fig. 99: HDFS Plugin Architecture

Installation

This article describes how to install Bacula Enterprise Hadoop Distributed File System (HDFS) Plugin.

Prerequisites

- The plugin is based on Java, so the Java version 8 or greater is needed
- Network access from the File Daemon where you install the plugin to the HDFS cluster nodes.

HDFS Installation with BIM

In order to install the HDFS Plugin with BIM, install the File Daemon with BIM and choose to install the HDFS Plugin during the FD installation.

For more details on the plugin installation process with BIM, click here.

HDFS Installation with Package Manager

On the Bacula File Daemon that you want to connect to your HDFS instance, extend the repository file for your package manager to contain a section for the HDFS plugin. For example, in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseHdfsPlugin]
name=Bacula Enterprise Hdfs Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/hdfs/
↪@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

On Debian Jessie, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪jessie-64/ jessie main
deb https://www.baculasystems.com/dl/@customer-string@/debs/hdfs/@version@/
↪jessie-64/ jessie hdfs
```

Once the repository is configured for your system, the package *bacula-enterprise-hdfs-plugin* can be installed with `yum install` or `apt-get install`.

```
  # yum install bacula-enterprise-hdfs-plugin
or
  # apt-get update
  # apt-get install bacula-enterprise-hdfs-plugin
```

Configuration

The following chapter presents the information on HDFS Plugin parameters, estimation and backup parameters, and restore parameters.

Plugin Parameters

The following parameters affect any type of HDFS Plugin Job (Backup, Estimation or Restore).

- `url=<string>` specifies the URL of the HDFS instance. This parameter is mandatory.
- `user=<string>` specifies the User who owns the `root path "/"` in the HDFS instance. Bacula needs to know this user in order to create snapshots in the system. This parameter is mandatory.

Plugin Estimation and Backup Parameters

- `include=<string>` specifies which files should be backed up from the HDFS System. This parameter is optional. There may be more than one `include` parameter.
- `regexinclude=<regex>` specifies, using a Regular Expression, which files should be backed up from the HDFS System. This parameter is optional. There may be more than one `regexinclude` parameter.
- `exclude=<string>` specifies which files should NOT be backed up from the HDFS System. This parameter is optional. There may be more than one `exclude` parameter.
- `regexexclude=<regex>` specifies, using a Regular Expression, which files should NOT be backed up from the HDFS System. This parameter is optional. There may be more than one `regexexclude` parameter.

If none of the optional paramaters `include`, `regexinclude`, `exclude` or `regexexclude` are specified then all files from the Hadoop File System to which the user `bacula` has access will be backed up.

Plugin Restore Parameters

- `user=<string>` specifies an account where restore will be performed. This parameter is optional. If not set, the `user` parameter from the backup Job will be used.
- `url=<string>` specifies the URL of the HDFS system during a restore. This parameter is optional. If not set, the `url=<string>` parameter from the backup Job will be used.
- `restore_local=<yes or no>` specifies that the files should be restored to a local directory based on the `where=` restore job parameter. This parameter is optional and defaults to no.

Fileset Examples

In the example below, all files inside the path `btest1` will be backed up.

```
Fileset {
 Name = FS_Hdfs
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*"
 }
}
```

In the example below, all files that do not end with `tmp` inside the path `btest1` will be backed up.

```
Fileset {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*␣
↪exclude=*tmp"
 }
}
```

This example is the same as the `exclude` one above, but using `regexexclude` instead:

```
Fileset {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*␣
↪regexclude=.*\\.tmp\\Z(?ms)"
 }
}
```

In the example below, all files that end with `.pdf` inside the path `path1` will be backed up.

```
Fileset {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*␣
↪regexinclude=.*\\.pdf\\Z(?ms)"
 }
}
```

In the example below, all files will be backed up.

```
Fileset {
  Name = FS_Hdfs_everything
  Include {
    Options {
      Compression = LZO
    }
    Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000"
  }
}
```

Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise HDFS Plugin**.

Backup

Assuming that we have the following Job configured in bacula-dir.conf:

```
JobDefs {
  Name = BackupJob
  Type = Backup
  Pool = Default
  Storage = File
  Messages = Standard
  Priority = 10
  Client=127.0.0.1-fd
  Write Bootstrap = "/home/hdev/bacula-cloud/regress/working/%n-%f.bsr"
}

Job {
  Name = PluginHdfsTest
  JobDefs = BackupJob
  Fileset= FS_Hdfs
}

Fileset {
 Name = FS_Hdfs
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*"
 }
}
```

We can run this Job using the bconsole program:

```
run job=PluginHdfsTest
Using Catalog "MyCatalog"
Run Backup job
JobName:  PluginHdfsTest
Level:    Full
Client:   127.0.0.1-fd
Fileset:  TestPluginHdfsSet
Pool:     Default (From Job resource)
Storage:  File (From Job resource)
When:     2020-04-06 12:19:10
Priority: 10
OK to run? (yes/mod/no): yes
Job queued. JobId=1
wait
You have messages.
messages
06-abr 12:29 127.0.0.1-dir JobId 1: Start Backup JobId 1, Job=PluginHdfsTest.
```

(continues on next page)

(continued from previous page)

```
↪2020-04-06_12.29.12_05
06-abr 12:29 127.0.0.1-dir JobId 1: Using Device "FileStorage" to write.
06-abr 12:29 127.0.0.1-sd JobId 1: Wrote label to prelabeled Volume
↪"TestVolume001" on File device "FileStorage" (/home/hdev/bacula-cloud/
↪regress/tmp)
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Starting HDFS Plugin Job
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Finished reading HDFS Plugin Params
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Starting backup
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Finishing HDFS Plugin Job
06-abr 12:29 127.0.0.1-sd JobId 1: Elapsed time=00:00:01, Transfer rate=3.157␣
↪K Bytes/second
06-abr 12:29 127.0.0.1-sd JobId 1: Sending spooled attrs to the Director.␣
↪Despooling 3,581 bytes ...
06-abr 12:29 127.0.0.1-dir JobId 1: Bacula 127.0.0.1-dir 12.4.0 (20Dec19):
  Build OS:               x86_64-pc-linux-gnu ubuntu 18.04
  JobId:                  1
  Job:                    PluginHdfsTest.2020-04-06_12.29.12_05
  Backup Level:           Full
  Client:                 "127.0.0.1-fd" 12.4.0 (20Dec19) x86_64-pc-linux-gnu,
↪ubuntu,18.04
  Fileset:                "TestPluginHdfsSet" 2020-04-06 12:29:10
  Pool:                   "Default" (From Job resource)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         06-abr-2020 12:29:12
  Start time:             06-abr-2020 12:29:14
  End time:               06-abr-2020 12:29:17
  Elapsed time:           3 secs
  Priority:               10
  FD Files Written:       13
  SD Files Written:       13
  FD Bytes Written:       60 (60 B)
  SD Bytes Written:       3,157 (3.157 KB)
  Rate:                   0.0 KB/s
  Software Compression:   None
  Comm Line Compression:  9.8% 1.1:1
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               no
  Volume name(s):         TestVolume001
  Volume Session Id:      1
  Volume Session Time:    1586186949
  Last Volume Bytes:      4,586 (4.586 KB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK

06-abr 12:29 127.0.0.1-dir JobId 1: Begin pruning Jobs older than 6 months .
06-abr 12:29 127.0.0.1-dir JobId 1: No Jobs found to prune.
06-abr 12:29 127.0.0.1-dir JobId 1: Begin pruning Files.
```

(continues on next page)

(continued from previous page)

```
06-abr 12:29 127.0.0.1-dir JobId 1: No Files found to prune.
06-abr 12:29 127.0.0.1-dir JobId 1: End auto prune.

list files jobid=1
Using Catalog "MyCatalog"
+-------------------------------------+
| filename                            |
+-------------------------------------+
| /@hdfs/btest1/files/A/A0A/bFa5csqF |
| /@hdfs/btest1/files/A/A0A/bEa4csqE |
| /@hdfs/btest1/files/A/A0A/bDa3csqD |
| /@hdfs/btest1/files/A/A0A/bCa2csqC |
| /@hdfs/btest1/files/A/A0A/bBa1csqB |
| /@hdfs/btest1/files/C/aCa2aaaC     |
| /@hdfs/btest1/files/F/aFa5aaaF     |
| /@hdfs/btest1/files/B/aBa1aaaB     |
| /@hdfs/btest1/files/bAa0csqA       |
| /@hdfs/btest1/files/E/aEa4aaaE     |
| /@hdfs/btest1/files/A/aAa0aaaA     |
| /@hdfs/btest1/files/D/aDa3aaaD     |
+-------------------------------------+

+-------+----------------+---------------------+------+-------+----------+----
↪------+-----------+
| jobid | name           | starttime           | type | level | jobfiles |␣
↪jobbytes | jobstatus |
+-------+----------------+---------------------+------+-------+----------+----
↪------+-----------+
|     1 | PluginHdfsTest | 2020-04-06 12:29:14 | B    | F     |       13 |   ␣
↪   60 | T         |
+-------+----------------+---------------------+------+-------+----------+----
↪------+-----------+
quit
```

Restore

To restore the backup performed in the last section into the same HDFS System, inside the directory `bacula-restores` we also use bconsole:

```
restore jobid=1 where=/bacula-restores all done yes
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
12 files inserted into the tree and marked for extraction.
Bootstrap records written to /home/hdev/bacula-cloud/regress/working/127.0.0.
↪1-dir.restore.1.bsr

The Job will require the following (*=>InChanger):
```

(continues on next page)

(continued from previous page)

```
   Volume(s)                  Storage(s)                 SD Device(s)
===========================================================================

    TestVolume001              File                       FileStorage

Volumes marked with "*" are in the Autochanger.


12 files selected to be restored.

Automatically selected Client: 127.0.0.1-fd
Using Catalog "MyCatalog"
Job queued. JobId=5
wait
You have messages.
messages
06-abr 12:29 127.0.0.1-dir JobId 5: Start Restore Job RestoreFiles.2020-04-06_
↪12.29.39_16
06-abr 12:29 127.0.0.1-dir JobId 5: Restoring files from JobId(s) 1
06-abr 12:29 127.0.0.1-dir JobId 5: Using Device "FileStorage" to read.
06-abr 12:29 127.0.0.1-sd JobId 5: Ready to read from volume "TestVolume001"␣
↪on File device "FileStorage" (/home/hdev/bacula-cloud/regress/tmp).
06-abr 12:29 127.0.0.1-sd JobId 5: Forward spacing Volume "TestVolume001" to␣
↪addr=239
06-abr 12:29 127.0.0.1-sd JobId 5: Elapsed time=00:00:01, Transfer rate=2.142␣
↪K Bytes/second
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Starting HDFS Plugin Job
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Finished reading HDFS Plugin Params
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Starting restore
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Finishing HDFS Plugin Job
06-abr 12:29 127.0.0.1-dir JobId 5: Bacula 127.0.0.1-dir 12.4.0 (20Dec19):
  Build OS:               x86_64-pc-linux-gnu ubuntu 18.04
  JobId:                  5
  Job:                    RestoreFiles.2020-04-06_12.29.39_16
  Restore Client:         127.0.0.1-fd
  Where:                  /bacula-restores
  Replace:                Always
  Start time:             06-abr-2020 12:29:41
  End time:               06-abr-2020 12:29:45
  Elapsed time:           4 secs
  Files Expected:         12
  Files Restored:         12
  Bytes Restored:         60 (60 B)
  Rate:                   0.0 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK

06-abr 12:29 127.0.0.1-dir JobId 5: Begin pruning Jobs older than 6 months .
06-abr 12:29 127.0.0.1-dir JobId 5: No Jobs found to prune.
06-abr 12:29 127.0.0.1-dir JobId 5: Begin pruning Files.
```

(continues on next page)

(continued from previous page)

```
06-abr 12:29 127.0.0.1-dir JobId 5: No Files found to prune.
06-abr 12:29 127.0.0.1-dir JobId 5: End auto prune.
```

Useful Commands

List files inside a directory

```
# hadoop fs -ls /user/hadoop/file1
```

Upload a file

```
# hadoop fs -put /local-files/file1.txt /hdfs-path
```

Upload many files

```
# hadoop fs -put /local-files /hdfs-path
```

Download many files

```
# hadoop fs -get /hdfs-path /local-path
```

For a complete set of commands and options, refer to the Hadoop documentation:

- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html
- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsSnapshots.html

Limitations

The following article presents limitations of HDFS Plugin.

- The HDFS plugin requires snapshot enabled in the HDFS file system, available from Hadoop 3.3.1. If you cannot enable snapshot, please use fuse and mount the HDFS locally on the system running the FD and the SD.
- The creation time of a file cannot be backed up.
- Empty directories and directory attributes cannot be backed up.
- Bacula's Accurate backup mode is not supported. You will receive a warning message if it is applied.
- The current implementation of the plugin cannot backup ACL and Extended Attributes.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5 Databases

Important: Database solutions are used with the File Daemon.

5.1 MSSQL

The MSSQL VDI (Virtual Device Interface) and MSSQL VSS (Volume Shadow Copy Service) Plugins for Bacula Enterprise are both designed to facilitate backups of Microsoft SQL Server databases, but they operate differently and offer distinct features.

They seamlessly integrate with Bacula Enterprise's scheduling, cataloging, and management functionalities, enabling automated and uniform backup operations. Each plugin guarantees that SQL Server databases are preserved in a consistent state, thereby mitigating the risk of data corruption and ensuring dependable restores. Furthermore, both plugins are tailored for optimal performance with Microsoft SQL Server, utilizing various technologies to meet backup and recovery objectives.

- *MSSQL VDI Plugin* offers:
- Direct interaction with SQL Server via VDI.
- Full, differential, and log backups.
- High-performance backup operations.
- Granular control over backup types and restoration processes.

Thus, MSSQL VDI Plugin is best suited for environments that are heavily focused on SQL Server, offering fine-grained control over backup types, including differential and log backups.

- *MSSQL VSS Plugin* offers:
- Volume Shadow Copy Service used for backup.
- System-wide consistency ensured across multiple services.
- Full database and log backups supported via snapshots.
- Backup management simplified in mixed application environments.

Thus, MSSQL VSS Plugin is ideal for mixed environments where multiple applications might be sharing storage, as it coordinates the backup across different services beyond just SQL Server, ensuring overall system consistency.

MSSQL VDI Plugin

Overview

This user's guide presents how to use the MSSQL Plugin feature with **Bacula Enterprise**.

Features Summary

Bacula Systems provides a plugin for Microsoft SQL Server for **Bacula Enterprise** named `mssql-fd.dll`. The MSSQL Plugin provides the following main features:

- Full and Differential support
- Incremental (Log) level support
- Database level backup
- Ability to include/exclude databases from the backup job
- Support for "Copy Only" backups
- Restore MSSQL backup files to disk
- Send the backup stream directly to the Storage Daemon without requiring much local disk space.
- *Point in time recovery restore*

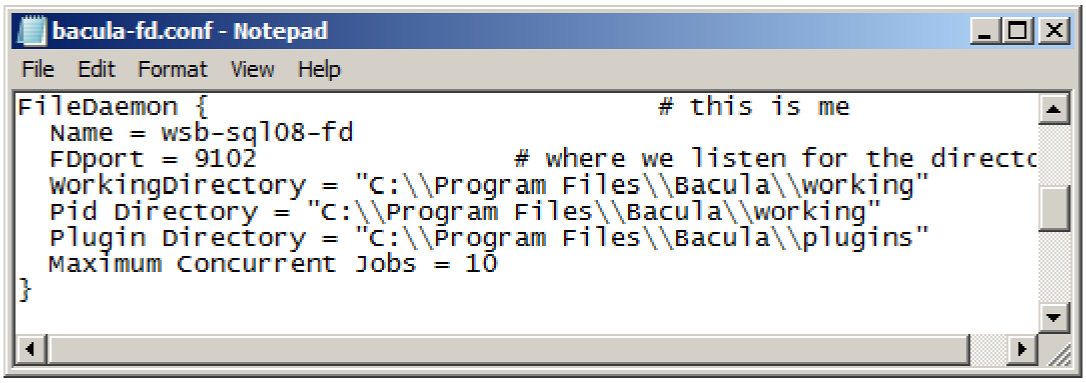
The MSSQL VDI plugin is compatible with Copy/Migration jobs. Please read the CopyMigrationJobsReplication for more information.

Scope

This document will present solutions for **Bacula Enterprise** 8.4 and later, which are not applicable to prior versions. The MSSQL Plugin has been tested and is supported from Windows Server 2003 R2 up to Windows Server 2019 and from Microsoft SQL Server 2005 up to 2019.

Installation

You must ensure that the `mssql-driver.dll` is in the Bacula program directory and the `mssql-fd.dll` plugin is in the plugins directory on the FD doing the backup (done by default with the installer), and that the `Plugin Directory` directive line is present and enabled in the FD's configuration file `bacula-fd.conf`. An example configuration file and status output of a client with the MSSQL plugin available is shown below.



```
FileDaemon {                                    # this is me
  Name = wsb-sql08-fd
  FDport = 9102                        # where we listen for the directc
  WorkingDirectory = "C:\\Program Files\\Bacula\\working"
  Pid Directory = "C:\\Program Files\\Bacula\\working"
  Plugin Directory = "C:\\Program Files\\Bacula\\plugins"
  Maximum Concurrent Jobs = 10
}
```

Fig. 100: File Daemon Configuration Excerpt with "Plugin Directory" Line

```
*status client
Automatically selected Client: win2008-fd
Connecting to Client win2008-fd at win2008-64-r2:9102

win-fd Version: 8.4.3 (30 November 2015)  VSS Linux Cross-compile Win64
Daemon started 11-Dec-15 05:13. Jobs: run=1 running=0.
Microsoft Windows Server 2008 R2 Standard Edition (build 7600), 64-bit
 Heap: heap=0 smbytes=348,013 max_bytes=481,620 bufs=134 max_bufs=162
 Sizes: boffset_t=8 size_t=8 debug=0 trace=1 mode=0,2010 bwlimit=0kB/s
 Plugin: alldrives-fd.dll mssql-fd.dll
```

If the SQL Server database is running under an account that is not `NT AUTHORIZED/SYSTEM`, it will be mandatory to configure the SQL Server instance to allow the Bacula File Daemon service account to connect and perform backup operations. By default, the Bacula File Daemon service runs under the `NT AUTHORIZED/SYSTEM` account.

The permission `sysadmin` can be granted with the following SQL command:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT AUTHORITY\SYSTEM]
```

Configuration

Note: In case of a clustered MSSQL environment, or AlwaysOn Availability Groups, the Bacula Enterprise MSSQL VDI plugin will not automatically detect the configuration. Precautions need to be taken on both the MSSQL Server and Bacula Plugin side to make sure databases are adequately protected in such an environment.

Please contact Bacula Systems (support@baculasystems.com) for help on configuring the MSSQL VDI plugin if in doubt.

To activate the MSSQL plugin you have to put the following into the Include section of the File Set which will be used to back up the SQL Server data:

```
Plugin = "mssql"
```

This will back up all SQL server databases (`tempdb` is excluded by default).

The plugin directive must be specified exactly as shown above.

If everything is set up correctly as above then the backup will include the SQL server data. The SQL server data files backed up will appear in a **bconsole** or **bat** restore as follows:

```
/@mssql/MSSQLSERVER/master/data.bak
/@mssql/MSSQLSERVER/production/data.bak
...
etc
```

It is possible to select different instances or databases to be backed up with the following parameters:

- `instance`
- `database`
- `include`
- `exclude`

Full, Differential and Logs (Incremental) backups are supported.

Following the creation of a new database, you should run a Full backup of the SQL server data. A new database will not be backed up at a different level, and a Differential backup will automatically be upgraded to a Full backup on this specific new database.

The MSSQL Plugin does not use VSS snapshots to perform the backup so, unless some disk folder is present in the fileset, **Enable VSS** can be set to "no".

A complete example of the Job setup for MSSQL Server data is shown below:

```
Fileset {
  Name = MSSQL
  Enable VSS = no      # VSS is not required
  Include {
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: database=production"
  }
```

(continues on next page)

(continued from previous page)

```
}

Job {
  Name = MSSQL08
  File Set = MSSQL
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Differential
}
```

```
Fileset {
  Name = MSSQL09
  Enable VSS = no      # VSS is not required
  Include {
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: abort_on_error exclude=test1 exclude=test2 copyonly"
  }
}

Job {
  Name = MSSQL09
  File Set = MSSQL09
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Full
}
```

```
Fileset {
  Name = MSSQL10
  Enable VSS = no      # VSS is not required
  Include {
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: include=test2 include=prod1 include=r7*"
  }
}
Job {
  Name = MSSQL10
  File Set = MSSQL10
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Full
}
```

```
Fileset {
  Name = SQLExpress
  Enable VSS = no      # VSS is not required
  Include {
```

(continues on next page)

(continued from previous page)

```
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: instance=SQLExpress hostname=."
  }
}

Job {
  Name = SQLExpress
  File Set = SQLExpress
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Differential
}
```

Attention: New in Bacula Enterprise 12.8.0

MS SQL Server Multi Instances Backup Job

```
Fileset {
  Name = SQLAllInstances
  Enable VSS = no      # VSS is not required
  Include {
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: all_instances"
  }
}

Job {
  Name = SQLAllInstances
  File Set = SQLAllInstances
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Incremental
}
```

Attention: New in Bacula Enterprise 12.8.0

MS SQL Server 2 Instances Backup Job

```
Fileset {
  Name = SQL2Instances
  Enable VSS = no      # VSS is not required
  Include {
    Options {
      Signature = SHA1
    }
    Plugin = "mssql: instance=MSSQLSERVER instance=PRODUCTION"
  }
}

Job {
  Name = SQL2Instances
  File Set = SQL2Instances
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Incremental
}
```

Plugin Options

- **database=<glob>** Specifies the name of the databases to backup. This parameter is optional. By default, all databases (except `tempdb`) will be backed up.
- **include=<glob>** Specifies the names of databases to backup. It is possible to specify the `include` parameter multiple times on the plugin command line.

  If a database that was explicitly specified is not found, a warning or error message will be printed to the Job report. The **abort_on_error** option is adhered to.
- **exclude=<glob>** Specifies the names of databases to exclude from the backup. It is possible to specify the `exclude` parameter multiple times on the plugin command line.
- **user=<str>** The username used to connect to the MSSQL instance. If the user is part of a domain, please also specify the domain parameter below. If no domain is specified the user parameter will refer to a local account. This parameter is optional, and if not set, the `bacula-fd` service account will be used to connect to the MSSQL instance.
- **domain=<str>** The domain of the user used to connect to the MSSQL instance. This parameter is optional. If not set and no user is set, the `bacula-fd` service account will be used to connect to the MSSQL instance. If not set and a user is set, the user is assumed to be a local account.
- **password=<str>** The password used to connect to the MSSQL instance. This parameter is optional, and if set, the password might be printed in various places such as `status client` output, job log or debug messages.
- **passfile=<file>** The file where the password can be found. This parameter is optional. The plugin will use the first line (limited to 127 characters) as the connection password.
- **authtype=[windows | server]** The authentication type used to connect to the MSSQL instance. This parameter is optional. By default the `Windows` authentication type is used.
- **instance=<str>** The instance name used to connect to the MSSQL instance. This parameter is optional. By default, the instance name is “MSSQLSERVER”.

  > **Attention: New in Bacula Enterprise 12.8.0**
  >
  > Multiple `instance` parameters are allowed.

- > **Attention: New in Bacula Enterprise 12.8.0**

all_instances This parameter is optional. If set, the MSSQL plugin will list and backup all instances defined on the SQL Server. The same authentication settings should be available on each instance.

- **hostname=<str>** This parameter is optional. If set, the `hostname` string will be used in the `Server` connection string of the ODBC driver. If not set, the plugin will configure automatically the ODBC driver to use `(localdb)` or `(local)`. This advanced option might be used in conjunction with the `instance` option, often, it is necessary to specify "`.`" as `hostname` parameter. Ex: `instance=MYINSTANCE hostname=.`
- **abort_on_error** By default, if the plugin is not able to reach the MSSQL instance, or to find an explicitly named database, an error will be generated and the job will continue. Some users might prefer to abort the Job, which will happen if this option is set.
- **copyonly[=incremental|all]** A copy-only backup is a MSSQL backup that is independent of the sequence of conventional MSSQL backups. The next Differential or Transaction Log (Incremental) backup will not use it. If the `incremental` option is set, only Transaction Log (Incremental) backups will use the option. The last Incremental job will contain all the information necessary to perform a point in time recovery.
- **fullbackup** Force the level of the MSSQL backup. If specified in the plugin command line, a job can run with the incremental level for standard files, and the databases backed up with the plugin will always be backed up like with a Full backup. (Available since 8.4.11)
- **skipreadonly** Skip read only databases when the backup level is incremental (BACKUP LOG) and the last modification/creation time for tables and views is prior to the last backup. By default, read only databases are upgraded to Full. (Available since 8.6.20)
- **dblayout** Store each database layout as a RestoreObject in the Bacula Catalog with a Full backup. The layout can be displayed in `bconsole` with `list restoreobjects` command. (Available since 8.6.20)
- **lock_timeout=<int>** Use the `SET LOCK_TIMEOUT` before issuing queries. (Available since 8.6.20)
- **buffercount=<int>** Specifies the total number of I/O buffers to be used for the backup operation. You can specify any positive integer; however, large numbers of buffers might cause "out of memory" errors because of inadequate virtual address space in the Sqlservr.exe process. This parameter is optional and the default value is 10.
- **blocksize=<bytes>** Specifies the physical block size, in bytes. The supported sizes are 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (64 KB) bytes. This parameter is optional and the default is 65536.
- **maxtransfersize=<bytes>** Specifies the largest unit of transfer in bytes to be used between the MSSQL Server and the backup media. The possible values are multiples of 65536 bytes (64 KB) ranging up to 4194304 bytes (4 MB). This parameter is optional and the default is 65536.
- **connection_string=<str>** Specifies the ODBC connection string. This parameter is optional.
- **checksum=<No/Yes>**

Available since Bacula Enterprise 8.11.6

Use the `WITH CHECKSUM` option in the backup query. This parameter is optional and the default is no.

- **driver=<str>**

Available since Bacula Enterprise 10.2.3

The driver name used in the ODBC connection string. Default is SQL Server. If the `connection_string` option is set, it will overwrite this option.

- **target_backup_recovery_models=<str>**

  Available since Bacula Enterprise 10.2.4

  Only the databases with the specified recovery model are backuped. The other databases are skipped with warning. The parameter should be on of the following:

  - `SIMPLE`
  - `FULL`
  - `BULK_LOGGED`

- **simple_recovery_models_incremental_action=<str>**

  Available since Bacula Enterprise 10.2.4

  Specifies the behavior when incremental backup is requested over a simple recovery model database from the following parameters:

  - `make_full` : The incremental backup is automatically upgraded to full.
  - `ignore_with_error` : The backup is skipped with an error (the error translated to a warning at the job level).
  - `ignore` : The backup is skipped silently.

Backup

Full Backup

The Full backup saves the database files and MSSQL to provide complete protection against media failure. If one or more data files are damaged, media recovery can restore all committed transactions. In-process transactions are rolled back. The master and the mbdb databases are always backed up in this mode.

Differential Backup

A differential backup is based on the most recent, previous full database backup. A differential backup captures only the data that has changed since that full backup. When using the Differential backup feature, the backup chain is very critical. If for some reason, the Full backup used as referenced by MSSQL is not available, the Differential data will not be usable. The plugin uses different techniques to avoid this problem, so if a problem is detected, the Differential database backup might be automatically upgraded to a Full backup.

Transaction Log Backup

The "Transaction Log Backup" MSSQL feature is implemented as the "Incremental" level with Bacula. The database must be configured with the full recovery model or bulk-logged recovery model. If the database uses the simple recovery model, the MSSQL file will be truncated after each checkpoint. The full restore will be possible, but not the restore to a point in time. For more information, see https://msdn.microsoft.com/en-us/library/ms189275.aspx.

Point in Time Restore (PITR)

Point In Time Restore (PITR) requires the database to be configured with the full recovery model. If the database uses the simple recovery model, the MSSQL file will be truncated after each checkpoint. For more information, see https://msdn.microsoft.com/en-us/library/ms189275.aspx.

MSSQL Database Configuration

The master database must be backed up. If master is damaged in some way, for example because of media failure, an instance of MSSQL may not be able to start. In this event, it is necessary to rebuild master, and then restore the database from a backup. Only full database backups of master can be created. See https://technet.microsoft.com/en-us/library/aa213839%28v=sql.80%29.aspx for more information.

Restore

You can use all the regular ways to start a restore. However, you must make sure that if restoring differential data, the previous full backup is also restored. This happens automatically if you start the restore, in `bconsole`, using the restore options 5 or 12. In the file tree generated, you should mark either complete databases or databases instances.

Restore Options

It is possible to restore the data with different scenarios using the following options:

- The common restore "Where" parameter (**where=<path>**)
- The common restore "Replace" parameter (**replace=<never|always>**)
- The Plugin restore option accessible in the "Plugin Options" menu at the restore prompt (Item 13).

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.9.bsr
Where:           c:/tmp
Replace:         Never
Fileset:         Full Set
Backup Client:   win2008-fd
Restore Client:  win2008-fd
Storage:         File
When:            2016-02-22 12:02:56
Catalog:         MyCatalog
Priority:        10
```

(continues on next page)

(continued from previous page)

```
Plugin Options:  *None*
OK to run? (yes/mod/no): mod                       <--------------------
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13              <------------------
Automatically selected : mssql: database=db29187
Plugin Restore Options
instance:             *None*
database:             *None*
username:             *None*
password:             *None*
domain:               *None*
hostname:             *None*
authtype:             *None*
recovery:             *None*               (yes)
stop_before_mark:     *None*
stop_at_mark:         *None*
stop_at:              *None*
restricted_user:      *None*               (no)
verify:               *None*               (no)
connection_string:    *None*
checksum:             *None*               (no)
driver:               *None*               (SQL Server)

Use above plugin configuration? (yes/mod/no): mod  <-------------------
You have the following choices:
You have the following choices:
     1: instance (Instance used to restore)
     2: database (New database name)
     3: username (Username used for restore)
     4: password (Password used for restore)
     5: domain (Domain name of user (default to local))
     6: hostname (Server ODBC parameter (default to (local/localdb)))
     7: authtype (Authentication type (server or windows))
     8: recovery (Start Recovery)
     9: stop_before_mark (Stop the recovery before a mark (STOPBEFOREMARK).
↪{lsn:lsn_number | mark_name})
    10: stop_at_mark (Stop the recovery at a mark (STOPATMARK). {lsn:lsn_
↪number | mark_name})
    11: stop_at (Stop at (STOPAT). {datetime})
```

(continues on next page)

(continued from previous page)

```
    12: restricted_user (Restrict access to the restored database)
    13: verify (Verify backup integrity)
    14: connection_string (ODBC Connection string)
    15: checksum (Restore using the WITH CHECKSUM option)
    16: driver (ODBC Driver Name)

Select parameter to modify (1-15):
```

The Plugin restore options are:

- **instance=<str>** The instance name used to connect to the MSSQL instance. This parameter is optional, and if not set, the restore will use the value set during at the backup time. By default, the instance name is “MSSQLSERVER”.
- **database=<name>** Specifies the name of the databases to restore. This parameter is optional. By default, the plugin will use the `where` option to determine the name of the new database. If both `where` and `database` are set to a valid database name, `database` will be used. A valid database name can contain the following characters: `A-Za-z0-9#_`
- **username=<str>** The username used to connect to the MSSQL instance. This parameter is optional, and if not set, the restore will use the value set during at the backup time.
- **password=<str>** The password used to connect to the MSSQL instance. This parameter is optional and if not set, the restore will use the value set during at the backup time.
- **domain=<filestr>** The domain used to connect to the MSSQL instance. This parameter is optional and if not set, the restore will use the value set during at the backup time.
- **hostname=<str>** The MSSQL server host name.
- **authtype=[windows | server]** The authentification type. Windows is default.
- **recovery=<Yes/no>** Specifies if the database will use the RECOVERY or the NORECOVERY option during the restore. By default, the restored database will be recovered.
- **stop_before_mark=<markname>** Use the WITH STOPBEFOREMARK = ’<point>’ clause to specify that the log record that is immediately before the mark is the recovery point. The point can be a LSN number or a mark_name.
- **stop_at_mark=<markname>** Use the WITH STOPATMARK = ’<point>’ clause to specify that the marked transaction is the recovery point. STOPATMARK rolls forward to the mark and includes the marked transaction in the roll forward. The point can be a LSN number or a mark_name.
- **stop_at=<datetime>** Use the WITH STOPAT = ’<datetime>’ clause to specify that the date time is the recovery point.
- **restricted_user=<No/yes>** Use the WITH RESTRICT_USER clause to restrict access to the restored database. The default is no.
- **connection_string=<str>** Specifies the ODBC connection string. This parameter is optional.
- **checksum=<No/yes>**

  Available since Bacula Enterprise 8.11.6

  Use the WITH CHECKSUM clause to the restored database. The default is no.

- **driver=<str>**

Available since Bacula Enterprise 10.2.3

The driver name used in the ODBC connection string. Default is SQL Server. If the `connection_string` option is set, it will overwrite this option.

On BWeb Management Suite, the Plugin Options are available in the restore tab.

Point In Time Restore

This topic is relevant only for SQL Server databases that use the full or bulk-logged recovery models. Under the bulk-logged recovery model, if a log backup contains bulk-logged changes, point-in-time recovery is not possible to a point within that backup. The database must be recovered to the end of the MSSQL backup.

More information can be found on https://msdn.microsoft.com/en-us/library/ms179451.aspx

It is possible to do Point In Time Restore of a MSSQL database directly from the MSSQL Plugin. It is also possible to restore files locally and do the operation from the Microsoft SQL Server Mangement Console to have more options.

LSN Information

LSNs are used internally during a RESTORE sequence to track the point in time to which data has been restored. When a backup is restored, the data is restored to the LSN corresponding to the point in time at which the backup was taken. More information can be found on https://msdn.microsoft.com/en-us/library/ms190925.aspx.

The LSN of a log record at which a given backup and restore event occurred is viewable using one or more of the following:

- Bacula Backup job output
- Log file names
- msdb.backupset table
- msdb.backupfile table

During a backup job with MSSQL Plugin, the following information about
LSN numbers will be displayed in the Job output:

```
win-fd JobId 3: LSN for "db29187": First: 42000146037, Last: 44000172001
```

The **First LSN** number corresponds to the last LSN of the last transaction logs backup. It can be the very first Full backup, or the last transactional backup (Incremental). The **Last LSN** number corresponds to the last transaction recorded in the log.

With a MSSQL backup (Incremental), the file name associated with this database in the Incremental job will be:

```
/@mssql/MSSQLSERVER/db29187/log-42000162001.trn
```

The number in the name, here **42000162001** corresponds to the last LSN of the previous job (Full or Incremental).

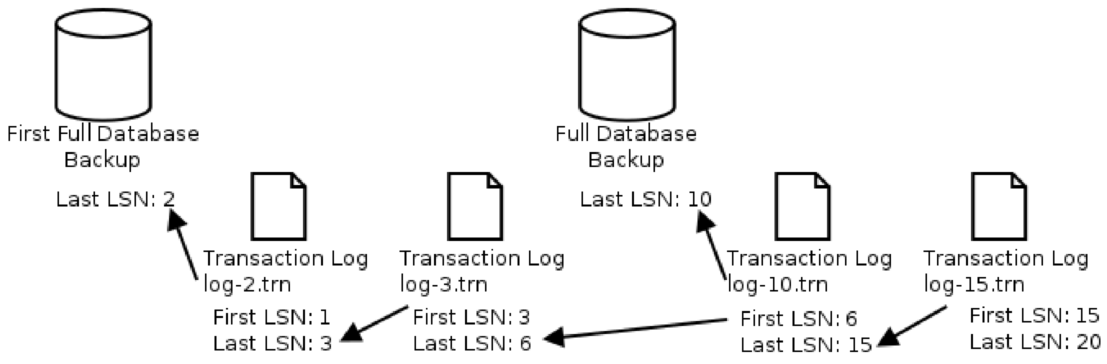


Fig. 101: First, Last and Filename LSNs

In the example shown above, if the administrator needs to restore the database at the state that corresponds to LSN 14, it can be done with the following actions:

- Use restore menu option 5
- Browse the database directory "/@mssql/db29187"
- Select last Full backup file "data.bak" (LSN: 10)
- Select incremental backup "log-10.trn"
- Specify the stop_at_mark option to "LSN:14"
- Run the restore job

or if the last full backup is not available but the previous full backup is.

- Use restore menu option 3, select the relevant jobids
- Browse the database directory "/@mssql/db29187"
- Select Full backup file "data.bak" (LSN: 2)
- Select incremental backups "log-2.trn", "log-3.trn", "log-10.trn"
- Specify the stop_at_mark option to "LSN:14"
- Run the restore job

Restore Scenarios Overview

Table 34: Restore Scenarios

Description	wher	regexwl	databa	Example
Restore files **to disk**	Path			`where=c:/tmp`
Restore original database to MSSQL				`where=/`
Restore with a new name to MSSQL	Name			`where=newdb`
Restore with a new name to MSSQL			Name	`database=newdb`
Restore with a new name and file relocation to MSSQL	Path		Name	`where=c:/tmp` `database=newdb`
Restore with a new name and individual file relocation to MSSQL		Regex	Name	`regexwhere=!CLUSTER!MSSQLSERVER!` `database=newdb`

Restore Using Advanced Relocation

In some cases, a database can use different disks to store the data.

```
c:\data\db1.MDF
e:\logs\db1_log.LDF
```

Available since Bacula Enterprise 8.6.17

Using the Bacula `RegexWhere` function, it is possible to manipulate each file and generate new filenames. The destination directories must exist prior to the restore.

In the following example, the database will be renamed and the path will be adapted to the new server.

```
db1                 ->  db1-old                     !db1!db1-old!i
c:\data\db1.MDF     ->  f:\data\db1-old.MDF         !c:!f:!i
e:\logs\db1_log.LDF ->  g:\data\db1-old_log.MDF     !e:!g:!i
```

The resulting `RegexWhere` will be something like:

```
 *restore regexwhere="!db1!db1-old!i,!c:!f:!i,!e:!g:!i"
...
  database:            db1-old
```

Note that the **database** name must be specified in the Plugin Options menu, else, files will be stored on disk and the SQL restore commands will have to be executed manually.

To convert a path, the windows path separator must be escaped correctly in the console:

```
 *restore regexwhere="!c:\\\\data\\\\db1!c:\\data2\\db1!i"
or
 *restore regexwhere="!c:.data.db1!c:/data2/db1!i"
or
 *restore regexwhere="!c:\\\\data\\\\db1!c:/data2/db1!i"
```

Restore With Same Name

To restore a database with the same name, the `where` parameter should be empty or "/" and the `replace=` flag should be set to `always` or the original database should be dropped first.

```
* restore where=/ replace=always
...
Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/127.0.0.1-dir.restore.1.bsr
Where:           /
Replace:         Always
Fileset:         Full Set
Backup Client:   win2008-fd
Restore Client:  win2008-fd
```

(continues on next page)

(continued from previous page)

```
Storage:         File
When:            2015-12-14 09:53:36
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):
```

Restore Database With a New Name

To restore a database with a new name, it might be required to relocate database files on disk. It depends if the original database is still present.

If the original database is no longer available, the `where` parameter or the "Plugin Options" `database` can contain the new database name, and the plugin will automatically handle the database creation with the new name.

If the original database is still required, the `where` parameter is used to relocate files on disk, and the new database name should be be set with the "Plugin Options" menu with the `database` option. The `layout.dat` must be selected in the restore tree.

```
* restore where=c:/tmp replace=always
...
Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/127.0.0.1-dir.restore.1.bsr
Where:           c:/tmp
Replace:         Always
Fileset:         Full Set
Backup Client:   win2008-fd
Restore Client:  win2008-fd
Storage:         File
When:            2015-12-14 09:53:36
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): mod                       <------------------
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
```

(continues on next page)

(continued from previous page)

```
Select parameter to modify (1-13): 13           <-----------------
Automatically selected : mssql: database=db29187
Plugin Restore Options
instance:            *None*
database:            *None*
username:            *None*
password:            *None*
domain:              *None*
recovery:            *None*              (yes)
stop_before_mark:    *None*
stop_at_mark:        *None*
stop_at:             *None*
Use above plugin configuration? (yes/mod/no): mod  <------------------
You have the following choices:
   1: instance (Instance used to restore)
   2: database (New database name)
   3: username (Username used for restore)
   4: password (Password used for restore)
   5: domain (Domain name of user (default to local))
   6: recovery (Start Recovery)
   7: stop_before_mark (Stop the recovery before a mark (STOPBEFOREMARK).
   8: stop_at_mark (Stop the recovery at a mark (STOPATMARK).
   9: stop_at (Stop at (STOPAT). {datetime})
Select parameter to modify (1-9): 2             <------------------
Please enter a value for database: newdb        <------------------
Use above plugin configuration? (yes/mod/no): yes  <------------------

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/127.0.0.1-dir.restore.1.bsr
Where:           c:/tmp
Replace:         Always
Fileset:         Full Set
Backup Client:   win2008-fd
Restore Client:  win2008-fd
Storage:         File
When:            2015-12-14 09:53:36
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User Specified
OK to run? (yes/mod/no): yes                    <-----------------
```

Restore to Local Disk

When specifying `where=c:/path/`, files will be restored to the local filesystem and the MSSQL administrator can use a TSQL or the Microsoft SQL Server Mangement Console to restore the database. SQL commands needed to restore the database are printed in the Job output as showed in the next example.

```
* restore where=c:/tmp

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Cancel
Select item:  (1-13): 5
Automatically selected Client: win2008-fd
+-------+-------+----------+----------+---------------------+---------------+
| jobid | level | jobfiles | jobbytes | starttime           | volumename    |
+-------+-------+----------+----------+---------------------+---------------+
|     1 | F     |        3 |   65,771 | 2015-12-14 09:52:31 | TestVolume001 |
|     2 | I     |        2 |   65,771 | 2015-12-14 09:52:42 | TestVolume001 |
|     3 | I     |        2 |   65,771 | 2015-12-14 09:52:52 | TestVolume001 |
+-------+-------+----------+----------+---------------------+---------------+
You have selected the following JobIds: 1,2,3

Building directory tree for JobId(s) 1,2,3 ...
6 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @mssql
cwd is: /@mssql/
$ cd MSSQLSERVER
cwd is: /@mssql/MSSQLSERVER/
$ m db1684
6 files marked.
```

(continues on next page)

(continued from previous page)

```
$ done
Bootstrap records written to /opt/bacula/working/127.0.0.1-dir.restore.1.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    TestVolume001             File                      FileStorage

Volumes marked with "*" are in the Autochanger.


2 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/127.0.0.1-dir.restore.1.bsr
Where:           /tmp
Replace:         Always
Fileset:         Full Set
Backup Client:   win2008-fd
Restore Client:  win2008-fd
Storage:         File
When:            2015-12-14 09:53:36
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=6
wait
You have messages.
* messages

$ done

17:18 dir JobId 6: Start Restore Job RestoreFiles.2015-12-14_17.18.18_14
17:18 dir JobId 6: Using Device "FileStorage" to read.
17:18 sd JobId 6: Ready to read from volume "TestVolume001" on file device
↪"FileStorage" (/tmp/regress/tmp).
17:18 sd JobId 6: Forward spacing Volume "TestVolume001" to file:block 0:224.
17:18 fd JobId 6: RESTORE DATABASE [db1684] FROM DISK='c:/tmp/@mssql/
↪MSSQLSERVER/db1684/data.bak'
                    WITH BLOCKSIZE=65536, BUFFERCOUNT=10,␣
↪MAXTRANSFERSIZE=65536, NORECOVERY , REPLACE
17:18 fd JobId 6: RESTORE DATABASE [db1684]
                     FROM DISK='c:/tmp/@mssql/MSSQLSERVER/db1684/log-
↪34000000014400001.bak'
                     WITH BLOCKSIZE=65536, BUFFERCOUNT=10,␣
↪MAXTRANSFERSIZE=65536, NORECOVERY
17:18 fd JobId 6: RESTORE DATABASE [db1684]
                     FROM DISK='c:/tmp/@mssql/MSSQLSERVER/db1684/log-
```

(continues on next page)

(continued from previous page)

```
↪340000000018400001.bak'
                     WITH BLOCKSIZE=65536, BUFFERCOUNT=10,␣
↪MAXTRANSFERSIZE=65536, NORECOVERY
17:18 fd JobId 6: RESTORE DATABASE [db1684]
                     FROM DISK='c:/tmp/@mssql/MSSQLSERVER/db1684/log-
↪340000000029100001.bak'
                     WITH BLOCKSIZE=65536, BUFFERCOUNT=10,␣
↪MAXTRANSFERSIZE=65536, NORECOVERY
17:18 sd JobId 6: End of Volume at file 0 on device "FileStorage" (/tmp/
↪regress/tmp), Volume "TestVolume001"
17:18 fd JobId 6: RESTORE DATABASE [db1684]
                     FROM DISK='c:/tmp/@mssql/MSSQLSERVER/db1684/log-
↪360000000017200001.bak'
                     WITH BLOCKSIZE=65536, BUFFERCOUNT=10,␣
↪MAXTRANSFERSIZE=65536, NORECOVERY
17:18 sd JobId 6: Elapsed time=00:00:01, Transfer rate=9.372 M Bytes/second
17:18 fd JobId 6: RESTORE DATABASE [db1684]
17:18 dir JobId 6: Bacula dir 8.4.8 (22Feb16):
  Build OS:               x86_64-unknown-linux-gnu archlinux
  JobId:                  6
  Job:                    RestoreFiles.2015-12-11_17.18.18_14
  Restore Client:         win2008-fd
  Start time:             14-Dec-2015 17:18:20
  End time:               14-Dec-2015 17:18:22
  Files Expected:         6
  Files Restored:         6
  Bytes Restored:         9,371,785
  Rate:                   4685.9 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Restore the "master" Database

Instructions on how to restore the "master" database are detailed in this article: https://technet.microsoft.com/en-us/library/aa213839%28v=sql.80%29.aspx

Database in *restoring* State

At the end of a restore, if the plugin option `recovery` was set to `no`, the restored database will be in the "restoring" state. To end the restore process, the recovery process must be run. It can be done with the following SQL command:

```
RESTORE [yourdatabase] WITH RECOVERY;
```

Always On Availability Groups

Availability groups can be created in MSSQL to provide high data availability over WSFC cluster nodes, as detailed in this article: https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server?view=sql-server-2017

It can be interesting depending on the system scale to perform backups on secondary replicas to free up time and resource costs from the primary replica. The Bacula MSSQL Plugin is compatible with this type of architecture, with some restrictions.

Backup

When avalability group(s) are defined, automated backup preference should also be specified to indicate the type of backup strategy (prefer secondary, primary only, etc.). The MSSQL plugin enforces this preference, only allowing backup of the prefered replica.

Note that depending on replica, not all backup type are allowed. Secondary will support copy-only full database backups and log backups while differential is not supported. This needs to be taken in account when creating Bacula's backup Jobs. Refer to this article for more details: https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/active-secondaries-backup-on-secondary-replicas-always-on-availability-groups?view=sql-server-2017

Restore

Database restores can not be performed when they are part of one or many availability groups. This is detected by the plugin and the database restore will be skipped when this is the case. To workaround this, you need to remove the database from the availability group(s), then restore the database and eventually put it back in its original availability group(s). More details can be found here: https://social.technet.microsoft.com/wiki/contents/articles/28148.restoring-a-backup-database-to-an-availability-group-sql-server.aspx

Limitations

The current version of the plugin has the following limitations:

- The plugin can have only one Plugin command line in the Fileset. To backup multiple databases in the same Job, it is possible to use the **include** parameter multiple times on the plugin command string.
- To restore a database with a new name and a new location, the `layout.dat` file must be selected in the restore process. The relocation function will not work correctly if data files were added after the last Full backup job.
- It is not possible to restore multiple databases with a new name in a single restore job. The restore should be performed in different restore jobs.
- Database snapshots are automatically excluded from the backup.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

These limitations will be addressed in a future version.

MSSQL VSS Plugin

Attention: The MSSQL VSS plugin is deprecated

Support for the MSSQL VSS plugin will stop in September 2022. All instances of Microsoft SQL Server should use the mssqlvdi-plugin for future backups.

Please do not run any differential backup with the MSSQL VSS plugin.

In order to access the MSSQL VDI plugin, please contact our Support Team. The mssqlvss-to-mssqlvdi will guide you through the migration process.

Overview

This white paper presents how to use the Microsoft Windows VSS plugin's SQl Server support with **Bacula Enterprise** version 6.0 or newer. These solutions are not applicable to prior versions. This document is intended to be used by **Bacula Enterprise** administrators.

Bacula Windows VSS Plugin

Bacula Systems provides a single plugin for Bacula Enterprise named `vss-fd.dll` that permits you to backup a number of different components on Windows machines. One of those components is Microsoft SQL Server (MSSQL), which is the subject of this white paper.

Backing up and restoring MSSQL databases is supported with Full and Differential level backups. It is not possible to do Incremental backups because Microsoft does not support that backup level for the SQL Server product.

To activate the MSSQL component you have to put the following into the Include section of the File Set which will be used to back up the SQL Server data:

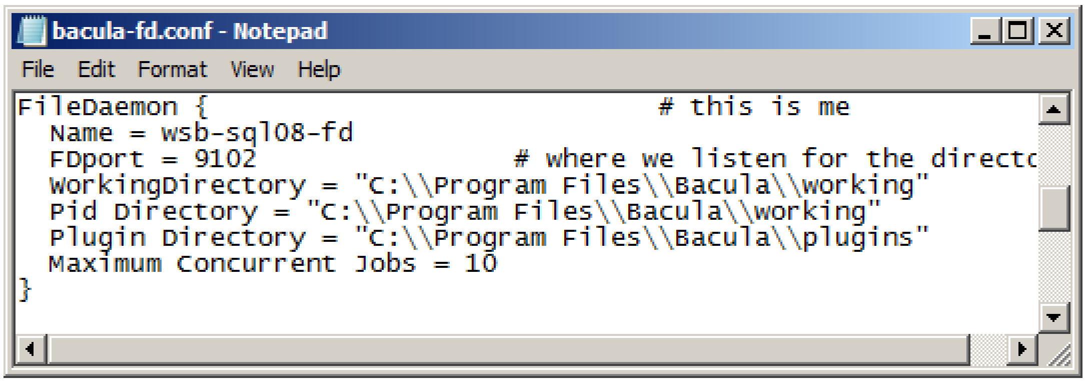
```
Plugin = "vss:/@MSSQL/"
```

This will back up all SQL server data except for those databases owned by Sharepoint, if that VSS plugin component is also specified.

The plugin directive must be specified exactly as shown above. A Job may have one or more of the vss plugin components specified.

You must ensure that the `vss-fd.dll` plugin is in the plugins directory on the FD doing the backup (done by default with the installer), and that the `Plugin Directory` directive line is present and enabled in the FD's configuration file `bacula-fd.conf`. An example configuration file is shown in figure *File Daemon*

Configuration Excerpt with "Plugin Directory" Line. The status output of a client with the VSS plugin available is shown in figure *Status of a File Daemon with VSS Plugin Available*.



```
FileDaemon {                                     # this is me
  Name = wsb-sql08-fd
  FDport = 9102                         # where we listen for the directo
  WorkingDirectory = "C:\\Program Files\\Bacula\\working"
  Pid Directory = "C:\\Program Files\\Bacula\\working"
  Plugin Directory = "C:\\Program Files\\Bacula\\plugins"
  Maximum Concurrent Jobs = 10
}
```

Fig. 102: File Daemon Configuration Excerpt with "Plugin Directory" Line

```
*status client=wsb-sql08-fd
Connecting to Client wsb-sql08-fd at wsb-sql08:9102

wsb-sql08-fd Version: 6.0.1 (02 Apr 2012)  VSS Linux Cross-compile Win64
Daemon started 20-Apr-12 13:14. Jobs: run=15 running=0.
Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1 (build ↙
→7601), 64-bit
 Heap: heap=0 smbytes=1,061,455 ...
 Sizes: boffset_t=8 size_t=8 debug=0 ...
 Plugin: alldrives-fd.dll delta-fd.dll vss-fd.dll
```

Fig. 103: Status of a File Daemon with VSS Plugin Available

BMR and VSS

The VSS plugin will not work correctly during a Bare Metal Recovery because Microsoft does not include the VSS infrastructure in WinPE, the environment used during Bare Metal Recovery. As a consequence, any backup you do with the VSS plugin cannot be used for a Bare Metal Recovery. To have a good backup for BMR purposes, you must run the Bacula FD **without** using the `Plugin` = directive in your Fileset (i.e. the plugin must not be used).

Backup

If everything is set up correctly as above then the backup will include the SQL server data. The SQL server data files backed up will appear in a bconsole or bat restore like:

```
/@MSSQL/
...
etc
```

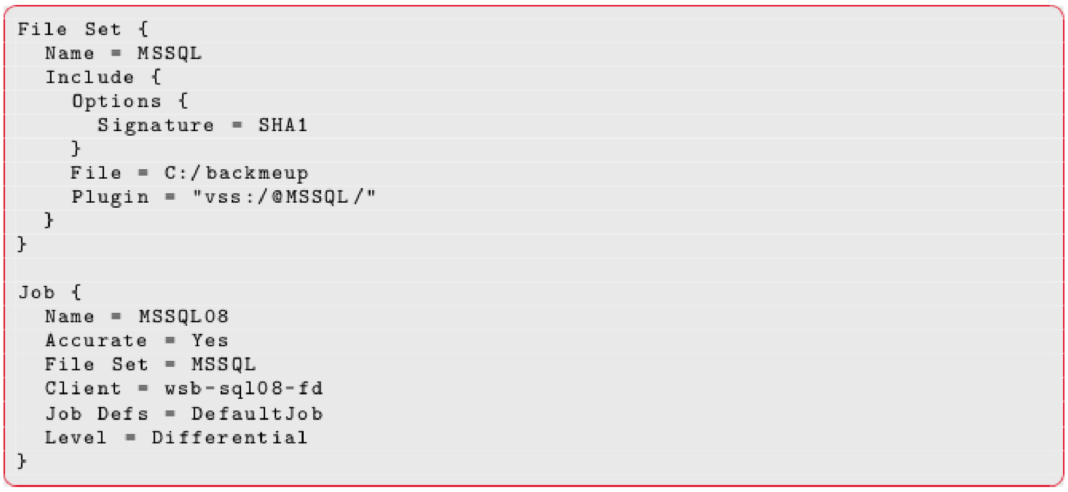
Only a backup of the complete SQL server data is supported, i. e. all database server instances, all databases and all tables are being backed up. It is not currently possible to back up only parts of the SQL server data like certain server instances, databases, or tables.

Both Full and Differential backups are supported. Microsoft does not support Incremental backups for MSSQL. If you run one, you will get errors.

Following the creation of a new database, you should run a Full backup of the SQL server data. A new database will not be backed up until a Full backup has been done, and Differential backups will fail on this specific new database.

As Windows does not update the time and date of modification of all SQL server files, you **must** use the **Accurate** Job option if you want correct Differential backups. If you do not use this option, the plugin will print a warning message, and restores will probably fail.

A complete example of the Job setup for MS SQL Server data looks as shown in figure *MS SQL Server Backup Job*. As for all VSS-enabled components, it is the administrator's responsibility to make sure that the required VSS snapshots are created by explicitly mentioning at least one file or directory for each drive where data that is handled by the plugin is stored. In the example, we use the file `c:/backmeup` to ensure this.



```
File Set {
  Name = MSSQL
  Include {
    Options {
      Signature = SHA1
    }
    File = C:/backmeup
    Plugin = "vss:/@MSSQL/"
  }
}

Job {
  Name = MSSQL08
  Accurate = Yes
  File Set = MSSQL
  Client = wsb-sql08-fd
  Job Defs = DefaultJob
  Level = Differential
}
```

Fig. 104: MS SQL Server Backup Job

Note the inclusion of `c:/backmeup`, which is required to ensure that **Bacula** creates the required VSS snapshot of the drive the backed up data is stored on. If SQL Server data is also stored on other drives, you need to create similar `File =`-lines for these drives, too[1].

```
File Set {
  Name = MSSQL-TestDB
  Include {
    Options {
      Signature = SHA1
    }
    File = C:/backmeup
    # backup only TestDB on the server
    Plugin = "vss:/@MSSQL/ cinclude=*/TestDB cexclude=*"
  }
}
```

In this example, only the database TestDB will be included in the backup. Use of multiple `cinclude` parameters is possible in the Plugin command line.

[1] Starting with version 12.5, specifying the volumes is not mandatory anymore

Restore

To restore you can use all regular ways to start a restore. However, you must make sure that, if restoring differential data, the previous full backup is also restored. This happens automatically if you start the restore, in `bconsole`, using the restore options 5 or 12. In the file tree generated, you should either mark complete database instances or databases. Note that it is important **not** to include master databases in a restore as those have to be handled specially. We show an example of that below.

Additionally, a database must be restored to the MSSQL server from which is has been backed up; restoring to another machine is not possible.[2]

The **master** database cannot be restored while the MSSQL server is running. Thus you must take care not to select the master database for a regular restore. If you do, the restore will fail. Accordingly, we provide instructions how to restore the **master** database of a MSSQL server instance, see chapter *Restoring master Databases*.

All other databases can be restored while the system is running. The Windows VSS writer will automatically unmount each database before it is restored, then apply all the outstanding log files if any exist, and finally remount the database.

It is important to understand that, using relocation of the restored files, data can be restored to a different and even a new database. This may be useful, but it may also be unintended. We recommend to make sure that any relocation is either turned off, or the effects are well understood. See figure *Relocated Database* for an example of how things may look when relocating to a default location.

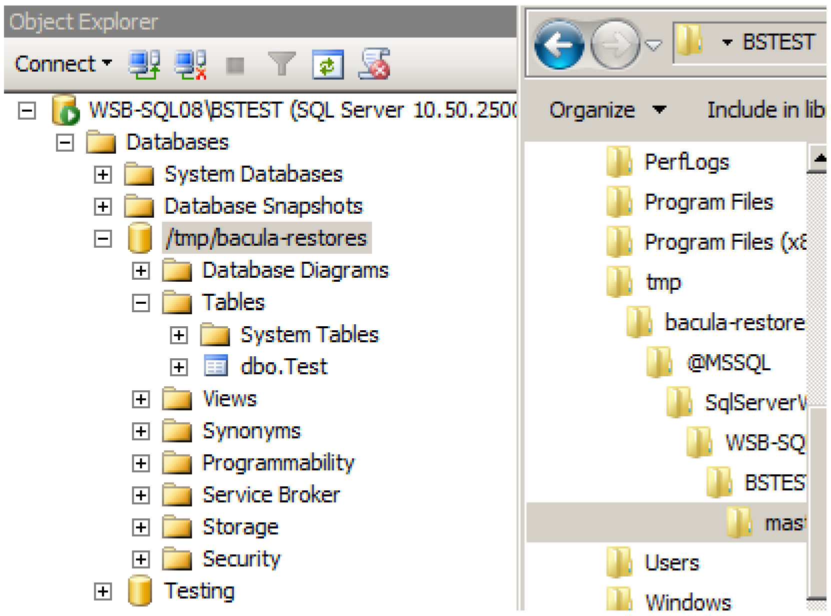
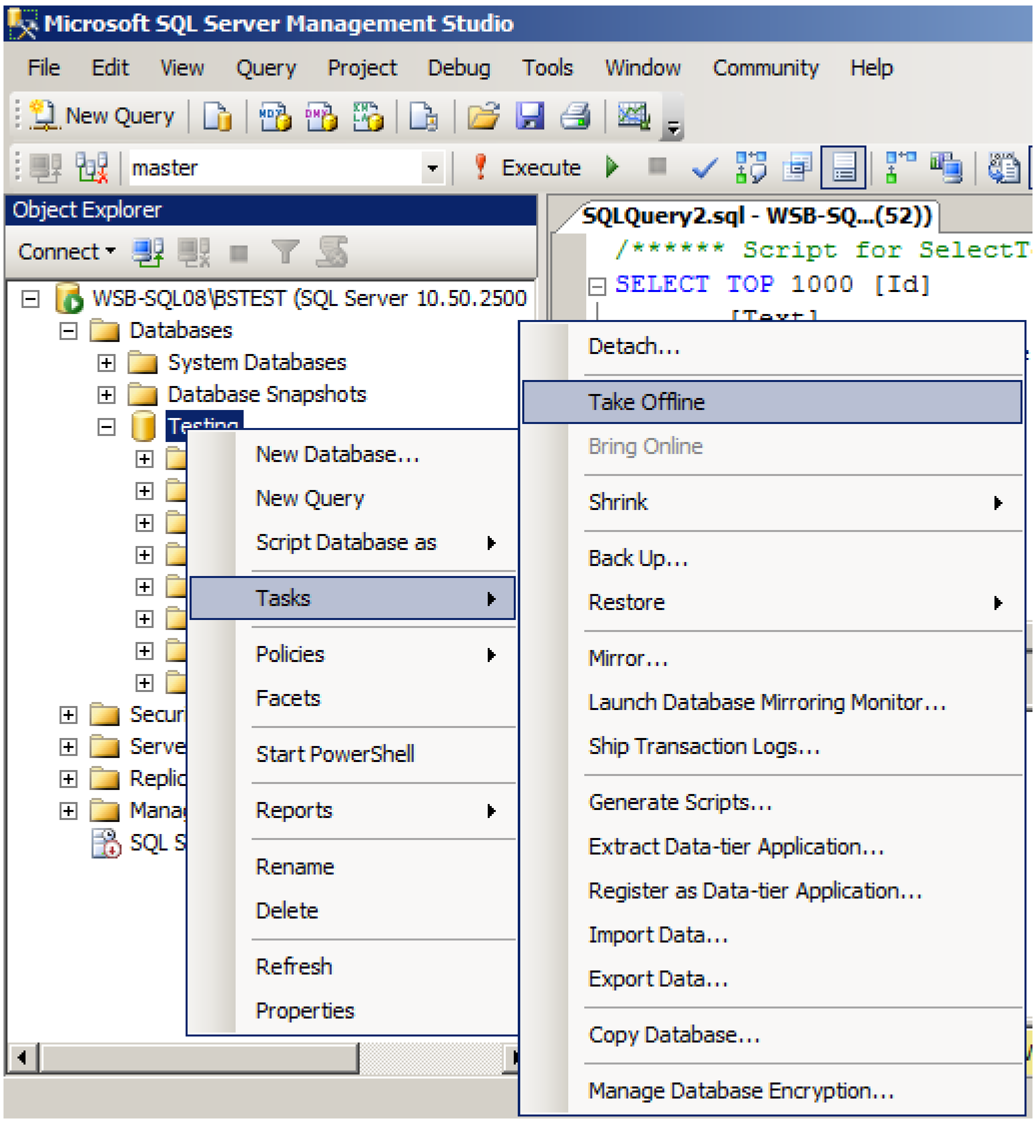


Fig. 105: Relocated Database

This result would probably not be what you expect – the result is a new database with a name that most likely does not match any reasonable naming scheme.

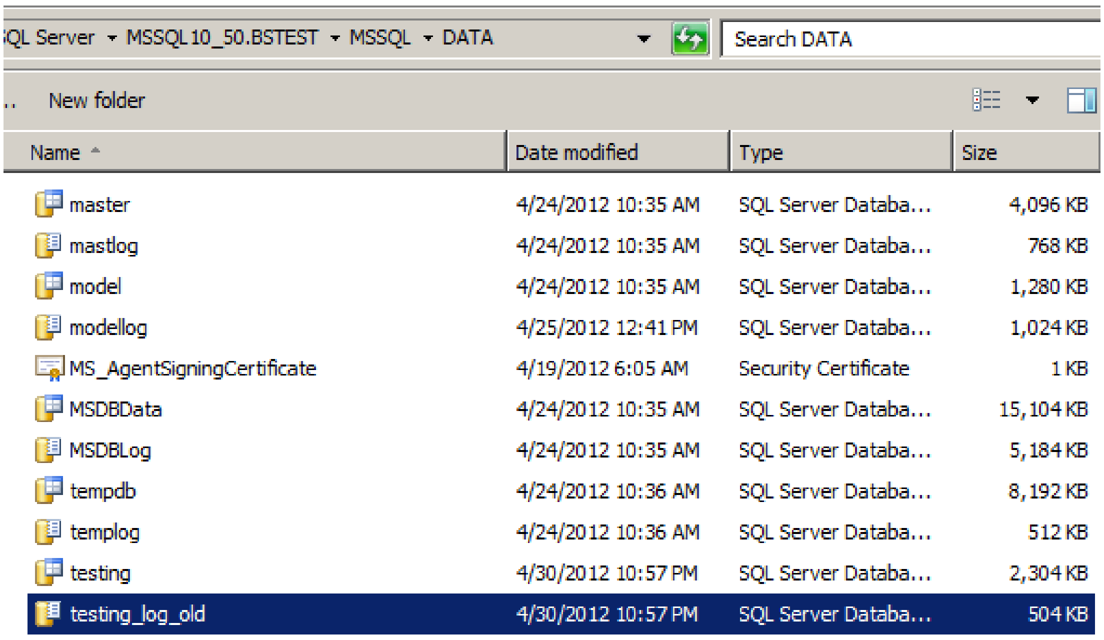
If you restore SQL Server data, it is important to understand that, during the restore process, **all** existing database log files in the database directory will be applied. For this reason, it can be important to remove old log files. This can only be done when the database is offline, so in this situation, the restore sequence may be like this:

[2] The MS SQL plugin is not intended to be used to migrate data. To do that, you should use specialized tools. In the simplest case, a complete SQl data dump may be created, moved to the new machine, and fed to the database locally.



```
/****** Script for SelectT
SELECT TOP 1000 [Id]
      ,[Text]
```

Fig. 106: Off-Line a Database in SQL Server Management Studio



Name	Date modified	Type	Size
master	4/24/2012 10:35 AM	SQL Server Databa...	4,096 KB
mastlog	4/24/2012 10:35 AM	SQL Server Databa...	768 KB
model	4/24/2012 10:35 AM	SQL Server Databa...	1,280 KB
modellog	4/25/2012 12:41 PM	SQL Server Databa...	1,024 KB
MS_AgentSigningCertificate	4/19/2012 6:05 AM	Security Certificate	1 KB
MSDBData	4/24/2012 10:35 AM	SQL Server Databa...	15,104 KB
MSDBLog	4/24/2012 10:35 AM	SQL Server Databa...	5,184 KB
tempdb	4/24/2012 10:36 AM	SQL Server Databa...	8,192 KB
templog	4/24/2012 10:36 AM	SQL Server Databa...	512 KB
testing	4/30/2012 10:57 PM	SQL Server Databa...	2,304 KB
testing_log_old	4/30/2012 10:57 PM	SQL Server Databa...	504 KB

Fig. 107: Renaming a Log File in Explorer

- Take database offline. This can be done through the `Microsoft SQL Server Management Studio` by right-clicking on the database in the Explorer pane, clicking “Tasks”, then “Take Offline”. This can be seen in figure *Off-Line a Database in SQL Server Management Studio*.
- Delete, move or rename the database log files. An example is shown in figure *Renaming a Log File in Explorer*.
- Run the actual restore.
- Bring the database online again, similar to the procedure in step 1.

Example

We assume that a correct backup of MSSQL data exists and you start the restore with option 5 in `bconsole`’s **restore** command, mark the complete tree of data backed up by the MSSQL component of the VSS plugin, then finally do `lsmark @MSSQL` to show all the files selected to be restored. Then the output you see should be similar to that presented in figure *Output for lsmark Command with SQL Server 2005 Data Marked*.

Note, the MSSQL files are generally under @MSSQL/MSDEWriter as in figure *Output for lsmark Command with SQL Server 2005 Data Marked* if you are running MSSQL 2005 on Windows Server 2003. If you are running MSSQL 2008 on Windows Server 2008, they will be under @MSSQL/SqlServerWriter.

Following that part of the data tree is the name of the MSSQL server host, in this case RUFUS-WIN2003, possibly the database server instance, and then the various databases.

In figure *Output for lsmark Command with SQL Server 2005 Data Marked* you can see that the name of the database master immediately follows RUFUS-WIN2003. The other databases, model, and msdb are at the same indentation level as master. So, for the restore to work, in the above example, you will need to unmark all the items that are associated with database master and below. If you have only selected database msdb for restoration, you would have output that looks like the one shown in figure *Marked Files to Restore Excluding master Database*.

```
$ mark @M*
14 files marked.
$ lsmark
*@MSSQL/
  *MSDEWriter/
    *RUFUS-WIN2003/
      *master/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *master.mdf
                    *mastlog.ldf
      *model/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *model.mdf
                    *modellog.ldf
      *msdb/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *msdbdata.mdf
                    *msdblog.ldf
```

Fig. 108: Output for lsmark Command with SQL Server 2005 Data Marked

```
$ lsmark
*@MSSQL/
  *MSDEWriter/
    *RUFUS-WIN2003/
      *msdb/
        *:component_info_4f771ba1
        *c:/
          *program files/
            *microsoft sql server/
              *mssql.1/
                *mssql/
                  *data/
                    *msdbdata.mdf
                    *msdblog.ldf
```

Fig. 109: Marked Files to Restore Excluding master Database

Restoring master Databases

To restore the SQL Server **master** database, the only reliable way is to turn off plugins, restore the database files to some location, and copy the restored files to their original location.

To turn off the VSS plugin, use `notepad` to edit the **file deamon** (fd)'s configuration file, `bacula-fd.conf`, put a hash sign "#" in front of the `plugin directory` line, and re-start the **file deamon** (fd). A typical command sequence (in a command window with elevated privileges!) would look something like this:

```
net stop bacula-fd
notepad "C:\Program Files\Bacula\bacula-fd.conf"
net start bacula-fd
```

```
$ ls
master.mdf
mastlog.ldf
$ pwd
cwd is: /@MSSQL/SqlServerWriter/WSB-SQL08/BSTEST/master/c:/program files/
↪microsoft
 sql server/mssql10_50.bstest/mssql/data/
$ mark *
2 files marked.
```

This example shows SQL Server 2010 paths.

When restoring, you should navigate directly to the directory containing the data files, which will pe represented in the virtual directory tree similarly to what we show in figure **fig:masterrestore**. In there, mark the data files (there should be two: `master.mdf` and `mastlog.ldf`), and continue the restore process. Using file relocation features to put the files into a new location is strongly recommended.

After the restore of the data files, you have to shut down the SQL server instance, move the restored data files to their correct location, and start the server instance again.

Afterwards, make sure to turn on plugins for the again.

Restoring the **master** database while MSSQL server is running is not possible.

Plugin Notes

Windows VSS Plugin Items to Note

- One file from each drive needed by the plugins must be explicitly listed in File Set used. This is to ensure that the main **Bacula** code does a snapshot of all the required drives. At a later time, we will find a way to accomplish this automatically.
- When doing a backup that is to be used for Bare Metal Recovery, do **not** use the VSS plugin. The reason is that during a Bare Metal Recovery, VSS is not available nor are the writers from the various components that are needed to do the restore. You might do a full backup to be used with a Bare Metal Recovery once a month or once a week, and all other days, do a backup using the VSS plugin, but under a different Job name. Then to restore your system, use the last Full non-VSS backup during the bare metal restoration of your system, and after rebooting do a restore with the VSS plugin to get everything fully up to date.

General Plugin Items to Note

- The ’estimate’ command does not handle plugins. When estimating a job that uses plugins, an error message regarding the plugin will be displayed. However, backup jobs will use the plugin.
- The File Set Include Option `CheckFileChanges = Yes` does not work with plugin-generated data. Thus, you must not use that Option in the Include section of the Fileset where you specify using the MSSQL plugin.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Problem Resolution

Most problems that can happen are reported in the job report **Bacula** sends. In the following table, we have collected information about common problems and the suggested resolution.

Job Report Message	Cause	Resolution
Unable to do a Differential backup of MSSQL master. Database excluded.		The master table is only backed up at full level. This is desing limitation of MS SQL Server.[3]
Warning: VSS Writer "SqlServer-Writer" has invalid sttate. ERR=The writer vetoed the shadow copy creation creation process during the backup preparation state.	Various	Check for other messages in the Job Report
VSS Writer (PrepareForBackup): "SqlServerWriter", State: 0x7 (VSS_WS_FAILED_AT_PREPARE_BACKUP)		
Warning: VSS Writer "SqlServer-Writer" has invalid sttate. ERR=The writer vetoed the shadow copy creation creation process during the backup preparation state.	Various	Check for other messages in the Job Report
SQL writer error: Backup type 2 not supported.	Incremental	Do not run incremental use Full and Differential levels only
Error: Unstable writer state=13: Restore skipped for file: /@MSSQL/SqlServerWriter/… /… /master/:component_info_… Writer="SqlServerWriter" … Sqllib error: OLEDB Error encountered calling ICommandText::Execute. hr = 0x80040e14.		
SQLSTATE: 42000, Native Error: 3013 Error state: 1, Severity: 16 … Error message: RESTORE master WITH SNAPSHOT is not supported. To restore master from a snapshot backup, stop the service and copy the data and log file.		The master table can not be restored with SQL Server running normally. Follow instructions given above
Error: Unstable writer state=0: Restore skipped for file: /@MSSQL/SqlServerWriter/… /… /master/:component_info_… Writer="SqlServerWriter" Sqllib error: OLEDB Error encountered calling IDBInitialize::Initialize. hr = 0x80004005 SQLSTATE: 42000, Native Error: 18461 Error state: 1, Severity: 14 Source: Microsoft SQL Server Native Client 10.0 Error message: Login failed for user 'NT AUTHORITYSYSTEM'. Reason: Server is in single user mode. Only one administrator can connect at this time.		
DBPROP_INIT_DATASOURCE: WSB-SQL08BSTEST DBPROP_INIT_CATALOG: master DBPROP_AUTH_INTEGRATED: SSPI		Follow instructions above to restore master table

[3] An explanation can be found at a Microsoft Web site

VSS MSSQL Plugin Deprecation - Migration Instructions to VDI MSSQL Plugin

The VSS MSSQL Plugin is being phased out due to its limitations therefore we recommend using the MSSQL VDI plugin instead. Unfortunately they are not compatible, therefore the two plugins must remain installed until the data in for the VSS MSQL plugin expires. Please find below some highlights on transtioning between the two plugins.

The mssqlvdi-plugin documentation offers to review all configuration options.

Requirements

- MSSQL Server user with administrative credentials
- MSSQL Server Instance name
- Recommended: A list of the database names to backup

Job Configuration

- Ideally create one job per database
- Unlike cinclude/cexclude there is no need to refer to the virtual VSS writer path. The include/exclude parameters refer to database names or patterns. If a single database is to be copied the parameter database=DatabaseName is preferred. Please find some equivalent examples below:
- Include just the database TestDB:

```
VSS MSSQL: Plugin = "vss:/MSSQL/ cinclude=*/TestDB cexclude=*"
VDI MSSQL: Plugin = "mssql: database=TestDB"
```

- Include just databases database1 database2 and database3

```
VSS MSSQL: Plugin = "vss:/MSSQL/ cinclude=*/database1 cinclude=*/database2 ␣
↪cinclude=*/database3 cexclude=*"
VDI MSSQL: Plugin = "mssql: include=database1 include=database2␣
↪include=database3 exclude=*"
```

- Just exclude database msdb from backup

```
VSS MSSQL: Plugin = "vss:/MSSQL/ cexclude=msdb"
VDI MSSQL: Plugin = "mssql: exclude=msdb"
```

- If Instance name is different than the default it is necessary to configure the plugin with at least these parameters:

```
Plugin = "mssql: instance=SQLExpress hostname=."
```

- Check if the recovery model of the databases as Simple recovery model only allows Full and Differential backups to set the default backup level accordingly, as Bacula's default is incremental.
- If the MSSQL server is an always-on cluster please refer to the section *Always On Availability Groups* in order to decide the backup strategy regarding primary/secondary replicas

Attention: The first job will be upgraded to a full backup The first MSSQL VDI backup will be a full regardless any pre-existing MSSQL VSS backup and there can be no further MSSQL VSS backup after the first MSSQL VDI backup.

Restore

Bacula is able to detect what plugin was used to perform the backup and act accordingly if just the fileset is changed using this procedure . The file tree is substantially different when using the new plugin. For the VSS MSQL plugin the tree looks like below:

```
/@MSSQL/SqlServerWriter/WSB-SQL08/BSYSTEST/database/c:/program files/
↪microsoft sql server/mssql10_50.bstest/mssql/data/database
```

While the file tree for the MSSQL VDI plugin for the same database for the BSYSTEST instance looks like below:

```
/@mssql/BSYSTEST/database
```

5.2 DB2 Plugin

Executive summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership.

This document is intended to provide insight into the considerations and processes required to successfully implement DB2 backup technique.

DB2

Architecture

The Bacula Enterprise DB2 Plugin relies on the extensive DB2 backup and restore API.

Features

The Bacula Enterprise DB2 Plugin can :

- use a named pipe to transfer data
- list and backup all databases from an instance
- select the databases to backup
- handle ONLINE/OFFLINE backups automatically
- detect if a database can support incremental and differential backup level
- stop/start automatically a database to do an OFFLINE backup
- backup a database executed via Docker
- automatically backup the database schema
- restore a database backup into a new database automatically
- ease the incremental restore procedure
- keep track of the backup timestamp
- control all steps and report errors to the job log
- restore a database dump file into a local directory for manual restore

Installation of the Plugin

On the Bacula File Daemon that you want to use, extend the repository file for your package manager to contain a section for the plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseDB2Plugin]
name=Bacula Enterprise DB2 Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/db2/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

or in Debian Stretch, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪jessie-64/ stretch main
deb https://www.baculasystems.com/dl/@customer-string@/debs/db2/@version@/
↪jessie-64/ stretch db2
```

Then perform a `yum update` or `apt-get update`, and after that the package *bacula-enterprise-db2-plugin* can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

Configuration

Plugin Command Line Parameters

The following options can be used in the Fileset Plugin definition:

- **can_stop** If the database configuration doesn't permit online backups, the plugin can stop the target database for the backup. If this parameter is not set, the database will not be backed up and an error will be raised in the Job.
- **use_sudo** Use the sudo command to prefix all `db2` commands. Used in conjunction with `unix_user`.
- **unix_user=<str>** Unix user to use to run `db2` commands. By default commands are executed under the Bacula FileDaemon service account (root) or set to `instance` parameter if used. Ex: `unix_user=db2inst1`
- **abort_on_error** Abort immediately the job if a serious error is found. By default, a Job error is raised, but other files and databases are backed up.
- **timeout=<int>** Number of seconds used to run various commands or wait to connect `DB2` during a backup. The default value is 120 seconds. Ex: `timeout=5mins`
- **ctl_dir=<path>** Directory accessible to both the DB2 service and the Bacula File Daemon service. The Bacula DB2 Plugin will create command files inside this directory, and the DB2 instance user will execute them. The DB2 instance user must be able to read and create files inside this directory. The default value is `/tmp` and must be adapted to your configuration for security reasons. Ex: `ctl_dir=/data/backup`

  ```
  # mkdir /data/backup
  # chmod 700 /data/backup
  # chown db2inst1 /data/backup
  ```

- **user_config_dir=<path>** Directory used to store local customization files for backup and restore procedures. (See ref:*custom-script* for more information)
- **bin_dir=<str>** String that is added to all `DB2` commands. It can be a directory outside the PATH, or some specific options such as a docker command. Ex: `bin_dir=/db2/config`
- **user=<str>** DB2 user account used in db2 commands.
- **password=<str>** DB2 user password used in db2 commands.

- **instance=<str>** DB2 instance name. This string is compared to the DB2 instance that will be used in the backup/restore procedure. If the instance doesn't match the instance found on the system, the plugin command will be skipped or the job will be aborted. Ex: `instance=db2inst1`
- **database=<glob>** DB2 database glob pattern to include in the backup. This glob expression is used to filter the database list found on the DB2 server. If the parameter is set and the selected database list is empty, an error will be raised. Ex: `database=prod*`

DB2 Docker Configuration

When using Docker DB2 package, the database directory must be shared with the main system, and Bacula can use this directory to communicate with DB2.

For example, if your Docker image was started with the following command:

```
mkdir -p /data/ctl
docker run -itd --name mydb2 -e DBNAME=testdb -v /data:/data \
  -e DB2INST1_PASSWORD=XXX -e LICENSE=accept -p 50000:50000  \
  --privileged=true ibmcom/db2
```

The Fileset will look like:

```
Plugin = "db2: bin_dir=\\"docker exec mydb2 su - db2inst1 -c \\" ctl_dir=/
↪data/ctl"
```

Where `mydb2` is the DB2 docker container name, `db2inst1` is the DB2 unix user inside the Docker container and `/data` is the volume shared between the host system and the docker container.

Plugin Restore Options

- **database** New database name. Take precedence over where parameter
- **user** Username
- **password** Password
- **unix_user** Unix user to use
- **instance** Instance where to restore
- **restore_options** Restore options passed to db2 restore command
- **replace_existing** Add the `REPLACE EXISTING` parameter to the restore command
- **rollforward_options** Rollforward options passed to db2 rollforward command
- **logtarget** `LOGTARGET` option to use when restoring the final image. Ex: `'/tmp/'`

Plugin Configuration File

During the Backup and the Restore operations, the Bacula FileDaemon DB2 Plugin will generate a set of dynamic scripts on disk to call the DB2 backup API.

example:

```
#!/bin/sh
export DB2INSTANCE=db2inst1

BDB2_BIN_DIR=""
BDB2_DATABASE="DB1"
BDB2_JOB_LEVEL=F
BDB2_FIFO="/data/tmp/fifo.5"
BDB2_INCLUDE_LOGS=""
BDB2_BACKUP_LEVEL=""
BDB2_STOP_DB=YES
BDB2_EXTRA_ARGS="WITHOUT PROMPTING"
BDB2_CMD='${BDB2_BIN_DIR}db2 -vst BACKUP DB "${BDB2_DATABASE}" ${BDB2_BACKUP_
↪LEVEL}
            to "${BDB2_FIFO}" ${BDB2_INCLUDE_LOGS} ${BDB2_EXTRA_ARGS}'
BDB2_VERSION=1

################################################################
if [ -f "/data/custom/db2_backup" ]
 then
    . "/data/custom/db2_backup"
 fi

if [ $BDB2_STOP_DB = YES ]; then
  echo Stopping the database...
  ${BDB2_BIN_DIR}db2 -v CONNECT TO "${BDB2_DATABASE}"
  ${BDB2_BIN_DIR}db2 -v LIST APPLICATION FOR DATABASE "${BDB2_DATABASE}"
  ${BDB2_BIN_DIR}db2 -v QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
  ${BDB2_BIN_DIR}db2 TERMINATE
  ${BDB2_BIN_DIR}db2 -v DEACTIVATE DATABASE "${BDB2_DATABASE}"
fi
mkfifo "${BDB2_FIFO}"
eval ${BDB2_CMD}
ret=$?
rm -f "${BDB2_FIFO}"
if [ $BDB2_STOP_DB = YES ]; then
  echo Starting the database...
  ${BDB2_BIN_DIR}db2 -v ACTIVATE DATABASE "${BDB2_DATABASE}"
  ${BDB2_BIN_DIR}db2 -v CONNECT TO "${BDB2_DATABASE}"
  ${BDB2_BIN_DIR}db2 -v UNQUIESCE DATABASE
  ${BDB2_BIN_DIR}db2 TERMINATE
fi
exit $ret
```

These scripts are defining a set of variables to complete the operation. It is possible to overwrite the variables to customize the process execution. The `user_config_dir` can be used to configure where the custom configuration file will be stored.

In the following example, to use the DB2 COMPRESS option, it is possible to edit the file /data/custom/db2_backup file and overwrite the BDB2_EXTRA_ARGS variable to use the custom COMPRESS option. Note, that in this example, it is possible to use Bacula builtin compression instead.

```
db2inst1:~# grep db2 /opt/bacula/etc/bacula-dir.conf
Plugin = "db2: user_config_dir=/data/custom ctl_dir=/data/tmp"

db2inst1:~# cat /data/custom/db2_backup
BDB2_EXTRA_ARGS="COMPRESS WITHOUT PROMPTING"
```

If the modification is more advanced, it is possible to control the Bacula FileDaemon DB2 shell protocol version.

```
db2inst1:~# cat /data/custom/db2_backup
if [ $BDB2_SHELL_VERSION = 1 ]; then
   BDB2_EXTRA_ARGS="COMPRESS WITHOUT PROMPTING"
fi
```

To optimize the format of the backup images for Aligned device or Global Endpoint Deduplication storage, it is possible to use:

```
db2inst1:~# cat /data/custom/db2_backup
if [ $BDB2_SHELL_VERSION = 1 ]; then
   BDB2_EXTRA_ARGS="DEDUP_DEVICE WITHOUT PROMPTING"
fi
```

DB2 Plugin Customization Interface Version 1

- db2_backup, db2_size, db2_dump_schema, db2_instance, db2_config, db2_list user files
  - BDB2_BIN_DIR=<string> Directory or string to prepend to all db2 commands.
  - BDB2_DATABASE=<string> Current database name.
  - BDB2_SHELL_VERSION=<int> Current Bacula Enteprise DB2 shell version.
- db2_backup user file
  - BDB2_BACKUP_LEVEL=<string> DB2 option to specify the backup level. Ex: BDB2_BACKUP_LEVEL=INCREMENTAL DELTA
  - BDB2_CMD=<string> Command to be executed via eval.
  - BDB2_EXTRA_ARGS=<string> DB2 database backup command arguments. Ex: BDB2_EXTRA_ARGS=WITHOUT PROMPTING
  - BDB2_FIFO=<path> Path to the communication Unix named pipe used by the script.
  - BDB2_INCLUDE_LOGS=<string> DB2 option to include LOGS in the backup if the database supports it. Ex: BDB2_INCLUDE_LOGS=INCLUDE LOGS
  - BDB2_JOB_LEVEL=<char> Current Bacula backup job level. Can be "F" for Full, "I" for Incremental and "D" for Differential.
  - BDB2_ONLINE=<string> DB2 ONLINE option if the database supports it. Ex BDB2_ONLINE=ONLINE

  - BDB2_STOP_DB=<bool> If the DB2 database must be stopped during the backup. Ex: BDB2_STOP_DB=YES
  - BDB2_USER_STRING=<string> User and password configuration string passed to dblook schema analyzer.

- db2_restore
  - BDB2_NEWDATABASE The target database name
  - BDB2_REPLACE If the restore must replace a database. Ex: BDB2_REPLACE=REPLACE EXISTING
  - BDB2_LOGTARGET is used when restoring the final image. (It is also possible to modify the option via bacula restore menu). Ex: BDB2_LOGTARGET=LOGTARGET '/tmp/'
  - BDB2_TAKEN Ex. BDB2_TAKEN=TAKEN AT 20190815120000

These variables can be configured in the user_config_dir.

Examples

```
Fileset {
 Name = DB2_ALL_DB
 Description = "Backup all DB2 databases"
 Include {
   Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1"
 }
}

Fileset {
 Name = DB2_OFFLINE
 Description = "Backup all DB2 databases OFFLINE"
 Include {
   Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 can_stop database=mydb"
 }
}
```

Backup

During a backup Job, the Bacula DB2 Plugin will analyze the configuration and adjust the parameters automatically.

For example, in the following example, Bacula will handle a database named DB2, as the database is configured with the TRACKMOD and the LOGARCHMETH1 options, Bacula will be able to run incremental and differential job level and backup database logs.

```
Fileset {
 Name = FS_DB2_DB
 Description = "Backup DB2 database"
 Include {
   Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 database=DB2"
 }
}
```

(continues on next page)

(continued from previous page)

```
Job {
 Name = J_DB2_DB
 Fileset = FS_DB2_DB
 Client = db2-fd
 JobDefs = Defaults
}
```

The Full job output will look like the following:

```
dir JobId 1: No prior or suitable Full backup found in catalog. Doing FULL␣
↪backup.
dir JobId 1: Start Backup JobId 1, Job=J_DB2_DB.2019-08-13_17.07.03_04
dir JobId 1: Using Device "FileStorage" to write.
sd JobId 1: Wrote label to prelabeled Volume "Vol001" on File device
↪"FileStorage" (/storage)
fd JobId 1: DB2: -- No userid was specified, db2look tries to use Environment␣
↪variable USER
fd JobId 1: DB2: -- USER is: DB2INST1
fd JobId 1: DB2: -- Creating DDL for table(s)
fd JobId 1: DB2: -- Binding package automatically ...
fd JobId 1: DB2: -- Bind is successful
fd JobId 1: DB2: -- Binding package automatically ...
fd JobId 1: DB2: -- Bind is successful
fd JobId 1: DB2: Doing backup of database "DB2"
fd JobId 1: DB2: BACKUP DATABASE DB2 ONLINE to /data/tmp/fifo.1 INCLUDE LOGS␣
↪WITHOUT PROMPTING
fd JobId 1: DB2: Backup successful. The timestamp for this backup image is :␣
↪20190813150713
sd JobId 1: Elapsed time=00:00:09, Transfer rate=1.586 M Bytes/second
sd JobId 1: Sending spooled attrs to the Director. Despooling 1,753 bytes ...
dir JobId 1: Bacula db2-dir 12.0.1 (05Aug19):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  1
  Job:                    J_DB2_DB.2019-08-13_17.07.03_04
  Backup Level:           Full (upgraded from Incremental)
  Client:                 "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
↪archlinux,
  Fileset:                "FS_DB2_DB" 2019-08-13 17:06:58
  Pool:                   "Default" (From Job resource)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         13-Aug-2019 17:07:03
  Start time:             13-Aug-2019 17:07:05
  End time:               13-Aug-2019 17:07:14
  Elapsed time:           9 secs
  Priority:               10
  FD Files Written:       7
  SD Files Written:       7
  FD Bytes Written:       14,280,473 (14.28 MB)
  SD Bytes Written:       14,282,545 (14.28 MB)
  Rate:                   1586.7 KB/s
```

(continues on next page)

(continued from previous page)

```
  Software Compression:   92.9% 14.1:1
  Comm Line Compression:  4.3% 1.0:1
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               no
  Volume name(s):         Vol001
  Volume Session Id:      1
  Volume Session Time:    1565708807
  Last Volume Bytes:      14,328,370 (14.32 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK

*list files jobid=1
Using Catalog "MyCatalog"
+-------------------------------+
| filename                      |
+-------------------------------+
| /@DB2/db2inst1/DB2/schema.sql |
| /@DB2/db2inst1/DB2/data.sql.F |
| /@DB2/db2inst1/DB2/           |
| /@DB2/                        |
| /@DB2/db2inst1/               |
+-------------------------------+
```

The Incremental job output will look like the following:

```
dir JobId 2: Start Backup JobId 2, Job=J_DB2_DB.2019-08-13_17.07.15_05
dir JobId 2: Using Device "FileStorage" to write.
sd JobId 2: Volume "Vol001" previously written, moving to end of data.
fd JobId 2: DB2: -- No userid was specified, db2look tries to use Environment␣
↪variable USER
fd JobId 2: DB2: -- USER is: DB2INST1
fd JobId 2: DB2: -- Creating DDL for table(s)
fd JobId 2: DB2: Doing backup of database "DB2"
fd JobId 2: DB2: BACKUP DATABASE DB2 ONLINE INCREMENTAL DELTA to /data/tmp/
↪fifo.2 INCLUDE LOGS
                WITHOUT PROMPTING
fd JobId 2: DB2: Backup successful. The timestamp for this backup image is :␣
↪20190813150723
sd JobId 2: Elapsed time=00:00:07, Transfer rate=36.31 K Bytes/second
sd JobId 2: Sending spooled attrs to the Director. Despooling 1,131 bytes ...
dir JobId 2: Bacula db2-dir 12.0.1 (05Aug19):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  2
  Job:                    J_DB2_DB.2019-08-13_17.07.15_05
  Backup Level:           Incremental, since=2019-08-13 17:07:05
  Client:                 "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
↪archlinux,
  Fileset:                "FS_DB2_DB" 2019-08-13 17:06:58
```

(continues on next page)

(continued from previous page)

```
  Pool:                    "Default" (From Job resource)
  Catalog:                 "MyCatalog" (From Client resource)
  Storage:                 "File" (From Job resource)
  Scheduled time:          13-Aug-2019 17:07:15
  Start time:              13-Aug-2019 17:07:17
  End time:                13-Aug-2019 17:07:24
  Elapsed time:            7 secs
  Priority:                10
  FD Files Written:        6
  SD Files Written:        6
  FD Bytes Written:        252,832 (252.8 KB)
  SD Bytes Written:        254,221 (254.2 KB)
  Rate:                    36.1 KB/s
  Software Compression:    99.5% 198.5:1
  Comm Line Compression:   80.7% 5.2:1
  Snapshot/VSS:            no
  Encryption:              no
  Accurate:                no
  Volume name(s):          Vol001
  Volume Session Id:       2
  Volume Session Time:     1565708807
  Last Volume Bytes:       14,592,649 (14.59 MB)
  Non-fatal FD errors:     0
  SD Errors:               0
  FD termination status:   OK
  SD termination status:   OK
  Termination:             Backup OK

*list files jobid=2
Using Catalog "MyCatalog"
+------------------------------------------------+
| filename                                       |
+------------------------------------------------+
| /@DB2/db2inst1/DB2/data.sql.I.20190813170822 |
| /@DB2/db2inst1/DB2/schema.sql                  |
| /@DB2/db2inst1/DB2/                            |
| /@DB2/db2inst1/                                |
| /@DB2/                                         |
+------------------------------------------------+
```

In the following example, the DB3 and DB1 databases are not configured to allow ONLINE backups. The plugin will stop the databases automatically during the backup. If a user or an application is connected to the database, the plugin will terminate all connections automatically.

```
Fileset {
 Name = FS_DB2_DB3
 Description = "Backup DB1/DB3 database"
 Include {
   Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 database=DB[13] can_stop
↪"
 }
}
```

(continues on next page)

(continued from previous page)

```
Job {
 Name = J_DB3_DB
 Fileset = FS_DB3_DB
 Client = db2-fd
 JobDefs = Defaults
}
```

```
dir JobId 5: Start Backup JobId 5, Job=J_DB3_DB.2019-08-13_17.07.43_08
dir JobId 5: Using Device "FileStorage" to write.
sd JobId 5: Volume "Vol001" previously written, moving to end of data.
fd JobId 5: DB2: -- No userid was specified, db2look tries to use Environment␣
↪variable USER
fd JobId 5: DB2: -- USER is: DB2INST1
fd JobId 5: DB2: -- Creating DDL for table(s)
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: Doing backup of database "DB3"
fd JobId 5: DB2: Stopping the database...
fd JobId 5: DB2: CONNECT TO DB3
fd JobId 5: DB2:    Database Connection Information
fd JobId 5: DB2:  Database server        = DB2/LINUXX8664 11.5.0.0
fd JobId 5: DB2:  SQL authorization ID   = DB2INST1
fd JobId 5: DB2:  Local database alias   = DB3
fd JobId 5: DB2: LIST APPLICATION FOR DATABASE DB3
fd JobId 5: DB2: Auth Id  Application    Appl.      Application Id             ␣
↪    DB        # of
fd JobId 5: DB2:          Name           Handle                                ␣
↪     Name    Agents
fd JobId 5: DB2: -------- -------------- ---------- --------------------------
↪--- -------- -----
fd JobId 5: DB2: DB2INST1 db2bp          996        *LOCAL.db2inst1.
↪190813151420  DB3      1
fd JobId 5: DB2: QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
fd JobId 5: DB2: DB20000I  The QUIESCE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: DB20000I  The TERMINATE command completed successfully.
fd JobId 5: DB2: DEACTIVATE DATABASE DB3
fd JobId 5: DB2: DB20000I  The DEACTIVATE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: BACKUP DATABASE DB3 to /database//tmp/fifo.5 WITHOUT␣
↪PROMPTING
fd JobId 5: DB2: Backup successful. The timestamp for this backup image is :␣
↪20190813150759
fd JobId 5: DB2: Starting the database...
fd JobId 5: DB2: ACTIVATE DATABASE DB3
fd JobId 5: DB2: DB20000I  The ACTIVATE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: CONNECT TO DB3
```

(continues on next page)

(continued from previous page)

```
fd JobId 5: DB2:    Database Connection Information
fd JobId 5: DB2:  Database server        = DB2/LINUXX8664 11.5.0.0
fd JobId 5: DB2:  SQL authorization ID   = DB2INST1
fd JobId 5: DB2:  Local database alias   = DB3
fd JobId 5: DB2: UNQUIESCE DATABASE
fd JobId 5: DB2: DB20000I  The UNQUIESCE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: DB20000I  The TERMINATE command completed successfully.
fd JobId 5: DB2: -- No userid was specified, db2look tries to use Environment␣
↪variable USER
fd JobId 5: DB2: -- USER is: DB2INST1
fd JobId 5: DB2: -- Creating DDL for table(s)
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: Doing backup of database "DB1"
fd JobId 5: DB2: Stopping the database...
fd JobId 5: DB2: CONNECT TO DB1
fd JobId 5: DB2:    Database Connection Information
fd JobId 5: DB2:  Database server        = DB2/LINUXX8664 11.5.0.0
fd JobId 5: DB2:  SQL authorization ID   = DB2INST1
fd JobId 5: DB2:  Local database alias   = DB1
fd JobId 5: DB2: LIST APPLICATION FOR DATABASE DB1
fd JobId 5: DB2: Auth Id  Application    Appl.      Application Id              ␣
↪    DB        # of
fd JobId 5: DB2:          Name           Handle                                 ␣
↪    Name    Agents
fd JobId 5: DB2: -------- -------------- ---------- --------------------------
↪--- -------- -----
fd JobId 5: DB2: DB2INST1 db2bp          1052       *LOCAL.db2inst1.
↪190813151516  DB1      1
fd JobId 5: DB2: QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
fd JobId 5: DB2: DB20000I  The QUIESCE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: DB20000I  The TERMINATE command completed successfully.
fd JobId 5: DB2: DEACTIVATE DATABASE DB1
fd JobId 5: DB2: DB20000I  The DEACTIVATE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: BACKUP DATABASE DB1 to /database//tmp/fifo.5 WITHOUT␣
↪PROMPTING
fd JobId 5: DB2: Backup successful. The timestamp for this backup image is :␣
↪20190813150813
fd JobId 5: DB2: Starting the database...
fd JobId 5: DB2: ACTIVATE DATABASE DB1
fd JobId 5: DB2: DB20000I  The ACTIVATE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: CONNECT TO DB1
fd JobId 5: DB2:    Database Connection Information
fd JobId 5: DB2:  Database server        = DB2/LINUXX8664 11.5.0.0
fd JobId 5: DB2:  SQL authorization ID   = DB2INST1
fd JobId 5: DB2:  Local database alias   = DB1
```

(continues on next page)

(continued from previous page)

```
fd JobId 5: DB2: UNQUIESCE DATABASE
fd JobId 5: DB2: DB20000I  The UNQUIESCE DATABASE command completed␣
↪successfully.
fd JobId 5: DB2: DB20000I  The TERMINATE command completed successfully.
sd JobId 5: Elapsed time=00:00:31, Transfer rate=6.386 M Bytes/second
sd JobId 5: Sending spooled attrs to the Director. Despooling 2,494 bytes ...
dir JobId 5: Bacula db2-dir 12.0.1 (05Aug19):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  5
  Job:                    J_DB3_DB.2019-08-13_17.07.43_08
  Backup Level:           Full (upgraded from Incremental)
  Client:                 "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
↪archlinux,
  Fileset:                "FS_DB3_DB" 2019-08-13 17:07:43
  Pool:                   "Default" (From Job resource)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         13-Aug-2019 17:07:43
  Start time:             13-Aug-2019 17:07:45
  End time:               13-Aug-2019 17:08:16
  Elapsed time:           31 secs
  Priority:               10
  FD Files Written:       11
  SD Files Written:       11
  FD Bytes Written:       197,964,193 (197.9 MB)
  SD Bytes Written:       197,967,386 (197.9 MB)
  Rate:                   6385.9 KB/s
  Software Compression:   43.9% 1.8:1
  Comm Line Compression:  84.2% 6.3:1
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               no
  Volume name(s):         Vol001
  Volume Session Id:      5
  Volume Session Time:    1565708807
  Last Volume Bytes:      213,265,390 (213.2 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK

*list files jobid=5
+-------------------------------+
| filename                      |
+-------------------------------+
| /@DB2/                        |
| /@DB2/db2inst1/               |
| /@DB2/db2inst1/DB3/schema.sql |
| /@DB2/db2inst1/DB3/data.sql.F |
| /@DB2/db2inst1/DB3/           |
| /@DB2/db2inst1/DB1/schema.sql |
```

(continues on next page)

(continued from previous page)

```
| /@DB2/db2inst1/DB1/data.sql.F |
| /@DB2/db2inst1/DB1/           |
+-------------------------------+
```

Restore

The Bacula Enterprise DB2 plugin is designed to help DB2 administrator manage the restore procedure.

Restoring a Single Full Database

To restore a single database with the Bacula Enterprise DB2 plugin, the appropriate files from the database directory are selected during the restore process.

To restore the database with its original name, the selection should only contain the data file (`data.sql.*`).

To restore a single database to a new name, the file `data.sql.*` must be selected. The **where** parameter is used to specify the new database name. If **where** is set to a single word consisting of only `a..z`, `0-9` and `_`, Bacula will create the specified database and restore the data into it. On DB2, the database name has some restrictions, please see DB2 documentation on that subject.

```
* restore where=bacula2
...
cwd is: /
$ cd /@DB2/db2inst1/BACULA
cwd is: /@DB2/db2inst1/BACULA
$ m data.sql.F
$ lsmark
*data.sql.F
```

If the `replace` parameter is set to , Bacula will check the database list, and will abort the Job if the database being restored already exists.

Restoring an Incremental/Differential Backup

The following procedure describes how to restore a database to a point in time.

1. Select the final image to be restored. This is the usually just the most recent data.sql.* file from your Bacula backups of the database. Access this using the normal Bacula restore procedure, selecting the jobs for the client up to the current date, and then mark and restore just the most recent backup. Please see the following code snippets for an example. This will restore the database configurations and necessary initial information.
2. Next, the restore command is run again, and all the database backups back to the most recent full are selected. Make sure that the job selected in step 1 is included. The 'mark *' command should select the correct data as long as the job listing is the same as in step 1. Bacula will restore each of the required full, differential or incremental backupimages, in the order in which they were produced, on top of the baseline image restored in Step 1. The last image restored will extract database logs automatically.
3. Once all the data is restored, the roolforward command must be executed to complete the backup and replay all necessary log files.

During a Bacula incremental restore operation, only the logs included in the target image of the restore operation will be retrieved from the backup image. Any logs that are included in intermediate images referenced during the incremental restore process will not be extracted from those backup images.

Once the database is restored, the ROOLFORWARD command is needed to finish the restore.

```
*restore client=sap-fd where=test1 select
+-------+-------+----------+------------+---------------------+---------------
↪+
| jobid | level | jobfiles | jobbytes   | starttime           | volumename    ␣
↪|
+-------+-------+----------+------------+---------------------+---------------
↪+
|     1 | F     |        7 | 14,296,244 | 2019-08-15 11:13:43 | TestVolume001␣
↪|
|     4 | D     |        6 |    253,359 | 2019-08-15 11:14:14 | TestVolume001␣
↪|
|     6 | I     |        6 |    253,715 | 2019-08-15 11:15:05 | TestVolume001␣
↪|
|     7 | I     |        6 |    253,770 | 2019-08-15 11:15:16 | TestVolume001␣
↪|
+-------+-------+----------+------------+---------------------+---------------
↪+
You have selected the following JobIds: 1,4,6,7

Building directory tree for JobId(s) 1,4,6,7 ...
5 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @DB2
cwd is: /@DB2/
$ cd db2inst1/
cwd is: /@DB2/db2inst1/
$ cd DB2
cwd is: /@DB2/db2inst1/DB2/
$ ls
data.sql.D.20190815110819
data.sql.F
data.sql.I.20190815110810
data.sql.I.20190815110820
schema.sql
$ m data.sql.I.20190815110820
1 file marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
```

(continues on next page)

(continued from previous page)

```
===========================================================================

    TestVolume001             File                        FileStorage

Volumes marked with "*" are in the Autochanger.


1 file selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/working/127.0.0.1-dir.restore.2.bsr
Where:           test1
Replace:         Always
Fileset:         Full Set
Backup Client:   db2-fd
Restore Client:  db2-fd
Storage:         File
When:            2019-08-15 11:25:10
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=18
*messages
dir JobId 18: Start Restore Job RestoreFiles.2019-08-15_11.25.22_03
dir JobId 18: Restoring files from JobId(s) 1,4,6,7
dir JobId 18: Using Device "FileStorage" to read.
sd JobId 18: Ready to read from volume "TestVolume001" on File device
↪"FileStorage" (/tmp).
sd JobId 18: Forward spacing Volume "TestVolume001" to addr=196768511
sd JobId 18: Elapsed time=00:00:01, Transfer rate=243.8 K Bytes/second
fd JobId 18: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.18
               TAKEN AT 20190815091522 INTO test1 LOGTARGET DEFAULT WITHOUT␣
↪PROMPTING
fd JobId 18: DB2: DB20000I  The RESTORE DATABASE command completed␣
↪successfully.
dir JobId 18: Bacula 127.0.0.1-dir 12.0.1 (05Aug19):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  18
  Job:                    RestoreFiles.2019-08-15_11.25.22_03
  Restore Client:         127.0.0.1-fd
  Where:                  test1
  Replace:                Always
  Start time:             15-Aug-2019 11:25:24
  End time:               15-Aug-2019 11:25:28
  Elapsed time:           4 secs
  Files Expected:         1
  Files Restored:         1
  Bytes Restored:         50,376,704 (50.37 MB)
  Rate:                   12594.2 KB/s
  FD Errors:              0
```

(continues on next page)

(continued from previous page)

```
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

At this step, the DB2 database was created on the server, and you need now to select all files to perform the restore.

```
*restore client=sap-fd where=test1 select
+-------+-------+----------+------------+---------------------+---------------
↪+
| jobid | level | jobfiles | jobbytes   | starttime           | volumename    ␣
↪|
+-------+-------+----------+------------+---------------------+---------------
↪+
|     1 | F     |        7 | 14,296,244 | 2019-08-15 11:13:43 | TestVolume001␣
↪|
|     4 | D     |        6 |    253,359 | 2019-08-15 11:14:14 | TestVolume001␣
↪|
|     6 | I     |        6 |    253,715 | 2019-08-15 11:15:05 | TestVolume001␣
↪|
|     7 | I     |        6 |    253,770 | 2019-08-15 11:15:16 | TestVolume001␣
↪|
+-------+-------+----------+------------+---------------------+---------------
↪+
You have selected the following JobIds: 1,4,6,7

Building directory tree for JobId(s) 1,4,6,7 ...
5 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @DB2
cwd is: /@DB2/
$ cd db2inst1/
cwd is: /@DB2/db2inst1/
$ cd DB2
cwd is: /@DB2/db2inst1/DB2/
$ m *
5 files marked.
$ lsmark
*data.sql.D.20190815110819
*data.sql.F
*data.sql.I.20190815110810
*data.sql.I.20190815110820
*schema.sql
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.1.bsr
```

(continues on next page)

(continued from previous page)

```
The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    TestVolume001             File                      FileStorage

Volumes marked with "*" are in the Autochanger.


5 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.1.bsr
Where:           test1
Replace:         Always
Fileset:         Full Set
Backup Client:   db2-fd
Restore Client:  db2-fd
Storage:         File
When:            2019-08-15 11:40:54
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): Job queued. JobId=19
*messages
dir JobId 19: Start Restore Job RestoreFiles.2019-08-15_11.40.56_03
dir JobId 19: Restoring files from JobId(s) 1,4,6,7
dir JobId 19: Using Device "FileStorage" to read.
sd JobId 19: Ready to read from volume "TestVolume001" on File device
↪"FileStorage" (/tmp).
sd JobId 19: Forward spacing Volume "TestVolume001" to addr=223
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
               TAKEN AT 20190815091351 INTO test1 WITHOUT PROMPTING
sd JobId 19: Elapsed time=00:00:03, Transfer rate=5.009 M Bytes/second
fd JobId 19: DB2: DB20000I  The RESTORE DATABASE command completed␣
↪successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
               TAKEN AT 20190815091420 INTO test1 WITHOUT PROMPTING
fd JobId 19: DB2: DB20000I  The RESTORE DATABASE command completed␣
↪successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
               TAKEN AT 20190815091511 INTO test1 WITHOUT PROMPTING
fd JobId 19: DB2: DB20000I  The RESTORE DATABASE command completed␣
↪successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
               TAKEN AT 20190815091522 INTO test1 LOGTARGET DEFAULT WITHOUT␣
↪PROMPTING
fd JobId 19: DB2: DB20000I  The RESTORE DATABASE command completed␣
↪successfully.
fd JobId 19: DB2: No ROLLFORWARD option speficied. Database is in pending mode
```

(continues on next page)

(continued from previous page)

```
db2 ROLLFORWARD DATABASE test1 TO END OF LOGS
db2 ROLLFORWARD DATABASE test1 COMPLETE
15-Aug 11:41 127.0.0.1-dir JobId 19: Bacula 127.0.0.1-dir 12.0.1 (05Aug19):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  19
  Job:                    RestoreFiles.2019-08-15_11.40.56_03
  Restore Client:         127.0.0.1-fd
  Where:                  test1
  Replace:                Always
  Start time:             15-Aug-2019 11:40:58
  End time:               15-Aug-2019 11:41:26
  Elapsed time:           28 secs
  Files Expected:         5
  Files Restored:         5
  Bytes Restored:         352,538,624 (352.5 MB)
  Rate:                   12590.7 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

To access the database, you must now execute the ROLLFORWARD function up to the point in time that is needed.

```
[db2inst1@db2 tmp]$ db2 ROLLFORWARD DATABASE test1 COMPLETE

                                 Rollforward Status

 Input database alias                   = test1
 Number of members have returned status = 1

 Member ID                              = 0
 Rollforward status                     = not pending
 Next log file to be read               =
 Log files processed                    =  -
 Last committed transaction             = 2019-08-15-09.15.22.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.

[db2inst1@db2 tmp]$ db2 connect to test1

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.5.0.0
 SQL authorization ID   = DB2INST1
 Local database alias   = TEST1

[db2inst1@db2 tmp]$ db2 'select * from a order by when desc limit 2'

VAL         WHEN
----------- --------------------------
          1 2019-08-15-09.15.17.792096
```

(continues on next page)

(continued from previous page)

```
          1 2019-08-15-09.15.13.121704

  2 record(s) selected.
```

Restoring Dump Files to a Directory

To restore DB2 images to a directory, the **where** parameter needs to be set to indicate an existing directory.

```
* restore where=/tmp
```

If the **where** parameter is a directory (containing /), Bacula will restore all files into this directory. Doing so, it is possible to use `db2 restore` directly and restore only particular contents.

```
[db2inst1@db2 tmp]$ cd /tmp/@DB2/db2inst1/DB2
[db2inst1@db2 DB2]$ mkfifo myfifo
[db2inst1@db2 DB2]$ cat data.sql.F > myfifo &
[db2inst1@db2 DB2]$ db2 restore dbtest from myfifo
```

Limitations

The Bacula Enterprise DB2 plugin has been written to help backup administrator manage and recover DB2 databases. The Plugin doesn't implement all DB2 backup features. For example, it is not possible to backup a particular tablespace with the plugin natively. For very advanced and granular use, it is possible to use the bpipe plugin. Please contact Bacula Systems for assistance with the bpipe plugin.

The XBA API is not part of the Bacula Enterprise DB2 Plugin.

The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5.3 MySQL Plugin

This chapter aims at presenting the reader with information about the Bacula Enterprise MySQL Plugin. The document briefly defines the scope of its operations, describes the target technology of the plugin, and presents its main features and various techniques and strategies to backup MySQL with Bacula Enterprise.

Scope

The information presented here applies to **Bacula Enterprise**.

This plugin supports MySQL 4.0.x, 4.1.x, 5.0.x, 5.5.x, 5.6.x., 8.0 as well as MariaDB 10.x.

If you want to back up MySQL versions 8.1 or above, please use BE 18.0.4 or above

Features

The MySQL Plugin is designed to simplify backup and restore of your MySQL server. The backup administrator doesn't need to know MySQL backup techniques or how to write complex scripts. The Plugin will automatically backup essential information such as configuration or user definitions. The MySQL Plugin supports both Dump and binary backup techniques.

The MySQL Plugin is compatible with Copy/Migration jobs. Read the CopyMigrationJobsReplication for more information.

Backup Strategies

The following article presents backup strategies for the MySQL Plugin.

Choosing Between Binary and Dump Modes

The following table article aims at helping you choose between backup techniques supported by the Bacula Enterprise MySQL Plugin. Major functionalities such as being able to restore your database at any point in time, or being able to filter objects during backup or restore should guide you. It is also possible to combine Dump and Binary techniques for the same server.

Functionality	Dump	Binary
Can restore directly a single object (table, schema…)	Yes 1	No
Backup speed	Slow	Fast
Restore speed	Very Slow	Fast
Backup size	Small	Big
Can restore at any point in time	Yes	Yes
Incremental/Differential support	Yes	Yes
Online backup	Yes	Yes
Consistent	Yes	Yes
Can restore to previous major version of MySQL	Yes 2	No
Can restore to newer major version of MySQL	Yes	No

1 To restore a single object, the dump file must be edited.

2 To restore an SQL dump to a previous version of MySQL, you might have to edit the SQL file if you use features that are not available in the previous version. Generally, restoring to a previous version of MySQL is neither supported nor guaranteed.

Dump Mode

During a database's life, MySQL generates logs that can be used to replicate and/or protect your database using P.I.T.R (Point In Time Recovery).

By default, the MySQL Plugin will dump each database one at a time. This means that if you restore the entire server, the databases will be consistent separately, because they were not backed up at exactly the same time, the databases will not be globally consistent. To address this issue, the Bacula Enterprise MySQL Plugin will also save log files generated during the backup. These log files may later be played back to ensure that the databases are consistent at a particular point in time.

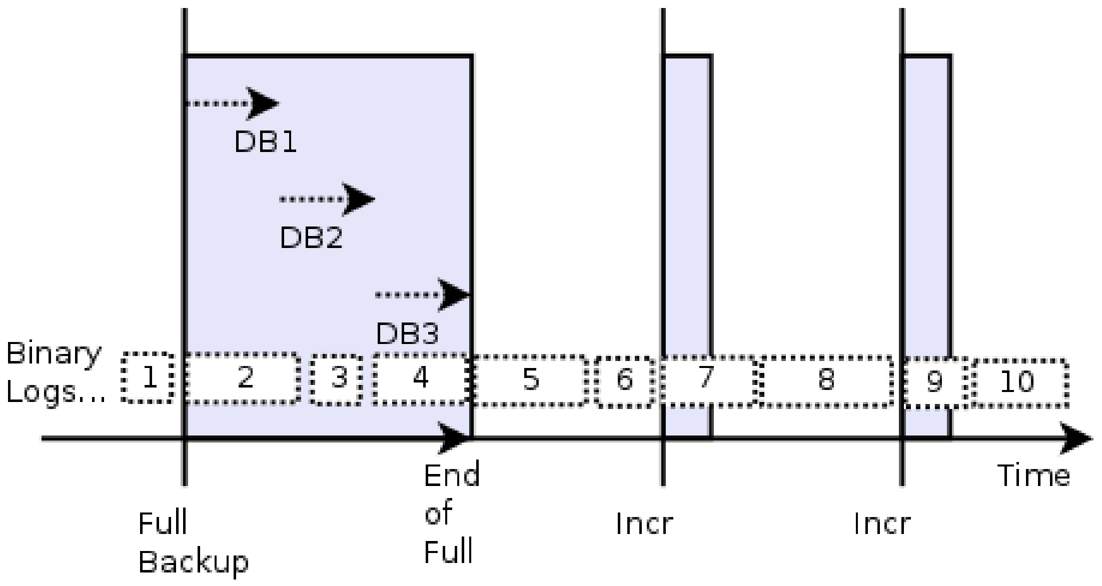


Fig. 110: Interaction between Backup and Binary Logs

In the example presented in the figure above, during the backup of the databases “DB1”, “DB2” and “DB3” (that can take several hours), 3 log files (logs 2, 3, and 4) were generated, and will be included in the Full backup.

The next Incremental or Differential backup will save only new binary logs generated after the Full. To ensure that only one copy of each log file is included in your backup, you should activate the Accurate option for your Job.

In the example shown above, the first Incremental job after the Full backup will include logs 5 and 6, and the second Incremental job will include logs 7 and 8. A Differential backup would include log files 5, 6, 7 and 8.

When you use the `all_databases` option on the Plugin command line, all databases will be dumped at the same time, and the log files will not be flushed at the end of the Full backup, but logs generated before the end of the job are included in the backup. In the example shown in the figure below, the Full backup will generate a single dump `all-databases.sql` and will include log files 2 and 3, but not log file 4. The first subsequent Incremental backup will include log files 4, 5 and 6.

Binary Mode - Percona XtraBackup and Mariabackup

In binary mode, the MySQL Plugin uses Percona XtraBackup, which is an open-source hot backup utility for MySQL based servers that doesn’t need to lock your database during the backup. The Percona technology uses techniques that ensure consistency of the whole backup.

It can back up data from InnoDB, XtraDB, and MyISAM tables on unmodified MySQL 5.0, 5.1, 5.5, and 5.7 servers, as well as a Percona Server with XtraDB. MariaDB is supported with Percona xtrabackup, but starting with Bacula Enterprise 12.6, `Mariabackup` is the recommended backup tool for MariaDB 10.1 and above.

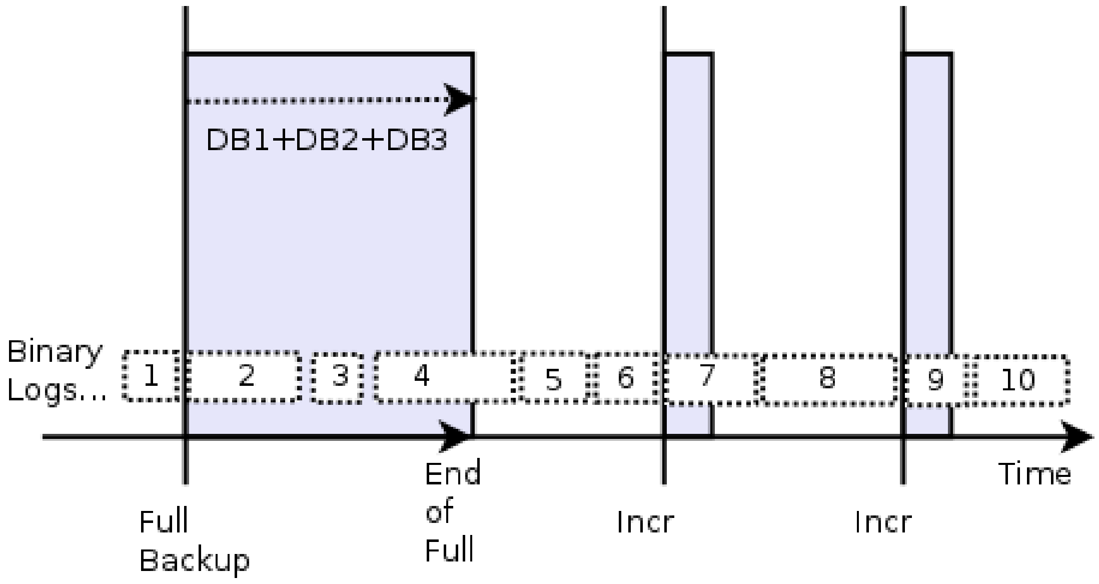


Fig. 111: Interaction between `all_databases` option and Binary Logs

Installation

The following chapter aims at presenting how to install the MySQL Plugin as well as complementary software.

MySQL Plugin Installation

Prerequisites

As with all Bacula plugins, you must specify the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

MySQL Plugin Installation with BIM

In order to install the MySQL Plugin with BIM, install the File Daemon with BIM and choose to install the MySQL Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

MySQL Plugin Installation with Package Manager

The MySQL Plugin is available as a Bacula Enterprise package.

You must also install the plugin on the Client where your MySQL server resides. The MySQL client package, usually “mysql-client” should also be installed, tools such as `mysqldump` and `mysql` must be available to the plugin.

After installing the MySQL Plugin, the File Daemon must be restarted.

Percona Tools Installation

Note: Starting with Bacula Enterprise 12.6, use MariabackupInstallation with MariaDB 10.1 and above instead.

When using Binary Mode, you must install the Percona xtrabackup tool and ensure that the `innobackupex`, xtrabackup, and xbstream programs are properly installed and are available to the plugin.

RPMs and Debs are available on the Percona web site and must be installed prior to usage of the Bacula plugin. More information about Percona and installing the needed programs can be found at:

https://docs.percona.com/percona-xtrabackup/

Mariabackup Installation

Note: Recommended with MariaDB 10.1 and above.

`Mariabackup` is included with MariaDB 10.1.23 and later. It can also be installed with a package manager. See:

https://mariadb.com/kb/en/mariabackup-overview/#installing-with-a-package-manager

Configuration

The following chapter aims at presenting how to configure the MySQL Plugin.

Automatic Objects Integration

Since Bacula version 16.0.7, a new solution has been introduced, so that each object can be backed up separately with different Jobs to maximize the throughput and the resiliency. It is highly recommended to use this new solution for that purpose - *Automatic Object Integration (Scan Plugin)*. See an example for MySQL.

MySQL Specific Configuration

In order to use Point-In-Time Recovery feature of MySQL, you need to enable `log_bin` in the MySQL configuration file and then restart the MySQL server. The procedure may differ between major MySQL versions, so we advise you to read the MySQL documentation corresponding to your server version.

```
log_bin = hostname-bin
```

or

```
log_bin = /U01/hostname-bin
```

If you change the `log_bin` parameter after a Full backup, you will need to schedule a new Full backup to back up binary logs in Incremental level properly.

The Bacula Enterprise MySQL Plugin usually is able to detect the `log_bin` path, however, in some cases you might need to specify the `mysqld` configuration file using `config_file` or the `logbin_dir` plugin command option.

By default, the MySQL Plugin uses the `root` account to dump and read MySQL files. On some systems, it is possible to use the `mysql` account. However, on RedHat systems, this account is locked and this is not possible.

Binary Mode Configuration

With the Binary option, the MySQL Plugin uses `Percona` tools to handle Full, Incremental and Differential backups. You must install those tools before using Binary Mode. For more information, see the PerconaToolsInstallation.

```
Job {
 Name = "MySQL-BIN"
 Client = laptop1-fd
 Fileset = FS_mysql
 ...
}

Fileset {
 Name = FS_mysql
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "mysql: mode=binary abort_on_error"
 }
}
```

When backing up in Binary mode, the MySQL Plugin also accepts the parameters listed in the table (recommended to open in a new tab).

Binary Backup General

When backup using the `mode=binary` plugin option is done, the database will be backed up in binary mode. But since there are multiple databases (even in the case of a single user database), the database will not be consistent when a restore is done. However, during the binary backup, the Percona tools will save and restore the MySQL binary logs that will permit making the databases consistent.

Making the databases consistent is, in Percona terminology, called “Prepare”. This prepare operation is commonly performed when the databases are restored. They are restored to a temporary location, then made consistent using the Prepare option on the Percona tools prior to actually modifying the live database. This Prepare operation requires having sufficient disk for twice the database size, and it consumes CPU and I/O during the process. During the restoration of a large database, the time and resources that the Prepare phase requires can be significantly high, particularly for large databases.

Binary Backup with Prepare

Rather than doing the Prepare work to make the database consistent at restore time, the Prepare can be performed by the plugin automatically during the backup phase by adding the plugin option `prepare` keyword. Prepare has two options `fd` (default) and `sd`.

When the `prepare=fd` option is specified, the prepare will be done on the File Daemon machine at backup time prior to sending the prepared binary data to the Storage Daemon.

Note: Doing the prepare during the backup allows the restore to be done faster (particularly for large databases), but it requires more disk space, CPU and I/O resources. The additional resources might be undesirable if the File Daemon is running on a critical server (see below for a possible solution to this case).

As an alternative to doing the prepare on the File Daemon, it can be done on the Storage Daemon by using the plugin option `prepare=sd`. With this option there is no additional disk space, CPU or I/O required on the File Daemon. However, the additional disk, CPU, and I/O will be used on the Storage Daemon.

Note: If several File Daemons use the `prepare=sd` option at the same time, the load on the Storage Daemon can increase significantly. By having robust Storage daemons or several Storage Daemons, one can largely mitigate the extra overhead imposed on them.

It is possible to specify xtrabackup options in `/etc/my.cnf` in a **[xtrabackup]** section.

Note: The `prepare` option (either for the FD or the SD) works only with a Full backup. Incremental or Differential backups cannot use the prepare option.

Binary Mode Options

Table 35: MySQL Plugin Options in Binary Mode

Option	Comment	Default	Example
mode	Enable Binary backup	dump	mode=binary
unix_user	MySQL Unix user	root	unix_user=mysql
service	MySQL server information	main	service=main
prepare	Use Percona prepare on the File daemon	fd	prepare=fd
prepare	Use Percona prepare on the Storage daemon		prepare=sd
user	MySQL super user	root	user=root
password	MySQL super password		password=xx
backup_softw	MySQL backend (mariadb, xtrabackup, mysql). Used to determine the tools to use.``10``	mysql	backup_software=mariadb
bin_dir	MySQL binaries location		bin_dir=/5.1/bin
bin_format	Binary format (tar or xbstream)	xbstream	bin_format=tar
config_file	Path to my.cnf mysqld configuration file	/etc/mysql/	
mycnf_dir	Path where the MySQL connection `.my.cnf` file is stored		mycnf_dir=/opt/bacula/etc
extra_file	Path to mysql connection file [1]	/root/extra.	
tmp_dir	`WorkingDirectory` Where the plugin will create files and scripts for the database backup. [2]		tmp_dir=/othertmp
timeout	Timeout for SQL queries [3]	60 seconds	timeout=1200
abort_on_err	Abort the job if we have MySQL connection problems [4]		abort_on_error=true
xtrabackup_tmpd	Set `tmpdir` option when doing binary backup [5]		xtrabackup_tmpdir=/tmp/test
xtrabackup_args	Set additional options when doing binary backup [6]		xtrabackup_args="–compress"
restore_extract	Extract xbstream archive automatically at restore [7]		restore_extract
innobackupex_tmpdir	Deprecated [8]		innobackupex_tmpdir=/tmp/test
innobackupex_args	Deprecated [9]		innobackupex_args="–compress"

1 Available with Bacula Enterprise 8.2.4 and later.

2 Available with Bacula Enterprise 6.6.6 and later.

3 Available with Bacula Enteprise 8.6.15.

4 Available with Bacula Enterprise 8.2.9 and later.

5 Available with Bacula Enterprise 8.2.8 and later.

6 Available with Bacula Enterprise 8.2.9 and later.

7 Available with Bacula Enterprise 10.2 and later.

8 Available with Bacula Enterprise 8.2.8 and later. Deprecated since Bacula Enterprise 10.2.

9 Available with Bacula Enterprise 8.2.9 and later. Deprecated since Bacula Enterprise 10.2.

10 May be necessary only in special scenarios such as binary bundle package installations. Available with Bacula Enterprise 16.0.8 and later.

Note that the tar option for bin_format is not compatible with Incremental backups, so only the Full backup will be stored in tar format. Incremental backups will use the xbstream output format.

Dump Mode Configuration

With the Dump option (as opposed to the Binary option), the MySQL server should be configured with the binary log option to perform Incremental and Differential backups.

Note: If the mode plugin option is not specified, the backup will default to mode=dump.

```
Job {
 Name = "Mysql-dump"
 Client = laptop1-fd
 Fileset = FS_mysql_dump
 ...
}

Fileset {
 Name = FS_mysql_dump
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = mysql
 }
}
```

In the above example, the plugin will detect and back up all databases of your server. This simple configuration will work only if the root account is able to connect to the MySQL database. For more complex configurations, refer to the options table.

```
Fileset {
 Name = FS_mysql
 Include {
...
   Plugin = "mysql: database=bacula"
   Plugin = "mysql: database=master"
 }
}
```

In the above example, the plugin will backup databases bacula and master.

```
Fileset {
 Name = FS_mysql
 Include {
...
   Plugin = "mysql: unix_user=admin tmp_dir=/tmp"
 }
}
```

In the above example, the plugin will backup databases using the "admin" Unix user account. This account should be able to connect with all permissions to all databases that you want to dump. You need to make sure that the `tmp_dir` will be writable to your user.

In Dump mode, the MySQL plugin also accepts the parameters listed in the following DumpMode-MySQLPluginOptions.

```
Fileset {
 Name = FS_mysql_dump
 Include {
...
   Plugin = "mysql: user=rob dump_opt=\"--ignore-table=db_name.tbl_name\""
 }
}
```

In this example, the MySQL Plugin will use MySQL account "rob" to perform a dump backup of all databases, and skip the table tbl_name in database db_name.

In order to use `sudo` wrapper, you need to comment out the following option in `/etc/sudoers`.

```
Defaults requiretty
```

The MySQL Plugin permits different dump options for each MySQL version:

Table 36: MySQL Dump options

Version	Option
>= 4.1.18	`--single-transaction --opt --extended-insert --create-options --default-character-set=utf8`
>= 5.0	4.1.18 options + `--routines --master-data=2`
>= 4.1	`--single-transaction --opt --extended-insert -all --default-character-set=utf8`
3.x	`--skip-lock-tables --opt --extended-insert --create-options --default-character-set=utf8`

Backup Level in Dump Mode

When using Dump mode, depending on the Job level, the MySQL Plugin will do the following:

- For **Full** backups, the Plugin will backup all databases and logs generated during the backup.
- For **Incremental** backups, the Plugin will flush the current log and will backup all logs generated since the last backup.
- For **Differential** backups, the Plugin will flush the current log and will backup all logs generated since the last Full backup.

Dump Mode Options

Table 37: MySQL Plugin Options in Dump Mode

Option	Comment	Default	Example
dump_opt	This string will be passed to the mysqldump command		dump_opt="-X"
exclude_tabl	Exclude a table. Multiple parameter is allowed. If the argument points to a file on the client, each line will be excluded. 5		exclude_table=mysql
unix_user	Unix user to use for MySQL commands	root	unix_user=robert
service	MySQL server name		service=main
mycnf_dir	Path where MySQL .my.cnf file is stored		my_cnf=/tmp
use_sudo	Use sudo instead to run MySQL commands (when not root)		use_sudo
database	Will backup on databases matching this string		database=prod*
all_databa	Will generate a single dump of all databases		
bin_dir	MySQL binaries location		bin_dir=/opt/mysql/bin
user	MySQL super user	root	user=root
password	MySQL super password		password=xx
logbin_dir	mysqld log_bin directory		
config_file	Path to my.cnf mysqld configuration file	/etc/mysql/my.cn	
extra_file	Path to mysql connection file 1	/root/my.cnf	
character_set	Character set used to dump data	utf8	character_set=utf8
tmp_dir	Where the MySQL plugin will create files and scripts for the database backup 2	/tmp	tmp_dir=/othertmp
timeout	Timeout for SQL queries 3	60 seconds	timeout=1200
skip_missi	Send a SKIPPED message instead of a warning when a database disapears during a job 6		skip_missing_db
abort_on_	Abort the job if we have MySQL connection problems 4		abort_on_error=true
backup_so	MySQL backend (mariadb, mysql). Used to determine the tools to use.``7``	mysql	backup_software=mariadb

1 Available with Bacula Enterprise 8.2.4 and later.

2 Available with Bacula Enterprise 6.6.6 and later.

3 Available with Bacula Enteprise 8.6.15.

4 Available with Bacula Enterprise 8.2.0 and later.

5 Available with Bacula Enterprise 14.0 and later.

6 Available with Bacula Enterprise 14.0.4 and later.

7 May be necessary only in special scenarios such as binary bundle package installations. Available with

Bacula Enterprise 16.0.8 and later.

MySQL Connection Information

If you are using specific connection options such as:

- TCP connection
- Non-standard port
- Password
- and others,

it is possible to create a configuration file to store these settings and use them with the Bacula Enterprise Plugin. For example, it avoids having the password exposed on the Plugin command string.

The connection file should be specified with the `extra_file` plugin command line option. In the connection file, the `[client]` section is a shortcut for all required context.

Note: The the `extra_file` plugin option was introduced in Bacula Enterprise 8.2.4 and available with MySQL 5.0.6.

```
# cat /opt/bacula/etc/database1.cnf
[client]
user=admin
password=admin1
socket=/tmp/mysql.sock

# Plugin = "mysql: extra_file=/opt/bacula/etc/database1.cnf"
```

In `bin` backup mode, the xtrabackup tool doesn't read the `.my.cnf` to get connection information. To use the binary backup mode, it is mandatory to specify the password on the Plugin command line or to use the `extra_file` plugin command option.

It is also possible to use the user specific `.my.cnf` MySQL ini file that should contain information for `client` programs such as `mysql`, `mysqldump`.

```
# comment
[client]
password=rootroot
```

MySQL programs will search the `.my.cnf` file in the HOME directory by default. With the Bacula Enterprise MySQL Plugin, the `.my.cnf` file can be stored anywhere on your system. The use of the `mycnf_dir` Fileset option permits to specify the directory where this file is stored.

```
# cat /opt/bacula/etc/.my.cnf
[client]
user=admin
password=admin1

# Plugin = "mysql: mycnf_dir=/opt/bacula/etc"
```

Testing Database Access Configuration

You can use the Bacula `estimate` command to verify that the MySQL plugin is well configured.

```
* estimate listing job=my-test
...
```

If the `estimate` or the job output display the following error,

```
Error: Can't reach MySQL server to get database config.
```

you should check that the Bacula Enterprise MySQL Plugin can retrieve information using the `mysql` command running as the `mysql` user on the Client.

You must ensure that your `unix_user` user can connect to the MySQL server without any password prompt. By default, the `unix_user` is root.

If you need to specify options such as `-h localhost` in the `mysql` command line, you will need to use a my.cnf file as described in MySQL Connection Information.

Error Log and Debug Information

With the MySQL Plugin, MySQL error messages generated by commands are automatically included in the job log.

For example:

```
Fatal error: Can't reach MySQL server to get database config. ERR=268435457
Error: ERROR 1045 (28000): Access denied for user 'backup_user'@'localhost'␣
↪(using password: YES)
```

If you need more details about the MySQL errors or if no error message is present in the backup job log, you can enable debug level 200 on File Daemon to not delete log files in the `/tmp` directory of the File Daemon host at the end of the job.

```
* setdebug level=200 trace=1 options=t client=mysqlserver-fd
```

For example, after enabling debug level 200 on `mysqlserver-fd`, the following files will be generated:

```
root@mysqlserver:/tmp# ls
cmd.24.sh.4OQxiQ
cmd.24.sh.OCuiKJ
cmd.24.sql.0jRa7W
cmd.24.sql.SxlbWM
grants.24.sql.yiwPHT
mysql.24.log
```

```
  Error: Pipe close error 2 on /@MYSQL/main/MyDatabase/data.sql
(sh -c 'mysqldump --opt --extended-insert --create-options --single-
↪transaction
       --default-character-set=utf8 --routines --errorParameter "MyDatabase"
       2>>/tmp/mysql.24.log'): ERR=Child exited with code 2
```

We can see in the output log file an incorrect parameter configured in the Fileset plugin line:

```
root@mysqlserver:/tmp# cat mysql.24.log
mysqldump: unknown option '--errorParameter'
```

Note: If you often encounter errors of the lost connection to MySQL server, visit MySQL Community documentation for advice.

Operations

The following chapter aims at presenting possible operations with the MySQL Plugin.

Backup

Estimate Information

The estimate command will display all information found by the MySQL plugin. For Dump mode, Bacula cannot estimate the dump size for databases, so it will display database size instead.

Backup Information in Dump Mode

The MySQL Plugin will generate the following files entries in the Bacula catalog for a server having a single database “test”.

```
/etc/mysql/my.cnf
@MYSQL/main/gobal-grants.sql
@MYSQL/main/settings.txt

@MYSQL/main/test/createdb.sql
@MYSQL/main/test/schema.sql
@MYSQL/main/test/data.sql
@MYSQL/main/test/grants.sql

@MYSQL/main/logs/mysql-bin.000001
```

Table 38: Backup Content in Dump Mode

File	Context	Comment
global-grants.sql	global	List of all users, their password and specific options
settings.txt	global	Current variables for the mysql server
my.cnf	global	MySQL server configuration
createdb.sql	database	Database creation script
schema.sql	database	Schema database creation script
data.sql	database	Database data in dump format
grants.sql	database	List of all users associated to the database

Restore

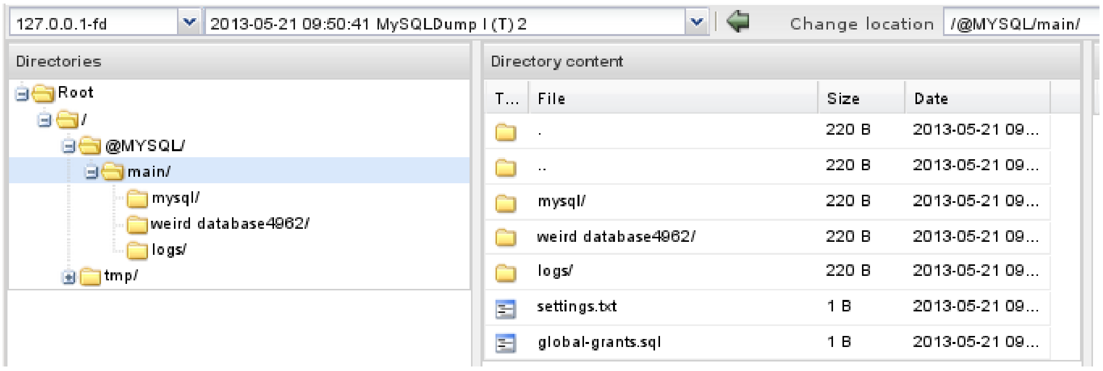
Restoring Using Dumps

Restoring Users and Roles

To restore roles and users to your MySQL server, you just select the `global-grants.sql` file located in: `/@MYSQL/<service>/global-grants.sql`.

Important: With MySQL >= 5.7, users must be created prior to the GRANT command with: `CREATE USER username`

Then, using `where=/` or `where=` the plugin will load this SQL file into your database. If some roles already exist, errors will be printed in the Job log. Note that it is possible to restore the `global-grants.sql` file to a local directory, edit the file and load it with `mysql` to restore only a particular selection.

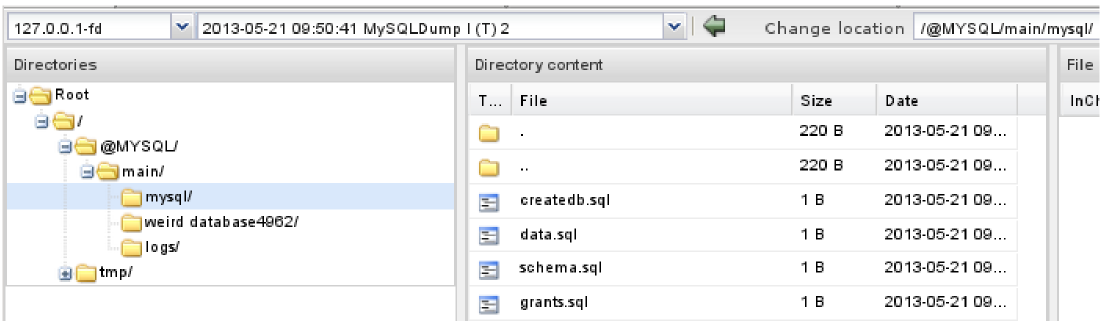


T...	File	Size	Date
🗀	.	220 B	2013-05-21 09...
🗀	..	220 B	2013-05-21 09...
🗀	mysql/	220 B	2013-05-21 09...
🗀	weird database4962/	220 B	2013-05-21 09...
🗀	logs/	220 B	2013-05-21 09...
🗎	settings.txt	1 B	2013-05-21 09...
🗎	global-grants.sql	1 B	2013-05-21 09...

Fig. 112: MySQL Server content during restore

Restoring Single Database

To restore a single database with the Bacula Enterprise MySQL Plugin, you need only to select the database directory in the restore command, the selection should contain the data file (`data.sql`) and the database creation script (`createdb.sql`).



T...	File	Size	Date
🗀	.	220 B	2013-05-21 09...
🗀	..	220 B	2013-05-21 09...
🗎	createdb.sql	1 B	2013-05-21 09...
🗎	data.sql	1 B	2013-05-21 09...
🗎	schema.sql	1 B	2013-05-21 09...
🗎	grants.sql	1 B	2013-05-21 09...

Fig. 113: Database content during restore

When the database directory is selected, you can use the `where` parameter to restore the database to a new database. If you set `where` to a single word that contains only `a..z`, `0-9`, `.` and `_`, Bacula will create the specified database and restore data into it.

```
* restore where=bacula.old
```

If you set the `replace` parameter to `never`, Bacula will check the database list, and will abort the Job if the database currently restored already exists.

Using `replace=always` is not recommended.

If the `where` parameter is a directory (containing /), Bacula will restore all files into this directory. Doing so, you will be able to use `mysql` directly and do manual restores.

Restoring Dump Files To Directory

To restore SQL dumps to a directory, you set the `where` parameter to a valid directory.

```
* restore where=/tmp
```

Although the database name below is `mydb`, the following `.sql` files should be restored in the `/tmp`:

```
# ls -lR /opt/bacula/archive/bacula-restores/
/opt/bacula/archive/bacula-restores/:
total 4
drwxr-xr-x 3 root root 4096 Nov 18 10:20 @MYSQL

/opt/bacula/archive/bacula-restores/@MYSQL:
total 4
drwxr-xr-x 3 root root 4096 Nov 18 10:20 main

/opt/bacula/archive/bacula-restores/@MYSQL/main:
total 4
drwxrwsr-x 2 root 1001 4096 Nov 18 10:23 mydb

/opt/bacula/archive/bacula-restores/@MYSQL/main/mydb:
total 12
-rw-r----- 1 root root    0 Nov 15 16:04 createdb.sql
-rw-r----- 1 root root 2675 Nov 18 10:22 data.sql
-rw-r----- 1 root root   18 Nov 15 16:04 grants.sql
-rw-r----- 1 root root 2072 Nov 18 10:23 schema.sql
```

The database(s) can be restored using the `mysql` commands:

```
mysql --host <mysql_server_ip_or_FQDN> -uroot -p"<password>" --execute
↪"CREATE DATABASE mydb"
mysql --host <mysql_server_ip_or_FQDN> -uroot -p"<password>" --database mydb
↪< schema.sql
mysql --host <mysql_server_ip_or_FQDN> -uroot -p"<password>" --database mydb
↪< data.sql
mysql --host <mysql_server_ip_or_FQDN> -uroot -p"<password>" --database mydb
↪< grants.sql
```

Refer to the MySQL documentation for more details about the MySQL Command-Line Client options: https://dev.mysql.com/doc/.

PITR Using Binary Logs

Point-In-Time Recovery refers to recovery of data changes made up to a given point in time. Typically, this type of recovery is performed after restoring a full backup that brings the server to its state as of the time the backup was made.

To restore data from the binary log, aside from adding the `logs/` folder from the plugin file tree, you also must know the name and location of the current binary log files when the backup was made. This information is available in the "CHANGE MASTER" line on the top of the `data.sql` file.

```
-- Position to start replication or point-in-time recovery from

-- CHANGE MASTER TO MASTER_LOG_FILE='sql-bin.000004', MASTER_LOG_POS=2083;
```

This information is also printed in the Bacula job report when restoring a dump directly into a new database using `where=newdb` parameter.

```
...
Found MASTER_LOG position sql-bin.000004:2083 for "database5276"
...
```

Once you have this information and all log files generated between the Full backup and the point in time when you want to restore, you need to use the `mysqlbinlog` program.

```
# mysqlbinlog -j 2083 sql-bin.000004 sql-bin.000005...
```

This command will generate an SQL script that you can load into your restored database to run the *recover* process. You may want to stop the *recover* process in a middle of a log file, for that, `mysqlbinlog` provides several options such as `--stop-datetime` to control this behavior. Refer to the `mysqlbinlog` documentation for all parameters:

http://dev.mysql.com/doc/refman/5.1/en/mysqlbinlog.html.

As the output of `mysqlbinlog` program is an SQL script, you can also edit the script to fit your needs. For example, if the database has a new name, you will need to edit the SQL script to change database references.

```
# mysqlbinlog -j 2083 mysql-bin.000004 ... | \
   sed 's/use `orgname`/use `newname`/'    | \
   mysql -u root newname
```

For more information on PITR with MySQL, refer to the MySQL documentation:

https://dev.mysql.com/doc/refman/8.0/en/point-in-time-recovery.html

Restoring Single Table

To restore a single item such as a table, you currently need to restore the dump file to a directory and use the `mysql` command.

```
$ sed -n -e '/Table structure for table .mytable.$/,/Table structure for␣
↪table/p' data.sql
```

The above `sed` command will extract the table structure, the index and the data from the dump.

Restoring Complete Server

To restore the all-databases and the server configuration, just select all files located in `/@MYSQL/<service>` except bin-log in `logs` directory, use `replace=always` and `where=/` options.

If you are using MySQL BinLog, you will need to apply bin logs after the restore using instruction described in pitrbinlog.

Restoring Using MySQL Command-Line Tool

It is possible to use the MySQL Command-Line Tool to restore from the dump files generated by backup jobs using the MySQL plugin in Dump mode.

The MySQL plugin generate, in a Dump mode backup, the files related in Table DumpMode-MySQLPluginOptions.

The MySQL plugin will generate the following files entries in the Bacula catalog for a server having a single database "test":

```
/etc/mysql/my.cnf
@MYSQL/main/gobal-grants.sql
@MYSQL/main/settings.txt

@MYSQL/main/test/createdb.sql
@MYSQL/main/test/schema.sql
@MYSQL/main/test/data.sql
@MYSQL/main/test/grants.sql
```

To restore the single database "test", please follow the below sequence of commands.

At this time, the createdb.sql file is a dummy file used to create the database if selected in the restore process. We might enhance the plugin to add the SQL command used to create a database. So please manually create the database:

```
# mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
...
mysql> create database test;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
```

Then restore schema, data and privileges:

```
# mysql -u root --database=test < /path/to/\@MYSQL/main/test/schema.sql
# mysql -u root --database=test < /path/to/\@MYSQL/main/test/data.sql
# mysql -u root --database=test < /path/to/\@MYSQL/main/test/grants.sql
```

When restoring the MySQL server or the "mysql" database, it may be needed to restore global privileges:

```
# mysql -u root < /path/to/\@MYSQL/main/global-grants.sql
```

The “settings.txt” file contains the current variables for the MySQL server. This file is not used automatically by the restore process. Its content can be used to restore the current MySQL server settings or to re-configure a MySQL server on a new system, for example.

Restoring Complete Server Using Binary Mode (Percona)

In binary mode, the Bacula MySQL Plugin uses the Percona xtrabackup tools. You must have the Percona tools installed.

You can find useful information in the Percona manual: http://www.percona.com/doc/percona-xtrabackup/?id=percona-xtrabackup:start

The details of the restore depend on whether or not you used the `prepare` option or not. If you did use the `prepare` option, please see the next section RestoringWithAutomaticExtraction, otherwise please see the section entitled RestoringBinaryModeBackupWithoutPrepare.

Restoring with Automatic Extraction

Available since Bacula Enterprise 10.2.3

Bacula can extract xbstream automatically at restore.

To do so, either specify `restore_extract` in the Plugin line

```
Plugin = "mysql: restore_extract"
```

Or overwrite the behavior in bconsole by modifying the plugin Options.

```
Automatically selected Client: localhost-fd
Using Catalog "MyCatalog"
Run Restore job
...
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: Fileset
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : mysql: mode=bin"
Plugin Restore Options
```

(continues on next page)

(continued from previous page)

```
Option                Current Value        Default Value
manual_restore:       *None*               (yes)
extract_restored_xbstream: *None*                (no)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
     1: manual_restore (Do not try to start the recover process)
     2: extract_restored_xbstream (When xbstream is restored, automatically␣
↪extract it)
Select parameter to modify (1-2): 2
Please enter a value for extract_restored_xbstream: yes
Plugin Restore Options
Option                Current Value        Default Value
manual_restore:       *None*               (yes)
extract_restored_xbstream: yes                   (no)
Use above plugin configuration? (yes/mod/no): yes
```

Restoring Binary Mode Backup with Prepare

If the Prepare has already been done during the backup because you used the `prepare` option on the plugin, you will not need to manually do the Prepare. This can save considerable time.

First, you should use the Bacula `restore` and select a Full backup to be restored. Once the files are restored to a suitable directory, you will find something similar to the following directory structure:

```
@MYSQL/main/all-databases.xbstream
@MYSQL/main/my.cnf
@MYSQL/main/mysql.dat
```

Once you have restored the backup content with Bacula, if you didn't choose automatic extraction with `restore_extract` or `extract_restored_xbstream`, you can then restore the data using the -x option on xbstream (or `mbstream` for MariaDB version 10 or greater) as in the following example:

```
% cd @MYSQL/main
%ls
all-databases.xbstream my.cnf   mysql.dat
# Now extract the all-databases.xbstream
% xbstream -x < all-databases.xbstream
% ls
all-databases.xbstream mysql              xtrabackup_checkpoints
backup-my.cnf          performance_schema  xtrabackup_info
ib_buffer_pool         regress            xtrabackup_logfile
ibdata1                sys
my.cnf                 testdb
```

Now the files in the local directory are ready to be used by the server. The `--copy-back` option on `innobackupex` will copy the prepared data back to its original location as defined by the datadir in your `my.cnf`. Note that you can use `--defaults-file=/path/to/my.cnf` to specify the `my.cnf` configuration file.

However, before innobackupex will allow you to overwrite the original MySQL data files, you must either move them or remove them. For example, either:

```
% rm -rf /var/lib/mysql
```

or

```
% mv /var/lib/mysql /var/lib/mysql.old
```

If you are running on a server where the MySQL data files are kept in a different directory, you will need to adjust the above paths.

Now you can actually copy the files back with:

```
% innobackupex --copy-back $PWD
...
120604 02:58:44  innobackupex: completed OK!
```

For MariaDB version 10 or greater, use the `mariabackup` command with the parameter `--innobackupex` and the same arguments instead:

```
% mariabackup --innobackupex --copy-back $PWD
```

You should check the file permissions after copying the data back. You may need to adjust them with something like:

```
% chown -R mysql:mysql /var/lib/mysql
```

Now the datadir contains the restored data. You are ready to re-start the server, typically with something like:

```
% service mysql start
```

If the MySQL server does not start it may be due to SELinux or AppArmor. If SELinux is enabled, please reset the security context as **root** with the command below:

```
# restorecon -R /var/lib/mysql/
```

If the restart issue is due to AppArmor denials, until you solve the problems, you can temporarily disable apparmor with the following command:

```
% apparmor_parser -R /etc/apparmor.d/usr.sbin.mysqld
```

The opposite of the above is the following:

```
% apparmor_parser -a /etc/apparmor.d/usr.sbin.mysqld
```

Note: In the above two examples, `mysqld` is spelled with a d on the end in contrast to the service name where the name is simply `mysql`.

Finally, if you want to completely remove your database and create a new empty one, the following commands may help:

```
% rm -rf /var/lib/mysql          # Remove any old database setup
% mysql_install_db -u mysql      # Install new database
% systemctl unmask mysql.service # Emables the service for systemd
% service mysql start            # start the service.
```

Restoring Binary Mode Backup without Prepare

If you have done your backup with the `prepare` keyword on the plugin directive you should go back to the previous section section as the restored backup prepare has already been done.

Once you have restored the backup content with Bacula, files using the `tar` format should be extracted with `tar -i` option. With xbstream format, if you didn't choose automatic extraction with `restore_extract` or `extract_restored_xbstream`, you can extract data with the `-x` option.

```
% cd @MYSQL/main
% xbstream -x < all-databases.xbstream
% ls
all-databases.xbstream  ibdata1.delta          performance_schema
xtrabackup_logfile      ibdata1.meta           testdb
backup-my.cnf           xtrabackup_checkpoints mysql
xtrabackup_binary       xtrabackup_binlog_info
```

When the files are uncompressed you can prepare the backup with the `--apply-log` option of the `innobackupex` tool. If you plan to apply incremental backups, you need also to use the `--redo-only` option. For MariaDB the `mariabackup` command must have the `--innobackupex` parameter so that it will mimic `innobackupex` below.

```
% innobackupex --apply-log --redo-only $PWD
...
120604 02:50:02  innobackupex: completed OK!
```

Each Incremental should be extracted in a specific directory, then they should be applied to the base data.

```
% mkdir incr1
% cd incr1
% xbstream -x < ../all-databases-1220202.xbstream
% cd ..
% innobackupex --apply-log --redo-only --incremental-dir=incr1 $PWD
...
120604 02:51:02  innobackupex: completed OK!

% mkdir incr2
% cd incr2
% xbstream -x < ../all-databases-1320402.xbstream
% cd ..
% innobackupex --apply-log --redo-only --incremental-dir=incr2 $PWD
...
120604 02:52:02  innobackupex: completed OK!
```

When the files are uncompressed you can prepare the backup with the `--apply-log` option of the `innobackupex` tool:

```
% innobackupex --apply-log $PWD
...
120604 02:51:02  innobackupex: completed OK!
```

Now the files in the local directory are ready to be used by the server. The `--copy-back` option will copy the prepared data back to its original location as defined by the datadir in your `my.cnf`. Note that you can use `--defaults-file=/path/to/my.cnf` to specify the `my.cnf` configuration file.

```
% innobackupex --copy-back $PWD
...
120604 02:58:44  innobackupex: completed OK!
```

You should check the file permissions after copying the data back. You may need to adjust them with something like:

```
% chown -R mysql:mysql /var/lib/mysql
```

Now the datadir contains the restored data. You are ready to start the server.

Use Cases

The following article presents use cases for the MySQL Plugin.

Percona XtraDB Cluster in Kubernetes

This document outlines the procedure for utilizing the Bacula MySQL Plugin running in a container to perform backups of the Percona XtraDB Cluster.

For additional details regarding Percona XtraDB Cluster, visit: https://www.percona.com

The software versions used in this use case:

- Percona XtraDB Cluster: *Server version: 8.0.36-28.1 Percona XtraDB Cluster (GPL), Release rel28, Revision bfb687f, WSREP version 26.1.4.3*
- MySQL Client: *mysql Ver 8.0.37 for Linux on x86_64 (MySQL Community Server - GPL)*
- Percona XtraBackup: *xtrabackup version 8.0.35-31 based on MySQL server 8.0.35 Linux (x86_64) (revision id: 55ec21d7)*

Bacula File Daemon and MySQL Plugin Kubernetes Deployment

Dockerfile

Use the following Dockerfile to build a Bacula File Daemon with the MySQL Plugin installed image.

```
# cat Dockerfile
FROM debian:bullseye
LABEL maintainer="Bacula Systems SA"
LABEL version="18.0.5"
LABEL name="Bacula Enterprise Edition Client"
LABEL vendor="BACULA SYSTEMS SA"
LABEL summary="This is a Bacula File Daemon with the MySQL plugin"
LABEL description="This image contains a Bacula File Daemon which allows␣
↪connection between this pod resources and the Bacula Director."
# Update image
RUN apt update
# install required dependencies for the mysql client and Percona XtraBackup
RUN apt install lz4 ca-certificates curl gnupg lsb-release wget -y
# install the Bacula File Daemon and the MySQL Plugin
```

(continues on next page)

(continued from previous page)

```
RUN mkdir -p /etc/apt/keyrings
RUN curl -fsSL https://qa.baculasystems.com/dl/QA_SL_2-
↪adsfJLU783jklAKjd667aJKNyX/BaculaSystems-Public-Signature-08-2017.asc -o /
↪etc/apt/keyrings/bacula.asc
RUN chmod a+r /etc/apt/keyrings/bacula.asc
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
↪bacula.asc] https://qa.baculasystems.com/dl/QA_SL_2-
↪adsfJLU783jklAKjd667aJKNyX/debs/bin/18.0.5/bullseye-64/ bullseye main" > /
↪etc/apt/sources.list.d/Bacula-Enterprise-Edition.list
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
↪bacula.asc] https://qa.baculasystems.com/dl/QA_SL_2-
↪adsfJLU783jklAKjd667aJKNyX/debs/mysql/18.0.5/bullseye-64/ bullseye mysql" >␣
↪/etc/apt/sources.list.d/Bacula-Enterprise-Edition-mysql-plugin.list
RUN apt-get update
RUN apt-get install -y bacula-enterprise-client bacula-enterprise-mysql-plugin
# download and install the mysql client for Percona/MySQL 8.0
RUN wget https://repo.mysql.com/mysql-apt-config_0.8.29-1_all.deb
RUN DEBIAN_FRONTEND=noninteractive dpkg -i mysql-apt-config_0.8.29-1_all.deb
RUN apt-get update
RUN apt -y install mysql-client
RUN apt list --installed | grep mysql > /tmp/apt_mysql.txt
# download and install Percona XtraBackup 8.0
RUN wget https://repo.percona.com/apt/percona-release_latest.$(lsb_release -
↪sc)_all.deb
RUN DEBIAN_FRONTEND=noninteractive dpkg -i percona-release_latest.$(lsb_
↪release -sc)_all.deb
RUN percona-release enable pxb-80 release
RUN apt -y install percona-xtrabackup-80
RUN apt autoremove
# use the bacula-fd.conf file previously configured for a specific Bacula␣
↪Director / configuration
RUN rm /opt/bacula/etc/bacula-fd.conf
COPY bacula-fd.conf /opt/bacula/etc/bacula-fd.conf
# copy required my.cnf and .my.cnf files to the bacula-fd with the specific␣
↪MySQL configuration needed to access the Percona cluster
COPY my.cnf /opt/bacula/etc
COPY .my.cnf /opt/bacula/etc
#  expose bacula-fd port
EXPOSE 9102
USER root
# Start the Bacula File Daemon service
CMD ["/opt/bacula/bin/bacula-fd", "-f"]
```

The `.my.cnf` file used in this use case:

```
~/bacula-fd-mysql# cat .my.cnf
[client]
host=10.43.198.239 <--- The ClusterIP address of the pxc-db-haproxy
user=root
password=2R7D9D.6V1I+5Z,F$ <--- root password get from the `pxc-db-
↪secrets`
```

```
# kubectl get service -n pxc
NAME                              TYPE        CLUSTER-IP      ␣
↪EXTERNAL-IP   PORT(S)                              AGE
bacula-fd                         ClusterIP   10.43.80.87     10.0.
↪97.201   9102/TCP                             5d22h
my-db-pxc-db-haproxy              ClusterIP   10.43.4.195     <none>␣
↪        3306/TCP,3309/TCP,33062/TCP,33060/TCP   12d
my-db-pxc-db-haproxy-replicas     ClusterIP   10.43.54.231    <none>␣
↪        3306/TCP                                12d
my-db-pxc-db-pxc                  ClusterIP   None            <none>␣
↪        3306/TCP,33062/TCP,33060/TCP            12d
my-db-pxc-db-pxc-unready          ClusterIP   None            <none>␣
↪        3306/TCP,33062/TCP,33060/TCP            12d
percona-xtradb-cluster-operator   ClusterIP   10.43.248.174   <none>␣
↪        443/TCP                                 12d
pxc-db-haproxy                    ClusterIP   10.43.198.239   <none>␣
↪        3306/TCP,3309/TCP,33062/TCP,33060/TCP   12d
```

```
# kubectl -n pxc get secrets pxc-db-secrets -o jsonpath="{.data.root}
↪" | base64 --decode
```

Build an image using the Dockerfile, and tag/push it to a local registry in the Kubernetes Cluster.

Important Notes

- The *mysql-client* and the *Percona XtraBackup for MySQL* versions installed must be compatible with the Percona XtraDB Cluster MySQL version.
- Have a `.my.cnf` file with user-specific options to allow the connection from the MySQL Plugin to the Percona XtraDB Cluster.
- Have the `my.cnf` file used by the Percona XtraDB Cluster, or use a Kubernetes configmap.
- Have the `bacula-fd.conf` file previously configured (the Director the File Daemon will use, for example), or use a Kubernetes configmap for a valid `bacula-fd.conf` file.
- It is possible to use any base image - Debian Bullseye is used in this use case - but dependencies and external programs versions could change.

Create bacula-fd Deployment in Kubernetes Cluster

After creating the *bacula-fd* deployment in the Kubernetes Cluster, attach one of the *datadir* persistent volumes, used by the Percona Cluster, to the *bacula-fd* pod, and mount the volumes in the *bacula-fd* container.

It is important to provide an external access for the communication between the Bacula File Daemon in the Kubernetes Cluster and both the Director and the Storage Daemon, when the services run outside the Kubernetes Cluster.

In this use case, the *EXTERNAL-IP* configured to the *bacula-fd* service is the ip address of one of the Kubernetes Master nodes:

```
# kubectl get service -n pxc
NAME                                 TYPE        CLUSTER-IP      ␣
↪EXTERNAL-IP   PORT(S)                                   AGE
bacula-fd                            ClusterIP   10.43.80.87     10.0.
↪97.201   9102/TCP                                  5d22h
```

An External Load Balancer or Ingress can be used as well.

Backup and Restore using bacula-fd Service in Kubernetes Cluster

Using the MySQL Plugin, both DUMP and Binary modes can be used to backup the Percona XtraDB Cluster.

For details about the MySQL Plugin backup configuration, refer to the *main MySQL Plugin page*.

Fileset Configuration

The Fileset configuration used in this use case for DUMP and Binary mode backups:

```
Fileset {
    Name = "percona-dump-fileset"
    Include {
      Options {
        IgnoreCase = yes
        OneFs = no
        Signature = Md5
      }
      Plugin = "mysql: debug verbose abort_on_error mycnf_dir=\"/opt/
↪bacula/etc/\" config_file=\"/opt/bacula/etc/my.cnf\""
    }
    Exclude {
    }
}
```

```
Fileset {
    Name = "percona-binary-fileset"
    Include {
      Options {
        IgnoreCase = yes
        OneFs = no
        Signature = Md5
      }
      Plugin = "mysql: debug abort_on_error mode=\"binary\" mycnf_
↪dir=\"/opt/bacula/etc\" config_file=\"/opt/bacula/etc/my.cnf\"␣
↪backup_software=\"xtrabackup\""
    }
    Exclude {
    }
}
```

Restore - BINARY mode - Complete Server - without preparation

To allow a restore of the data in the datadir Percona Cluster persistent volumes, one of the datadir persistent volumes must be mounted to the *bacula-fd* container as ReadWrite Access Mode.

It is recommended to stop the Percona XtraDB Cluster to perform the complete server restore:

```
# kubectl -n pxc patch --type=merge --patch='{ "spec": { "pause":␣
↪true } }' pxc pxc-db
perconaxtradbcluster.pxc.percona.com/pxc-db patched
```

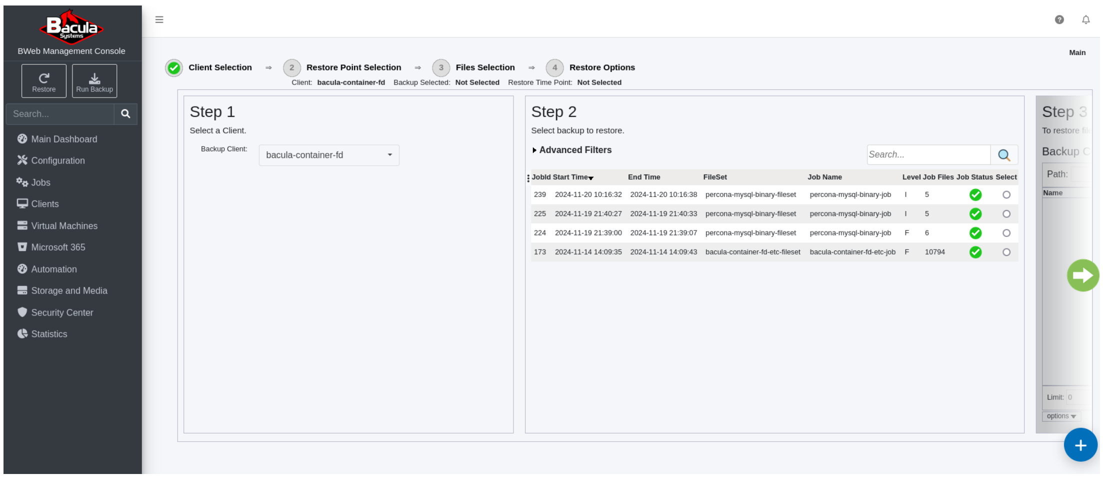
And remove all, if any, existent files from the current `/var/lib/mysql` directory in the datadir persistent volume. This operation can be done from the *bacula-fd* container.

Restore Files into Local Directory

Run a restore by selecting either Full or Incremental job corresponding to the specific point in time when the MySQL server should be restored. The files will be restored in a local directory of the *bacula-fd* container:

```
root@bacula-fd-86bfd77786-98qxm:/opt/bacula/archive/bacula-restores#␣
↪ls -lR
.:
total 8
drwxrwsr-x 3 root 1001 4096 Nov 20 14:43 @MYSQL
drwxr-xr-x 3 root root 4096 Nov 21 09:05 opt

./@MYSQL:
total 4
drwxrwsr-x 2 root 1001 4096 Nov 20 14:43 main

./@MYSQL/main:
total 82764
-rw-r----- 1 root 1001  2020282 Nov 20 14:41 all-databases.34206973.
↪xbstream
-rw-r----- 1 root 1001  1954746 Nov 20 14:42 all-databases.34208728.
↪xbstream
-rw-r----- 1 root 1001 80760446 Nov 20 14:40 all-databases.xbstream
-rw-r--r-- 1 root root        9 Nov 15 14:13 my.cnf
-rw-r--r-- 1 root root      536 Nov 20 14:43 mysql.dat
```

This is an example to restore one Full and two Incremental jobs using BWeb:



Client: **bacula-container-fd** Backup Selected: **Not Selected** Restore Time Point: **Not Selected**

Step 1

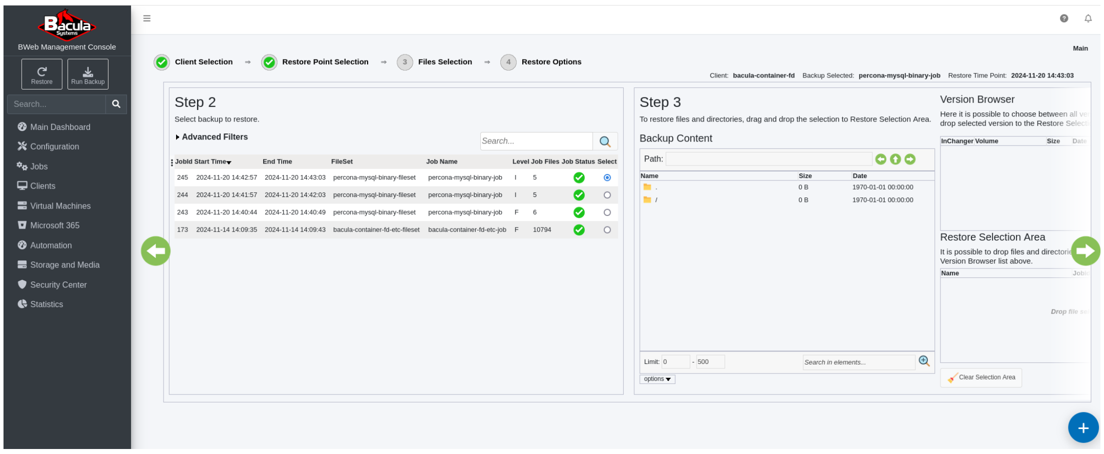
Select a Client.

Backup Client: bacula-container-fd

Step 2

Select backup to restore.

JobId	Start Time	End Time	FileSet	Job Name	Level	Job Files	Job Status	Select
239	2024-11-20 10:16:32	2024-11-20 10:16:38	percona-mysql-binary-fileset	percona-mysql-binary-job	I	5	✓	○
225	2024-11-19 21:40:27	2024-11-19 21:40:33	percona-mysql-binary-fileset	percona-mysql-binary-job	I	5	✓	○
224	2024-11-19 21:39:00	2024-11-19 21:39:07	percona-mysql-binary-fileset	percona-mysql-binary-job	F	6	✓	○
173	2024-11-14 14:09:35	2024-11-14 14:09:43	bacula-container-fd-etc-fileset	bacula-container-fd-etc-job	F	10794	✓	○



Client: **bacula-container-fd** Backup Selected: **percona-mysql-binary-job** Restore Time Point: **2024-11-20 14:43:03**

Step 2

Select backup to restore.

JobId	Start Time	End Time	FileSet	Job Name	Level	Job Files	Job Status	Select
245	2024-11-20 14:42:57	2024-11-20 14:43:03	percona-mysql-binary-fileset	percona-mysql-binary-job	I	5	✓	●
244	2024-11-20 14:41:57	2024-11-20 14:42:03	percona-mysql-binary-fileset	percona-mysql-binary-job	I	5	✓	○
243	2024-11-20 14:40:44	2024-11-20 14:40:49	percona-mysql-binary-fileset	percona-mysql-binary-job	F	6	✓	○
173	2024-11-14 14:09:35	2024-11-14 14:09:43	bacula-container-fd-etc-fileset	bacula-container-fd-etc-job	F	10794	✓	○

Step 3

To restore files and directories, drag and drop the selection to Restore Selection Area.

Backup Content

Path:

Name	Size	Date
.	0 B	1970-01-01 00:00:00
/	0 B	1970-01-01 00:00:00

Limit: 0 - 500

Version Browser

Here it is possible to choose between all ve
drop selected version to the Restore Selecti

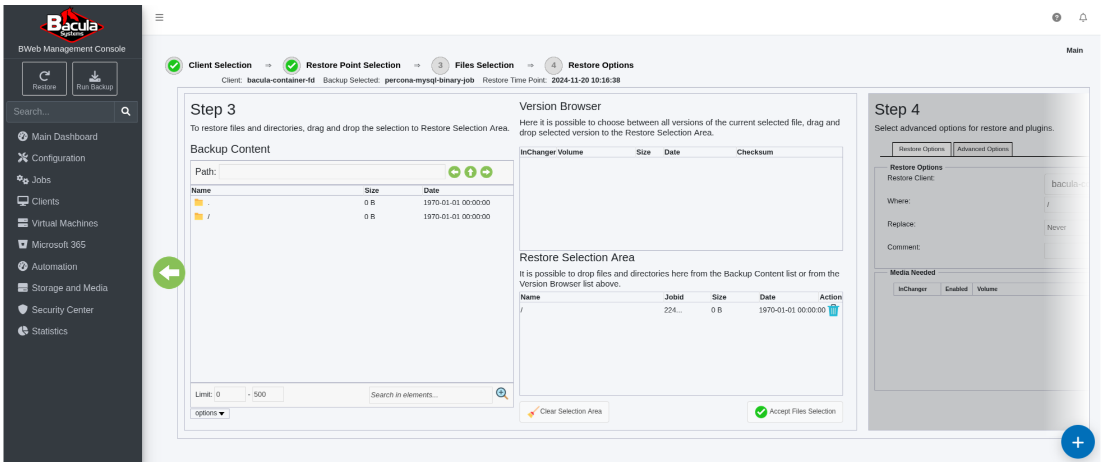
InChanger	Volume	Size	Date

Restore Selection Area

It is possible to drop files and directorie
Version Browser list above.

Name	Jobid

Drop file se



Client: **bacula-container-fd** Backup Selected: **percona-mysql-binary-job** Restore Time Point: **2024-11-20 10:16:38**

Step 3

To restore files and directories, drag and drop the selection to Restore Selection Area.

Backup Content

Path:

Name	Size	Date
.	0 B	1970-01-01 00:00:00
/	0 B	1970-01-01 00:00:00

Limit: 0 - 500

Version Browser

Here it is possible to choose between all versions of the current selected file, drag and drop selected version to the Restore Selection Area.

InChanger	Volume	Size	Date	Checksum

Restore Selection Area

It is possible to drop files and directories here from the Backup Content list or from the Version Browser list above.

Name	Jobid	Size	Date	Action
/	224...	0 B	1970-01-01 00:00:00	

Step 4

Select advanced options for restore and plugins.

Restore Options

Restore Client: bacula-c

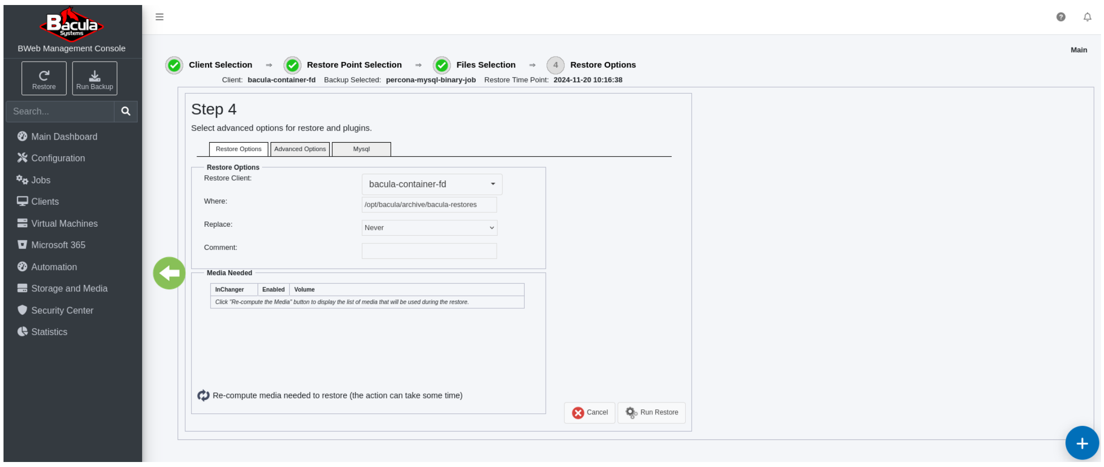
Where: /

Replace: Never

Comment:

Media Needed

InChanger	Enabled	Volume



Client Selection → Restore Point Selection → Files Selection → 4 Restore Options

Client: bacula-container-fd Backup Selected: percona-mysql-binary-job Restore Time Point: 2024-11-20 10:16:38

Step 4

Select advanced options for restore and plugins.

Restore Options | Advanced Options | Mysql

Restore Options

Restore Client: bacula-container-fd

Where: /opt/bacula/archive/bacula-restores

Replace: Never

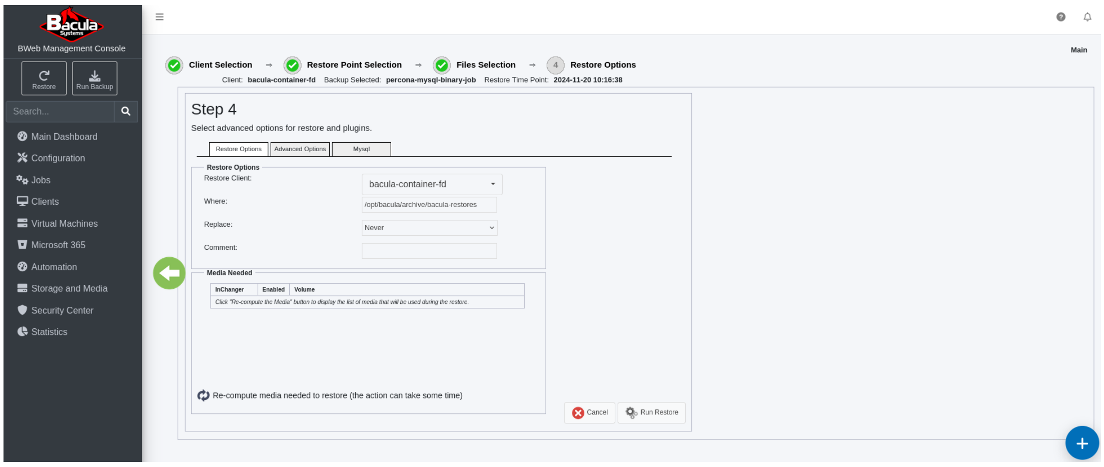
Comment:

Media Needed

InChanger	Enabled	Volume
Click "Re-compute the Media" button to display the list of media that will be used during the restore.		

Re-compute media needed to restore (the action can take some time)

Cancel Run Restore


Client Selection → Restore Point Selection → Files Selection → 4 Restore Options

Client: bacula-container-fd Backup Selected: percona-mysql-binary-job Restore Time Point: 2024-11-20 10:16:38

Step 4

Select advanced options for restore and plugins.

Restore Options | Advanced Options | Mysql

Restore Options

Restore Client: bacula-container-fd

Where: /opt/bacula/archive/bacula-restores

Replace: Never

Comment:

Media Needed

InChanger | Enabled | Volume

Click "Re-compute the Media" button to display the list of media that will be used during the restore.

Re-compute media needed to restore (the action can take some time)

Cancel Run Restore

Procedure using xtrabackup to Restore Data into datadir in Cluster Persistent Volumes

1. `cd /opt/bacula/archive/bacula-restores` to create a directory for the Full, and one for each Incremental job:
   - `mkdir full`
   - `mkdir incr1`
   - `mkdir incr2`
2. `cd @MYSQL/main/` in the directory you have restored the files.
3. decompress the `.xbstream` files in the corresponding `full`, `incr1` and `incr2` directories:

- `cd /opt/bacula/archive/bacula-restores/full && xbstream -x < /opt/bacula/archive/bacula-restores/@MYSQL/main/all-databases.xbstream` (Full job)
- `cd /opt/bacula/archive/bacula-restores/incr1 && xbstream -x < /opt/bacula/archive/bacula-restores/@MYSQL/main/all-databases.34206973.xbstream` (first Incremental job)
- `cd /opt/bacula/archive/bacula-restores/incr1 && xbstream -x < /opt/bacula/archive/bacula-restores/@MYSQL/main/all-databases.34208728.xbstream` (second and latest Incremental job)

4. Prepare Full and the Incremental jobs:
   - Prepare the Full job using the *–apply-log-only* option, in the Full level job directory:

     `xtrabackup --prepare --apply-log-only --target-dir=/opt/bacula/archive/bacula-restores/full`
   - Apply the Incremental backup to the Full backup using the *–apply-log-only* option:

     `xtrabackup --prepare --apply-log-only --incremental-dir=incr1 --target-dir=/opt/bacula/archive/bacula-restores/full`
   - Apply the latest Incremental backup to the Full backup **not using** the *–apply-log-only* option:

     `xtrabackup --prepare --incremental-dir=incr2 --target-dir=/opt/bacula/archive/bacula-restores/full`
5. Copy the backup to the datadir of the server/cluster:

   `xtrabackup --datadir=/var/lib/mysql --copy-back --target-dir=/opt/bacula/archive/bacula-restores/full`

Start Percona XtraDB Cluster

```
# kubectl -n pxc patch --type=merge --patch='{ "spec": { "pause": false } }' pxc pxc-db
perconaxtradbcluster.pxc.percona.com/pxc-db patched
```

Go back to the *Use Cases* page.

Go back to the *MySQL Plugin* page.

Limitations

- To backup multiple MySQL instances with the `binary` method and the xtrabackup tool, you must define multiple Fileset and multiple Job resources.
- The xtrabackup tool doesn't know how to read `.my.cnf` file to get user and password information. Thus you must specify the password in the plugin command line or to use the `extra_file` option in addition to the `mycnf_dir` parameter.
- The Percona `prepare` option may only be used with Full backups. If you attempt to use the `prepare` option with an Incremental or Differential Percona backup, the backup will continue without the `prepare` option.

- The Percona `prepare` option is incompatible with Bacula Encryption, Compression, and ACL options. If you use any of those options with the `prepare` option, the resulting backup will probably be unrestorable.

Note: All Percona backups use the xbstream program to backup the data. The xbstream program automatically uses compression.

- The Percona `prepare` option only works with backups of a single MySQL instance. There may be multiple databases within that instance that are backed up.
- With Percona `prepare=fd` the sized of the all-databases.xbstream as shown in the Bacula Catalog will alwas be reported as `-1`. This is because the stream is created on the fly with no intermediary file.
- Most of the MD5 signatures for a Percona `prepare` will not be valid either because the file never existed on disk, or because the file was modified without recomputing the requested checksum.
- In the current Percona `prepare=sd` implementation the Storage daemon's `Working Directory` is used for placement of the temporary files. Consequently, it should be on a very fast device (RAID or SSD) and must be sufficiently large to handle the maximum database size for as many clients that can run simultaneously.
- The backup of the Percona `prepare=fd` may include a few left over files of the Prepare process that are not really needed for a proper backup.
- The Percona tools are tailor made for each operating system and for each version of MySQL. Therefore you must be very careful about upgrading either MySQL and/or the Percona tools. Please test carefully before trying to put them into production. Apparently with Ubuntu 16.04 the Percona tools that are part of their distribution were not upgraded when they added a newer version of MySQL. Consequently just doing an upgrade of that system can lead to Backup failures. In general, the Percona site has the most current versions you need for each MySQL version and also has a matrix of which versions work together.
- The testing of the Percona `prepare` features was done with `mysql Ver 14.14 Distrib 5.7.22` and `Percona version 2.4.11`.
- As noted above doing a restore of a MySQL database on a system that uses AppArmor (Debian based, e.g. Debian, Ubuntu, ...) can run into AppArmor permissions problems. Thus we strongly recommend that you try doing a Full restore of your MySQL installation in a test environment prior to putting it into production.
- If you have MySQL binary mode backups and you plan to upgrade your MySQL server operating system, please confirm that Percona xtrabackup tools are available for the platform you plan to upgrade the system. MySQL binary mode backups use the Percona xtrabackup tools and they must be available for the new operating system version/platform of your MySQL server.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5.4 Oracle Plugin

Overview

This user's guide presents various techniques and strategies to backup Oracle with **Bacula Enterprise**.

Scope

This paper will present solutions for **Bacula Enterprise** 6.4 and later, which are not applicable to prior versions.

The Oracle Plugin has been tested with AIX 7.1 and 7.3 versions as well as a variety of Linux distributions.

Convention Used In This Guide

- <SID> Anything between < and > should be adapted by the user, for example, <SID> should be replaced by your current ORACLE_SID. If you ORACLE_SID is `TEST` a file written as init<SID>.ora will become initTEST.ora.
- % means that the command should be run with a normal user such as `oracle`
- # means that the command should be run with the root account.
- RMAN> means that the command should by run inside a `rman` session.
- SQL> means that the command should by run inside a `sqlplus` session

Presentation

The Bacula Enterprise Oracle Plugin is designed to simplify the backup and restore procedure of your Oracle Database instance, the backup administrator don't need to learn about internals of Oracle backup techniques or write complex scripts. The Bacula Enterprise Oracle plugin supports both dump and Point In Time Recovery (PITR) with RMAN backup techniques.

In both modes, the plugin will also backup essential information which is part of Oracle DBAs best practices.

When using RMAN mode, the Bacula Enterprise Plugin is able to do incremental and differential backup of the database at a block level.

In 6.4.x and later version of the Bacula Enterprise, the Oracle Plugin implements the RMAN API SBT version 2 that can avoid writing files to local disk first. Note that the Proxy Copy feature is not implemented. The `oracle-sbt` Plugin requires a Director running in 6.4.x version.

This plugin is available on Linux platforms 32/64bit supported by Oracle, and supports Oracle Database versions from 11.x to 21c, including Oracle RAC.

The Oracle plugin is compatible with Copy/Migration jobs. Please read the CopyMigrationJobsReplication for more information.

Using Oracle Plugin

Choosing Between Dump and RMAN

Please note that Oracle, as of version 11g, does no longer support dumping and loading for backup purposes. Existing backup schemes and recovery processes may need to be updated.

Warning: Oracle 11 and later deprecate dump mode!

The following table might help you to choose between backup techniques supported by the Bacula Enterprise Oracle Plugin. Major functionnalities such as beeing able to restore your database to any point in time, or being able to filter objects during backup or restore should guide you. It is also quite common to combine Dump and RMAN PITR techniques for the same cluster.

RMAN also allows you to use advanced techniques where you can send the data through the Bacula Enterprise SBT interface to tape, and, for example, keep it on disk for fast restore and disaster recovery.

	Dump	RMAN	RMAN SBT
Can restore directly a single object (table, schema…)	Yes	No	No
Can restore directly a single file (index, db, tbs…)	No	Yes	Yes
Backup speed	Low	High	High
Restore speed	Low	High	High*
Backup size	Small	Big	Big
Size on local disk during backup	Nothing	Entire Backup	Nothing
Size on local disk during restore	Nothing	Entire Restore	Objects needed
Can restore to any point in time	No	Yes	Yes
Incremen tal/Differential support	No	Yes	Yes
Can restore in parallel	Yes	Yes	Yes
Online backup	Yes	Yes	Yes
Consistent	Yes	Yes	Yes
Can restore to previous major version of Oracle	*No*	No	No
Can restore to newer major version of Oracle	Yes	?	?
Can backup large database (> 50GB)	No	Yes	Yes

** When using RMAN SBT interface, the restore speed depends on various elements such as the type of media (tape or disk), the network speed, the availability of the Storage Daemon device, etc… In Bacula Enterprise 6.4 and later, the Storage Daemon device is able to use the same disk volume for multiple concurrent restores, so it's possible to use concurrent jobs to restore and backup the data.*

Installation

The Oracle plugin is available as a Bacula Enterprise package for all supported Oracle platforms.

```
bacula-enterprise-oracle-6.4.0-1.rh5.i586.rpm
```

You need to install this plugin on the Client where your Oracle database resides. The Plugin assumes that your instances are listed in `/etc/oratab` and the Oracle Unix user "`oracle`" is member of the DBA Unix group (usually "`dba`" or "`oracle`").

The package contains the following files:

```
/opt/bacula/scripts/bs_oracle_restore.pl
/opt/bacula/scripts/install-sbt-libobk.sh
/opt/bacula/plugins/oracle-fd.so
/opt/bacula/plugins/oracle-sbt-fd.so
```

On Debian system, if you plan to use the Oracle RMAN interface, you will need to install the bacula-enterprise-console package.

Plugin Configuration

As with all Bacula plugins, you must to specify the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

In order to send commands to the Oracle database, the Bacula Enterprise Oracle Plugin must share files on disk with Oracle. When using packages provided by Bacula Systems, these files are located on `/opt/bacula/oracle` and the directory permissions should be:

```
% ls -ld /opt/bacula/oracle
drwxrwx--- 13 root dba 4096 Mar 28 14:04 /opt/bacula/oracle
```

where `dba` is the main group of the `oracle` Unix user. This permission is automatically set when installing packages, however, if your Oracle Unix user is not "`oracle`", you may need to manually set permissions on this directory yourself and make sure that your changes are still in effect after each upgrade of the Bacula Enterprise Oracle Plugin package.

Some files that must be preserved between backups are located in:

```
/opt/bacula/working/<SID>/
```

If the Oracle Plugin is unable to find them during the backup, it will display a message and force a Full backup at the RMAN level.

RMAN SBT Configuration

This part of the user's guide will describe how to install and configure properly the Bacula Enterprise RMAN interface with Oracle and RMAN.

When running a backup or a restore from RMAN, RMAN will need to contact the Bacula Enterprise Director to get information about files and volumes, or run backup and restore jobs. This communication involves shared FIFO command files, and the `bconsole` program.

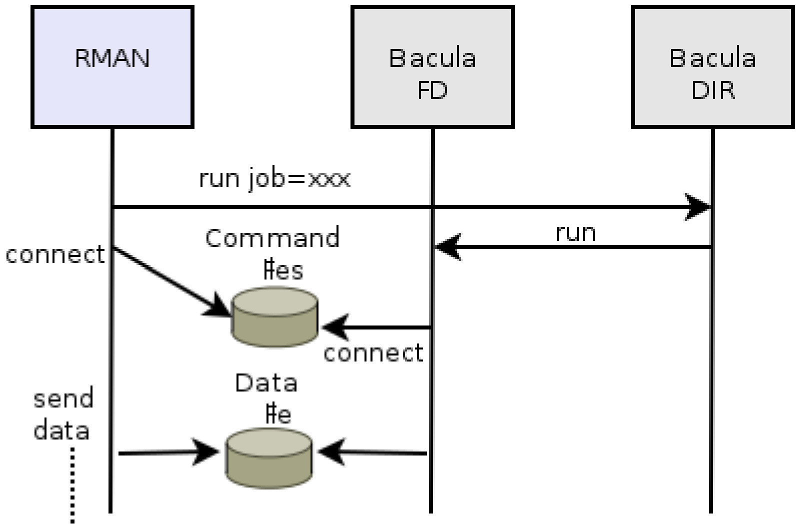


Fig. 114: Interaction Between RMAN and Bacula

When using the `oracle-sbt-fd` plugin, the Director will not be able to start a backup from Bconsole or from a Schedule. Only RMAN will be able to initiate the session and start a Backup. Note that you can still run a regular system backup of your Oracle server, and then, use a RunScript to call RMAN automatically.

Bacula Configuration

When using the RMAN interface, Bacula console `bconsole` should be installed, and the console should be able to connect to your Director and have access to the local Client, the backup Job and other Pool specifications.

The access to the Director via `bconsole`, a restricted console should be properly configured on the Client:

```
# cat /opt/bacula/oracle/bconsole.conf

# Bacula User Agent (or Console) Configuration File
#

Director {
  Name = "oracle11-dir"
  DirPort = 9101
  Address = 192.168.0.46
  Password = "NotUsed"
}

```

(continues on next page)

(continued from previous page)

```
Console {
  Name = "oracle-fd"
  Password = "pass"
}
```

To use a restricted console, you may use the following Console definition:

```
Client {
  Name = oracle-fd
  Maximum Concurrent Jobs = 10
  ...
}
Console {
  Name = oracle-fd
  Password = "pass"

  CommandACL = .bvfs_lsfiles, .bvfs_get_volumes, use, .bvfs_get_jobids, wait
  CommandACL = .bvfs_restore, .bvfs_cleanup, restore, run, gui, .jobs, quit

  # These commands are used only by the install-sbt-libobk.sh test
  # procedure and can be commented out after the installation
  CommandACL = show, status

  ClientACL  = oracle-fd
  JobACL     = SBT-Backup, RestoreJob

  CatalogACL = *all*
  StorageACL = *all*
  FilesetACL = *all*
  PoolACL    = *all*
  WhereACL   = /

  # Available with Bacula 8.8
  DirectoryAcl = *all*
  UserIdAcl = *all*
}

Job {
  Name    = SBT-Backup
  Fileset = SBT-Fileset
  Client  = oracle-fd
  Maximum Concurrent Jobs = 10

  # Adapt the following resources
  # to your settings
  Messages = Standard
  Pool     = Default
  Storage  = File
}
Fileset {
  Name = SBT-Fileset
  Include {
```

(continues on next page)

(continued from previous page)

```
    Options {
       Signature = MD5
    }
    Plugin = oracle-sbt
  }
}
```

The unix "`oracle`" user should be able to execute `bconsole` and read the associated configuration file `bconsole.conf`, which is **not the default configuration**. You can copy the configuration file to the `/opt/bacula/oracle` directory with the following unix commands:

```
cp /opt/bacula/etc/bconsole.conf /opt/bacula/oracle
chown oracle:dba  /opt/bacula/oracle/bconsole.conf
chmod go-rxw /opt/bacula/oracle/bconsole.conf
```

Running Parallel Jobs

In order to run Backup or Restore using multiple channels, you need to ensure that all required resources in Bacula are properly configured using `Maximum Concurrent Jobs` directive to allow concurrent jobs.

- Director: Director (ex: 100)
- Director: Client (ex: 10)
- Director: Job (ex: 10)
- Director: Storage (ex: 10)
- Storage: Storage (ex: 100)
- Storage: Device (ex: 10 or 10 devices grouped in a Virtual Changer)
- Client: FileDaemon (ex: 10)

To allow concurrent Backup and Restore jobs using the same Director Storage resource, the configuration must use a Virtual Changer disk device. See *Disk Backup* whitepaper about this specific configuration.

libobk Installation

Once packages are installed, the `libobk.so` file will be present in `/opt/bacula/lib` directory.

```
# ls -l /opt/bacula/lib/libobk*
-r--r--r-- 1 root root 95654 2013-03-02 01:00 libobk.so

# ln -s /opt/bacula/lib/libobk.so $ORACLE_HOME/lib/
# ls -l $ORACLE_HOME/lib/libobk.so
lrwxrwxrwx 1 root root 34 2013-03-22 01:01 libobk.so -> /opt/bacula/lib/
↪libobk.so
```

You basically need to add a link in $ORACLE_HOME/lib/ to `libobk.so` in `/opt/bacula/lib/`.

It can be done using the `install-sbt-libobk.sh` script available under `/opt/bacula/scripts`

```
# /opt/bacula/scripts/install-sbt-libobk.sh install
```

Once done, we advise you to shutdown all oracle instances on this client. Sometimes Oracle will take a shared library (such as ours) and place it into its shared memory area. Then each time RMAN runs it will use the library it has in shared memory, not the one on disk. So even though this upgrade will replace the file on disk, Oracle may ignore the newly upgraded library.

When the library is installed and configured, it's time to test if the linking has been done properly.

As `oracle` unix user, run RMAN and allocate a tape channel.

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';

using target database control file instead of recovery catalog
allocated channel: ORA_MAINT_SBT_TAPE_1
channel ORA_MAINT_SBT_TAPE_1: SID=42 device type=SBT_TAPE
channel ORA_MAINT_SBT_TAPE_1: Bacula Enterprise Oracle SBT Plugin 1.0.0.2
```

Storage Consideration

Oracle imposes to the Media Manager, Bacula Enterprise, to not multiplex data streams from two concurrent API sessions onto the same sequential device. It means that if you are using tape based storage for your Oracle backup, you must use different tape devices for each concurrent backup jobs. This restriction doesn't apply to disk based storage. This limitation implies specially longer restore time.

Bacula SBT Configuration

The libobk should be configured with the `/opt/bacula/oracle/sbt.conf` file or by using the RMAN SEND command. The keywords presented Table **OptionsSBT** are accepted.

Table 39: SBT libobk Configuration

Option	Comment	Example
client	Bacula Client name	client=oracle-fd
restoreclient	Bacula Client name used for restore	restoreclient=oracle-fd
job	Bacula Backup Job name	job=SBT-Backup
bconsole	Bconsole command with arguments	bconsole=”/tmp/bconsole -n”
restorejob	Bacula Restore Job name. If multiple restore jobs are defined in your configuration and this option is not used, the SBT plugin will choose automatically the first restore Job defined.	restorejob=RestoreFiles
waitjobcompletion	Wait for Job completion at the end of the SBT session. The default is to finish the SBT session as soon as possible. Note that this option can be used only when starting the Backup from RMAN.	waitjobcompletion
update	Type of update (local/catalog). When the filename is present in the local catalog, the Plugin will reply directly to RMAN without contacting Bacula Director. Use update=force to force Bacula Director check.	update=force
jobopt	Extra Job options	jobopt=”spooldata=no”
backupdir	Directory for the local catalog	backupdir= /opt/bacula/oracle
ctrlfile	Base path of control files	ctrlfile=/tmp/oracle
ctrltimeout	Timeout when connecting with Bacula	ctrltimeout=300
retry	Number of retry when connecting to Bacula	retry=30
localdir	Local data file directory where the SBT plugin will check first before calling a Bacula restore.	localdir=/tmp/@ORACLE/sbt
level	Bacula Job level	level=full
catalog	Bacula Catalog name.	catalog=”MyCatalog2”
trace	Path to the trace file.	trace=/tmp/log.txt
debug	Debug level.	debug=50
comment	Job comment.	comment=”Initiated by me”

The minimal configuration file will require the `client`, `job` and `bconsole` options to be set. Note that if the configuration item contains spaces (such as bconsole), you need to use double quotes to enclose it.

```
# cat /opt/bacula/oracle/sbt.conf
client=oracle-fd
job=OracleBackup
```

(continues on next page)

(continued from previous page)

```
bconsole="/opt/bacula/oracle/bconsole -n -c /opt/bacula/oracle/bconsole.conf"
```

It is possible to overwrite these settings using the RMAN SEND command.

```
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;

 SEND 'job=OtherBackup jobopt="spooldata=yes storage=File1"';
 SEND 'retry=10';
}
```

The SEND command is limited to 512 bytes, so this is possible to use multiple SEND commands to set all options. In general, this is a good idea to avoid long path names when using the RMAN interface.

In the following example, we allocate multiple channels and we can send different parameters to each of them:

```
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
 SEND CHANNEL c1 'job=OtherBackup jobopt="spooldata=yes storage=File1"';
 SEND CHANNEL c2 'retry=10';
}
```

Some other RMAN driver providers are using the environment variables to pass parameters to the RMAN driver (`ENV=(waitjobcompletion)`). Bacula Enterprise Oracle RMAN driver uses only the SEND command.

Fileset Configuration

The Oracle RMAN plugin (`oracle-sbt`) is accepting parameters in the Job Fileset described in Table **OptionsPluginSBT**.

Table 40: Oracle SBT Plugin Options

Option	Default	Comment	Example
unix_user	oracle	Unix user to use for Oracle commands	unix_user=rob
ctrlfile	/opt/bacula/oracl	Path to the control file shared between the Plugin and RMAN.	ctrl-file=/tmp/base

Testing sbt.conf Configuration

To test the Bacula Enterprise Oracle SBT Plugin configuration, the following command can be used as `root` user:

```
oracle# /opt/bacula/scripts/install-sbt-libobk.sh test
1000 OK: oracle11-dir Version: 6.4.1 (24 May 2013)
INFO: Connection to the Director OK
INFO: Connection from the Director to the Client OK
INFO: Plugin installed correctly
INFO: Job found on the Director
INFO: Fileset configured on the Director
```

If a connection error is detected, a message will be displayed. It is useless to run any RMAN backup until the connection is properly configured.

Internal Bacula SBT Catalog

The Bacula Enterprise libobk will use a local catalog to store information about all files. These information may be outdated, so you can use the `update=force` in the sbt.conf file or in the SEND command, to force lookup Bacula catalog.

```
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c2 DEVICE TYPE disk;
 SEND 'update=force client=oracle-fd';

 CROSSCHECK  ARCHIVELOG ALL;
 DELETE EXPIRED ARCHIVELOG ALL;
 CROSSCHECK BACKUP;
 DELETE NOPROMPT FORCE OBSOLETE;
}
```

The catalog is stored by default in `/opt/bacula/oracle/bacula-sbt.cat` and can be a part of the regular system backup.

RMAN Backup Retention

When using RMAN SBT plugin of the Bacula Enterprise, the backup retention defined in RMAN should match the Bacula volume or job retention. When RMAN will send commands to delete backup files, Bacula will not try to purge or delete anything.

Backup Examples

The following example will start 3 Bacula backup jobs in parallel and RMAN will send data into them using some kind of round robbin. If RMAN is not able to contact Bacula for one or more channel, RMAN will automatically send the data to the available channel. It means that if your Storage or the Director is busy (limited by the number of devices or the Maximum Concurrent Jobs setting), RMAN will manage the situation automatically.

```
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c3 DEVICE TYPE sbt;
 BACKUP INCREMENTAL LEVEL 0 DATABASE  plus archivelog;
}
```

In this example, RMAN will use 3 Bacula backup jobs to backup 3 datafiles.

```
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c3 DEVICE TYPE sbt;
 BACKUP DATAFILE 1,2,3;
}
```

Stub job to trigger RMAN jobs through Bacula's scheduler

In this example Bacula is used to create a job that triggers the respective RMAN job on the oracle server. As any Bacula job requires a fileset, create an empty fileset:

```
Fileset {
  Name = "None"
```

(continues on next page)

(continued from previous page)

```
  Enable VSS = no
}
```

Create a proper schedule, if it does not exist already. In this example:

- A full backup on Sunday night
- An incremental backup every hour at the minute 30
- A differential backup every night of the week but Sunday

```
Schedule {
   Name = "OracleSchedule"
   run = Level=Full Sun at 21:00
   run = Level=Incremental hourly at 00:30
   run = Level=Differential Mon-Sat at 21:00
}
```

Create the job to perform the backup of database **SID=ORCLDB**:

```
Job {
   Name = "RunRMANBackup"
   Type = "Backup"
   Client = "oracle-fd"
   Fileset = "None"
   JobDefs = "DefaultJob"
   Runscript {
     Command = "su - oracle -c /opt/bacula/oracle/trigger-rman-backup.sh %l␣
↪ORCLDB"
     RunsOnClient = yes
     RunsWhen = After
   }
   Schedule = "OracleSchedule"
 }
```

The script **trigger-rman-backup.sh** would be as below:

```
#!/bin/bash
#
# Copyright (C) 2000-2024 Bacula Systems
# License: BSD 2-Clause; see file LICENSE-FOSS
#
# Bacula Job configuration:
# ClientRunBeforeJob = "sudo -u oracle /etc/bacula/scripts/orabacula.sh %l
↪<SID>"
# OR
# ClientRunBeforeJob = "sudo -u oracle /etc/bacula/scripts/orabacula.sh␣
↪archivelog <SID>"
# FOR ARHIVE LOG ONLY.

[ "$2" = "" ] && echo -e "Not enough parameters.\nUsage: $0 JobLevel <SID>" &&
↪ exit 1

export TMP=/tmp
```

(continues on next page)

(continued from previous page)

```
export TMPDIR=$TMP
export ORACLE_SID=$2
export ORAENV_ASK=NO
. oraenv
export ORAENV_ASK=YES

case "$1" in
    "Full")
      BPAR="DATABASE plus archivelog"
    ;;
    "Differential")
      BPAR="INCREMENTAL LEVEL 0 DATABASE plus archivelog"
    ;;
    "Incremental")
      BPAR="INCREMENTAL LEVEL 1 DATABASE plus archivelog"
    ;;
    "archivelog")
      BPAR="ARCHIVELOG FROM TIME 'SYSDATE-1'"
    ;;
    *)
      echo "Invalid Backup level informed"
      exit 2
    ;;
esac

[ ! "$BPAR" = "" ] && $ORACLE_HOME/bin/rman target / <<EOF
RUN {
ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
ALLOCATE CHANNEL c3 DEVICE TYPE sbt;
BACKUP ${BPAR};
}
QUIT
EOF
```

RMAN Mode Configuration

This part of the user's guide will describe how to configure RMAN to work properly with the non-SBT part of the Bacula Enterprise Oracle Plugin.

The current version of the Bacula Enterprise Oracle Plugin currently supports only databases running with ARCHIVELOG mode enabled.

ARCHIVELOG Oracle Configuration

In order to use the RMAN backup mode, the database must be in ARCHIVELOG mode. To verify how your database is configured you can use the following SQL command.

```
oracle$ sqlplus / as sysdba

SQL> SELECT LOG_MODE FROM SYS.V$DATABASE;

```

(continues on next page)

(continued from previous page)

```
LOG_MODE
-----------
ARCHIVELOG
```

To activate the archive mode of your database, you can use the `ALTER DATABASE ARCHIVELOG` command on a non-open state as `SYSDBA`.

- Stop the database with `SHUTDOWN`
- Backup the database
- Edit your `init<SID>.ora` file to configure the archive log destination
- Start your database without opening it with `STARTUP MOUNT`
- Change the archive mode with `ALTER DATABASE ARCHIVELOG;` and open it with `ALTER DATABASE OPEN;`
- Stop the database with `SHUTDOWN IMMEDIATE`
- Backup the database again, because changing the ARCHIVELOG will update control files and will make old backups unusable.

Bacula Enterprise Oracle Plugin will create RMAN backup set into a sub directory of the archive log destination defined in the init<SID>.ora file.

Optimize Incremental Backup

RMAN's change tracking feature for incremental backups improves incremental backup performance by recording changed blocks in each data file in a change tracking file. If change tracking is enabled, RMAN uses the change tracking file to identify changed blocks for incremental backup, thus avoiding the need to scan every block in the data file.

After enabling change tracking, the first Full backup still has to scan the entire data file, as the change tracking file does not yet reflect the status of the blocks. Any subsequent incremental backups that use this Full backup as parent will take advantage of the change tracking file.

The following SQL command, run as sysdba, permits activating the change tracking feature and use the file "/path/to/file" as destination of the activity log. (Note that the file must be in a valid directory where the Oracle user can write)

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE  '/path/to/file';
SQL> ALTER DATABASE OPEN;
```

RMAN Backup Retention

When using RMAN mode of the Bacula Enterprise Oracle Plugin, each Bacula job will execute RMAN to generate a backup set. We advise you to configure RMAN in order to delete old files after some time period. Although, this can be done just after the end of the backup, we advise you to keep data on disk a bit longer to avoid any gaps in your point-in-time recovery capability. The following command will configure the Oracle retention period of 7 days, which should be sufficient providing you do some sort of backup at least once every 7 days.

```
RMAN>  CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

See the Oracle RMAN manual for more information http://docs.oracle.com/cd/B28359_01/backup.111/b28270/rcmconfb.htm#i1019318

Oracle Plugin Configuration for RMAN

With the RMAN point-in-time recovery (PITR) option, the Bacula Oracle plugin requires Accurate mode information to correctly handle incremental and differential backups, thus you must enable the Accurate option in your Job resource. Note that when combined with the plugin, the Accurate option is used to ensure that all new files are saved by Bacula, but will not mark old files as deleted from previous backups as they most likely will still be useful.

```
Job {
 Name = "Oracle-RMAN"
 Client = laptop1-fd
 Fileset = FS_oracle
 Accurate = yes
 ...
}

Fileset {
 Name = FS_oracle
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "oracle: mode=rman"
 }
}
```

In the RMAN mode, the Oracle plugin also accepts additional options on the Plugin command line listed in the table below:

Table 41: Oracle Plugin Options in RMAN Mode

Option	Comment	Default	Example
mode	Needed to enable RMAN PITR backup	dump	mode=rman
oracle_user	Oracle Unix super user	oracle	oracle_user=oracle10
sid	Oracle SID		sid=XE
ORA-CLE_SID	Oracle SID		ORACLE_SID=XE
ORA-CLE_HOME	Oracle HOME		ORACLE_HOME =/opt/oracle/..
verbose	Display RMAN output in the Job	0	verbose=1
sbt	Use SBT in RMAN script , available since version 6.4.3		sbt
ctrlfile	Base path of control files when using SBT	ctrl-file=/tmp/oracl	

Schedule Consideration for RMAN

If you wish to be able to restore to any point in time between the last Incremental and the next Full or Differential backup you must arrange your backups carefully so as not to leave a gap. One way is to schedule an Incremental backup immediately before your next Full or Differential.

Another way is to configure the RMAN Retention Policy to include previous archivelogs. For example:

```
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
or
RMAN> CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
```

Either of above options, the Full backup will contain previous archivelog files required to restore to a point between the last Incremental and the Full backup. To restore to a point between both jobs, you need to restore files from both groups of backup. Note, if you have implemented the above policy, there is no need to do an Incremental backup immediately before your Full since the Full will find the old logs and save them.

Customize RMAN Scripts

The Bacula Enterprise Oracle Plugin allows you to customize the RMAN backup script by creating special files in `/opt/bacula/etc`:

- `oracle_before_full_backup.rman`
- `oracle_before_incr_backup.rman`
- `oracle_before_diff_backup.rman`

You may want to add special actions to these files such as:

```
% cat /opt/bacula/etc/oracle_before_full_backup.rman
BACKUP ARCHIVELOG FROM TIME 'SYSDATE-2';
```

In this example, the Bacula Enterprise Oracle Plugin will include a backup of all Archivelogs generated for the last two days. It will allow you to cover the gap between the last Incremental backup and the current Full backup as described in *2.5.5*. Note that if you configure the Retention Policy the Full backup will automatically include Archivelogs generated between the two backups that have not been backed up.

If you want to exclude tablespaces from the backup, you can use the following RMAN script.

```
% cat /opt/bacula/etc/oracle_before_full_backup.rman
CONFIGURE EXCLUDE FOR TABLESPACE cwmlite;
CONFIGURE EXCLUDE FOR TABLESPACE example;
```

In this example, the RMAN backup will exclude the two tablespaces “cwmlite” and “example”. Note that this setting is saved in RMAN configuration and will stay across RMAN sessions.

Tip: Use `CONFIGURE EXCLUDE FOR TABLESPACE cwmlite CLEAR;` to remove this exception.

Important, you can’t change the configuration of the disk device or it will reset previous configuration made by Bacula and break the job.

Dump Configuration

Warning: Oracle 11 and later deprecate dump mode!

With the Dump option, Bacula cannot perform Incremental or Differential backups, but the procedure to backup and restore is very simple, and this method is suitable for small or medium databases that don’t need Point-In-Time recovery capabilities. Also, please be aware that Oracle database 11g and newer no longer support dump for backup purposes.

```
Job {
 Name = "Oracle-dump"
 Client = laptop1-fd
 Fileset = FS_oracle_dump
 ...
}

Fileset {
 Name = FS_oracle_dump
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = oracle
 }
}
```

In this example, the plugin will detect and backup all databases on your server. Instances will be detected using information in `/etc/oratab`. You can also specify `ORACLE_HOME` and `ORACLE_SID` in Plugin command line.

```
Fileset {
 Name = FS_oracle
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = "oracle: schema=bacula"
   Plugin = "oracle: schema=master"
 }
}
```

In this example, the plugin will backup databases `bacula` and `master`.

In the Dump mode, the Oracle plugin also accepts additional options on the Plugin command line listed in table *2.4*.

Table 42: Oracle Plugin Options in Dump Mode

Option	Default	Comment	Example
mode	dump	Configure the Plugin backup method	mode=dump
dump_opt	CONSISTENT=Y GRANTS=y	This string will be passed to the exp command	dump_opt=""
unix_user	oracle	Unix user to use for Oracle commands	unix_user=rob
oracle_user	/ as sysdba	Oracle user to use for Oracle commands	oracle_user="scott/tiger"
use_sudo		Use sudo instead to run Oracle commands (when not root)	use_sudo
compress	Y	Use exp compression. Y/N	compress=N
schema		Will backup schema matching this string	schema=PROD*
instance		Will backup instances (SID) matching this string	instance=PROD*
sid		Instance (SID) to backup	sid=PROD
ORACLE_HOME		ORACLE_HOME to use with **sid**	ORACLE_HOME=/ora

```
Fileset {
 Name = FS_oracle_dump
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "oracle: unix_user=rob dump_opt=\"TABLES=temp\""
 }
}
```

In this example, the Oracle plugin will use Unix account "rob" to perform a dump backup of table named "temp". The Oracle Plugin expects the "rob" account to be a member of the dba Unix group in order to directly access Oracle using "/ as sydba".

Testing Database Access

You can use the `estimate` command to verify that the Oracle plugin is correctly configured.

```
* estimate listing job=oracle-test
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
-rw-r--r--  1 oracle dba   1949 2012-06-06 21:55:20  /@ORACLE/XE/users.sql
-rw-r--r--  1 oracle dba   5240 2012-06-06 21:55:22  /@ORACLE/XE/FLOWS/user.
↪sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:22  /@ORACLE/XE/FLOWS/data.
↪dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:22  /@ORACLE/XE/FLOWS
-rw-r--r--  1 oracle dba   1028 2012-06-06 21:55:23  /@ORACLE/XE/HR/user.sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:23  /@ORACLE/XE/HR/data.dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:23  /@ORACLE/XE/HR
-rw-r--r--  1 oracle dba    360 2012-06-06 21:55:23  /@ORACLE/XE/OUTLN/user.
↪sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:23  /@ORACLE/XE/OUTLN/data.
↪dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:23  /@ORACLE/XE/OUTLN
```

(continues on next page)

(continued from previous page)

```
-rw-r--r--  1 oracle dba   2941 2012-06-06 21:55:24  /@ORACLE/XE/SYS/user.sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:24  /@ORACLE/XE/SYS/data.dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:24  /@ORACLE/XE/SYS
-rw-r--r--  1 oracle dba   2857 2012-06-06 21:55:24  /@ORACLE/XE/SYSTEM/user.
↪sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:24  /@ORACLE/XE/SYSTEM/data.
↪dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:24  /@ORACLE/XE/SYSTEM
-rw-r--r--  1 oracle dba    233 2012-06-06 21:55:25  /@ORACLE/XE/TSMSYS/user.
↪sql
-rw-r-----  1 oracle dba     -1 2012-06-06 21:55:25  /@ORACLE/XE/TSMSYS/data.
↪dmp
drwxr-x---  2 oracle dba   4096 2012-06-06 21:55:25  /@ORACLE/XE/TSMSYS
2000 OK estimate files=25 bytes=36,643
```

Notice that all the files backed up by the Bacula Oracle plugin are under a pseudo-directory named /@ORACLE/.

In order to use the sudo wrapper, you need to comment out:

```
Defaults requiretty
```

by putting a pound sign (#) in front of it in the /etc/sudoers file.

Estimate Information

The estimate command will display all discovered information by the Oracle plugin. Note that with the dump mode, Bacula can't compute dump size for databases and will display -1 instead. On RMAN PITR mode, Bacula will not use RMAN to generate backup set, so it will display only system files and the flash recovery area.

Backup Files in RMAN Mode

In RMAN mode, Bacula Enterprise Oracle Plugin will not be cataloged under the pseudo-directory named /@ORACLE/ but will be under their original location. For example:

```
/etc/oratab
/ora10/flash_recovery_area/<Jobname>/<SID>/last_control_file
/ora10/flash_recovery_area/<Jobname>/<SID>/restore_query_file.txt
/ora10/product/10.2.0/server/network/admin/tnsnames.ora
/ora10/product/10.2.0/server/network/admin/listener.ora
/ora10/product/10.2.0/server/dbs/orapw<SID>
```

The Plugin will also generate some extra files in order to help you in case of a disaster situation, such as datafiles.txt or logfiles.txt as shown in Table *2.5*.

Table 43: Files Generated During RMAN Backup

File	Comment
dbid.txt	DBID of the current instance
tnsname.ora	Backup of ORACLE_HOME/network/admin/tnsname.ora
control<SID>.txt	Last control file generated after the backup
init<SID>.ora	Parameter file for the current instance
last_control_file	File that contains the name of the last control file backup
datafiles.txt	List of all data files
logfiles.txt	List of all log files
tempfiles.txt	List of all temp files
restore_query_file.txt	Special file needed for automatic restore
orpw<SID>	Password file for this instance
c-0000-YYYYMMDD-00	Control file
9tncsu2b_1_1	Backup Set

RMAN Backup Level

In Bacula Enterprise terminology, jobs may have the following **Level**

- **Full** A backup that includes all files, it corresponds to the `INCREMENTAL LEVEL 0` RMAN level. This level will allow RMAN to do subsequent incremental backups.
- **Incremental** A backup that includes all files changed since the last Full, Differential, or Incremental backup started. It corresponds to the `INCREMENTAL LEVEL 1` or `INCREMENTAL FROM SCN` RMAN level. Depending on the current state of the RMAN catalog, Bacula will choose one level or the other, depending on the SCN sequence.
- **Differential** A backup that includes all files changed since the last Full save started. It corresponds to the `INCREMENTAL LEVEL 1 CUMULATIVE` RMAN level.

Example of Job Output

The Bacula Enterprise Oracle Plugin will display in the Job output the full result of the RMAN command executed.

```
dir: Start Backup JobId 7, Job=Oracle.2012-06-11_20.00.07_10
dir: Using Device "FileStorage"
sd: Recycled volume "TestVolume001" on device "FileStorage" (/storage), all␣
↪previous data lost.
fd: Calling RMAN for orcl
fd: connected to target database: ORCL (DBID=1229390655)
fd: using target database control file instead of recovery catalog
fd: RMAN> RUN {
fd: 2>  CROSSCHECK  ARCHIVELOG ALL;
fd: 3>  DELETE EXPIRED ARCHIVELOG ALL;
fd: 4>  CROSSCHECK BACKUP;
fd: 5>  DELETE NOPROMPT FORCE OBSOLETE;
fd: 6>  CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK
        TO '/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/%F';
fd: 7>  CONFIGURE CONTROLFILE AUTOBACKUP ON;
fd: 8>  CONFIGURE BACKUP OPTIMIZATION OFF;
fd: 9>  CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2G
        FORMAT '/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/%U';
fd: 10> # @/tmp/oracle_before_full_backup.rman
```

(continues on next page)

(continued from previous page)

```
fd: 11>  BACKUP INCREMENTAL LEVEL 0 DATABASE  PLUS ARCHIVELOG;
fd: 12>  SQL "ALTER DATABASE BACKUP CONTROLFILE TO TRACE
         AS ''/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/
↪controlorcl.txt'' REUSE";
fd: 13>  SQL "CREATE pfile=''/app/oracle/flash_area/Oracle.2012-06-11_20.00.
↪07_10/orcl/initorcl.ora''
         FROM spfile";
fd: 14> }
fd: 15>
fd: allocated channel: ORA_DISK_1
fd: channel ORA_DISK_1: SID=33 device type=DISK
fd: validation succeeded for archived log
fd: Crosschecked 28 objects
fd: released channel: ORA_DISK_1
fd: allocated channel: ORA_DISK_1
fd: channel ORA_DISK_1: SID=33 device type=DISK
fd: specification does not match any archived log in the repository
fd: using channel ORA_DISK_1
fd: crosschecked backup piece: found to be 'AVAILABLE'
fd: backup piece handle=/app/oracle/flash_area/Oracle2.2012-06-08_15.33.06_28/
↪orcl/1pnd0fqi_1_1 RECID=96 STAMP=785..
...
fd: Crosschecked 3 objects
fd: RMAN retention policy will be applied to the command
fd: RMAN retention policy is set to redundancy 1
fd: using channel ORA_DISK_1
fd: no obsolete backups found
fd: CONFIGURE BACKUP OPTIMIZATION OFF;
fd: released channel: ORA_DISK_1
fd: Starting backup at 2012-06-11_11:00:22
fd: current log archived
fd: allocated channel: ORA_DISK_1
fd: channel ORA_DISK_1: SID=33 device type=DISK
fd: channel ORA_DISK_1: starting archived log backup set
fd: channel ORA_DISK_1: specifying archived log(s) in backup set
fd: input archived log thread=1 sequence=556 RECID=6 STAMP=785300484
...
fd: channel ORA_DISK_1: starting piece 1 at 2012-06-11_11:00:23
fd: channel ORA_DISK_1: finished piece 1 at 2012-06-11_11:00:24
fd: piece handle=/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/
↪1rnd8si7_1_1 tag=TAG20120611T110023
fd: channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
fd: Finished backup at 2012-06-11_11:00:24
fd: Starting backup at 2012-06-11_11:00:24
fd: using channel ORA_DISK_1
fd: channel ORA_DISK_1: starting incremental level 0 datafile backup set
fd: channel ORA_DISK_1: specifying datafile(s) in backup set
fd: input datafile file number=00002 name=/app/oracle/oradata/orcl/sysaux01.
↪dbf
...
fd: channel ORA_DISK_1: starting piece 1 at 2012-06-11_11:00:24
fd: channel ORA_DISK_1: finished piece 1 at 2012-06-11_11:01:19
```

(continues on next page)

(continued from previous page)

```
fd: piece handle=/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/
↪1snd8si8_1_1 tag=TAG20120611T110024
fd: channel ORA_DISK_1: backup set complete, elapsed time: 00:00:55
fd: Finished backup at 2012-06-11_11:01:19
fd: Starting backup at 2012-06-11_11:01:19
fd: current log archived
fd: using channel ORA_DISK_1
fd: channel ORA_DISK_1: starting archived log backup set
fd: channel ORA_DISK_1: specifying archived log(s) in backup set
fd: input archived log thread=1 sequence=585 RECID=35 STAMP=785674879
fd: channel ORA_DISK_1: starting piece 1 at 2012-06-11_11:01:19
fd: channel ORA_DISK_1: finished piece 1 at 2012-06-11_11:01:20
fd: piece handle=/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/
↪1tnd8sjv_1_1 tag=TAG20120611T110119
fd: channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
fd: Finished backup at 2012-06-11_11:01:20
fd: Starting Control File and SPFILE Autobackup at 2012-06-11_11:01:20
fd: piece handle=/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/c-
↪1229390655-20120611-00
fd: Finished Control File and SPFILE Autobackup at 2012-06-11_11:01:21
fd: sql statement: ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS
    ''/app/oracle/flash_area/Oracle.2012-06-11_20.00.07_10/orcl/controlorcl.
↪txt'' REUSE
fd: sql statement: CREATE pfile=''/app/oracle/flash_area/Oracle.2012-06-11_20.
↪00.07_10/orcl/initorcl.ora''
    FROM spfile
fd: Recovery Manager complete.
sd: Job write elapsed time = 00:02:29, Transfer rate = 5.577 M Bytes/second
sd: Sending spooled attrs to the Director. Despooling 7,972 bytes ...
dir: Bacula dir 6.0.2 (01May12):
  Build OS:               i686-pc-linux-gnu redhat
  JobId:                  7
  Job:                    Oracle.2012-06-11_20.00.07_10
  Backup Level:           Full (upgraded from Incremental)
  Client:                 "127.0.0.2-fd" 6.0.2 (01May12) i686-pc-linux-gnu,
↪redhat,
  Fileset:                "OracleSQLRMAN" 2012-06-11 14:26:50
  Pool:                   "Default" (From Job resource)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         11-juin-2012 20:00:07
  Start time:             11-juin-2012 20:00:09
  End time:               11-juin-2012 20:02:41
  Elapsed time:           2 mins 32 secs
  Priority:               10
  FD Files Written:       25
  SD Files Written:       25
  FD Bytes Written:       831,041,382 (831.0 MB)
  SD Bytes Written:       831,045,562 (831.0 MB)
  Rate:                   5467.4 KB/s
  Software Compression:   60.4 %
  VSS:                    no
```

(continues on next page)

(continued from previous page)

```
  Encryption:             no
  Accurate:               yes
  Volume name(s):         TestVolume001
  Volume Session Id:      1
  Volume Session Time:    1339432273
  Last Volume Bytes:      831,895,725 (831.8 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Backup Information in Dump Mode

The Oracle plugin will generate the following files for a `MAIN` instance having a single database "test".

```
/@ORACLE/MAIN/TEST/user.sql
/@ORACLE/MAIN/TEST/tables.sql
/@ORACLE/MAIN/TEST/data.dmp

/@ORACLE/MAIN/users.sql
/@ORACLE/MAIN/tnsnames.ora
/@ORACLE/MAIN/listener.ora
/@ORACLE/MAIN/orapwMAIN
/@ORACLE/MAIN/datafiles.txt
/@ORACLE/MAIN/logfiles.txt
/@ORACLE/MAIN/init.ora

/@ORACLE/oratab
```

Table 44: Files Generated During Dump Backup

File	Context	Comment
users.sql	global	List of all users, their password and specific options
datafiles.txt	global	List of all data files, for information only
tempfiles.txt	global	List of all temp files, for information only
logfiles.txt	global	List of all log files, for information only
tnsnames.ora	global	Content of the tnsnames.ora
listener.ora	global	Content of the listener configuration file
orapw<SID>	global	Password file of the current instance
init.ora	global	Oracle configuration file
user.sql	schema	Schema definition with all information about GRANT, PASSWORD, etc.
data.dmp	schema	Database data in exp format, contains everything needed to restore
tables.sql	schema	Tables and indexes definition in SQL format

Oracle RAC Configuration

Since the RAC determines which node will provide the data, the plugin must be installed on each node. It also needs to be configured with the same password in its bacula-fd.conf. On the director at least one client needs to be added pointing to the cluster ip address and again, the same password set in the bacula-fd.conf of every node. The parameter *update=force* will be used to avoid the local catalog becoming outdated.

Restore Scenarios

Before starting a restore or recovery procedure, we strongly advise you to run a backup of you database.

Restoring using RMAN SBT Interface

Like when doing a Backup job, to restore objects with RMAN, the connection between RMAN and Bacula should be functional. If resouces are available, everything will be managed by RMAN automatically.

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;

RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt;
 ALLOCATE CHANNEL c2 DEVICE TYPE sbt;
 SET UNTIL TIME "to_date('2013-05-31_10:20:00','YYYY-MM-DD_HH24:MI:SS')";
 RESTORE DATABASE;
 RECOVER DATABASE;
}
ALTER DATABASE OPEN RESETLOGS;
```

In this example, RMAN will restore the database at a certain point in time defined by the `UNTIL` command. More information can be found on Oracle RMAN documentation. http://docs.oracle.com/cd/B28359_01/server.111/b28294/rman.htm#i1024051

If you restore RMAN files into a local directory and the `localdir` option is defined in `sbt.conf`, the RMAN plugin will look the `localdir` before starting a Bacula restore session. If the requested file is present, the RMAN Plugin will use it directly.

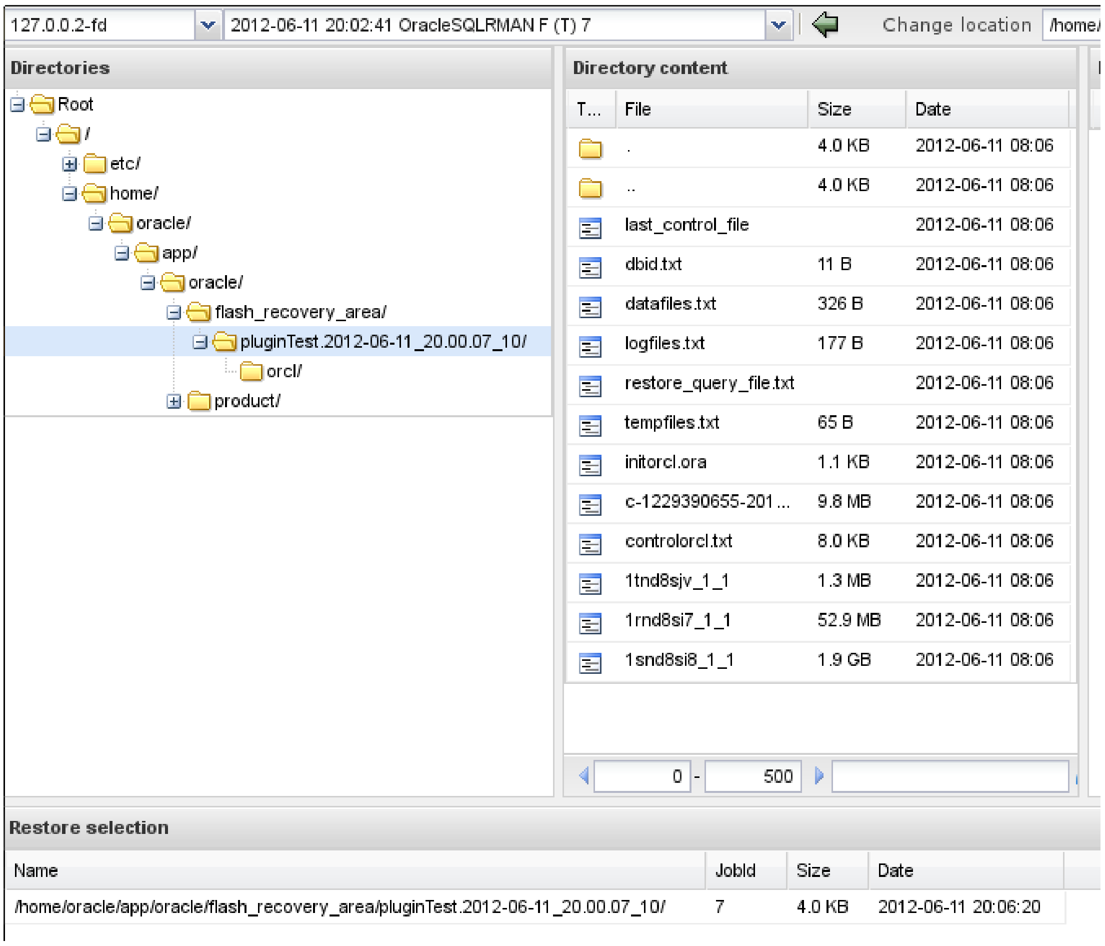
Restoring using RMAN with bs_oracle_restore.pl

Once you have restored the contents of the RMAN backup to your system (Fig 2.2) with the bconsole restore command or with BAT/BWeb, the Bacula Enterprise Oracle Plugin allows you to automate some RMAN operations through a wrapper called `bs_oracle_restore.pl`. This script is menu driven and allows you to:

- Restore the original database to a certain Point-In-Time
- Clone your database whether or not it is still available (currently in beta testing)

Once you have restored the contents of the backup to a given point in time, you should run the `bs_oracle_restore.pl` script with `restore_query_file.txt` file as argument.

Note that you only need to restore files that are not on your system, RMAN will use files that are still in the flash recovery area to perform the restore.



T...	File	Size	Date
	.	4.0 KB	2012-06-11 08:06
	..	4.0 KB	2012-06-11 08:06
	last_control_file		2012-06-11 08:06
	dbid.txt	11 B	2012-06-11 08:06
	datafiles.txt	326 B	2012-06-11 08:06
	logfiles.txt	177 B	2012-06-11 08:06
	restore_query_file.txt		2012-06-11 08:06
	tempfiles.txt	65 B	2012-06-11 08:06
	initorcl.ora	1.1 KB	2012-06-11 08:06
	c-1229390655-201 ...	9.8 MB	2012-06-11 08:06
	controlorcl.txt	8.0 KB	2012-06-11 08:06
	1tnd8sjv_1_1	1.3 MB	2012-06-11 08:06
	1rnd8si7_1_1	52.9 MB	2012-06-11 08:06
	1snd8si8_1_1	1.9 GB	2012-06-11 08:06

0 - 500

Name	JobId	Size	Date
/home/oracle/app/oracle/flash_recovery_area/pluginTest.2012-06-11_20.00.07_10/	7	4.0 KB	2012-06-11 20:06:20

Fig. 115: RMAN file selection when restoring using BWeb

In the next example, you will find the file named restore_query_file.txt in the directory where you restored the files with Bacula. If your backup was in /u01/flash/Test.2012-06-06_12-00-00, and you restored it using where=/tmp, the restore_query_file.txt should be in /tmp/u01/flash/Test.2012-06-06_12-00-00/restore_query_file.txt

```
# /opt/bacula/scripts/bs_oracle_restore.pl /path/to/restore_query_file.txt
Bacula Enterprise Oracle Restore Tool 0.9

Do you want to:
 1- restore the original database
```

The Bacula Enterprise Oracle restore script can be called with the -testing option to have access to restore procedures that are currently in testing phase by Bacula Systems.

```
# /opt/bacula/scripts/bs_oracle_restore.pl --testing /path/to/restore_query_
↪file.txt
Bacula Enterprise Oracle Restore Tool 0.9

Do you want to:
  1- restore the original database

The following restore modes are available but still being beta tested
  2- restore the database into a clone
  3- restore the database to a different location
```

Performing Database Point-In-Time Recovery

RMAN can perform recovery of the whole database to a specified past time, SCN, or log sequence number. This type of recovery is sometimes called incomplete recovery because it does not completely use all of the available redo information.

The restore wrapper bs_oracle_restore.pl will detect from files restored with Bacula Enterprise parameters that you can use during the restore.

```
Do you want to:
        1- restore to a certain point-in-time
        2- restore to a certain scn

Choose restore mode (1-2): 1

Getting the range of recoverable backups
Please input the time to which you want to restore
between 2012-06-05_15:17:16 and 2012-06-05_15:36:09

(YYYY-MM-DD_HH24:MI:SS): 2012-06-05_15:35:00

INFO: Mounting database in mount state
The database is in open state,
do you really want to shutdown the database now (y/N): y
INFO: Call RMAN to restore the database
Opening database

```

(continues on next page)

(continued from previous page)

```
BE CAREFUL, we are about to open the database in RESETLOGS mode.

Do you want to continue ? (no will exit) (y/N): y
Opening database resetlogs
```

The `bs_oracle_restore.pl` will scan the backup directory, then detect the right Incarnation in order to restore files as expected.

At the end of the restore process, the database should be in "OPEN" state. The `bs_oracle_restore.pl` will perform all the step necessary to recover the database. If you are familiar with RMAN, you may want to do these steps manually.

Using bs_oracle_restore.pl Script Without Restoring First

In some case, if RMAN backup sets are still present on disk, you may want to skip the Bacula restore and run directly the `bs_oracle_restore.pl` script. For that, just use the `-D` option and point to the flash back recovery area where are located files generated during the last backup.

```
# /opt/bacula/scripts/bs_oracle_restore.pl -D /path/to/flash/job/SID
```

Restoring Directly with RMAN

Once you restored your files with Bacula, you need to scan the backup directory to include files in the RMAN catalog.

```
RMAN>  CATALOG START WITH '/path/to/restore' NOPROMPT;
```

Then, you should have all backups registered and you can list them with:

```
RMAN>  LIST BACKUP SUMMARY;

List of Backups
===============
Key    TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------ -- -- - ----------- --------------- ------- ------- ---------- ---
458    B  0  A DISK        12-JUN-12       1       1       YES        ␣
↪TAG20120612
459    B  A  A DISK        13-JUN-12       1       1       YES        ␣
↪TAG20120613
461    B  A  A DISK        14-JUN-12       1       1       YES        ␣
↪TAG20120614
462    B  1  A DISK        15-JUN-12       1       1       YES        ␣
↪TAG20120615

...
```

Note that the `LIST BACKUP` can display information about specific objects such as:

- Archivelogs
- Datafiles
- Controlfile
- ...

To start the restore process, the database should not be in open state.

```
RMAN> shutdown immediate;
RMAN> startup mount;
```

Then, you can set a `UNTIL` clause and start your recovery

```
RMAN> RUN {
2> RESTORE DATAFILE 1;
3> RECOVER DATAFILE 1;
3> }

Starting restore at 15-JUN-12
using channel ORA_DISK_1
...
```

RMAN is a very powerful tool with many options, for more information, see the RMAN documentation available on http://docs.oracle.com

Sometime, RMAN may ask to restore files and finaly find out that some of these files are not required to perform the restore. In this case, the communication link with Bacula is stopped by RMAN without specific notification. In this situation, the RMAN job output is correct, but the Bacula restore job report has an error such as:

```
2017-03-20 13:28:12 oracle-fd JobId 1: Error: restore.c:1388 Write error
   at byte=0 block=0 write_len=65536 lerror=0
   on /@ORACLE/sbt/c-3461191666-20170321-02: ERR=Success
```

If the RMAN job report is OK, then the Bacula error can be ignored.

Restoring using Dump

Restoring Users

To restore users/schema to your Oracle instance, you just select the `users.sql` file located in `/@ORACLE/<SID>/users.sql`

Then, using `where=/` or `where=` the plugin will load this SQL file to your database. If some roles already exist, errors will be printed in the Job log. It is also possible to restore the `users.sql` file to a local directory, edit the file and load it with `sqlplus` to restore any selected part of the file.

```
% sqlplus / as sysdba @users.sql
```

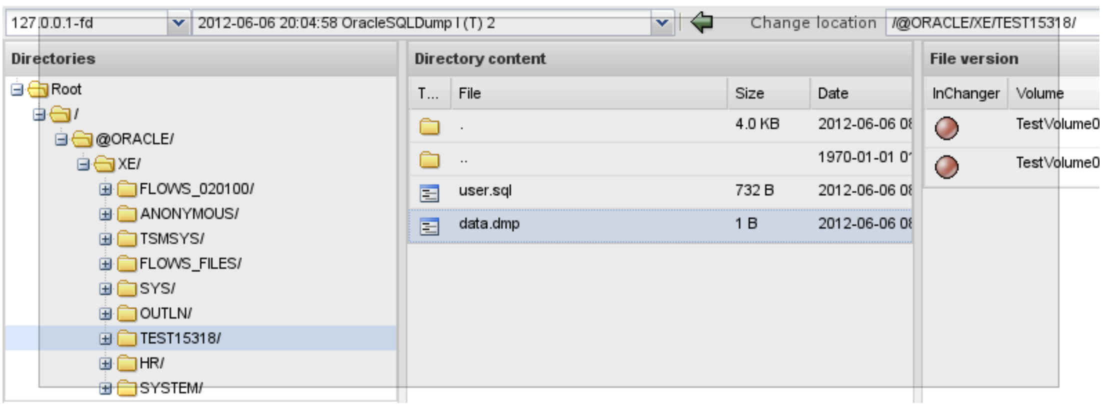
Restoring a Single Database

To restore a single schema with the Bacula Enterprise Oracle plugin, you must select the schema directory during the restore command, the selection should contain the data file (`data.dmp`) and the schema creation script (`user.sql`).

When the database directory is selected, you can use the `where` parameter to restore the schema to a new schema, with a different name. In order to create a new schema name, you must set `where` to a single word that contains only `A..Z`, `0-9`, and `_`, Bacula will then create the specified schema and restore data into it.

```
* restore where=BACULAOLD
```

We advise you to always use schema names in capital letters, Bacula Enterprise Oracle Plugin will recreate the new schema using exactly the same name that you provided in the `where=` parameter. If you mix



127.0.0.1-fd | 2012-06-06 20:04:58 OracleSQLDump1 (T) 2 | Change location | /@ORACLE/XE/TEST15318/

Directories

```
Root
  /
    @ORACLE/
      XE/
        FLOWS_020100/
        ANONYMOUS/
        TSMSYS/
        FLOWS_FILES/
        SYS/
        OUTLN/
        TEST15318/
        HR/
        SYSTEM/
```

Directory content

T...	File	Size	Date
	.	4.0 KB	2012-06-06 0
	..		1970-01-01 0
	user.sql	732 B	2012-06-06 0
	data.dmp	1 B	2012-06-06 0

File version

InChanger	Volume
●	TestVolume0
●	TestVolume0

Fig. 116: Database content with dumps with BWeb

upper and lowercase characters in the name, it can lead to a situation where you will need to enclose the schema name with quotes to access it.

Once restored, you may need to reset the password of the schema that you just created using the same parameters as the original schema. To do so, use:

```
SQL> ALTER USER BACULAOLD IDENTIFIED BY APASSWORD;
```

If you set the `replace` parameter to `never`, Bacula will check the schema list, and will abort the Job if the schema currently restored already exists.

Using `replace=always` is not recommended, because it can overwrite existing files.

If the `where` parameter is a directory (containing /), Bacula will restore all files into this directory. Doing so, you will be able to use :term"*IMP* directly and restore only triggers, tables, indexes, etc…

Restoring a Single Table

To restore a single object such as a table from your dump, you must first restore the dump file to a local directory, then use the :term"*IMP* tool to import the needed object. See the Oracle :term"*IMP* documentation for more information.

Restoring Dump Files to a Directory

To restore SQL dumps to a directory, you can set the `where` parameter to any valid directory.

```
* restore where=/tmp
```

The Bacula restore process will create the following directories when restoring the schema SYS in the Oracle SID XE, and will restore selected files into it.

```
/tmp/@ORACLE/MAIN/XE
```

Restoring the Entire Database

To restore the all databases and the database configuration, just all files located in `/@ORACLE/<service>`, use `replace=always` and `where=/`.

Note: Oracle documentation:

- http://docs.oracle.com
- http://www.orafaq.com

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5.5 Sybase Plugin

Overview

This user guide presents various techniques and strategies to backup Sybase Adaptive Server Enterprise with Bacula Enterprise.

Scope

This paper will present solutions for Bacula Enterprise version 10 and later, which are not applicable to prior versions.

Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. This white paper presents various strategies to backup Sybase Adaptive Server Enterprise using the Sybase Plugin with Bacula Enterprise version 10.

The Bacula Enterprise Sybase Plugin is designed to simplify the backup and restore operations of your Sybase Adaptive Server Enterprise. The backup administrator does not need to learn about internals of Sybase backup techniques or write complex scripts. The Bacula Enterprise Sybase Plugin supports Point In Time Recovery (PITR) with Sybase Backup Server Archive API backup and restore techniques.

The Bacula Enterprise Plugin is able to do incremental and differential backup of the database at a block level. This plugin is available on 32 and 64-bit Linux platforms supported by Sybase, and supports Sybase ASE 12.5, 15.5, 15.7 and 16.0.

Features Summary

- Sybase database online backup and restore using *Sybase Backup Server Archive API*.
- Support for **Full Database** backup and restore, **Cumulative Database** (incremental) backup and restore, **Database Transactions** backup and restore.
- Database backup levels are smoothly mapped to Bacula Enterprise backups levels.
- All backup or restore operations are managed by the database administrator.
- Ability to restore any database backup to an alternate location.
- Direct support of Point In Time Recovery (PITR) with database transaction restoration.

Conventions Used In This Guide

- `<SYBSERVER>` Anything between < and > should be adapted by the user, for example, `<SYBSERVER>` should be replaced by your current Sybase service name. If your Sybase service name is `SYBASE16` a variable `SYS=<SYBSERVER>` will become `SYS=SYBASE16`.
- The `%` prompt means that the command should be run with a normal user such as `sybase`.
- A `#` prompt implies that the command should be run with the `root` account.
- `N>` means that the command should be run inside an `isql` utility session, where 'N' is an integer sequential number, i. e. 1, 2, 3, etc.
- `*` indicates that the command should be run as a `bconsole` command.

Using the Sybase Plugin

Installation

The Sybase plugin is available as a Bacula Enterprise package for all supported Sybase platforms.

```
bacula-enterprise-sybase-plugin.<version>.[rpm,deb]
```

This plugin has to be installed on the Client where the Sybase database resides. The package will install the following files:

```
/opt/bacula/scripts/bacula-sybase.sh
/opt/bacula/scripts/install-sybase.sh
/opt/bacula/scripts/sybase-backup.sh
/opt/bacula/lib/libsybacula.so
```

Note: On a Debian system it is necessary to install the `bacula-enterprise-console` package.

Plugin Configuration

As with all Bacula plugins, the **Plugin Directory** directive in the **FileDaemon** (or) resource of the bacula-fd.conf file has to be set:

```
FileDaemon {
   Name = sybase-fd
   (...)
   Plugin Directory = /opt/bacula/plugins
}
```

In order to send commands to the Sybase database, the Bacula Enterprise Sybase Plugin must share files on disk with Sybase. When using packages provided by Bacula Systems, these files are located in /opt/bacula/sybase and the directory permissions should allow to read and write files for the Sybase user, i.e.:

```
# ls -ld /opt/bacula/sybase/
drwxr-xr-x. 2 sybase sybase 85 11-22 09:40 /opt/bacula/sybase/
```

where sybase is the main user and group of the Sybase Database Unix user. These permissions are automatically set when installing packages, but, if the actual Sybase Unix user is not "sybase", it will be necessary to manually change permissions on this directory and make sure that the changes are still in effect after an upgrade of the Bacula Enterprise Sybase Plugin package.

Sybase SBT Configuration

This section describes how to properly install and configure the Bacula Enterprise SBT interface with the Sybase Backup Server.

When running a backup or a restore from Sybase, Sybase Backup Server will need to contact the Bacula Enterprise Director to get information about files and volumes, or to run backup and restore jobs. This communication involves shared FIFO command files, and the bconsole program.

When using the sybase-sbt-fd plugin, the Director will not be able to start a backup through bconsole or directly with a scheduled job. Only the Sybase Backup Server will be able to initiate the session and start a backup. Note that it is still possible run a regular system backup of your Sybase server, and, along with it, use a **RunScript** to call Sybase Backup Server automatically.

Bacula Configuration

When using the SBT interface, the Bacula console bconsole needs to be installed, and the console should be able to connect to your Director and have access to the local **Client**, the backup **Jobs** used for Sybase database backups, and any required **Pools**. To access the Director via bconsole, a restricted console will need to be confiured on the Director, and a bconsole configuration file needs to be configured on the client:

```
# cat /opt/bacula/etc/bconsole-sybase.conf
# Bacula User Agent (or Console) Configuration File
Director {
   Name = "sybase-dir"
   DirPort = 9101
   Address = 192.168.0.46
```

(continues on next page)

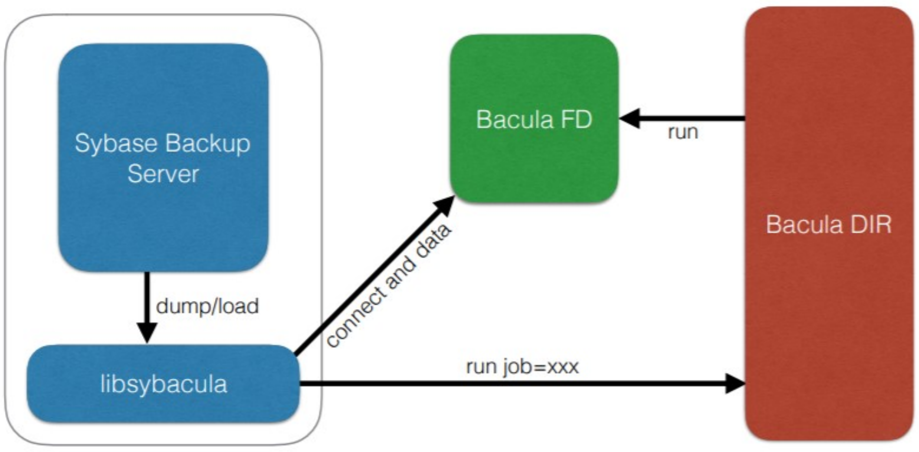


Fig. 117: Interaction Between Sybase Backup Server and Bacula

(continued from previous page)

```
   Password = "NotUsed"
}
Console {
   Name = "sybase-fd"
   Password = "pass"
}
```

To use a restricted console, the following or a similar **Console** definition and other required resources – **Client**, **Job**, **Pool** and **Fileset** must be explicitly allowed per ACL in the Director's configuration:

```
Client {
   Name = sybase-fd
   Maximum Concurrent Jobs = 10
}

Console {
   Name = sybase-fd
   Password = "pass"
   CommandACL = .bvfs_lsfiles, .bvfs_get_volumes, use, .bvfs_get_jobids, wait
   CommandACL = .bvfs_restore, .bvfs_cleanup, restore, run, gui, .jobs, quit
   # These commands are used only by the install-sybase.sh test
   # procedure and can be commented out after the installation
   CommandACL = show, status

   ClientACL = sybase-fd
   JobACL = SBT-Backup, RestoreJob
   CatalogACL = *all*
```

(continues on next page)

(continued from previous page)

```
   StorageACL = *all*
   FilesetACL = *all*
   PoolACL = *all*
   WhereACL =/
   DirectoryAcl = *all*
   UserIdAcl = *all*
}

Job {
   Name = SBT-Backup
   Fileset = SBT-Fileset
   Client = sybase-fd
   Maximum Concurrent Jobs = 10
   # Adapt the following resources
   # to your settings
   Messages = Standard
   Pool = Default
   Storage = File
}

Fileset {
   Name = SBT-Fileset
   Include {
      Options {
         Signature = MD5
      }
      Plugin = "sybase-sbt:"
   }
}
```

The unix "sybase" user or, more precisely, the user account used by the Sybase installation should be able to execute `bconsole` and read the associated configuration file `bconsole-sybase.conf`, which is **not the default configuration**. It is possible to copy the configuration file to `/opt/bacula/etc/bconsole-sybase.conf` with the following unix commands:

```
# cp /opt/bacula/etc/bconsole.conf /opt/bacula/etc/bconsole-sybase.conf
# chown sybase:sybase /opt/bacula/etc/bconsole-sybase.conf
# chmod go-rxw /opt/bacula/etc/bconsole-sybase.conf
```

Running Parallel Jobs

In order to run backups or restores using multiple stripes (check *Stripes*), the required resources in Bacula need to be properly configured using the `Maximum Concurrent Jobs` directive to allow concurrent jobs:

- Director: **Director** (ex: 100)
- Director: **Client** (ex: 10)
- Director: **Job** (ex: 10)
- Director: **Storage** (ex: 10)
- Storage: **Storage** (ex: 100)

- Storage: **Device** (ex: 10, or 10 devices grouped in a Virtual Changer)
- Client: **FileDaemon** (ex: 10)

To allow concurrent backup and restore jobs using the same Director Storage resource, the configuration should use a Virtual Changer disk device. See the *Disk Backup* whitepaper about this specific configuration.

Backup Server Module Installation – `libsybacula`

Once packages are installed, the Sybase Backup Server module `libsybacula.so` file will be present in the `/opt/bacula/lib` directory. A symbolic link to it, in `$SYBASE/$SYBASE_ASE/lib/` is the most convenient way to ensure it is available to the Sybase `backupserver` program. This can be done using the `install-sybase.sh` script available under `/opt/bacula/scripts`:

```
# /opt/bacula/scripts/install-sybase.sh install
Enter the SYBASE base install [/opt/sybase]:
Enter the SYBASE_ASE location at <SYBASE>/[ASE-16_0]:
Enter the unix Sybase account name [sybase]:
Enter the SYBASE service name [SYBASE16]:
INFO: write sybase.env file
INFO: linking sybacula module
INFO: Installation SBT interface OK.
```

or this may be executed manually:

```
# ln -s /opt/bacula/lib/libsybacula.so $SYBASE/$SYBASE_ASE/lib/
# ls -l $SYBASE/$SYBASE_ASE/lib/libsybacula.so
lrwxrwxrwx. 1 root root 30 11-21 09:05 libsybacula.so -> /opt/bacula/lib/
↪libsybacula.so
```

As the library is loaded by the Sybase Backup Server on demand, there is no need to restart any Sybase component after successful installation.

During a manual installation, it is helpfu to insert the proper installation values into the `/opt/bacula/sybase/sybase.env` file which is used by other plugin tools, i. e. `sybase-backup.sh`. This step is optional. The file is automatically generated by the `install-sybase.sh` script.

```
# cat /opt/bacula/sybase/sybase.env
SYBASE=/opt/sybase
SYBASE_ASE=ASE-16_0
SYBASE_OCS=OCS-16_0
SYBASE_USER=sybase
SYBSERVER=SYBASE16
SAPASSWORD=secret
```

Here, `SYBSERVER` is a Sybase server name and `SAPASSWORD` a password for the `sa` user, the default database administrator user name.

Storage Consideration

Sybase imposes to the Media Manager, Bacula Enterprise, to not multiplex data streams from two concurrent stripes of the same backup job onto the same sequential device. This means that, when tape-based storage is used for Sybase backups, different tape devices have to be used for each concurrent backup job. This restriction does not apply to disk based storage. This limitation implies longer restore times.

Bacula SBT Configuration

The `libsybacula.so` can be configured in the `/opt/bacula/sybase/sbt.conf` or files, or using the proper arguments to Sybase `dump` or `load` commands. Supported parameters are:

client
: is a Bacula Client name, i.e. `client=sybase-fd`.

restoreclient
: is a Bacula Client name used for restore, such as `restoreclient=sybase-fd`.

job
: is a Bacula Backup Job name used for backup, i. e. `job=SBT-Backup`.

bconsole
: is a `bconsole` command with arguments used to communicate with Bacula Director, i.e. `bconsole="/tmp/bconsole -n"`. The default is `bconsole -u 60 -n -c /opt/bacula/etc/bconsole-sybase.conf`.

restorejob
: is a Bacula Restore Job name. If multiple restore jobs are defined in Bacula's configuration and this option is not used, the Sybase SBT Plugin will automatically choose the first restore Job defined. For example: `restorejob=RestoreFiles`

  waitjobcompletion sets the Sybase plugin to wait for Job completion at the end of the backup or restore session. The default is to finish the session as soon as possible. Example: `waitjobcompletion`.

jobopt
: sets additional Job options the same way they can be passed in a `bconsole run` command, for example `jobopt="spooldata=no"`.

ctrlfile
: is the base path to the control files used to communicate between the Sybase Backup Server and the Plugin, i.e. `ctrlfile=/tmp/sybase`. The default is `/opt/bacula/sybase/sbt`.

ctrltimeout
: is the timeout to apply when connecting with the Bacula Director, i.e. `ctrltimeout=300`.

retry
: determines the number of retries when connecting to Bacula, for example `retry=30`. Note that, in a reasonably configured environment, this option should not be necessary or may be set to a relatively low value.

catalog
: is a Bacula Catalog name, such as `catalog="MyCatalog 2"`. With a Bacula instance which uses one catalog only – which should be the case for most production instances – this option will not be needed.

trace
indicates the path to the trace file. Check **tracing** for more information. Example: `trace=/tmp/log.txt`.

debug
is the debug level to apply when tracing. Again, see **tracing** for more information. Example: `debug=50`.

The minimal configuration file will require the **client** and **job** options to be set. Note that if a configuration item contains spaces (such as the **bconsole** example above), it needs to be enclosed in double quotes.

Note: The Sybase Backup Server Archive API limits the total size of archive device options (`libsybacula` parameters above) to 127 characters. If the needed parameter string exceeds this limit, a configuration file `sbt.conf` must be used.

```
# cat /opt/bacula/sybase/sbt.conf
client=sybase-fd
job=SBT-Backup
```

Fileset Configuration

The Sybase SBT plugin (`sybase-sbt`) accepts the following parameters in the Jobs **Fileset** configuration:

ctrlfile
is the path to the control files used to communicate between Sybase Backup Server and the Plugin, i.e. `ctrlfile=/tmp/sybase`. The default is `/opt/bacula/sybase/sbt`.

Testing `sbt.conf` Configuration

To test the Bacula Enterprise Sybase SBT Plugin configuration, the following command can be used as the `root` user:

```
# /opt/bacula/scripts/install-sybase.sh test
INFO: Testing ...
INFO: Connection to the Director OK
INFO: Connection from the Director to the Client OK
INFO: Plugin installed correctly
INFO: Job found on the Director
INFO: Fileset configured on the Director
INFO: RestoreJob found on the Director
INFO: Fileset configured for the Restore job on the Director
INFO: Testing OK.
```

Backup

The **Bacula Enterprise** Sybase SBT Plugin supports different backup levels to backup Sybase database or transaction dumps. These levels are mapped as follows:

- Sybase Full Database dump – Bacula Full level
- Sybase Incremental (Cumulative) Database dump – Bacula Differential level
- Sybase Transaction dump – Bacula Incremental level

Full Database Backup

This is the most basic and simple database backup. It will generate a backup copy of the entire database, including the transaction logs.

To perform a Full Database backup execute a command like the following:

```
1> dump database pubs2 to 'sybacula::'
2> go
```

For more information about the `dump` command and additional parameters check the Sybase documentation: *Reference Manual: Commands 16.0*, *Commands*.

It is advisable to use a `sybase-backup.sh` script to execute a full database backup manually or through a Bacula backup job using the **Run Script** Job sub-resource:

```
# /opt/bacula/scripts/sybase-backup.sh pubs2 Full
```

This would save a single dump file like: `D.pubs2.2018-11-21.14:22:46.000` where: '`D`' indicates a full dump, '`pubs2` is the database name, '`2018-11-21.14:22:46`' is the backup time stamp and '`000`' is a stripe number.

Incremental Database Backup

The standard full database backup copies entire databases to the backup system (MMS in Sybase terminology) which for large databases can take time and consumes a lot of other resources (archive storage space, cpu, network, etc.). For this reason, Full Database Backups are executed less often, but transaction backups more frequently. In this case, a full recovery to the most current state requires a number of transaction dump loads which consumes resources as well and increase the potentially critical restore time.

To optimize both backup and restore operations it is possible to use Database (Cumulative) Incremental backups which create a copy of all the pages in the database that have been modified since the last full database dump. This backup allows for:

- Reduced recovery time – Recovery time is minimized when compared to a strategy that uses transaction log dumps.
- Reduced backup size, especially for databases that contain big read-only tables.
- Improved backup performance, especially on database loads that have substantial updates on only a subset of the database pages.
- Incremental backups of low-durability databases.

Incremental database backups are a comparatively new feature, available as of Sybase ASE 15.7 SP100. For more information, please see the Sybase documentation: *New Features Guide Adaptive Server Enterprise 15.7 SP100*, section *Enhancements to Backup and Restore*.

To perform a (cumulative) Incremental backup, execute a command such as:

```
1> dump database pubs2 cumulative to 'sybacula::'
2> go
```

More information about the `dump` command and its parameters can be found in the Sybase documentation: *Reference Manual: Commands 16.0*, *Commands*.

Of course, a `sybase-backup.sh` script to trigger a cumulative database backup manually or from a Bacula backup job using the **RunScript** directive is possible:

```
# /opt/bacula/scripts/sybase-backup.sh pubs2 Differential
```

This would save a single dump file like `C.pubs2.2018-11-22.18:04:42.000` where: '`C`' indicates a cumulative dump, '`pubs2` is a database name, '`2018-11-22.18:04:42`' the backup time stamp and '`000`' is a stripe number.

Database Transaction Backup

This type of database backup will make a copy of a transaction log, and removes the inactive portion of the log, provided the dump transaction is not running concurrently with another database dump.

To perform a Database Transaction backup execute a command such as

```
1> dump transaction pubs2 to 'sybacula::'
2> go
```

More information about the `dump` command and its additional parameters are available in the Sybase documentation *Reference Manual: Commands 16.0* in section *Commands*.

Again, a `sybase-backup.sh` script to execute a transaction database backup manually or by a Bacula backup job could be:

```
# /opt/bacula/scripts/sybase-backup.sh pubs2 Incremental
```

This would store a single dump file like: `T.pubs2.2018-11-20.18:11:00.000` where: '`T`' indicates a transaction dump, '`pubs2` is a database name, '`2018-11-20.18:11:00`' is the database backup time stamp and '`000`' a stripe number.

Stripes

A standard `dump database` or command described above runs a single backup stream which corresponds to the execution of a single Bacula job. This solution is convenient for small to moderately sized databases. When backing up large database sets, multiple streams can be used for database and transaction dump or load operations.

To use multiple streams, a suitable `dump` or command which includes `stripe on 'sybacula::'` options is needed. It is possible to use as many `stripe on 'sybacula::'` options as required. Each stripe option added to the command will execute an additional concurrent job with Bacula, which implies that concurrency limits will need to be verified or adjusted. Please see *Running Parallel Jobs* for more detailed information.

The following example executes a full database backup job with three concurrent backup streams, and accordingly three Bacula backup jobs.

```
1> dump database pubs2 to 'sybacula::'
2>  stripe on 'sybacula::'
3>  stripe on 'sybacula::'
4> go
```

Multiple stream functionality is available for any backup or restore job type. The multiple stripes functionality can be used for particular backup levels, for example with Full Database dumps, leaving other backup levels with a single stream only.

Up to 32 streams, including the main stream in the `to 'sybacula::'` clause, can be used. This is a Sybase ASE limit.

If a backup has been performed with multiple stripes, a restore needs to be run with an identical number of stripes.

Database System Objects

System Database Backups

The Bacula Enterprise Sybase SBT Plugin can be used to backup and restore the system databases `master`, `tempdb`, `model`, and `sybsystemdb` like any other user databases. These backups will be available in the Bacula catalog like any other database.

```
# /opt/bacula/scripts/bacula-sybase.sh
Updating cache ...
Searching catalog ... Jobs found.
Full dump:      D.master.2018-11-22.17_15_01
Full dump:      D.model.2018-11-22.17_21_21
Full dump:      D.sybsystemdb.2018-11-22.17_21_29
```

Scheduling Backups

The following considerations should be taken into account when scheduling Sybase Database backups.

- Any type of dump can be run while a Sybase ASE database is running. However, any database backup could slow down the system by consuming resources like disk I/O, network throughput, CPU cycles, etc. It is recommended to execute large dumps (especially full database dumps) during lower database utilization periods.
- Implementing cumulative database backups will reduce backup size, especially with databases that hold big read-only tables.
- The `master` database should be backed up regularly and frequently. In addition to regular backups, `master` should be backed up after each `create database`, `alter database`, and `disk init` command issued.
- The `model` database should be backed up whenever it is modified.
- The `dump database` backup command should be run immediately after creating a database, to make a copy of the entire database. `dump transaction` can not be run on a new database until a full database dump has been performed.

- Each time a cross-database constraint is added, or a table containing a cross-database constraint is dropped, both of the affected databases should be dumped.

Warning: Loading earlier dumps of these databases can cause database corruption. This is a Sybase ASE limitation.

- Use the *SAP Adaptive Server Enterprise*, section *System Administration Guide* documentation to plan for backup and recovery.

Restore

Listing Database Dumps

Before any database restore or recovery is started, a list of available database or transaction dumps is needed to prepare for recovery. To list all available database dumps, the `/opt/bacula/scripts/bacula-sybase.sh` script can be used. It simply queries a Bacula catalog and displays information in human readable format.

```
# /opt/bacula/scripts/bacula-sybase.sh
Updating cache ...
Searching catalog ... Jobs found.

Full dump:       D.pubs2.2018-11-19.14_11_07
  Transaction dump: T.pubs2.2018-11-19.14_11_12
  Transaction dump: T.pubs2.2018-11-19.14_11_15
  Transaction dump: T.pubs2.2018-11-19.14_11_18
 Incremental dump:   C.pubs2.2018-11-19.14_11_21
  Transaction dump: T.pubs2.2018-11-19.14_11_25
  Transaction dump: T.pubs2.2018-11-19.14_11_28
  Transaction dump: T.pubs2.2018-11-19.14_11_30
```

In the above example, the values like `D.pubs2.2018-11-19.14:11:07` need to be used for the `dump=...` option of a Sybase restore session. More details are available in the following sections.

The above information can be obtained manually by browsing the Bacula catalog with the `bconsole` or `BWeb` applications. In this case, the saved filenames like `D.pubs2.2018-11-19.14:11:07.000` are of interest, where the suffix `.000` shows the stripe number of a particular dump file saved.

When database dumps were performed using the `striped on 'sybacula::'` parameter of the `dump` command, then `bacula-sybase.sh` will show how many stripes were used for any particular dump, i.e.:

```
# /opt/bacula/scripts/bacula-sybase.sh
Updating cache ...
Searching catalog ... Jobs found.

Full dump:       D.pubs2.2018-11-19.14_46_47 - stripped on 2
  Transaction dump: T.pubs2.2018-11-19.14_46_55 - stripped on 2
  Transaction dump: T.pubs2.2018-11-19.14_46_58 - stripped on 2
  Transaction dump: T.pubs2.2018-11-19.14_47_01 - stripped on 2
 Incremental dump:   C.pubs2.2018-11-19.14_47_05 - stripped on 2
  Transaction dump: T.pubs2.2018-11-19.14_47_09 - stripped on 2
```

(continues on next page)

(continued from previous page)

```
  Transaction dump: T.pubs2.2018-11-19.14_47_12 - stripped on 2
  Transaction dump: T.pubs2.2018-11-19.14_47_15 - stripped on 2
```

The string `"stripped on 2"`[1] above indicates that two `"stripe on 'sybacula::'"` parameters will be needed to restore. More information on stripes can be found at *Stripes*.

Restoring a Database from a Full Backup

To restore a database it is necessary to always start from a Full Database backup. A command along with the selection of the required database dumpsusing the `dump=...` parameter is the starting point. The list of dumps to load can be obtained with the `bacula-sybase.sh` script as described in section *Listing Database Dumps*. The command might be executed as follows:

```
1> load database pubs2 from 'sybacula::dump=D.pubs2.2018-11-20.18_09_50'
2> go
```

If the Full Database backup was performed with multiple streams, the correct `load database` command needs to be prepared as described in section *Stripes*. The additional `stripe on` parameters of this command should not use any `dump=...` options, as the correct dump filenames will be selected automatically. The same procedure applies for any other backup type.

After a load operation has finished, the Sybase ASE does not bring the database online automatically, allowing the administrator to load subsequent cumulative or transaction log backups. Thus, after a restore process has been finalized, it is necessary to bring the restored database online manually, which is done with the command:

```
1> online database pubs2
2> go
```

More information about the `load` and commands and their additional parameters can be found in the Sybase documentation in the *Reference Manual: Commands 16.0*, section *Commands*.

Restoring a Database from a Differential Backup

Database restores of full backups can be completed by loading a subsequent differential backups. In most cases, the latest differential backup will be needed, as that consists of all changes recorded since the previous Full database dump. It is not required to restore any previous cumulative dumps – differential Backups in Baculas terminology – for successful recovery.

To execute a differential restore, the command to use is `load database cumulative` with the proper dump file in the `dump=...` parameter. Which file to use may be determined by using the `bacula-sybase.sh` script as described in section *Listing Database Dumps*. Thus, a command might be such as

```
1> load database pubs2 cumulative from 'sybacula::dump=C.pubs2.2018-11-20.18_
↪15_38'
2> go
```

Again, if the differential – "cumulative incremental" in Sybase terminology – backup was performed with multiple streams, the correct `load database` command has to be prepared as described in *Stripes*.

[1] The typo "stripped" instead of "striped" will be fixed in a later edition of this manual, and in the software itself.

The additional `stripe on` parameters of this command should not use any `dump=...` options, as in this case the correct dump filenames will be determined automatically by the plugin.

To finalize the restore now, if no subsequent transaction log backups will be needed, the `online database` command is used to bring the restored database online.

```
1> online database pubs2
2> go
```

More information about the `load` and commands and their parameters is available in the Sybase documentation, in *Reference Manual: Commands 16.0*, section *Commands*.

Restoring Database Transactions and PITR

Database recovery can be continued after loading a full dump and a subsequent cumulative incremental backup (described above) by loading and applying required transaction logs. All available and required database transactions should be selected and loaded. Restoration to a distinct point in time (Point in Time Recovery or PITR) before the latest transaction logged is possible by following the procedure in section *Loading Transaction Logs to a Point In Time (PITR)*.

For each transaction log backup – incremental backup in Bacula's terminology – a separate `load transaction` command is required. This command needs to be executed with the required database dump in the `dump=...` parameter. Which transaction log backup to restore may be determined using the `bacula-sybase.sh` script as described in section *Listing Database Dumps*.

An example command would be:

```
1> load transaction pubs2 from 'sybacula::dump=T.pubs2.2018-11-21.10_18_10'
2> go
```

Note that transaction logs need to be applied in their correct order, namely, oldest first, and no logs can be skipped.

Also note that this step needs to be repeated until all required or available transaction dumps are loaded to the restored database.

If transaction backups were performed with multiple streams, the correct `load transaction` command needs to be prepared as described in section *Stripes*. The additional `stripe on` parameters of this `load` command should not use any `dump=...` options. In this case also, the correct dump filenames will be chosen automatically. This is the same procedure for any backup type.

After all available or required transaction dumps were loaded to the restored database, the `online database` command is used to bring this database online:

```
1> online database pubs2
2> go
```

For more information about the `load` and commands and their parameters, the Sybase documentation should be consulted: *Reference Manual: Commands 16.0*, section *Commands*.

Loading Transaction Logs to a Point In Time (PITR)

It is possible to recover a database up to a specified point in time (Point In Time Recovery) in its transaction logs. To do so, the `until_time` option of the `load transaction` command is used. This is useful if, for example, a user inadvertently drops an important table. In this case, the `until_time` option would be set to a time just before the table was dropped.

To use the `until_time` option effectively after data has been destroyed, it is necessary to know the exact time the error occurred. In `isql`, the current timestamp can be determined using the `select getdate` command immediately after the time of the error:

```
1> select convert(char(26), getdate(), 109)
2> go
 --------------------------
 Nov 22 2018  2:06:30:610PM
```

After backing up the transaction log containing the error and loading the most recent database dump, plus possible cumulative (or differential) backups, all transaction logs that were created after the database was last dumped will be restored. The final one, containing the error, will then be loaded with the `until_time` option:

```
1> load transaction pubs2 from "sybacula::dump=T.pubs2.2018-11-22.12_07_30"
2> with until_time = "Nov 22 2018  2:06:00:0PM"
3> go
```

Note that in this example, we assume the time stamp was taken very quickly after the error was caused.

After a transaction log is applied with the `until_time` option, Adaptive Server restarts the database's log sequence. This implies that, until the database is dumped again, subsequent transaction logs after the load transaction using `until_time` can not be applied. You **must** dump the database before another transaction log backup can be done.

Restoring System Databases

In order to `load` the master database, the Sybase ASE engine must run in **single-user** mode. Please see the Sybase documentation, *SAP Adaptive Server Enterprise 16.0*, part *System Administration Guide: Volume 2* for more information about restoring system databases and required procedures.

Restoring Dumps to a Local Directory

With the `sybase-sbt` plugin it is possible to restore any database backups to a directory for manual database restore and recovery procedures. This feature is invoked by setting a proper `where=...` restore parameter pointing to a local directory of your choice.

Note that these restores need to be started through Bacula's command interface, for example with `bconsole`:

```
* restore client=sybase-fd fileset=SBT before="2018-11-21 10:00:00" \
    where=/tmp/restores select all done
Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /opt/bacula/working/sybase-dir.restore.5.bsr
```

(continues on next page)

(continued from previous page)

```
Where:           /tmp/restores
Replace:         Always
Fileset:         Full Set
Backup Client:   sybase-fd
Restore Client:  sybase-fd
Storage:         File1
When:            2018-11-22 15:32:51
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):
```

Note that, if stripes were used for such backups, some of the striped dump files will not be restored using the above method. It is necessary to restore missing striped dump files using the `JobId=...` restore parameter instead of the `before=...` option.

More Information

Troubleshooting

Connection Errors

In case an error message like

```
Msg 4002, Level 14, State 1:
Server 'SYBASECENTOS':
Login failed.
CT-LIBRARY error:
    ct_connect(): protocol specific layer: external error: \
        The attempt to connect to the server failed.
```

is encountered using a script like `sybase-backup.sh`, the contents of the file `/opt/bacula/sybase/sybase.env`, which contains parameters used to connect to the Sybase dataserver, should be checked.

Of particular importance are the `SYBSERVER` and `SAPASSWORD` settings.

Tracing

Job tracing information can be generated using the following parameters to the `dump` or `load` commands:

- `trace=<file>` provides the name of a file to store tracing information from the `libsybacula.so` module in.
- `debug=<level>` to set up a specific debug level for tracing. The higher the level, the more detailed the tracing information will be.

If the a `trace=...` parameter is omitted, but the `debug` one is provided, the default trace file will be `sybacula.<PID>.trace` in the or `C:/Program Files/Bacula/sybase/` directory.

An example is

```
1> dump database pubs2 to 'sybacula::debug=100'
2> go
```

Limitations

- Database backup and restore of compressed data must occur on the same platform. This is a Sybase Backup Server limitation.
- Automatic restore of striped backups to a local directory could skip some striped dump files. Manual restore using the JobId parameter is required. This limitation does not apply when restoring to a Sybase ASE dataserver.
- No more than 32 streams, including the main one in the `to 'sybacula::'` clause, can be used. This is a Sybase ASE limitation.
- Job estimation is not supported. This limitation might be removed in the future.
- Plugin listing mode is not supported. This limitation might be removed in the future.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5.6 PostgreSQL Plugin

This chapter aims at presenting the reader with information about the Bacula Enterprise PostgreSQL Plugin. The document briefly defines the scope of its operations, describes the target technology of the Plugin, and presents its main features and various techniques and strategies to backup PostgreSQL with Bacula Enterprise.

Scope

This Plugin is available for 32 and 64-bit Linux platforms, and supports all officially supported PostgreSQL versions since version 8.4.

Features

The PostgreSQL Plugin is designed to simplify backup and restore procedure of PostgreSQL clusters, so that the backup administrator does not need to know about internals of Postgres backup techniques or write complex scripts. The Plugin will automatically take care of backing up essential information such as configuration, users definition or tablespace.

The PostgreSQL Plugin supports both Dump and Point In Time Recovery (PITR) backup techniques.

The PostgreSQL Plugin is compatible with Copy/Migration jobs. Read the CopyMigrationJobsReplication for more information.

Backup Strategies

The following article presents the comparison of backup strategies for the PostgreSQL Plugin.

Choosing Between PITR and Dump

The following table helps choosing between backup techniques supported by the Bacula Enterprise PostgreSQL Plugin. Major functionalities such as being able to restore databases to any point in time, or being able to filter objects during backup or restore should be used to guide through the backup design. It is quite common to combine Dump and PITR techniques for the same Cluster. In the table, the Custom format corresponds to the Custom Dump format of pg_dump and the Dump format of our Plugin corresponds to the plain format of pg_dump.

Note: Regardless the backup method used, no temporary local disk space is necessary to save any temporary file.

Table 45: PostgreSQL dump vs PITR backup

	Custom 1	Dump	PITR
Can restore directly a single object (table, schema, . . .)	Yes	No	No
Backup speed	Slow	Slow	Fast
Restore speed	Slow	Very Slow	Fast
Backup size	Small	Small	Big
Can restore at any point in time	No	No	Yes
Incremental/Differential support	No	No	Yes
Can restore in parallel	Yes 2	No	–
Online backup	Yes	Yes	Yes
Consistent	Yes	Yes	Yes
Can restore to previous major version of PostgreSQL	No	Yes 3	No
Can restore to newer major version of PostgreSQL	Yes	Yes	No

1 Custom Dump format is the default.

2 Run the most time-consuming parts of pg_restore - those which load data, create indexes, or create constraints - using multiple concurrent jobs. This option can dramatically reduce time to restore a large database to a server running on a multiprocessor machine. It requires to store the Dump to the disk first.

3 To restore a SQL plain Dump to a previous version of PostgreSQL, you might have to edit the SQL file if you are using some features that are not available in the previous version. Generally, restoring to a previous version of PostgreSQL is not supported or not guaranteed.

Installation

Prerequisites

As with all Bacula plugins, the **Plugin Directory** directive in the **FileDaemon** resource of the `bacula-fd.conf` file needs to be set:

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

PostgreSQL Installation with BIM

In order to install the PostgreSQL Plugin with BIM, install the File Daemon with BIM and choose to install the PostgreSQL Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

PostgreSQL Plugin Installation with Package Manager

The PostgreSQL Plugin is available as a Bacula Enterprise package for all supported platforms.

The Plugin must be installed on the primary node in the PostgreSQL Cluster. The PostgreSQL client software package, usually "postgresql-client", should also be installed as it provides tools such as `pg_dump` and `psql` which are used by the plugin, or may be useful for diagnostic purposes.

In order to get information and/or data from the PostgreSQL database, the Bacula Enterprise Plugin uses PostgreSQL standard commands using the unix "postgres" user. This user account needs to be able to connect to the Cluster without interactive password dialog. Such a configuration (which is the default one) can be achieved using the following entry in `pg_hba.conf`. The following entry should be the first one in the list:

```
local    all    postgres        ident
```

If remote databases need to be accessed, or disabling the interactive password authentication method is not an option, it is possible to define a `pgpass` file.

Configuration

The following chapter aims at presenting how to configure the PostgreSQL Plugin.

Automatic Object Integration

Since Bacula version 16.0.7, a new solution has been introduced, so that each object can be backed up separately with different Jobs to maximize the throughput and the resiliency. It is highly recommended to use this new solution for that purpose - *Automatic Object Integration (Scan Plugin)*. See an example for PostgreSQL.

PITR Configuration

PostgreSQL Server Configuration for PITR

In order to use the Point In Time Recovery feature of PostgreSQL, WAL Archiving needs to be enabled. The procedure differs between major PostgreSQL version, so we advise to read the PostgreSQL documentation corresponding to the Cluster version; for PostgreSQL version 9.1, for example, it can be found here: http://www.postgresql.org/docs/9.1/static/continuous-archiving.html

Basically, on 8.4, the `archive_command` and the `archive_mode` settings need to be configured.

```
# on 8.3 - 8.4
archive_mode = on
archive_command = 'test ! -f /mnt/waldir/%f && cp %p /mnt/waldir/%f'
```

For 9.x, it is needed to configure `archive_command`, `wal_level`, and `archive_mode`.

```
# on 9.0 - 9.1
wal_level = archive
archive_mode = on
archive_command = 'test ! -f /mnt/waldir/%f && cp %p /mnt/waldir/%f'
```

The `/mnt/waldir` directory should be purged from time to time when backup jobs are successful. We do not recommend to remove WAL files just after a backup job. If something is going wrong with the backup job and files are no longer on the database server, it will not be possible to easily re-start the backup job.

```
# This example will remove WAL files older than 14 days
find /mnt/waldir -type f -mtime +14 -exec rm -f {} \;
```

As shown below, the PostgreSQL plugin needs to know where WAL files are stored after archiving. In this example, `archive_dir` would point to `/mnt/waldir`.

Note: It may be useful to compress WAL files in the `archive_command` using something like:

```
```

archive_command = ‘test ! -f /mnt/waldir/%f.gz && gzip -c %p > /mnt/waldir/%f.gz’

In this case, the **restore_command** in `recovery.conf` will need to be modified during the restore, as the PostgreSQL Plugin will not be able to reverse the custom `archive_command` automatically. It is good practice to put the needed `restore_command` command as comment into the `postgresql.conf`.

PostgreSQL Plugin Configuration for PITR

With the PITR option, the PostgreSQL Plugin uses Accurate mode information to handle Differential backups, thus the **Accurate** option in the backup Job resource needs to be enabled. If Accurate mode is not used in Differential level backups, orphan data files in the Cluster directory may occur when restoring. Note that using Accurate mode is not mandatory only Full and Incremental backups are planned.

```
Job {
 Name = "Postgresql-PITR"
 Client = laptop1-fd
 Fileset = FS_postgresql
 Accurate = yes
 ...
}

Fileset {
 Name = FS_postgresql
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = "postgresql: mode=pitr"
```

(continues on next page)

(continued from previous page)

```
  }
}
```

In the PITR mode, the PostgreSQL Plugin also accepts the parameters listed here.

If PostgreSQL clusters are planned to be restored using the file relocation feature of Bacula, it is useful to use the **PrefixLinks** directive in the prepared Restore Job.

For example, if symbolic links in the PGDATA cluster directory are used, like for pg_wal[1] or tablespaces, the default Restore Job will re-create those symbolic links pointing to their original locations and not the restored datas directory, which will break PostgreSQLs recovery process.

On the other hand, if restored files are planned to be moved to their original locations later, outside of Bacula's restore process, all files linked with absolute names will be broken.

PostgreSQL Plugin Options in PITR Mode

Table 46: PostgreSQL Plugin Options in PITR Mode

Option	Comment	Default	Example
mode=pit	Needed to enable PITR backup.	custom	
abort_on_	Abort the job after a connection error with PostgreSQL (available with Bacula Enterprise 8.2.0 and later).	not set	abort_on_error
archive_d	Should point to where you are archiving WAL with the `archive_command`.	pg_wa	
bin_dir	PostgreSQL binaries location.		bin_dir=/opt/pg9.1/bin
fast_backu	Use fast backup option in pg_backup_start() procedure. It will create a peak of IO if used.	false	fast_backup
pgpass	Path to PostgreSQL password file.		pgpass=/etc/pgpass
user	PostgreSQL Unix super user. A standard user cannot be used here.	postgres	user=dba
service	PostgreSQL connection information.		service=main
timeout	Specify a custom timeout (in secs) for commands sent to PostgreSQL.	300	timeout=600
tmp_dir	Where the plugin will create files and scripts for the database backup (available with Bacula Enterprise 6.6.6 and later).	`/tmp`	tmp_dir=/othertmp
use_sudo	Use sudo to execute Postgresql commands.		use_sudo
unix_user	Use specific unix user to execute Postgresql commands. The unix user owning the cluster is required.	postgres	unix_user=bob

[1] In old versions of postgresql (<= 9.x), pg_wal was pg_xlog - it was renamed in version 10.

Backup Level in PITR

When using PITR mode, depending on the Job level, the PostgreSQL Plugin will do the following:

- For a **Full** backup, the Plugin will backup the data directory and all WAL files generated during the backup.
- During an **Incremental** backup, the Plugin will force the switch of the current WAL, and will backup WAL files generated since the previous backup.
- During a **Differential** backup, the Plugin will backup data files that changed since the latest Full backup, and it will backup WAL files generated during the backup.

Note: It is not possible to run two concurrent Full or Differential jobs at the same time.

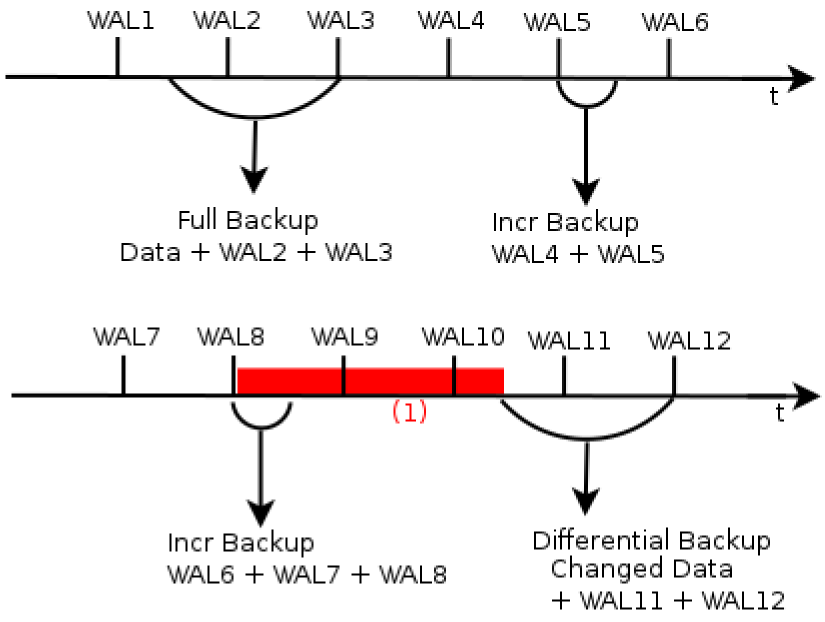


Fig. 118: Backup Level Impact in PITR Mode

where (1) is period between the latest Incremental and the new Full or Differential backup.

Note: It is recommended to visit ScheduleConsiderationForPITR for further information.

Note that replaying a long list of WAL files may take considerable time on a large system with lot of activities.

Schedule Consideration for PITR

In order to be able to restore to any point in time between the latest Incremental and a previous Full or Differential backup (see the (1) area in figure pg-level), it is good practice to schedule an Incremental to the same time of Full or Differential backups. The **Maximum Concurrent Jobs** setting on the Client and the Job resource should allow to run two jobs concurrently.

```
Schedule {
  Name = SCH_PostgreSQL

  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sun at 23:05
}
```

Without this schedule configuration, it will not be possible to restore to a specific point in time during the period between the latest Incremental backup and the next Full or Differential backup. Note that disabling `Allow Duplicate Jobs` prevents starting two Full backup Job at the same time.

Dump Configuration

With the Dump option, Bacula cannot perform Incremental or Differential backup.

```
Job {
 Name = "Postgresql-dump"
 Client = laptop1-fd
 Fileset = FS_postgresql_dump
 ...
}

Fileset {
 Name = FS_postgresql_dump
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = postgresql
 }
}
```

With the above example, the Plugin will detect and backup all databases of the Cluster.

```
Fileset {
 Name = FS_postgresql
 Include {
   Options {
     Signature = MD5
     Compression = GZIP
   }
   Plugin = "postgresql: database=bacula"
   Plugin = "postgresql: database=master"
```

(continues on next page)

(continued from previous page)

```
 }
}
```

In this example, the Plugin will backup only the databases `bacula` and `master`.

In Dump mode, the PostgreSQL Plugin also accepts the parameters listed here.

```
Fileset {
 Name = FS_postgresql_dump
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "postgresql: use_sudo user=rob dump_opt=\"-T temp\""
 }
}
```

In this example, the PostgreSQL Plugin will use the Unix account "rob" to perform a Custom Dump backup with the PostgreSQL "rob" account excluding tables named "temp".

In order to use the `sudo` wrapper, it is needed to comment out the following option in `/etc/sudoers`.

```
Defaults requiretty
```

PostgreSQL Plugin Options in Dump Mode

Table 47: PostgreSQL Plugin Options in Dump Mode

Option	Comment	Default	Example
dump_opt	This string will be passed to the pg_dump command. 1	-c -b -F p	dump_opt="-c"
user	PostgreSQL user to use for PostgreSQL commands.	postgres	user=rob
unix_user	Unix user to use for PostgreSQL commands. 2	set to `user`	unix_user=pg1
service	pg_service to use for PostgreSQL commands.		service=main
pgpass	Path to PostgreSQL password file.		pgpass=/etc/pgpass
use_sudo	Use sudo instead to run PostgreSQL commands (when not root).		use_sudo
compress	Use pg_dump compression level. 0-9, 0 is off.	0	compress=5
database	Will backup on databases matching this string.		database=prod*
exclude	Will exclude databases matching this string. 5		exclude=test*
bin_dir	PostgreSQL binaries location.		bin_dir=/opt/pg9.1/bin
tmp_dir	Where the plugin will create files and scripts for the database backup. 3	/tmp	tmp_dir=/othertmp
abort_on_err	Abort the job after a connection error with PostgreSQL. 4	not set	abort_on_error
timeout	Specify a custom timeout (in secs) for commands sent to PostgreSQL.	60 seconds	timeout=120

1 The dump_opt option cannot be used to backup remote servers. Please use PGSERVICE instead

2 Available with Bacula Enterprise 8.4.12 and later.

3 Available with Bacula Enterprise 6.6.6 and later.

4 Available with Bacula Enterprise 8.2.0 and later.

5 Available with Bacula Enterprise 14.0.5 and later.

Note: The `database` and `exclude` options support regex.

Service Connection Information

The connection service file allows PostgreSQL connection parameters to be associated with a single service name. That service name can then be specified by a PostgreSQL connection, and the associated settings will be used.

The connection service file can be a per-user service file at `` /.pg_service.conf``, can have its location specified by the environment variable **PGSERVICEFILE**, or it can be a system-wide file at `/etc/pg_service.conf` or in the directory specified by the environment variable **PGSYSCONFDIR**. If service definitions with the same name exist in the user and the system files, the user file takes precedence.

The file uses an INI file format where the section name represents the service name and the parameters are connection parameters; see http://www.postgresql.org/docs/9.1/static/libpq-connect.html for a list. For example:

```
# comment
[main]
port=5433
```

If a password is needed to connect, the `pgpass` option in the Plugin command string can be used to make the Plugin define the needed **PGPASSFILE** environment variable.

Testing Database Access

The `estimate` command can be used to verify that the PostgreSQL Plugin is well configured.

```
* estimate listing job=pg-test
...
```

If `estimate` or the job output displays the following error:

```
Error: Can't reach PostgreSQL server to get database config.
```

the next steps should be to verify that the Bacula Enterprise PostgreSQL plugin can retrieve information using the `psql` command as "postgres" user on the Client.

To verify if the "postgres" user can connect to the PostgreSQL Cluster, the `psql -l` command can be used, and it should list all databases in the Cluster:

```
postgres%  psql -l
   List of databases
   Name    |  Owner   |
-----------+----------+
```

(continues on next page)

(continued from previous page)

```
postgres   |   xxx
template0  |   xxx
template1  |   xxx
```

If options such as **-h localhost** are needed on the `psql` command line, a service file as described in **servicecon** will be required.

Operations

The following chapter aims at presenting possible operations with the PostgreSQL Plugin.

Backup

Estimate Information

The `estimate` command will display all information found by the PostgreSQL Plugin.

Note: In Dump mode, Bacula can not compute the Dump size for databases, so it will display database size instead.

Backup Information in Dump Mode

The PostgreSQL Plugin will generate the following files for a Cluster containing the single database "test":

```
@PG/main/roles.sql
@PG/main/postgresql.conf
@PG/main/pg_hba.conf
@PG/main/pg_ident.conf
@PG/main/tablespaces.sql

@PG/main/test/createdb.sql
@PG/main/test/schema.sql
@PG/main/test/data.sqlc
```

Table 48: Backup Content in Dump Mode

File	Context	Comment
roles.sql	global	List of all users, their password and specific options
postgresql.conf	global	PostgreSQL cluster configuration
pg_hba.conf	global	Client connection configuration
pg_ident.conf	global	Client connection configuration
tablespaces.sql	global	Tablespaces configuration for the PostgreSQL cluster
createdb.sql	database	Database creation script
schema.sql	database	Schema database creation script
data.sqlc	database	Database data in custom format, contains everything needed to restore
data.sql	database	Database data in dump format

Restore

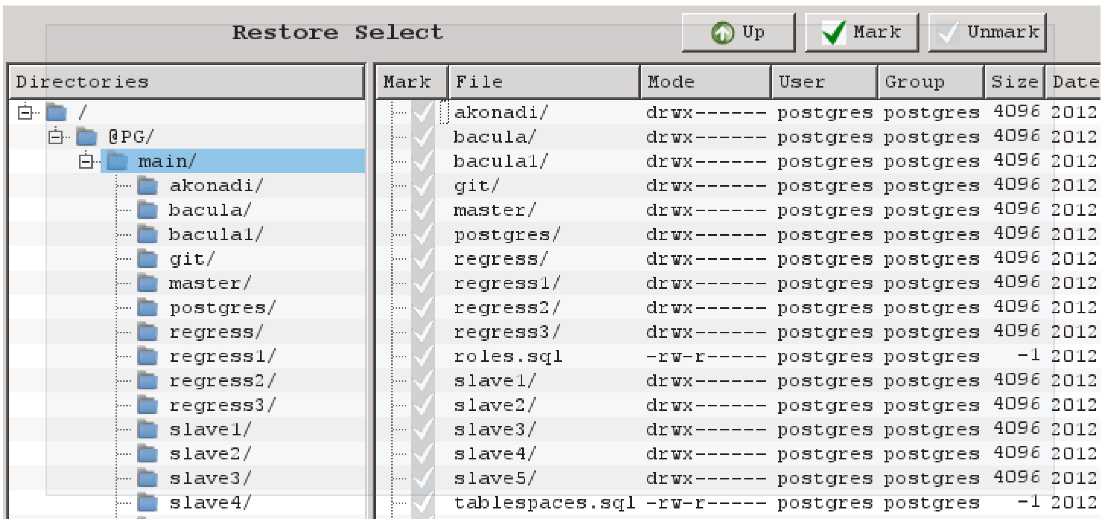
Restoring Using Dumps

Restoring Users and Roles

To restore roles and users to a PostgreSQL Cluster, the `roles.sql` file located in `/@PG/<service>/roles.sql` needs to be selected.

Then, using **where=/** or **where=**, the Plugin will load this SQL file to the database. If some roles already exist, errors will be printed to the Job log.

Note: It is possible to restore the `roles.sql` file to a local directory, edit it, and load it using `psql` to restore only a selection of its original contents.



```
Directories
/
  @PG/
    main/
      akonadi/
      bacula/
      bacula1/
      git/
      master/
      postgres/
      regress/
      regress1/
      regress2/
      regress3/
      slave1/
      slave2/
      slave3/
      slave4/
```

Mark	File	Mode	User	Group	Size	Date
✓	akonadi/	drwx------	postgres	postgres	4096	2012
✓	bacula/	drwx------	postgres	postgres	4096	2012
✓	bacula1/	drwx------	postgres	postgres	4096	2012
✓	git/	drwx------	postgres	postgres	4096	2012
✓	master/	drwx------	postgres	postgres	4096	2012
✓	postgres/	drwx------	postgres	postgres	4096	2012
✓	regress/	drwx------	postgres	postgres	4096	2012
✓	regress1/	drwx------	postgres	postgres	4096	2012
✓	regress2/	drwx------	postgres	postgres	4096	2012
✓	regress3/	drwx------	postgres	postgres	4096	2012
✓	roles.sql	-rw-r-----	postgres	postgres	-1	2012
✓	slave1/	drwx------	postgres	postgres	4096	2012
✓	slave2/	drwx------	postgres	postgres	4096	2012
✓	slave3/	drwx------	postgres	postgres	4096	2012
✓	slave4/	drwx------	postgres	postgres	4096	2012
✓	slave5/	drwx------	postgres	postgres	4096	2012
✓	tablespaces.sql	-rw-r-----	postgres	postgres	-1	2012

Fig. 119: PostgreSQL Cluster Contents During Restore

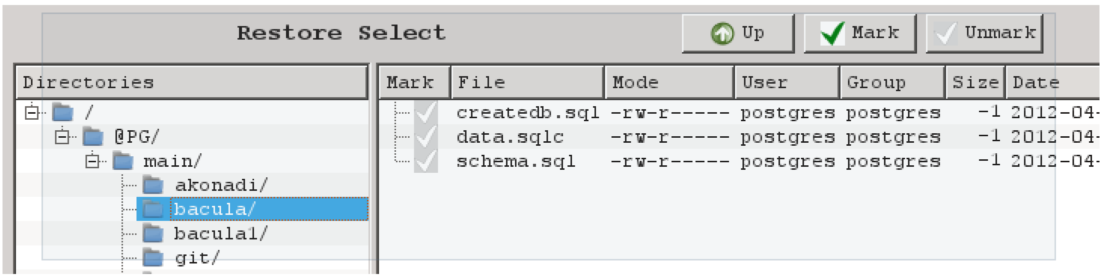
Restoring Database Structure

To restore only the database structure using the Bacula Enterprise Postgresql Plugin, the file `createdb.sql` located in the database directory needs to be selected during the restore process. To recreate the SQL database schema, the `schema.sql` file is used which contains all commands needed to recreate the database schema. The `schema.sql` file must be restored to disk and loaded manually into the database using the `psql` command.

Restoring Single Database

To restore a single database with the Bacula Enterprise Postgresql Plugin, the appropriate files from the database directory are selected during the restore process.

To restore the database with its original name, the selection should only contain the data file (`data.sqlc` or `data.sql`). If the `createdb.sql` file is also selected, harmless messages might be printed during the restore.



```
Directories
/
  @PG/
    main/
      akonadi/
      bacula/
      bacula1/
      git/
```

Mark	File	Mode	User	Group	Size	Date
✓	createdb.sql	-rw-r-----	postgres	postgres	-1	2012-04-
✓	data.sqlc	-rw-r-----	postgres	postgres	-1	2012-04-
✓	schema.sql	-rw-r-----	postgres	postgres	-1	2012-04-

Fig. 120: Database Contents During Restore

To restore a single database to a new name, the two files `createdb.sql` and `data.sqlc` (or `data.sql`) must be selected. The **where** parameter is used to specify the new database name. If **where** is set to a single word consisting of only `a..z`, `0-9` and `_`, Bacula will create the specified database and restore the data into it.

```
* restore where=baculaold
...
cwd is: /
$ cd /@PG/main/bacula
cwd is: /@PG/main/bacula/
$ m data.sqlc
$ m createdb.sql
$ ls
schema.sql
*data.sqlc
*createdb.sql
```

If the restore process has an error such as `ERROR: database "xxx" already exists`, the `createdb.sql` can be skipped in the restore selection.

If the **replace** parameter is set to **never**, Bacula will check the database list, and will abort the Job if the database currently restored already exists.

Using **replace=always** is not recommended.

If the **where** parameter is a directory (containing `/`), Bacula will restore all files into this directory. Doing so, it is possible to use `pg_restore` directly and restore only particular contents, such as triggers, tables, indexes, etc.

Note: Some databases such as `template1`, `postgresql` or databases with active users can not be replaced.

Restoring Dump Files to Directory

To restore SQL dumps to a directory, the **where** parameter needs to be set to indicate an existing directory.

```
* restore where=/tmp
```

Restoring Single Table

To restore a single item such as a table, it is currently needed to restore the dump file to a directory and use the `pg_restore` command.

Restoring Complete Cluster Using PITR

Useful information for this disaster recovery scenario can be found in the PostgreSQL manual, for example at: http://www.postgresql.org/docs/9.1/static/continuous-archiving.html

The overall process is as follows:

1. Stop the server, if it's running.
2. If the space to do so is available, the whole Cluster data directory and any tablespaces should be copied to a temporary location in case they are needed later.

   It is very important to use the "-a" parameter to copy the files, so that the files retain their ownership

   ```
   root@dc-u24Postgres16db-test:/var/lib/postgresql/16/main# cp -a pg_wal/*␣
   ↪/tmp/pg_wal.backup/
   root@dc-u24Postgres16db-test:/var/lib/postgresql/16/main# ls -lhtr /tmp/
   ↪pg_wal.backup/
   total 97M
   -rw------- 1 postgres postgres  16M Jul  2 17:18 000000010000000000000001
   -rw------- 1 postgres postgres  16M Jul  2 17:18 000000010000000000000002
   -rw------- 1 postgres postgres  349 Jul  2 17:18␣
   ↪000000010000000000000002.00000028.backup
   -rw------- 1 postgres postgres  16M Jul  2 17:26 000000010000000000000003
   -rw------- 1 postgres postgres  16M Jul  2 17:26 000000010000000000000004
   -rw------- 1 postgres postgres  370 Jul  2 17:26␣
   ↪000000010000000000000004.00000028.backup
   -rw------- 1 postgres postgres  16M Jul  2 17:36 000000010000000000000005
   drwx------ 2 postgres postgres 4.0K Jul  2 17:36 archive_status
   -rw------- 1 postgres postgres  16M Jul  2 17:36 000000010000000000000006
   root@dc-u24Postgres16db-test:/var/lib/postgresql/16/main#
   ```

   Note: This precaution will require having enough free space to hold two copies of the existing databases. If enough space is not available, at least the contents of the `pg_wal` subdirectory of the Cluster data directory should be copied, as it may contain logs which were not archived before the system went down.

3. Clean out all existing files and subdirectories below the Cluster data directory and the root directories of any tablespaces being used.

4. Restore the database files from the backups. If tablespaces are used, it is strongly recommended to verify that the symbolic links in `pg_tblspc/` were correctly restored. The **PrefixLinks** restore Job option can be useful here.

5. Any files present in `pg_wal` can be removed; these came from the backup and are therefore probably obsolete rather than current. Normally, this directory should be empty after a restore.

6. If there are unarchived WAL segment files that were saved in the step 2, they need to be copied back into `pg_wal/` (it is best to copy, not move them, so that the unmodified ones are available if a problem occurs and the process needs to be done again).

7. The recovery command file `recovery.conf.sample` inside the Cluster data directory may need to be edited and renamed to `postgresql.recovery.conf`. It may be useful to temporarily modify `pg_hba.conf` to prevent ordinary users from connecting until the recovery has been verified.

8. Start the server. The server will go into recovery mode and proceed to read through the archived WAL files it needs. Should the recovery be terminated because of an external error, the server can simply be restarted and it will continue recovery. Upon completion of the recovery process, the server will rename `recovery.conf` to `recovery.done` (to prevent accidentally re-entering recovery mode in case of a later crash) and then commence normal database operations.

   ```
   # su postgres
   $ cd /path/to/the/data/directory
   $ mv recovery.conf.sample recovery.conf
   $ vi recovery.conf
   $ pg_ctl -D $PWD start
   ```

9. The contents of the databases restored should be verified to ensure it was recovered to the desired state. If not, return to step 1.

10. If all is well, users can be allowed to connect by restoring `pg_hba.conf` to its normal contents.

Warning: About tablespaces and symlinks:

When applying logs, PostgreSQL needs to create the tablespace directory to re-create the tablespace, and PostgreSQL doesn't support the relocation. So, when replaying logs, it will overwrite or fail on this operation.

Limitations

- Postgres developers didn't implement the backup routines with data deduplication in mind, therefore the results may not be ideal. However the `cluster` command may be used at times to enhance the deduplication ratio as it physically reorders the data according to the index information. However, notice that this command requires exclusive lock while it is running, and also may quite heavy on CPU and I/O resources.
- In PITR mode, each Job should contain a single Cluster. To backup multiple clusters installed on the same client, multiple jobs are needed.
- In Dump mode, PostgreSQL version 8.3 and earlier may not support automatic restore through the Bacula Enterprise PostgreSQL plugin. The `pg_restore` program can produce the following error: `WARNING: invalid creation date in header`. In this case, it is needed to restore data to a directory, and run `pg_restore` in a terminal.
- The **clean** option requires PostgreSQL 9.1 or later.

- The backup will include all files in the data directory and tablespaces, including the configuration files and any additional files placed in the directory by third parties, except certain temporary files managed by PostgreSQL. But only regular files and directories are copied, except that symbolic links used for tablespaces are preserved. Symbolic links pointing to certain directories known to PostgreSQL are copied as empty directories. Other symbolic links and special device files are skipped. See: https://www.postgresql.org/docs/current/protocol-replication.html for the precise details.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

5.7 Amazon RDS Plugin

The following article aims at presenting the reader with information about the **Bacula Enterprise Amazon RDS Plugin**.

Amazon Relational Database Service (Amazon RDS) is a managed database service provided by Amazon Web Services (AWS) that simplifies the setup, operation, and scaling of relational databases in the cloud. RDS supports several popular database engines, including PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server. The service can be deployed via managed Amazon EC2 instances, or in fully managed format with Amazon Aurora. It includes limited automated backups in the AWS Cloud, software patching, and database monitoring, as well as automatic failover and replication to enhance availability and reliability. Additionally, RDS provides scalable storage and compute resources, enabling users to adjust capacity as needed without experiencing downtime.

The Bacula Enterprise Amazon RDS Plugin provides backup and restore of database instances (whether standalone or clustered) running in Amazon RDS, managing the whole snapshot livecycle without limitations. It also offers the optional feature of automatically exporting data to an S3 bucket in the AWS cloud and to extract it to send it to any other storage system managed by Bacula Enterprise, whether in the cloud or on-premise (including any type of disk, tape or alternative cloud). The entire process operates in an agent-less manner, providing a high degree of flexibility for executing various operations and delivering superior performance when the underlying network is suitable for the backup target. Once the data from a database has been protected by Bacula Enterprise, it can be restored to a new Cloud RDS instance or to a on-premise database engine, allowing the user to use this plugin as an export tool if there is a need to migrate data to a different engine.

Through subchapters, more in-depth information can be found about the topics below.

Scope

Bacula Enterprise Amazon RDS Plugin currently supports backup and restore database instances or clusters deployed via the Amazon RDS service, provided there is a connection to the appropriate Amazon RDS API.

When export and download service is used, access to the Amazon S3 service and API is also required.

This plugin has been available since **Bacula Enterprise 18.2**, and needs to be deployed in a Linux host.

Features

The main feature of the **Bacula Enterprise Amazon RDS Plugin** is its capability to provide backup and restore for Amazon RDS instances or clusters.

The primary backup layer manages the snapshots of the database instances or clusters, executing them from Bacula Enterprise according to a user-defined schedule and maintaining the desired number of snapshots. On top of this, the plugin can export data to an S3 bucket and to backup the contents of the export operation, resulting in backup jobs that include one Apache Parquet archive for each table, along with several metadata files pertaining to the database and the snapshot. Finally, the plugin is able to manage the complete livecycle of the snapshots created and the exported data. All elements can be retained or cleaned up in a flexible, governed by the configuration settings of Bacula Enterprise.

Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It provides high performance compression and encoding schemes to handle complex data in bulk. For further details, refer to: - https://parquet.apache.org/

Apache Parquet is an independent database format, and the Bacula Enterprise Amazon RDS Plugin offers the ability to restore protected files to a PostgreSQL or MySQL database running on the same or a completely different underlying architecture. This includes both cloud and on-premise solutions, and it is not required that the source database is the same engine as the destination database, which means this plugin can be used as a migration tool for the protected data.

General Features

Below, there is a list of general features this plugin offers:

- Backup and restore of Amazon RDS instances
- Backup and restore of Amazon RDS clusters
- Amazon AWS API-based backups
- Automatic multi-threaded download or upload operations
- Network resiliency mechanisms
- Instances and volume discovery capabilities
- List instances and snapshots through query or list commands
- Auto-generation capabilities when combined with the Scan Plugin
- Restore data to Amazon RDS from the previously generated snapshots in the cloud
  - To the original location (region, availability zone, etc.)
  - To a different location (region, availability zone, etc.)
  - With the same original instance or cluster configuration
  - With a different instance or cluster configuration
  - Point-in-Time recovery
- Restore data from snapshots done during the backup or from any existing identifier
- Restore data to a running database engine from the data stored in Apache Parquet format
- Full & Incremental backup levels
  - Snapshots are Incremental backups by nature in Amazon RDS, as they are based in a non-visible EBS layer

  - Exports to S3 include always all data from the selected tables
- Advanced instance selection for backup, filtering by any parameter
- Ability to select whether to export data to S3
- Ability to select whether to download data from S3
- Ability to select if all tables or just part of them should be exported to S3 using Apache Parquet format
- Automatic snapshot cleanup before and after backups based on the specified retention
- Configurable automatic S3 bucket cleanup after backups

Architecture

Bacula Enterprise Amazon RDS Plugin is a Bacula File Daemon plugin that utilizes the Amazon Web Services APIs to interact with the Amazon Cloud (launching operations in the Amazon Cloud, retrieving data from it and feeding it at restore time). The plugin runs a Java Daemon which employs the official Amazon AWS SDK version 2.x built by Amazon.

All the information is obtained through secure and encrypted HTTPS queries to AWS from the File Daemon (and via the aforementioned Java Daemon), where the plugin is installed. The specific URLs will depend on the region and the operation being performed.

To get more information about AWS implied APIs, visit:

- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/ProgrammingGuide.html
- https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html

The metadata for each backed up element is stored in Bacula in JSON format, which encompasses database instances, cluster instances and snapshots of both elements. The files related to the database exports are stored in S3 and are temporarily downloaded to the host where the plugin operates, prior to being sent to the Storage Daemon.

The backup and restore processes use different parallelization techniques in order to maximize performance and overcome latency times when communicating with AWS Parallelization. Additionally, the plugin also supports the parallelization of multiple backup jobs.

Below, there is a simplified vision of the architecture of this plugin within a generic **Bacula Enterprise** deployment:

Installation

This article describes how to install Bacula Enterprise Amazon RDS Plugin.

Prerequisites

- The Bacula File Daemon and the Amazon RDS Plugin need to be installed on the host that will connect to the AWS Cloud.
- The plugin must be deployed in a host running Linux. It is possible to use any of the supported **Linux** distributions of Bacula Enterprise, including RHEL, Debian, Ubuntu or Suse Linux Enterprise Server as some examples.

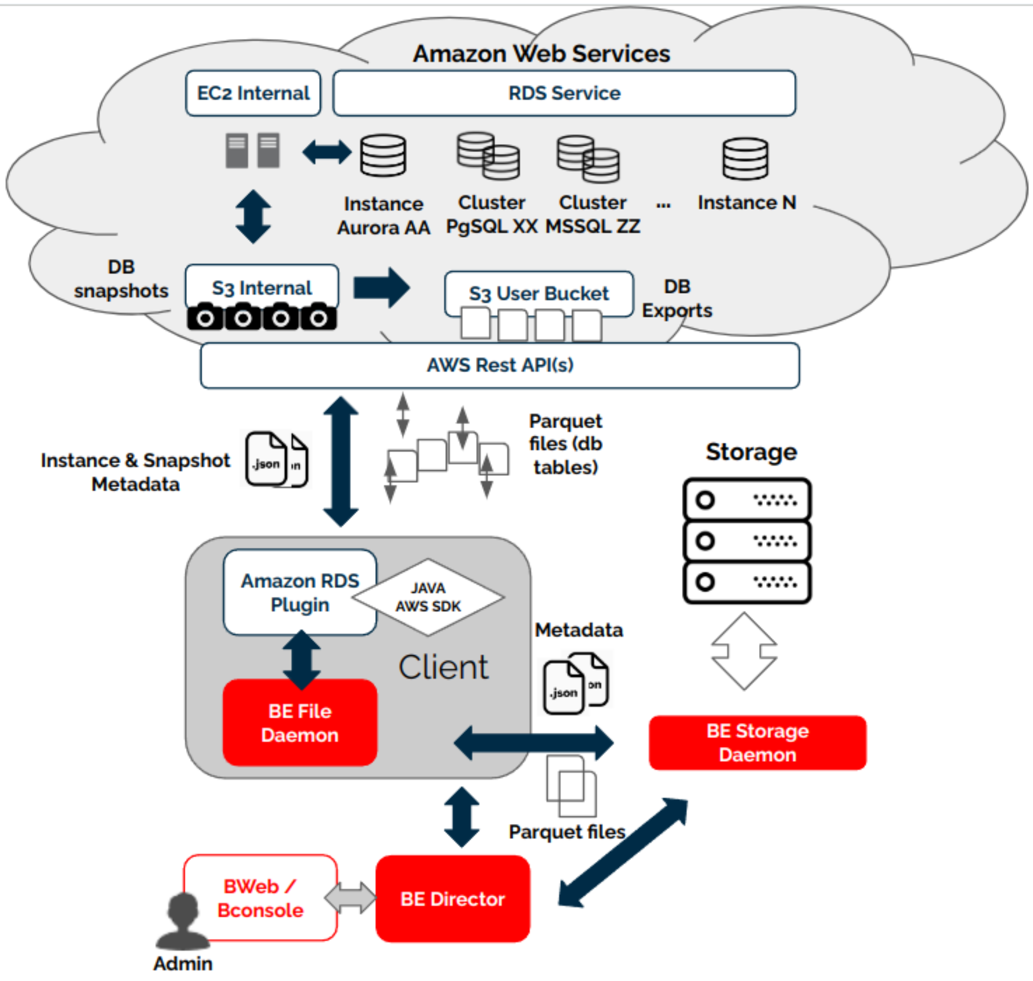


Fig. 121: Amazon RDS Plugin Architecture

- The plugin works via a Java daemon, therefore Java needs to be installed into the host using a JRE or JDK package (openjdk-11-jre for example). The Java environment should be version 11 or higher, and the Java binary must be accessible in the system PATH.
- The memory and computational requirements can vary significantly based on the configuration and usage of the plugin, such as parallelization and data size for backup etc. It is recommended to have a minimum of **4GB RAM** on the server where the File Daemon runs. By default, each job may use up to 512Mb of RAM in demanding scenarios, although it is typically less. In certain cases, this could be higher. Memory limits can be adjusted (see: Out of memory).
- Amazon Web Services REST APIs are used to perform all plugin operations. Therefore, they must be accessible through HTTPS from the host where Bacula FD and the plugin will be deployed.
- In order to fetch data, an access key comprising a key and secret is used to connect to AWS. Proper RDS and S3 permissions need to be associated to that access key. Details about how to configure such access key are given in the next sections.

Installation Methods

- *Amazon RDS Plugin Installation with BIM* (recommended)
- *Amazon RDS Plugin Installation with Package Managers*
- *Amazon RDS Plugin Manual Installation*

Amazon RDS Plugin Installation with BIM

The recommended way to install any Bacula component, including any daemon and any plugin, is using the Bacula Installation Manager (BIM).

Steps

1. Install File Daemon.
2. Select the amazon-rds key in the plugin selection step.

Result

Amazon RDS Plugin is installed. Click here for more details.

For more information, visit Linux: Bacula Enterprise Installation with BIM.

Amazon RDS Plugin Installation with Package Managers

Another way to install this plugin involves using the package manager of the used distribution.

In this instance, Debian Bullseye serves as an example of a base operative system. The process will be very similar in any version of Debian or any other Debian-based distribution (e.g., Ubuntu). In the case of using RPM-based distributions, like Red Hat Linux, Oracle Linux or Suse Linux, the primary distinction lies in switching to the appropriate package manager for that distribution, typically *yum*.

Steps

1. Add the repository file suitable for the existing customer subscription and the Debian version utilized. An example would be `/etc/apt/sources.list.d/bacula.list` with the following content:

Listing 263: **APT repositories**

```
# Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/
↪bullseye-64/ bullseye main
deb https://www.baculasystems.com/dl/@customer-string@/debs/amazon-rds/
↪@version@/bullseye-64/ bullseye amazon-rds
```

2. Run the apt update:

Listing 264: **APT update**

```
apt update
```

3. Install the plugin using:

Listing 265: **APT install**

```
apt install bacula-enterprise-amazon-rds-plugin
```

The plugin has two distinct packages that should be installed automatically with the command shown:

- `bacula-enterprise-amazon-rds-plugin`
- `bacula-enterprise-amazon-rds-plugin-libs`

Result

Amazon RDS Plugin is installed. Click here for more details.

Amazon RDS Plugin Manual Installation

Manual installation of the packages may be done after downloading the packages, and then using the package manager to install them.

1. Download the following packages from your Bacula Systems download area.
   - bacula-enterprise-amazon-rds-plugin
   - bacula-enterprise-amazon-rds-plugin-libs
2. After downloading the packages, install them with the command:

Listing 266: **Manual dpkg install**

```
dpkg -i bacula-enterprise-*
```

Result

Amazon RDS Plugin is installed. Click here for more details.

Result

The installation package includes the following components:

- Jar libraries in `/opt/bacula/lib` (such as `bacula-amazon-rds-plugin-x.x.x.jar` and `bacula-amazon-rds-plugin-libs-x.x.x.jar`). Note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a following message: "Jar version:X.X.X".
- A plugin connection file (`amazon-rds-fd.so`) in the plugins directory (usually `/opt/bacula/plugins`). Note that amazon-rds acronym refers to Amazon Elastic Compute Cloud.
- A backend file (`amazon-rds-backend`) that invokes the jar files in `/opt/bacula/bin`. This backend file searches for the most recent `bacula-amazon-rds-plugin-x.x.x.jar` file in order to launch it, even though there should generally be only one file present.

Once the plugin is installed, it should be possible to verify it loaded through a status client command in bconsole (`Plugin:` line must contain `amazon-rds`):

Listing 267: **Status client**

```
*st client
Automatically selected Client: 127.0.0.1-fd
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102

127.0.0.1-fd Version: 18.0.0 (20 Nov 2023)  x86_64-pc-linux-gnu ubuntu 22.04
Daemon started 14-abr-23 10:14. Jobs: run=2 running=0 max=100.
Ulimits: nofile=1024 memlock=2026356736 status=ok
Heap: heap=827,392 smbytes=436,939 max_bytes=5,100,087 bufs=153 max_bufs=248
Sizes: boffset_t=8 size_t=8 debug=600 trace=1 mode=1,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL 3.0.2 15 Mar 2022
APIs: !GPFS
Plugin: bpipe(2) amazon-rds(1.0.0)
```

Configuration

The following chapter presents the information on how to configure the plugin. Backup jobs with Bacula Enterprise Amazon RDS Plugin function similarly to other Bacula Enterprise backup jobs after the appropriate Fileset has been established and the access key with necessary permissions is provided.

In the following sections we present how to configure an AWS access key with the associated permissions to use this plugin, and then the parameters to set up a Fileset to invoke Amazon RDS Plugin through a job.

Restore parameters are discussed in the Restore section of *Operations chapter*.

Access Key Configuration

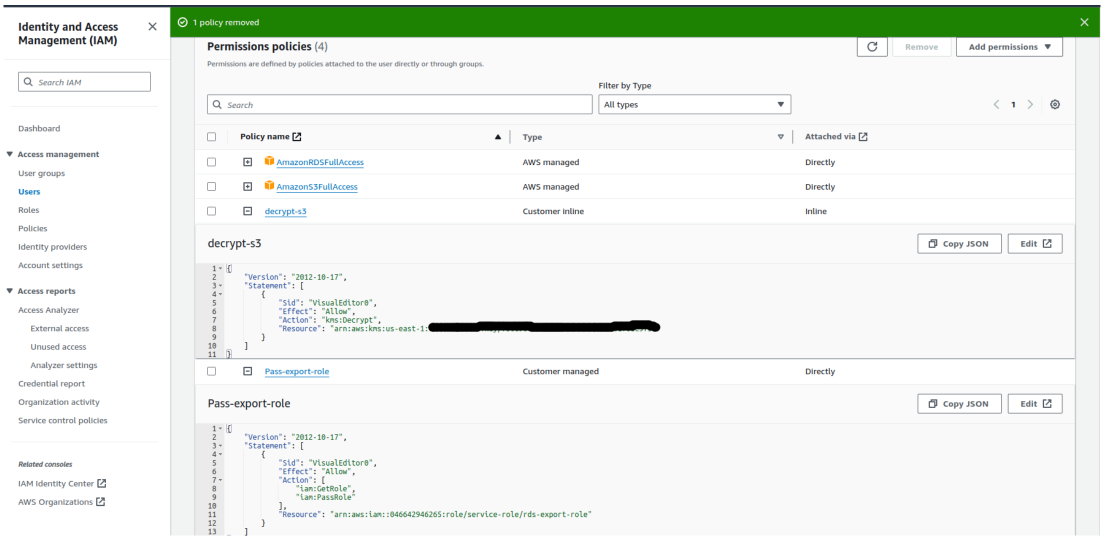
The utilization of Bacula Enterprise Amazon RDS Plugin requires the involvement of an AWS IAM service user who possesses a specific set of permissions outlined below.

Subsequently, the user must establish an access key, which should be configured as Fileset parameters. This configuration enables the plugin to effectively retrieve or write data during backup or restore operations.

The plugin needs the following set of permissions to work appropriately:

- Full RDS service permissions for the target instances or clusters
- Full S3 service permissions or permissions to fully manage the destination bucket to be used for the export operation, enabling snapshots to be exported to it and allowing data to be retrieved and sent to the SD
- The user needs to be associated with an export role that needs to be created beforehand (refer to the details below)
- The user needs to have decrypt permissions using a KMS Service Key that needs to be created in advance (refer to the details below)

As an example, the final set of permissions of the user should resemble the image provided below:



Permissions policies (4)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AmazonRDSFullAccess	AWS managed	Directly
AmazonS3FullAccess	AWS managed	Directly
decrypt-s3	Customer inline	Inline
Pass-export-role	Customer managed	Directly

decrypt-s3

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-1:[redacted]
        }
    ]
}
```

Pass-export-role

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::046642946265:role/service-role/rds-export-role"
        }
    ]
```

Fig. 122: RDS Permissions Summary for the user

Note that for export operations to succeed, all of the following parameters must be defined: `export_bucket`, `export_role`, `export_key`, `access_key`, `secret_key`.

Export Role

It is necessary to grant tasks write-access permission for the snapshot export tasks to the Amazon S3 bucket designated for exporting the backups. This can be accomplished in 3 steps with a user with permissions to manage policies and roles within the IAM service:

Listing 268: **Export Role Procedure**

```
# 1. Create a policy using the proper bucket name instead of EXAMPLE-BUCKET
$ aws iam create-policy  --policy-name ExportPolicy --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
         {
               "Sid": "ExportPolicy",
               "Effect": "Allow",
               "Action": [
                  "s3:PutObject*",
                  "s3:ListBucket",
                  "s3:GetObject*",
                  "s3:DeleteObject*",
                  "s3:GetBucketLocation"
               ],
               "Resource": [
                  "arn:aws:s3:::EXAMPLE-BUCKET",
                  "arn:aws:s3:::EXAMPLE-BUCKET/*"
               ]
         }
      ]
   }'

   # 2. Create a IAM role, so that Amazon RDS can assume this IAM role on␣
↪your behalf to access your Amazon S3 buckets.
   aws iam create-role  --role-name rds-s3-export-role  --assume-role-policy-
↪document '{
      "Version": "2012-10-17",
      "Statement": [
         {
            "Effect": "Allow",
            "Principal": {
               "Service": "export.rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
         }
      ]
      }'

   # 3. Attach the IAM policy to the just created role
   aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-
↪s3-export-role

   # 4. In the User Management screen, associate the created role to the user␣
↪that will contain the access key for Bacula Enterprise Amazon RDS Plugin
```

For more information, consult: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_

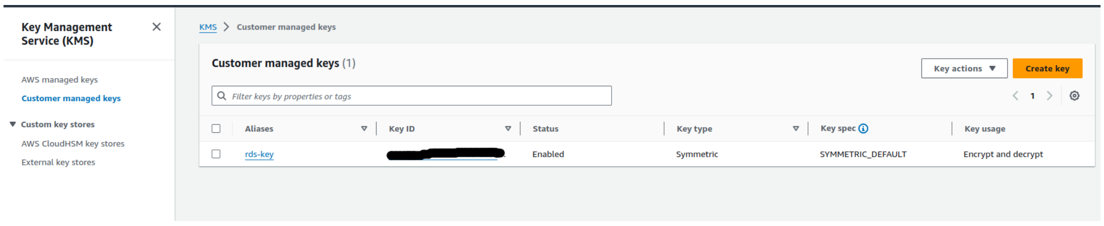
ExportSnapshot.html#USER_ExportSnapshot.SetupIAMRole

Encryption Key

Create a symmetric encryption AWS KMS key for the server-side encryption. The KMS key is used by the snapshot export task to set up AWS KMS server-side encryption when writing the export data to S3.

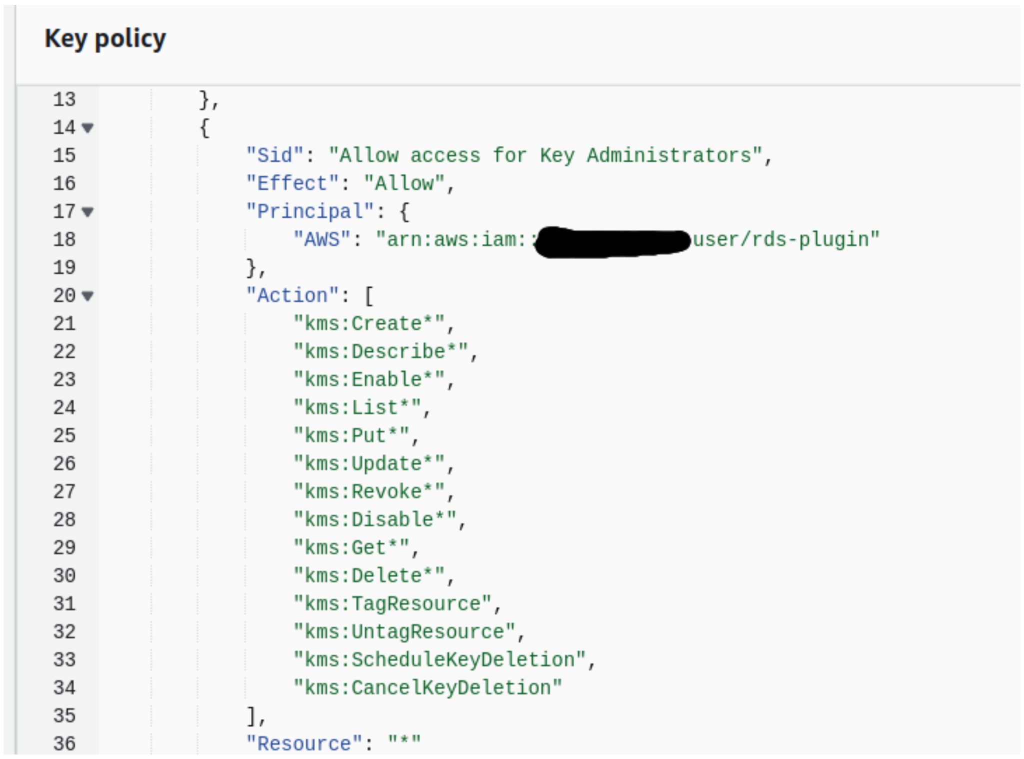
The KMS key policy must include both the kms:CreateGrant and kms:DescribeKey permissions. For more information on using KMS keys in Amazon RDS, visit AWS KMS key management official documentation.

Below is an example of a KMS-created key along with its policy.



Aliases	Key ID	Status	Key type	Key spec	Key usage
rds-key	[redacted]	Enabled	Symmetric	SYMMETRIC_DEFAULT	Encrypt and decrypt

Fig. 123: KMS Key



Key policy

```
    },
    {
        "Sid": "Allow access for Key Administrators",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam:[redacted]user/rds-plugin"
        },
        "Action": [
            "kms:Create*",
            "kms:Describe*",
            "kms:Enable*",
            "kms:List*",
            "kms:Put*",
            "kms:Update*",
            "kms:Revoke*",
            "kms:Disable*",
            "kms:Get*",
            "kms:Delete*",
            "kms:TagResource",
            "kms:UntagResource",
            "kms:ScheduleKeyDeletion",
            "kms:CancelKeyDeletion"
        ],
        "Resource": "*"
```

Fig. 124: KMS Key Policy

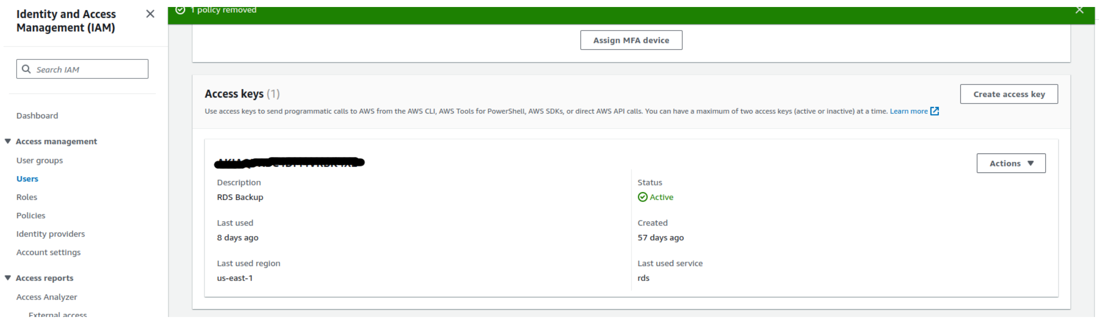
After creating the key, the IAM user employed for the Amazon RDS Plugin needs to reference the key with a policy similar to what was shown before, using the proper ARN. Below, the policy example from it:

Listing 269: **Policy for kms key in the user**

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-1:046642946265:key/xxxxxxx-
↪yyyy-zzzz-aaaa-bbbbbbbbbbbbbb"
      }
   ]
}
```

Access Key

After the user has been set up, it is essential to navigate to *access keys* and generate a new one:



Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

[redacted]

Description: RDS Backup
Status: Active
Last used: 8 days ago
Created: 57 days ago
Last used region: us-east-1
Last used service: rds

Fig. 125: Amazon RDS Access Key

The Id of the key and the associated secret are the parameters to configure in the plugin Fileset in `access_key` and `access_secret` parameters. Adding also the `region` parameter should be enough to allow the plugin to connect to the target dataset to protect.

See also:

- Go to Fileset Configuration

Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular Filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to invoke the Amazon RDS Plugin:

Listing 270: **Fileset RDS**

```
Fileset {
   Name = FS_RDS
   Include {
```

(continues on next page)

(continued from previous page)

```
      Options {
        signature = MD5
        ...
      }
      Plugin = "amazon-rds: <amazon-rds-parameter-1>=<amazon-rds-value-1>
↪<amazon-rds-parameter-2>=<amazon-rds-value-2> ..."
   }
}
```

It is **strongly recommended** to use only one `Plugin` line within each Fileset. The plugin offers the needed flexibility to combine various module backups within the same plugin line. In instances where multiple rds servers exist, it is essential to employ distinct Filesets and separate jobs.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ",".

Below, in the subsections, there are lists that present all the parameters you can use to control Amazon RDS Plugin behavior.

Fileset Connection Parameters

The following parameters control the connection of the Amazon RDS Plugin AWS.

Option	Require	Default	Values	Example	Description
access_	**Yes**		String: access key with proper permissions	AKIAIOS-FODNN7EXAMPL	Access key defined in the desired AWS account with access to the target instances and having the right permissions
secret_	**Yes**		String: secret key associated to provided access key	wJalrXUtn-FEMI/K7MDENG/b	Secret associated to the access key
region	No	eu-west-1	String: region code	eu-east-1	Existing region in AWS where the instances to backup reside

Fileset Backup Parameters

The following list of parameters controls what will be incorporated into the corresponding backup:

Option	Require	Default	Values	Example	Description
instanc	No		Valid database instance or cluster identifiers separated by ‘,’	myInstance1, myInstance2	Comma separated list of database instance or cluster identifiers to backup. If no instance is provided the plugin will list all of them and will backup them
instanc	No		Valid database instance or cluster identifiers separated by ‘,’	excludeExample, i-1233sablkbl	Comma separated list of database instance or cluster identifiers to exclude from backup. If this is the only parameter found for selection, all elements will be included and this list will be excluded
instanc	No		Valid regex	.*-includedSuf	Backup matching instances or clusters by its identifier. Please, only provide list parameters (instances + instances_exclude) or regex ones. But do not try to combine them
instanc	No		Valid regex	.*-excludedSuf	Exclude matching instances or clusters by its identifier. Please, only provide list parameters (instances + instances_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for user selection, all instances will be included and matching instances will be excluded
instanc	No		Tag keys or values (format: tagkey1=value	backup=yes	Backup instances or clusters containing the specified tags
instanc	No		Tag keys or values (format: tagkey1=value	backup=no	Exclude instances or clusters containing the specified tags
start_	No		0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Backup only the list of ebs volumes indicated by this parameter from the selected instances. If not specified, all disks from each instance will be backed up
snapshots_	No	60	Integer	100	Retention or number of snapshots to be kept in Amazon RDS Service. Older ones exceeding this number will be removed
export_	No		String representing an existing bucket in the region	mybucket	Name of the bucket to send the exported parquet files if export is enabled
export_	No		String representing a folder name	myfolder	Optional folder inside the bucket to store the exported files
export_	No		String representing a IAM Role name with the proper export permissions	arn:aws:iam role/rds-export-role	Name of the s3 export role needed to perform export operations
ex-	No		String rep-	75gt9800	ARN of the AWS Encryption key needed to per-

Note: When the `export_download = no` parameter is set, the backup is not sent to the Bacula storage, but there is still the metadata that is copied to the Bacula storage. This is necessary to restore the data backed up in the user's S3 bucket later.

Fileset Common Parameters

These parameters control the generic characteristics of the behavior of the Amazon RDS Plugin. You may also encounter these parameters in other Bacula Enterprise Plugins that produce similar effects, so you might already be acquainted with them.

Option	Required	Default	Values	Example	Description
abor	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
config_fi	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/ rds.settings	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset
log	No	/opt/bacula rds/amazon rds-debug.log	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/amazo rds.log	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/amazon-rds/amazon-rds-debug.log* files containing debut information which is more complete with a greater debug number
path	No	/opt/bacula	An existing path with enough permissions for File Daemon to create any internal (and usually temporary) plugin file	/mnt/my-vol/	Uses this path to store metadata and temporary files

This is an example of a `config_file` with the FilesetRDSConnectionParameters options:

```
# cat /opt/bacula/etc/amazon-rds.settings
region=us-east-1
```

(continues on next page)

(continued from previous page)

```
access_key=AKIAQTESTKEY12134g
secret_key=m23480ahpqwre894qwrffsfdSecretExample
```

Fileset Tuning Parameters

The following parameters can be utilized to adjust the plugin's behavior for increased flexibility in challenging network conditions, high job concurrency scenarios, and similar situations.

It is generally unnecessary to alter them in most situations.

Option	Require	Default	Values	Example	Description
concurrent_threads	No	500	1-1000	100	Number of maximum concurrent backup threads running in parallel in order to download blocks from a given EBS volume
api_timeout	No	9000	Positive integer (seconds)	2000	Total timeout for AWS API calls
api_read_timeo	No	600	Positive integer (seconds)	5000	Timeout for AWS API calls to respond
general_network_r	No	600	Positive integer (number of retries)	10	Number of retries for failed requests to the AWS API
general_network_d	No	50	Positive integer (seconds)	100	Delay between retries to the AWS API
snapshot_operations	No	1800	Positive integer (seconds)	3600	Timeout for snapshot related operations (like make snapshot) before giving up
export_operations	No	1800	Positive integer (seconds)	3600	Timeout for export operations before giving up
instance_operatio	No	1800	Positive integer (seconds)	3600	Timeout for instance related operations (like waiting for them to be in an operational status) before giving up

Fileset Advanced Parameters

The following parameters are advanced ones, and they should not be modified in the great majority of cases:

Option	Re-quire	Default	Values	Ex-am-ple	Description
stream_sl	No	1	Positive integer (1/10 sec-conds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_n	No	120	Positive integer (sec-onds)	360	Max wait time for FD to answer packet requests
time_max	No	86400	Positive integer (sec-onds)	4320	Maximum time to wait to overwrite a debug log that was marked as being used by other process
log-ging_max	No	50MB	String size	300M	Maximum size of a single debug log fileGenerates the working/amazon-rds/amazon-rds-debug.log* files containing debut information which is more complete with a greater debug number
log-ging_max	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rollin	No	amazon-rds.log.%d{ MMM}.log.	String file name pattern	...	Log pattern for rotated log files
split_conf	No	=	Charac-ter	:	Character to be used in config_file parameter as separator for keys and values
pub-lisher_qu	No	1200	Positive integer (sec-onds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The `.log` files are rotated automatically. Currently, each file can be 50Mb at maximum and the plugin will keep 25 files.
  - This behavior can be changed using the internal advanced parameters: `logging_max_file_size` and `logging_max_backup_index`.
- The `.err` file can show contents even if no actual error has occurred during the jobs. It can also present contents even if debugging is disabled. This file is not rotated, but it is generally anticipated to remain small in size. If you still need to rotate it, you can include it in a general rotating tool such as `logrotate`.
- Backups in parallel and also failed backups will generate several log files. For example: amazon-rds-debug-0.log, amazon-rds-debug-1.log, and so on.

Fileset Examples

In this section, some Fileset examples are presented:

Listing 271: **Fileset: for a selection of instances by name, just snapshot**

```
Fileset {
   Name = fs-amazon-rds-instances-a-b
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: region=us-east-1 access_key=AKIAQTESTKEY12134g␣
↪secret_key=m23480ahpqwre894qwrffsfdSecretExample instances=myInstanceA,
↪myInstanceB"
   }
}
```

Listing 272: **Fileset: using a config file**

```
Fileset {
   Name = fs-amazon-rds-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances=myInstanceA "
   }
}

Config file contents in stored in the same File Daemon host in /opt/bacula/
↪etc/amazon-rds.settings:
region=us-east-1
access_key=AKIAQTESTKEY12134g
secret_key=m23480ahpqwre894qwrffsfdSecretExample
```

Listing 273: **Fileset: Export and download the database**

```
Fileset {
   Name = fs-amazon-rds-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances=myInstanceA export_bucket=myExportBucket export_
↪role=arn:aws:iam::046642946265:role/service-role/rds-export-role export_
↪key=75gt9800-719e-44b1-63h0-4197d0gt9015"
   }
}
```

Listing 274: **Fileset: Backup instances contaning tag bacula**

```
Fileset {
   Name = fs-amazon-rds-instances
   Include {
```

(continues on next page)

(continued from previous page)

```
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances_tags=\"bacula\""
   }
}
```

Listing 275: **Fileset: Backup instances contaning tag backup=yes**

```
Fileset {
   Name = fs-amazon-rds-instances
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances_tags=\"backup=yes\""
   }
}
```

Listing 276: **Fileset: Tweak snapshot retention and start instances if they are offline**

```
Fileset {
   Name = fs-amazon-rds-instance-a
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances=myInstanceA start_instances=yes snapshot_retention=100"
   }
}
```

Listing 277: **Fileset: by tag, but exclude A**

```
Fileset {
   Name = fs-amazon-rds-instances
   Include {
      Options { signature = MD5 }
      Plugin = "amazon-rds: config_file=/opt/bacula/etc/amazon-rds.settings␣
↪instances_tags=\"backup=yes\" instances_exclude=myInstanceA"
   }
}
```

Best Practices

The following article presents best practices regarding jobs distribution, concurrency and performance.

Jobs Distribution

It is recommended to split the target backup between different instances or clusters, ideally having one job per instance or cluster.

Following this recommendation, errors in a single job will not compromise the whole backup cycle, allowing for successful backups of certain instances or clusters, even if others encounter issues. This also makes it easier to identify the cause of any error.

Concurrency

When using Amazon RDS APIs, it is possible to find a variety of boundaries that need to be considered. We highlight some of them below:

- Amazon RDS limits: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Limits.html
- Capabilities of the host serving the RDS Service
- Service usage during the backup window

If a boundary is crossed, the corresponding request will usually fail. Bacula Amazon RDS Plugin is prepared to wait for a certain amount of time and then retry it, thus offering a degree of resiliency. However, it is crucial to plan an adequate strategy to backup all the elements without frequently approaching any boundaries. This entails managing the number of concurrent requests made during the backup window.

The recommended strategy to backup a new environment is to plan a step-by-step testing scenario prior to deployment, where the number of instances and the concurrency of the jobs are increased progressively. Another important aspect is the timing schedule, as some boundaries are related to time-frames (i.e., the number of request per time unit). If you detect you are reaching boundaries when running all your backups during a single day of the week, try to increase the time window and distribute the load throughout it in order to enhance performance results.

Note that from an architectural and AWS service point of view, you can also consider to:

- Run your File Daemon directly in the cloud (if your SD is also in the cloud)
- Run your Storage Daemon and File Daemon in the same host, so you skip one network hop in the process (recommended)
- Use a dedicated AWS connection (https://aws.amazon.com/directconnect/)

Performance

The performance of the plugin is highly dependent on various external factors:

- Latency and bandwidth to AWS
- Network infrastructure
- FD Host hardware
- FD Load

- Ratio number of elements/size
- And many more.

In short, it is not possible to establish an exact reference for the duration required to complete a backup.

Typically, backups that do not use the export and download functions will proceed as quickly as AWS permits However, when using the export and download functions, the aforementioned factors become increasingly pertinent, as certain files will be created in S3 that must subsequently be downloaded.

As a guideline concerning the number of records per table:

- Numerous small tables to protect: More files (one file per table) per time period, but smaller speed (MB/s).
- Large tables to protect: Fewer files (one file per table) per time period, but higher speed (MB/s).

It is also important to note that the snapshot cleanup process requires some time to finalize, in addition to the export operation itself.

It is recommended to benchmark your own environment based on your specific requirements and needs.

There are various strategies available to use this plugin, so it is recommended to evaluate which option best meets your needs prior to deploying the jobs across your entire environment, ensuring optimal results:

- You can have a job per instance or cluster (recommended)
- You can have multiple instances per job or even all your instances or clusters in the same job (not recommended)
- You can split your workload through a schedule, or try to run all your jobs together
- You can specifically select the instances you want to backup or backup them all
- You can specifically select the clusters you want to backup or backup them all
- You can specifically select the tables you want to export and backup or process them all
- You can run your File Daemon on premise or in the cloud
- You can use default internet connection to AWS or use a dedicated AWS connection (https://aws.amazon.com/directconnect/)
- You can run your Storage Daemon and File Daemon in the same host, so you skip one network hop in the process (recommended).

Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise Amazon RDS Plugin**.

Backup

The overall backup operation involves the following steps:

1. Identify all selected target database instances or clusters.
2. Trigger a snapshot of the selected targets. This snapshot is inherently incremental compared to the previous one (if applicable).
3. If export is enabled, trigger an export operation of the selected tables to S3 (to a bucket configured in the plugin).
4. If download is enabled, download the exported files and store them in Bacula (send them to the Storage Daemon).
5. If export delete option is enabled, delete the exported files from the S3 bucket.
6. Remove old snapshots according to the established snapshot retention policy (which also includes deleting associated export files if the `export_delete` is enabled).

Backup File Structure

A single database backup with export and download enabled will produce contents in the backup similar to the following ones:

Listing 278: **Files protected with the backup**

```
$ dir
----------   0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/
-rw-r-----   1 nobody   nogroup         645  2024-06-06 14:38:30  /@amazon-
↪rds/jg-pg-aws/export_info_plugintest-20240606-135513-03.json
-rw-r-----   1 nobody   nogroup       72801  2024-06-06 14:38:30  /@amazon-
↪rds/jg-pg-aws/export_tables_info_plugintest-20240606-135513-03_from_1_to_38.
↪json
-rw-r-----   1 nobody   nogroup        3192  2024-03-28 11:27:41  /@amazon-
↪rds/jg-pg-aws/jg-pg-aws.i.json
-rw-r-----   1 nobody   nogroup        1103  1970-01-01 00:59:59  /@amazon-
↪rds/jg-pg-aws/plugintest-20240606-135513-03.isnap.json

$ pwd
cwd is: /@amazon-rds/jg-pg-aws/

$ cd bacula
cwd is: /@amazon-rds/jg-pg-aws/bacula/

$ dir
----------   0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.basefiles/
----------   0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.cdimages/
----------   0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.client/
----------   0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.counters/
```

(continues on next page)

(continued from previous page)

```
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.device/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.events/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.file/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.fileevents/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.filemedia/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.fileset/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.job/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.jobhisto/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.jobmedia/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.location/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.locationlog/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.log/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.malwaremd5/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.malwaresha256/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.media/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.mediatype/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.metaattachment/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.metaemail/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.object/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.path/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.pathhierarchy/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.pathvisibility/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.pool/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.restoreobject/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.snapshot/
----------   0 root     root            0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.status/
```

(continues on next page)

(continued from previous page)

```
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.storage/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.tagclient/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.tagjob/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.tagmedia/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.tagobject/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.unsavedfiles/
----------    0 root     root              0  1970-01-01 01:00:00  /@amazon-
↪rds/jg-pg-aws/bacula/public.version/

cd public.status/1
cwd is: /@amazon-rds/jg-pg-aws/bacula/public.status/1/

$ dir
-rw-r-----    1 nobody   nogroup        1421  2024-06-06 14:36:49  /@amazon-
↪rds/jg-pg-aws/bacula/public.status/1/part-00000-e8ed76c2-8a29-49c7-9e0f-
↪771a666ebe74-c000.gz.parquet
```

The JSON files represent information about the database instance, the snapshot and the export operation. Additionally, there exists a directory for each table of the database, wherein one can find an Apache Parquet file containing the data of that specific table.

Snapshot and Exports Livecycle

The Bacula Enterprise Amazon RDS Plugin controls the whole livecycle of the snapshots created within the Amazon Web Services Cloud, as well as the livecycle of the data exported to S3.

Amazon RDS Snapshots are actually Amazon EBS Snapshots containing also certain transaction logs. The EBS layer is transparent to the user and cannot be directly controlled.

The database snapshots are sent to an S3 bucket, which is also not directly controllable. When an RDS instance or cluster snapshot is created, Amazon automatically manages these details in the background, creating the appropriate EBS device snapshot and storing the data in S3. Similarly, when a snapshot is removed from RDS, the corresponding data in S3 will be automatically removed.

Bacula Enterprise Amazon RDS Plugin will remove exported files from S3 after each backup if the `export_delete` flag is enabled.

Snapshots that exceed the `snapshots_retention` value will be deleted. Their associated exports, if they still exist, will also be deleted if `export_delete` flag is enabled.

Restore

Restore Procedure

Bacula Enterprise Amazon RDS Plugin offers four distinct restore modes:

1. Generate a new instance from a given snapshot (from the backup or from an arbitrary snapshot identifier).
2. Generate a new instance with a specific state using Point-In-Time Recovery.
3. Restore metadata or Apache Parquet exported data locally to the filesystem for subsequent processing.
4. Restore Apache Parquet information into a running MySQL or PostgreSQL database.

Depending on the desired result, the restore operation should be parametrized differently.

Restore Parameters

Amazon RDS Plugin is able to perform a raw restore to any local filesystem mounted over the host where the File Daemon operates, or directly to the Amazon RDS environment. The restore method is selected determined by the value of the `where` parameter at restore time:

- Empty or '/' (example: `where=/`) -> Amazon RDS restore will be triggered
- Any other path for where (example: `where=/tmp`) -> Raw restore to the local file system will be triggered.

This principle generally applies to other Bacula plugins as well. Nevertheless, in this specific plugin, it is advised against using the raw restore method by the File Daemon. This recommendation stems from the fact that the raw restore method retains the headers generated during the backup process in the volume disk files. If the intention is to restore the filesystem, one can trigger an Amazon RDS restore using the `where=/` parameter. It is important to also use the restore variable `to_local_path` and specify the desired destination. By doing so, the mentioned headers will be automatically removed, resulting in a disk image that is appropriate for any future purposes.

When using the Amazon RDS restore method, the following parameters are available to control the restore behavior within the "Plugin Options" menu during a BConsole restore session:

Option	Required	Default	Values	Example	Description
instance	No	Generated from Restore job name	String for the new instance or cluster that will be generated with the restore. Only a-zA-Z or '-' characters are allowed	MyRestoredInstance	Set the identifier for a new restored database instance or cluster. If none is provided and the restore operation needs to create a new instance or cluster, the name will be generated from the restore job
from_	No		Existing snapshot identifier in RDS service	my-snap-2024-05-16-12-30	Set a snapshot identifier, existing in AWS to restore an instance or cluster directly from it. If not specified, the identifier will be get from the backup information
instance	No	Original backup value	Accepted instance type in Amazon RDS. Defined in: https://aws.amazon.com/rds/instance-types/	c7g.medium	Set database instance class
availability_zo	No	Original backup value	String: availability zone existing in: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZo html	us-east-1a	Set the destination availability zone for the instance(s) and its/their volume(s)
parameter_gr	No	Original backup value	String: Name of the parameter group	mygroup	Set the name(s) of the parameter groups that will be applied to the restored database instance (separated by ',')
vpc_s	No	Original backup value	String: Name of the security group id	sg-03237oe449	Set the id(s) of the VPC Security Groups that will be applied to the restored database instance (separated by ',')
tags	No		List of key value strings: key1=value1, key2=value2		Set tags to the restored instance(s) (tag1key:tag1value,tag2key:tag2value…)
multi_	No	Original backup value	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	yes	Configure the restored database instance as Multi AZ
db_po	No	Original backup value	Integer	2140	Set the database port
storage_ty	No	Original backup value	Accepted volume type in Amazon EBS. Defined in: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html		Set the instance storage type
storage_th	No	Original backup value	Positive integer		Set the instance storage throughput
iops	No	Original backup value	Positive integer		Iops value for instance storage

Restore Use Cases

The following restore scenarios are supported, and steps to execute them are described:

a) Generate a new instance or cluster from a the snapshot that was taken during the selected backup

   1. Run a restore session selecting appropriate backup jobs
   2. Select all the contents of the backup
   3. Use `Where=/`

b) Generate a new instance or cluster from an arbitrary snapshot that is existing in Amazon RDS, and configure a specific new instance id:

   - Follow previously described 'a' scenario.
   - Set `instance_identifier` with the value of the desired new name
   - Set `from_snapshot_id` with the value of the desired previous snapshots

c) Restore an instance in Point-In-Time Recovery mode:

- Follow previously described 'a' scenario
- Before confirming the restore operation, set `pitr_from_instance_id` with the desired instance or cluster to use, or let the restore used the instance from the backup job selected
- You may want to run a `.query` command (parameter=instance) to check the last available restore time for your instance
- Set `pitr_date` with the desired value, or set `pitr_latest` to yes.

d) Restore an instance to Amazon RDS, but adjust any configuration parameter of it (advanced):

- Follow previously described 'a', 'b' or 'c' scenarios.
- Adjust any desired parameter from the following list: `instance_class`, `availability_zone`, `parameter_groups`, `vpc_security_groups_ids`, `tags`, `multi_az`, `db_port`, `storage_type`, `storage_throughput`, `iops`
- Note that the configuration you set here will be applied as entered. Therefore, the consistency of that configuration will depend on all elements being correct (existing and consistent) at Amazon RDS side for your particular infrastructure

e) Restore an instance to Amazon RDS, but to a different location:

- Follow previously described 'a', 'b', 'c' or 'd' scenarios.
- Adjust `region`, `access_key`, `secret_key` with the destination values

f) Restore instance Apache Parquet files to a running database in the File Daemon host:

- Follow previously described 'a' scenario.
- Adjust database connection with 'parquet_xxx' parameters
- By default, if no host is established, a socket connection will be attempted. If a host is specified, the connection will use TCP/IP
- Pay attention to the `parquet_save_mode`, which will allow you to control how the restore should behave in terms of finding previous values in your database

g) Restore instance files or volume files to a local directory:

   1. Run a restore session selecting appropriate backup jobs

2. Select the desired files (or all of them) inside a given `i-xxxxxxx/` folder
3. Use `Where=/`
4. Adjust `to_local_path` to the desired path

Restore Example Session

Note: It is also possible to run backup or restore operations from any of the Bacula Graphical User Interfaces.

Listing 279: **Restore bconsole session**

```
restore jobid=1 Client=127.0.0.1-fd where="/"
Using Catalog "MyCatalog"
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
41 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd "/@amazon_rds/"
Invalid path given.
cwd is: /
$ dir
----------   0 root     root               0  1970-01-01 01:00:00  /@amazon-
↪rds/
$ estimate
41 total files; 0 marked to be restored; 0 bytes.
$ mark *
41 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.3.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

   TEST-2024-06-06:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


41 files selected to be restored.

Using Catalog "MyCatalog"
```

(continues on next page)

(continued from previous page)

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.3.bsr
Where:           /
Replace:         Always
FileSet:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2024-06-06 14:39:13
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: FileSet
   5: Restore Client
   6: When
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : amazon-rds: region="us-east-1" access_key=
↪"AKIAQVXBC4DM4VRBK4XL" secret_key="hqvQac/ZikF6+E9AGDOwGjWPMyrz0K6zAC1eQ9KL
↪" instances="jg-pg-aws" start_instances=yes snapshots_retention=2 export_
↪bucket=j-bacula-rds export_role=arn:aws:iam::046642946265:role/service-role/
↪rds-export-role export_key=73ed9600-749e-47b5-93f0-6761f0ac9734 export_
↪delete=yes debug=6
Plugin Restore Options
Option                        Current Value        Default Value
instance_identifier:          *None*               (*none*)
from_snapshot_id:             *None*               (*none*)
instance_class:               *None*               (*none*)
availability_zone:            *None*               (*none*)
parameter_groups:             *None*               (*none*)
vpc_security_groups_ids:      *None*               (*none*)
tags:                         *None*               (*none*)
multi_az:                     *None*               (*none*)
db_port:                      *None*               (*none*)
storage_type:                 *None*               (*none*)
storage_throughput:           *None*               (*none*)
iops:                         *None*               (*none*)
pitr_from_instance_id:        *None*               (*none*)
pitr_date:                    *None*               (*none*)
pitr_latest:                  *None*               (*none*)
```

(continues on next page)

(continued from previous page)

```
parquet_db_socket:              *None*              (*none*)
parquet_db_engine:              *None*              (*none*)
parquet_db_host:                *None*              (*none*)
parquet_db_name:                *None*              (*none*)
parquet_db_port:                *None*              (*none*)
parquet_db_user:                *None*              (*none*)
parquet_db_password:            *None*              (*none*)
parquet_save_mode:              *None*              (*none*)
access_key:                     *None*              (*none*)
secret_key:                     *None*              (*none*)
region:                         *None*              (*none*)
debug:                          *None*              (*none*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_identifier (Set the identifier for a new restored database␣
↪instance or cluster)
   2: from_snapshot_id (Set a snapshot identifier, exisiting in AWS to␣
↪restore an instance or cluster directly from it)
   3: instance_class (Set database instance class)
   4: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   5: parameter_groups (Set the name(s) of the parameter groups that will be␣
↪applied to the restored database instance (separated by ','))
   6: vpc_security_groups_ids (Set the id(s) of the VPC Security Groups that␣
↪will be applied to the restored database instance (separated by ','))
   7: tags (Set tags to the restored instance(s) (tag1key:tag1value,
↪tag2key:tag2value...))
   8: multi_az (Configure the restored database instance as Multi AZ)
   9: db_port (Set the database port)
   10: storage_type (Set the instance storage type)
   11: storage_throughput (Set the instance storage throughput)
   12: iops (Iops value for instance storage)
   13: pitr_from_instance_id (Set the instance identifier that will reference␣
↪the instance to be used to proceed with a Point In Time Recovery restore)
   14: pitr_date (Sets the date for Point in Time Recovery restore)
   15: pitr_latest (Run a Point in Time Recovery restore using the latest␣
↪date available)
   16: parquet_db_socket (Set the socket path to use a socket based␣
↪connection to restore parquet data from the backup directly into a database)
   17: parquet_db_engine (Set the engine (postgresql or mysql) to restore␣
↪parquet data from the backup directly into a database)
   18: parquet_db_host (Set the host to connect and restore parquet data from␣
↪the backup directly into a database)
   19: parquet_db_name (Set the database name to connect and restore parquet␣
↪data from the backup directly into a database)
   20: parquet_db_port (Set the database port to connect and restore parquet␣
↪data from the backup directly into a database)
   21: parquet_db_user (Set the database user to connect and restore parquet␣
↪data from the backup directly into a database)
   22: parquet_db_password (Set the database password to connect and restore␣
↪parquet data from the backup directly into a database)
   23: parquet_save_mode (Set the restore mode for a parquet restore: Append,␣
```

(continues on next page)

(continued from previous page)

```
↪Overwrite, ErrorIfExists or Ignore)
   24: access_key (Set a different access key to access to the destination)
   25: secret_key (Set a different secret key to access to the destination)
   26: region (Set the destination region)
   27: debug (Change debug level)
Select parameter to modify (1-27): 17
Please enter a value for parquet_db_engine: mysql
Plugin Restore Options
Option                         Current Value        Default Value
instance_identifier:           *None*               (*none*)
from_snapshot_id:              *None*               (*none*)
instance_class:                *None*               (*none*)
availability_zone:             *None*               (*none*)
parameter_groups:              *None*               (*none*)
vpc_security_groups_ids:       *None*               (*none*)
tags:                          *None*               (*none*)
multi_az:                      *None*               (*none*)
db_port:                       *None*               (*none*)
storage_type:                  *None*               (*none*)
storage_throughput:            *None*               (*none*)
iops:                          *None*               (*none*)
pitr_from_instance_id:         *None*               (*none*)
pitr_date:                     *None*               (*none*)
pitr_latest:                   *None*               (*none*)
parquet_db_socket:             *None*               (*none*)
parquet_db_engine:             mysql                (*none*)
parquet_db_host:               *None*               (*none*)
parquet_db_name:               *None*               (*none*)
parquet_db_port:               *None*               (*none*)
parquet_db_user:               *None*               (*none*)
parquet_db_password:           *None*               (*none*)
parquet_save_mode:             *None*               (*none*)
access_key:                    *None*               (*none*)
secret_key:                    *None*               (*none*)
region:                        *None*               (*none*)
debug:                         *None*               (*none*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_identifier (Set the identifier for a new restored database␣
↪instance or cluster)
   2: from_snapshot_id (Set a snapshot identifier, exisiting in AWS to␣
↪restore an instance or cluster directly from it)
   3: instance_class (Set database instance class)
   4: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   5: parameter_groups (Set the name(s) of the parameter groups that will be␣
↪applied to the restored database instance (separated by ','))
   6: vpc_security_groups_ids (Set the id(s) of the VPC Security Groups that␣
↪will be applied to the restored database instance (separated by ','))
   7: tags (Set tags to the restored instance(s) (tag1key:tag1value,
↪tag2key:tag2value...))
   8: multi_az (Configure the restored database instance as Multi AZ)
```

(continues on next page)

(continued from previous page)

```
     9: db_port (Set the database port)
     10: storage_type (Set the instance storage type)
     11: storage_throughput (Set the instance storage throughput)
     12: iops (Iops value for instance storage)
     13: pitr_from_instance_id (Set the instance identifier that will reference␣
↪the instance to be used to proceed with a Point In Time Recovery restore)
     14: pitr_date (Sets the date for Point in Time Recovery restore)
     15: pitr_latest (Run a Point in Time Recovery restore using the latest␣
↪date available)
     16: parquet_db_socket (Set the socket path to use a socket based␣
↪connection to restore parquet data from the backup directly into a database)
     17: parquet_db_engine (Set the engine (postgresql or mysql) to restore␣
↪parquet data from the backup directly into a database)
     18: parquet_db_host (Set the host to connect and restore parquet data from␣
↪the backup directly into a database)
     19: parquet_db_name (Set the database name to connect and restore parquet␣
↪data from the backup directly into a database)
     20: parquet_db_port (Set the database port to connect and restore parquet␣
↪data from the backup directly into a database)
     21: parquet_db_user (Set the database user to connect and restore parquet␣
↪data from the backup directly into a database)
     22: parquet_db_password (Set the database password to connect and restore␣
↪parquet data from the backup directly into a database)
     23: parquet_save_mode (Set the restore mode for a parquet restore: Append,␣
↪Overwrite, ErrorIfExists or Ignore)
     24: access_key (Set a different access key to access to the destination)
     25: secret_key (Set a different secret key to access to the destination)
     26: region (Set the destination region)
     27: debug (Change debug level)
Select parameter to modify (1-27): 19
Please enter a value for parquet_db_name: test
Plugin Restore Options
Option                         Current Value        Default Value
instance_identifier:           *None*               (*none*)
from_snapshot_id:              *None*               (*none*)
instance_class:                *None*               (*none*)
availability_zone:             *None*               (*none*)
parameter_groups:              *None*               (*none*)
vpc_security_groups_ids:       *None*               (*none*)
tags:                          *None*               (*none*)
multi_az:                      *None*               (*none*)
db_port:                       *None*               (*none*)
storage_type:                  *None*               (*none*)
storage_throughput:            *None*               (*none*)
iops:                          *None*               (*none*)
pitr_from_instance_id:         *None*               (*none*)
pitr_date:                     *None*               (*none*)
pitr_latest:                   *None*               (*none*)
parquet_db_socket:             *None*               (*none*)
parquet_db_engine:             mysql                (*none*)
parquet_db_host:               *None*               (*none*)
parquet_db_name:               test                 (*none*)
```

(continues on next page)

(continued from previous page)

```
parquet_db_port:              *None*              (*none*)
parquet_db_user:              *None*              (*none*)
parquet_db_password:          *None*              (*none*)
parquet_save_mode:            *None*              (*none*)
access_key:                   *None*              (*none*)
secret_key:                   *None*              (*none*)
region:                       *None*              (*none*)
debug:                        *None*              (*none*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_identifier (Set the identifier for a new restored database␣
↪instance or cluster)
   2: from_snapshot_id (Set a snapshot identifier, exisiting in AWS to␣
↪restore an instance or cluster directly from it)
   3: instance_class (Set database instance class)
   4: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   5: parameter_groups (Set the name(s) of the parameter groups that will be␣
↪applied to the restored database instance (separated by ','))
   6: vpc_security_groups_ids (Set the id(s) of the VPC Security Groups that␣
↪will be applied to the restored database instance (separated by ','))
   7: tags (Set tags to the restored instance(s) (tag1key:tag1value,
↪tag2key:tag2value...))
   8: multi_az (Configure the restored database instance as Multi AZ)
   9: db_port (Set the database port)
   10: storage_type (Set the instance storage type)
   11: storage_throughput (Set the instance storage throughput)
   12: iops (Iops value for instance storage)
   13: pitr_from_instance_id (Set the instance identifier that will reference␣
↪the instance to be used to proceed with a Point In Time Recovery restore)
   14: pitr_date (Sets the date for Point in Time Recovery restore)
   15: pitr_latest (Run a Point in Time Recovery restore using the latest␣
↪date available)
   16: parquet_db_socket (Set the socket path to use a socket based␣
↪connection to restore parquet data from the backup directly into a database)
   17: parquet_db_engine (Set the engine (postgresql or mysql) to restore␣
↪parquet data from the backup directly into a database)
   18: parquet_db_host (Set the host to connect and restore parquet data from␣
↪the backup directly into a database)
   19: parquet_db_name (Set the database name to connect and restore parquet␣
↪data from the backup directly into a database)
   20: parquet_db_port (Set the database port to connect and restore parquet␣
↪data from the backup directly into a database)
   21: parquet_db_user (Set the database user to connect and restore parquet␣
↪data from the backup directly into a database)
   22: parquet_db_password (Set the database password to connect and restore␣
↪parquet data from the backup directly into a database)
   23: parquet_save_mode (Set the restore mode for a parquet restore: Append,␣
↪Overwrite, ErrorIfExists or Ignore)
   24: access_key (Set a different access key to access to the destination)
   25: secret_key (Set a different secret key to access to the destination)
   26: region (Set the destination region)
```

(continues on next page)

(continued from previous page)

```
   27: debug (Change debug level)
Select parameter to modify (1-27): 21
Please enter a value for parquet_db_user: root
Plugin Restore Options
Option                         Current Value        Default Value
instance_identifier:           *None*               (*none*)
from_snapshot_id:              *None*               (*none*)
instance_class:                *None*               (*none*)
availability_zone:             *None*               (*none*)
parameter_groups:              *None*               (*none*)
vpc_security_groups_ids:       *None*               (*none*)
tags:                          *None*               (*none*)
multi_az:                      *None*               (*none*)
db_port:                       *None*               (*none*)
storage_type:                  *None*               (*none*)
storage_throughput:            *None*               (*none*)
iops:                          *None*               (*none*)
pitr_from_instance_id:         *None*               (*none*)
pitr_date:                     *None*               (*none*)
pitr_latest:                   *None*               (*none*)
parquet_db_socket:             *None*               (*none*)
parquet_db_engine:             mysql                (*none*)
parquet_db_host:               *None*               (*none*)
parquet_db_name:               test                 (*none*)
parquet_db_port:               *None*               (*none*)
parquet_db_user:               root                 (*none*)
parquet_db_password:           *None*               (*none*)
parquet_save_mode:             *None*               (*none*)
access_key:                    *None*               (*none*)
secret_key:                    *None*               (*none*)
region:                        *None*               (*none*)
debug:                         *None*               (*none*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_identifier (Set the identifier for a new restored database␣
↪instance or cluster)
   2: from_snapshot_id (Set a snapshot identifier, exisiting in AWS to␣
↪restore an instance or cluster directly from it)
   3: instance_class (Set database instance class)
   4: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   5: parameter_groups (Set the name(s) of the parameter groups that will be␣
↪applied to the restored database instance (separated by ','))
   6: vpc_security_groups_ids (Set the id(s) of the VPC Security Groups that␣
↪will be applied to the restored database instance (separated by ','))
   7: tags (Set tags to the restored instance(s) (tag1key:tag1value,
↪tag2key:tag2value...))
   8: multi_az (Configure the restored database instance as Multi AZ)
   9: db_port (Set the database port)
   10: storage_type (Set the instance storage type)
   11: storage_throughput (Set the instance storage throughput)
   12: iops (Iops value for instance storage)
```

(continues on next page)

(continued from previous page)

```
   13: pitr_from_instance_id (Set the instance identifier that will reference␣
↪the instance to be used to proceed with a Point In Time Recovery restore)
   14: pitr_date (Sets the date for Point in Time Recovery restore)
   15: pitr_latest (Run a Point in Time Recovery restore using the latest␣
↪date available)
   16: parquet_db_socket (Set the socket path to use a socket based␣
↪connection to restore parquet data from the backup directly into a database)
   17: parquet_db_engine (Set the engine (postgresql or mysql) to restore␣
↪parquet data from the backup directly into a database)
   18: parquet_db_host (Set the host to connect and restore parquet data from␣
↪the backup directly into a database)
   19: parquet_db_name (Set the database name to connect and restore parquet␣
↪data from the backup directly into a database)
   20: parquet_db_port (Set the database port to connect and restore parquet␣
↪data from the backup directly into a database)
   21: parquet_db_user (Set the database user to connect and restore parquet␣
↪data from the backup directly into a database)
   22: parquet_db_password (Set the database password to connect and restore␣
↪parquet data from the backup directly into a database)
   23: parquet_save_mode (Set the restore mode for a parquet restore: Append,␣
↪Overwrite, ErrorIfExists or Ignore)
   24: access_key (Set a different access key to access to the destination)
   25: secret_key (Set a different secret key to access to the destination)
   26: region (Set the destination region)
   27: debug (Change debug level)
Select parameter to modify (1-27): 22
Please enter a value for parquet_db_password: root
Plugin Restore Options
Option                         Current Value        Default Value
instance_identifier:           *None*               (*none*)
from_snapshot_id:              *None*               (*none*)
instance_class:                *None*               (*none*)
availability_zone:             *None*               (*none*)
parameter_groups:              *None*               (*none*)
vpc_security_groups_ids:       *None*               (*none*)
tags:                          *None*               (*none*)
multi_az:                      *None*               (*none*)
db_port:                       *None*               (*none*)
storage_type:                  *None*               (*none*)
storage_throughput:            *None*               (*none*)
iops:                          *None*               (*none*)
pitr_from_instance_id:         *None*               (*none*)
pitr_date:                     *None*               (*none*)
pitr_latest:                   *None*               (*none*)
parquet_db_socket:             *None*               (*none*)
parquet_db_engine:             mysql                (*none*)
parquet_db_host:               *None*               (*none*)
parquet_db_name:               test                 (*none*)
parquet_db_port:               *None*               (*none*)
parquet_db_user:               root                 (*none*)
parquet_db_password:           root                 (*none*)
parquet_save_mode:             *None*               (*none*)
```

(continues on next page)

(continued from previous page)

```
access_key:                    *None*              (*none*)
secret_key:                    *None*              (*none*)
region:                        *None*              (*none*)
debug:                         *None*              (*none*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: instance_identifier (Set the identifier for a new restored database␣
↪instance or cluster)
   2: from_snapshot_id (Set a snapshot identifier, exisiting in AWS to␣
↪restore an instance or cluster directly from it)
   3: instance_class (Set database instance class)
   4: availability_zone (Set the destination availability zone for the␣
↪instance(s) and its/their volume(s))
   5: parameter_groups (Set the name(s) of the parameter groups that will be␣
↪applied to the restored database instance (separated by ','))
   6: vpc_security_groups_ids (Set the id(s) of the VPC Security Groups that␣
↪will be applied to the restored database instance (separated by ','))
   7: tags (Set tags to the restored instance(s) (tag1key:tag1value,
↪tag2key:tag2value...))
   8: multi_az (Configure the restored database instance as Multi AZ)
   9: db_port (Set the database port)
   10: storage_type (Set the instance storage type)
   11: storage_throughput (Set the instance storage throughput)
   12: iops (Iops value for instance storage)
   13: pitr_from_instance_id (Set the instance identifier that will reference␣
↪the instance to be used to proceed with a Point In Time Recovery restore)
   14: pitr_date (Sets the date for Point in Time Recovery restore)
   15: pitr_latest (Run a Point in Time Recovery restore using the latest␣
↪date available)
   16: parquet_db_socket (Set the socket path to use a socket based␣
↪connection to restore parquet data from the backup directly into a database)
   17: parquet_db_engine (Set the engine (postgresql or mysql) to restore␣
↪parquet data from the backup directly into a database)
   18: parquet_db_host (Set the host to connect and restore parquet data from␣
↪the backup directly into a database)
   19: parquet_db_name (Set the database name to connect and restore parquet␣
↪data from the backup directly into a database)
   20: parquet_db_port (Set the database port to connect and restore parquet␣
↪data from the backup directly into a database)
   21: parquet_db_user (Set the database user to connect and restore parquet␣
↪data from the backup directly into a database)
   22: parquet_db_password (Set the database password to connect and restore␣
↪parquet data from the backup directly into a database)
   23: parquet_save_mode (Set the restore mode for a parquet restore: Append,␣
↪Overwrite, ErrorIfExists or Ignore)
   24: access_key (Set a different access key to access to the destination)
   25: secret_key (Set a different secret key to access to the destination)
   26: region (Set the destination region)
   27: debug (Change debug level)
Select parameter to modify (1-27): 23
Please enter a value for parquet_save_mode: Overwrite
Plugin Restore Options
```

(continues on next page)

(continued from previous page)

```
Option                         Current Value       Default Value
instance_identifier:           *None*              (*none*)
from_snapshot_id:              *None*              (*none*)
instance_class:                *None*              (*none*)
availability_zone:             *None*              (*none*)
parameter_groups:              *None*              (*none*)
vpc_security_groups_ids:       *None*              (*none*)
tags:                          *None*              (*none*)
multi_az:                      *None*              (*none*)
db_port:                       *None*              (*none*)
storage_type:                  *None*              (*none*)
storage_throughput:            *None*              (*none*)
iops:                          *None*              (*none*)
pitr_from_instance_id:         *None*              (*none*)
pitr_date:                     *None*              (*none*)
pitr_latest:                   *None*              (*none*)
parquet_db_socket:             *None*              (*none*)
parquet_db_engine:             mysql               (*none*)
parquet_db_host:               *None*              (*none*)
parquet_db_name:               test                (*none*)
parquet_db_port:               *None*              (*none*)
parquet_db_user:               root                (*none*)
parquet_db_password:           root                (*none*)
parquet_save_mode:             Overwrite           (*none*)
access_key:                    *None*              (*none*)
secret_key:                    *None*              (*none*)
region:                        *None*              (*none*)
debug:                         *None*              (*none*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.3.bsr
Where:           /
Replace:         Always
FileSet:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2024-06-06 14:39:13
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
Job queued. JobId=4
```

Listing 280: **Restore job result**

```
list joblog jobid=4
*list joblog jobid=4
| 127.0.0.1-dir JobId 4: Start Restore Job RestoreFiles.2024-06-06_14.39.14_
↪13                         |
```

(continues on next page)

(continued from previous page)

```
| 127.0.0.1-dir JobId 4: Restoring files from JobId(s) 1                         ␣
↪                          |
| 127.0.0.1-dir JobId 4: Connected to Storage "File" at 127.0.0.1:8103 with␣
↪TLS                         |
| 127.0.0.1-dir JobId 4: Using Device "FileStorage" to read.                     ␣
↪                          |
| 127.0.0.1-dir JobId 4: Connected to Client "127.0.0.1-fd" at 127.0.0.1:8102␣
↪with TLS                  |
| 127.0.0.1-fd JobId 4: Connected to Storage at 127.0.0.1:8103 with TLS          ␣
↪                          |
| 127.0.0.1-sd JobId 4: Ready to read from volume "TEST-2024-06-06:0" on File␣
↪device "FileStorage" (/tmp/regress/tmp). |
| 127.0.0.1-sd JobId 4: Forward spacing Volume "TEST-2024-06-06:0" to␣
↪addr=247                          |
| 127.0.0.1-sd JobId 4: Elapsed time=00:00:01, Transfer rate=159.2 K Bytes/
↪second                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Plugin log of this job available in: /tmp/
↪regress/working/amazon-rds/amazon-rds-debug-0.log |
| 127.0.0.1-fd JobId 4: amazon-rds: Jar Version: 1.0.0                           ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Starting backend restore process             ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: No port provided. Using default port for␣
↪mysql:3306                  |
| 127.0.0.1-fd JobId 4: amazon-rds: Instance identifier was not set              ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: From snapshot id was not set                 ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Pitr latest : false                          ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Pitr date was not set                        ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Pitr from instance id was not set            ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Parquet DB Name: test                        ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Parquet DB User: root                        ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:cdimages with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:cdimages restored                      ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:fileevents with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:fileevents restored                    ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:filemedia with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:filemedia restored                     ␣
↪                          |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:restoreobject with 0␣
↪records to database: test in mode:overwrite |
```

(continues on next page)

(continued from previous page)

```
| 127.0.0.1-fd JobId 4: amazon-rds: Table:restoreobject restored                ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:events with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:events restored                       ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:malwaremd5 with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:malwaremd5 restored                   ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:mediatype with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:mediatype restored                    ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:file with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:file restored                         ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:pool with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:pool restored                         ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:metaattachment with 0␣
↪records to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:metaattachment restored               ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:location with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:location restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:status with 28 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:status restored                       ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:counters with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:counters restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:storage with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:storage restored                      ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:unsavedfiles with 0␣
↪records to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:unsavedfiles restored                 ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:path with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:path restored                         ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:client with 0 records to␣
↪database: test in mode:overwrite |
```

(continues on next page)

(continued from previous page)

```
| 127.0.0.1-fd JobId 4: amazon-rds: Table:client restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:jobmedia with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:jobmedia restored                   ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:basefiles with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:basefiles restored                  ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:tagclient with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:tagclient restored                  ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:log with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:log restored                        ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:malwaresha256 with 0␣
↪records to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:malwaresha256 restored              ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:tagobject with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:tagobject restored                  ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:snapshot with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:snapshot restored                   ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:object with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:object restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:tagjob with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:tagjob restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:tagmedia with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:tagmedia restored                   ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:device with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:device restored                     ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:jobhisto with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:jobhisto restored                   ␣
↪                            |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:pathhierarchy with 0␣
↪records to database: test in mode:overwrite |
```

(continues on next page)

(continued from previous page)

```
| 127.0.0.1-fd JobId 4: amazon-rds: Table:pathhierarchy restored                ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:media with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:media restored                        ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:pathvisibility with 0␣
↪records to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:pathvisibility restored               ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:job with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:job restored                          ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:fileset with 0 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:fileset restored                      ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:metaemail with 0 records␣
↪to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:metaemail restored                    ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:locationlog with 0␣
↪records to database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:locationlog restored                  ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: Restoring table:version with 1 records to␣
↪database: test in mode:overwrite |
| 127.0.0.1-fd JobId 4: amazon-rds: Table:version restored                      ␣
↪                         |
| 127.0.0.1-fd JobId 4: amazon-rds: No more items to restore. Restore ended     ␣
↪                         |
| 127.0.0.1-dir JobId 4: Bacula 127.0.0.1-dir 18.0.3 (22May24):
Build OS:               x86_64-pc-linux-gnu ubuntu 22.04
JobId:                  4
Job:                    RestoreFiles.2024-06-06_14.39.14_13
Restore Client:         "127.0.0.1-fd" 18.0.3 (22May24) x86_64-pc-linux-gnu,
↪ubuntu,22.04
Where:                  /
Replace:                Always
Start time:             06-jun-2024 14:39:16
End time:               06-jun-2024 14:39:27
Elapsed time:           11 secs
Files Expected:         41
Files Restored:         41
Bytes Restored:         118,684 (118.6 KB)
Rate:                   10.8 KB/s
FD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:            Restore OK |
| 127.0.0.1-dir JobId 4: Begin pruning Jobs older than 6 months .               ␣
```

(continues on next page)

(continued from previous page)

```
↪                           |
| 127.0.0.1-dir JobId 4: No Jobs found to prune.                            ␣
↪                           |
| 127.0.0.1-dir JobId 4: Begin pruning Files.                               ␣
↪                           |
| 127.0.0.1-dir JobId 4: No Files found to prune.                           ␣
↪                           |
| 127.0.0.1-dir JobId 4: End auto prune.                                    ␣
↪                           |
```

List & Query

It is possible to list database instances or clusters using the bconsole `.ls` command or through a `.query` command.

Any Fileset parameter can be employed in the `plugin=""` value to filter the results,
similar to how the backup operates.

Below, there are several examples:

List instances:

Listing 281: **Query example: List all instances**

```
.query plugin="amazon-rds: region=us-east-1 access_key=AKIAQTESTKEY12134g␣
↪secret_key=m23480ahpqwre894qwrffsfdSecretExample" client=127.0.0.1-fd␣
↪parameter="instance"
instance=jg-pg-aws
instanceId=jg-pg-aws
jg-pg-aws.dbName=bacula
jg-pg-aws.status=starting
jg-pg-aws.createTime=2024-03-28 10:27:41
jg-pg-aws.latestRestorableTime=2024-06-04 11:14:35
```

Note: The previous query command can be used to retrieve the `latestRestorableTime`, which is useful for any Point-In-Time recovery operation.

List snapshots of a given instance:

Listing 282: **Query example: Get all snapshots of a given instance**

```
*.query client=127.0.0.1-fd plugin="amazon-rds: region=us-east-1 access_
↪key=AKIAQTESTKEY12134g secret_key=m23480ahpqwre894qwrffsfdSecretExample␣
↪instances=jg-pg-aws" parameter=snapshot
snapshot=plugintest-20240527-114110-03
snapshotId=plugintest-20240527-114110-03
plugintest-20240527-114110-03.instanceId=jg-pg-aws
plugintest-20240527-114110-03.status=available
plugintest-20240527-114110-03.createTime=2024-05-27 09:41:25
snapshot=plugintest-20240527-124713-03
snapshotId=plugintest-20240527-124713-03
```

(continues on next page)

(continued from previous page)

```
plugintest-20240527-124713-03.instanceId=jg-pg-aws
plugintest-20240527-124713-03.status=available
plugintest-20240527-124713-03.createTime=2024-05-27 10:48:05
snapshot=rds:jg-pg-aws-2024-05-16-12-30
snapshotId=rds:jg-pg-aws-2024-05-16-12-30
rds:jg-pg-aws-2024-05-16-12-30.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-16-12-30.status=available
rds:jg-pg-aws-2024-05-16-12-30.createTime=2024-05-16 12:30:54
snapshot=rds:jg-pg-aws-2024-05-17-12-30
snapshotId=rds:jg-pg-aws-2024-05-17-12-30
rds:jg-pg-aws-2024-05-17-12-30.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-17-12-30.status=available
rds:jg-pg-aws-2024-05-17-12-30.createTime=2024-05-17 12:31:00
snapshot=rds:jg-pg-aws-2024-05-24-16-02
snapshotId=rds:jg-pg-aws-2024-05-24-16-02
rds:jg-pg-aws-2024-05-24-16-02.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-24-16-02.status=available
rds:jg-pg-aws-2024-05-24-16-02.createTime=2024-05-24 16:02:49
snapshot=rds:jg-pg-aws-2024-05-25-12-30
snapshotId=rds:jg-pg-aws-2024-05-25-12-30
rds:jg-pg-aws-2024-05-25-12-30.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-25-12-30.status=available
rds:jg-pg-aws-2024-05-25-12-30.createTime=2024-05-25 12:30:44
snapshot=rds:jg-pg-aws-2024-05-26-12-30
snapshotId=rds:jg-pg-aws-2024-05-26-12-30
rds:jg-pg-aws-2024-05-26-12-30.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-26-12-30.status=available
rds:jg-pg-aws-2024-05-26-12-30.createTime=2024-05-26 12:30:38
snapshot=rds:jg-pg-aws-2024-05-27-09-41
snapshotId=rds:jg-pg-aws-2024-05-27-09-41
rds:jg-pg-aws-2024-05-27-09-41.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-27-09-41.status=available
rds:jg-pg-aws-2024-05-27-09-41.createTime=2024-05-27 09:41:25
snapshot=rds:jg-pg-aws-2024-05-27-10-47
snapshotId=rds:jg-pg-aws-2024-05-27-10-47
rds:jg-pg-aws-2024-05-27-10-47.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-05-27-10-47.status=available
rds:jg-pg-aws-2024-05-27-10-47.createTime=2024-05-27 10:48:05
snapshot=rds:jg-pg-aws-2024-06-03-11-41
snapshotId=rds:jg-pg-aws-2024-06-03-11-41
rds:jg-pg-aws-2024-06-03-11-41.instanceId=jg-pg-aws
rds:jg-pg-aws-2024-06-03-11-41.status=available
rds:jg-pg-aws-2024-06-03-11-41.createTime=2024-06-03 11:41:04
```

Note: The previous query command can be used to retrieve the snapshot ids currently available in the cloud. They can be used to restore the instance to that particular state by using the `from_snapshot_id` restore parameter.

Limitations

The following article presents limitations of Amazon RDS Plugin.

When the export to S3 function is not used, Amazon RDS plugin will not store actual data in Bacula volumes, as it will merely function as a Snapshot manager within the AWS cloud.

In instances where a new Amazon RDS instance or cluster is desired for restoring purposes, the source Snapshot must be present in the Amazon RDS service to be selected for use.

Point-in-Time Recovery function is also based on the existing Amazon RDS configuration and previous snapshots. To use this feature, is necessary to enable the Amazon RDS “Automated Backups” function and establish a retention period, even if other snapshots are managed through Bacula Enterprise Amazon RDS Plugin. With that function enabled, RDS service will upload transaction logs for database instances to Amazon S3 every five minutes. To see the most recent restorable time for a database instance, one can employ the `.query` command with the `instance` parameter command and examine the value returned in the `LatestRestorableTime` field.

The export function to S3, and consequently, the backup with the plugin using Apache Parquet format, contains **only the data** of the protected databases. Indexes and any additional information are excluded from both the export and the backup stored in Bacula.

Cloud Costs

As you may already be aware, storing data in the cloud incurs additional expenses.

Data transfer needs to be considered as well. While uploading data is typically free or incurs minimal costs, downloading it is usually not free, and charges will apply based on each operation and the volume of data transferred.

Amazon employs a pricing model for each of its storage tiers. Additionally, costs will differ depending on the region you use. More information are to be found here:

- https://aws.amazon.com/s3/pricing/

Exporting an Amazon RDS backup to S3 and subsequently downloading it will result in certain charges. Keep it in mind and schedule this operation according to a frequency that aligns with your needs and convenience.

Troubleshooting

In this article, there are suggested solutions to common situations that can cause trouble during the usage of the Amazon RDS Plugin.

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (which can be found in the `amazon-ec2-debug.err` file), you are very likely using a high level of concurrency through internal `concurrent_threads` parameter and/or parallel jobs.

To address this issue, you may consider the following options:

a) Reduce `concurrent_threads` parameter.

b) Reduce the number of jobs running in parallel.

c) If neither of the above is feasible, you should increase the JVM memory (you may also need to increase the underlying host physical memory)

To increase the JVM memory, you will need to:

Create the following file: `/opt/bacula/etc/amazon_rds_backend.conf`.

Below, an example of the contents:

Listing 283: **Limits file contents**

```
AMAZON_RDS_JVM_MIN=2G
AMAZON_RDS_JVM_MAX=8G
```

Those values will define the MIN (`AMAZON_RDS_JVM_MIN`) and MAX (`AMAZON_RDS_JVM_MAX`) memory values assigned to the JVM Heap size. Exercise caution if you are running jobs in parallel, as large values and several jobs at once could quickly consume all available memory on your host.

The `/opt/bacula/etc/AMAZON_RDS_backend.conf` will remain unchanged during package upgrades, ensuring that your memory configurations are retained.

Throttling

The API usage in the Bacula Enterprise Amazon RDS Plugin is generally less intensive compared to certain other plugins, as the operations (such as make snapshot, remove snapshot, list instances) are of a higher level, and the number of these operations is not expected to reach the boundaries. Nevertheless, it is advisable to review limits of this service at the following link: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Limits.html

If you ever experience throttling, the recommended solution is to reduce concurrency with one of the different strategies mentioned in the *Best Practices* section.

6 NAS

Important: NAS solutions are used with the File Daemon.

6.1 NDMP Plugin

This chapter aims at presenting the reader with information about the **Bacula Enterprise Network Data Management Protocol (NDMP) Plugin**. The document briefly defines the scope of its operations, describes the target technology of the plugin, and presents its main features and various techniques and strategies to backup NAS filers using the NDMP or NFS/CIFS protocols with Bacula Enterprise.

Scope

The current version of the NDMP Plugin supports several NAS vendors, like NetApp, HPE, EMC, and Huawei, among others. With NetApp's DataONTAP, we tested versions 8.x and 9.x.

Note: Vendor implementations of NDMP differ, so we cannot guarantee that all NDMP implementations will work with our NDMP Plugin.

Features

The following chapter describes the features of NDMP Plugin:

- File to Server mode
- ACL support
- File Restore via the Single Item Restore technique
- NDMP Dump/Tar
- NetApp SMTAPE
- Full, Incremental and Differential support for NDMP Dump/Tar backup
- Full for NDMP SMTAPE backup
- Full, Incremental and Differential support for NDMP SMTAPE backup NetApp
- NetAppCABConfiguration
- Copy and Migration jobs

Available since Bacula Enterprise 12.4.0

The Bacula Enterprise NDMP Plugin can analyse the native NetApp Dump format, the EMC Unity DUMP format, and the universal (i.e. supported by many different vendors) Tar format to ensure a high deduplication ratio with Bacula Systems' ged. For other NDMP vendors, *Aligned Volumes* may give good results with the appropriate storage system.

Backup Strategies

This article aims at presenting possible backup strategies for the NDMP Plugin.

Backing Up Using NDMP

NDMP has the main advantage of speed over the NFS/CIFS protocols for backing up NAS filers. Note that NDMP is a control protocol, not a backup format. The NDMP backup format is arbitrary (implementation-specific) and defined individually by each storage platform vendor. Thus in general, an NDMP backup made on one vendor's NAS system cannot be restored to a different vendor's NAS.

NDMP can be used in 3 ways:

- **Filer to Server**: Filer backing up across the LAN to your Bacula server.
- **Filer to Self**: Filer backing up to an attached tape drive.

- **Filer to Filer**: Filer backing up across the network to another filer's tape drive.

At this time, the **Bacula Enterprise** NDMP Plugin supports only the **Filer to Server** mode. In this mode, **Bacula** encapsulates NDMP backup images inside the standard Bacula backup data stream, allowing it to be multiplexed with other backup streams and providing capabilities such as arbitrary (Bacula) storage device use, compression, checksum verification, etc. Inside the saved data stream that contains the NDMP stream, the actual NDMP data remains opaque (vendor specific format, sometimes proprietary).

When using NDMP provided by the NAS vendor, attributes such as ACLs are included in the backup stream and handled correctly by the NAS vendor on backup and restore. Determining which files should be saved is also part of the NAS vendor's job, and Bacula has no way to ensure that all files will be protected with technologies such as our Accurate Mode.

Since Bacula does not implement the NDMP Direct Access Recovery (DAR) protocol (which is reputed to be very slow), doing a single file restore is possible via the Single Item Restore technique (`use_hist` option).

Restoring a single file with the NDMP Plugin without the Single Item Restore is possible by restoring the NDMP datastream to a local machine (same one that has the NDMP Plugin), then opening the resulting file. This requires to know the vendor data format and to have a program that can read it. Since it involves restoring the whole backup and then extracting the desired files, it isn't very efficient.

An alternative, simple, and fast technique is to make use of file system snapshots made on the NAS to keep a history of file system states. These snapshots remain on the NAS box and can thus easily be accessed to restore individual files. In this manner, the NDMP backups are used as a disaster recovery mechanism only, rather than a means of restoring individual files.

In short, the best use of the NDMP Plugin is to protect the NAS from disaster situations, while using snapshots on the NAS box allows to recover files quickly on a daily basis.

Note: It is also possible to backup a particular snapshot with the NDMP Plugin.

Using NetApp SMTAPE Backup Option

The SMTAPE backup option essentially makes a raw binary dump of the selected NAS partitions (volumes). The big advantage of the SMTAPE feature is the speed, especially for volumes with millions of files on them. The standard backup option (DUMP) is much too slow to effectively backup large, multi-terabyte volumes, which are the norm these days. For Full backups, SMTAPE will significantly outperform DUMP on speed. Also, because it includes the entire snapshot history, users can reduce the number of snapshots they keep on their primary volume since the backups include them.

For example, keeping 30 days worth of snapshots on your filer and doing monthly full backups with SMTAPE, and keeping those for years would give you access to your entire snapshot history for the time period without taking up excessive space on your primary volumes. With large capacity tapes, that turns into a pretty granular data protection setup. If you had hourly snapshots, you could potentially get anything back from any hour going back many years just by doing a full backup every month. The recovery path is a bit complex but manageable.

Available since Bacula Enterprise 12.4.0

Recent versions of SMTAPE now support the Incremental backup feature.

There are pros and cons to both approaches, but SMTAPE as an NDMP backup method solves a particularly painful problem for many enterprise users. It is a useful feature, and as long as NetApp is supporting its use by backup applications, it should be considered as first choice backup method.

See the section about SMTAPE Incremental/Differential configuration SMTAPEConfiguration for more information.

Backing Up Using NFS/CIFS without NDMP

Using NFS or CIFS to mount NAS volumes on a File Daemon machine allows to back up NAS files without the NDMP Plugin the same way it would be done with a local filesystem backup. However, when dealing with millions of small files, directory walking and network latency tend to decrease the backup throughput dramatically. By using techniques such as "Incremental forever" and Virtual Full backups that save only new and modified files, you can, however, achieve reasonable storage space consumption in this situation.

The Virtual Full feature reads the latest Full backup and all the Incrementals and merges them to create a new Full backup that looks like a "real" Full backup, but without having to access the NAS device. This kind of backup is sometimes referred to as a synthetic Full. During a Virtual Full backup, the data will be copied directly from one **Bacula** Volume to another, without accessing the NAS. See VirtualFullJobs for more information on this subject.

Most NAS systems provide file system snapshot capabilities, and some of them also provide capabilities to avoid the file system walk to find modified or new files, which is a function typically built on top of the snapshot comparison. If those features are accessible, it is possible to perform a file level backups with both better consistency and less overhead using "Incremental Accelerator". Bacula Systems offers such functionality for *NetApp NAS systems* and the nutanix_hfc plugins.

Using the NFS or CIFS protocols to do a complete restore of a large NAS file system can be excessively time consuming especially in urgent disaster recovery situations.

In addition, Access Control Lists (ACLs) are not always handled correctly using NFS/CIFS. Unix POSIX ACLs can be handled directly by **Bacula**, but backing up Windows ACLs needs special techniques (see the SavingACLs chapter).

Strategies Comparison

Table 49: Backup Strategies Comparison

Features	NFS Mount	NDMP dump/tar	NDMP SMTAPE
Incremental backup	Yes	Yes	No 3
Accurate support	Yes	Yes 1	Yes 1
ACL	No/Yes 1	Yes	Yes
Exclusion/Inclusion	Yes	Yes	No
Cross-filer recovery	Yes	No 1	No 1

Backup Side	NFS Mount	NDMP dump/tar	NDMP SMTAPE
Network load	Medium	Low	High
Backup size	Small	Small	Large 1
Average backup speed	Slow	Fast	Fast 1
Speed with many files	Slow	Fast	Very Fast

Restore Side	NFS Mount	NDMP dump/tar	NDMP SMTAPE
Network load	Low/Medium	High	High
Single File Restore	Very Fast	Slow	Slow
File Level Restore	Yes	Yes 2	No
Disaster Recovery	No	Yes	Yes
NetApp Cluster Support	Yes	Yes	Yes

1 Platform or configuration dependent

2 With use_hist option

3 Incremental and Differential backup support with NetApp only. See SMTAPEConfiguration for more information.

Installation

The NDMP Plugin is installed along the FileDaemon. To avoid network load between the FD and the SD, please install it on the same system of a SD.

It is possible to install the NDMP Plugin on any other host, but doing so is less efficient because it will cause the data to traverse two network links – once from the NAS to the File Daemon, and a second time from the File Daemon to the Storage Daemon.

If you plan to use the SMTAPE Incremental feature, it is required to install the NetAp OnTap API on your system. See NDMPNetAppInstallation for more information.

Prerequisites

The **Plugin Directory** option in the **File Daemon** resource should point to where the `ndmp-fd.so` plugin is installed, which usually is `/opt/bacula/plugins`

```
FileDaemon {
   Name = bacula-fd
   Plugin Directory = /opt/bacula/plugins
   ...
}
```

The File Daemon should have direct network access to the NAS, which can be verified using `telnet`, for example:

```
# telnet nasbox 10000
Connected to nasbox.
Escape character is '^]'.
```

Note: NDMP, when doing the actual backup, connects back from the filer to the host initiating the NDMP session. This may require firewalls to be configured accordingly. Using the **data_port_range** option, it is possible to control which TCP ports the NAS will attempt to use, allowing for more restricted firewall configuration. When the network connection is not possible, the error `NDMP4_CONNECT_ERR` can be issued.

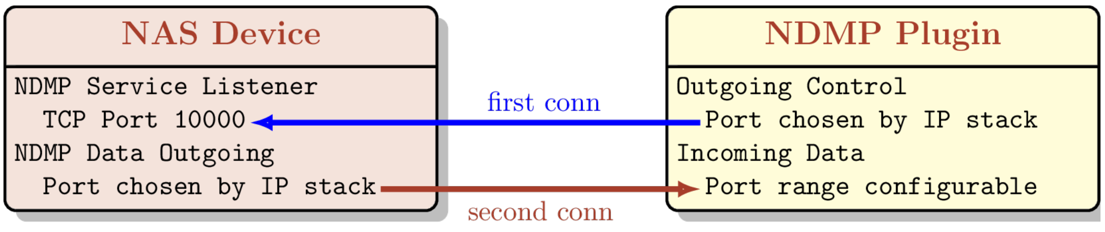


Fig. 126: Network Connections

The job specific configuration of the NDMP Plugin is part of the Fileset that is used and provided as options of the plugin directive.

Depending on NAS setup, the authentication method can be configured to be **md5** or **text**.

NDMP Plugin Installation

This article describes how to install Bacula Enterprise NDMP Plugin.

NDMP Installation with BIM

In order to install the NDMP Plugin with BIM, install the File Daemon with BIM and choose to install the NDMP Plugin during the FD installation.

Click here for more details on the plugin installation process with BIM.

Installation with Package Manager

Installation of the Bacula Enterprise NDMP Plugin is most easily done by adding the repository file suitable for the existing subscription and the distributions package manager configuration.

Deb based Linux distribution

An example would be `/etc/apt/sources.list.d/bacula.list` for deb based Linux distributions with the following content:

```
#Bacula Enterprise
[...]
deb https://www.baculasystems.com/dl/@customer-string@/debs/ndmp/@version@/
↪stretch-64/ stretch ndmp
```

After that, a run of `apt-get update` is needed. Then, the plugin can be installed using `apt-get install bacula-enterprise-ndmp`

RPMS based Linux distribution

On RHEL or alternative distributions, extend the repository file for your package manager to contain a section for the plugin - `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer@/rpms/bin/@version@/rhel7-
↪64/
enabled=1
protect=0
gpgcheck=0

[BaculaNDMPPlugin]
name=Bacula Enterprise NDMP Plugin
baseurl=https://www.baculasystems.com/dl/@customer@/rpms/ndmp/@version@/rhel7-
↪64/
enabled=1
protect=0
gpgcheck=0
```

Then, perform a `yum update` or `dnf update` and after that the package can be installed with `yum install bacula-enterprise-ndmp` or `dnf install bacula-enterprise-ndmp`.

Manual installation of the packages, can be done after downloading the proper files from the Bacula Systems provided download area, and then using the low-level package manager (`rpm` or `dpkg`) to perform the plugin installation.

NetApp OnTAP Python API Installation

To install the NetApp OnTAP Python API, run the following commands as root:

```
# pip3 install netapp-ontap --target /opt/bacula/python
```

Unfortunately, the information needed to access the OnTAP API is usually different from the NDMP information, so it requires to specify a few new parameters on the plugin command line such as:

- `ontap_user`
- `ontap_password`
- `ontap_host`
- `ontap_vserver`
- `ontap_ignore_ssl`
- `ontap_profile`

To test the connectivity with the OnTAP API interface of your NetApp system, you may use the following command to list the volumes on the vserver:

```
# /opt/bacula/bin/snapmgr.py -e "vserver" -u "user" -f "host" -p "password" -
↪x -V
vol1
```

(continues on next page)

(continued from previous page)

```
svm1_root
backup_data1_1
```

If you have an error such as `401 Client Error:  Unauthorized for url`, you may have to use an other interface to query the OnTAP data.

Configuration

The NDMP Plugin configuration is done on the **client side** (File Daemon) and takes place in the **bacula-fd.conf** file.

NDMP Plugin Configuration

The following chapter presents the information on NDMP Plugin configuration.

Automatic Objects Integration

Since Bacula version 16.0.7, a new solution has been introduced, so that each object can be backed up separately with different Jobs to maximize the throughput and the resiliency. It is highly recommended to use this new solution for that purpose - *Automatic Object Integration (Scan Plugin)*. See an example for NDMP.

Custom Volume Format

Starting with Bacula Enterprise 10.2.4, the volume format is automatically detected from the NDMP system. The `volume_format` option should not be required anymore.

The NDMP Plugin should be aware of the volume structure in order to detect if the administrator wants to restore to a new volume (`where=/dev/vol_tmp`) or inside a subdirectory of the targeted volume (`where=/tmp`).

The NDMP Plugin will automatically detect the following structure:

- `/vol/` (used on NetApp)
- `/root_vdm/` (used on sone EMC hardware)

If your volumes use a different naming scheme, you **must** use the **volume_format** option.

```
/dev/volume_home          -> volume_format=/dev/
/rootvolume/volume_tmp    -> volume_format=/rootvolume/
/VG/volume_var            -> volume_format=/VG/
```

The actual volume format and the allowed flexibility is vendor specific. In general, if volumes can be created with plain names and no path in the NAS configuration frontend, a fixed volume format will be in use.

```
Fileset {
 Name = NDMPFS
 ...
 Include {
```

(continues on next page)

(continued from previous page)

```
   Plugin = "ndmp:host=nasbox user=root pass=root file=/dev/vol1 volume_
↪format=/dev/"
 }
}
```

For example, on NetApp, the volume format is:

```
/vol/vol1
/vol/vol2
...
```

The common part to distinguish volumes and directories is **/vol/**. On Solaris with ZFS, volumes are stored in **/dev/**.

```
/dev/Volume01
/dev/Volume02
/dev/Volume03
```

If volumes are organized like in this example:

```
/FS1/
/FS2/
/FS3/
```

The **volume_format** should be configured as **/FS** – note that there is **no trailing slash** here!

Fileset Examples

This section will present various File Set examples.

```
Fileset {
 Name = NDMP
 Include {
  Plugin = "ndmp:host=nasbox user=root pass=root volume=/vol/vol1"
 }
}
```

```
Fileset {
 Name = NDMP_home
 Include {
  Plugin="ndmp:host=nas user=root pass=root volume=/vol/vol2 file=/home"
 }
}
```

```
Fileset {
 Name = NDMP_home
 Include {
  Plugin="ndmp:host=nas user=root pass=root volume=/vol/vol2 type=SMTAPE"
 }
}
```

```
Fileset {
 Name = NDMP_snapshot
 Include {
  Plugin="ndmp:host=nas user=root pass=root volume=/vol/vol2/.snapshot/snap1"
 }
}
```

```
Fileset {
 Name = NDMP_Incremental4ever
 Include {
  Plugin="ndmp:profile=netapp volume=/vol/vol2 use_base_date"
 }
}
```

```
Fileset {
 Name = NDMP_SMTAPE
 Include {
   # The plugin line should be on a single line
   Plugin = "ndmp:host=nasbox user=root pass=root volume=/svm1/vol2
                 ontap_user=admin ontap_password=pass
                 ontap_host=ontapsrv ontap_vserver=svm1 ontap_ignore_ssl"
 }
}
```

Using ndmp.conf to Store Username and Password

If exposing the access credentials of the NAS on the Plugin command line is undesirable, it is possible to use file `ndmp.conf`, stored on the File Daemon host, to save your credentials.

In the following example, the profile `root` will refer to the NAS host `nasbox`, the connection will use MD5 as authentication method ("/m"), the username will be `root` and the password will be `password`.

```
# cat /opt/bacula/etc/ndmp.conf
[--root]
-D nasbox/m,root,password
```

The format of the **-D** parameter is the following:

```
HOST[:PORT][/FLAGS][,USERNAME,PASSWORD]
```

Where:

Option	Description
HOST	Is the host name or IP address of the NAS
:POR	Optional port number. If not given the port number is 10000.
USER	A user name that will be recognized by the NAS. Whether this is a general user name or a special account within the NAS is implementation dependent.
PASS	The password corresponding to the USERNAME. The password field should not contain special characters such as: “, ’, !, $, / or #.
/ FLAG	Optional flags to indicate desired NDMP version or authentication method. The default version is negotiated to be the highest possible. The default authentication method is text (NDMP_AUTH_TEXT).
2	Use NDMP version 2.
3	Use NDMP version 3.
4	Use NDMP version 4.
n	Use no authentication (NDMP_AUTH_NONE).
t	Use text authentication (NDMP_AUTH_TEXT). The user name and password are sent over the network as clear text (unencrypted).
m	Use MD5 challenge/response authentication (NDMP_AUTH_MD5). The remote NAS is asked for a challenge. The password is used as the shared secret and is never sent over the network.

In the following example, the profile “11” will refer to the NAS host “10.1.1.11”, the connection will use MD5 as authentication method (“/m”), the username will be `root` and the password will be `mypassword`. A second profile “12” is also shown in this example `ndmp.conf` file.

```
[root@lxbackup ~]# cat /opt/bacula/etc/ndmp.conf
[--11]
-D 10.1.1.11/m,root,mypassword

[--12]
-D 10.1.1.12/m,root,mypassword2
```

In the following example two **Fileset** resources are configured – one for each of the two NAS profiles configured in the `ndmp.conf` example above.

```
Fileset {
 Name = NDMP_11
 Include {
  Plugin = "ndmp: host=10.1.1.11 profile=11 volume=/vol/LAN_backup type=smtape
↪"
 }
}

Fileset {
 Name = NDMP_12
 Include {
  Plugin = "ndmp: host=10.1.1.12 profile=12 volume=/vol/DMZ_backup type=smtape
↪"
 }
}
```

Note: The `ndmp.conf` file is different from the `ontap.conf` file described here ontapconf.

SMTAPE Incremental/Differential Configuration

When using SMTAPE with Incremental or Differential backup level, the NDMP Plugin has to manage snapshots via the NetApp OnTAP API. The REST API used by the NDMP Plugin is available on NetApp 9.6 and later.

Saving Access Control Lists (ACLs)

When using the NDMP Plugin, all metadata, including ACLs, should be backed up and restored correctly, as this functionality is provided by vendor built-in tools. Thus, problems in this area are in scope of vendor support and not due to particular backup or restore software.

ACLs Using CIFS

If a CIFS volume is mounted on Linux system, it may not be possible to backup ACLs. In order to reliably back up ACLs on CIFS volumes these must be mounted on a Windows server with a **Bacula** File Daemon installed. If a large amount of data is involved, avoiding to send the data over the network twice my be achieved by installing the Windows server in a KVM host on the Storage Daemon server. By doing so, it will be possible to backup Windows ACLs and reduce the network load.

This virtual server should be started only during the NAS backup. Contact Bacula Systems for more information on this configuration.

Newer Linux distributions with recent Samba tools may actually allow to back up and restore security-related file metadata, though this is not fully supported by Bacula at this time. Please contact **Bacula Systems** support for more information.

In particular for file systems which are accessed both by Windows systems via CIFS and by NFS, and which are storing file ACLs for both access methods, a non-NDMP backup preserving all ACL metadata is often impossible, or would cause considerable resource consumption.

ACLs Using NFS

To be able to backup file Access Control Lists (ACLs) through NFS mounts, ACL support needs to be enabled with the mount command as follows:

```
# mount -t nfs -o soft,intr,proto=tcp,acl nas:/vol/vol1 /nas/vol/vol1
```

In addition, it is needed to add the **ACL Support** option to the Bacula File Set used:

```
Fileset {
 Name = FS_NAS_VOL1
 Options {
   ACL Support = yes
   ...
 }
```

(continues on next page)

(continued from previous page)

```
 File = /nas/vol/vol1
}
```

Plugin Options

Table 50: NDMP Plugin Options

Option	Required	Default	Info	Example
host	Yes		NAS Hostname	host=192.168.0.1
user	Yes		Username	user=root
pass	Yes		Password	pass=password
port	No	10000	Connection Port	port=10010
auth	No	md5	md5, text, none	auth=text
profile	No		Profile name	profile=root
volume	Yes		Volume to backup	volume=/vol/vol0
volume_format	No		Specific Volume format	volume_format=/vol/
file	No	/	Directory to backup	file=/home
timeout	No	5mins	Timeout for various NDMP related commands	timeout=5h
type	No	dump	dump, tar, SMTAPE, config 1	type=dump
data_port_ra	No		Data port range	data_port_range=2000-3000
abort_on_err	No		Abort job on error	abort_on_error
debug	No		Enable debug	debug
use_cab 2	No		Enable the NetApp Cluster Aware Backup Protocol	use_cab
use_hist	No		Store HIST information in the catalog - need to adjust timeout	use_hist
limit_hist	No		Limit the path depth of the NDMP history stored in the catalog	limit_hist=3
use_base_da 3	No		Enable Incremental forever using BASE_DATE variable	use_base_date
hist_retentio	No		Prune HIST files stored in working/ndmp after the specified time	hist_retention=30days
max_level 3	No	9	Maximum number of DUMP level allowed by the NDMP vendor	max_level=32
ontap_profile 4	No		ontap.conf reference	ontap_profile=name1
ontap_host	No		OnTAP hostname	ontap_host=netapp.lan
ontap_user	No		OnTAP username	ontap_user=admin
ontap_passwor	No		OnTAP password	ontap_password=xxx
ontap_vserver	No		OnTAP vserver name	ontap_vserver=svm1
ontap_use_http	No		Use OnTAP HTTPS access	ontap_use_https

1 The `config` option is deprecated since Bacula Enterprise 10.2.4

2 use_cab option is available as of Bacula Enterprise 16.0.2

3 use_base_date and max_level options are available as of Bacula Enterprise 16.0

4 OnTAP options are available as of Bacula Enterprise 12.2.4

Table 51: NDMP Plugin Restore Options

Option	Required	Default	Info	Example
dest_volume	No		Destination volume to restore to	dest_volume=/vol/vol1

NetApp Configuration

OnTAP Configuration

The OnTAP parameters can be stored in `/opt/bacula/etc/ontap.conf` and be referenced with `ontap_profile` parameter on the plugin command line.

```
# cat /opt/bacula/etc/ontap.conf
[name1]
ontap_user=admin
ontap_password=MyPassword
ontap_vserver=svm1
ontap_ignore_ssl=True
ontap_host=10.0.2.3
```

```
Plugin = "ndmp:ontap_profile=name1 volume=/svm1/v1 host=netapp user=root␣
↪pass=xxx type=smtape"
```

NetApp Cluster Aware Backup (CAB) Configuration

With NetApp NDMP systems, it is possible to configure the Bacula Enterprise NDMP Plugin to contact the node hosting a volume for the NDMP session via the NDMP CAB Extension.

```
Fileset {
   Name = "ndmp-CAB"
   Include {
     Plugin = "ndmp: user=x password=y host=svm1 volume=/volume1 use_cab"
   }
}
```

NetApp Cluster Backup Using OnTAP API

With NetApp NDMP systems, it is also possible to configure the Bacula Enterprise NDMP Plugin to contact the node hosting a volume for the NDMP session via the OnTap API. The CAB extension is the recommended configuration to use.

Note: Available since Bacula Enterprise 16.0.1

For this configuration, it is required to configure the following:

- Install the python OnTap API via `pip3 install netapp-ontap --target=/opt/bacula/python`
- Create and configure the `/opt/bacula/etc/ontap.conf` file with OnTAP API related information
- Create and configure the `/opt/bacula/etc/ndmp.conf` file with NDMP related information
- One NDMP LIF per cluster node (role data)
- Use the DUMP method

In the following example, the OnTAP configuration should allow Bacula to query the OnTAP api.

```
# cat /opt/bacula/etc/ontap.conf
[svm1]
ontap_host=netapp-ontap-api.lan
ontap_user=bacula
ontap_password=AVeryGoodPassword?
ontap_vserver=svm1
ontap_ignore_ssl=True
```

In the following ndmp configuration file, the NDMP protocol must be configured on one of the node of the cluster.

```
# cat /opt/bacula/etc/ndmp.conf
[--svm1]
-D netapp-cluster1.lan/m,bacula,Xlkdflkdd
```

The `snapmgr.py` tool can be used to test the NetApp cluster configuration with the following command:

```
# /opt/bacula/bin/snapmgr.py -o svm1 -M
volume=/svm1/volume1 aggr=ONTAP97_02_FC_1
volume=/svm1/svm1_root node=ONTAP97-01 aggr=ONTAP97_01_FC_1 ip=10.0.110.93
```

The final Fileset plugin command line should look like the following:

```
Fileset {
   Name = "ndmp-ONTAP"
   Include {
     Plugin = "ndmp: profile=svm1 ontap_profile=svm1 host=svm1 volume=/svm1/
↪volume1"
   }
}
```

Some FlexGroup volumes may cause issues with the detection, it is possible to assign statically the node to connect to in the `ontap.conf` file.

```
# cat /opt/bacula/etc/ontap.conf
[svm1]
ontap_user = bacula
....

[svm1.flexgroup]
volume1 = 1.2.3.4
volume2 = 1.2.3.4
```

NAS Volume Configuration and Testing

Starting with Bacula Enterprise 10.2.4, the configuration will be automatically stored with the dump file. It is no longer required to specify this plugin command in the Fileset.

By specifying **type=config** as a NDMP plugin argument, Bacula will backup a configuration overview of all the NAS volumes to the file `/@ndmp/<host>.config`. To display the configuration, this file can be restored, and Bacula will display the NAS volume configuration in the Job log as shown in the example:

```
Plugin = "ndmp:host=nasbox user=root pass=pass type=config"
```

```
...
JobId 3: QR  File system /vol/vol2
JobId 3: QR    physdev
JobId 3: QR    unsupported 0x0
JobId 3: QR    type       WAFL
JobId 3: QR    status     online
JobId 3: QR    space      805306 total, 118784 used, 8052948 avail
JobId 3: QR    inodes     288238 total, 100 used
JobId 3: QR    empty default env
JobId 3: QR
JobId 3: QR  File system /vol/vol1
JobId 3: QR    physdev
JobId 3: QR    unsupported 0x0
JobId 3: QR    type       WAFL
JobId 3: QR    status     online
JobId 3: QR    space      8589938 total, 6594560 used, 858334 avail
JobId 3: QR    inodes     307450 total, 1703 used
JobId 3: QR    empty default env
JobId 3: QR
...
```

The NetApp master configuration file is located by default in `/vol/vol0/etc`. If the complete NAS setup should be restorable from scratch, the directory `/vol/vol0/etc` must be included in the backup Fileset.

The **type=config** option can also be used to check the Network configuration and the connection information between the Bacula Enterprise NDMP Plugin and the NAS. If this backup job is not successful, check the configuration, in particular network and firewall related, and the access password.

Operations

The following chapter describes details regarding NDMP Plugin operations.

Backup

Backups created by the NDMP Plugin can be freely copied or migrated using the respective Job types: Copy Job and Migration Job of **Bacula Enterprise** as of version 10.

Restore

File Level Recovery

Using snapshot options available directly on the NAS allows to restore any file to any point in time represented by a snapshot. Since snapshots have little to do with data protection, using NDMP with native format or NetApp SMTAPE option can be used to protect your NAS system from a disaster situation. Combining both techniques provides data protection with some flexibility.

On NDMP systems that support DAR, the plugin option `use_hist` can be used to enable the "Single Item Restore" feature. With the `use_hist` option, all the filenames included in a `dump` are stored in the Bacula catalog, and a `restore` session will have the ability to choose individual files or directories to restore. Only the selected files are restored on the NDMP destination system. The correct `dump` file is automatically selected during the file selection process.

Restoring Using NFS/CIFS

Restoring through NFS/CIFS can be done as usual with `bconsole`, `BAT` or `BWeb`.

Restoring Using NDMP

To perform a full restore of an NDMP protocol backup, select one of the *.dump, *.tar, or *.smtape files in the `/@ndmp/` virtual directory. It is possible to change the `where` parameter to restore to an alternate location. However, the `RegexWhere` option is not available with this method.

The `where` parameter may refer to a directory inside the original volume, or a directory inside an alternate volume. The **volume_format** option permits to configure how the NDMP Plugin will distinguish volumes and directories (see customvolumeformat for more information).

In a case that there is a need to explicitly define an alternate volume to restore to, the `dest_volume` plugin option can be used. This option can be used by modifying the plugin arguments at restore time.

Below, additional preparations for SMTAPE data stream restores are described.

In the following example, if the `dest_volume` is, during the restore, set to **/vol/vol1**, the NDMP stream will be restored to the directory `/restored` in volume `vol1`.

```
* restore where=/restored
```

In the following example, when `dest_volume` is not set, during the restore, the NDMP stream will be restored to the original volume, into the directory `/tmp`.

```
* restore where=/tmp
```

When restoring an NDMP data stream, it is possible that the filer device can not apply all metadata, in particular ACLs. Such situations need to be resolved after the restore itself, usually by checking and possibly adjusting permissions, ownership and ACLs of restored files through a system mounting the restored file system. In those situations, it might be important to observe if permissions or ownership are inherited through directory hierarchies.

Restoring Selected Files from NDMP Streams

If the `use_hist` option has been applied with NDMP based backups, information about individual files will be added to Bacula's Catalog. To restore selected files, not complete volumes, you would navigate into the file system tree below the **@ndmp/** directory, pick individual files as usual, and finalize the restore preparation in the usual way, too. It is required to restore to an NDMP Plugin in this case, and the `where=` option will still be pointing to a directory inside the target volume. The volume to restore to would also be indicated by the plugin option `dest_volume`.

Note: Such single file restores will still read the complete NDMP data stream and send it to the filer device, which is responsible to apply the file selection during restoration.

Note that **SMTAPE** format data streams do not provide single file restore capabilities.

Restoring Files without NDMP Host

In some cases, it might be necessary to extract files generated by the NDMP host by first restoring the NAS backup to a NDMP Plugin host machine, then using programs such as `restore` or `tar` on the restored data stream. For that purpose, select the respective NDMP data stream (not the individual files, in case `use_hist` is applied) and the `where=` option with the following format, i.e. with a greater-than sign directly followed by the absolute target path on the FD host:

```
* restore where=">/tmp"
```

NDMP files would then be extracted to the **local** `/tmp/` directory.

Note: The quotes in the above example must be used only when using command line argument, they are not needed when editing options via the restore menu.

Restoring with NetApp SMTAPE

The SMTAPE option using the path to the root of the volume (e.g., `/vol/foo`) will back up and restore complete volumes. This is closer to backing up and restoring a hard disk partition using `dd` or `partimage`. Bacula has no knowledge of the internal structure of the SMTAPE data because, when the volume is brought back online after the restore, the filer will simply recognize it as a WAFL filesystem and proceed on its own.

To restore a NetApp volume with the SMTAPE option, the target volume volume must have the type Data Protection (DP) and needs to be restricted, which can be done with the `volume` command on the NetApp host:

```
netapp> volume create -volume restore -aggregate agg1_data -size 5GB
   -state online -policy default -type DP -autosize-mode grow_shrink
   -snapshot-policy none -foreground true

netapp> volume restrict -volume restore
Volume "restore" is now restricted.
```

Then, it is needed to specify `where=/vol/restore` in the restore command.

Warning: With this method, any data on the volume would be overwritten by the previously saved data.

After restore is complete, the volume is in a read-only state. To make the volume writeable, you need to either submit the `snapmirror break` command or use the Data ONTAP APIs.

Best Practices

- While it is technically possible to specify multiple volumes in a single **Fileset** resource configuration (by using multiple plugin command lines), this is not necessarily the best way to perform multiple volume backups. It is strongly recommended that one **Fileset** definition is created per each filer volume being backed up.
- By default, if one of your volume fails to back up in a "multi-volume" backup job, the main Bacula job will terminate "Backup OK – with warnings." The JobStatus for jobs that terminate "Backup OK" and "Backup OK – with warnings" are not differentiated in the Catalog. They are both 'T', so this means that you will have to carefully monitor your backup job logs in case some volume backups fail and pay attention to the JobErrors field in the job summaries.
- To address this issue, there is a plugin option called "abort_on_error", which causes Bacula to immediately fail the job as soon as an error is detected while backing up a volume. However, if you use this option, and the backup of volume number 5 in a list of 10 volumes fails, then the whole job will be failed, and volumes 6-10 will not be backed up during that job's run.
- A 1:1 configuration (one volume backed up per job) means that the "abort_on_error" option will make more sense to enable in each job, so you will immediately know when a volume fails to backup since the Bacula job will terminate with a "Backup failed" message and 'f' in the Catalog for the job.
- With a 1:1 volume/job configuration, re-running a specific volume backup job is simple to do after the cause of the failure is investigated and fixed.
- In the example about the 10 volumes, without a 1:1 configuration, there is no way to re-run a backup of just the one volume that failed to back up.

All the above best practice are applied when using NDMPAutomaticObjectsIntegration.

Limitations

- The NDMP Plugin uses the job name to track the NDMP backup LEVEL information. It is not possible to mix multiple Filesets with a single job and the same NAS host and the same volumes.
- The NDMP password should not contain special characters such as: ", ', !, $, / or #
- With Single Item Restore, the number of files to be restored in a single session should not be too big. Some filers may not react very well with very large file lists.
- The Single Item Restore feature is not able to look into the Bacula backup stream. If a single file is selected inside a 1 TB NDMP dump file, 1TB will be read from Bacula's source volumes. The `DIRECT` variable cannot be set to "Yes".
- The NDMP dump does not provide information about deleted files. Bacula's Accurate Mode is not supported.
- With the Single Item Restore feature enabled, the NDMP HIST file needs to be analyzed in memory at the end of the NDMP backup session on the File Daemon system. The memory needed for this operation depends on the number of files and the tree structure. On some advanced situation, it is possible to configure a custom NDMP "HIST file" scanner by creating a symlink pointing to `/opt/bacula/bin/custom_ndmp_idx_dump`.
- With the Single Item Restore feature enabled, all the files and directories included in a NDMP dump are stored in the Bacula Catalog. To build the restore tree with the "restore" bconsole command, the Director needs to allocate an average of 170B per file in memory.
- The history file (HIST) is stored after each backup session in the working directory. This file is needed to perform a Single Item Restore session. If the file is not present in the directory, the restore may not be successful. The history file is included automatically in the backup. It is possible to restore the history file associated with a given NDMP dump prior to the file restore. When selected, the history file is automatically restored to the working directory.
- The Single Item Restore feature has been tested on NetApp and EMC ISILON systems with 50,000 files.
- The Single Item Restore feature has not been tested intensively with accented characters.
- Permission on directories that are automatically created during a restore with the Single Item feature may not be correct.
- The HIST option is supported only with the directory or node file history format. To correctly set this option and verify that it is enabled on EMC Isilion devices, the following commands in the Isilon's CLI interface may be issued:

```
NAS1-4# isi ndmp settings variables modify /ifs/path/to/specific/share␣
↪HIST D

NAS1-4# isi ndmp settings variables list
Path                             Name                 Value
----------------------------------------------------------
/ifs/path/to/specific/share      HIST                 D
```

- It is not possible to restore individual files from a dump during the local restore (with `where=">/path/"` option).
- The `replace` option is not respected when using the Single Item Restore feature.
- For NDMP Single Item Restores in bconsole, if you wish to use `Enter a list of files to restore` option (option 7), or the `file=` option on the bconsole "restore" command line, a file

with the list of filenames to restore must also include the names of the NDMP dump that holds the file(s) that you wish to restore.

- This plugin does not support backups made using the NDMP Tar format provided by EMC Unity systems.
- The native **dump** format often has a serious limitation of eight for the number of Incremental Jobs that you can run between two Full or Differential backups. Making more than eight Incrementals will not allow you to restore your system accurately. All files will be correctly restored, but subsequent restores of Incremental backups will not recreate the file system as it appeared during the final incremental backup.

  With some NAS vendors such as NetApp, it is possible to set more incremental levels using the `max_level` plugin command parameter (up to 32 with NetApp), or use the BASE_DATE incremental forever backup strategy with the `use_base_date` plugin command parameter.
- The NDMP Plugin is not compatible with the `sparse` Fileset option.
- Backups created using the NDMP Plugin are not compatible with Virtual Full jobs. Do not attempt to combine these two backup strategies as you will not be able to properly restore NDMP Plugin jobs from Virtual Full backups.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

6.2 HFC

NetApp HFC (Incremental Accelerator)

Overview

This user's guide presents various techniques and strategies to backup NetApp NAS having a huge number of files with Bacula Enterprise.

Scope

This paper will present solutions for **Bacula Enterprise** 6.0.6 and later, which are not applicable to prior versions.

Presentation

The plugin Incremental Accelerator for NetApp is designed to simplify and the optimize backup and restore performance of your NetApp NAS hosting a huge number of files.

When using the plugin, for Incremental backup Bacula Enterprise will query the NetApp for a previous backup snapshot then quickly determine a list of all files modified since the last backup instead of having to walk through the entire filesystem. Once Bacula has the backup list, it will use a standard network share (such as NFS or CIFS) to access files.

In order to compute the file list modified since the last backup, Bacula must store a few Snapshots on the NetApp device. The Incremental Accelerator Plugin will manage the snapshot list to minimize resource usage.

This plugin is available for RHEL 5 and 6 (32 / 64 bit) and SUSE Linux Enterprise Server 11 and 12 (64 bit) and supports NetApp 7-mode 7.3.6, 8.0, 8.1, 8.2 and 8.3, as well as all versions in c-mode.

Most recent product tests have been performed with SUSE Linux Enterprise Server 12 SP3 and NetApp Release 9.1.

Using The Incremental Accelerator for NetApp

Installation

The Incremental Accelerator for NetApp Plugin is available as a Bacula Enterprise package for all supported platforms. The package is composed of:

- `/opt/bacula/plugins/netapp-hfc-fd.so` Bacula Enterprise File Daemon plugin
- `/opt/bacula/bin/snapmgr` Bacula Enterprise NetApp Snapshot Manager

You must install this plugin on a Client machine that has network access to the NetApp filer. Bacula Systems advises you to use the Client that resides on your Storage Daemon so that the File daemon to Storage daemon data transfers are made internally rather than across the network.

Download the NetApp SDK installation files from the NetApp Support site for All Platforms: http://mysupport.netapp.com/NOW/cgi-bin/software/?product=NetApp+Manageability+SDK&platform=All+Platforms. Note: A form is required to download the software. Please complete and state that you want to use the SDK for Data ONTAP with Perl, and in the solution field enter "NetApp SDK"

```
% unzip netapp-manageability-sdk-5.4.zip netapp-manageability-sdk-5.4/lib/
↪perl/NetApp/*
Archive: netapp-manageability-sdk-5.4.zip
```

(continues on next page)

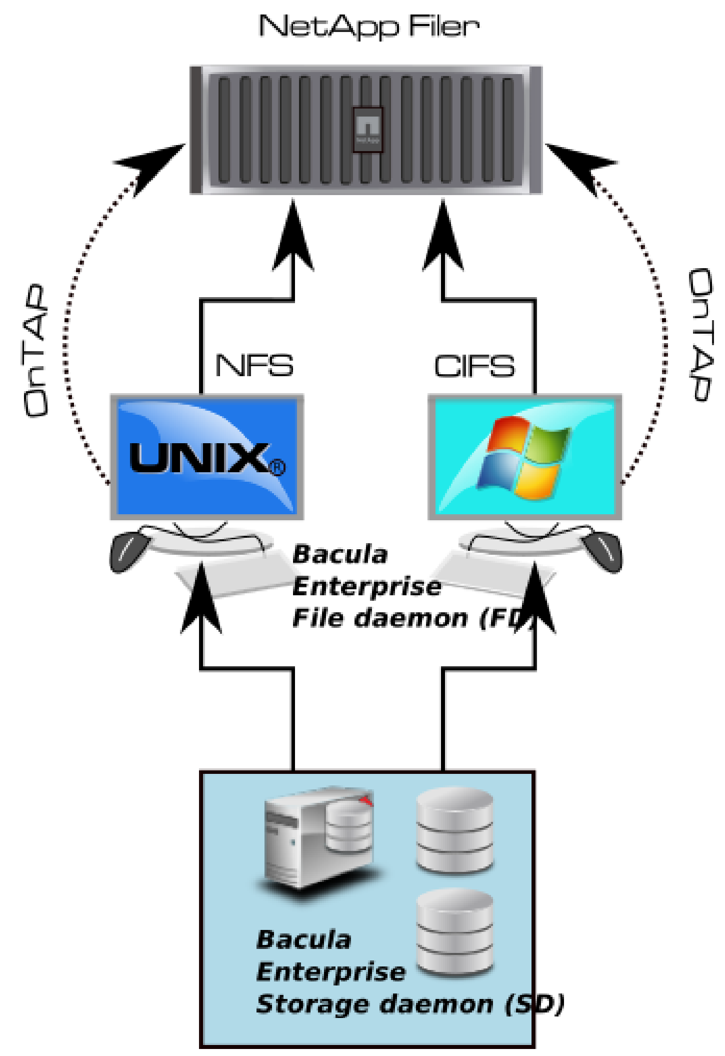

NetApp Filer
OnTAP
NFS
CIFS
OnTAP
UNIX®
Bacula Enterprise File daemon (FD)
Bacula Enterprise Storage daemon (SD)

Fig. 127: NetApp Incremental Accelerator Feature

(continued from previous page)

```
creating: netapp-manageability-sdk-5.4/lib/perl/NetApp/
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/DfmErrno.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/NaElement.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/NaErrno.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/NaServer.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/ONTAPILogParser.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/ONTAPITestContainer.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/SdkEnv.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/Test.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/OCUMAPI.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/OCUMClassicAPI.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/Ontap7ModeAPI.pm
inflating: netapp-manageability-sdk-5.4/lib/perl/NetApp/OntapClusterAPI.pm

% mv netapp-manageability-sdk-5.4/lib/perl/NetApp /opt/bacula/bin/
```

In order to save Windows extended file attributes such as NTFS ACLs, you will need to install the Incremental Accelerator for NetApp Plugin on a Windows server, this functionality will be available in a future version.

Configuration

As with all Bacula plugins, you must to specify the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

Your Backup Job should be defined as:

```
Job {
 Name = "NetApp-HFC"
 Client = storage-fd
 Fileset = FS_netapp
 ...
}

Fileset {
 Name = FS_netapp
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "netapp-hfc: host=nas1 password=pw exclude=vol0"
 }
}

Fileset {
```

(continues on next page)

(continued from previous page)

```
 Name = FS_netapp_cmode
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "netapp-hfc: host=nas1 password=pw exclude=vol0 vserver=vserver1␣
↪https"
 }
}
```

In this example, all NetApp volumes have to be mounted in the `/nas1` directory (the `mount_base` plugin option permits to configure the base mount point). The second Fileset `FS_netapp_cmode` is valid for a “Cluster Mode”.

```
% ls /nas1
vol1   vol2   vol3   vol4
```

The Incremental Accelerator for NetApp plugin accepts the parameters listed in table ‘Incremental Accelerator Plugin Options’.

Table 52: Incremental Accelerator Plugin Options

Option	*Required**	**Default**	**Info**	**Example**
host	Yes		NAS Hostname	`host=19 2.168.0.1`
user	Yes	root	Username	`user=root`
password	Yes		Password	`password =password`
volume	No		Volume to backup	`vo lume=vol0`
include	No	*	Volumes to backup	`incl ude=vol1*`
exclude	No		Volumes to exclude from backup	`exc lude=vol0`
mount_bas	No	*/host*	Where to find the volume mounted using NFS / CIFS	`mount_ base=/ mnt`
key	No	Job Name	Advanced snapshot pruning control	`key=MyKey`
snap_dir	No	.snapshot	Name of the directory where snapshots are visible	`snap_ dir=. snap`
vserver	No		Vserver name. This option is required for C-Mode support	`vserver =vserver1`
https	No	No	Use HTTPS with NetApp	`https`

Note: Because the NetApp Incremental Accelerator Plugin uses a dynamic `strip` Fileset option, the Fileset Include section should include **ONLY** the NetApp plugin; mixing other plugins or normal files with the HFC Plugin is not supported.

Note also that the NetApp filer needs to be configured to allow management access on the data access network interface, and a user with http access and the “vsadmin” role assigned should be used for this plugin. In c-mode, this may require creation of a user account on the SVM being backed up.

Snapshot Pruning

The NetApp SnapDiff API is based on Snapshots to generate the list of the changes. Once a snapshot is no longer required, the Bacula NetApp HFC Plugin will prune the snapshot.

The snapshot registry that is maintained by the plugin is stored in the Bacula working directory. The registry uses the Job name as a key to compute the snapshot chain. If you are doing multiple backups with the same Job or you have the same volume name in different context, we advise to set the extra `key` parameter in the Plugin command line to uniquely identify your jobs.

```
Fileset {
 Name = FS_netapp_cmode
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "netapp-hfc: host=n1 password=pw volume=vol0 vserver=v1 key=prod␣
↪https"
 }
}
```

External Password Storage

To avoid storing passwords in the Bacula Director configuration file, you can create a special configuration file located where the Plugin is installed and store the password in it.

```
% cat /opt/bacula/etc/snapmgr.conf
nas1:root = mypassword
netapp2:admin = myotherpassword
```

The first line will define the password for the root account on the "nas1" nas box. The Bacula NetApp HFC Plugin will use this entry with the following plugin command:

```
plugin = "netapp-hfc: host=nas1 user=root exclude=vol0"
```

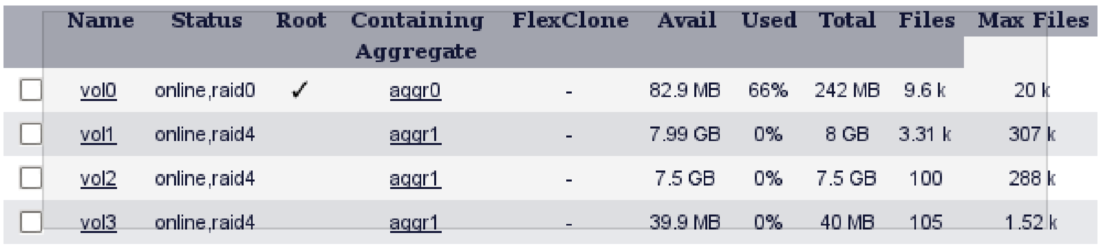
You can have multiple lines in the `snapmgr.conf`.

NFS Configuration

The NetApp HFC Plugin will detect the list of all volumes from the NetApp device, then the Plugin will assume that each volume is mounted and available under `mount_base` directory. For example, if you have the following volumes defined on your NetApp device (Fig *NetApp Volume List*).

You will need to mount each of them under your `mount_base` directory. For example, in `/mnt`:

```
% ls -l /mnt
drwxr-xr-x 2 root root 4096 May 23 14:52 vol0
drwxr-xr-x 2 root root 4096 May 23 14:52 vol1
drwxr-xr-x 2 root root 4096 May 23 14:52 vol2
drwxr-xr-x 2 root root 4096 May 22 10:07 vol3
```



	Name	Status	Root	Containing Aggregate	FlexClone	Avail	Used	Total	Files	Max Files
☐	vol0	online,raid0	✓	aggr0	-	82.9 MB	66%	242 MB	9.6 k	20 k
☐	vol1	online,raid4		aggr1	-	7.99 GB	0%	8 GB	3.31 k	307 k
☐	vol2	online,raid4		aggr1	-	7.5 GB	0%	7.5 GB	100	288 k
☐	vol3	online,raid4		aggr1	-	39.9 MB	0%	40 MB	105	1.52 k

Fig. 128: NetApp Volume List

We advise you to create a directory with your NetApp host name and mount volumes under it. For example, instead of using `/mnt` as the previous example was showing, using `/netapp-box` will simplify restore operation and file identification during restore. (Fig *NetApp and Volume Structure*).

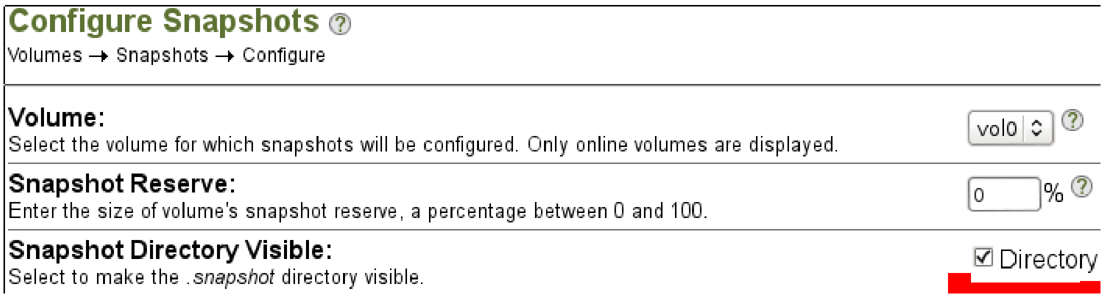
```
% ls -l /netapp-box
drwxr-xr-x 2 root root 4096 May 23 14:52 vol0
drwxr-xr-x 2 root root 4096 May 23 14:52 vol1
...
```

The following article will give you information about how to find optimal settings for NFS mounts http://nfs.sourceforge.net/nfs-howto/ar01s05.html

For example, you may want to mount NFS shares with the following options:

```
% mount -t nfs -o soft,intr nas1:/vol/vol3 /nas1/vol3
```

Snapshots should be visible in the `.snapshots` directory.



Configure Snapshots

Volumes → Snapshots → Configure

Volume: vol0

Select the volume for which snapshots will be configured. Only online volumes are displayed.

Snapshot Reserve: 0 %

Enter the size of volume's snapshot reserve, a percentage between 0 and 100.

Snapshot Directory Visible: [✓ Directory]

Select to make the *.snapshot* directory visible.

Fig. 129: Configuration for Snapshot Directory

Snapshot Management

The Incremental Accelerator for NetApp Plugin uses Snapshots to determine which files changed since the last backup. The Plugin will keep a number of Snapshots on your NetApp filer. Depending on your strategy, the Plugin will keep 3 to 4 Snapshots per volume.

Volume Management

To list all volumes available on the NetApp system, the following command might be used:

```
 # /opt/bacula/bin/snapmgr -V -f host -u user -p pass

or

 # /opt/bacula/bin/snapmgr -V -f host -u user -p pass -e vserver1

or

 # /opt/bacula/bin/snapmgr -s -V -f host.with.https -u user -p pass -e␣
↪vserver1
```

Where, `host` is the NetApp hostname or address, `user` is the username used to connect the NetApp and `pass` is the required password.

With NetApp c-mode, the `vserver` option is required.

In version prior to 12.4.1, `snapmgr` error messages can be found in `working/snapmgr.err` or `/tmp/snapmgr.err`. Starting with 12.4.1, error messages are automatically included in the Bacula Job log.

Estimate Information

The information returned by the `estimate` command will give you the list of all files that have changed. However, be careful when using the `estimate` command with the level Full (default). It will result in a heavy load of your NetApp filer if it contains many files.

Accurate Mode

By default, the Incremental Accelerator for NetApp Plugin backup result will be similar to what a traditional backup traversing the whole filesystem would do. Some operations (such as deleting or moving) on a directory (see figure *Directory Structure When Restoring (Without Accurate Support)*) will lead to *non accurate* directory structure when restoring.

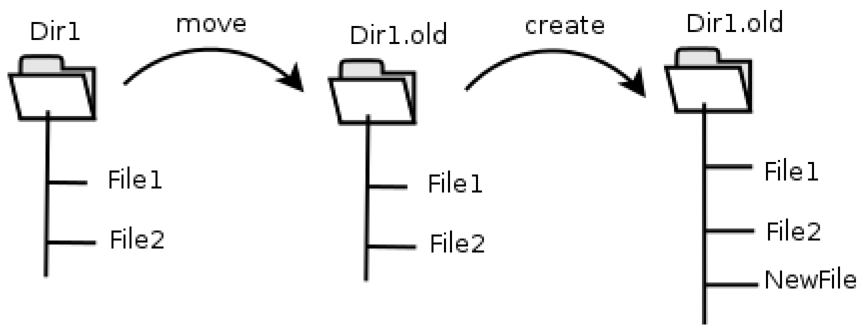


Fig. 130: Directory Structure When Restoring (Without Accurate Support)

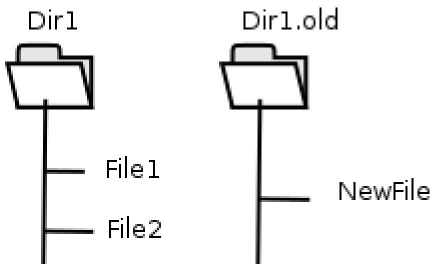


Fig. 131: Directory Structure When Restoring (Without Accurate Support)

This issue can be solved using Bacula's Accurate feature in your Job definition. However, be aware that the Accurate feature will consume additional memory on your File Daemon machine and CPU cycles on your catalog database system, and when dealing with million of files, you need to properly configure your File Daemon hardware. Using Bacula Accurate mode is transparent for the NetApp system.

Bacula Enterprise allows you to turn Accurate mode on or off in the Job Schedule. Doing that, you are able to, for example, turn the Accurate mode on only for weekly differential jobs. This can provide a good compromise by having a completely accurate weekly backup without the additional overhead each day.

Backup Information

During a backup, some essential information is stored in the `netapp-hfc.dat` file located in the `working` directory that is required for sequences of Jobs, so please take care not to remove the `netapp-hfc.dat` files.

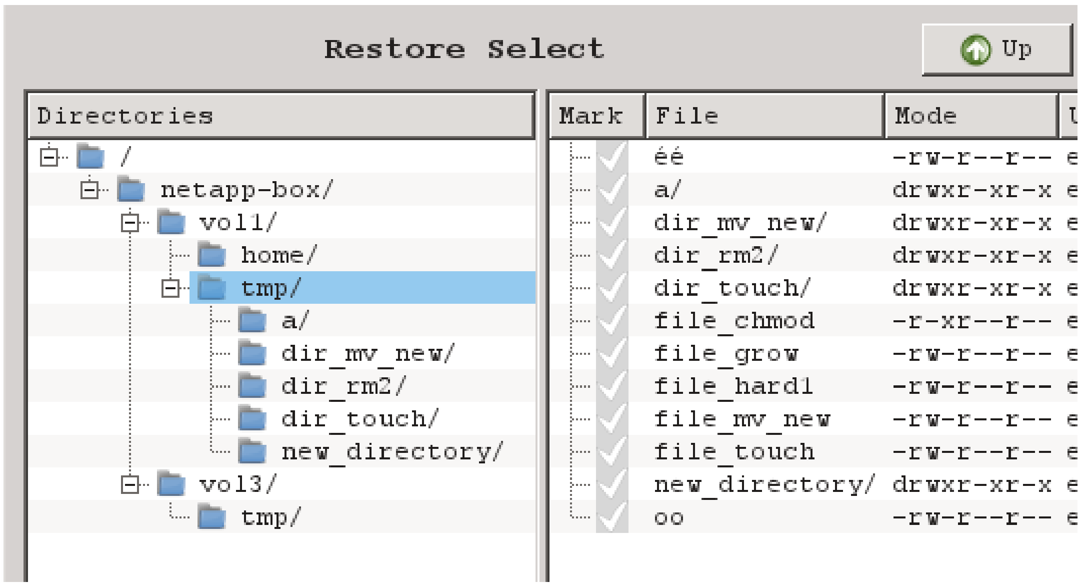
Restore Scenarios

During restoration, files will be presented in a simple virtual filesystem structure.

```
/<netapp host name>/<volume>/<files and directories>/...
```

For example (Fig *NetApp and Volume Structure*), the file `/etc/passwd` in the volume `vol0` of the NetApp device `netapp-box` will be stored as:

```
/netapp-box/vol0/etc/passwd
```



```
Directories
/
  netapp-box/
    vol1/
      home/
      tmp/
        a/
        dir_mv_new/
        dir_rm2/
        dir_touch/
        new_directory/
    vol3/
      tmp/
```

Mark	File	Mode	U
✓	éé	-rw-r--r--	e
✓	a/	drwxr-xr-x	e
✓	dir_mv_new/	drwxr-xr-x	e
✓	dir_rm2/	drwxr-xr-x	e
✓	dir_touch/	drwxr-xr-x	e
✓	file_chmod	-r-xr--r--	e
✓	file_grow	-rw-r--r--	e
✓	file_hardl	-rw-r--r--	e
✓	file_mv_new	-rw-r--r--	e
✓	file_touch	-rw-r--r--	e
✓	new_directory/	drwxr-xr-x	e
✓	oo	-rw-r--r--	e

Fig. 132: NetApp and Volume Structure

Limitations

- NetApp Data ONTAP version 7.3.5 and lower did not properly support Unicode characters in filenames outside of 7-bit ASCII. All NetApp versions after 7.3.5 support the full range of Unicode characters.
- The Incremental Accelerator plugin only works with NetApp appliances.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Nutanix HFC (Incremental Accelerator)

Overview

This user's guide presents various techniques and strategies to backup a Nutanix NAS having a large number of files with Bacula Enterprise.

Scope

This paper will present solutions for **Bacula Enterprise** 14.0 and later, which are not applicable to prior versions.

Presentation

The Nutanix Incremental Accelerator plugin is designed to simplify and the optimize backup and restore performance of your Nutanix NAS hosting a large number of files.

When using the plugin for Incremental backups, Bacula Enterprise will query the Nutanix REST API for a previous backup snapshot then quickly determine a list of all files modified since the last backup instead of having to walk through the entire filesystem. Once Bacula has the backup list, it will use a standard network share (such as NFS or CIFS) to access the files.

In order to compute the file list modified since the last backup, Bacula must store a few Snapshots on the Nutanix device. The Incremental Accelerator Plugin will manage the snapshot list to minimize resource usage.

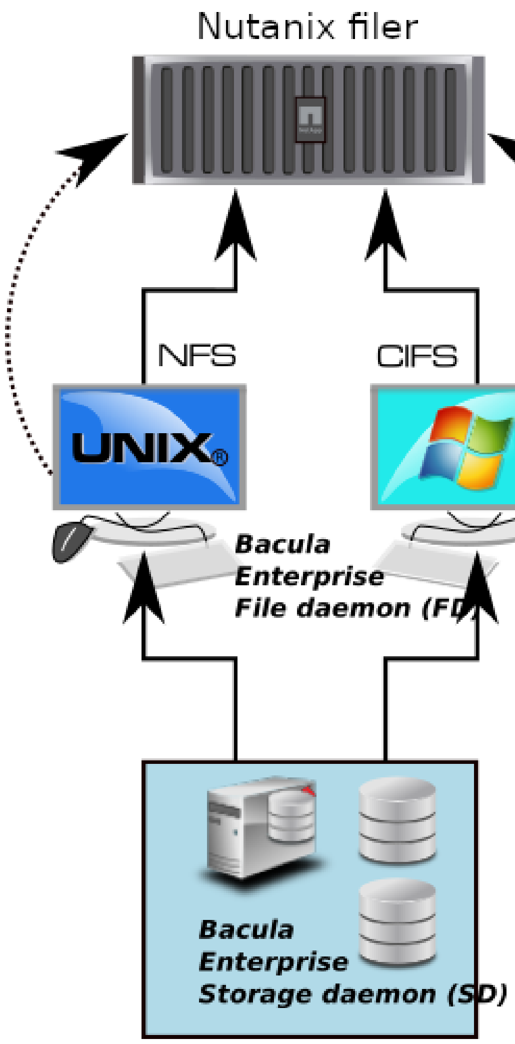

Nutanix filer
NFS
CIFS
UNIX®
Bacula Enterprise File daemon (FD)
Bacula Enterprise Storage daemon (SD)

Fig. 133: Nutanix Incremental Accelerator Feature

Installation

The Nutanix Incremental Accelerator plugin is available as a Bacula Enterprise package for all supported platforms. The package is comprised of:

- `/opt/bacula/plugins/nutanix-hfc-fd.so` Bacula Enterprise File Daemon plugin
- `/opt/bacula/bin/nutanix_backend` Bacula Enterprise Nutanix Manager

You must install this plugin on a Client machine that has network access to the Nutanix filer. Bacula Systems advises you to use the Client that resides on your Storage Daemon so that the File daemon to Storage daemon data transfers are made internally rather than across the network.

Configuration

As with all Bacula plugins, you must to specify the `Plugin Directory` directive in the `FileDaemon` resource of the `bacula-fd.conf` file.

```
FileDaemon {
  Name = test-fd
  ...
  Plugin Directory = /opt/bacula/plugins
}
```

Your Backup Job should be defined as:

```
Job {
 Name = "Nutanix-HFC"
 Client = storage-fd
 Fileset = FS_nutanix
 ...
}

Fileset {
 Name = FS_nutanix
 Include {
   Options {
     Signature = MD5
   }
   Plugin = "nutanix-hfc: host=nutanix-box password=pw exclude=vol0"
 }
}
```

In this example, all Nutanix volumes have to be mounted in the `/nutanix-box` directory (the `mount_base` plugin option permits to configure the base mount point).

```
% ls /nutanix-box
vol1   vol2   vol3   vol4
```

The Nutanix Incremental Accelerator plugin accepts the parameters listed in table 'Incremental Accelerator Plugin Options'.

Table 53: Incremental Accelerator Plugin Options

Option	Required	Default	Info	Example
host	Yes		NAS Hostname	host=192.168.0.1 host=nutanix-box
user	Yes	root	Username	user=root
password	Yes		Password	password=password
volume	No		Volume to	volume=vol0
include	No	*	Volumes to backup	include=vol1*
exclude	No		Volumes to exclude from backup	exclude=vol0
mount_base	No	/nutanix-box	Where to find the volume mounted using NFS / CIFS	mount_base=/nutanix-box

Note: Because the Nutanix Incremental Accelerator Plugin uses a dynamic `strip` Fileset option, the Fileset Include section should include **ONLY** the Nutanix plugin; mixing other plugins or normal files with the HFC Plugin is not supported.

External Password Storage

To avoid storing passwords in the Bacula Director configuration file, you can create a special configuration file located on the Client where the plugin is installed and store the password in it.

```
% cat /opt/bacula/etc/snapmgr.conf
nutanix-box:root = mypassword
```

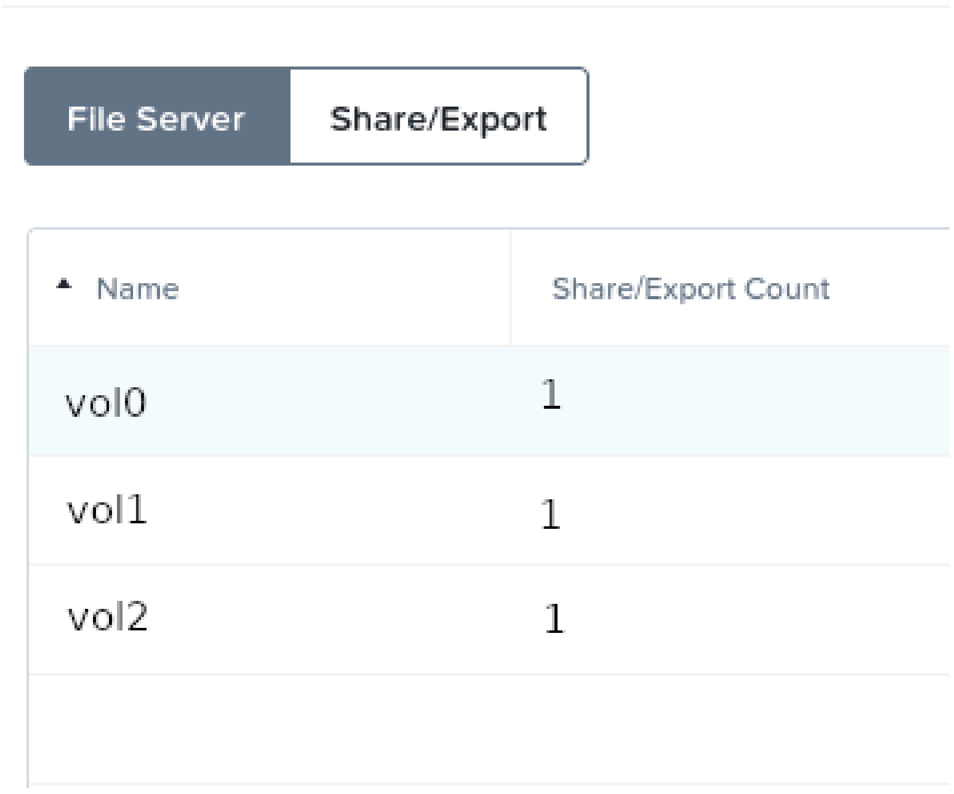
The first line will define the password for the root account on the nas box named “nutanix-box”. The Bacula Nutanix HFC Plugin will use this entry with the following plugin command:

```
plugin = "nutanix-hfc: host=nutanix-box user=root exclude=vol0"
```

You can have multiple lines in the `snapmgr.conf` file.

NFS Configuration

The Nutanix HFC plugin will detect the list of all volumes from the Nutanix device, then the plugin will assume that each volume is mounted and available under the `mount_base` directory. For example, if you have the following volumes defined on your Nutanix device (Fig *Nutanix Volume List*).



Name	Share/Export Count
vol0	1
vol1	1
vol2	1

Fig. 134: Nutanix Volume List

You will need to mount each of them under your `mount_base` directory. For example, in `/mnt`:

```
% ls -l /mnt
drwxr-xr-x 2 root root 4096 May 23 14:52 vol0
drwxr-xr-x 2 root root 4096 May 23 14:52 vol1
drwxr-xr-x 2 root root 4096 May 23 14:52 vol2
```

We advise you to create a directory with your Nutanix host name and mount volumes under it. For example, instead of using `/mnt` as the previous example was showing, using `/nutanix-box` will simplify restore operation and file identification during restore. (Fig *Nutanix and Volume Structure*).

```
% ls -l /nutanix-box
drwxr-xr-x 2 root root 4096 May 23 14:52 vol0
drwxr-xr-x 2 root root 4096 May 23 14:52 vol1
...
```

The following article will give you information about how to find optimal settings for NFS mounts http://nfs.sourceforge.net/nfs-howto/ar01s05.html

For example, you may want to mount NFS shares with the following options:

```
% mount -t nfs -o soft,intr nutanix-box:/vol/vol2 /nutanix-box/vol2
```

Snapshot Management

The Nutanix Incremental Accelerator plugin uses Snapshots to determine which files have changed since the last backup. The plugin will keep a number of Snapshots on your Nutanix filer. Depending on your strategy, the plugin will keep 3 to 4 Snapshots per volume.

Snapshots are visible in the hidden `.snapshot` directory at the top level directory in each volume share.

Snapshot Pruning

The Nutanix Incremental API is based on Snapshots to generate the list of the changes. Once a snapshot is no longer required, the Nutanix Incremental Accelerator HFC plugin will prune the snapshot.

The snapshot registry that is maintained by the plugin is stored in the Bacula working directory. The registry uses the Job name as a key to compute the snapshot chain.

Estimate Information

The information returned by the bconsole `estimate listing level=incremental` command will give you the list of all files that have changed. However, be careful when using the `estimate` command with the level Full (default if level=incremental is not specified). It will result in a heavy load of your Nutanix filer if it contains many files.

Accurate Mode

By default, the Nutanix Incremental Accelerator plugin backup results will be similar to what a traditional backup traversing the whole filesystem would do. Some operations (such as deleting or moving) on a directory (see figure *Directory Structure When Restoring (Without Accurate Support)*) will lead to *non accurate* directory structure when restoring.

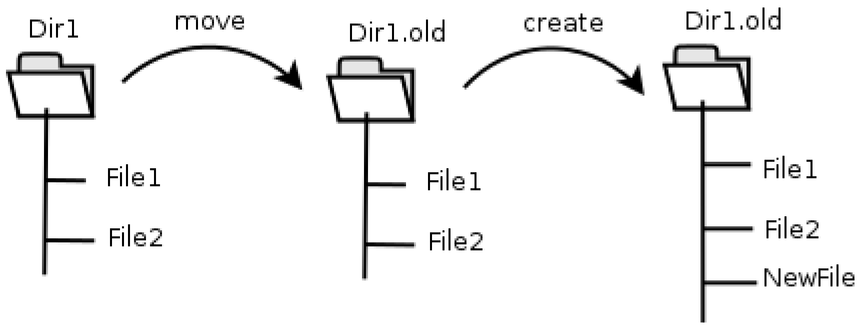


Fig. 135: Directory Structure When Restoring (Without Accurate Support)

This issue can be solved using Bacula's Accurate feature in your Job definition. However, be aware that the Accurate feature will consume additional memory on your File Daemon machine and CPU cycles on your catalog database system, and when dealing with million of files, you need to properly configure your File Daemon hardware. Using Bacula Accurate mode is transparent for the Nutanix system.

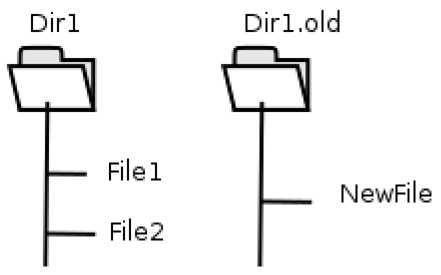


Fig. 136: Directory Structure When Restoring (Without Accurate Support)

Bacula Enterprise allows you to turn Accurate mode on or off in the Schedule. Using this method, you are able to, for example, turn the Accurate mode on only for weekly differential jobs. This can provide a good compromise by having a completely accurate weekly backup without the additional overhead each day.

Backup Information

During a backup, some essential information is stored in the `nutanix-hfc.dat` file located in the Bacula `working` directory that is required for sequences of Jobs, so please take care not to remove the `nutanix-hfc.dat` files.

Restore Scenarios

During restoration, files will be presented in a simple virtual filesystem structure.

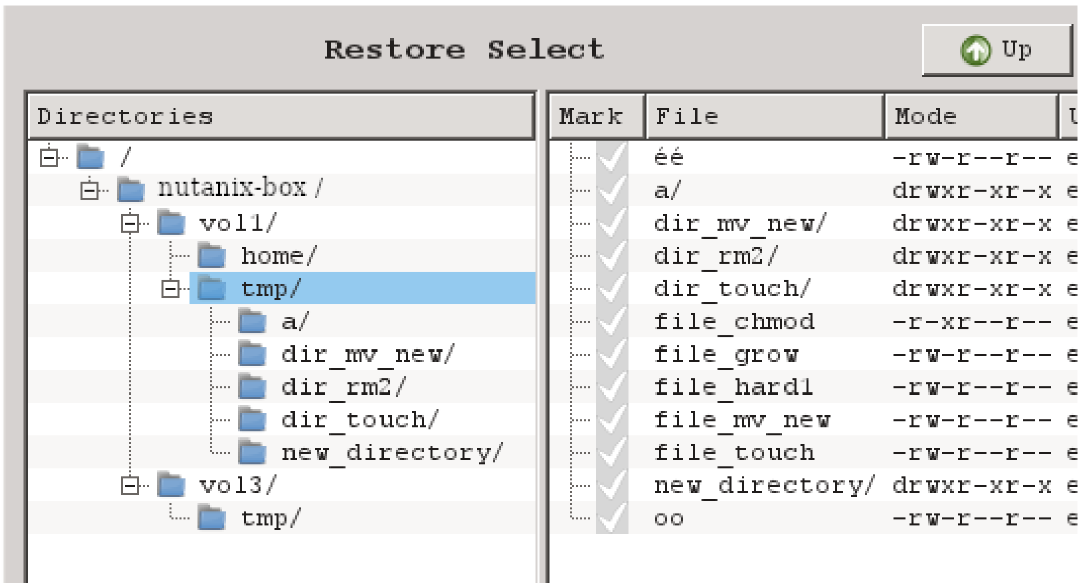
```
/<nutanix host name>/<volume>/<files and directories>/...
```

For example (Fig *Nutanix and Volume Structure*), the file `/etc/passwd` in the volume `vol0` of the Nutanix device `nutanix-box` will be stored as:

```
/nutanix-box/vol0/etc/passwd
```

Limitations

- As noted above, if you want to backup all Windows file attributes, you must mount the Nutanix volume using the CIFS protocol, and use a Bacula Windows File daemon with the Nutanix Incremental Accelerator plugin, which is not available at this time. Consequently restoring Windows files will restore only the file attributes that can be mapped into Unix file permissions.
- The Nutanix Incremental Accelerator plugin only works with Nutanix appliances.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.



```
Directories
/
  nutanix-box /
    vol1/
      home/
      tmp/
        a/
        dir_mv_new/
        dir_rm2/
        dir_touch/
        new_directory/
    vol3/
      tmp/
```

```
Mark  File            Mode
      éé              -rw-r--r--
      a/              drwxr-xr-x
      dir_mv_new/     drwxr-xr-x
      dir_rm2/        drwxr-xr-x
      dir_touch/      drwxr-xr-x
      file_chmod      -r-xr--r--
      file_grow       -rw-r--r--
      file_hardl      -rw-r--r--
      file_mv_new     -rw-r--r--
      file_touch      -rw-r--r--
      new_directory/  drwxr-xr-x
      oo              -rw-r--r--
```

Fig. 137: Nutanix and Volume Structure

6.3 Quobyte Plugin

This following chapter presents the strategy to backup Quobyte NAS with Bacula Enterprise.

Through subchapters, more in-depth information can be found about the following topics:

Scope

This plugin is available for **Bacula Enterprise 18.0** and later.

Features

The support for Quobyte is designed to simplify and optimize the backup and restore performance of your Quobyte storage volumes.

When using the File Daemon Snapshot feature, Bacula Enterprise will query the Quobyte server to detect tenants and volumes, and create snapshots for the current Job. The Quobyte network share (fuse.quobyte) is required to access files.

With Bacula Enterprise, snaphotting and backing up Quobyte volumes is supported in two different ways:

- The **bsnapshot** Bacula plugin now supports Quobyte volumes in addition to LVM, ZFS, and btrfs volumes.
- The new **quobyte-fd** plugin, when used in conjunction with the **scan_plugin** plugin, supports automatically identifying Quobyte volumes, and creating a Job/Fileset pair for each Quobyte volume identified.

Note: See Quobyte example for the *Scan Plugin*.

Installation

Snapshot support for Quobyte volumes is available with the standard Bacula Enterprise **bsnapshot** package for all supported platforms.

To perform snapshot backups of Quobyte volumes, it is necessary to install the **bacula-enterprise-snapshot package** on a Client machine that has network access to the Quobyte filer. It is advised to use the Client that resides on your Storage Daemon so that the File Daemon to Storage Daemon data transfers are made internally rather than across the network.

If you intend to use the Automation Center (or **scan_plugin**) with Quobyte to automatically create one Bacula Job/Fileset per Quobyte volume, you will need to install the **bacula-enterprise-quobyte-plugin** package.

It is also necessary to install the Quobyte client package on the Bacula Client machine and this system must have access to the Quobyte tenants and volumes (located under */quobyte* by default).

Note: For further information, refer to the Quobyte documentation.

Configuration

The following chapter presents information on Quobyte configuration.

Quobyte Configuration

The Quobyte filesystem must be mounted and available on your Bacula Client machine. The configuration can be done through the `/etc/quobyte/client.cfg` or when using systemd `/etc/quobyte/client-service.cfg`.

```
root# cat /etc/quobyte/client-service.cfg
# By default these are the options for the quobyte systemd service
# quobyte-client.service

multi-tenant
# Where to mount to
mount_point=/quobyte
registry=f7sskksge.myquobyte.net
uuid=yyyy-xxx-zzzz-aaaa-bbbbbbbbb
```

Mount the Quobyte filesystem to your machine:

```
root# mkdir /quobyte

root# systemctl enable --now quobyte-client

root# ls /quobyte
tenant_1
tenant_2
```

Install the `qmgmt` Quobyte tool and test the access.

Note: Refer to the Quobyte installation guide for more information.

Configuration for Built-in bsnapshot Support

Your Backup Job and Fileset should be defined as:

```
 Job {
  Name = "Quobyte"
  Client = storage-fd
  Fileset = quobyte
  ...
 }

 Fileset {
  Name = quobyte

  # Enable bsnapshot support for Quobyte volumes
  Enable Snapshot = yes

  Include {
   Options {
    Signature = sha1
    # Include all subvolumes
    onefs = no
   }
  File = /quobyte
}
```

Access to the Quobyte cluster can be configured in the `/opt/bacula/etc/bsnapshot.conf` bsnapshot configuration file.

```
root# cat /opt/bacula/etc/bsnapshot.conf
env="QUOBYTE_API=http://10.10.10.17:7860"
env="QUOBYTE_USER=admin"
env="QUOBYTE_PASSWORD=Thisisasecurepassword"
```

Note: Refer to the Quobyte documentation to use an API Key if needed.

In this example, all Quobyte volumes are available under the `/quobyte` directory.

It is possible to backup a subset of volumes or tenants by adapting the Fileset configuration:

```
Fileset {
 Name = quobyte_all_tenants

 # Enable Snapshot support for Quobyte
 Enable Snapshot = yes

 Include {
```

(continues on next page)

(continued from previous page)

```
  Options {
   Compression = zstd
   Signature = sha1
   onefs = no
  }
 File = /quobyte/tenant_1
 File = /quobyte/tenant_2
}
```

```
 Fileset {
  Name = quobyte_tenant_1_vol1

  # Enable Snapshot support for Quobyte
  Enable Snapshot = yes

  Include {
   Options {
    Signature = Sha1
  }
  File = /quobyte/tenant_1/vol1
}
```

Job output example using the **bsnapshot** plugin on a Quobyte volume:

```
JobId 879: Start Backup JobId 879, Job=QuobyteVol1.2024-01-16_12.01.
↪51_11
JobId 879: Connected to Storage "NULL" at wa-quobyte-dir:9103 with␣
↪TLS
JobId 879: Using Device "NULL" to write.
JobId 879: Connected to Client "wa-quobyte-dir-fd" at wa-quobyte-
↪dir:9102 with encryption
JobId 879: Connected to Storage at wa-quobyte-dir:9103 with␣
↪encryption
JobId 879: Wrote label to prelabeled Volume "Vol-0038" on TAPE␣
↪device "LTO" (/dev/lto)
JobId 879:    Create Snapshot for /quobyte/tenant_1/vol1
JobId 879:    Delete Snapshot for /quobyte/tenant_1/vol1
JobId 879: Elapsed time=00:36:33, Transfer rate=3.161 M Bytes/second
JobId 879: Sending spooled attrs to the Director. Despooling 634,551␣
↪bytes ...
JobId 879: Bacula Enterprise wa-quobyte-dir-dir 18.0.0 (11Jan24):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise␣
↪redhat (Core)
  JobId:                  879
  Job:                    QuobyteVol1.2024-01-16_12.01.51_11
  Backup Level:           Full
  Client:                 "wa-quobyte-fd" 18.0.0 (11Jan24) x86_64-pc-
↪linux-gnu,redhat,(Core)
  Fileset:                "quobyte_tenant_1_vol1" 2023-12-18 14:58:55
  Pool:                   "DiskBackup365d" (From Job resource)
  Catalog:                "BaculaCatalog" (From Client resource)
  Storage:                "TAPE" (From Command input)
```

(continues on next page)

(continued from previous page)

```
  Scheduled time:         16-jan-2024 12:01:51
  Start time:             16-jan-2024 12:02:53
  End time:               16-jan-2024 12:39:27
  Elapsed time:           36 mins 34 secs
  Priority:               10
  FD Files Written:       2,532
  SD Files Written:       2,532
  FD Bytes Written:       6,933,339,531 (6.933 GB)
  SD Bytes Written:       6,933,732,834 (6.933 GB)
  Rate:                   3160.1 KB/s
  Software Compression:   None
  Comm Line Compression:  6.9% 1.1:1
  Snapshot/VSS:           yes
  Encryption:             no
  Accurate:               no
  Volume name(s):         Vol-0038
  Volume Session Id:      325
  Volume Session Time:    1702658652
  Last Volume Bytes:      6,939,807,455 (6.939 GB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Configuration with the quobyte-fd Plugin

The **quobyte-fd** plugin can be used to optimize the Quobyte support. In this scenario, to maximize the throughput, the idea is to create one Bacula Job and Fileset per volume available on Quobyte. The Automation Center (or **scan_plugin**) was designed to automate the Bacula integration.

The Quobyte Plugin is installed with the **bacula-enterprise-quobyte-plugin** package using the standard package management tools.

Example Job and Fileset using the **Quobyte** Plugin:

```
Job {
 Name = "quobyte_tenant_1_vol1"
 Type = "Backup"
 Client = "wa-quobyte-dir-fd"
 Fileset = "quobyte_tenant_1_vol1"
 ...
}
```

```
Fileset {
 Name = "quobyte_tenant_1_vol1"
  Include {
  Options {
  Signature = sha1
  }
  Plugin = "quobyte: volume=\"/quobyte/tenant_1/vol1\" abort_on_error"
```

(continues on next page)

(continued from previous page)

```
 }
}
```

Job output example using the Quobyte Plugin on a Quobyte volume:

```
JobId 1665: Start Backup JobId 1665, Job=quobyte_tenant_1_vol1.2024-06-10_13.
↪05.11_49
JobId 1665: Connected to Storage "Dedup2Autochanger" at 10.0.99.122:9103␣
↪without encryption
JobId 1665: Using Device "Dedup2Autochanger_Dev7" to write.
JobId 1665: Connected to Client "wa-quobyte-dir-fd" at wa-quobyte-dir:9102␣
↪with TLS
JobId 1665: Enabling Snapshot for the current Fileset
JobId 1665: Connected to Storage at 10.0.99.122:9103 with TLS
JobId 1665: Volume "Dedup2_30d-0114" previously written, moving to end of␣
↪data.
JobId 1665: Ready to append to end of Volume "Dedup2_30d-0114" size=2,041,142
JobId 1665:    Create Snapshot for /quobyte/tenant_1/vol1
JobId 1665:    Delete Snapshot for /quobyte/tenant_1/vol1
JobId 1665: Elapsed time=00:00:29, Transfer rate=19.57 K Bytes/second
JobId 1665: Sending spooled attrs to the Director. Despooling 501,201 bytes ..
↪.
JobId 1665: Bacula Enterprise wa-quobyte-dir-dir 18.0.3 (05Jun24):
  Build OS:               x86_64-redhat-linux-gnu-bacula-enterprise redhat␣
↪(Core)
  JobId:                  1665
  Job:                    quobyte_tenant_1_vol1.2024-06-10_13.05.11_49
  Backup Level:           Full
  Client:                 "wa-quobyte-dir-fd" 18.0.3 (05Jun24) x86_64-redhat-
↪linux-gnu-bacula-enterprise,redhat,(Core)
  Fileset:                "quobyte_tenant_1_vol1" 2024-06-07 16:49:48
  Pool:                   "Dedup2_30d" (From Job resource)
  Catalog:                "BaculaCatalog" (From Client resource)
  Storage:                "Dedup2Autochanger" (From Pool resource)
  Scheduled time:         10-Jun-2024 13:05:11
  Start time:             10-Jun-2024 13:05:19
  End time:               10-Jun-2024 13:05:48
  Elapsed time:           29 secs
  Priority:               10
  FD Files Written:       2,536
  SD Files Written:       2,536
  FD Bytes Written:       42,498,943 (42.49 MB)
  SD Bytes Written:       567,655 (567.6 KB)
  Rate:                   1465.5 KB/s
  Software Compression:   98.7% 74.9:1
  Comm Line Compression:  55.6% 2.3:1
  Snapshot/VSS:           yes
  Encryption:             no
  Accurate:               yes
  Volume name(s):         Dedup2_30d-0114
  Volume Session Id:      28
  Volume Session Time:    1717576935
```

(continues on next page)

(continued from previous page)

```
  Last Volume Bytes:      meta: 2,667,617 (2.667 MB) aligned: 176,396,059␣
↪(176.3 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Automatic Object Integration

Since Bacula version 16.0.7, a new solution has been introduced, so that each object can be backed up separately with different Jobs to maximize the throughput and the resiliency. It is highly recommended to use this new solution for that purpose - *Automatic Object Integration (Scan Plugin)*. See an example for Quobyte.

7 Endpoint

Important: Endpoint solutions are used with the File Daemon.

7.1 BMR

WinBMR

Introduction

Bare Metal Recovery as part of a Disaster Recovery strategy allows for much quicker (re-)installations of original software and sophisticated automatic deployment and configuration management systems. For Windows environments, where automatic deployment and configuration management can become quite opaque, Bare Metal Recovery can be an important part of a Disaster Recovery plan. Bare Metal Recovery can save a lot of time and effort in virtualized environments, where all hardware looks identical to the guest operating systems (though it should be noted that, in virtualized environments, virtual machine backups backed by LUN snapshots in the SAN can be much more effective).

For these reasons, Bacula Systems has developed procedures and tools to provide Bare Metal Recovery capabilities to its customers. In this user's guide, we introduce the tools and related procedures for Windows systems by guiding you through a complete setup and test run of the Bacula Systems Windows Bare Metal Recovery procedure.

File-Based or Image-Based

In general, for any sort of Bare Metal Recovery, a complete backup of the original system is required. This can be done not only at the file level, where individual files are backed up, but also on the disk image level, where complete disk contents is backed up.

While both approaches have their pros and cons, we focus on the file based approach here. For image level backups, additional information can be found in the Bacula Systems white paper "VMware Virtual Machine Backup with Bacula Enterprise".

Installation Environment

Bacula Enterprise Director and Storage daemon components, used in the backup and restore of your Windows system, can run on any supported platform. Note that you will need to modify the Bacula Director configuration, so it may be reasonable to set up a test installation in your network and use that until you are satisfied that your production backup system will not be negatively affected by your work.

We assume that you already have Bacula Enterprise installed including the required network connectivity i.e. all routers and firewalls involved should allow Bacula traffic as needed. In particular, this means that you may need to allow connections from all machines you consider valid targets for BMR to the Bacula Director.

Since Bare Metal Recovery is used as a means to get critical systems up and running quickly, it is important to ensure that the procedures planned actually work. A good deal of this testing and fine-tuning today can be done in virtualized environments; however, Bacula Systems recommends testing on your physical hardware as well – only then can you be sure of your procedures.

Prerequisites

For testing and probably fine-tuning, we recommend that you define a "real" server system to use as the source of backups, and one unused server system as the target for a test of Bare Metal Recovery. You should only run this in virtual machines if your production environment is also virtualized. It may be necessary to pre-install drivers in the source system that match the hardware devices that will be required at recovery time if the expected replacement hardware differs from the existing. This can save time installing drivers during BMR.

The source and the target servers don't need to have Internet access, but this can make any troubleshooting procedure with our support easier.

Bacula Systems recommends that you go through the whole procedure to configure Recovery Media several times. We also suggest that you update your Disaster Recovery manual during those sessions to make sure your documented procedures really match what an operator will encounter. So, even though the final goal of doing Bare Metal Recovery is to save time, you should set aside at least one working day to get things set up, configured, created, tested and documented.

Installation

The installation of the Bacula Bare Metal Recovery for Windows Server is done in separate steps.

Downloading Files

- **winbmr-rescue-3.x.y.iso**, this ISO image is used to boot your system for recovery. You can customize it with the ISO configurator.
- **winbmr-iso-configurator.exe** This executable is used to pre-configure the ISO image with your Bacula Director settings and passwords. You can also configure network addresses and setup VNC for remote access.

The WindowsBMR Plugin has been included into the main Bacula Enterprise Client package since Bacula Enterprise versions 6.2.7 and 6.4.3, so no additional software installation is required on the client.

Configuring the WindowsBMR Rescue ISO

You can preset your Bacula configuration inside the Recovery Image itself to simplify the restore procedure. To customize your rescue CD-ROM, you must use `winbmr-iso-configurator.exe`. Start the program, locate your ISO file, and edit the fields you want to customize. When done, you can restart the program to check the values and change them whenever needed.

None of these fields are mandatory and setting a value for one will only pre-fill the field in the rescue application. You will still be able to modify this value at recovery time.

Any of these fields can also be initialized using the command line with the appropriate option. Run the program with the `-h` option to get a list of all these options. The name of the options are also included under parenthesis below for reference. You can see how the `GUI` looks like at figure **SetupConfig**.

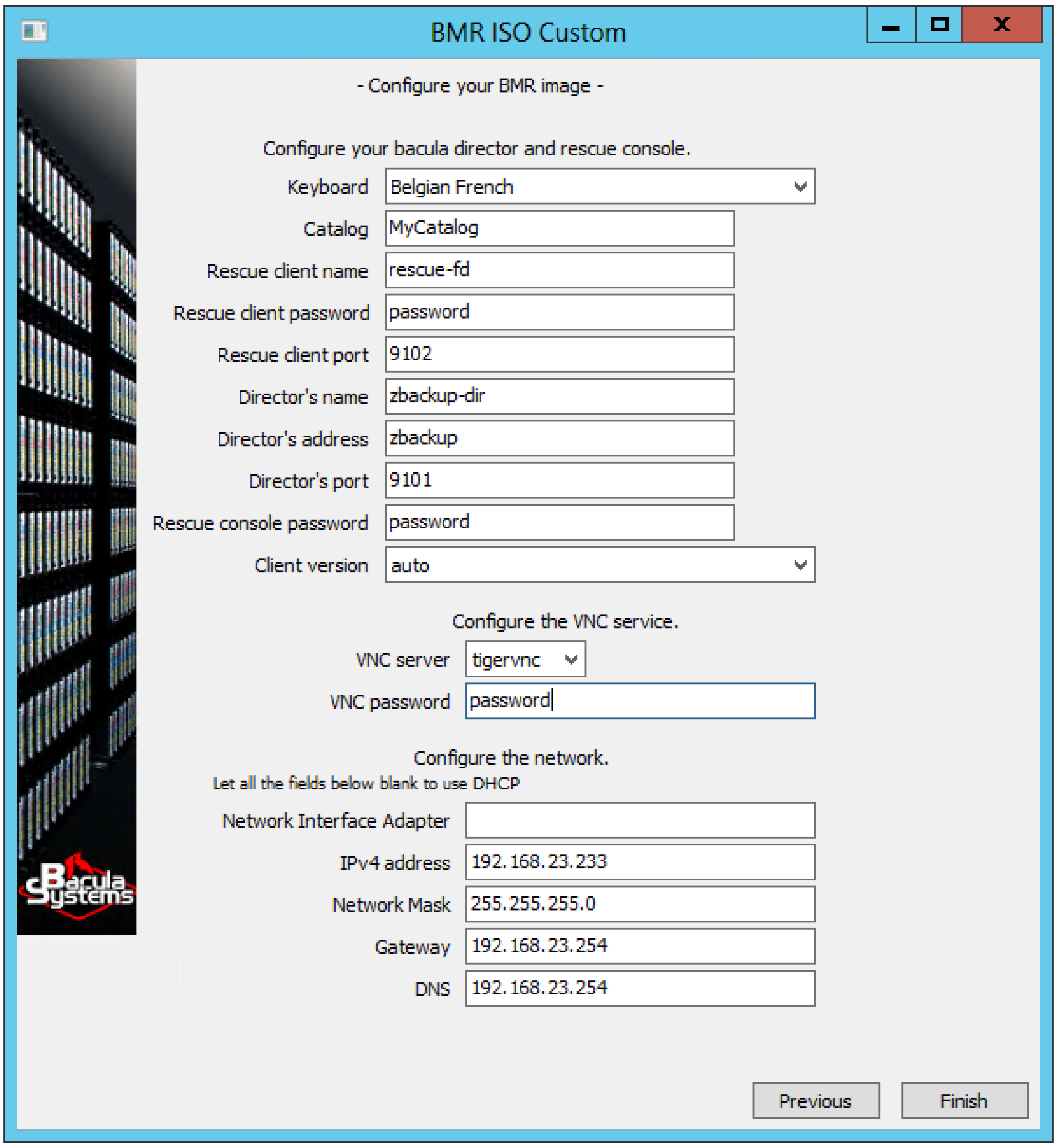
The first field is the keyboard.

- Keyboard(`-keyboard`, "US" by default). You can get a list of available keyboard layouts using the option `-list-kb`. Don't forget to use quote for name that contains spaces.

Then come the fields related to your Bacula setup. These are identical to the fields shown in the configuration screen of the rescue program, figure **BaculaConfig**.

- Your Director's Catalog (`-catalog`, set to "MyCatalog" in this document). If you have only one catalog, you can keep it blank because this catalog will be automatically selected by the rescue program even if you have given another name here.
- Rescue client name (`-cli-name`, set to "rescue-fd" in this document), this is the name of the **bacula-fd** client running on the machine being restored.
- Rescue client password (`-cli-pass`, set to "password" in this document)
- Rescue client port (`-cli-port`, 9102 by default)
- Director's name (`-dir-name`, set to "zbackup-dir" in this document))

- Director's address (`-dir-address`, set to "zbackup" in this document))
- Director's port (`-dir-port`, 9101 by default)
- Rescue console password (`-cons-pass`, set to "password" in this document)
- Client version (`-cli-version`, auto by default). To support a wider range of infrastructures, multiple versions of the Bacula FileDaemon are embedded into the image. You can select one specific version or select "auto" and let WindowsBMR choose the appropriate version regarding your director version at recovery time. You can get a list of available client versions using the option `-list-cli-version`). "auto" selects the latest version of the client compatible with your Director.



Configure your bacula director and rescue console.

Keyboard	Belgian French
Catalog	MyCatalog
Rescue client name	rescue-fd
Rescue client password	password
Rescue client port	9102
Director's name	zbackup-dir
Director's address	zbackup
Director's port	9101
Rescue console password	password
Client version	auto

Configure the VNC service.

VNC server	tigervnc
VNC password	password

Configure the network.
Let all the fields below blank to use DHCP

Network Interface Adapter	
IPv4 address	192.168.23.233
Network Mask	255.255.255.0
Gateway	192.168.23.254
DNS	192.168.23.254

Fig. 138: Customize the ISO Image

Next is the VNC configuration data:

- VNC Server (`-vnc-server`, set to "tigervnc" in this document). You can keep VNC disabled or choose between TightVNC and TigerVNC . If you have some refresh issues with one VNC server, try the other. However, only TigerVNC supports IPv6 and SSL.

- VNC password (`-vnc-password`, set to "password" in this document).

You can assign a static IP address to your machine. If you want to use DHCP instead let the `IPv4 address` field blank.

- Network Interface Adapter (`-net-ifname`, set to "<blank>" in this document). When left blank, the rescue program will assign the address to the first adapter that is connected.
- IPv4 address (`-net-ipaddr`, set to "192.168.23.233" in this document).
- Network Mask (`-net-netmask`, set to "255.255.255.0" in this document).
- Gateway (`-net-gateway`, set to "192.168.23.254" in this document).
- DNS (`-net-dns`, set to "192.168.23.254" in this document).

The command line interface has some special options

- `-iso FILENAME` use the given ISO image filename.
- `-no-gui` will not show the `GUI` interface.
- `-list-kb` shows a list of known keyboard layouts.
- `-list-cli-version` shows a list of known client versions.

For example to setup some parameters of your ISO image without starting the GUI, you can use :

```
C:\> winbmr-iso-configurator.exe --no-gui --iso winbmr-rescue-3.4.0.iso
   --cli-name=rescue-fd --cli-pass=xxx --cli-port=9102 --cons-pass=xxx
   --dir-address=10.10.1.2 --dir-name=bacula-dir.lan --dir-port=9101
```

Preparing your Bacula Installation

Now it is time to configure your Bacula installation to add the WindowsBMR services. We will not discuss the details of normal configuration of Bacula here because we assume that your Bacula Enterprise is already installed and configured.

The BMR procedures require you to add some resources to the Director's configuration. The required additions can be grouped into several categories. You may freely choose Passwords and names for File Sets, and Clients etc.

WindowsBMR Plugin for Windows Clients

As noted, in versions later than 6.2.6 or 6.4.2, Bacula Enterprise includes the WindowsBMR plugin with the main installer `bacula-enterprise-win32-x.x.x.exe` or `bacula-enterprise-win64-x.x.x.exe`.

A BMR Enabled Client Job

Below is typical Job resource for a BMR-enabled backup:

```
Job {
  Name = "WinBMR-job"
  JobDefs = "DefaultJob"
  Level = Incremental
  Client = your-windows-fd
```

(continues on next page)

(continued from previous page)

```
  File Set = "WinBMR-set"
}
```

For those users who have used a WindowsBMR prior to version 3.0, please be aware that the new WindowsBMR version 3 is packaged as a plugin and does not need or use the **Client Run Before Job** setting in the Job resource as it was the case with previous versions.

File Set Considerations

The second part of setting up your WindowsBMR job is to include the WindowsBMR plugin in your Fileset resource.

Since BEE version 16.0.9, it is recommended to use the new *version=2* option, see below. You don't need to upgrade your old Fileset, it will continue to work as before.

The File Sets used to back up Windows systems for BMR purposes must contain the **Plugin = winbmr** line to enable the WindowsBMR plugin. When option *version=2* is not used, the plugin searches for and includes all drive letters automatically.

It is important for the backup to include all the files that are required by the system. A common mistake is to exclude files based only on their file extension, as illustrated in:

```
WildFile = "*.ldf"
```

Instead, you should specify a full path to limit the exclusion to a directory:

```
WildFile = "D:/Data/*.ldf"
```

Although we do not recommend it, if you are careful, you may exclude some drives by using the **Exclude** subresource. However, take care to exclude only existing drives, because unused letters are assigned by WindowsBMR to "hidden" partitions during the time of the backup. If you exclude one of the letters used by WindowsBMR then the matching partition will not be backed up and the system will probably not boot at restore time.

```
Fileset {
  Name = "WinBMR-set"
  Enable VSS = yes # default, but make sure it's enabled!
  Include {
    Options { Signature = MD5 }
    Plugin = winbmr
    # Plugin = "winbmr:exclude=F,G"  # not recommended
  }
}
```

Above, we show an example of the "exclude" option. Be careful to use the exact same notation, the double quotes are important.

The new *version=2* option has been designed to be used specifically in conjunction with the *File=/* directive inside the Fileset.

When the option *version=2* is used, the *winbmr* plugin will:

- not modify the Fileset that will be used by the File Daemon. This is a major difference with the normal *winbmr* behavior. When the "version=2" option is not used, the normal *winbmr* behavior will add the system/recovery disk if there is any present in the system.
- ignore any *exclude=XXX* option of the plugin as it cannot apply it to the Fileset because of the previous rule. You must use an *exclude option block* instead.

- mount any system/recovery partitions (if not already mounted) before the beginning of the backup and dismount it after the end of the backup (if it was not mounted). This behavior doesn't change.
- If the directive *File=/* is used, Bacula will include any system/recovery partition that is mounted at the time of the backup. Else, you must include it yourself usually using a *File=T:* directive in the Fileset. Be careful, the T: letter depends on the drive letters available when the plugin mounts the partition.
- Copy the content of the *EFI* partition (if any) into "C:BaculawinbmrpartitionsEFI". You must of course be sure that C: is backed up. *File=/* always backs up C: (if it is not excluded). If that's not the case, you have to add the *File="C:"* directive to your *Fileset* yourself.

Remember that in a Windows *Fileset*, the "File=/" directive will back up all the drive letters. If *OneFs=no* is specified (in the **last** *Option* block because others are ignored), all the mount points below the selected drives will also be included and recursively. A drive or a mount point will be selected only if it is not listed in the Exclude block. VSS will only snapshot the volumes (drives or mount point) that have been selected.

A typical Fileset using this new option looks as follows:

```
Fileset {
      Name = "Winbmr2-Fileset"
      Enable VSS = yes
      Include {
             Options {
                    signature = MD5
                    compression = LZO
                    # Implicitly Include directories that are mount point for␣
↪other Filesystems
                    OneFs = no
             }
             Options {
                # In this Option block, excludes files and directories
                    # that are not mount points or drives
                    Exclude = yes
                    OneFs = no # only the last one matter

        WildDir = "*/$Recycle.Bin"
        WildDir = "*/$RECYCLE.BIN"
        WildDir = "*/System Volume Information"
     }
     # use the new "version=2 " winbmr option with the File=/ directive
     Plugin = "winbmr: version=2"
     File = "/"
  }
  Exclude {
     # exclude mount points and drives here
     File="F:"
     File="C:/mount"
  }
}
```

This Fileset will also work well without the *winbmr* plugin line, of course you will lose the "BMR" feature, but the backup will be valid.

If for some reason you don't want to use *File=/*, you can replace it with something like:

```
File = "C:" # always if you are doing a winbmr backup
File = "D:"
File = "T:" # if you have a system partition, you must include it manually␣
↪and verify that the drive name match the one used by the winbmr plugin
```

And to answer a question that you have: If you have 3 volumes with mount points that are stacked on each other:

```
H:/ (volume1)
H:/mnt (volume2)
H:/mnt/submnt (volume3)
```

and a Fileset like this one:

```
Fileset {
  Name ="XXX"
  Enable VSS = yes
  Include {
    Options {
      Exclude = yes
      OneFs = no
    }
    File="H:"
  }
  Exclude {
    File="H:/mnt"
  }
}
```

it will back up the content of H:/ and H:/mnt/submnt but not goes into H:/mnt. It will create a snapshot for H:/ and H:/mnt/submnt but not for H:/mnt

Note that there was a bug in the *winbmr* plugin in version prior to 16.0.12 when using the "estimate" command. The system/recovery filesystem that have been mounted by the plugin were not dismounted. This doesn't not disturb the next *winbmr* backups. Since version 16.0.12, "estimate" doesn't mount any filesystem and doesn't update the partition layout.

Restore Job Considerations

A `restore` Job resource which does not have any `RunScript` directives is needed by the WindowsBMR restore process. Which restore job to use can be chosen by the user (see figure **SelectJob**) from a list of such jobs. The single restore job of a default Bacula configuration meets this requirement.

Handling of Windows Volume Mount Points

Note that all paths containing a mount point must be named using ASCII characters exclusively, as otherwise at restore time the mount point will not be re-created. This limitation will be removed in a future WindowsBMR version.

If you cannot assign a drive letter to your Volume, you must explicitly add the path to your mount point directory to your Fileset resource. At restore time choose "Manual Partitioning", create the Volume and mount the file system to its correct location before starting the restore. Do the same if the path of your mount point contains non-ASCII characters.

How the WindowsBMR Works

During backup, the WindowsBMR plugin analyzes host disks and partitions. It creates the directory `C:/Bacula/winbmr` and copies certain files and directories needed at restore time to that location (only a few MB). If a "Recovery" or a "System Reserved" partition is found, the plugin assigns an unused drive letter (usually the first free letter starting at T:) to it for the time of the backup. This drive letter is released at the end of the backup. If the system is EFI-enabled, the EFI partition is automatically mounted, its contents copied to `C:/Bacula/winbmr/partitions/EFI`, and the partition then unmounted.

The plugin adds all static volumes that have a drive letter assigned to the backed-up Fileset. As mentioned above you may exclude some drive letters using the "exclude" option, but be careful to not exclude an important drive or an unused letter (like T:) which the plugin might use for the "hidden" partitions.

Here are the directories and files you will find in the **C:/Bacula/winbmr** directory:

```
C:/Bacula/winbmr
C:/Bacula/winbmr/data
C:/Bacula/winbmr/data/bcd.bak
C:/Bacula/winbmr/data/bcd_active.txt
C:/Bacula/winbmr/data/bcd_all.txt
C:/Bacula/winbmr/data/disklayout.txt
C:/Bacula/winbmr/data/the_bacula_rescue_for_windows
C:/Bacula/winbmr/partitions
C:/Bacula/winbmr/partitions/EFI
...
```

Attention: The rescue process is case sensitive and expects to find the above files in this exact directory name: with an uppercase 'B' and lowercase 'w'.

The BMR Rescue Access

The Director configuration entries presented next are required to run actual BMR Restore Jobs.

The client defined below will run from the Recovery Media during the WindowsBMR restore process and is a special client that is used for all WindowsBMR restores and thus should have a different name from any of the clients used for backup. The "0.0.0.0" address will be updated during the recovery process with the correct address of the host (client) on which you are restoring. We recommend that you use the name `rescue-fd` and to not use this particular client for any non-BMR operation. Under BWeb, go to the configuration part under Console and click on the `Set BMR Console` wizard to create all necessary resources called `rescue-fd`.

```
Client {
  Name = rescue-fd        # cliname, same as Console
  Password = "xxxpassxxx" # clipass
  FDPort = 9102           # cliport

  Address = 0.0.0.0      # dummy address
  Catalog = MyCatalog
  File Retention = 30 days
  Job Retention = 6 months
  Auto Prune = Yes
}
```

The named console defined below is used for BMR operations. It is important that it allows access to all needed Catalogs (Bacula Systems recommends using only one Catalog, unless there are some very specific requirements involved), Clients, Jobs and Locations. Limiting Access to Commands, Storage, Jobs, Pools and File Sets can be safer, but it depends on your configuration. We will show how to tune access control lists (ACLs) later in this manual. For a first try we recommend to set all ACLs to “*all*”.

```
Console {
  Name = rescue-fd          # cliname, same Client
  Password = "xxxpassxxx"   # conspass

  CommandACL = *all*
  ClientACL = *all*
  CatalogACL = *all*
  JobACL = *all*
  StorageACL = *all*
  ScheduleACL = *all*
  PoolACL = *all*
  FilesetACL = *all*
  WhereACL = *all*
  # The next two ACLs are required when using
  # Bacula Enterprise 8.8.0 and above
  UserIdACL = *all*
  DirectoryACL = *all*
  # This last ACL is available when using
  # Bacula Enterprise 8.8.0 and above but is not required
  RestoreClientACL = *all*
}
```

The client and console resources shown above **must** have the same name. Also as mentioned above it would be a bad idea to use the name of a production server as doing so makes it possible to accidentally do a BMR restore to a production server thereby destroying it.

Using the Recovery Media

You now have a CDROM and an ISO image. For testing purposes, on physical servers, the CD-ROM is the best choice, while in virtual machines, the ISO image should be used. It is also possible to create a bootable USB flash drive from the ISO. Creating a bootable USB device from the ISO is described in a later chapter. If you need a bootable USB device please create it now before proceeding.

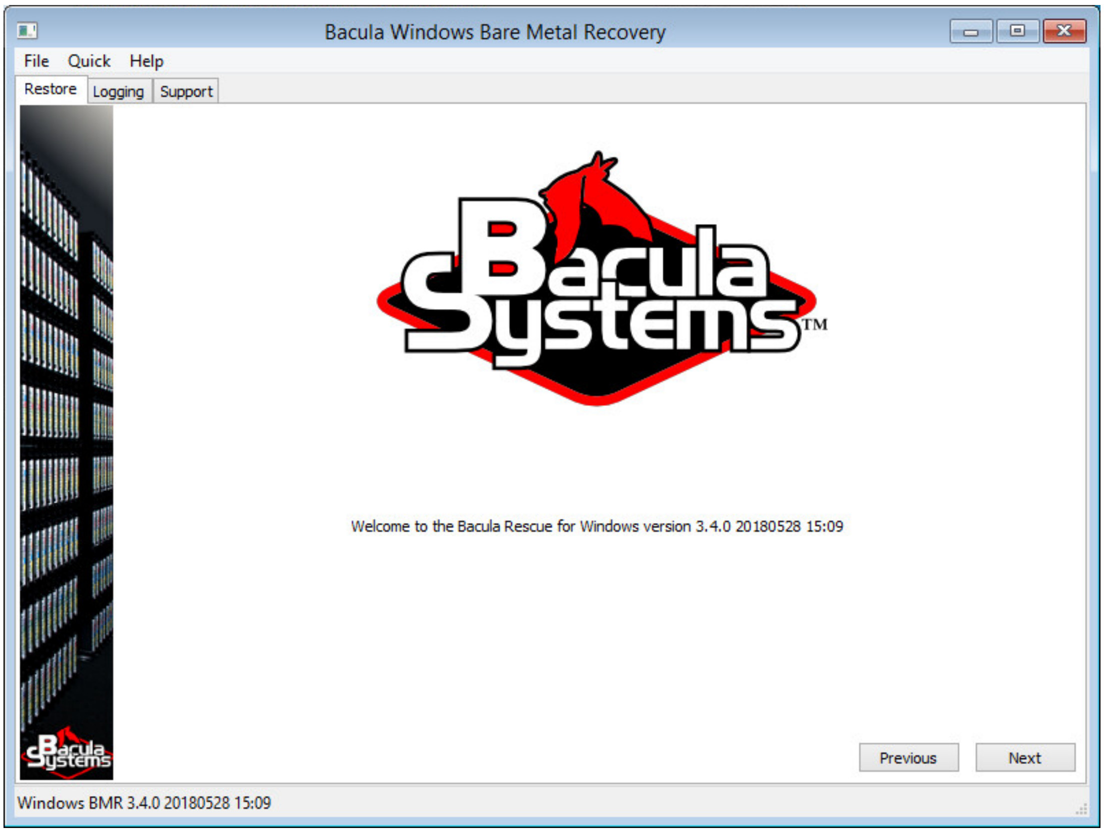
Starting the Recovery

Proceed by attaching the Recovery Media to your test recovery system, which should be set up with an empty disk (or several empty disks, if that matches your production needs). Make sure the test system will boot from the Recovery Media by setting the BIOS options accordingly. It may be necessary, for example, to disable *secure boot* while performing the recovery and re-enable it afterwards.

The actual recovery procedure is initiated and configured manually; although this has the disadvantage that physical or remote console access to the system is required, we consider this to be the safer approach for a procedure that could otherwise lead to data loss.

When your client to recover boots, it will start the minimal Windows on the Recovery Media, set up its network, and display the Bacula splash screen: (Figure **Splash**)

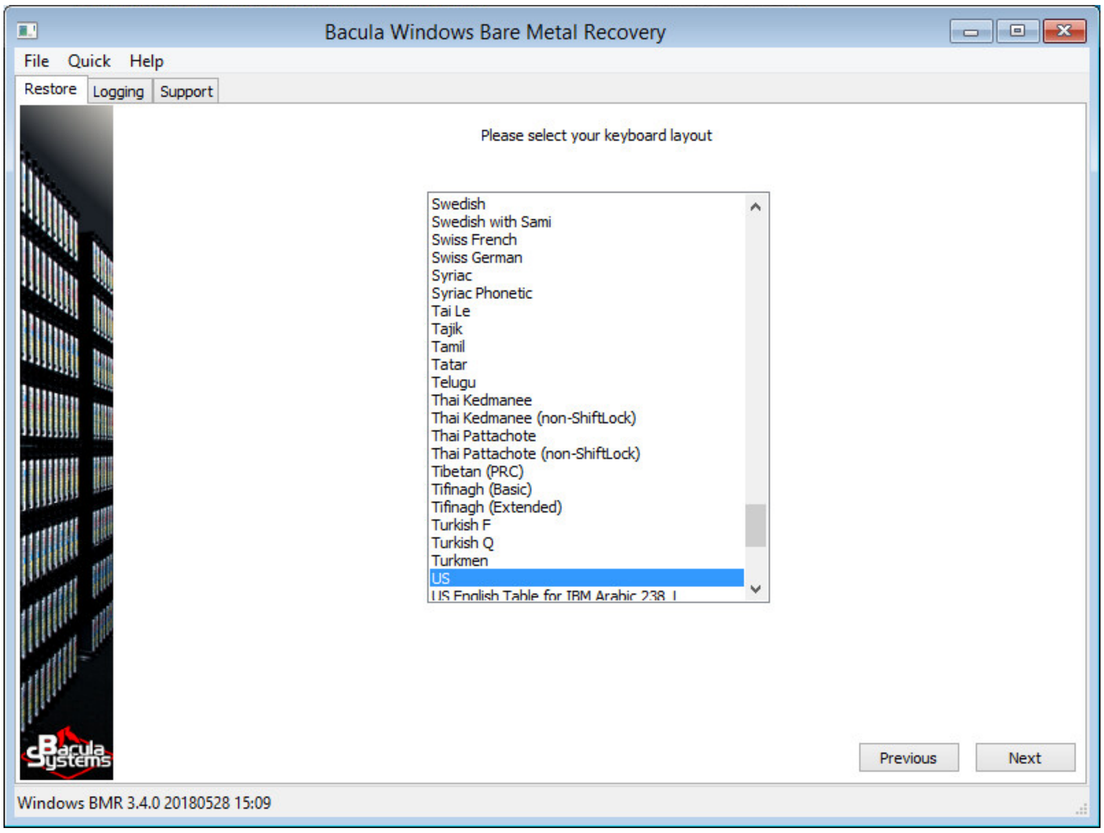
Click “Next” and select your keyboard layout in the list. You can double click on the line to avoid to have to click the “Next” button. After that the program will restart to take the keyboard change into account and return to prompt you.



Welcome to the Bacula Rescue for Windows version 3.4.0 20180528 15:09

Windows BMR 3.4.0 20180528 15:09

Fig. 139: Recovery System Booted



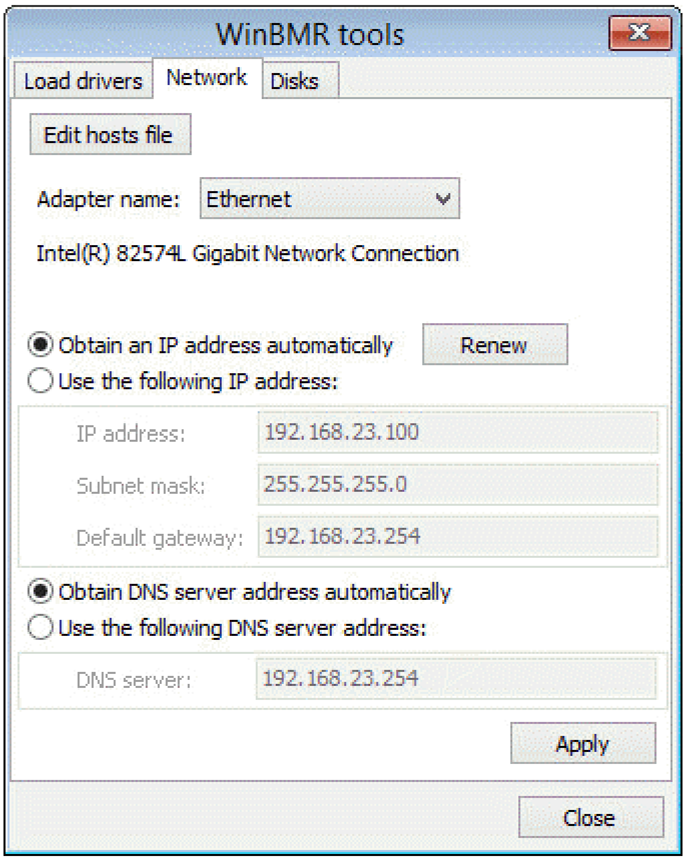
Bacula Windows Bare Metal Recovery
File Quick Help
Restore Logging Support
Previous
Next
Windows BMR 3.4.0 20180528 15:09

Please select your keyboard layout

Swedish
Swedish with Sami
Swiss French
Swiss German
Syriac
Syriac Phonetic
Tai Le
Tajik
Tamil
Tatar
Telugu
Thai Kedmanee
Thai Kedmanee (non-ShiftLock)
Thai Pattachote
Thai Pattachote (non-ShiftLock)
Tibetan (PRC)
Tifinagh (Basic)
Tifinagh (Extended)
Turkish F
Turkish Q
Turkmen
US
US English Table for IBM Arabic 238_L

Fig. 140: Select your Keyboard Layout

If your network is not configured with a DHCP server or if you did not pre-configure your ISO image with static addresses, you must configure a static IP address using the Network setup dialog box started from the Tools menu item in the File Menu.



Adapter name: Ethernet

Intel(R) 82574L Gigabit Network Connection

(●) Obtain an IP address automatically
( ) Use the following IP address:

IP address: 192.168.23.100
Subnet mask: 255.255.255.0
Default gateway: 192.168.23.254

(●) Obtain DNS server address automatically
( ) Use the following DNS server address:

DNS server: 192.168.23.254

Fig. 141: Configure the Network

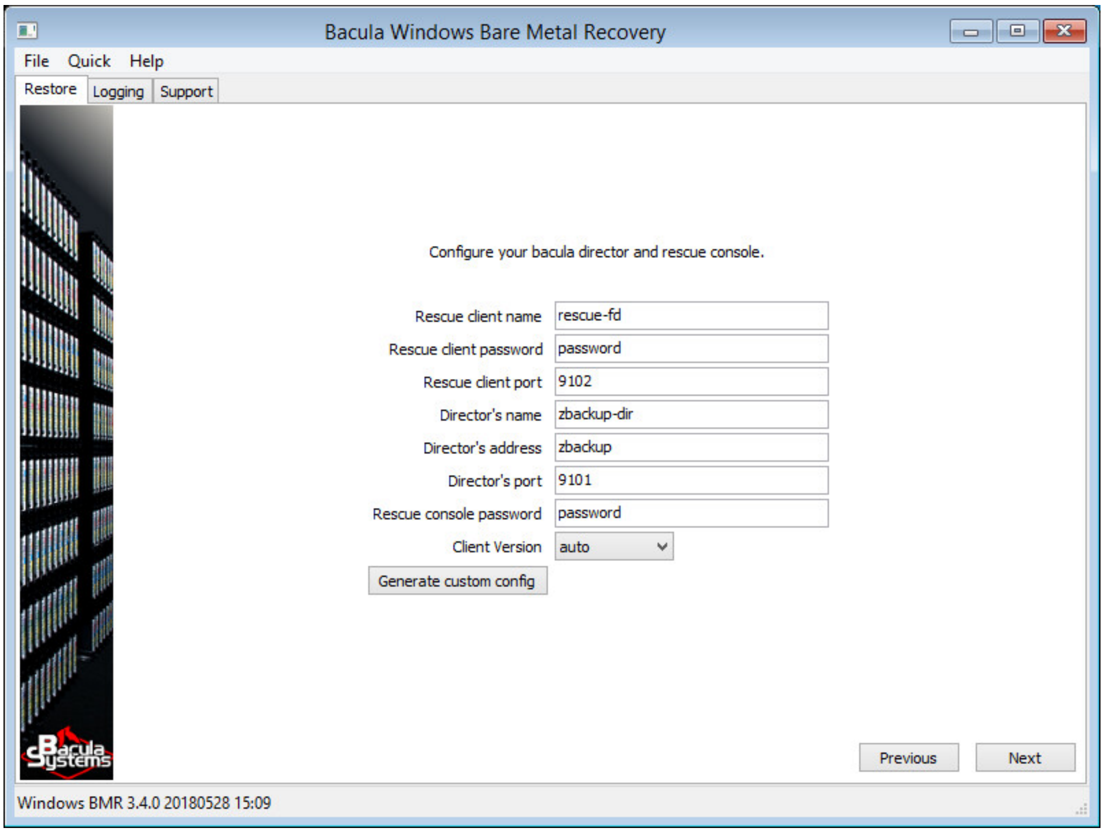
WindowsBMR now lets you review and modify the Bacula configuration (Figure **BaculaConfig**) that is stored on the Recovery Media. You can pre-configure your Recovery Media using the `configurator`. See the *Configuring the WindowsBMR Rescue ISO* to learn how and find some information about the available configuration options.

If the host cannot connect to your Director or if the Director cannot connect back to the recovery file daemon you will get an error message. Please read the message carefully to know which parameter is wrong and correct it. When your setup is OK, the program will go to the next screen.

When your director has more than one Catalog configured, WindowsBMR shows you this extra screen (Figure **SelectCatalog**) to select the catalog you want to use.

Then you will be presented with a screen to select the client you wish to recover (figure **SelectClient**). You can type some characters in the filter box to reduce the number of clients in the list. The match doesn't need to start at the beginning of the client name, but any sub string can match the filter. No wildcard characters are supported.

The next screen shows the date and time stamps of all successful backups run from the selected client. The list displays the date, the job level, `Incremental` or `Full`, the JobId and the name of the Job. Unfortunately we cannot filter the jobs to display only valid BMR-enabled backups in a reasonably short amount of time. If the selected job is not a valid BMR-enabled backup, the program will ask you to choose another one. At the bottom of the screen, the program will automatically select a restore job

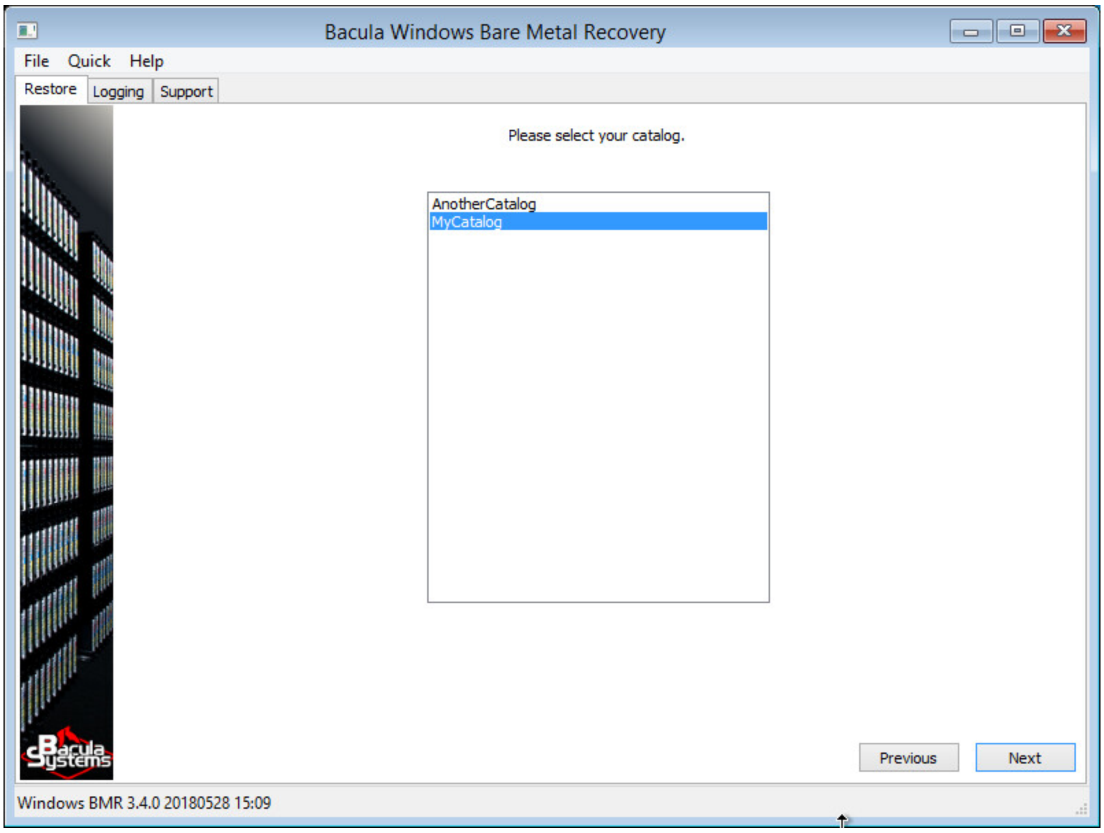


Configure your bacula director and rescue console.

Rescue client name	rescue-fd
Rescue client password	password
Rescue client port	9102
Director's name	zbackup-dir
Director's address	zbackup
Director's port	9101
Rescue console password	password
Client Version	auto

Windows BMR 3.4.0 20180528 15:09

Fig. 142: Configure Bacula

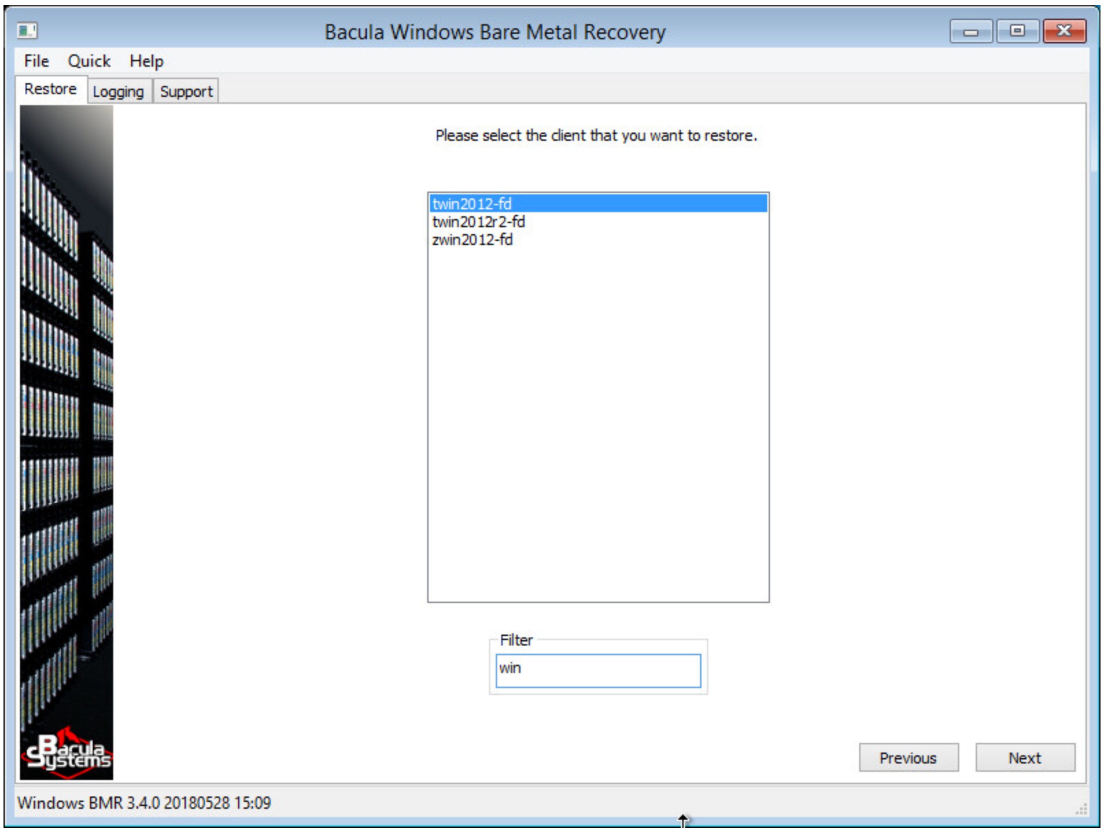


Please select your catalog.

AnotherCatalog
MyCatalog

Windows BMR 3.4.0 20180528 15:09

Fig. 143: Catalog Selection



Please select the client that you want to restore.

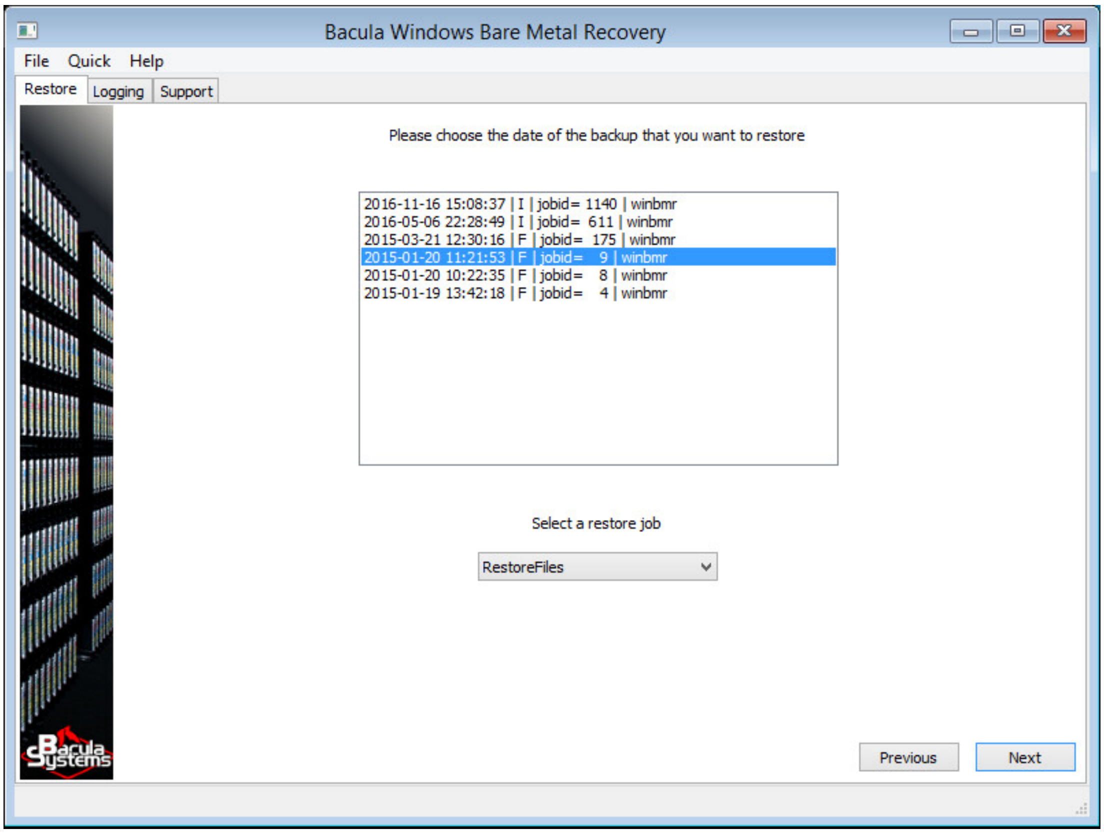
twin2012-fd
twin2012r2-fd
zwin2012-fd

Filter: win

Windows BMR 3.4.0 20180528 15:09

Fig. 144: Client Selection

template; you can choose a different one if needed. (Figure **SelectJob**).



Please choose the date of the backup that you want to restore

```
2016-11-16 15:08:37 | I | jobid= 1140 | winbmr
2016-05-06 22:28:49 | I | jobid=  611 | winbmr
2015-03-21 12:30:16 | F | jobid=  175 | winbmr
2015-01-20 11:21:53 | F | jobid=    9 | winbmr
2015-01-20 10:22:35 | F | jobid=    8 | winbmr
2015-01-19 13:42:18 | F | jobid=    4 | winbmr
```

Select a restore job

RestoreFiles

Fig. 145: State to Restore

Information, both from the backed up and the local system, is then collected and the next three screens will help you create partitions, format them and select to which of them the data will be restored.

Managing Disk Drives

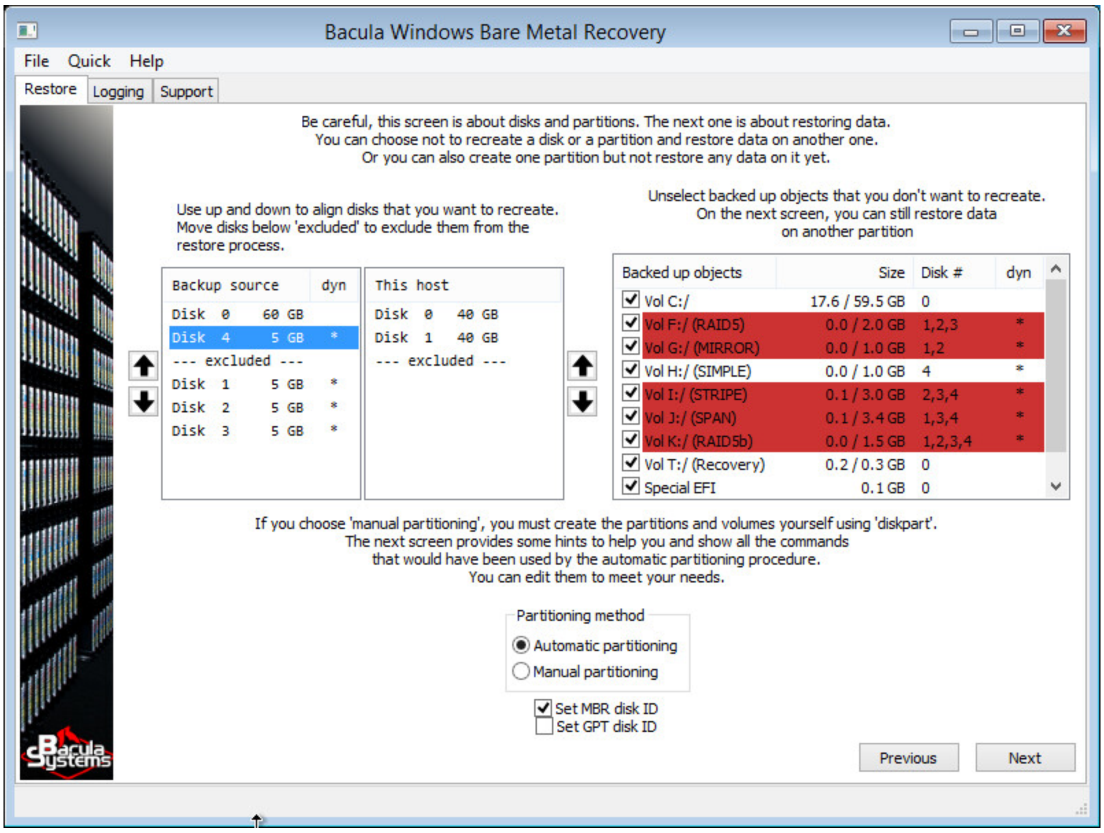
The BMR process will recommend that you remove any existing dynamic disk volumes on the destination disk and allow automatic partitioning. You can decide to use your existing partitions instead if you are confident they are correctly configured.

The next three screens may look difficult to understand, but most of the time the default settings are what you want and simply clicking “Next” will do the job. If the number of disks on the target host is different or if the disks are too small, then you will have to modify the configuration by making appropriate selections on these screens.

The screens have been designed to be very flexible and also allow partitioning operations to be done manually and finishing the restore within the tool. In most cases, however, the program will work correctly without extra operations.

To understand the details of these three screens it is important to be aware that a “Volume” is located on a partition **or a dynamic disk** and has been formatted and has a drive letter assigned. WindowsBMR can only handle “Volumes” that have a drive letter.

Some partitions that are not backed up by WindowsBMR (because they are not required), like the MSR, are not shown in the screens below, but they will be created because they are required. Other “system” partitions like the “EFI”, “System Reserved” and the “Recovery” partitions are usually hidden and not seen by the user. These partitions may be important in the boot process and are backed up by WindowsBMR. These can be excluded and merged with the C: drive at restore time. Most of the time, it is advisable to keep them in the restore process and the recovery will be successful. (Figure **Partitioning**).



Be careful, this screen is about disks and partitions. The next one is about restoring data.
You can choose not to recreate a disk or a partition and restore data on another one.
Or you can also create one partition but not restore any data on it yet.

Use up and down to align disks that you want to recreate.
Move disks below 'excluded' to exclude them from the restore process.

Unselect backed up objects that you don't want to recreate.
On the next screen, you can still restore data on another partition

```
Backup source      dyn
Disk  0    60 GB
Disk  4     5 GB    *
--- excluded ---
Disk  1     5 GB    *
Disk  2     5 GB    *
Disk  3     5 GB    *
```

```
This host
Disk  0    40 GB
Disk  1    40 GB
--- excluded ---
```

Backed up objects	Size	Disk #	dyn
✓ Vol C:/	17.6 / 59.5 GB	0	
✓ Vol F:/ (RAID5)	0.0 / 2.0 GB	1,2,3	*
✓ Vol G:/ (MIRROR)	0.0 / 1.0 GB	1,2	*
✓ Vol H:/ (SIMPLE)	0.0 / 1.0 GB	4	*
✓ Vol I:/ (STRIPE)	0.1 / 3.0 GB	2,3,4	*
✓ Vol J:/ (SPAN)	0.1 / 3.4 GB	1,3,4	*
✓ Vol K:/ (RAID5b)	0.0 / 1.5 GB	1,2,3,4	*
✓ Vol T:/ (Recovery)	0.2 / 0.3 GB	0	
✓ Special EFI	0.1 GB	0	

If you choose 'manual partitioning', you must create the partitions and volumes yourself using 'diskpart'.
The next screen provides some hints to help you and show all the commands
that would have been used by the automatic partitioning procedure.
You can edit them to meet your needs.

Partitioning method
● Automatic partitioning
○ Manual partitioning

✓ Set MBR disk ID
☐ Set GPT disk ID

Fig. 146: Disk Matching and Partitioning

The first of the three screens mentioned is about disk matching and partitioning. The goal is to match disks of the source host (original machine backed up) with disks on the target host (machine to which you are restoring which is usually the same as the source host but could be a replacement machine). It is possible that disks on the target host are of different capacity and appear differently numbered than those of the source host.

Use the two lists on the left to do the matching. Use the arrows on both sides to move disks up and down. Disks below “–excluded–” will be ignored. If the exclusions are on the left, “volumes” that are located on them (the source host) will not be created and are displayed in red in the list at the far right.

Disks below the “–excluded–” bar on the right will be left untouched (if you have data on these disks they will normally be unchanged by the restore process).

Be careful that the bootable disk on the source is aligned with one on the target. Usually both “disk 0” have to be aligned.

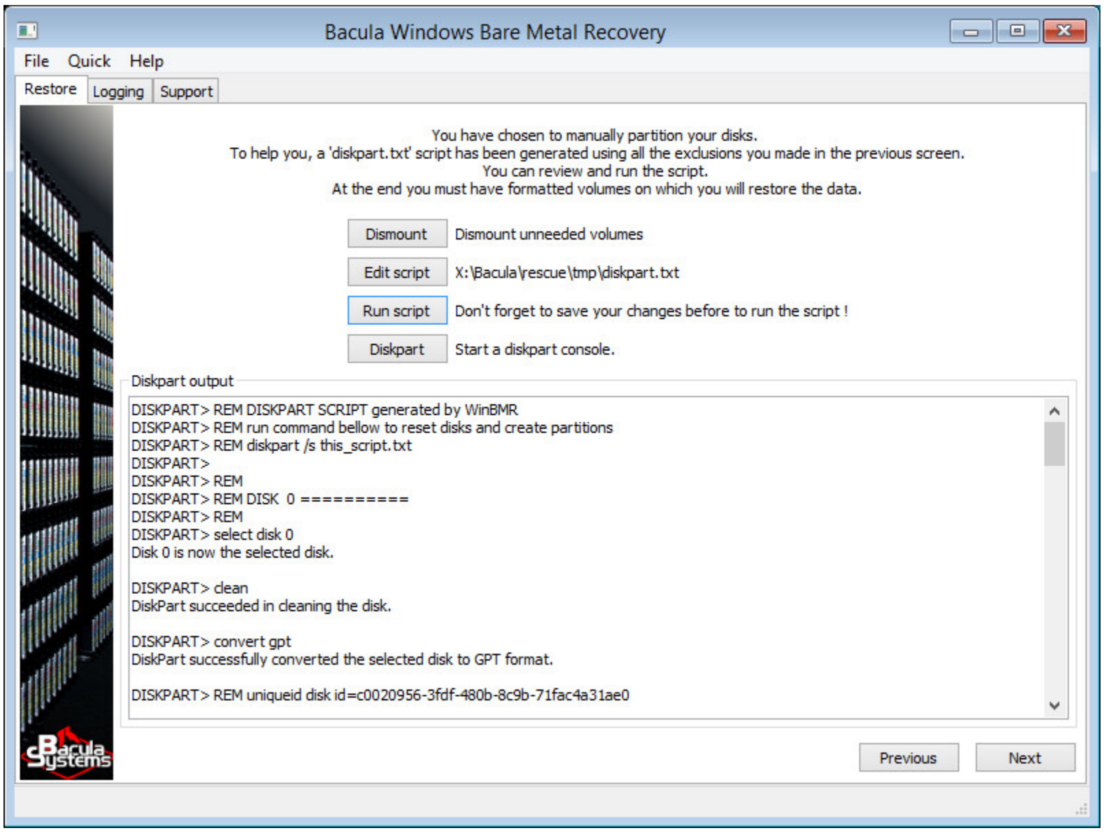
The list on the far right displays the “Volumes” that have been backed up. If you unselect a volume, it will not be created. The fact that the volume will not be created does not imply data can not be restored to another location! For example, if you have a C: and a D: drive on the same disk, you can skip the creation of the D: drive but still restore data from D: to the C: drive.

The “dyn” column on the left tells you if a disk is part of a dynamic “Volume”, and the last column on the right tells you which of these dynamic “Volumes” are on which disks. The column “size” on the right shows two values, the first one is the space used on the volume and the second one the size of the source volume. The space used is only an estimate and should always be less or equal to the size needed for the restore.

If the disk to which you are restoring your data is of a different size, the size of the last partition will be automatically adapted to fit the new disk size.

The "Set disk ID" checkboxes at the bottom allow you to decide if the re-partitioned disks should be assigned their original disk IDs. Windows may encounter problems if two disks with the same disk ID exist on the same machine. Our default setting provides the best user experience. You should change this setting only if you are aware of the resulting consequences. Please contact support with any questions.

If this interface is not flexible enough for your needs, you can select the "Manual partitioning" radio button. (Figure **ManualPartitioning**). If you want to create additional partitions, you must select the manual mode, as it is not possible to create new partitions with the current interface.



You have chosen to manually partition your disks.
To help you, a 'diskpart.txt' script has been generated using all the exclusions you made in the previous screen.
You can review and run the script.
At the end you must have formatted volumes on which you will restore the data.

Dismount unneeded volumes
X:\Bacula\rescue\tmp\diskpart.txt
Don't forget to save your changes before to run the script !
Start a diskpart console.

Diskpart output

```
DISKPART> REM DISKPART SCRIPT generated by WinBMR
DISKPART> REM run command bellow to reset disks and create partitions
DISKPART> REM diskpart /s this_script.txt
DISKPART>
DISKPART> REM
DISKPART> REM DISK  0 ==========
DISKPART> REM
DISKPART> select disk 0
Disk 0 is now the selected disk.

DISKPART> clean
DiskPart succeeded in cleaning the disk.

DISKPART> convert gpt
DiskPart successfully converted the selected disk to GPT format.

DISKPART> REM uniqueid disk id=c0020956-3fdf-480b-8c9b-71fac4a31ae0
```

Fig. 147: Manual Partitioning

At the end of the manual partitioning process, you **must** have created and formatted the "Volumes" to restore your data as well as all the system partitions. You **must have** assigned drive letters to be able to select them during the following steps.

To help you, this screen provides all the information that the auto-partitioning procedure would use. You are free to use your own tools, or even to have pre-partitioned the disk before booting the WindowsBMR media. Once the disk layout fits your needs, click "Next".

The steps you can do are the following: First you should understand that simply clicking "Dismount", then "Run Script" and then "Next", will do exactly what "auto-partitioning" mode does.

The "Dismount" button un-mounts all drives. Only the X: drive will stay after this, even the CDROM drive letter will have been dismounted. This will avoid drive letter collision later on, but please be aware that you must not use drive X:.

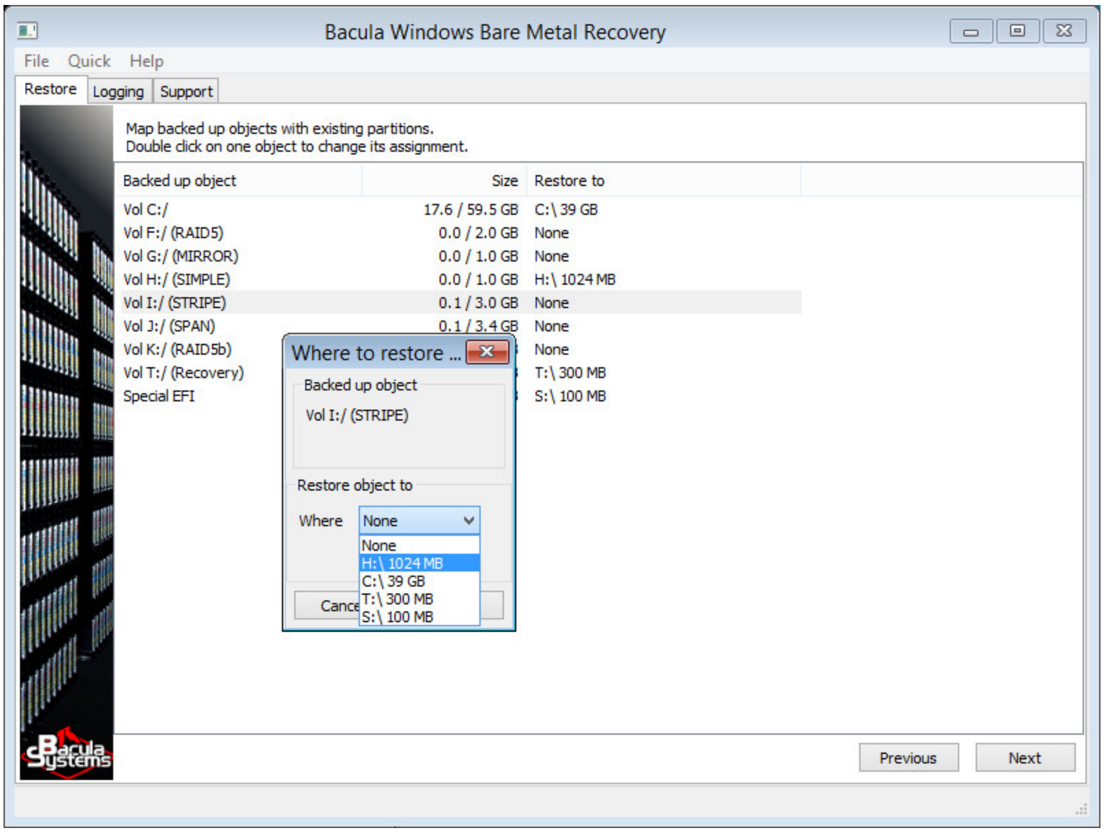
A script "diskpart.txt" has been generated for you. You can push the button "Edit script" to edit it. The script has been generated using the information from the previous screen. If you have excluded disks or partitions you will see the changes in the script. The script contains some comments to help you. The original size of the partition and the disk ID are sometimes commented out. You can uncomment the line or copy the size into the "create partition" command.

The original script cleans up existing partitions on each disk it will re-partition. Thus you can run the script repeatedly, without the need to manually clean up partitions in between.

When done you must save the file and click the "Run script" button to run it. You will see the script output in the text area below. Since the program does not detect errors at this point, please read this output carefully and search for any errors then modify your script until you corrected any and all errors.

Keep in mind that WindowsBMR uses drive letters. You must format and assign a drive letter to all the volumes that you want to use during the restore process.

After the automatic or the manual partitioning of your disks, the "Volume matching" screen is shown. (Figure **VolumeMatching**).



Map backed up objects with existing partitions.
Double click on one object to change its assignment.

Backed up object	Size	Restore to
Vol C:/	17.6 / 59.5 GB	C:\ 39 GB
Vol F:/ (RAID5)	0.0 / 2.0 GB	None
Vol G:/ (MIRROR)	0.0 / 1.0 GB	None
Vol H:/ (SIMPLE)	0.0 / 1.0 GB	H:\ 1024 MB
Vol I:/ (STRIPE)	0.1 / 3.0 GB	None
Vol J:/ (SPAN)	0.1 / 3.4 GB	None
Vol K:/ (RAID5b)		None
Vol T:/ (Recovery)		T:\ 300 MB
Special EFI		S:\ 100 MB

Backed up object: Vol I:/ (STRIPE)

Restore object to — Where: None

None
H:\ 1024 MB
C:\ 39 GB
T:\ 300 MB
S:\ 100 MB

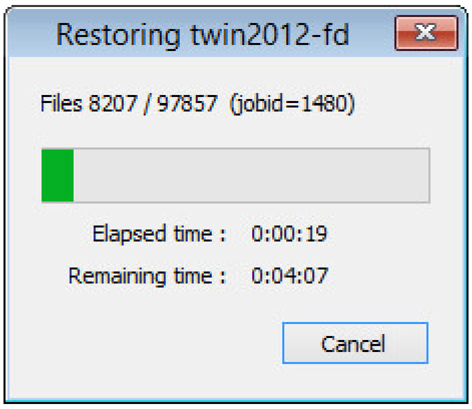
Fig. 148: Volume Matching

On this screen, you have to tell the program where you want to restore your data to. The objective is to match the drive letters of your source host to the drive letters of the target host. The program has automatically matched the "Volumes" using the same drive letters. Normally, C:/ (here we use the Bacula notation) will go to C:\ (with the Windows notation) and the same for other drive letters. If you have fewer disks or excluded some partitions, you will get orphaned "Volumes" that are going to nowhere (None). You can double-click on the right column to change the "Volume" assignment.

On the other hand, if you don't want to restore a "Volume", you can set it to None and possibly restore it later, after having rebooted the system.

Restore Progress

When ready, click "Next" to start the restore.



Restoring twin2012-fd

Files 8207 / 97857 (jobid=1480)

Elapsed time : 0:00:19

Remaining time : 0:04:07

Fig. 149: Restore Status

The (figure **RestoreStatus**) window shows you the progress of the restore.

You can "Cancel" the process and go back to make further changes, and start a new restore process.

When done, the (figure **Final**) screen shows the status of the restore and the status of the process making the host bootable. Error messages and success status are only indicative – you can know the final status only after having rebooted. Check any error or warning messages (if any) to see if they are critical for you and ask our support team if you are worried.

Click "Next" to reboot the system. If you have questions for our support team, download the log of the restore in advance, since our support may ask for it. See the "Troubleshooting" section to know how to get the bssupport.zip file.

When rebooting the machine don't forget to remove the rescue BMR disk / USB key to prevent booting into the rescue system again.

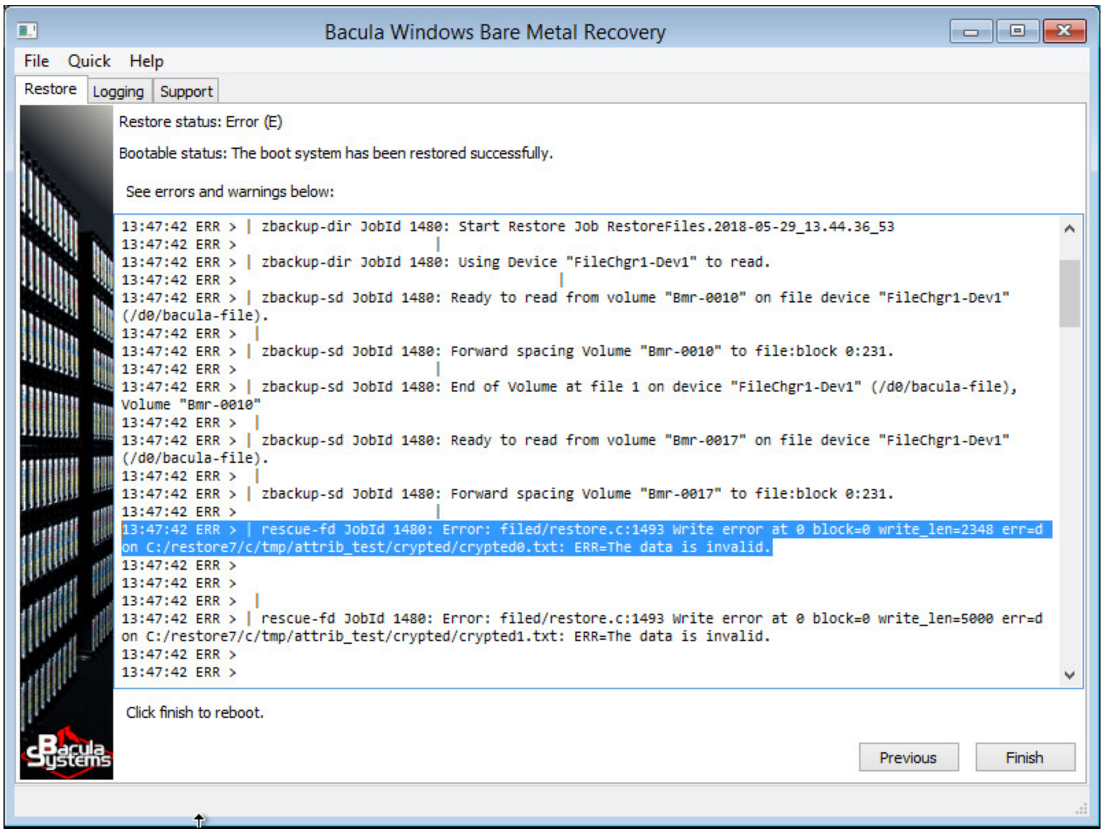
If don't want to reboot immediately, the command `wpeutil reboot` can be used to reboot later and the command `bsrescue` will start the bare-metal recovery program again.

Finalizing a Windows BMR Session

In some cases, after the actual BMR process is finished, some additional tasks may be needed.

For example, if the recovered system does not boot, repairing the system using the repair options of the original Windows installation media may be needed. Also, for Active Directory servers, it may be necessary to follow Microsoft's guidelines to get a consistent state of the AD databases and synchronize with other AD servers.

If your setup includes dynamic disks, you must import them in the freshly restored system after the reboot. You can do that from the disk manager or using "diskpart" by selecting one of the dynamic disk and using the "import" command:



Restore status: Error (E)

Bootable status: The boot system has been restored successfully.

See errors and warnings below:

```
13:47:42 ERR > | zbackup-dir JobId 1480: Start Restore Job RestoreFiles.2018-05-29_13.44.36_53
13:47:42 ERR >                          |
13:47:42 ERR > | zbackup-dir JobId 1480: Using Device "FileChgr1-Dev1" to read.
13:47:42 ERR >                                       |
13:47:42 ERR > | zbackup-sd JobId 1480: Ready to read from volume "Bmr-0010" on file device "FileChgr1-Dev1"
(/d0/bacula-file).
13:47:42 ERR >  |
13:47:42 ERR > | zbackup-sd JobId 1480: Forward spacing Volume "Bmr-0010" to file:block 0:231.
13:47:42 ERR >                          |
13:47:42 ERR > | zbackup-sd JobId 1480: End of Volume at file 1 on device "FileChgr1-Dev1" (/d0/bacula-file),
Volume "Bmr-0010"
13:47:42 ERR >  |
13:47:42 ERR > | zbackup-sd JobId 1480: Ready to read from volume "Bmr-0017" on file device "FileChgr1-Dev1"
(/d0/bacula-file).
13:47:42 ERR >  |
13:47:42 ERR > | zbackup-sd JobId 1480: Forward spacing Volume "Bmr-0017" to file:block 0:231.
13:47:42 ERR >                          |
13:47:42 ERR > | rescue-fd JobId 1480: Error: filed/restore.c:1493 Write error at 0 block=0 write_len=2348 err=d
on C:/restore7/c/tmp/attrib_test/crypted/crypted0.txt: ERR=The data is invalid.
13:47:42 ERR >
13:47:42 ERR >
13:47:42 ERR >  |
13:47:42 ERR > | rescue-fd JobId 1480: Error: filed/restore.c:1493 Write error at 0 block=0 write_len=5000 err=d
on C:/restore7/c/tmp/attrib_test/crypted/crypted1.txt: ERR=The data is invalid.
13:47:42 ERR >
13:47:42 ERR >
```

Click finish to reboot.

Fig. 150: Final

```
select disk <XX>
import
```

Remote or Head Less recovery using VNC

It is possible to do the BMR of a remote or a head less machine using the VNC protocol. You can choose between TightVNC and TigerVNC server. Only TigerVNC supports SSL connections. You must enable VNC and setup the password on the ISO image using our **winbmr-iso-configurator.exe** tools. See the *Configuring the WindowsBMR Rescue ISO* section to learn how.

The VNC server is started at startup. You must know the IP address of your machine to connect to it. The port used by VNC is the default one, 5900. Notice that the IP address is displayed by the console just before the WindowsBMR program starts, but if you don't have access to the console you'll need to know the IP address or be able to find it in another way.

More About the GUI

The GUI includes some "helpers" that can be useful in some circumstances.

The File Menu

The file menu allows you to start additional command prompts or the "Tools dialog box".

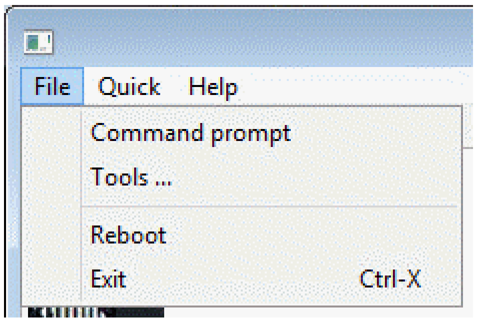


Fig. 151: The File Menu

The Quick Menu

The Quick menu can open text files that are used during the restore process. Some files may be unavailable depending on where you are in the restore process or the type of the system you are restoring :

Open Log
: open the log file

Open diskpart.txt
: open the diskpart script as used by the auto-partitioning.

Open diskpart_source.txt
: open the diskpart script as it would have been generated by the auto-partitioning process without any changes to the original system

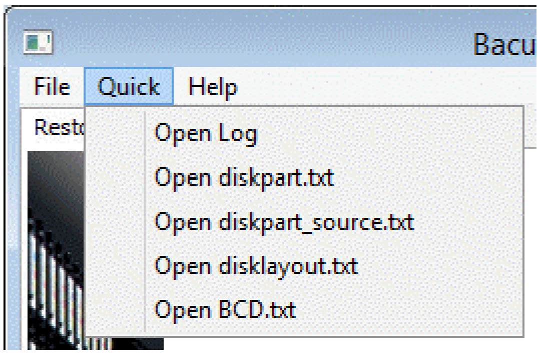


Fig. 152: The Quick Menu

Open disklayout.txt
open the file that the WindowsBMR plugin has generated at the backup time and that contains information about the disk and partition layout of the source host.

Open BCD.txt
open the text version of the BCD database on the source host.

The Tools Dialog Box

The tools dialog box allows you to load drivers, setup your network adapter or remove any existing partitions from your disks.

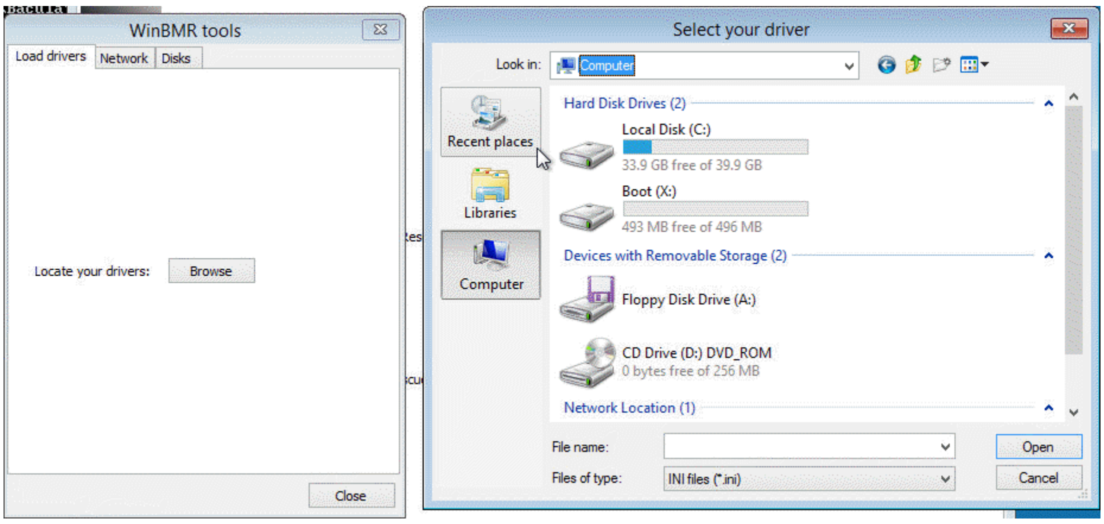
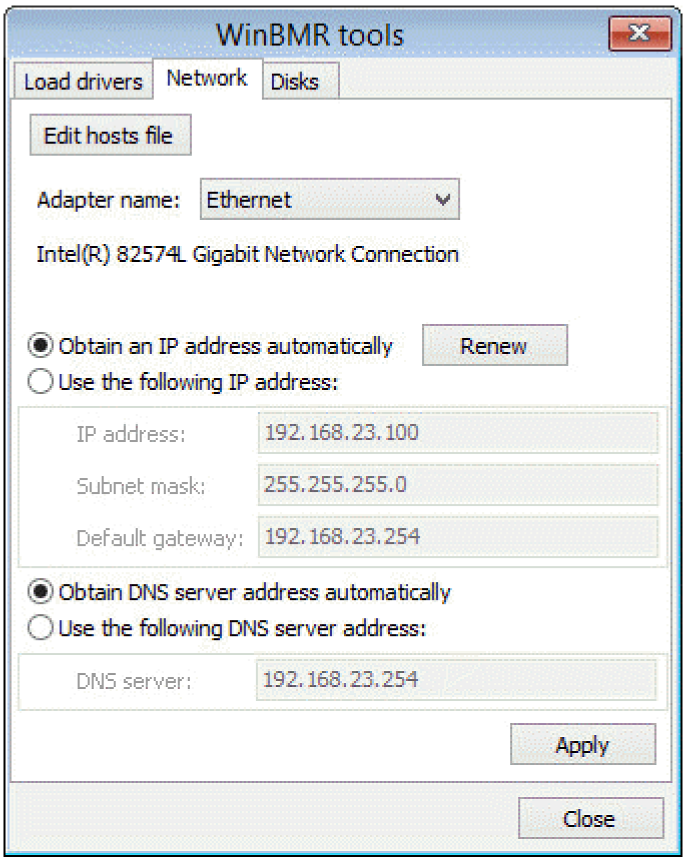


Fig. 153: Load Drivers



Adapter name: Ethernet

Intel(R) 82574L Gigabit Network Connection

(●) Obtain an IP address automatically

( ) Use the following IP address:

IP address: 192.168.23.100

Subnet mask: 255.255.255.0

Default gateway: 192.168.23.254

(●) Obtain DNS server address automatically

( ) Use the following DNS server address:

DNS server: 192.168.23.254

Fig. 154: Network Configuration

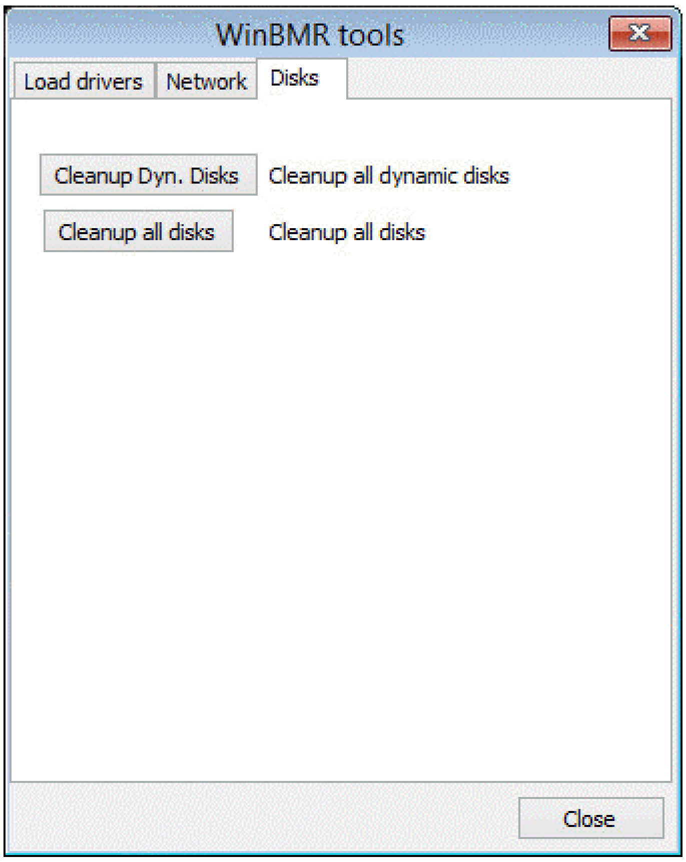
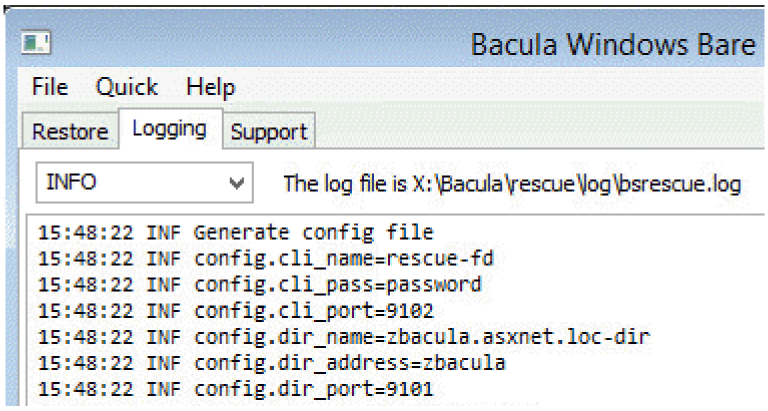


Fig. 155: Cleanup Disks

The Logging Tab

Next to the “Restore” tab is the Logging tab that displays all log messages of the program. You can filter messages by level.



```
15:48:22 INF Generate config file
15:48:22 INF config.cli_name=rescue-fd
15:48:22 INF config.cli_pass=password
15:48:22 INF config.cli_port=9102
15:48:22 INF config.dir_name=zbacula.asxnet.loc-dir
15:48:22 INF config.dir_address=zbacula
15:48:22 INF config.dir_port=9101
```

Fig. 156: Logging Tab

The Support Tab

The “Support” tab allows generation of a report file called “bssupport.zip”. The file can be uploaded directly to Bacula support through HTTPS protocol or downloaded through HTTP protocol via a built-in server. If you already have a ticket open, then supply the ticket number while sending the file to the support. If not, use any identifier that can help the support to easily identify your report.

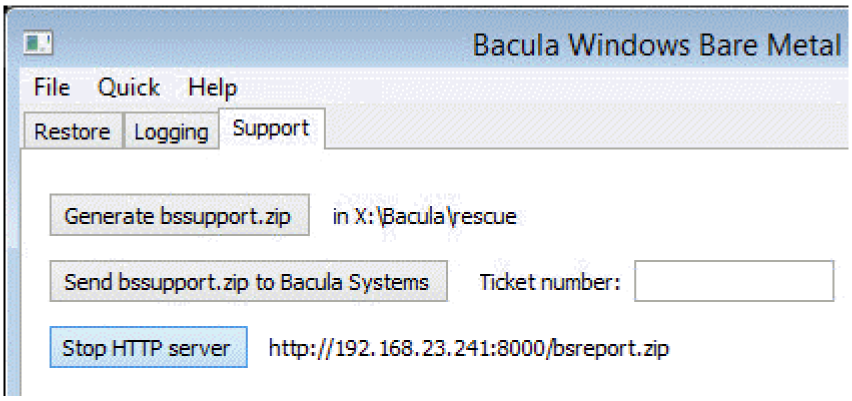


Fig. 157: Support Tab

Troubleshooting

Backup

Before starting any BMR test, you must be able to back up files on your source host with Bacula Enterprise. Then you must be sure that the WindowsBMR plugin is correctly installed and working. Use the command "status" in bconsole to check the status of your source client :

```
*status client=zwin2012b-fd
Connecting to Client zwin2012b-fd at zwin2012b:9102

win2012b-fd Version: 6.2.4 (04 May 2013) VSS Linux Cross-compile Win64

Daemon started 18-May-13 12:15. Jobs: run=2 running=0.
Microsoft (build 9200), 64-bit
Heap: heap=0 smbytes=132,812 max_bytes=394,019 bufs=110 max_bufs=228
Sizes: boffset_t=8 size_t=8 debug=0 trace=1 mode=0,2010 bwlimit=0kB/s
Plugin: alldrives-fd.dll winbmr-fd.dll
```

The "Plugin" line shows that the winbmr-fd.dll is installed.

You can get more debugging information from the "trace" file in "C:Program FilesBaculaworking", if you enable debug tracing from the console with:

```
*setdebug level=20 trace=1 client=zwin2012b-fd
Connecting to Client zwin2012b-fd at zwin2012b:9102
2000 OK setdebug=20 trace=1 hangup=0
```

Restore

All the critical information for restoration is compiled in the bssupport.zip file. This file can be generated, downloaded or directly sent to Bacula Systems support from the "support tab". Read the section about the "Support tab" for more information.

File transfer to and from the WindowsBMR environment

If you need to transfer files to or from the WindowsBMR environment, you can use:

- A USB stick
- FTP
- SCP (ssh copy)
- mount a network share using command "net use"

USB Stick

You can use the command below to display all connected drives and identify your USB key:

```
echo list volume | diskpart
```

Use the command "xcopy" to copy a directory recursively :

```
X:\Bacula\rescue\restore>xcopy temp h:\ /e
```

FTP

Here is a sample of using FTP to upload and download a file.

```
X:\Bacula\rescue\restore>ftp ftp.yourserver.net
Connected to ftp.yourserver.net.
220-Bienvenue,
User (ftp.yourserver.net:(none)): bacula
331 User bacula OK. Password required
Password: *********
230-User bacula has group access to:  users
230 OK. Current restricted directory is /
ftp> put temp\logs\stderr.log
200 PORT command successful
150 Connecting to port 49204
226-File successfully transferred
226 0.420 seconds (measured here), 186.14 Kbytes per second
ftp: 81855 bytes sent in 0.31Seconds 262.36Kbytes/sec.
ftp> get stderr.log
200 PORT command successful
150-Connecting to port 49205
150 78.1 kbytes to download
226-File successfully transferred
226 0.047 seconds (measured here), 1.67 Mbytes per second
ftp: 81852 bytes received in 0.08Seconds 1049.38Kbytes/sec.
ftp> quit
221-Goodbye. You uploaded 79 and downloaded 80 kbytes.
221 Logout.
```

SCP

More secure (and probably more common than ftp, today) is scp, a command that uses the SSH protocol. Since most Linux systems run the SSH daemon, it is easy to transfer a file using SCP.

```
X:\Bacula\rescue\restore>pscp temp\logs\stderr.log root@max:/tmp
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 46:75:4c:4b:41:b9:57:c2:d9:58:8b:1d:62:1a:54:18
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n) y
root@max's password:******
```

To download a file just swap the arguments:

```
X:\Bacula\rescue\restore> pscp root@max:/tmp/stderr.log .
```

Network Share

To mount a network share use the command "net use". This command is very capricious, if you have a problem, try to qualify / unqualify / tune the various options it supports.

```
X:\Bacula\rescue\restore> net use U: \\192.168.1.10\C$ /user:Administrator␣
↪password
```

- C$ is the administration share of drive C:, all drives have an administration share.
- You can also use a share Name, if the windows host exports any.
- "Administrator" is the local Administrator, use "DOMAINNAMEAdministrator" for the domain admin. You can also use any other local or domain user with the correct password: "HOST-NAMEusername". The administrator account name is localized, so it may be different for different localizations of Windows.
- Please take care to use the correct combination of username and password to get write access to the network share.
- Instead of the IP address you can also try to use the DNS or netbios name.

```
net use V: \\myserver.domain.com\myshare /user:LOCALCPU\LOCALUSER password
net use W: \\myserver\myshare /user:DOMAINNAME\DOMAINUSER password
```

Use the command `xcopy` to copy a directory recursively, for example:

```
X:\Bacula\rescue\restore>xcopy temp h:\ /e
```

Creating a bootable USB flash disk

To create a bootable USB device from the ISO, you must first prepare your USB key using diskpart. You must create a single partition and activate it to make it bootable, then format the partition as FAT32 and assign it a drive letter, to copy the content of the CDROM on it. Here is the output of the diskpart utility. Be careful, be sure to select the right disk; when you run the "clean" command, all data on the partition will be erased!

```
PS C:\Users\Administrator> diskpart

Microsoft DiskPart version 6.2.9200

Copyright (C) 1999-2012 Microsoft Corporation.
On computer: ZWIN2012

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           60 GB      0 B        *
  Disk 1    Online          100 GB    99 GB        *
  Disk 2    Online         2048 MB  2015 MB        *
  Disk 3    Online         7554 MB      0 B

DISKPART> select disk 3

```

(continues on next page)

(continued from previous page)

```
Disk 3 is now the selected disk.

DISKPART> clean

DiskPart succeeded in cleaning the disk.

DISKPART> create partition primary

DiskPart succeeded in creating the specified partition.

DISKPART> active

DiskPart marked the current partition as active.

DISKPART> format quick fs=fat32

  100 percent completed

DiskPart successfully formatted the volume.

DISKPART> assign

DiskPart successfully assigned the drive letter or mount point.

DISKPART> exit

Leaving DiskPart...
```

You need to copy all files from the ISO image / CD-ROM to the freshly formatted USB key, which can then be used to boot a system.

Securing the Rescue Console

Overview

The recommended ACLs setup for the rescue console is to use `*All*` for all ACLs. This allows you to restore any backup of any host using the same console. The drawback is that anybody getting the password of this console can restore any data to their own computer.

If you must delegate BMR privilege to people with limited permissions, you must create restricted consoles using Bacula ACL directives. For any of these consoles, you must create clients with the same name because WindowsBMR uses the `setip` command that requires that the client and the console use the same name.

Minimal ACLs

The minimal setup requires accesses to all resources needed to restore the backup. For example:

```
Console {
  Name = <rescue-fd>
  Password = <password>
  CatalogAcl = <catalog>
  ClientAcl = <rescue-fd>, <host-fd>
  JobAcl = <restore-job>, <winbmr-job>
  FilesetAcl = <restore-fileset>, <winbmr-fileset>
  PoolAcl = <pool>
  StorageAcl = <storage>
  WhereAcl = *all*
  # the following three ACLs are required when using Bacula Enterprise
  # version 8.1 or newer
  RestoreClientACL = <rescue-fd>
  UserIdACL = *all*
  DirectoryACL = *all*

  CommandAcl = .bvfs_cleanup, .bvfs_get_jobids, .bvfs_lsdirs, .bvfs_lsfiles, .
↪bvfs_restore
  CommandAcl = .bvfs_update, cancel, .catalog, .clients, gui, .jobs, list,␣
↪llist, q, quit
  CommandAcl = restore, setip, show, status, wait
  # .sql is not required anymore since WinBMR >= 3.4.0 and Bacula Enterprise >
↪ 8.4.2
  CommandAcl = .sql
}
```

Replace <`tag`> as follow :

rescue-fd
the name of the console and the file daemon.

host-fd
the host you want to restore.

restore-job
you must use a restore job that doesn't run any script to do the recovery.

winbmr-job
this is the WindowsBMR-enabled job used to backup your host. If you have multiple jobs, add them to the list.

restore-fileset
is the Fileset used by the <restore-job>.

winbmr-fileset
This is the Fileset used by the <winbmr-job>.

pool
the list of pools which hold data you want to restore.

storage
the storage devices which store data you want to restore.

In practice, there may be multiple jobs, stored on different pools located on different storages. You must add ACLs for all these jobs depending on your configuration.

If you want to restore multiple hosts from the same WindowsBMR console, you must merge all the ACLs together.

Troubleshooting ACLs

Before modifying your ACLs you should ensure that the WindowsBMR procedure works for your setup using ***All*** for all your ACLs. When you are sure, then you can try to fine tune your ACLs.

Each step in the BMR process requires different ACLs. When you get your clients and the **<rescue-fd>** listed in the GUI, you know that your `CatalogACL` and `ClientACL` are correct. When the list of dates is the expected one, you know that `FilesetACL` and `JobACL` are correct, too. If WindowsBMR doesn't show you the appropriate drive list, this is probably because the `PoolACL` or the `StorageACL` are misconfigured.

For finer troubleshooting, it is easier to test some commands in `bconsole` and look at the result. Before starting `bconsole`, ensure that you are connected to the director, which is the case if you can get the list of clients in the GUI. Use a command prompt, and type the command below to start bconsole.

```
X:\Bacula\rescue\executable\bconsole -c X:\Bacula\rescue\executable\bconsole.
↪conf
```

You must use the full path for the configuration file, a relative path will not work.

Every time you change an ACL in the configuration file, you must reload the configuration from another console using the `reload` command and restart the `bconsole` on the rescue client. Don't forget to re-run the command `setip` before trying a restore because every `reload` resets the IP address of the client.

First, test if you have access to the clients using the command "`.clients`". You must see your `<rescue-fd>` and all hosts that you want to be able to restore from this console.

Use "`.jobs type=R`" to list all restore jobs and verify that your `<restore-job>` is in the list.

Use "`list job=<winbmr-job>`" to see if you have access to the WindowsBMR job.

Use command "`setip`" without any argument to tell the director the current IP address of your console. The director will use this address as the one of the corresponding file daemon running on the same machine. If the command fails, check the `CommandACL` and make sure to use the same name for your console and the file daemon.

To test the `PoolACL` and the `StorageACL`, you have to try to restore at least one file. Before that you must be sure that your file daemon is running. Use "`status client=<rescue-fd>`" to get its status. The file that we will try to restore is `the_bacula_rescue_for_windows`, which is generated for every WindowsBMR-enabled backup. To list all BMR-enabled backups run "`restore client=<host-fd>`", in the menu select `2` to get a list of all jobs where a given file is saved, enter the name of the file and then select `13` to cancel the operation.

For example if my host is `zwin2003-fd`:

```
*restore client=zwin2003-fd

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.
```

(continues on next page)

(continued from previous page)

```
To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Cancel
Select item:  (1-13): 2
Enter Filename (no path):the_bacula_rescue_for_windows
+-------+-------------------+----------+------+------+-------+------------+
| JobId | Name              | Start    | Type | Stat | Files | Bytes      |
+-------+-------------------+----------+------+------+-------+------------+
| 7     | C:/...for_windows | 11:39:17 | B    | T    | 25250 | 5382372536 |
| 5     | C:/...for_windows | 23:05:07 | B    | T    | 21    | 372705     |
| 3     | C:/...for_windows | 21:14:30 | B    | T    | 25238 | 4853751303 |
+-------+-------------------+----------+------+------+-------+------------+
```

Here, the name has been truncated for display purposes. The full name is:

```
C:/Bacula/winbmr/data/the_bacula_rescue_for_windows
```

Take note of the JobId and the full path of the filename, taking care of the case of each characters. Then use the following command to start a restore :

```
restore client=<host-fd> restoreclient=<rescue-fd> jobid=<jobid> where=/
↪Bacula/temp file=<full_path>
```

In a test environment, the full command is:

```
restore client=zwin2003-fd restoreclient=rescue-fd jobid=7 where=/Bacula/temp␣
↪file=C:/Bacula/winbmr/data/the_bacula_rescue_for_windows
```

This last command should confirm that your Storage and Pool ACLs are correct.

Using Additional Hardware Drivers

Overview

If your hardware is not supported out of the box by our WindowsBMR system, you have the choice to load the required drivers manually or even embed them into the Rescue Image.

Download the missing drivers from the manufacturer or vendor web site. Choose a 64-bit version, and always prefer the Windows 8 or Windows 2012 version of the drivers. Older versions of the drivers could work too, if you cannot find more recent version.

Load Drivers from a USB Stick or a Floppy

Extract the drivers and associated files to a USB stick. Usually there are 3 files: `*.inf`, `*.sys` and `*.cat` like these:

```
26/08/2009  15:10           68,668 b57win32.cat
26/08/2009  15:10          181,388 b57win32.inf
26/08/2009  15:10          213,544 b57xp32.sys
```

At any time in the GUI you can open the menu item “Tools” in the “File” menu. A dialog box appears, where, in the tab “Load drivers”, you can click on the “Browse” button to locate the `.ini` file and then choose it and click “Open” to load the driver.

It is also possible to load the driver from the command line using the “drvload” command:

```
drvload filename.inf
```

To test if your network adapter or disk controller is working, use the commands `ipconfig` or `diskpart` as explained below.

Embedding Drivers into the Recovery Image

It is not possible to modify an ISO 9660 filesystem, but some free tools on the Internet can extract all data and rebuild a new ISO image including your changes. Ask Google for `"modify"` or `"edit"` `"iso"` `"images"` to find these tools. We have tried the free WinISO-5.3 with success.

You must copy your drivers into the `WinBMRDrv` directory at the root of the ISO image. This directory already exists and contains additional drivers. When looking for drivers, WindowsBMR does a recursive search in this directory, thus you can keep your drivers in separate directories. Remove the existing drivers if you think they conflict with your own drivers.

Test your Drivers

You can use the `ipconfig` and `diskpart` commands to see if your network adapter or your disk controller has been detected by your drivers.

Run `"ipconfig"` from any command prompt to display the list of all network adapters found on the host and their IP addresses.

`diskpart` can display the list of installed disk. Run the program from a command prompt and at the diskpart prompt type `"list disk"`.

Compatibility

Bacula Enterprise

WindowsBMR 3.4 is compatible with Bacula Enterprise 6.2.6 and later.

Hardware

The ISO image is 64-bit and must be run on a 64-bit Intel x86 compatible processor. You can use it to restore any 32-bit Windows OS, but the CPU must be 64-bits. If you are restoring on a virtual machine, you may be required to temporarily switch the CPU or the Virtual Machine OS to 64-bits. After the restore you may switch back into 32-bit mode. Your restore machine must have 1GB of RAM, but 2GB are recommended.

Windows

WindowsBMR 3.4 can do bare metal recovery of all 32 and 64-bit XP versions of Windows and later.

It has been successfully tested on:

- Windows XP 32-bit (English, French)
- Windows 7 64-bit (French)
- Windows Server 2003 32-bit (English)
- Windows Server 2008 32-bit (English, French)
- Windows Server 2008R2 (English, French)
- Windows Server 2008R2 (English) on (U)EFI-enabled system
- Windows Server 2012 (English) on legacy BIOS
- Windows Server 2012 (English) on (U)EFI-enabled system
- Windows Server 2016 (English) on legacy BIOS
- Windows 10 & 11, Home and Workstation, on (U)EFI-enabled system

Restrictions

The Bacula Enterprise VSS Plugin (vss-fd.dll)

The Bacula Enterprise VSS plugin (vss-fd.dll) cannot be used with BMR. In fact, when doing backups to be used for BMR, you must not use the vss plugin otherwise the BMR restore will fail. This is because our BMR uses Windows PE, which does not include VSS capability, so when Bacula attempts to restore data backed up by the vss-fd.dll plugin, it will fail.

Don't confuse the plugin with the VSS option in the Fileset! VSS in the Fileset is enabled by default and **must be enabled** for a WindowsBMR backup.

If you need to restore a backup made with the vss-fd.dll plugin, please do so after you have restored your system with the BMR.

Restrictions in Using the BMR

- As mentioned above, you cannot do a WindowsBMR restore with a Bacula backup that used the vss-fd.dll plugin. It will fail in Microsoft WinPE environment.
- The drive letter X: is used internally by the rescue system and cannot be used during the restore procedure. If you have an X: drive, you must choose "Manual Partitioning" and use another free drive letter to restore it, then change the letter assignment after the first reboot.
- If you restore your server on **different hardware** such as IDE, SATA, SAS, SCSI, you will have to manage driver problems. Microsoft Windows will probably ask you to re-enter the license key for Windows. This is also true when moving between different virtual infrastructure.
- The rescue system needs to connect to your Director using the Bacula bconsole protocol (9101/tcp by default). After connecting, the Director Daemon uses the address from which bconsole was executed to contact the Client being restored. If the Client is behind a firewall or uses network address translation (NAT) it is possible that the Director will not be able to connect to the rescue Client. To avoid this kind of problem, we recommend that you ensure that the machine to be rescued resides in the same network as the Director and the Storage Daemon.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

Known Issues

Disk Labels

Disk labels can contain non-ASCII characters. These characters are sometime replaced by "0" (zero) at restore time. This should have no consequences on the restored host.

Volume Mount Points

All Volume mount points to be restored must have paths that are ASCII characters. This restriction will be removed in a future WindowsBMR version.

The machine fails to boot following the :term:`BMR` restore

Users have reported that running the command below fixes the issue.

```
bcdboot c:\windows
```

You can reboot from the ISO image and run the command using the command prompt that appears after you have selected the keyboard layout.

A common reason for boot failure is the exclusion of files that are critical for the machine's startup by the user. Review your *Fileset* and verify if you have inadvertently excluded files due to their file extensions or the directories containing system files.

Troubleshooting

- QUESTION: I don't seem to have any connectivity with the WindowsBMR, what can I do?

  ANSWER: This is most probably because the drivers of your Ethernet NIC

  are not present on the WindowsBMR iso.

  The section "Use extra hardware drivers" explains how to solve this problem.

- QUESTION: It looks like the Bacula Director cannot connect to my client, I have checked that passwords and port match, but I still cannot connect, why?

  ANSWER: This happens most often if the machine that you are trying to restore is on a different subnet than your director; place the machine into the subnet where your Director is connected.

- QUESTION: I restored onto different hardware, and it doesn't reboot, why?

  ANSWER: This is most probably because your original Windows didn't include the drivers for your new hardware or Windows has not been set up to use these drivers as boot device. Restore onto the same hardware or install the required drivers in the original machine, add these drivers to the list of drivers on which the system can boot on and back up again.

- QUESTION: I restored onto the same hardware, but a different disk, and it doesn't reboot, why?

  ANSWER: This is because you still have the original partition in your BCD file; you can solve this by removing your original drive from the machine. It may also help to disable the original drive in the BIOS configuration.

  Please note that if you already rebooted and tried to repair the bootloader things can be broken – it is much better to have the original disk removed before the first attempt to boot the recovered system.

- QUESTION: The diskpart never ends the disk inventory. What can I do?

  ANSWER: Please check if some of the disks are made available through an HBA interface and disconnect it. The HBA disk may be restored after the basic server is restored.

Bare Metal Recovery for Linux

Executive Summary

This **Bacula Systems** User Guide explains how Bare Metal Recovery of Linux systems can be done with the LinuxBMR toolkit **Bacula Systems** provides.

The reader of this paper is expected to have a good understanding of Bacula in general, i. e. working with the configuration files and `bconsole` should be expected. Also, general system administration knowledge as required for the backed up environment is assumed. The terminology used with **Bacula** needs also to be known.

To make sure you understand the current state of this **Bacula Systems** LinuxBMR toolkit, please read the Release Notes.

As any component that is to become part of a Disaster Recovery solution, the LinuxBMR toolkit needs to be extensively tested on any hardware it will be used on. Test results, site- or system-specific procedures, and essential configuration should be documented in a Disaster Recovery manual.

Introduction to BMR

Bare Metal Recovery or BMR for short, is the term usually used to describe restoration of a complete operating system to new hardware without actually going through the operating system's installation procedure.

The main goal is to get from a new, empty machine (bare metal) to a fully functional operating system including all applications and data as quickly as possible. The real challenge is to set up the disk subsystem of the new machine in a way that closely resembles the original disk layout where data was backed up from, and to ensure the recovered operating system can be booted.

If the new hardware is of a different type than the machine the installed software was backed up from originally, BMR will very often not be possible. To be useful, BMR procedures need to take that into account. It is necessary to ensure that the BMR hardware being restored to is compatible with the source system.

In the example in table *BMR source and target examples*, some common situations for Linux users today are outlined. In particular it is important to understand that software written for 32-bit x86-CPUs can be run on 64-bit x86-CPUs, but not vice versa, and that software written to run on one CPU family, like Sparc or x86, will not run on any other CPU family.

Table 54: BMR source and target examples

Origin	New Hardware	Problems	BMR possible?
i386	i386	None	Yes
i386_64	i386_64	None	Yes
i386	i386_64	Existing software not optimized for hardware	Yes
i386_64	i386	64-bit software will not run on 32-bit hardware	No
i386	SPARC	Completely different hardware	No

Overview of Bacula Enterprise LinuxBMR

The **Bacula Enterprise** LinuxBMR toolkit consists of several parts, each designed to fulfill one of the tasks in a BMR-enabled backup and recovery environment. In addition, some configuration of **Bacula** is required to ensure the software is actually used. Accordingly, well-defined procedures of how to install, deploy and use the BMR backup tools as well as how to do a BMR exist.

We will describe the software, the required configuration, and how to use it together in more detail after giving an overview first.

The overall approach to allow BMR is to collect essential system information during the backup itself, and to use that information during a BMR session to set up the new hardware to allow immediate reuse of the restored system, applications and data after the BMR is finished.

For the actual backup, after configuring the backup jobs accordingly, no additional effort is required.

During recovery, the essential disk partitioning data is pulled from the existing backups, applied to the BMR machine so that the target system is set up as needed, and then the actual restore is initiated. These steps, plus selection of what client to restore to, and what point in time to recover from is all handled by an easy to use GUI interface which is part of **Bacula Systems**' LinuxBMR recovery image.

The LinuxBMR recovery image may be downloaded from your download area. This ISO image may be written to a CD, a DVD, or a USB key. It is also possible to customize the image with some information related to your infrastructure. The image size varies with different releases of the product, so it may exceed the usable capacity of CDs.

The important fact is that, unlike other BMR approaches, no source system specific information is stored on the recovery media itself, which means that only one site-specific disaster recovery media is required.

Setting up LinuxBMR

To set up Linux BMR for **Bacula Enterprise**, several steps are required.

This chapter guides through all the steps necessary to set up and configure the needed LinuxBMR infrastructure.

Configuration Overview

These can be grouped into the following categories:

- Installing Bacula Enterprise Client
- Enabling BMR in your backup jobs
- Preparing BMR restore
- Creating BMR rescue media.

Installing Bacula Enterprise Client

In order to protect a server with the LinuxBMR toolkit, the Bacula Enterprise Client needs to be installed on this server. The client package is available in your download area. You will need at least the bacula-enterprise-client and the bacula-enterprise-libs (or bacula-enterprise-common) packages.

The `Bacula-rescue.sh` program is executed on the Client to collect critical system information. This program is included in the client package beginning with the 8.8.0 release of Bacula Enterprise. If an older version is in use, the bacula-enterprise-client-bmr package needs to be installed on the client(s). This package is available in your download area.

Enabling BMR in Jobs

For the jobs that should be BMR-enabled, two requirements have to be met. One is that critical system information has to be generated and backed up, and the other is that all file systems that are required must be included in the job.

The first requirement – ensuring that the system information required for BMR is available – is actually quite simple: The script that generates this information needs to be executed prior to the actual backup. This is done with a **Run Script** for the BMR-enabled jobs. To make things simpler, we recommend to use a **JobDefs** resource for those jobs. An example JobDefs resource and the configuration for an actual LinuxBMR Job are shown below:

```
JobDefs {
  Name = "LinuxBMR-JD"
  Client = lsb-245-fd

  # This Fileset contains the entire Linux System
  File Set = "AllUnix-U1004like"

  Type = Backup
  Level = Incremental
  Storage = File
  Messages = Standard
  Pool = Tier1
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%c-%n.bsr"

  # Enable LinuxBMR
  ClientRunBeforeJob = "/opt/bacula/bin/Bacula-rescue.sh"
}
```

Note: Note the **ClientRunBeforeJob** directive which ensures that the `Bacula-rescue.sh` script is executed to collect the critical system information. Alternatively, a **RunScript** block can be used.

```
Job {
  Name = lsb-246-bmr
  JobDefs = LinuxBMR-JD
  Client = lsb-246-fd
}
```

The second requirement is to ensure that all data is actually backed up. Typically, this means **everything** except your specific application data such as SQL database files. This requires a **File Set** which includes all local file systems.

In general, this is not difficult to achieve. However, depending on your site's requirements, it may be helpful to create the **File Set** includes automatically, on the client side, when the job is run. An example of how to automatically include all local file systems with selected filesystem types is shown below (if you have other Linux filesystems in use, you will have to extend the list below):

```
Fileset {
 Name = AllUnix
 Include {
   Options {
      signature=MD5
      compression = LZO
      xattrsupport = yes
      aclsupport = yes
      onefs = no
      fstype = ext2, ext3, ext4, xfs, msdos
   }
   File = /
  }
  Exclude {
    File = /tmp                     # exclude temp files
    File = /var/tmp
    File = /opt/bacula/working/*    # exclude Bacula working files

    File = /var/lib/postgres/data   # ensure that your databases
                                    # are saved by an other way.
    File = /var/log/lastlog         # this file if not backed up as sparse can␣
↪take
                                    # 100's of GB and is recreated at first␣
↪boot

  }
}
```

Since Bacula Enterprise 18.2.0, `ext3`, `ext4`, `fat`, `vfat` and `fat32` are valid values for the `fstype` option in the Fileset. In Bacula Enterprise 18.2.0, these values are silently translated into `ext2` (for `ext3` and `ext4`) and `msdos` (for `fat`, `vfat` and `fat32`).

The *UEFI* partition is using a *VFAT* filesystem and needs the `msdos` type to be part of the list.

Preparing BMR Restore

To make the BMR configuration complete, a Rescue Client resource that will be used during restore needs to be defined:

```
Client {
  Name = rescue-fd                  # Use your rescue client name
  Address = 0.0.0.0                 # Will be set automatically by LinuxBMR
  Password = bacularescue           # USE YOUR OWN PASSWORD
  Catalog = MyCatalog
}
```

During a LinuxBMR restore session, the LinuxBMR system needs to contact the **Director**. To do so a Console is required in the Director's configuration. For security reasons a named Console is used, which may be configured with restricted permissions. A Console resource for use with the LinuxBMR toolkit could be set up as follows:

```
Console {
  Name = rescue-fd                # MUST HAVE same name as Client resource
  Password = bacularescue         # USE YOUR OWN!
  CommandACL = *all*
  ClientACL = *all*
  CatalogACL = *all*
  JobACL = *all*
  StorageACL = *all*
  ScheduleACL = *all*
  PoolACL = *all*
  FilesetACL = *all*
  WhereACL = *all*
  # The next two ACLs are required when using
  # Bacula Enterprise 8.8.0 and above
  UserIdACL = *all*
  DirectoryACL = *all*
  # This last ACL is available when using
  # Bacula Enterprise 8.8.0 and above but is not required
  RestoreClientACL = *all*
}
```

Finally a restore job resource which does not have any run scripts is needed by the LinuxBMR restore process.

Creating LinuxBMR Rescue Media

The LinuxBMR recovery image may be downloaded from your download area. This ISO image may be written to a CD, a DVD, or a USB key. It is also possible to customize the image with some information related to your infrastructure.

- LinuxBMR-rescue-amd64-2.0.0.iso

The size of the image can vary from version to version. The latest version is currently too big to fit on a CD media, but that may change for future releases, as we do our best to keep the size as small as possible. The ISO image file may be used directly on VMware or VirtualBox to boot a Virtual Machine.

On , tools like `Win32DiskImager` are available to write the image to a USB stick. This tool may be downloaded from https://sourceforge.net/projects/win32diskimager/

Warning: Attention!

On Linux, the `cp` or `dd` commands can be used to do the job. This needs to be done carefully, as the procedures described in this section will destroy anything already on the target! Make very sure that you use the correct device name for your USB stick. If you use the wrong device the result could be that all information on your hard disk is lost.

With current systems, a USB stick should be automatically recognized when inserted. If it is not, you should check that the usb-storage kernel module is loaded. When the USB stick is inserted, it will be mapped to a block storage device named /dev/sdX, where the "X" usually is a letter in the range a-z. You should be able to see to which device the USB stick was mapped by running the command dmesg after inserting it.

To write to your stick, you may have to turn off its write protection switch.

```
root# cp LinuxBMR-rescue-amd64-2.0.0.iso /dev/sdX
root# sync
```

Make sure to use /dev/sdX and not /dev/sdX1. This is a very common error.

Doing BMR

Bare-Metal Recovery itself is done by booting the target system from the BMR image which should have already been created as described in section *Creating LinuxBMR Rescue Media*. Depending on the target computer's setup, booting from the BMR image may require some BIOS settings changes or interaction during the boot process – check with the system's manual if unsure!

Starting the Recovery System

Once the BMR system boots, you are first presented with the Isolinux language selection screen which will look similar to the one in the screenshot below. Use the up or down arrow keys on the keyboard to get to your language and then press enter.

Note: You are selecting the language of the Linux Desktop environment, but the BMR tool itself is available only in English.

Then press F3 to select your keyboard mapping:

Finally select "Start Bacula LinuxBMR" and press Enter.

After a short time, the Recovery system will be booted into a graphical desktop as shown below:

Before starting the BMR process itself, it may be useful to verify that the automatically determined network settings fit your environment. This is especially important if there is more than one network card (i. e., a production network and a dedicated backup network), or VLANs are used and no automatic address assignment is available. In these cases, manual configuration of the network settings may be required.

To do that, click the Network Manager tray icon in the bottom right of the screen (left of the clock) as shown above.

Starting the BMR Process

Once the network is configured correctly, you can start the Bacula Rescue session by double-clicking its icon on the desktop.

The first screen presented is a welcome message:

The menu and the 3 tabs in the windows top left corner are explained in section *More about the GUI*. Click Next to continue.

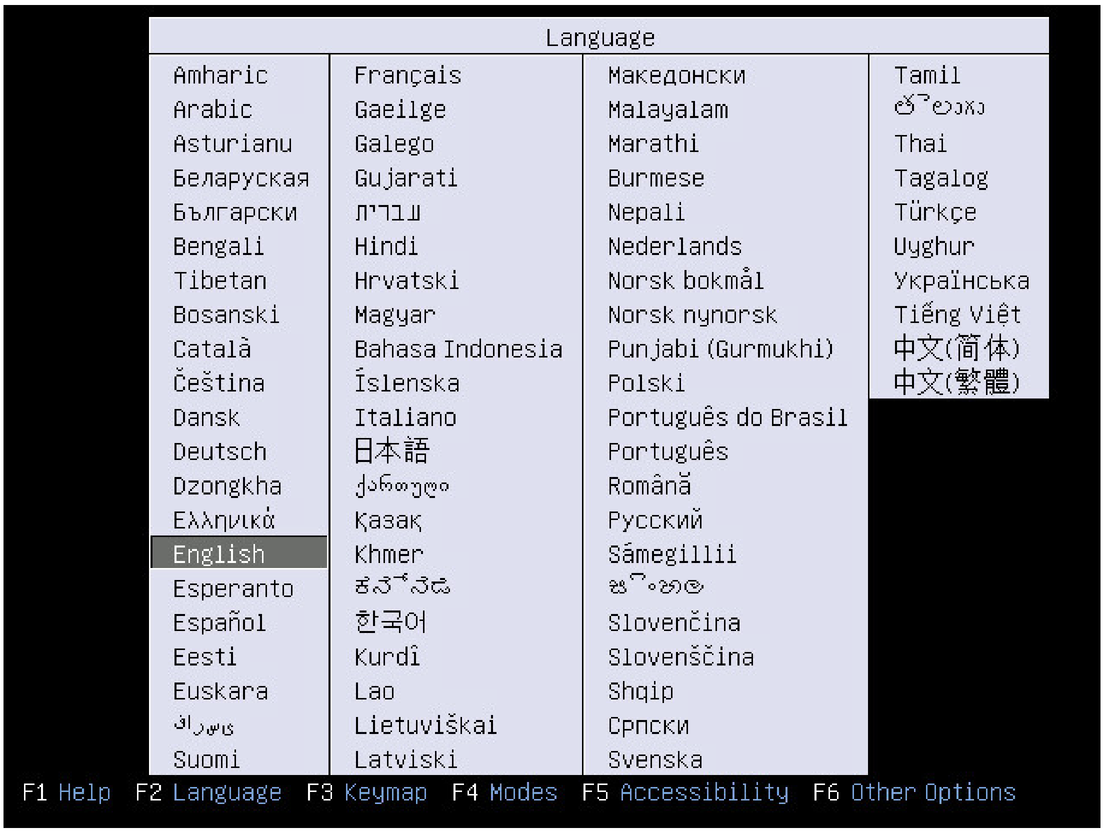


Fig. 158: The BMR Boot Screen – Language Selection

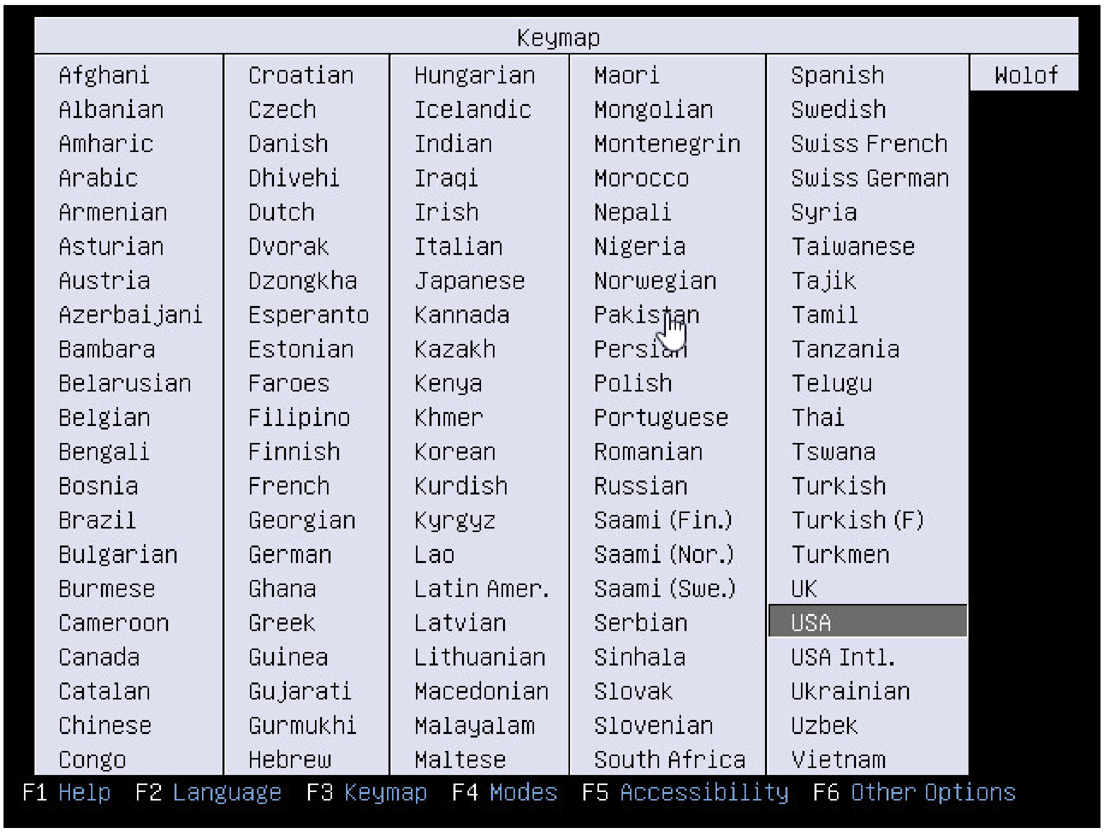


Fig. 159: The BMR Boot Screen – Keymap Selection

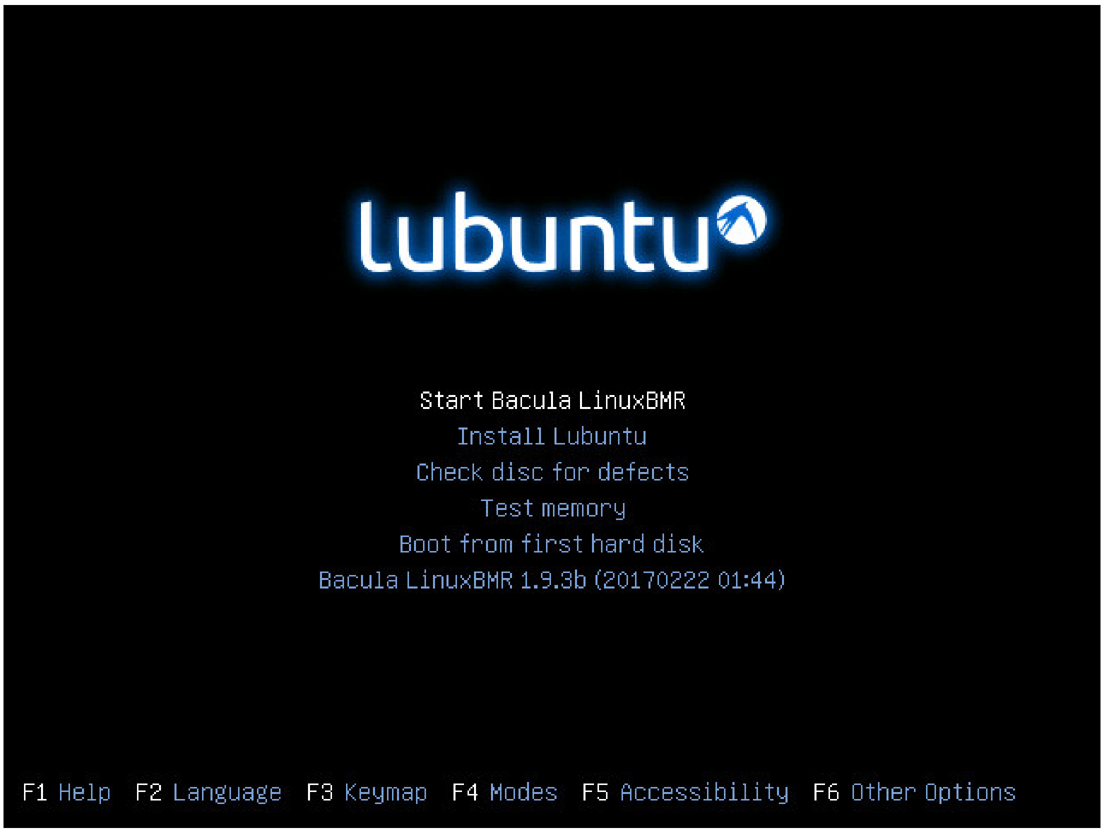

lubuntu
Start Bacula LinuxBMR
Install Lubuntu
Check disc for defects
Test memory
Boot from first hard disk
Bacula LinuxBMR 1.9.3b (20170222 01:44)
F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

Fig. 160: The BMR Boot Screen – Boot

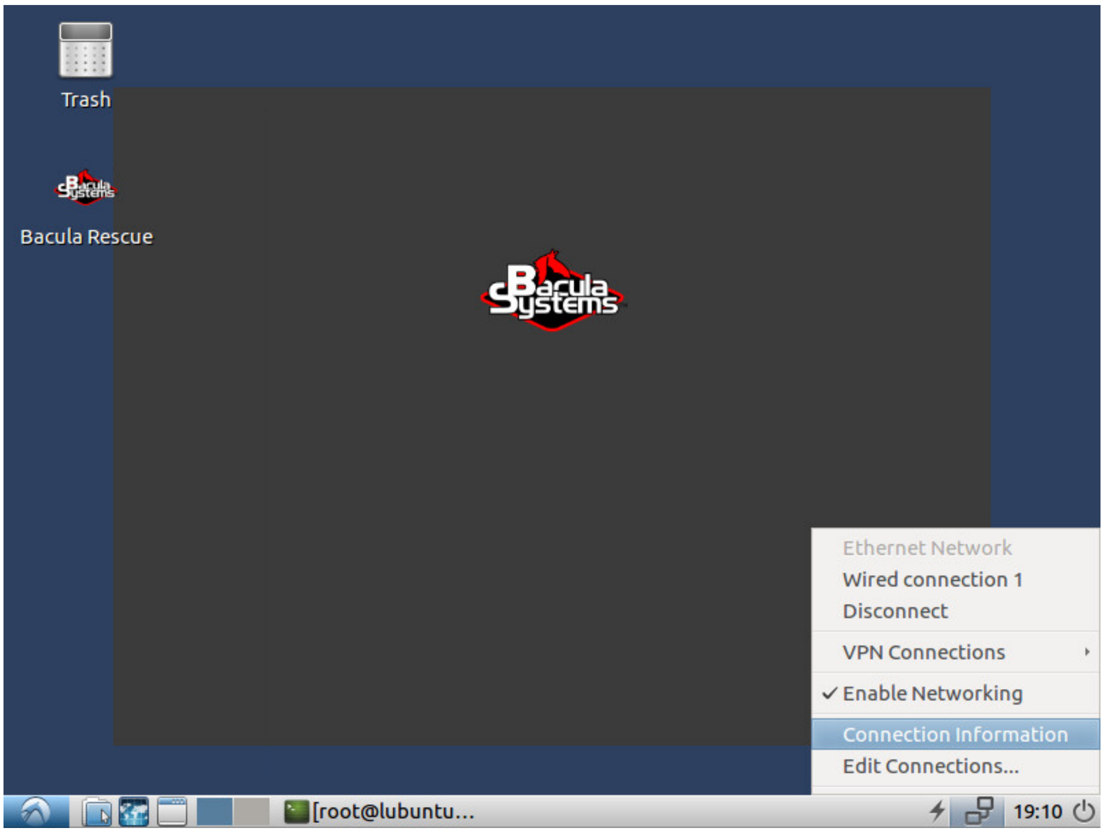
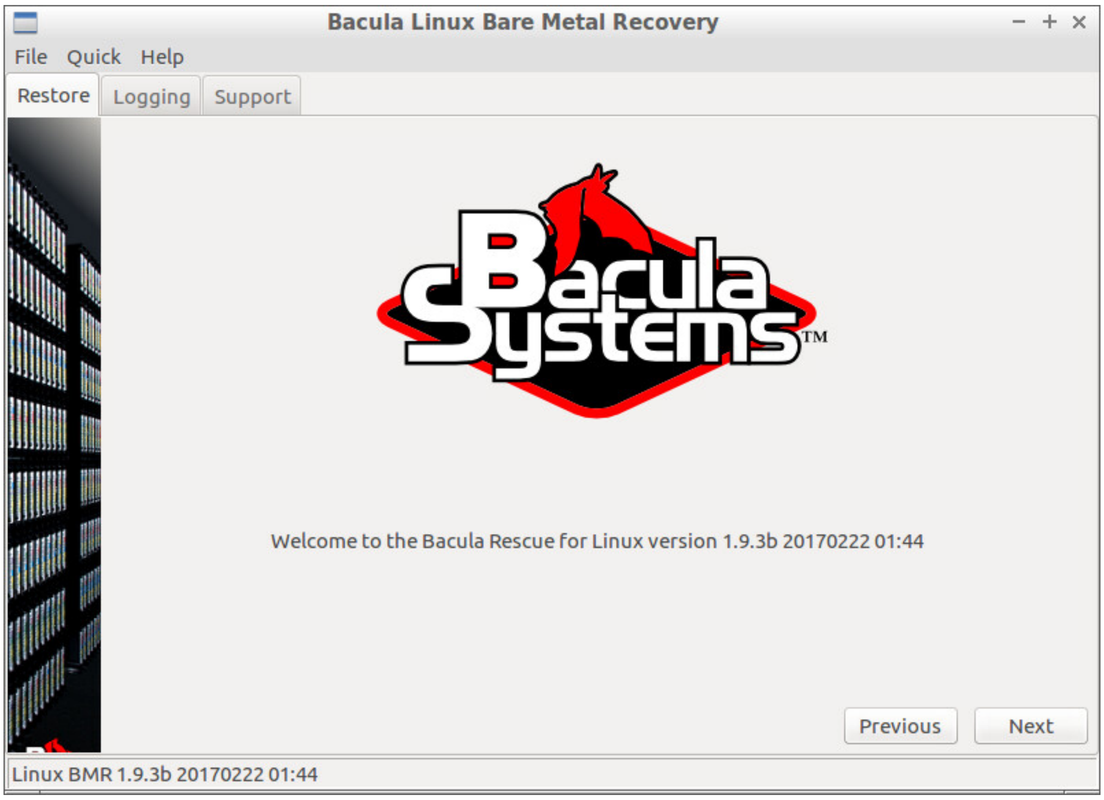


Fig. 161: The BMR System's Desktop



Welcome to the Bacula Rescue for Linux version 1.9.3b 20170222 01:44

Linux BMR 1.9.3b 20170222 01:44

Fig. 162: Welcome Page

The Configuration Screen

The next screen is the Configuration screen:

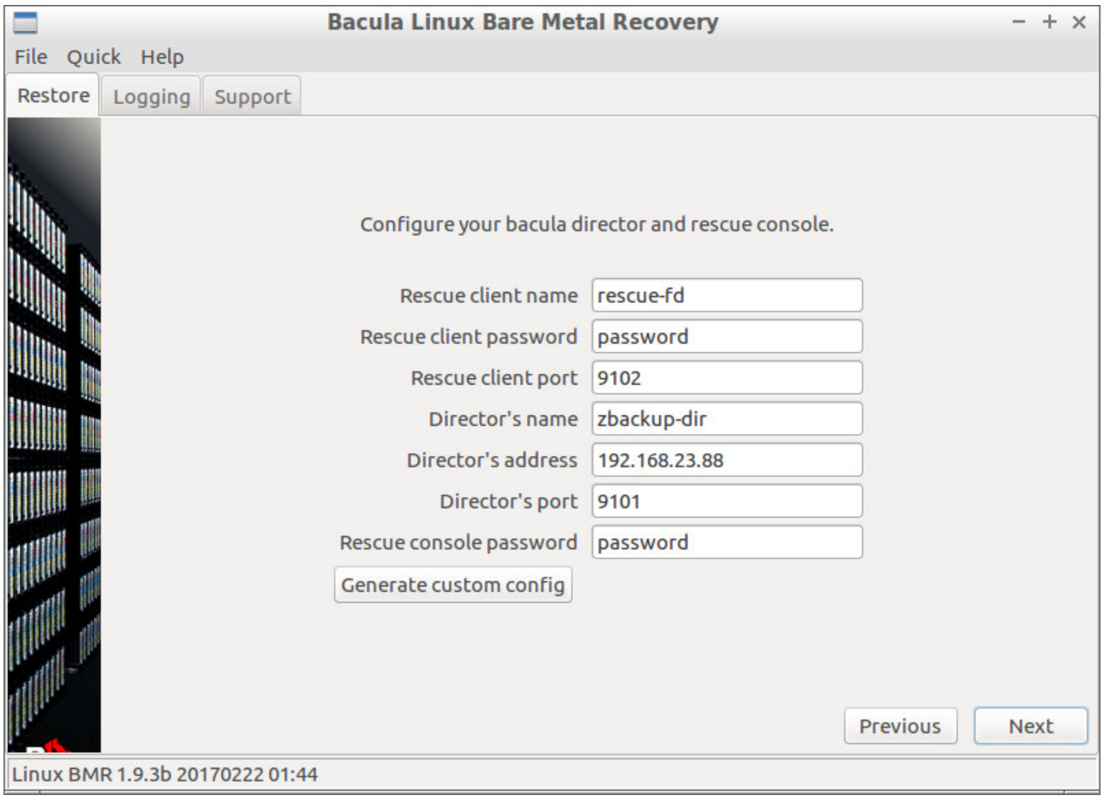
The Rescue client name, password and port have to match the ones defined in the the **Client** resource in *Preparing BMR Restore*.

The Director name, address and port are required to connect to your **Director**. Make sure you can connect to your Director from your machine: that the network is well configured, that the Name resolution (DNS) will resolve the name of your **Director** if you are not using an IP address, and that on the other side, your firewall will not block the incoming packages. The Rescue console password is the password set in *Preparing BMR Restore*. These values are used to create the client and console configuration files `bacula-fd.conf` and `bconsole.conf` in the directory `/bs-rescue`.

The "Generate custom config" button creates a copy of the client configuration file into `bacula-fd.custom` and opens it in a text editor. You may modify it and even add directives as needed. For example you can add TLS or data encryption to your configuration. Once saved, this file will be used by the BMR Client.

Click "Next" to check if the configuration is correct and then go to the next step (for connection troubleshooting see section *Connection Troubleshooting*).

Now the client daemon is running. It is accessible from the **Director**, and the client and server-side programs will communicate to restore data.



Configure your bacula director and rescue console.

Rescue client name	rescue-fd
Rescue client password	password
Rescue client port	9102
Director's name	zbackup-dir
Director's address	192.168.23.88
Director's port	9101
Rescue console password	password

Linux BMR 1.9.3b 20170222 01:44

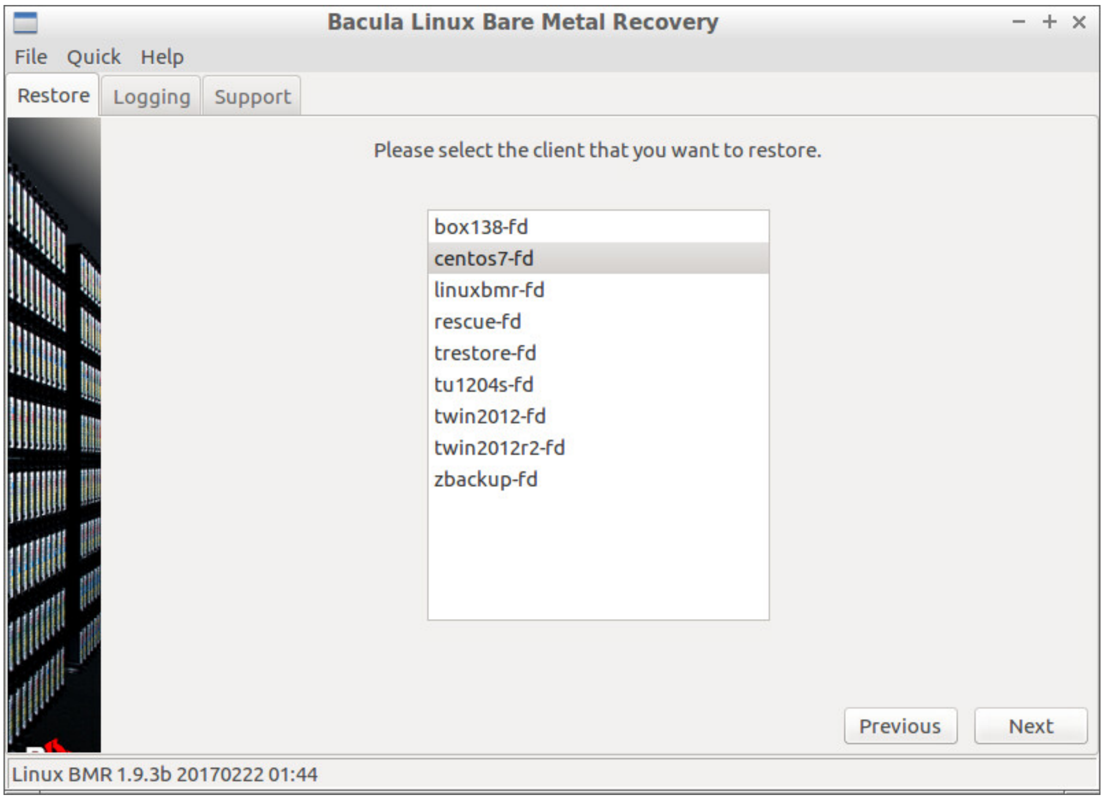
Fig. 163: Bacula Configuration

Selecting What to Restore

Now is the time to choose the system to be restored to the current host. For this purpose, the BMR tool presents you with a list of all clients known to the **Director** in alphabetical order, which looks like the one shown in the screenshot below. Select the correct client and continue by clicking the "Next" button. If you selected a client which does not have any backups available, a message stating that no backups are available is displayed and you can go back to the client selection step.

The next screen allows you to pick the date you want to restore from. All existing backups for the selected client are shown. You must select a BMR enabled backup. If the chosen job is not a good one, a dialog box will warn you. Previous versions of the LinuxBMR product were filtering the list of backups, to display only the BMR enabled backups, but this was sometimes too slow, so the filter has been removed (a simple and reliable solution is to clearly name all jobs that are BMR-enabled, for example with a job name prefix or suffix of "LinBMR").

A restore job that does not have any **RunScript** directives is automatically selected. You can choose a different one in the combobox at the bottom of the screen.



Please select the client that you want to restore.

box138-fd
centos7-fd
linuxbmr-fd
rescue-fd
trestore-fd
tu1204s-fd
twin2012-fd
twin2012r2-fd
zbackup-fd

Linux BMR 1.9.3b 20170222 01:44

Fig. 164: Source Client Selection

Selecting Where to Restore to

If a suitable backup was chosen, the BMR tool loads information about the disk layout of the source host. The same layout, or part of it will be reproduced on the target host, then the data will be restored to the freshly formated disk(s) and finally the tool will configure the boot loader.

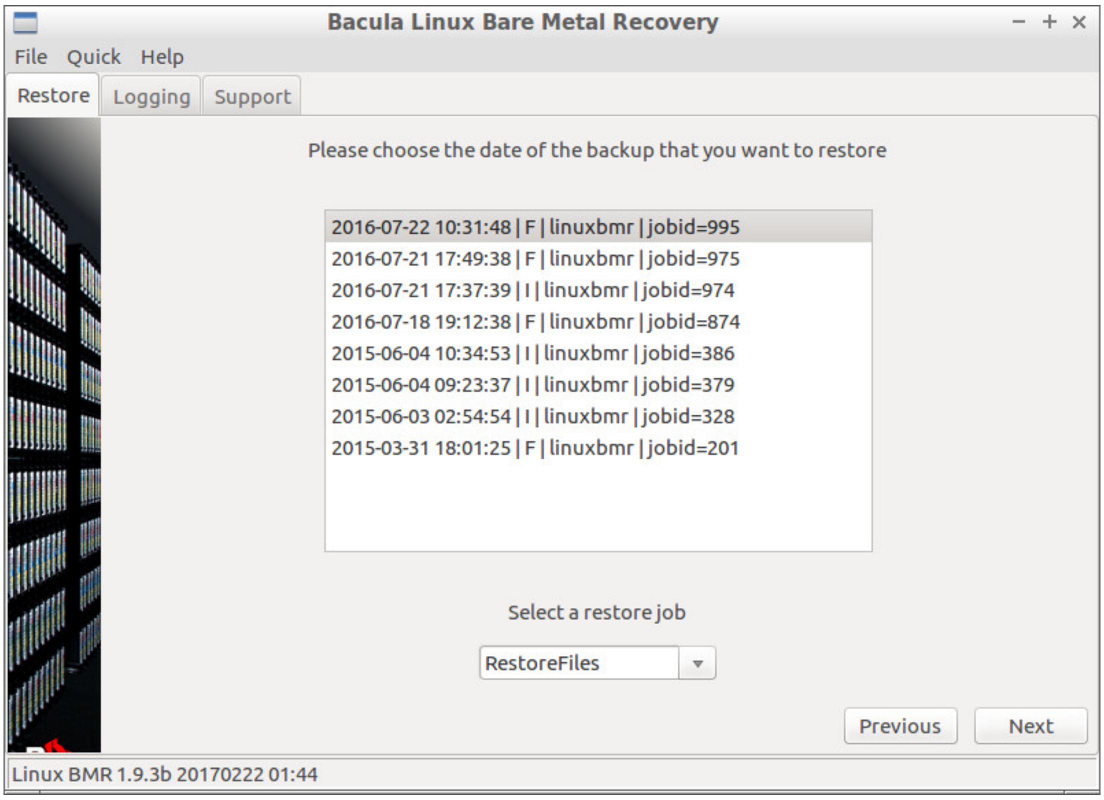
The process is split into the following operations:

- Disk mapping and exclusion
- Partition and volume resizing
- Partitioning
- Volume mapping and exclusion
- Restoring the data
- Configuring the boot loader

When the source and your target systems are identical, clicking “Next” will do the job most of the time. If the systems are different, the proposed setup will often give good results, but you should double check and then validate with “Next”, or adjust the inputs to meet your needs.

A small glossary for the following sections:

- A **disk** is a physical or virtual hard-disk.
- A **partition** is a part of a disk.



Please choose the date of the backup that you want to restore

2016-07-22 10:31:48 | F | linuxbmr | jobid=995
2016-07-21 17:49:38 | F | linuxbmr | jobid=975
2016-07-21 17:37:39 | I | linuxbmr | jobid=974
2016-07-18 19:12:38 | F | linuxbmr | jobid=874
2015-06-04 10:34:53 | I | linuxbmr | jobid=386
2015-06-04 09:23:37 | I | linuxbmr | jobid=379
2015-06-03 02:54:54 | I | linuxbmr | jobid=328
2015-03-31 18:01:25 | F | linuxbmr | jobid=201

Select a restore job

RestoreFiles

Linux BMR 1.9.3b 20170222 01:44

Fig. 165: Source Date Selection

- A **volume** is a partition, a logical volume built on top of LVM, or a device created by the Linux RAID Software, that holds a filesystem and can be mounted to a mount point.
- A **mount point** is a path inside the filesystem where a volume is mounted. We will often use the path as an alias to refer to the underlying volume.

Disk Mapping

The next screenshot shows what the Disk mapping screen looks like. The two lists on the left show the disks of the source host (from backup) and the ones available on the target host. The right panel shows how volumes and their related mount points depend on the underlying disks of the source host.

The task here is to map and align every disk of the source host to one disk of the target. This is useful if you have disks that are different in size or speed.

Use the up and down arrows on the left to move a selected disk up and down in the left list. Use the right arrows to move the selected disk on the right list.

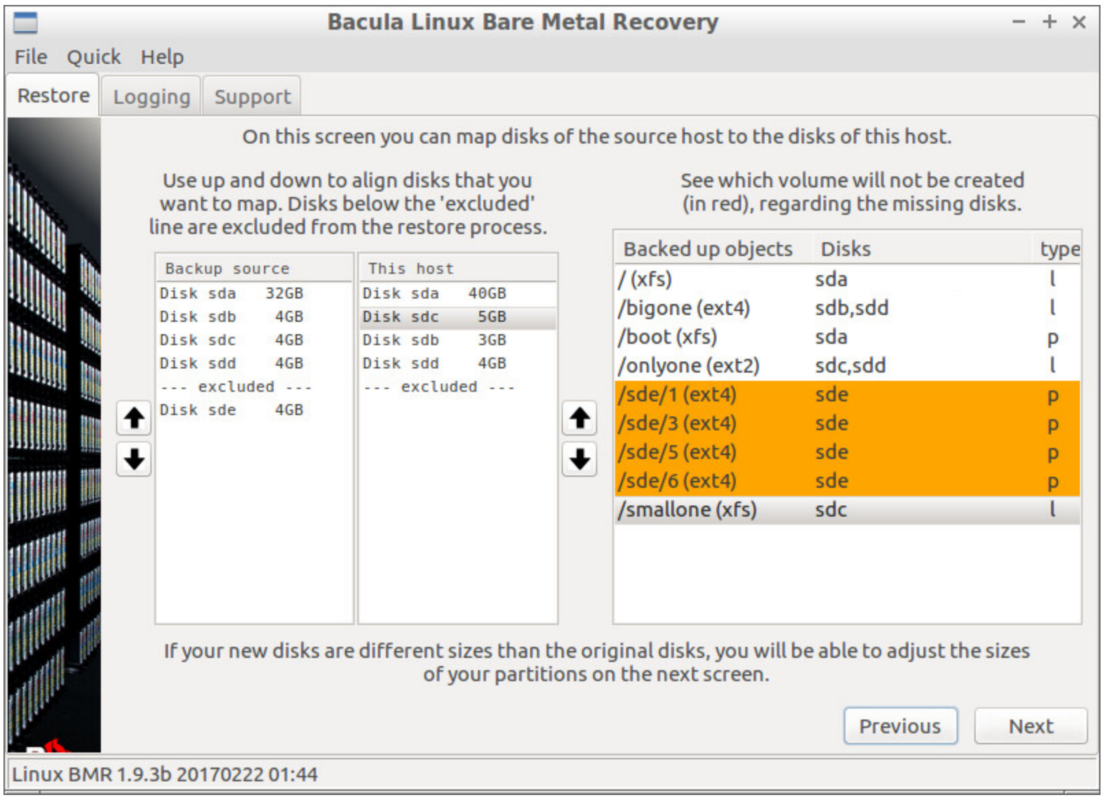
If the number of disks on both systems does not match, then some disks get excluded and are moved below the `- excluded -` separation line in order to have the same number of disks above the line for a one to one mapping.

You can move some disks below this separation line yourself to exclude them during the restore process.

Note: Existing `LVM` and `RAID` software information on these excluded disks **will be deleted** to avoid name collisions, but the partitions will remain untouched.

All volumes and filesystems that are related to these excluded disks will not be created and are shown in orange on the right panel.

The first disk of the target host will be the one the boot loader is installed to. It must also be chosen as the bootable disk in your BIOS. This is often the first disk in the list.



On this screen you can map disks of the source host to the disks of this host.

Use up and down to align disks that you want to map. Disks below the 'excluded' line are excluded from the restore process.

See which volume will not be created (in red), regarding the missing disks.

```
Backup source        This host
Disk sda    32GB     Disk sda    40GB
Disk sdb     4GB     Disk sdc     5GB
Disk sdc     4GB     Disk sdb     3GB
Disk sdd     4GB     Disk sdd     4GB
--- excluded ---     --- excluded ---
Disk sde     4GB
```

Backed up objects	Disks	type
/ (xfs)	sda	l
/bigone (ext4)	sdb,sdd	l
/boot (xfs)	sda	p
/onlyone (ext2)	sdc,sdd	l
/sde/1 (ext4)	sde	p
/sde/3 (ext4)	sde	p
/sde/5 (ext4)	sde	p
/sde/6 (ext4)	sde	p
/smallone (xfs)	sdc	l

If your new disks are different sizes than the original disks, you will be able to adjust the sizes of your partitions on the next screen.

Linux BMR 1.9.3b 20170222 01:44

Fig. 166: Disk Mapping between Source and Target hosts

In the screenshot above you can see that the target host does not have a 5th disk. Thus, the 4 volumes `/sde/[1356]` in orange on the the right panel will not be created. We could choose to exclude another disk but assume that `sde` is fine. You can also see that we swapped `sdb` and `sdc`, because the new `sdc` is bigger and we want to give this extra space to the `/bigone` volume.

Adjusting Partitions and Volumes Sizes

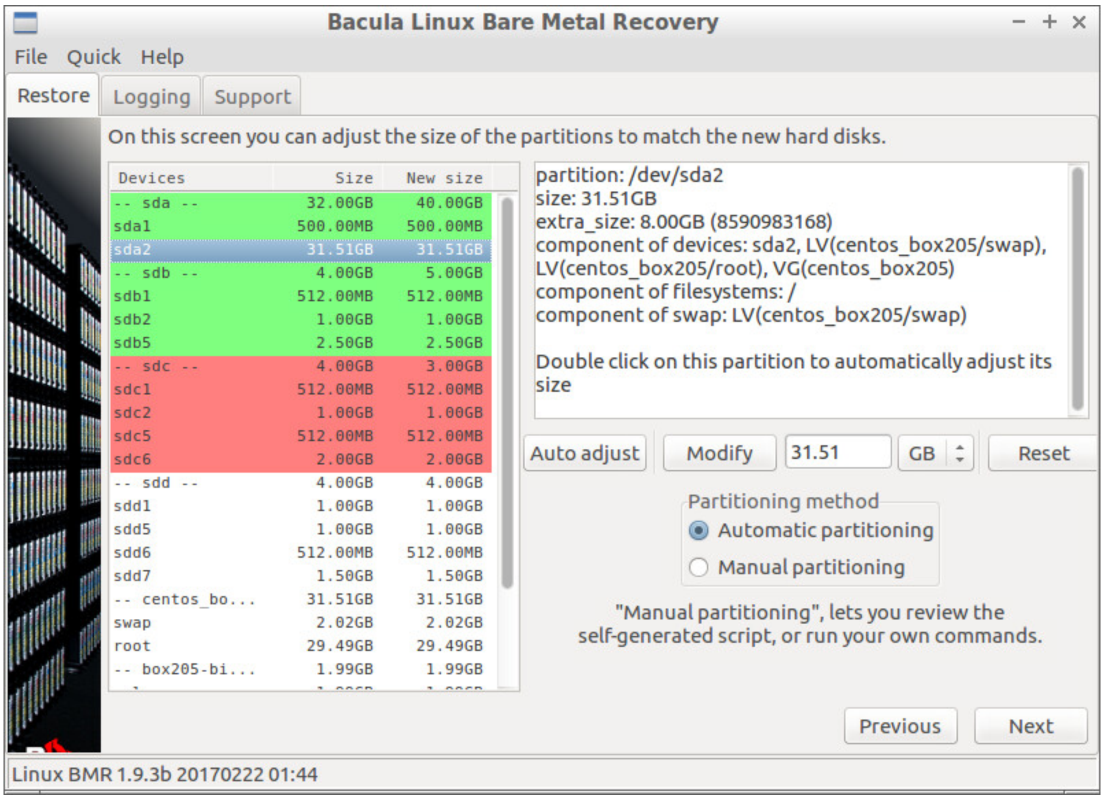
The same partitions and volumes as backed up will be created on the target host. You cannot add or remove any of them. If your disks are bigger or smaller than the source disks then you will have to choose which partitions or volumes will grow or shrink.

In the next screenshot you can see, on the left side, all partitions and volumes, and on the the right side the information related to the selected object on the left. For now, ignore the *Partitioning method* part on the bottom right that is described at the end of this section.

The areas in green have some extra space that can be assigned to the partitions and *Logical Volumes*. The areas in red will require some of their components to be shrunk.

`sda` is shown in green, used to have a size of 32GB on the source system and can now take advantage of 8 more GB. We can distribute this extra space between its partitions `sda1` and `sda2`.

To do that you must select the first partition `sda1`, then type the new size in the field at the right of the "Modify" button and click "Modify". You will see the new size allocated to the partition. Then select



On this screen you can adjust the size of the partitions to match the new hard disks.

```
Devices              Size      New size
-- sda --          32.00GB     40.00GB
sda1              500.00MB    500.00MB
sda2               31.51GB     31.51GB
-- sdb --           4.00GB      5.00GB
sdb1              512.00MB    512.00MB
sdb2                1.00GB      1.00GB
sdb5                2.50GB      2.50GB
-- sdc --           4.00GB      3.00GB
sdc1              512.00MB    512.00MB
sdc2                1.00GB      1.00GB
sdc5              512.00MB    512.00MB
sdc6                2.00GB      2.00GB
-- sdd --           4.00GB      4.00GB
sdd1                1.00GB      1.00GB
sdd5                1.00GB      1.00GB
sdd6              512.00MB    512.00MB
sdd7                1.50GB      1.50GB
-- centos_bo...    31.51GB     31.51GB
swap                2.02GB      2.02GB
root               29.49GB     29.49GB
-- box205-bi...     1.99GB      1.99GB
```

partition: /dev/sda2
size: 31.51GB
extra_size: 8.00GB (8590983168)
component of devices: sda2, LV(centos_box205/swap), LV(centos_box205/root), VG(centos_box205)
component of filesystems: /
component of swap: LV(centos_box205/swap)

Double click on this partition to automatically adjust its size

31.51 GB

Partitioning method
[✓ Automatic partitioning]
Manual partitioning

"Manual partitioning", lets you review the self-generated script, or run your own commands.

Linux BMR 1.9.3b 20170222 01:44

Fig. 167: Disk Resizing `sda`

the other partition and do the same. To go faster you can give it all the remaining space by clicking the "Auto adjust" button. When all the extra space is allocated the disk and its partitions are not highlighted anymore.

Often you don't want to divide the extra space and just give it all to one partition or volume. To go faster, double click on the partition that must grow (or shrink) and all the extra space is added (or lowered) to (or from) this partition or volume.

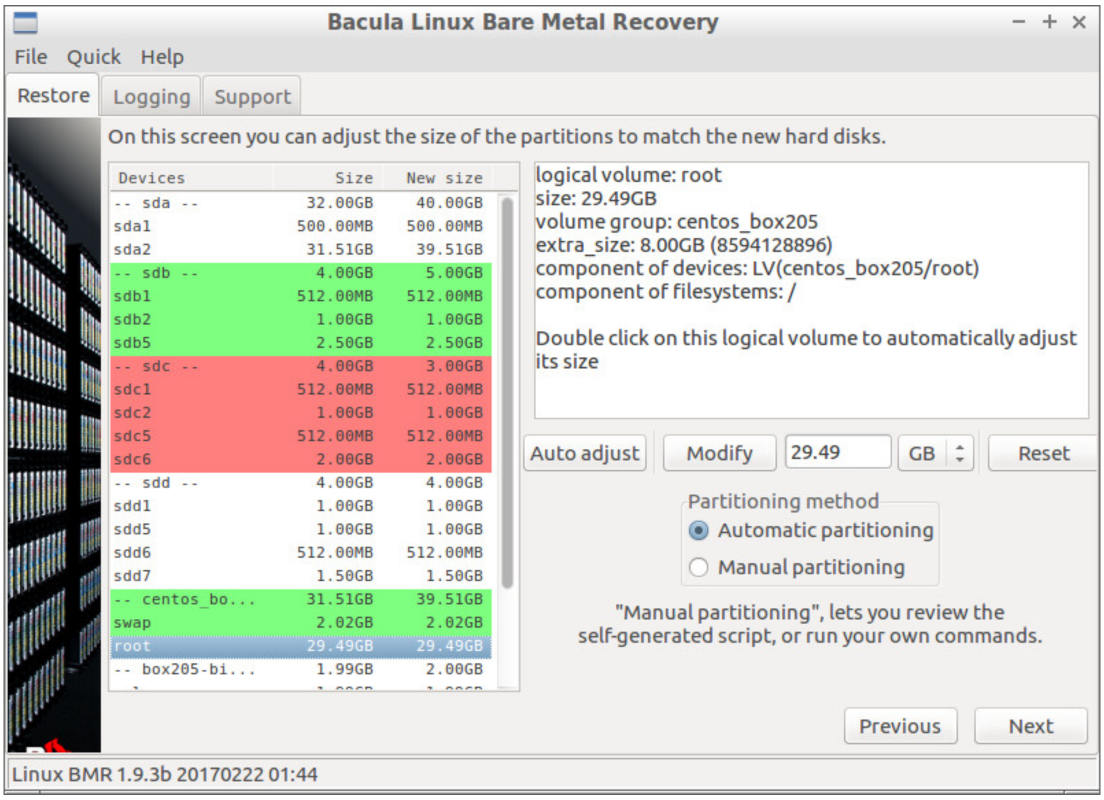
This is what we did with partition `sda2`. The `sda` disk is not green anymore, but the "Volume Group centos_box205" has become green. This *Volume Group* that got some extra space from the enlargement of `sda2` has 2 *Logical Volumes* that hold a `swap` area and the `root` filesystem. As we don't want to increase the swap space, we double-click the `root` volume to give it all the space. See the result in the next screenshot below.

Now we handle the `sdc` disk that is smaller on the new system. We must shrink one ore more partitions to save 1GB. Double clicking on `sdc1` would return an error because the partition is too small to take all space difference on its own. We choose to double click and shrink `sdc6` instead.

Finally we see that `sdc6` has lost 1GB.

And finally we increase the size of `sdb2` and give more space to the `/bigone` filesystem (not visible in screenshot because that filesystem is at the bottom of the list).

Note: Most of the sizes here are approximate because tools used to create the partitions and `LVM` objects depend on their own alignment rules and granularity constraints. To make it work, we let the tool make the adjustments on the last partition of the disk or the last Logical Volume of the Volume Group. This means that the last object will be a few KB or MB smaller than expected.



On this screen you can adjust the size of the partitions to match the new hard disks.

Devices	Size	New size
-- sda --	32.00GB	40.00GB
sda1	500.00MB	500.00MB
sda2	31.51GB	39.51GB
-- sdb --	4.00GB	5.00GB
sdb1	512.00MB	512.00MB
sdb2	1.00GB	1.00GB
sdb5	2.50GB	2.50GB
-- sdc --	4.00GB	3.00GB
sdc1	512.00MB	512.00MB
sdc2	1.00GB	1.00GB
sdc5	512.00MB	512.00MB
sdc6	2.00GB	2.00GB
-- sdd --	4.00GB	4.00GB
sdd1	1.00GB	1.00GB
sdd5	1.00GB	1.00GB
sdd6	512.00MB	512.00MB
sdd7	1.50GB	1.50GB
-- centos_bo...	31.51GB	39.51GB
swap	2.02GB	2.02GB
root	29.49GB	29.49GB
-- box205-bi...	1.99GB	2.00GB

logical volume: root
size: 29.49GB
volume group: centos_box205
extra_size: 8.00GB (8594128896)
component of devices: LV(centos_box205/root)
component of filesystems: /

Double click on this logical volume to automatically adjust its size

Partitioning method
[✓ Automatic partitioning]
Manual partitioning

"Manual partitioning", lets you review the self-generated script, or run your own commands.

Linux BMR 1.9.3b 20170222 01:44

Fig. 168: Disk Resizing Root

For your information, the program displays an entire disk or Volume Group in green or red if the difference between the expected size and the available size is more than 4MB.

Even though there should never be any green or red areas left, the program will not complain about it and will try to have the partitioning and LVM tools do the adjustment on the last partition or Logical Volume. If the adjustment cannot be done (for example because there is no more space to create any partition or *Logical Volume* the partitioning will fail.

If you need precise sizing you can tweak the partitioning script yourself, which is described in the next section.

Partitioning

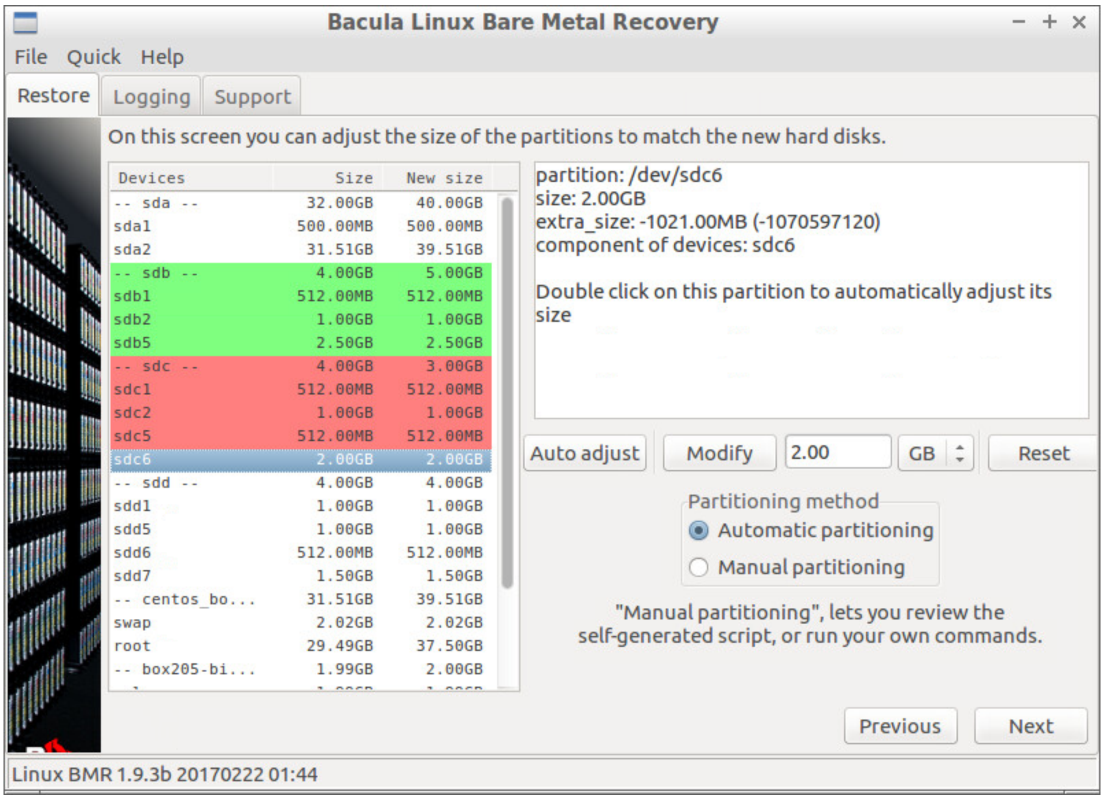
Finally we are done with the resizing and we can create the partitions and volumes. The BMR tool generates a shell script using the information coming from the source system and the choices made in the last two screens.

If you have selected "Automatic partitioning" and the script causes an error (see example below) you can read this section and fix the problem yourself, or contact the **Bacula Systems** Support Team.

If the generated script ran without error then the BMR tool skips the manual partitioning screen. Continue to read *Volume Mapping*.

If you are not interested in creating your own scripts or how to tweak the auto generated script, you can skip this section.

The script `mkpart-auto-generated.sh` in the list is the one that the BMR has generated. The two other scripts are here as a reminder that it is possible to write your own scripts in advance and store



On this screen you can adjust the size of the partitions to match the new hard disks.

Devices	Size	New size
-- sda --	32.00GB	40.00GB
sda1	500.00MB	500.00MB
sda2	31.51GB	39.51GB
-- sdb --	4.00GB	5.00GB
sdb1	512.00MB	512.00MB
sdb2	1.00GB	1.00GB
sdb5	2.50GB	2.50GB
-- sdc --	4.00GB	3.00GB
sdc1	512.00MB	512.00MB
sdc2	1.00GB	1.00GB
sdc5	512.00MB	512.00MB
sdc6	2.00GB	2.00GB
-- sdd --	4.00GB	4.00GB
sdd1	1.00GB	1.00GB
sdd5	1.00GB	1.00GB
sdd6	512.00MB	512.00MB
sdd7	1.50GB	1.50GB
-- centos_bo...	31.51GB	39.51GB
swap	2.02GB	2.02GB
root	29.49GB	37.50GB
-- box205-bi...	1.99GB	2.00GB

```
partition: /dev/sdc6
size: 2.00GB
extra_size: -1021.00MB (-1070597120)
component of devices: sdc6

Double click on this partition to automatically adjust its
size
```

"Manual partitioning", lets you review the self-generated script, or run your own commands.

Linux BMR 1.9.3b 20170222 01:44

Fig. 169: Disk Resizing sdc6

them in the `/opt/bacula/rescue/scripts` directory of your source host. Such custom scripts will be shown in the list if their names start with `mkpart-`.

You can edit and run the scripts using the "Edit script" button on the right. Their output is displayed in the small window at the bottom. The "Help" button provides information about what you have to do, how you can do it, and all the details about the source system.

Before leaving this screen, the BMR program reminds you that it is going to restore the data, and that the directory `/target` must be mounted and ready to receive the data.

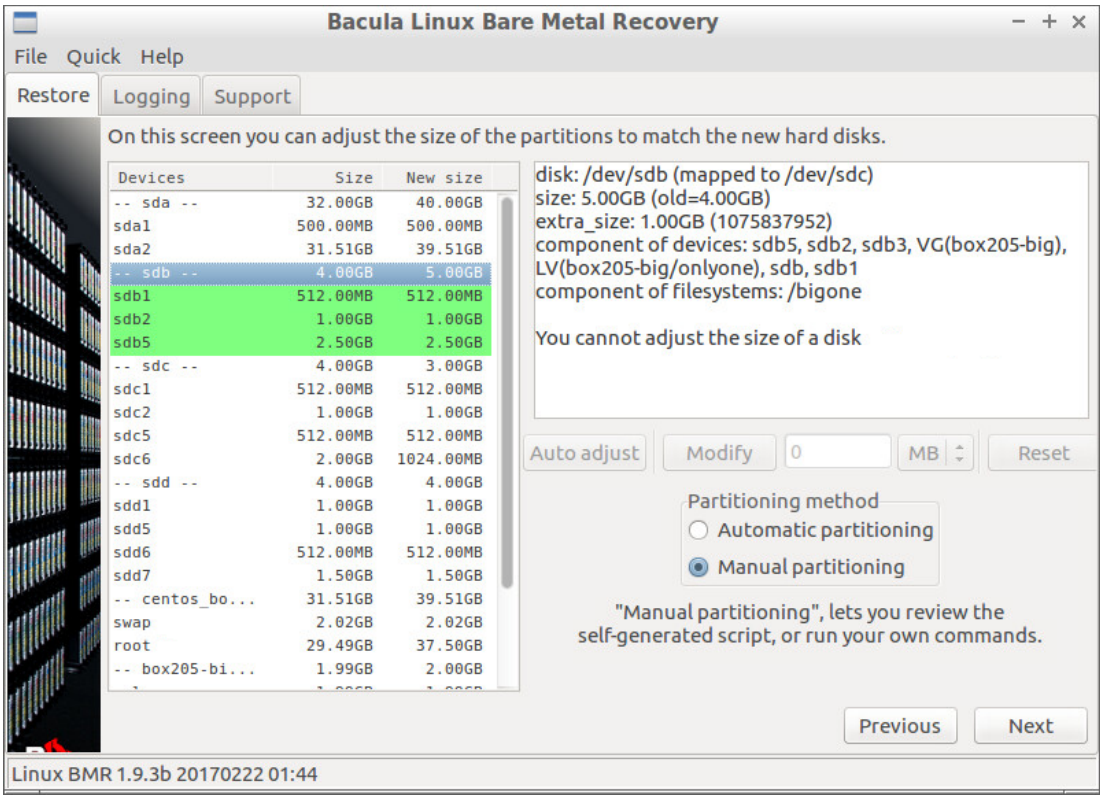
Volume Mapping

We are one step away from restoring the data and the purpose of the next screen is to match the volumes that have been backed up from the source host with the volumes that have been created by the Partitioning process on the target host.

In our example, we had no 5th disk to map the disk `sde`. As seen in the figure above, the program will not restore data in the directories that were stored on the missing disk – this is why it says *None* in the "Restore to" column. But we are in the process of redirecting the data in directory `/sde/1` to directory `/target/bigone` which we know to be big enough to fit the extra data.

You can double click on any volume of the source system and change the directory on the target system.

When done, click "Next" to start restoring your data. A progress bar will keep you informed about the status of the restore.



On this screen you can adjust the size of the partitions to match the new hard disks.

```
Devices            Size       New size
-- sda --          32.00GB    40.00GB
sda1               500.00MB   500.00MB
sda2               31.51GB    39.51GB
-- sdb --          4.00GB     5.00GB
sdb1               512.00MB   512.00MB
sdb2               1.00GB     1.00GB
sdb5               2.50GB     2.50GB
-- sdc --          4.00GB     3.00GB
sdc1               512.00MB   512.00MB
sdc2               1.00GB     1.00GB
sdc5               512.00MB   512.00MB
sdc6               2.00GB     1024.00MB
-- sdd --          4.00GB     4.00GB
sdd1               1.00GB     1.00GB
sdd5               1.00GB     1.00GB
sdd6               512.00MB   512.00MB
sdd7               1.50GB     1.50GB
-- centos_bo...    31.51GB    39.51GB
swap               2.02GB     2.02GB
root               29.49GB    37.50GB
-- box205-bi...    1.99GB     2.00GB
```

disk: /dev/sdb (mapped to /dev/sdc)
size: 5.00GB (old=4.00GB)
extra_size: 1.00GB (1075837952)
component of devices: sdb5, sdb2, sdb3, VG(box205-big), LV(box205-big/onlyone), sdb, sdb1
component of filesystems: /bigone

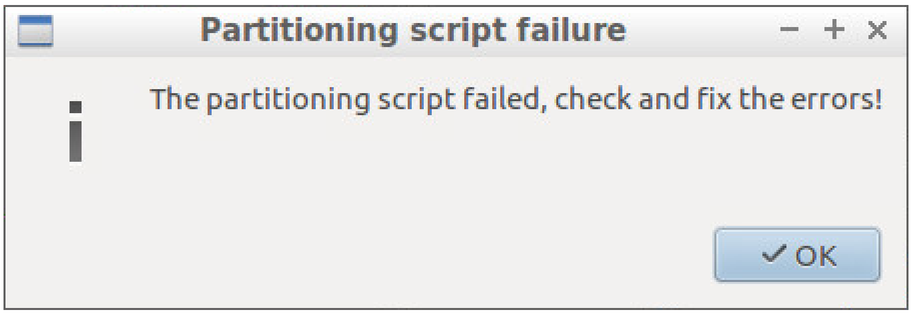
You cannot adjust the size of a disk

( ) Automatic partitioning
(●) Manual partitioning

"Manual partitioning", lets you review the self-generated script, or run your own commands.

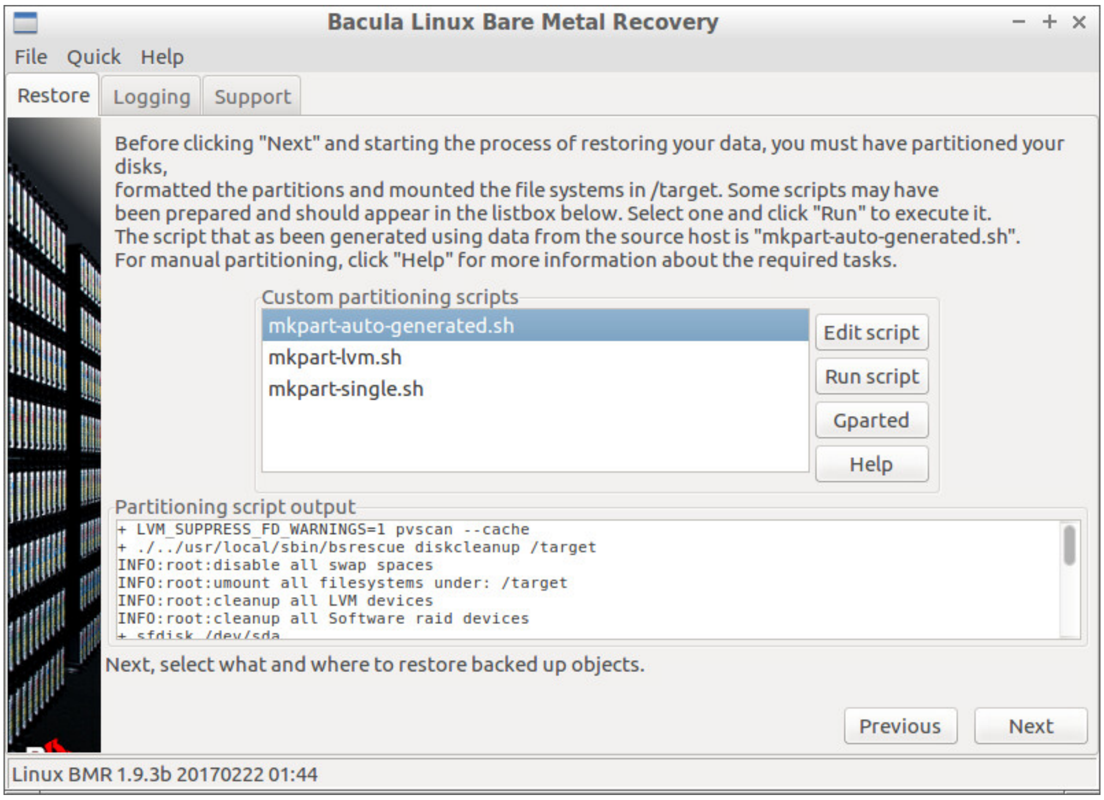
Linux BMR 1.9.3b 20170222 01:44

Fig. 170: Disk Resizing Final



The partitioning script failed, check and fix the errors!

Fig. 171: Partitioning Failure



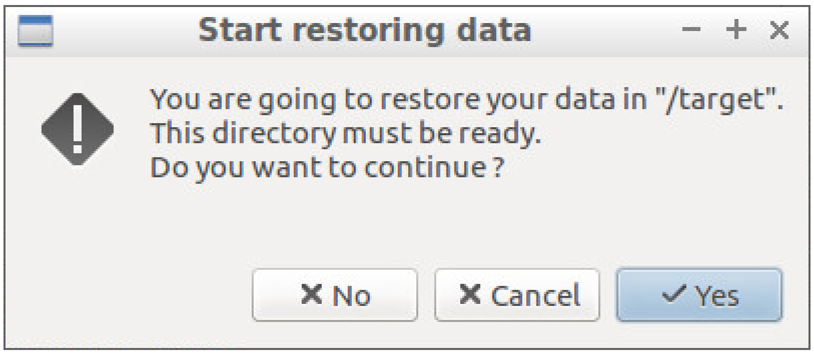
Before clicking "Next" and starting the process of restoring your data, you must have partitioned your disks,
formatted the partitions and mounted the file systems in /target. Some scripts may have
been prepared and should appear in the listbox below. Select one and click "Run" to execute it.
The script that as been generated using data from the source host is "mkpart-auto-generated.sh".
For manual partitioning, click "Help" for more information about the required tasks.

```
+ LVM_SUPPRESS_FD_WARNINGS=1 pvscan --cache
+ ./../usr/local/sbin/bsrescue diskcleanup /target
INFO:root:disable all swap spaces
INFO:root:umount all filesystems under: /target
INFO:root:cleanup all LVM devices
INFO:root:cleanup all Software raid devices
+ sfdisk /dev/sda
```

Next, select what and where to restore backed up objects.

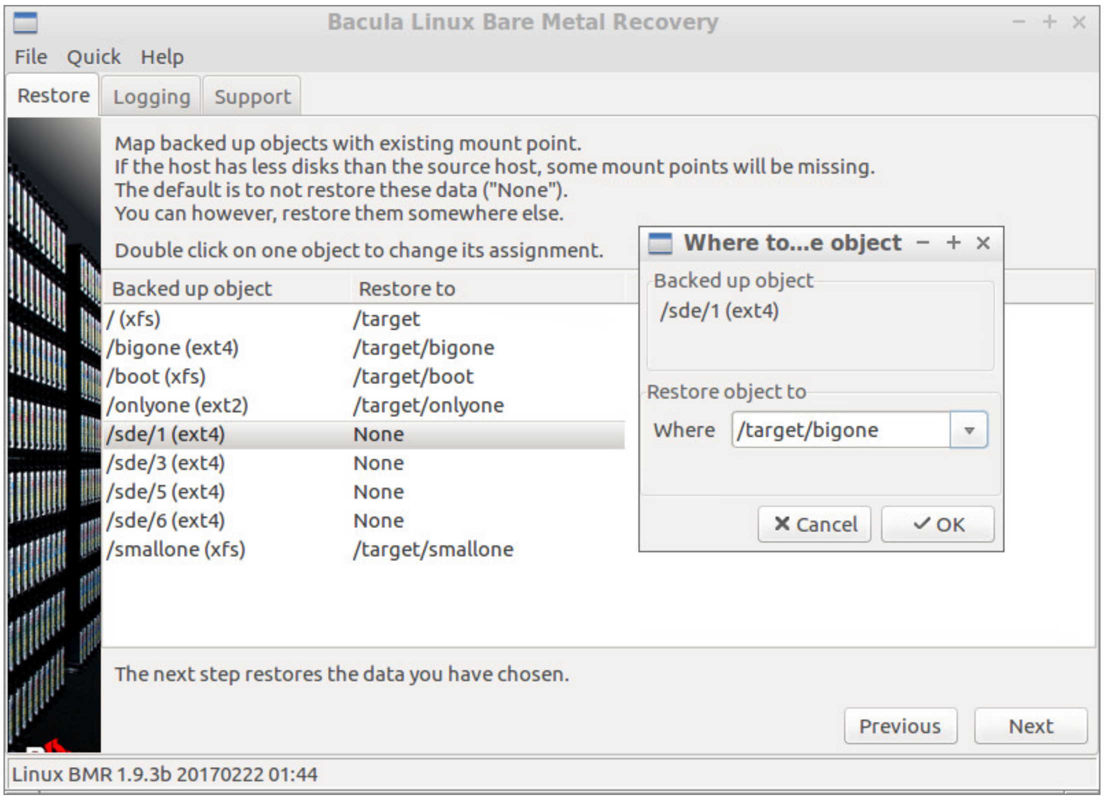
Linux BMR 1.9.3b 20170222 01:44

Fig. 172: Manual Partitioning



You are going to restore your data in "/target".
This directory must be ready.
Do you want to continue ?

Fig. 173: `/target` Must be Ready to get the Data



Map backed up objects with existing mount point.
If the host has less disks than the source host, some mount points will be missing.
The default is to not restore these data ("None").
You can however, restore them somewhere else.

Double click on one object to change its assignment.

Backed up object	Restore to
/ (xfs)	/target
/bigone (ext4)	/target/bigone
/boot (xfs)	/target/boot
/onlyone (ext2)	/target/onlyone
/sde/1 (ext4)	None
/sde/3 (ext4)	None
/sde/5 (ext4)	None
/sde/6 (ext4)	None
/smallone (xfs)	/target/smallone

Where to...e object

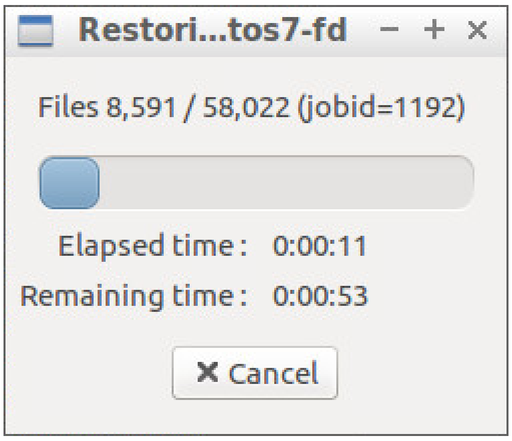
Backed up object
/sde/1 (ext4)

Restore object to
Where /target/bigone

The next step restores the data you have chosen.

Linux BMR 1.9.3b 20170222 01:44

Fig. 174: Volume Mapping



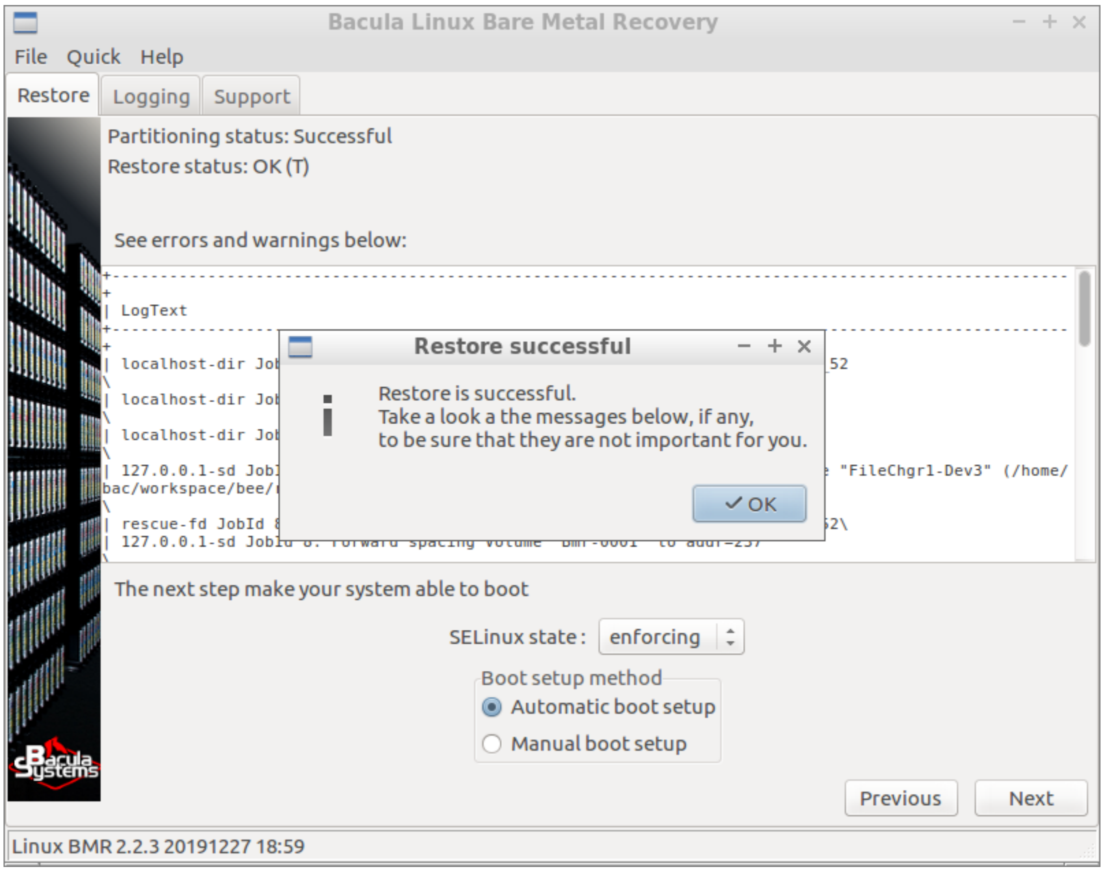
Files 8,591 / 58,022 (jobid=1192)

Elapsed time : 0:00:11
Remaining time : 0:00:53

Fig. 175: Restore Gauge

Restore and Boot Loader Configuration

Once restore of the data is complete, the BMR program invites you to look at the joblog of the restore job. Even if a successful restore would be good news, the log should be reviewed for warnings or other minor errors. The same screen in figure *After Restore* asks to choose between automatic and manual configuration of the boot loader.



Partitioning status: Successful
Restore status: OK (T)

See errors and warnings below:

```
+----------------------------------------------------------------------------------
+
| LogText
+----------------------------------------------------------------------------------
+
| localhost-dir Jo                                                              _52
\
| localhost-dir Jo
\
| localhost-dir Jo
\
| 127.0.0.1-sd JobI                                            "FileChgr1-Dev3" (/home/
bac/workspace/bee/
\
| rescue-fd JobId                                                               2\
| 127.0.0.1-sd JobI
\
```

Restore successful

Restore is successful.
Take a look a the messages below, if any,
to be sure that they are not important for you.

The next step make your system able to boot

SELinux state : enforcing

Boot setup method
- [✓] Automatic boot setup
- [ ] Manual boot setup

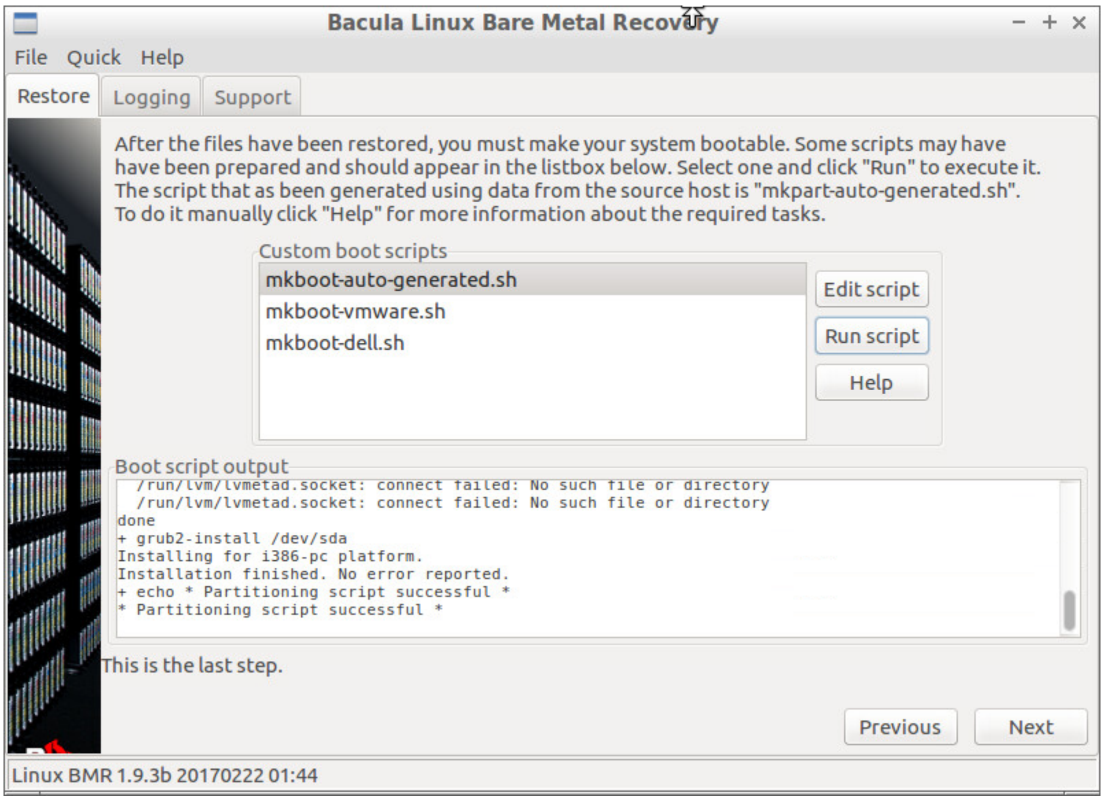
Linux BMR 2.2.3 20191227 18:59

Fig. 176: After Restore

The process is similar to the partitioning process seen before: The BMR program generates a shell script. If you choose "Manual boot setup" or if the auto-generated script has caused errors then the program enters the Boot Loader Manual Configuration screen presented in figure *Boot Loader Manual Configuration*.

You can change the SELinux mode using the combo box. This can be convenient if you have some problems at reboot time. Have a look at section *Restoring a system with SELinux enabled*.

The script `mkboot-auto-generated.sh` in the list is the one that the BMR program has generated. The two other scripts are here to remind you that you can prepare your own scripts in advance and store them in the `/opt/bacula/rescue/scripts` directory of the source host. These custom scripts will be available in the list if their names start with `mkboot-`. You can edit and run the scripts using the buttons on the right. The output is displayed in the small window at the bottom. The "Help" button provides a lot of information about what you have to do, how you can do it, and all the details about the source system.



After the files have been restored, you must make your system bootable. Some scripts may have have been prepared and should appear in the listbox below. Select one and click "Run" to execute it. The script that as been generated using data from the source host is "mkpart-auto-generated.sh". To do it manually click "Help" for more information about the required tasks.

```
  /run/lvm/lvmetad.socket: connect failed: No such file or directory
  /run/lvm/lvmetad.socket: connect failed: No such file or directory
done
+ grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
+ echo * Partitioning script successful *
* Partitioning script successful *
```

This is the last step.

Fig. 177: Boot Loader Manual Configuration

BMR Final

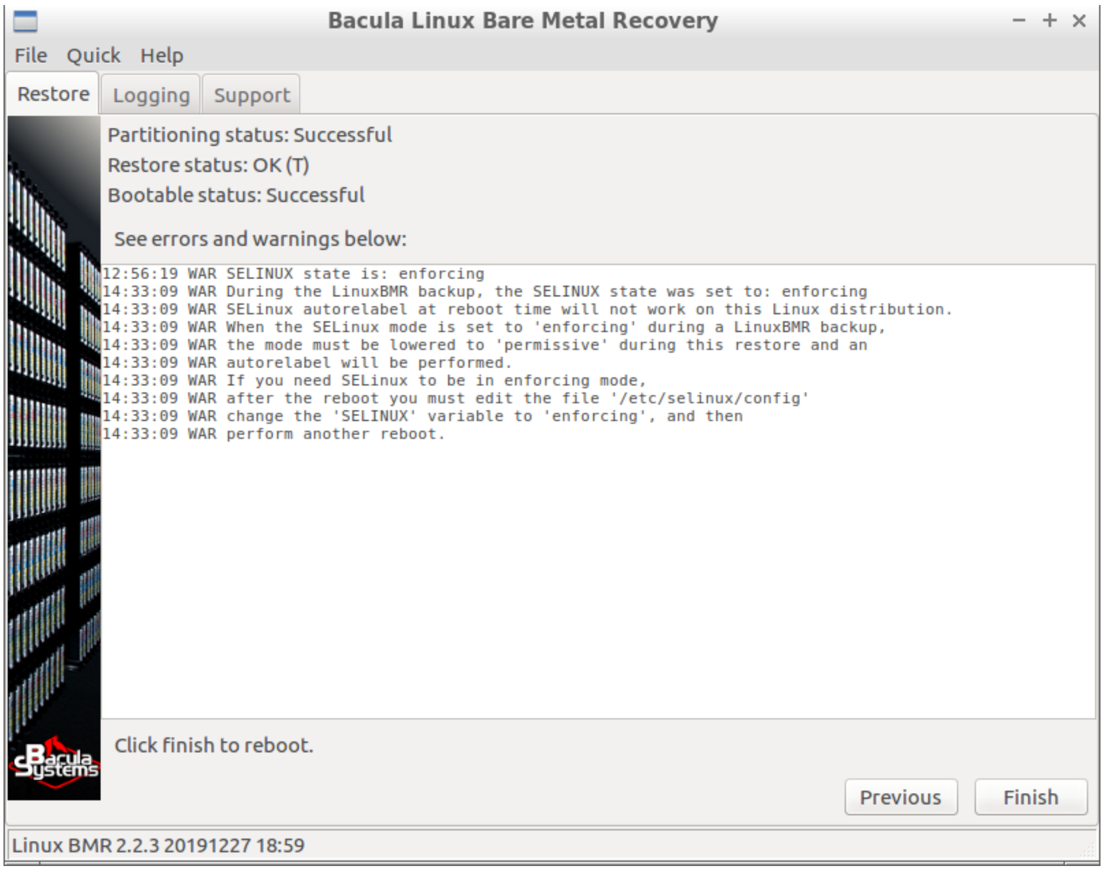
You have reached the final screen. It shows the status of all important operations and all events with a level or "warning" or "error" that the BMR program has encountered. Figure *BMR Final* also shows how mapped disk contents have been handled which caused some lines to be removed from the `fstab` file accordingly.

Before clicking the "Finish" button to reboot the system, you can do some interesting things:

You can inspect the restored data using any of the tools available on the Recovery system, or using a simple terminal. The restored data is in the the `/target` directory.

You can do some operations inside the restored system. Start a terminal, become root using the command `sudo su -` then type command `chroot /target` to operate in the freshly restored system like if it was your root system.

You can generate and download the report of the full BMR process in a file `bssupport.zip` that you could send to our support in case your system would not reboot or would not meet your expectations. The report can be generated in the "Support" tab described in section *The Support tab*.



Partitioning status: Successful
Restore status: OK (T)
Bootable status: Successful

See errors and warnings below:

```
12:56:19 WAR SELINUX state is: enforcing
14:33:09 WAR During the LinuxBMR backup, the SELINUX state was set to: enforcing
14:33:09 WAR SELinux autorelabel at reboot time will not work on this Linux distribution.
14:33:09 WAR When the SELinux mode is set to 'enforcing' during a LinuxBMR backup,
14:33:09 WAR the mode must be lowered to 'permissive' during this restore and an
14:33:09 WAR autorelabel will be performed.
14:33:09 WAR If you need SELinux to be in enforcing mode,
14:33:09 WAR after the reboot you must edit the file '/etc/selinux/config'
14:33:09 WAR change the 'SELINUX' variable to 'enforcing', and then
14:33:09 WAR perform another reboot.
```

Click finish to reboot.

Linux BMR 2.2.3 20191227 18:59

Fig. 178: BMR Final

Restoring a system with SELinux enabled

The BMR program can restore *SELinux* enabled systems. By default, the program creates the file `/.autorelabel` and lets the system reset all *SELinux* attributes at first reboot. The system often reboots a second time after the reset of the attributes.

Unfortunately, for unknown reasons this procedure doesn't work anymore with some recent Linux distributions when the *enforcing* mode is used. The workaround is to switch the mode to *permissive* just after the restore using the *combobox* in figure *Boot Loader Manual Configuration*, and let the system reset the attributes.

When the system is rebooted for the second time, the administrator has to login and edit the file `/etc/syslinux/selinux` to change the *SELinux* mode back to *enforcing* before rebooting one final time.

If you have problems at reboot time and you know that the *SELinux* mode is set to enforcing, then try to lower the mode to *permissive* as described above.

The BMR program detects this situation for RHEL 8 and automatically lowers the *SELinux* mode to *permissive* while configuring the *BOOT* and warns you in the log displayed just before reboot. In any case, a warning is displayed when *SELinux* mode was set to *enforcing* on the source system to get your attention about this problem.

The problem is known to show up also on Debian 10 but it is not yet handled by the BMR program, this up to you to change the *SELinux* mode in the dialog box, and follow the procedure.

Restoring a system with multipath enabled

LinuxBMR has supported *Multipath* since version 2.3.0. Backup and restore of RHEL 8 & 9 servers can be performed on both *Singlepath* and *Multipath* machines. However, other Linux distributions can only be backed up and restored on *SinglePath* machines. A list of supported Linux distributions will be provided subsequently. .. when?

When restoring on a *Multipath* setup, LinuxBMR does not configure or display any multipath devices; it only presents a single path while concealing the others. Prior to the creation of partitions, logical volumes and filesystems, LinuxBMR **deletes** the *hidden* device aliases to avoid any confusion and allows LVM and associated tools update the boot system. Additionally, LinuxBMR compels the Linux distribution to incorporate the multipath modules when regenerating the *initramfs*. It also updates */etc/multipath/bindings* to ensure consistent naming.

In contrast, when restoring on a *Singlepath* setup, LinuxBMR operates as usual, renaming devices as necessary without enforcing any multipath modules.

In both scenarios, the file */etc/lvm/devices/system.devices* is not restored to its original location, instead, it is renamed with a *.linuxbmr* extension due to containing outdated LVM UUID information that is no longer valid. To recreate the file, use the command: *lvmdevices –update*.

More about the GUI

File Menu

The File menu shown in figure *The File Menu* gives you access to a “Root terminal” and lets you “Exit” the program or “Reboot” the system.

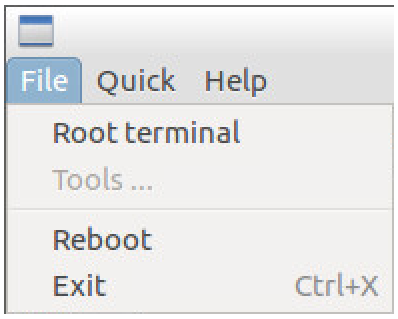


Fig. 179: The File Menu

Quick Menu

The Quick menu in figure *The Quick Menu* can open the log file of the BMR program.

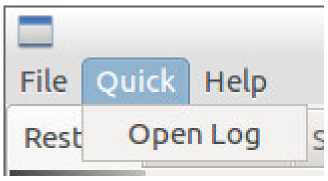


Fig. 180: The Quick Menu

Restore Tab

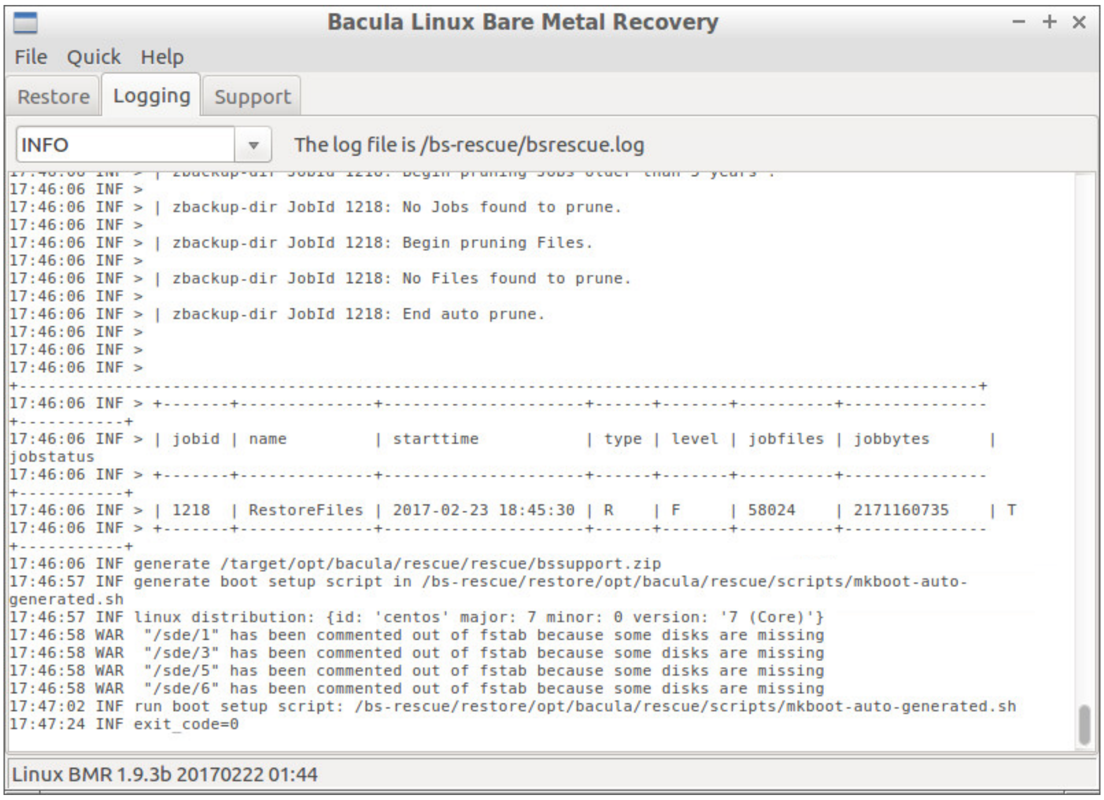
The restore tab has been described in the previous sections.

Logging Tab

Next to the “Restore” tab is the “Logging” tab that displays all log messages of the BMR program. Messages can be filtered by level.

Support Tab

The “Support” tab presented in figure *The Support tab* allows the generation of a report file `bssuport.zip` that contains information required by our support team to assist. The file can be uploaded directly to Bacula Systems’ ticketing system (using the https protocol), or downloaded through HTTP protocol via a built-in server. When uploading, if you already have a ticket open, then supply the ticket number when sending the file to Support. If not, use any identifier that can help the support team to easily identify your report.



The log file is /bs-rescue/bsrescue.log

```
17:46:06 INF > | zbackup-dir JobId 1218: Begin pruning Jobs older than 5 years .
17:46:06 INF >
17:46:06 INF > | zbackup-dir JobId 1218: No Jobs found to prune.
17:46:06 INF >
17:46:06 INF > | zbackup-dir JobId 1218: Begin pruning Files.
17:46:06 INF >
17:46:06 INF > | zbackup-dir JobId 1218: No Files found to prune.
17:46:06 INF >
17:46:06 INF > | zbackup-dir JobId 1218: End auto prune.
17:46:06 INF >
17:46:06 INF >
17:46:06 INF >
+-------------------------------------------------------------------------------------------+
17:46:06 INF > +-------+-------------+-------------------+------+-------+---------+-------------
+-----------+
17:46:06 INF > | jobid | name        | starttime         | type | level | jobfiles | jobbytes     |
jobstatus
17:46:06 INF > +-------+-------------+-------------------+------+-------+---------+-------------
+-----------+
17:46:06 INF > | 1218  | RestoreFiles | 2017-02-23 18:45:30 | R    | F     | 58024    | 2171160735    | T
17:46:06 INF > +-------+-------------+-------------------+------+-------+---------+-------------
+-----------+
17:46:06 INF generate /target/opt/bacula/rescue/rescue/bssupport.zip
17:46:57 INF generate boot setup script in /bs-rescue/restore/opt/bacula/rescue/scripts/mkboot-auto-
generated.sh
17:46:57 INF linux distribution: {id: 'centos' major: 7 minor: 0 version: '7 (Core)'}
17:46:58 WAR  "/sde/1" has been commented out of fstab because some disks are missing
17:46:58 WAR  "/sde/3" has been commented out of fstab because some disks are missing
17:46:58 WAR  "/sde/5" has been commented out of fstab because some disks are missing
17:46:58 WAR  "/sde/6" has been commented out of fstab because some disks are missing
17:47:02 INF run boot setup script: /bs-rescue/restore/opt/bacula/rescue/scripts/mkboot-auto-generated.sh
17:47:24 INF exit_code=0
```

Linux BMR 1.9.3b 20170222 01:44

Fig. 181: The Logging Tab

Connection Troubleshooting

Network problems or wrong values in the form shown in figure *Bacula Configuration*, can cause different errors. Here are some hints to solve them.

If you get the error message in figure **fig:bmr-config-no-dir-connect**, then you have a problem connecting to your **Director**. You can try to `ping` or `telnet` to your **Director**, or review the Director's fields in the configuration form.

If you get the error message shown in the screenshot below, then the **Director** can not reach your client. Try to `ping` or `telnet` to the client from the **Director** itself and compare the rescue client fields in the configuration form with the ones on the Director side.

If you get the error message in figure *Something is Wrong in the Console Configuration*, then the "Rescue client name" or "Rescue console password" do not match the ones used in your Console resource of the **Director**, as shown in figure **fig:cons**.

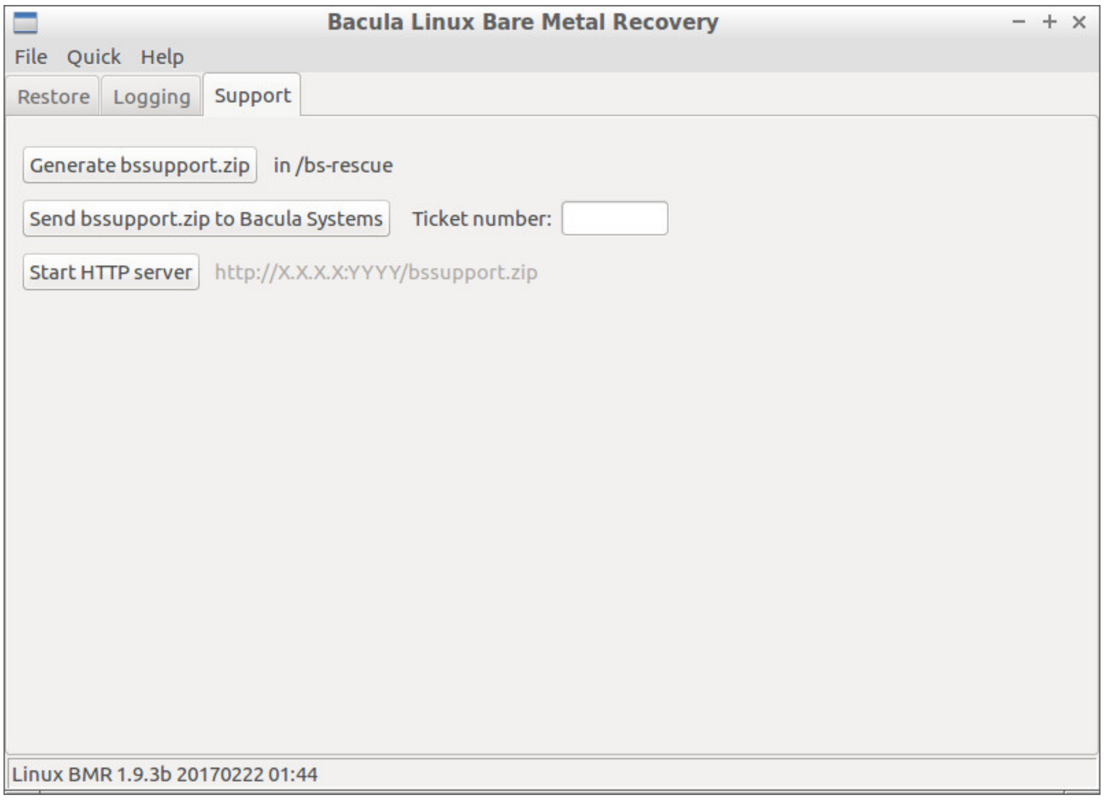
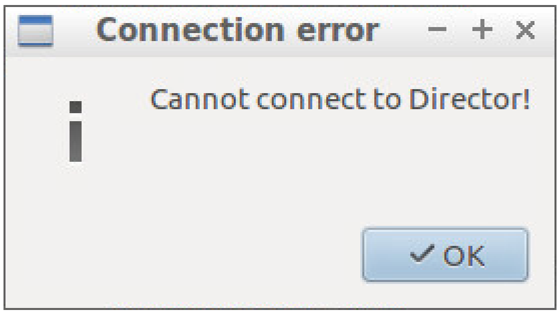
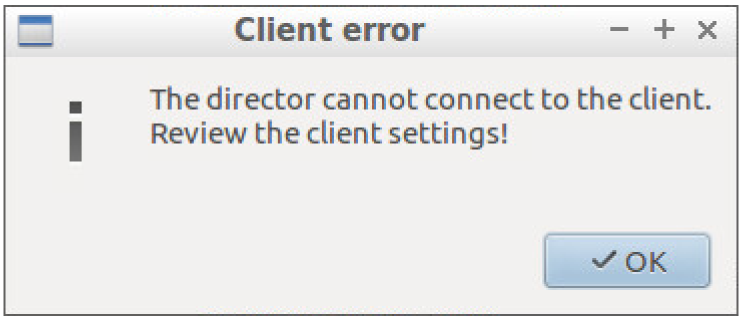

Bacula Linux Bare Metal Recovery
File Quick Help
Restore Logging Support
Generate bssupport.zip
in /bs-rescue
Send bssupport.zip to Bacula Systems
Ticket number:
Start HTTP server
http://X.X.X.X:YYYY/bssupport.zip
Linux BMR 1.9.3b 20170222 01:44

Fig. 182: The Support tab



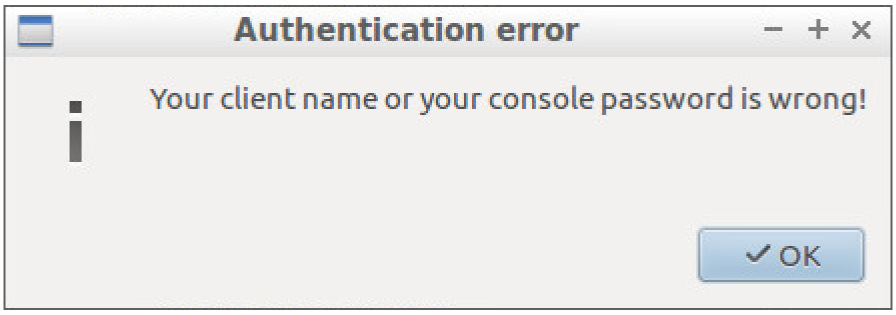
Cannot connect to Director!

Fig. 183: Something is Wrong Connecting to the Director



The director cannot connect to the client.
Review the client settings!

Fig. 184: Something is Wrong in the Client Configuration


Authentication error

Your client name or your console password is wrong!

Fig. 185: Something is Wrong in the Console Configuration

Compatibility

- The LinuxBMR runs with **Bacula Enterprise** versions 6 and newer.
- The LinuxBMR 2.2.4 version is compatible with the Bacula-rescue.sh script version 2.4.0 available from Bacula Enterprise 12.4.2 and in the LinuxBMR Plugin download area.
- The LinuxBMR has been successfully tested on RHEL 5, 6, 7, and 8, SUSE Linux Enterprise Server 11 with Grub Legacy, SUSE Linux Enterprise Server 12 with at least Service Pack 1 (btrfs is not supported), as well as Debian Linux 7, 8, 9, and 10, and Ubuntu LTS 14.04, 16.04, 18.04, and 20.04. Only 64bit versions of these distributions have been tested.
- The LinuxBMR supports `LVM`. LVM's advanced and `RAID` functionalities are not supported. *Multipath* is also not supported.
- `LVM` unallocated space is preserved and handled like an unused *Logical Volume* to preserve snapshot functionality.
- LinuxBMR supports the following filesytems: xfs, ext2, ext3, ext4. MSDOS / FAT are supported for UEFI boot partitions. zfs, btrfs and others are not.
- DOS and GPT partition tables are supported.
- Linux Software `RAID` and `LVM` on RAID will be supported in a future version.
- Partition encryption is not supported (this is planned for a future release).
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.
- Ask our Support Team for up to date information about supported and unsupported features. The most frequently requested features will be prioritized.

Important Considerations

The LinuxBMR is designed to be an extensible framework, not a static one-size-fits-all solution. Unsupported or problematic features can be handled manually if needed. In particular the disk partitioning and boot manager configurations have a manual mode. The scripts generated by the BMR using information from the source system can be used as an example to help the user.

Due to the very large compatibility matrix between hardware and software that LinuxBMR should support, LinuxBMR needs careful testing and configuration on actual production-like systems before it can become part of a reliable production-ready Disaster Recovery solution. Once a combination of hard- and software (for example, DELL R620 Perc H800 with RHEL 7 64bit) is validated, the product can safely be used on all servers that are using that combination.

7.2 Inventory Plugin

Overview

Features Summary

The **Bacula Enterprise inventory** plugin provides a framework that can be used to query components information for each Bacula client. The inventory information can be queried at will from bconsole.

Inventory Hooks

Inventory hooks are installed in `/opt/bacula/etc/inventory.d` and can be queried separately.

Basic

Linux

```
database-mysql.sh
database-postgresql.sh
```

Queries mysql and postgresql databases informations.

Windows

```
databases-mssql.ps1
hyperv-inventory.ps1
```

Queries mssql databases and hyper-V hypervisor information.

Installation

Configuration of the Bacula File Daemon

The `Plugin Directory` directive of the `File Daemon` resource in */opt/bacula/etc/bacula-fd.conf* should point to
the location where the `azure-vm-fd.so` plugin is installed. The default directory is: */opt/bacula/plugins*

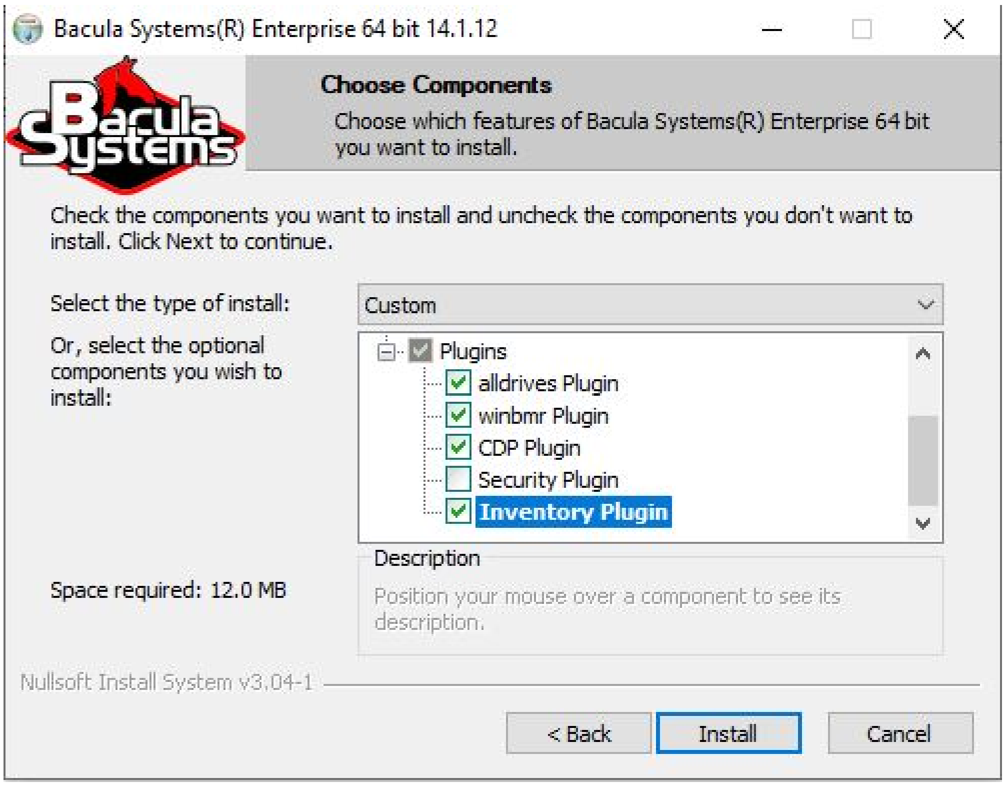
```
FileDaemon {
  Name = bacula-fd
  Plugin Directory = /opt/bacula/plugins
  ...
}
```

Installation of the Plugin

For more information about plugin installation see LinuxInstallFileDaemon.

Windows

The **Bacula Enterprise inventory** plugin is selectable as a component of the **File Daemon** windows installer.


Bacula Systems(R) Enterprise 64 bit 14.1.12

Choose Components
Choose which features of Bacula Systems(R) Enterprise 64 bit you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Custom

Or, select the optional components you wish to install:

- Plugins
  - [✓] alldrives Plugin
  - [✓] winbmr Plugin
  - [✓] CDP Plugin
  - [ ] Security Plugin
  - [✓] Inventory Plugin

Space required: 12.0 MB

Description
Position your mouse over a component to see its description.

Nullsoft Install System v3.04-1

Fig. 186: The inventory plugin in the File Daemon windows installer

Example

From **bconsole**, run the following command:

```
.query parameter=database* client=localhost-fd plugin="inventory:"
```

The output provided by the hook is a JSON object with the following information:

```
{
   "result": [
      {
         "source": "mysql",
         "type": "Database",
         "info": "mysql  Ver 14.14 Distrib 5.7.34, for Linux (x86_64) using ␣
↪EditLine wrapper",
         "version": 1,
         "runscript": [
            {
               "name": "clientrunbeforejob",
               "run": "systemctl stop mysql"
            },
            {
               "name": "clientrunafterjob",
               "run": "systemctl start mysql"
            }
         ],
         "status": 1
      },
      {
         "source": "postgresql",
         "type": "Database",
         "info": "psql (PostgreSQL) 13.9 (Debian 13.9-0+deb11u1)",
         "version": 1,
         "runscript": [
            {
               "name": "clientrunbeforejob",
               "run": "systemctl stop postgresql"
            },
            {
               "name": "clientrunafterjob",
               "run": "systemctl start postgresql"
            }
         ],
         "status": 1
      }
   ],
   "version": "1",
   "request": "*database*",
   "type": "inventory_report",
   "timesec": 1671718067,
   "hostname": "stretch",
   "uptime": 13698,
   "uname": "Linux stretch 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-
```

(continues on next page)

(continued from previous page)

```
↪13) x86_64"
}
```

Table 55: JSON fields

Option	Description
source version info error runscript	(String) Name of the hook (String) Version of the hook program (String) useful information (version, build, etc.) (Int) different from zero to raise an error (Array) suggestions on how to handle the component

Advanced

Hook Protocol Definition

inventory hooks can be written in any language. Some environment variables are passed to all hooks.

Table 56: Environnement variables

Option	Default	Description
BACULA_WORKINGDIR BACULA_SYSCONFDIR BACULA_BINDIR	/opt/bacula/working /opt/bacula/etc /opt/bacula/bin	Bacula Working directory Bacula Configuration directory Bacula Binary directory

Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

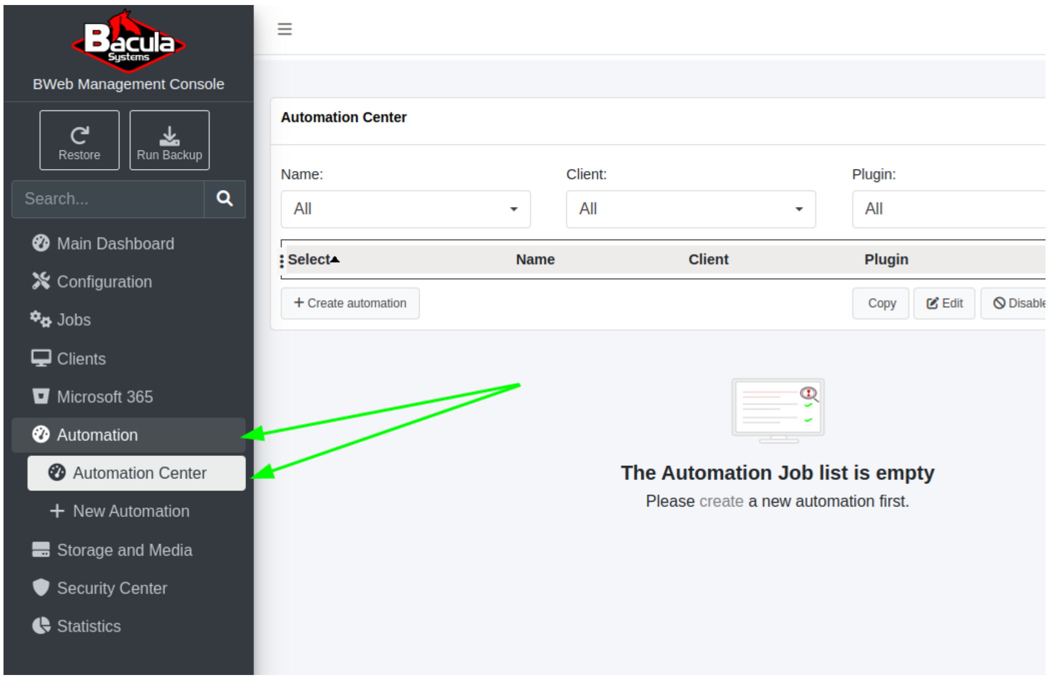
8 Automatic Object Integration (Scan Plugin)

This chapter will explain how to automatically configure Bacula with systems such as VMWare, HyperV or NetApp among others where each object can be backed up separately with different Jobs to maximize the throughput and the resiliency.

It is a best practice to setup a Job per virtual machine, per database or per NAS volume, however, it becomes difficult to adjust the Bacula configuration each time an object is added or deleted from the system. The Scan Plugin can query the plugin and create or delete the corresponding Bacula configuration automatically. A Job and a corresponding Fileset will be created for each object detected by the Plugin.

With some simple rules, Bacula can then detect and adjust it's configuration after each change done on the system, like a virtual machine, a database or a new volume added or deleted.

Important: **BWeb Automation Center** is recommended for use in place of manual configuration of the Scan plugin.



8.1 Installation

The **Scan Plugin** is part of the BWeb Management Center package `bacula-enterprise-bweb` that can be installed as a Director plugin, via BIM. The Bacula configuration must be managed by BWeb.

8.2 Usage

The Scan Plugin can be executed on demand or scheduled in an Admin Job.

An example of a Admin job:

```
Job {
  Name = A_db-sd_mysql
  Type = Admin
  JobDefs = BackupsToDisk
  RunBeforeJob = "/opt/bacula/bin/scan_plugin --jobdefs BackupsToDisk --
↪client db-fd --plugin mysql --plugin_option abort_on_error --commit_and_
↪reload"
}
```

8.3 Options

Table 57: Scan Plugin Options

Options	Required	Default	Info	Example
client <val>	Yes		Bacula Director Client resource name	`--client 127.0.0.1-fd`
parameter <val>	No		Plugin object parameter name.	`--parameter vm`
plugin <val>	Yes		Bacula Plugin name	`--plugin vsphere`
plugin_option <val>…	Yes		Plugin options	`--plugin_option use_sudo`
uuid <val>	No	client:plugi	UUID of the execution	`--uuid 03Mars23`
exclude <val>…	No		Pattern for object exclusion	`--exclude "test.*"`
include <val>…	No		Pattern for object inclusion	`--include "test.*"`
object-exclude <val>…	No		String for object exclusion	`--object-exclude "test"`
object-include <val>…	No		String for object inclusion	`--object-include "test"`
commit_and_reload	No	No	Apply the new configuration to the Director	`--commit_and_reload`
directive <val>…	No		Bacula Director Job directive	`--directive Priority=10`
fileset <val>	No	F_%c_%p_	Bacula Director Fileset name template	`--fileset database-%v`
fs_option <val>…	No		Bacula Director Fileset Option directive	`--fs_option signature=md5`
hook <val>	No		Execute a script for each new/old object	`--hook /scripts/editfstab`
job <val>	No	J_%c_%p_'	Bacula Director Job name template	`--job database-%v`
jobdefs <val>	Yes		Bacula Director JobDefs resource name	`--jobdefs BackupsToDisk`
remove_jobs	No	No	Remove resources that are no longer necessary	`--remove_jobs`
scan_option <val>…	No		Use a specific set of plugin option for the scan	`--scan_option user=root`
use_uuid	No	No	Use object UUID in the Fileset Plugin command	`--use_uuid`

Options in the table with a **...** can be used multiple times on the command line. Other options for the format output and the command line parsing are available with the `--help` option.

The **parameter** option shouldn't be required. If it is not the case, contact Bacula Systems Support.

The plugin_option parameter can be used as many times as you wish, with all the options you should have configured for the plugin specified in the plugin parameter. Check the specific documentation for the plugin to correctly setup the plugin_option values. They should be similar to the ones you use in the plugin line in the Fileset resource for the plugin.

The JobDefs is used to ease the configuration of the new Jobs created by the **Scan Plugin**.

```
JobDefs {
   Name = BackupsToDisk
   Type = Backup
   Pool = DiskBackup365
   Level = Incremental
   Messages = Default
   Storage = DiskAutochanger
   MaximumConcurrentJobs = 5
}
```

Specific options can be added with the `--directive** parameter`, e.g.:

```
--directive Priority=10 --directive MaximumRunTime=5h
```

By default, when an object is no longer available (like if a virtual machine is deleted), the Scan Plugin will disable the corresponding Job. With the `--remove_job` option, the Job and the Fileset can be deleted from the configuration.

The `--job` and `--fileset` parameters can use the following shortcuts:

- %c : Client name
- %p : Plugin name
- %v : Object name

Example:

```
--job %c-%p-%v --fileset fileset-%p-%v
```

The Scan Plugin will automatically adjust the name if the result string is too long or already exists.

8.4 Examples

The following example are executed as the **bacula** user. If the tool is executed as root, the permission on new files might not be correct. To switch to the **bacula** user, you can use `sudo -u bacula bash`

Many Plugins have the **abort_on_error** option that is advised with the Scan Plugin.

See the examples for specific plugins below:

MySQL Example

In the example presented below, the Scan Plugin will be used in an Admin Job and handle connect a mysql databases via the MySQL Plugin installed on the Bacula Client db-fd. Each database detected will be handled by a separated Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
Job {
  Name = A_db-sd_mysql
  Type = Admin
  JobDefs = BackupsToDisk
  RunsBeforeJob = "/opt/bacula/bin/scan_plugin --jobdefs BackupsToDisk --
↪client db-fd --plugin mysql --plugin_option abort_on_error --commit_and_
```

(continues on next page)

(continued from previous page)

```
↪reload"
}
```

NDMP Example

In the example presented below, the Scan Plugin will connect a NDMP server via the NDMP Plugin installed on the Bacula Client nas-fd. Each volume detected will be handled by a separate Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
bacula$ /opt/bacula/bin/scan_plugin --client nas-fd --plugin ndmp --jobdefs␣
↪BackupToDisks --commit_and_reload --plugin_option user=root --plugin_option␣
↪password=pwd --plugin_option host=netapp.lan --plugin_option abort_on_error

Report for scan_plugin 0.5 nas-fd:ndmp
----------------------------------------------
Client:         nas-fd
Plugin:         ndmp
Total Objects:  2
---------------------------------------------
 ~ updated, + added, X removed, D disabled
---------------------------------------------
 + J_nas-fd_ndmp_vol0
 + J_nas-fd_ndmp__dev_vol1

----------------------------------------------
Status: Configuration applied and reloaded
```

PostgreSQL Example

In the example presented below, the Scan Plugin will connect the local PostgreSQL database via the PostgreSQL Plugin installed on the Bacula Client www-fd. Each database detected will be handled by a separate Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
bacula$ /opt/bacula/bin/scan_plugin --client www-fd --plugin postgresql --
↪jobdefs BackupsToDisk --plugin_option abort_on_error

Report for scan_plugin 0.5 www-fd:postgresql
----------------------------------------------
Client:         www-fd
Plugin:         postgresql
Total Objects:  11
---------------------------------------------
 ~ updated, + added, X removed, D disabled
---------------------------------------------
 + J_www-fd_postgresql_other
 + J_www-fd_postgresql_weird database81225
 + J_www-fd_postgresql_regress
 + J_www-fd_postgresql_dev
 + J_www-fd_postgresql_template1
 + J_www-fd_postgresql_ssss1
```

(continues on next page)

(continued from previous page)

```
 + J_www-fd_postgresql_git
 + J_www-fd_postgresql_postgres
 + J_www-fd_postgresql_bacula
 + J_www-fd_postgresql_db1437565
 + J_www-fd_postgresql_master

----------------------------------------------
Status: Configuration not applied
```

The following configuration will be created:

```
Job {
  Name = "J_www-fd_postgresql_master"
  Description = "{\"parameter\":\"database\",\"id\":\"www-fd:postgresql\",\
↪"version\":\"0.5\",\"uuid\":\"uuid=www-fd:postgresql:master\",\"database\":\
↪"master\",\"plugin\":\"postgresql\",\"description\":\"Generated by scan_
↪plugin. Do not edit.\"}"
  Type = "Backup"
  Client = "www-fd"
  Fileset = "J_www-fd_postgresql_master"
  JobDefs = "BackupsToDisk"
}

Fileset {
  Name = "F_www-fd_postgresql_master"
  Description = "{\"parameter\":\"database\",\"version\":\"0.5\",\"id\":\"www-
↪fd:postgresql\",\"uuid\":\"uuid=www-fd:postgresql:master\",\"description\":\
↪"Generated by scan_plugin. Do not edit.\",\"plugin\":\"postgresql\",\
↪"database\":\":master\"}"
  Include {
    Options {
    }
    Plugin = "postgresql: database=\"master\" abort_on_error"
  }
}
```

M365 Example

In the example presented below, the Scan Plugin will be used in an Admin Job and connect via the Microsoft 365 Plugin installed on the Bacula Client saas-fd. Each user drive detected will be handled by a separated Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
Job {
  Name = A_m365_drive
  Type = Admin
  client = saas-fd
  JobDefs = BackupsToDisk
  RunsBeforeJob = "/opt/bacula/bin/scan_plugin --client %c --plugin m365 --
↪plugin_option \"config_file=/opt/bacula/etc/m365/TENANT/bacula_m365_config.
↪conf\" --plugin_option \"service=drive\" --scan_option 'config_file=/opt/
```

(continues on next page)

(continued from previous page)

```
↪bacula/etc/m365/TENANT/bacula_m365_config.conf' --jobdefs BackupsToDisk --
↪parameter user"
```

In the following example, the Scan Plugin will be used in an Admin Job and connect via the Microsoft 365 Plugin installed on the Bacula Client saas-fd. Each user e-mail detected will be handled by a separated Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
Job {
  Name = A_m365_email
  Type = Admin
  client = saas-fd
  JobDefs = BackupsToDisk
  RunsBeforeJob = "/opt/bacula/bin/scan_plugin --client \"%c\" --plugin m365 -
↪-plugin_option \"config_file=/opt/bacula/etc/m365/TENANT/bacula_m365_config.
↪conf\" --plugin_option \"service=email\" --scan_option \"config_file=/opt/
↪bacula/etc/m365/TENANT/bacula_m365_config.conf\" --jobdefs BackupsToDisk --
↪parameter user"
```

In the following example, the Scan Plugin will be used in an Admin Job and connect via the Microsoft 365 Plugin installed on the Bacula Client saas-fd. Each user sharepoint site detected will be handled by a separated Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

```
Job {
  Name = A_m365_sharepoint
  Type = Admin
  client = saas-fd
  JobDefs = BackupsToDisk
  RunsBeforeJob = "/opt/bacula/bin/scan_plugin --client \"%c\" --plugin m365 -
↪-plugin_option \"config_file=/opt/bacula/etc/m365/TENANT/bacula_m365_config.
↪conf\" --plugin_option \"service=sharepoint\" --scan_option \"config_file=/
↪opt/bacula/etc/m365/TENANT/bacula_m365_config.conf\" --jobdefs␣
↪BackupsToDisk --parameter site"
```

Quobyte Example

In the example presented below, the Scan Plugin will be used in an Admin Job and connect via the Quobyte Plugin.

```
JobId 1193: shell command: run BeforeJob "/opt/bacula/bin/scan_
↪plugin --json_conf_file "/opt/bacula/etc/conf.d/scan_plugin/wa-
↪quobyte-dir-fd_quobyte_connector.json""
JobId 1193: BeforeJob:
JobId 1193: BeforeJob: Report for scan_plugin 0.6 wa-quobyte-dir-fd-
↪quobyte-1710407060
JobId 1193: BeforeJob: ----------------------------------------------
JobId 1193: BeforeJob: Client:         wa-quobyte-dir-fd
JobId 1193: BeforeJob: Plugin:         quobyte
JobId 1193: BeforeJob: Total Objects:  3
JobId 1193: BeforeJob: ---------------------------------------------
JobId 1193: BeforeJob:  ~ updated, + added, X removed, D disabled
JobId 1193: BeforeJob: ---------------------------------------------
```

(continues on next page)

(continued from previous page)

```
JobId 1193: BeforeJob:  + J_wa-quobyte-dir-fd_quobyte__quobyte_waa_
↪vol3
JobId 1193: BeforeJob:  + J_wa-quobyte-dir-fd_quobyte__quobyte_waa_
↪vol2
JobId 1193: BeforeJob:  + J_wa-quobyte-dir-fd_quobyte__quobyte_waa_
↪vol1
JobId 1193: BeforeJob:
JobId 1193: BeforeJob: ----------------------------------------------
JobId 1193: BeforeJob: Status: Configuration applied and reloaded
JobId 1193: Start Admin JobId 1193, Job=A_wa-quobyte-dir-fd_quobyte.
↪2024-03-14_05.04.20_21
2024-03-14 05:04:24 q-dir JobId 1193: Bacula 18.0.2 (05Mar24): 14-
↪Mar-2024 05:04:24
 JobId:                  1193
 Job:                    A_wa-quobyte-dir-fd_quobyte.2024-03-14_05.
↪04.20_21
 Scheduled time:         14-Mar-2024 05:04:20
 Start time:             14-Mar-2024 05:04:24
 End time:               14-Mar-2024 05:04:24
 Termination:            Admin OK
```

See the **Scan Plugin** command line example below:

```
root# sudo -u bacula /opt/bacula/bin/scan_plugin --plugin quobyte --
↪jobdefs BackupsToDisk --uuid wa-quobyte-dir-fd-quobyte-1710407060 -
↪-client wa-quobyte-dir-fd --commit_and_reload 1 --fs_option␣
↪Signature=md5

Report for scan_plugin 0.6 wa-quobyte-dir-fd-quobyte-1710407060
----------------------------------------------
Client:         wa-quobyte-dir-fd
Plugin:         quobyte
Total Objects:  3
---------------------------------------------
 ~ updated, + added, X removed, D disabled
---------------------------------------------
 ~ J_wa-quobyte-dir-fd_quobyte__quobyte_waa_vol1
 ~ J_wa-quobyte-dir-fd_quobyte__quobyte_waa_vol2
 ~ J_wa-quobyte-dir-fd_quobyte__quobyte_waa_vol3

----------------------------------------------
Status: Configuration applied and reloaded
```

The resulting Jobs and Filesets will be created with the Quobyte Plugin. The Snapshot directive will be automatically forced to `yes` when the Quobyte Plugin is used and is not required to be set in the Fileset.

Proxmox Example

In the example presented below, the Scan Plugin will contact the Proxmox server via the Proxmox Plugin installed on the Bacula Client proxmox03-fd. Each virtual machine detected who's name ends with '-tmpl' will have a Job and corresponding Fileset created. The Job directives will be defined via the JobDefs **BackupsToDisk**.

The Proxmox Plugin must be correctly configured on the Bacula Client to be used by the **scan_plugin** script. Refer to the Proxmox configuration chapter to install and configure the Proxmox Plugin.

Using the bconsole **.ls** command, list the VMs hosted by the 'proxmox03-fd' Client ending in '-tmpl'.

```
bacula$ echo ".ls client=proxmox03-fd plugin=proxmox: path=vm" | bconsole |␣
↪grep ".*-tmpl"
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:31 ␣
↪centos7-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:33 ␣
↪centos8-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:37 ␣
↪debian12-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:40 ␣
↪debian11-tmpl
-rw-r-----   1 root     root           53687091200 2023-07-26 20:49:42 ␣
↪debian10-tmpl
-rw-r-----   1 root     root           32212254720 2023-07-26 20:49:47 ␣
↪debian9-tmpl
-rw-r-----   1 root     root           10737418240 2023-07-26 20:49:49 ␣
↪oraclelinux7-tmpl
-rw-r-----   1 root     root           10737418240 2023-07-26 20:49:52 ␣
↪oraclelinux8-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:54 ␣
↪oraclelinux9-tmpl
-rw-r-----   1 root     root           34359738368 2023-07-26 20:49:56 ␣
↪rhel8-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:49:58 ␣
↪rhel9-tmpl
-rw-r-----   1 root     root           10737418240 2023-07-26 20:50:06 ␣
↪sles125-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:50:09 ␣
↪sles154-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:50:26 ␣
↪ubuntu18-tmpl
-rw-r-----   1 root     root           26843545600 2023-07-26 20:50:37 ␣
↪ubuntu20-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:50:43 ␣
↪ubuntu22-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:50:46 ␣
↪rocky8-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:50:58 ␣
↪rocky9-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:51:45 ␣
↪almalinux8-tmpl
-rw-r-----   1 root     root           21474836480 2023-07-26 20:52:15 ␣
↪almalinux9-tmpl
```

Using the **–include** command line option for the **scan_plugin** script, create Jobs and Filesets only for VMs with names ending in *-tmpl*:

```
bacula$ /opt/bacula/bin/scan_plugin --client proxmox03-fd --plugin proxmox --
↪parameter vm --jobdefs BackupsToDisk --job %c_%v --fileset %c_%v --commit_
↪and_reload --include ".*-tmpl"

Report for scan_plugin 0.5 proxmox03-fd:proxmox
---------------------------------------------
Client:         proxmox03-fd
Plugin:         proxmox
Total Objects:  20
--------------------------------------------
~ updated, + added, X removed, D disabled
--------------------------------------------
+ proxmox03-fd_sles125-tmpl
+ proxmox03-fd_rocky8-tmpl
+ proxmox03-fd_rhel8-tmpl
+ proxmox03-fd_centos7-tmpl
+ proxmox03-fd_debian12-tmpl
+ proxmox03-fd_oraclelinux9-tmpl
+ proxmox03-fd_oraclelinux7-tmpl
+ proxmox03-fd_ubuntu20-tmpl
+ proxmox03-fd_debian10-tmpl
+ proxmox03-fd_almalinux9-tmpl
+ proxmox03-fd_ubuntu18-tmpl
+ proxmox03-fd_sles154-tmpl
+ proxmox03-fd_almalinux8-tmpl
+ proxmox03-fd_centos8-tmpl
+ proxmox03-fd_rocky9-tmpl
+ proxmox03-fd_oraclelinux8-tmpl
+ proxmox03-fd_debian11-tmpl
+ proxmox03-fd_ubuntu22-tmpl
+ proxmox03-fd_debian9-tmpl
+ proxmox03-fd_rhel9-tmpl

---------------------------------------------
Status: Configuration applied and reloaded
```

vSphere Example

In the example presented below, the Scan Plugin will contact the vCenter server via the vSphere Plugin installed on the Bacula Client vm-fd. Each virtual machine detected will be handled by a separate Job and Fileset. The Job directives will be defined via the JobDefs **BackupsToDisk**.

The vSphere Plugin must be correctly configured on the Bacula Client to be used by the Scan Plugin. Refer to the vSphere configuration chapter to configure **/opt/bacula/etc/vsphere_global.conf**.

Use the **vsphere-ctl** script to verify that the VMs can be listed managed by the vCenter server:

```
root@vm# /opt/bacula/bin/vsphere-ctl update
1: bacula
```

(continues on next page)

(continued from previous page)

```
2: www
3: mail
```

Example of a scan_plugin command:

```
bacula$ /opt/bacula/bin/scan_plugin --client vm-fd --plugin vsphere --jobdefs␣
↪BackupsToDisk --job J_%p_%v --commit_and_reload --plugin_option abort_on_
↪error

Report for scan_plugin 0.5 vm
----------------------------------------------
Client:         vm-fd
Plugin:         vsphere
Total Objects:  5
---------------------------------------------
 ~ updated, + added, X removed, D disabled
---------------------------------------------
 + J_vsphere_bacula
 + J_vsphere_www
 + J_vsphere_mail

----------------------------------------------
Status: Configuration applied and reloaded
```

OpenStack Example

Using the scan_plugin via Command Line to Setup OpenStack Backups

In the example presented below, the Scan Plugin will contact the OpenStack server via the OpenStack Plugin installed on the `bacula-proxy-vm-rhel9-fd`. For each instance detected in the OpenStack server, a separate Job and Fileset will be created. Also, it will use the JobDefs **BackupsToDisk** in the new Jobs created. By using the `--exclude bacula-proxy-vm-rhel9` option, a backup job for the bacula-proxy-vm-rhel9 will not be created.

```
# sudo -u bacula /opt/bacula/bin/scan_plugin --jobdefs BackupsToDisk --client␣
↪bacula-proxy-vm-rhel9-fd --plugin openstack --plugin_option proxy_
↪server=bacula-proxy-vm-rhel9 --scan_option proxy_server=bacula-proxy-vm-
↪rhel9 --exclude bacula-proxy-vm-rhel9 --commit_and_reload

Report for scan_plugin 0.8 bacula-proxy-vm-rhel9-fd:openstack
----------------------------------------------
Client:         bacula-proxy-vm-rhel9-fd
Plugin:         openstack
Objects Added:  2
---------------------------------------------
 ~ updated, = kept, + added, X removed, D disabled
---------------------------------------------
 + J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2
 + J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1

```

(continues on next page)

(continued from previous page)

```
----------------------------------------------
Status: Configuration applied and reloaded
```

The following configuration will be created:

```
# cat /opt/bacula/etc/conf.d/Director/am-r9-bee-bweb-openstack-tst-dir/Job/J_
↪bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1.cfg
Job {
  Name = "J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1"
  Description = "{\"description\":\"Generated by scan_plugin. Do not edit.\",\
↪"id\":\"bacula-proxy-vm-rhel9-fd:openstack\",\"parameter\":\"server\",\
↪"plugin\":\"openstack\",\"server\":\"cirros-instance1\",\"uuid\":\
↪"uuid=c37c9c5d-4296-41fc-9fc0-b8047ba2737b\",\"version\":\"0.8\"}"
  Type = "Backup"
  Client = "bacula-proxy-vm-rhel9-fd"
  Fileset = "F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1"
  JobDefs = "BackupsToDisk"
}

# cat /opt/bacula/etc/conf.d/Director/am-r9-bee-bweb-openstack-tst-dir/
↪Fileset/F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1.cfg
Fileset {
  Name = "F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1"
  Description = "{\"description\":\"Generated by scan_plugin. Do not edit.\",\
↪"id\":\"bacula-proxy-vm-rhel9-fd-openstack-1746790567\",\"parameter\":\
↪"server\",\"plugin\":\"openstack\",\"server\":\"cirros-instance1\",\"uuid\
↪":\"uuid=c37c9c5d-4296-41fc-9fc0-b8047ba2737b\",\"version\":\"0.8\"}"
  Include {
   Options {
   }
   Plugin = "openstack: server=\"cirros-instance1\"  neutron_network_backup=\
↪"true\" proxy_server=\"bacula-proxy-vm-rhel9\""
  }
}

# cat /opt/bacula/etc/conf.d/Director/am-r9-bee-bweb-openstack-tst-dir/Job/J_
↪bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2.cfg
Job {
  Name = "J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2"
  Description = "{\"description\":\"Generated by scan_plugin. Do not edit.\",\
↪"id\":\"bacula-proxy-vm-rhel9-fd:openstack\",\"parameter\":\"server\",\
↪"plugin\":\"openstack\",\"server\":\"cirros-instance2\",\"uuid\":\
↪"uuid=10440235-d07a-442b-acad-9e68bcdf3752\",\"version\":\"0.8\"}"
  Type = "Backup"
  Client = "bacula-proxy-vm-rhel9-fd"
  Fileset = "F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2"
  JobDefs = "BackupsToDisk"
}

# cat /opt/bacula/etc/conf.d/Director/am-r9-bee-bweb-openstack-tst-dir/
↪Fileset/F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2.cfg
Fileset {
```

(continues on next page)

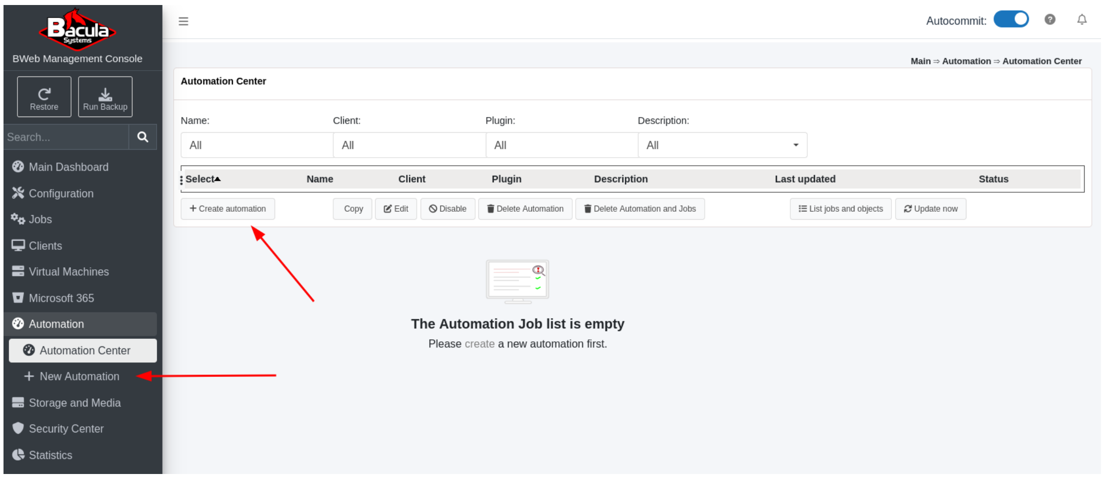
(continued from previous page)

```
  Name = "F_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2"
  Description = "{\"description\":\"Generated by scan_plugin. Do not edit.\",\
↪"id\":\"bacula-proxy-vm-rhel9-fd-openstack-1746790567\",\"parameter\":\
↪"server\",\"plugin\":\"openstack\",\"server\":\"cirros-instance2\",\"uuid\
↪":\"uuid=10440235-d07a-442b-acad-9e68bcdf3752\",\"version\":\"0.8\"}"
  Include {
   Options {
   }
   Plugin = "openstack: server=\"cirros-instance2\"  neutron_network_backup=\
↪"true\" proxy_server=\"bacula-proxy-vm-rhel9\""
  }
}
```

Using BWeb Automation to Setup OpenStack Backups

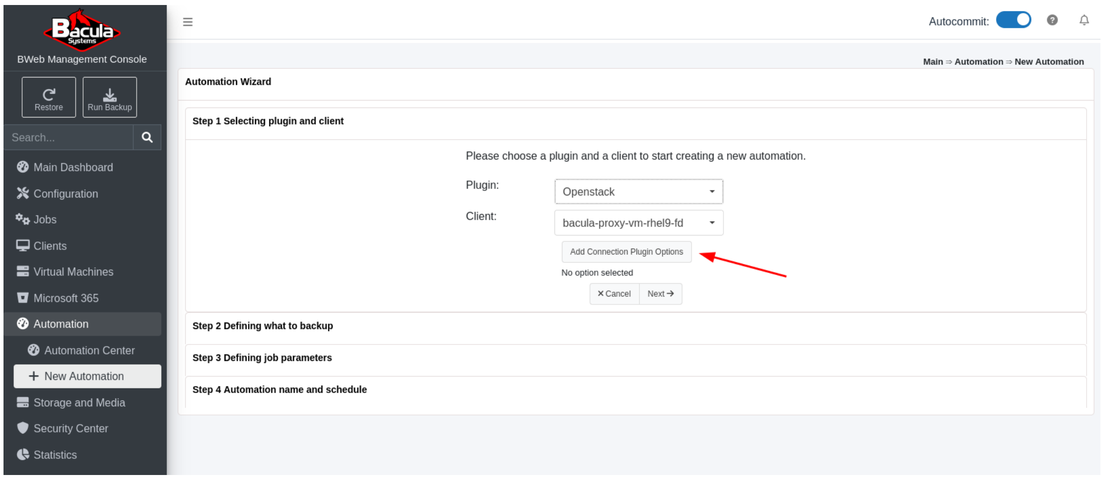
Using BWeb, it is possible to setup an Automation Job to run regularly, and add or remove Jobs from the instances found in the OpenStack server.

1) Click in either the `Create automation` button or the `New Automation` left menu.



The Automation Job list is empty
Please create a new automation first.

2) Select the OpenStack plugin and the Bacula proxy VM Client used to run the OpenStack backups, and choose "Add Connection Plugin Options" to setup connection parameters.



Automation Wizard

Step 1 Selecting plugin and client

Please choose a plugin and a client to start creating a new automation.

Plugin: Openstack

Client: bacula-proxy-vm-rhel9-fd

Add Connection Plugin Options

No option selected

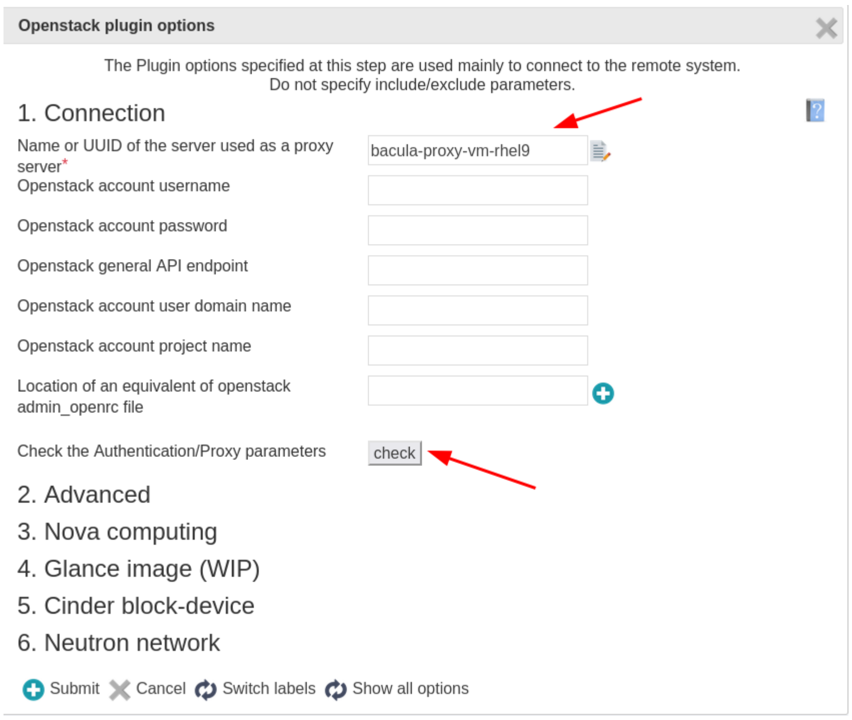
✕Cancel Next →

Step 2 Defining what to backup

Step 3 Defining job parameters

Step 4 Automation name and schedule

3) Enter the name or UUID of the Bacula proxy VM in the OpenStack server, and the check button to confirm the connection is successful. After the check button has reported a successful connection, submit the settings.



Openstack plugin options

The Plugin options specified at this step are used mainly to connect to the remote system.
Do not specify include/exclude parameters.

1. Connection

Name or UUID of the server used as a proxy server*	bacula-proxy-vm-rhel9
Openstack account username	
Openstack account password	
Openstack general API endpoint	
Openstack account user domain name	
Openstack account project name	
Location of an equivalent of openstack admin_openrc file	
Check the Authentication/Proxy parameters	check

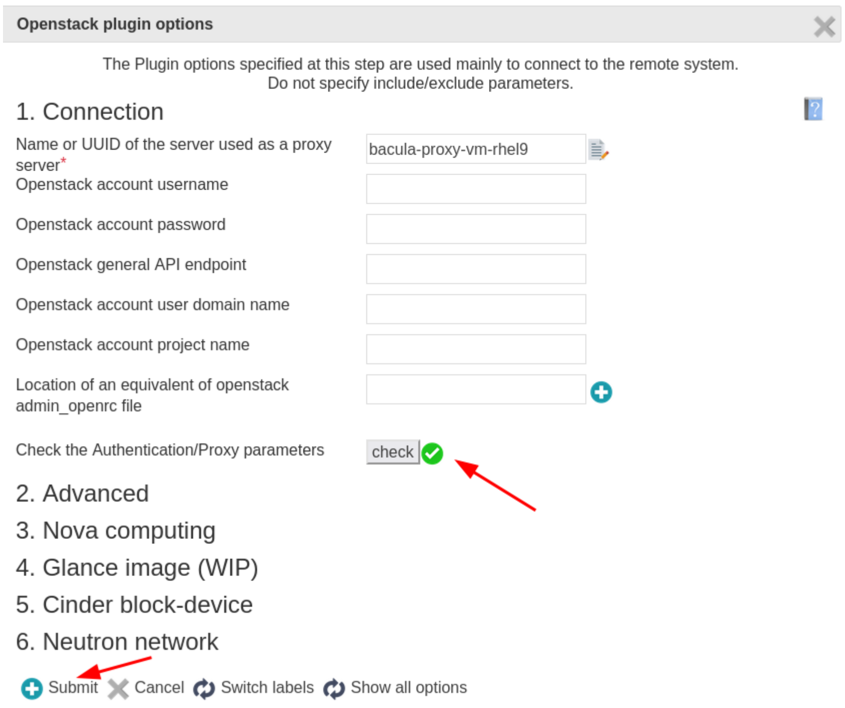
2. Advanced

3. Nova computing

4. Glance image (WIP)

5. Cinder block-device

6. Neutron network


Openstack plugin options

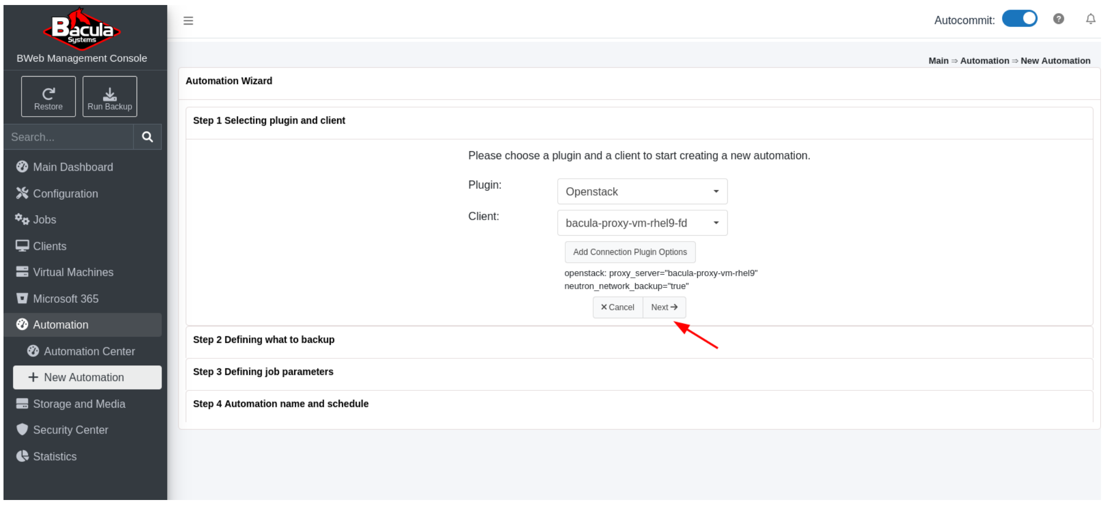
The Plugin options specified at this step are used mainly to connect to the remote system.
Do not specify include/exclude parameters.

1. Connection

Name or UUID of the server used as a proxy server*	bacula-proxy-vm-rhel9
Openstack account username	
Openstack account password	
Openstack general API endpoint	
Openstack account user domain name	
Openstack account project name	
Location of an equivalent of openstack admin_openrc file	
Check the Authentication/Proxy parameters	check ✓

2. Advanced
3. Nova computing
4. Glance image (WIP)
5. Cinder block-device
6. Neutron network

4) An example of how the OpenStack Plugin line should look like in the Fileset, and click in Next.



Automation Wizard

Step 1 Selecting plugin and client

Please choose a plugin and a client to start creating a new automation.

Plugin: Openstack

Client: bacula-proxy-vm-rhel9-fd

```
openstack: proxy_server="bacula-proxy-vm-rhel9"
neutron_network_backup="true"
```

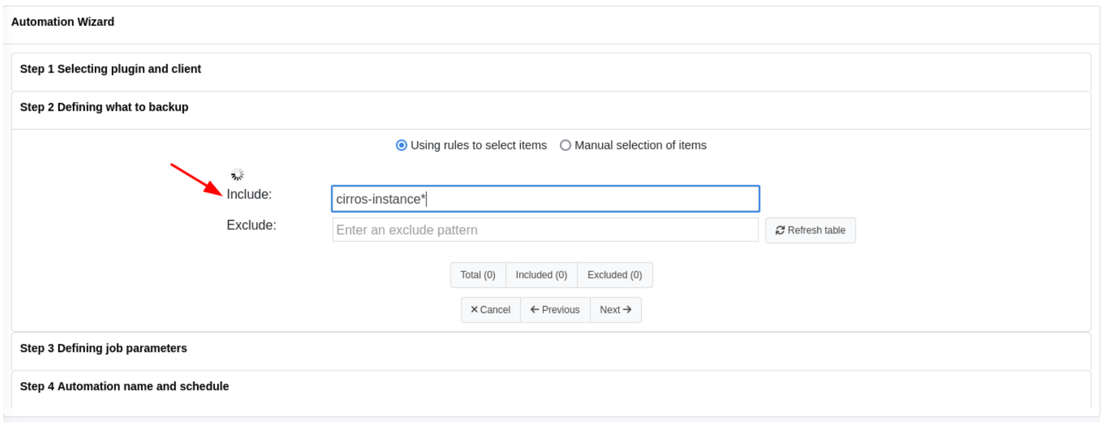
Step 2 Defining what to backup

Step 3 Defining job parameters

Step 4 Automation name and schedule

5) In Step 2, Defining what to backup, there are a few options:

a) using a regular expression to Include a set of instances


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

(● Using rules to select items) ○ Manual selection of items

Include: cirros-instance*

Exclude: Enter an exclude pattern

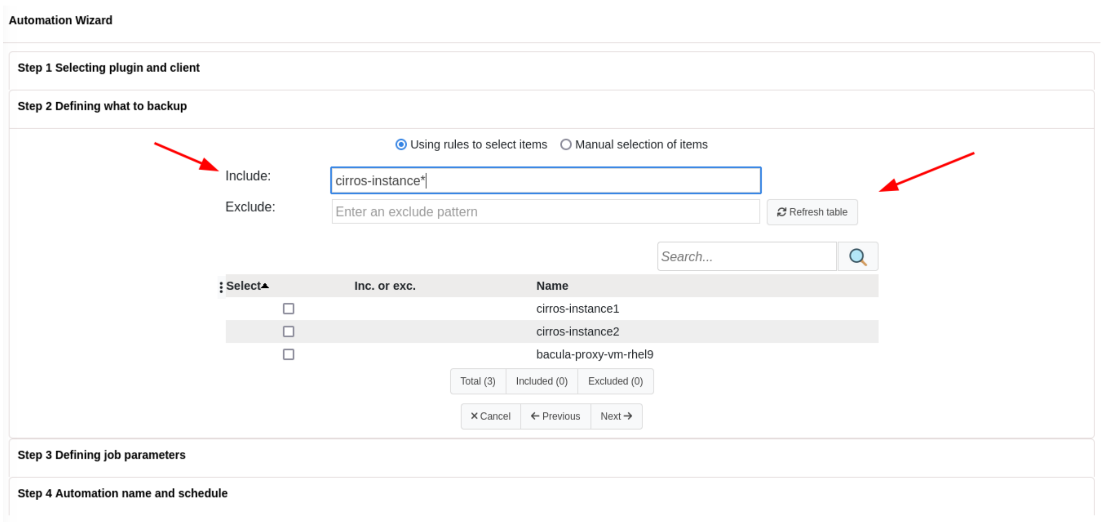
Refresh table

Total (0) | Included (0) | Excluded (0)

Cancel | Previous | Next

Step 3 Defining job parameters

Step 4 Automation name and schedule


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

(● Using rules to select items) ○ Manual selection of items

Include: cirros-instance*

Exclude: Enter an exclude pattern

Refresh table

Search...

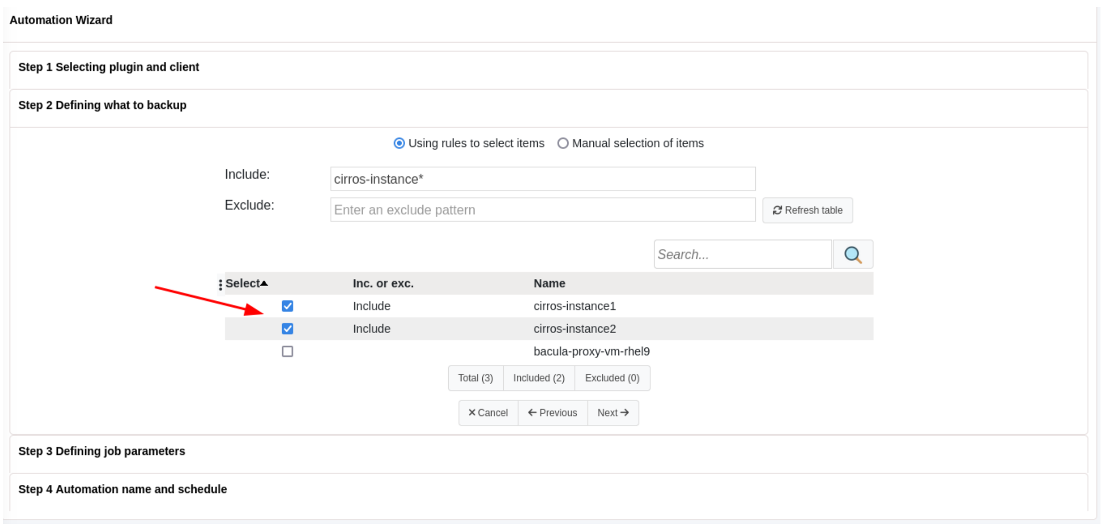
Select	Inc. or exc.	Name
☐		cirros-instance1
☐		cirros-instance2
☐		bacula-proxy-vm-rhel9

Total (3) | Included (0) | Excluded (0)

Cancel | Previous | Next

Step 3 Defining job parameters

Step 4 Automation name and schedule


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

(● Using rules to select items) ○ Manual selection of items

Include: cirros-instance*

Exclude: Enter an exclude pattern

Refresh table

Search...

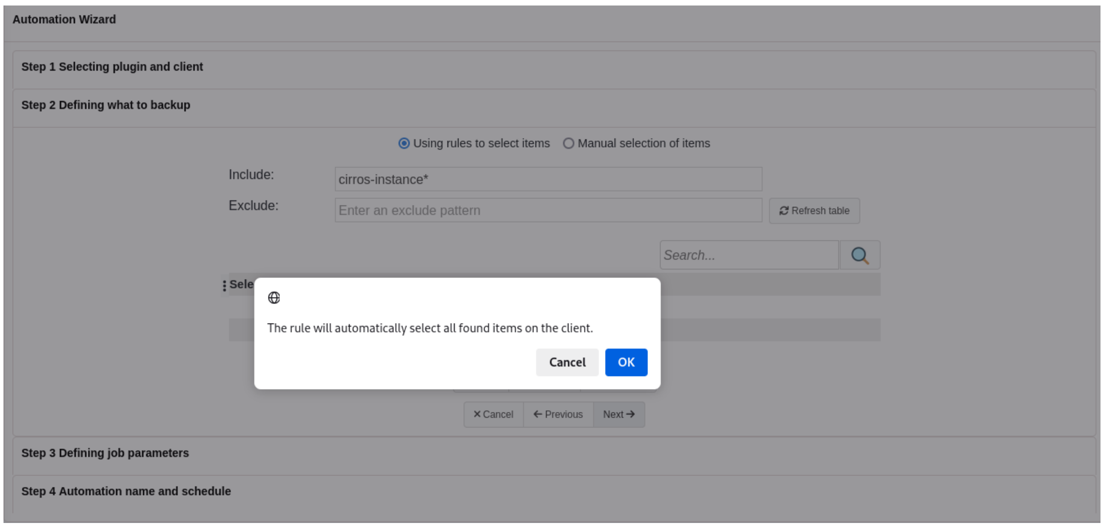
Select	Inc. or exc.	Name
☑	Include	cirros-instance1
☑	Include	cirros-instance2
☐		bacula-proxy-vm-rhel9

Total (3) | Included (2) | Excluded (0)

Cancel | Previous | Next

Step 3 Defining job parameters

Step 4 Automation name and schedule



Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

[✓ Using rules to select items] Manual selection of items

Include:	cirros-instance*
Exclude:	Enter an exclude pattern

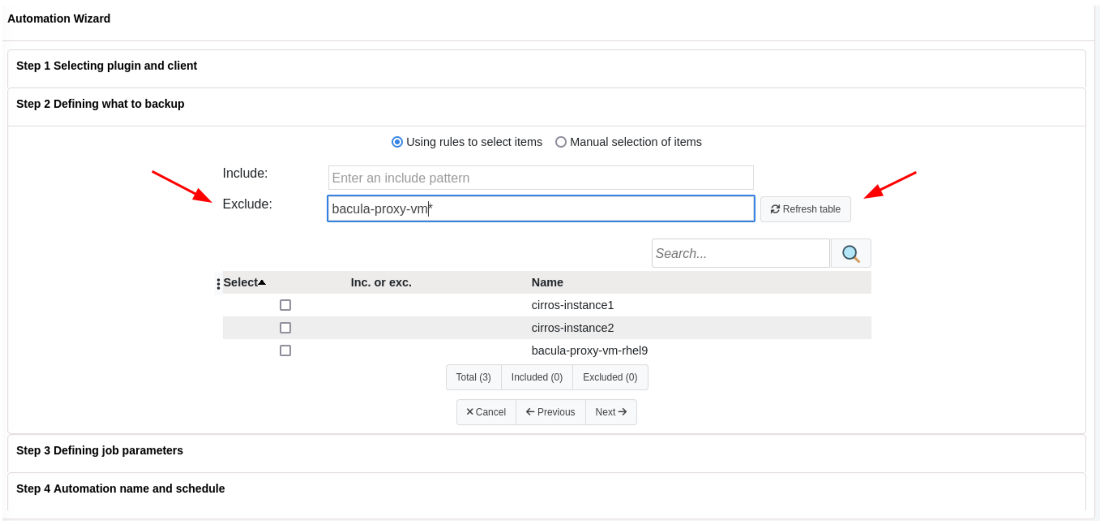
The rule will automatically select all found items on the client.

Cancel OK

Step 3 Defining job parameters

Step 4 Automation name and schedule

b) using a regular expression to Exclude a set of instances



Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

[✓ Using rules to select items] Manual selection of items

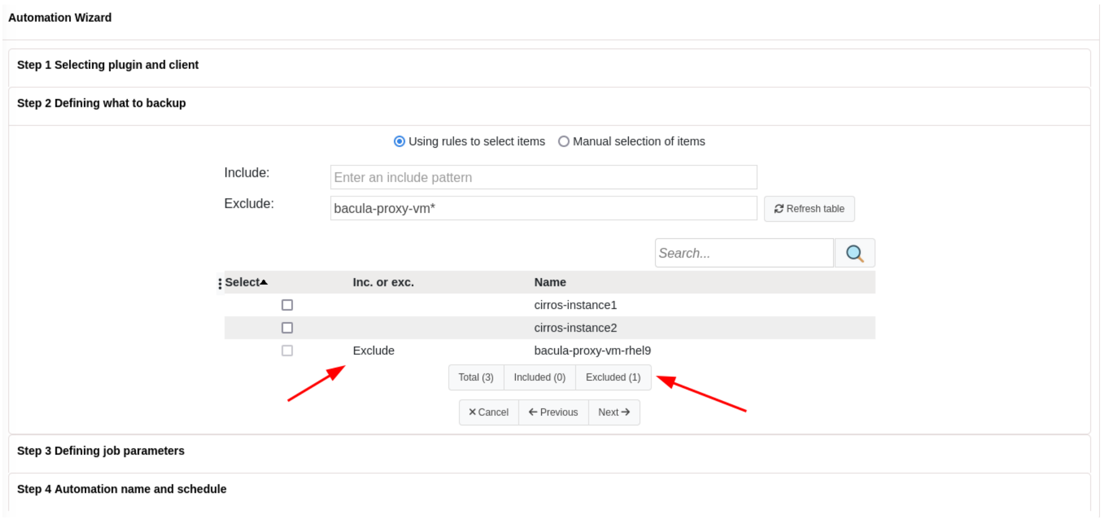
Include:	Enter an include pattern
Exclude:	bacula-proxy-vm*

Select	Inc. or exc.	Name
☐		cirros-instance1
☐		cirros-instance2
☐		bacula-proxy-vm-rhel9

Total (3) Included (0) Excluded (0)

Step 3 Defining job parameters

Step 4 Automation name and schedule



Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

[✓ Using rules to select items] Manual selection of items

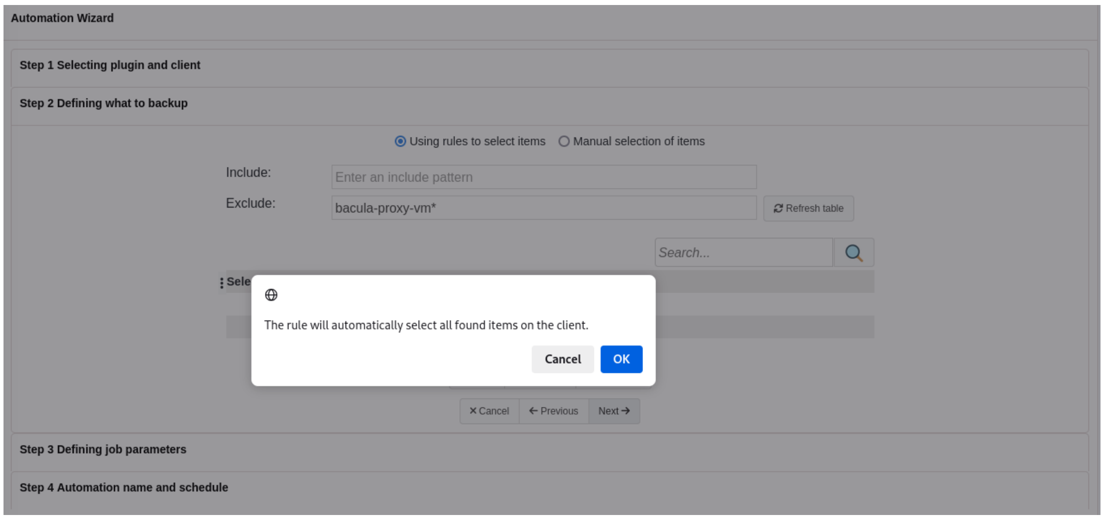
Include:	Enter an include pattern
Exclude:	bacula-proxy-vm*

Select	Inc. or exc.	Name
☐		cirros-instance1
☐		cirros-instance2
☐	Exclude	bacula-proxy-vm-rhel9

Total (3) Included (0) Excluded (1)

Step 3 Defining job parameters

Step 4 Automation name and schedule


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

[✓ Using rules to select items] Manual selection of items

Include: Enter an include pattern

Exclude: bacula-proxy-vm*

The rule will automatically select all found items on the client.

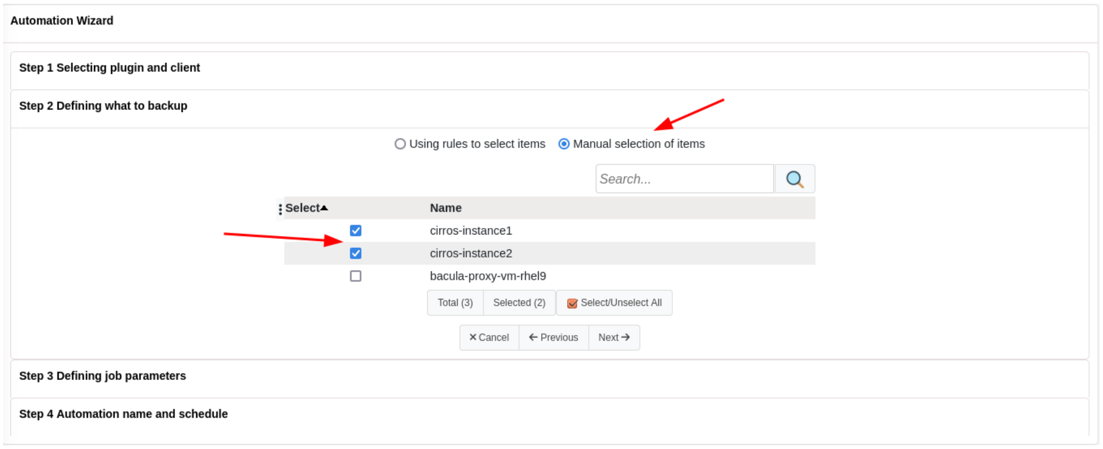
Step 3 Defining job parameters

Step 4 Automation name and schedule

Note: It is possible to use both the Include and Exclude options in the same automation Job.

---

c) manually selecting the items to be included in the OpenStack backup


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

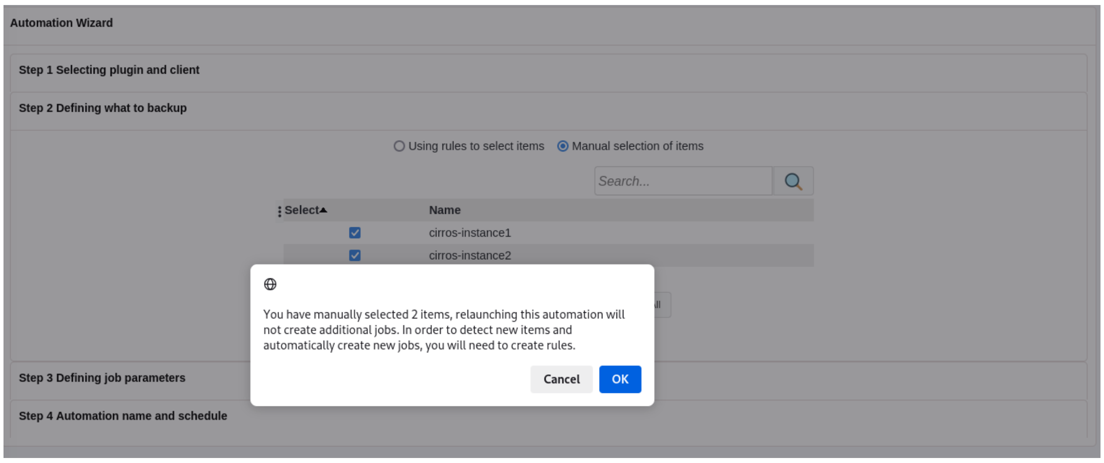
Using rules to select items [✓ Manual selection of items]

Select	Name
✓	cirros-instance1
✓	cirros-instance2
☐	bacula-proxy-vm-rhel9

Total (3) Selected (2) Select/Unselect All

Step 3 Defining job parameters

Step 4 Automation name and schedule


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

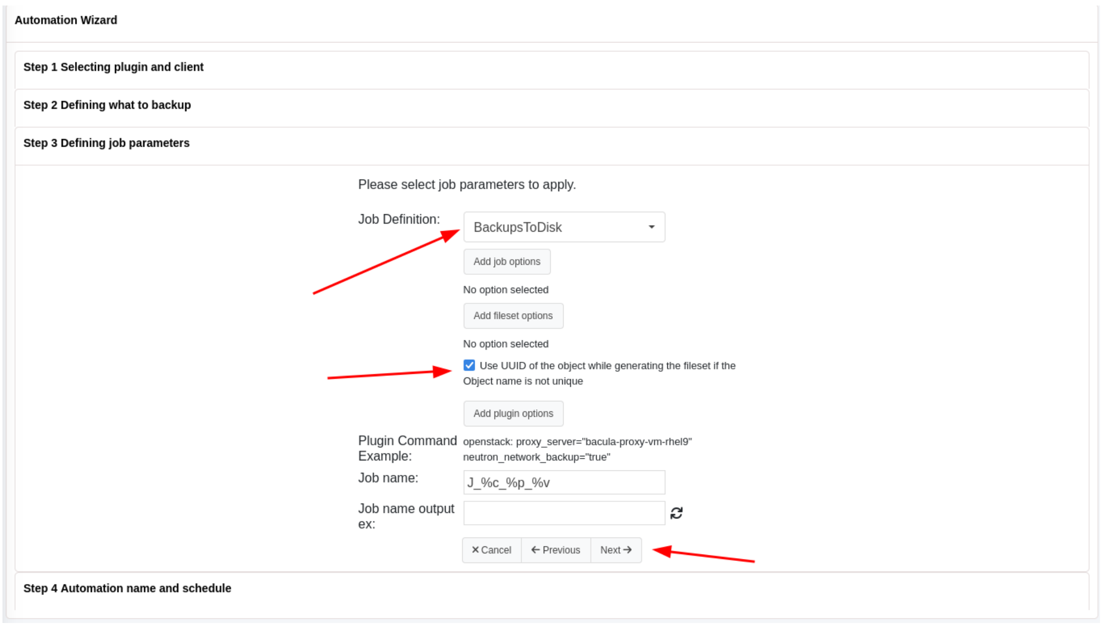
Using rules to select items [✓ Manual selection of items]

You have manually selected 2 items, relaunching this automation will not create additional jobs. In order to detect new items and automatically create new jobs, you will need to create rules.

Step 3 Defining job parameters

Step 4 Automation name and schedule

6) In Step 3, the Job parameters can be configured:


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

Step 3 Defining job parameters

Please select job parameters to apply.

Job Definition: BackupsToDisk

No option selected

No option selected

✓ Use UUID of the object while generating the fileset if the Object name is not unique

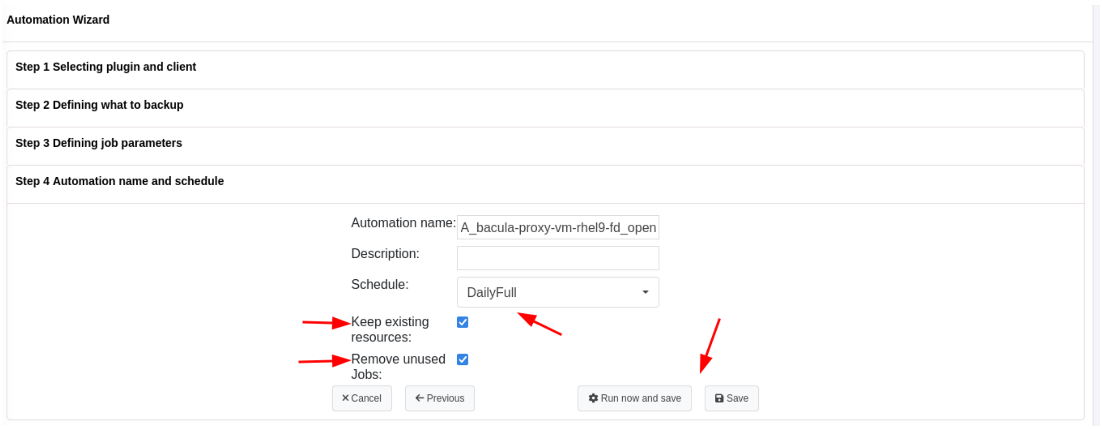
Plugin Command Example: openstack: proxy_server="bacula-proxy-vm-rhel9" neutron_network_backup="true"

Job name: J_%c_%p_%v

Job name output ex:

Step 4 Automation name and schedule

7) In Step 4, the Automation name, a Schedule to run the Automation, and other options can be configured:


Automation Wizard

Step 1 Selecting plugin and client

Step 2 Defining what to backup

Step 3 Defining job parameters

Step 4 Automation name and schedule

Automation name: A_bacula-proxy-vm-rhel9-fd_open

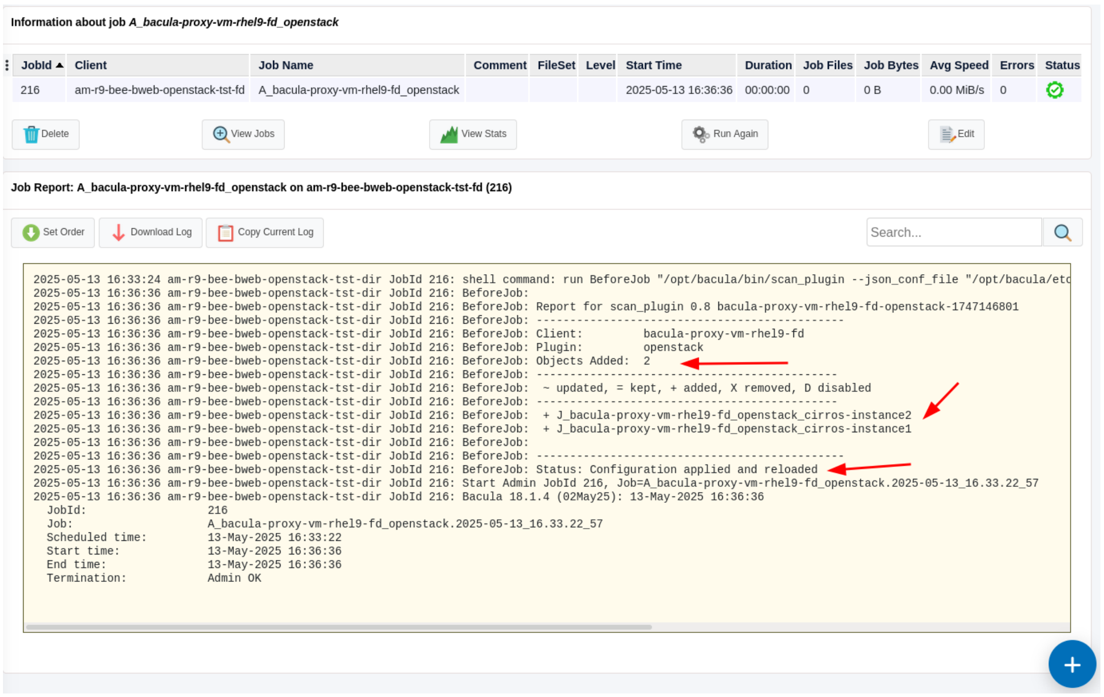
Description:

Schedule: DailyFull

Keep existing resources: ✓

Remove unused Jobs: ✓

8) By clicking in `Run now and save` in Step 4, the Automation Job is saved, it runs, and it adds two Jobs for both the `cirros_instance1` and the `cirros_instance2` objects:


Information about job *A_bacula-proxy-vm-rhel9-fd_openstack*

JobId ▲	Client	Job Name	Comment	FileSet	Level	Start Time	Duration	Job Files	Job Bytes	Avg Speed	Errors	Status
216	am-r9-bee-bweb-openstack-tst-fd	A_bacula-proxy-vm-rhel9-fd_openstack				2025-05-13 16:36:36	00:00:00	0	0 B	0.00 MiB/s	0	✓

Job Report: A_bacula-proxy-vm-rhel9-fd_openstack on am-r9-bee-bweb-openstack-tst-fd (216)

```
2025-05-13 16:33:24 am-r9-bee-bweb-openstack-tst-dir JobId 216: shell command: run BeforeJob "/opt/bacula/bin/scan_plugin --json_conf_file "/opt/bacula/etc
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob:
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: Report for scan_plugin 0.8 bacula-proxy-vm-rhel9-fd-openstack-1747146801
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: ---------------------------------------------
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: Client:         bacula-proxy-vm-rhel9-fd
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: Plugin:         openstack
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: Objects Added:  2
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: --------------------------------------------
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob:  ~ updated, = kept, + added, X removed, D disabled
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: --------------------------------------------
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob:  + J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance2
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob:  + J_bacula-proxy-vm-rhel9-fd_openstack_cirros-instance1
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob:
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: ---------------------------------------------
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: BeforeJob: Status: Configuration applied and reloaded
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: Start Admin JobId 216, Job=A_bacula-proxy-vm-rhel9-fd_openstack.2025-05-13_16.33.22_57
2025-05-13 16:36:36 am-r9-bee-bweb-openstack-tst-dir JobId 216: Bacula 18.1.4 (02May25): 13-May-2025 16:36:36
  JobId:                  216
  Job:                    A_bacula-proxy-vm-rhel9-fd_openstack.2025-05-13_16.33.22_57
  Scheduled time:         13-May-2025 16:33:22
  Start time:             13-May-2025 16:36:36
  End time:               13-May-2025 16:36:36
  Termination:            Admin OK
```

Also, Bacula offers a solution regarding automatic objects integration that works across plugins:

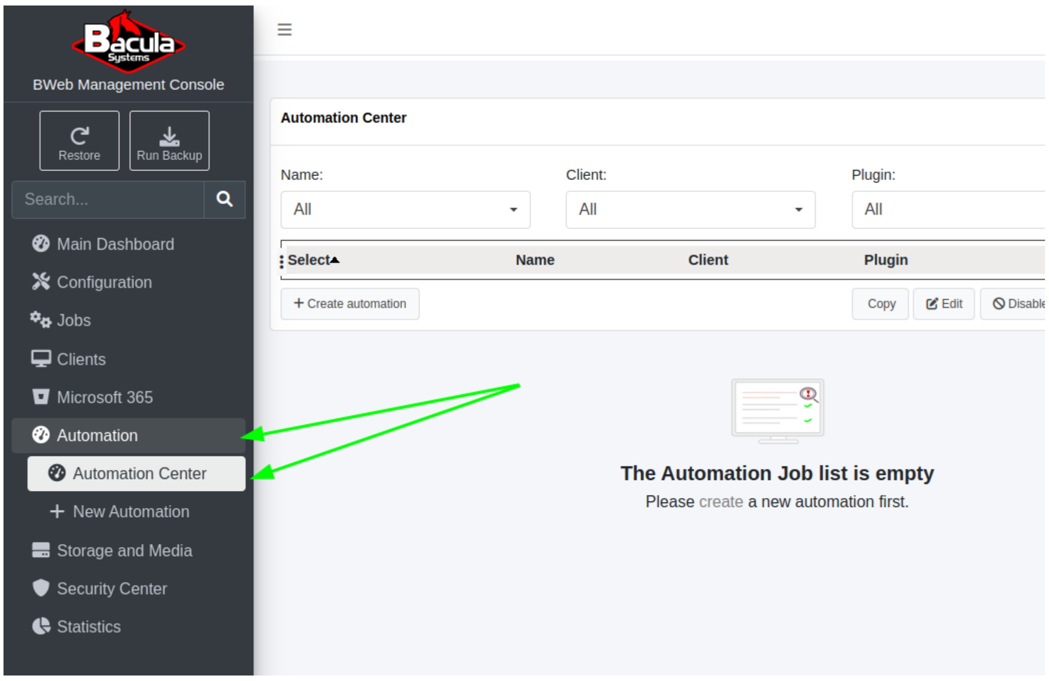
9 Automatic Object Integration (Scan Plugin)

This chapter will explain how to automatically configure Bacula with systems such as VMWare, HyperV or NetApp among others where each object can be backed up separately with different Jobs to maximize the throughput and the resiliency.

It is a best practice to setup a Job per virtual machine, per database or per NAS volume, however, it becomes difficult to adjust the Bacula configuration each time an object is added or deleted from the system. The Scan Plugin can query the plugin and create or delete the corresponding Bacula configuration automatically. A Job and a corresponding Fileset will be created for each object detected by the Plugin.

With some simple rules, Bacula can then detect and adjust it's configuration after each change done on the system, like a virtual machine, a database or a new volume added or deleted.

Important: BWeb Automation Center is recommended for use in place of manual configuration of the Scan plugin.



The Automation Job list is empty

Please create a new automation first.

Index