Bacula Enterprise Planning and Preparation

Bacula Systems Documentation

Contents

1 Backup Policy **2**
1.1 Framework for Backup Policy 2
1.2 Naming Resources 4
1.3 Schedules and Retentions 6
1.4 Backup Policy Configuration 7
1.5 Best Practices 9

2 Supported Platforms **9**
2.1 Windows Supported Platforms 9
2.2 Linux Supported Platforms 10
2.3 Other Supported Platforms 10

3 System Requirements **10**

Contents

The following chapter aims at advising you on how to plan and prepare to start with Bacula Enterprise.

1 Backup Policy

The following article presents information on Bacula backup policy.

1.1 Framework for Backup Policy

A backup policy helps manage users' expectations and provides specific guidance on the "who, what, when, and how" of the data backup and restore process. Collecting information about backing data up before it is needed helps prevent problems and delays that may be encountered when a user needs data from a backup. There are several benefits to documenting your data backup policy:

- It helps clarify the policies, procedures, and responsibilities.
- It will define:
  - where backups are located
  - who can access backups and how they can be contacted
  - how often data should be backed up
  - what kind of backups are performed
- Other policies or procedures that may already exist or that supersede the policy (such as contingency plans) are identified.
- A schedule for performing backups is well-defined.
- It will identify who is responsible for performing the backups and their contact information. This should include more than one person, in case the primary backup operator is unavailable.

- It will define who is responsible for checking that backups have been performed successfully, how and when they will perform this checking.
- A policy ensures data can be completely restored.
- A training plan for those responsible for performing the backups and for the users who may need to access the backups should be mandatory.
- The data Backup is should be fully automated, if technically possible.
- The policy will ensure that more than one copy of the backup exists and that it is not located in same location as the originating data.
- It will ensure that a variety of media are used to backup data, as each media type has its own inherent reliability issues.
- It will ensure that anyone new to the project or office can be given the documentation which will help inform them and provide guidance.

Defining a Data Backup Policy helps overview your infrastructure as well as your backup needs in order to create JobDefs, Schedules, Pools and Jobs that will be adequate for your environment.

Example

1. Scope of Policy
2. Purpose
3. Legal and Regulatory Obligations
4. Policy
   (a) Schedule of every important piece to be backed up
   (b) Storage of the first line Data Backup - Disk
   (c) Transport and storage of tapes
   (d) Tape Rotation and Storage
   (e) Regular data backup verification
   (f) Data recovery test in case of disaster recovery - scope and schedule
   (g) Restoration request process
   (h) Backup logs management
   (i) Backup monitoring
   (j) Backup Failure Management
   (k) Disposal of redundant/damaged tapes
5. Reporting - Role and Responsibilities
   (a) Backup and data recovery
   (b) Verifications
   (c) Disaster Recovery situation
   (d) Policy Implementation
   (e) Policy Review

Note: The information given in this article is for the purpose of information mainly. It needs to be adapted to the infrastructure, requirements, and possibly regulatory obligations of your organization.

1.2 Naming Resources

There are a lot of resources you will need to configure in Bacula. Therefore it is a very good practice to define and standardize the naming. The goal is to avoid ending up using Jobs named Job1, Job2, etc., which do not describe what they do and will be difficult to understand after several months of use. Your configuration will gain clarity.

Examples

Here are some naming examples for your configuration:

- Director/Storage Daemon(s)/Clients should be named with their location and hostname. If they are dedicated, one can add a description. At the end we add -dir / -sd / -fd

```
Europe-bacula1-dir
Iceland-baculabkp-DR-dir
NewYork-storage1-dedup-sd
Houston-clt2314-vsphere-fd
```

- Autochangers will be easy to find if their name starts with the Storage Daemons name, then a description and a trailing -autochanger

```
NewYork-storage1-dedup-xfs-autochanger
```

- Devices should refer to the naming of the Storage Daemon they are attached to, as well as, in the case of a Device inside an autochanger, its name and a number. A -device can be added at the end.

```
NewYork-storage1-dedup-xfs-autochanger-tape1-device
NewYork-storage1-dedup-xfs-autochanger-tape2-device
```

- Jobs will have the name of the client, the main function and a trailing -job:

```
ActiveDirectorySrv1-SystemState-job
DataServ-MySQL-job
RedHat6-vSphere-job
```

- JobDefs can be named according to the group of clients they for, with a trailing -jd

```
MainProduction-jd
TestMachines-jd
DedupHomes-jd
DataBase-jd
```

- Copy/Migration Jobs can have a trailing -copyjob or -migrationjob:

```
ActiveDirectory-SystemState-copyjob
DataServ-MySQL-migrationjob
RedHat6-vSphere-copyjob
```


- Filesets can be named according to what they do with a trailing -fs

```
WindowsAllDrives-fs
WindowsC-fs
Home-fs
TypicalUnixServer-fs
```

- Pools can be named according to the location or name of the Storage Daemon, the Media Type, the frequency, the level of the backup and a trailing -pool

```
ParisWeeklyDiskFull-pool
NewYorkDailyDedupFull-pool
SafeHouseMonthlyVirtualFull-pool
Bacula-sd2-ZFS-Differential-pool
```

- Clients could be named with the hostname and a trailing -fd

```
hrserver.domain-fd
ceolaptop.domain-fd
```

- Schedules might contain the frequency and/or the purpose, plus a trailing -sched

```
DailyMorningCatalog-sched
DailyEveningVM-sched
```

- Console should be named with the hostname and a description followed by a trailing -console

```
NewYorkSDrestrictedRestore-console
VPSalesBackup-console
```

- Messages resources can be customized per daemon or per Job and a trailing -messages:

```
Europe-bacula1-dir-messages
ActiveDirectory-SystemState-job-messages
```

Note:

- Don't use spaces in naming, it will work but requires the directives to be quoted.
- You can also define acronyms, like DR for Disaster Recovery, EX for External storage or countries such as FR for France. However, using a consistent and fixed scheme is essential.
- Use the **"Description"** directive to add even more details for your resources (when possible).
- Stick to your naming convention regarding "-", "_", format. Do not rename resources after they were set and used (See section sec:renamingresource on renaming resources for more details).
- Resource names are limited to a fixed number of characters. Currently the limit is defined to be 127 characters.

1.3 Schedules and Retentions

Schedules and retentions play a major role in calculating the requirements of a backup system. The number of tapes and the disk space required depend to a large extent on these two parameters. For instance, a 10MB backup taken every hour with a retention period of one month would require at least 7200MB of disk space to hold the full backup cycle.

Setup

Bacula will do exactly what you tell it to do. This is why you should be careful with Schedules and Retentions. If the frequency of your Schedule is high and the retentions are long, you will need a lot of disk space. In addition the catalog will be large, as its size is directly proportional to the number of jobs and the retention periods.

On the other hand, if you miss your retention and scheduling targets implied by policies, you may end up having your last Full backup recycled before a new Full has started.

Example

Let's say that we have a pool "Storage1-WeeklyFull-pool" with all retention times set to 7 days. Further we have an incremental pool "Storage1-DailyInc-pool" with the same retention times. The Full backups are scheduled each Monday night and Incrementals daily from Tuesday to Sunday:

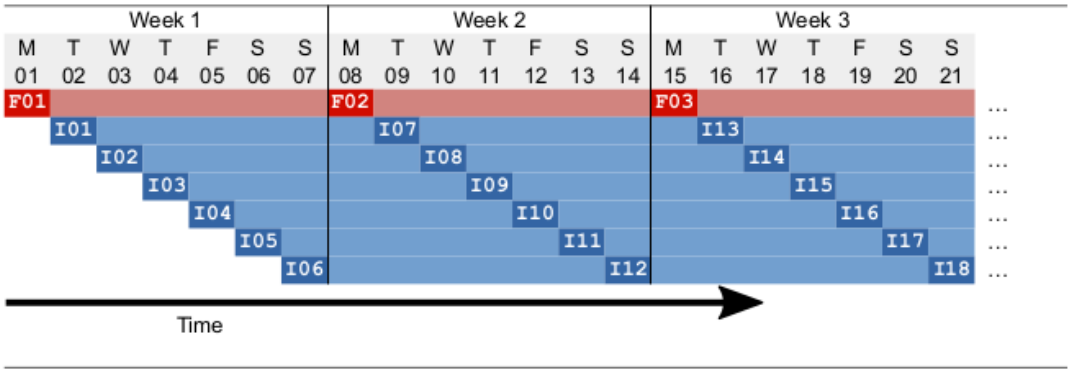

Fig. 1: Example for a simple schedule with 7 days of retention for Full and Incremental backups

We can see the **Full** backups in **red** and the *Incremental* backups in *blue*. The retention times are indicated by a lighter color. The first Full F01 will expire on the same day we take the next Full F02. This is dangerous: if for any reason the Full job F02 does not succeed, no Fulls are available at that point in time. In this case, setting the full pool retention to 8 or 9 days allows the backup administrator to re-run a Full manually (disabling the Incremental of the day while running). Alternatively, you can configure Maximum Full Interval to 7 days to force the promotion of an Incremental to a Full if no Fulls are available in the catalog for the last 7 days.

If you want to guarantee to your users that they can *go back in time* for 7 days, the retention times in the example scheme depicted above will be too short: Imagine a user on day 11 (Thursday) in the morning (I09 will be taken in the late evening) wanting to have a file back from the week before, and the only thing she knows is that she last saw the file on Friday of the previous week. Since the retention time of Full F01 has expired, all the Incremental backup that followed can no longer be used to construct the file tree, because Bacula will need a Full and all following Incremental for this. Thus at this point in time you will only be able to go back to the Monday of the same week when the most recent Full was taken.

Of course it could be that the file we are looking for has changed on day 5 (Friday of the week before) and was backed up in the Incremental I04, but this is a special case. If the file did not change it would not have been in that backup.

To accomplish the goal at all times with the above scheme the retention time for the Full should be at least 15 days.

Read further about Retention Period Configuration.

Note: The example above is a simple one, there are other ways of scheduling: using a specific week number (beware of 53 weeks years), or depending on a specific weekday (4th Sunday for example, beware of months with 5 Sundays). Theses variations must be well verified to avoid overlapping or, worse, missing a Full backup and thus creating a gap. In such a case you would be unable to restore in certain situations.

When planning schedules, it is essential to pay attention to corner cases, as sometimes the year has 53 weeks. or a month 5 weeks. In such cases if the retention period doesn't take these situations into account the previous full backup may be pruned or overwritten before it should. It is recommended to configure Bacula in such a way that at least two full backups co-exist for a while. It is better to be safe than sorry.

Along with storage space, the catalog also grows proportionally to the number of files and the frequency of the backups.

1.4 Backup Policy Configuration

The following article presents information on configuration of Bacula backup policy.

Retention Period Configuration

You can define several retention times in your pools, but also for each clients. The first rule is that the file retention must be lower or equal to job retention which should be lower or equal to the volume retention. Normally you should set up Retention so they match the following rule:

File Retention <= Job Retention <= Volume Retention

The longer you keep file entries in the Catalog, the bigger the catalog. As a rule of thumb you can expect at least 150 bytes per file or object stored in the catalog.

Once the File Retention and the Job Retention are over for all Jobs stored in a Volume, this volume can be reused at any time by Bacula for next backup Jobs.

Retention times are used by Bacula to recycle your volume so your tapes and disk Volumes can be re-used. That's why it is important to set them correctly so volumes are not recycled "too early" (see also figure fig:retention in section BEPPSchedulesAndRetentions). These retention settings must match your requirements.

In most cases, you should define **File Retention** and **Job Retention** in the `Client{}` resource, while the **Volume Retention** should be defined in the `Pool{}` resource.

In a Client resource:

```
Client {
   Neme = HRserv-fd # Human Ressources server File Daemon
   ...
   File Retention = 94608000 # 3 years in seconds
```

(continues on next page)

(continued from previous page)

```
   Job Retention  = 103680000 # 3 years and 4 months in seconds
}
```

In a Pool:

```
Pool {
   Name = 3years-pool
   Description = long term archiving Pool
   ...
   File Retention = 94608000 # 3 years in seconds
   Job Retention  = 103680000 # 3 years and 4 months in seconds
   Volume Retention = 106272000 # 3 years and 5 months in seconds
}
```

Note: If File retention is lower but not equal to Job Retention, you will, during the "gap" between those times, still be able to query the catalog for the Job but not directly for a file of this Job.

Pay great attention to retention times to avoid unwanted recycling or non-recycling.

Dedicated Restore Drive

When you set up your storage and drive, it can be useful to have a dedicated drive for restoration purpose as devices in Bacula cannot backup and restore at the same time. This applies to all device types. With virtual Autochangers (aka disk changers), you can create as many devices as you want, thus having dedicated ones fore restores plus a few spare ones is a good practice.

In order to have a dedicated device, you just need to add the Directive `Autoselect = no` to a device definition in your Storage Daemon configuration.

An example device dedicated for a restore:

```
Device {
   Name = RestoreDrive
   Media Type = File
   Archive Device = /pool/Backup_storage
   LabelMedia = yes
   Random Access = Yes
   AutomaticMount = yes
   RemovableMedia = no
   AlwaysOpen = no
   Autoselect = no     # this drive will not be selected for a backup
}
```

Note: In the case of a Tape Library, you can dedicate a tape drive or adjust Maximum Concurrent Job values in order to have one free "most of the time". Your restore Job will then be processed once a drive is free. Priorities can be used to ensure restore jobs run as soon as possible, and backup jobs can be stopped or cancelled to free up drives.

1.5 Best Practices

Compression in Bacula

Hardware and software data compression and communication line compression between daemons can be used independently from each other.

Hardware Compression

Hardware compression by tape drives is not directly configured through Bacula configuration files. Instead, it can be enabled or disabled with your tape drive, library, or the operating system following the vendor's instructions.

The default for most tape drives is to have compression enabled. It is strongly advised to keep this default value enabled, as hardware compression is usually faster and more efficient than software compression.

Bacula's software compression is configured in Fileset Include Options, can be controlled in a finely granular manner, and is independent from hardware compression. Combining both is usually not a good choice. When software compression is configured for some data, but needs to be disabled for backups going to particular storage, such as tape, the directive AllowCompression in the Director's Storage resources is available.

Communication Line Compression

Regarding "Communication Line Compression", Bacula Enterprise uses compression between daemons by default. It is advised to keep it enabled. If for any reason you need to disable this feature, you can do this by configuring `Comm Compression = no`. Refer to the directive Comm Compression. Note that the directive can be found in the DIR, the SD, the FD and the Console.

The compression algorithms used and the libraries implementing it are well tuned on x86 / AMD64 hardware, but often inefficient on other systems. In particular on RISC CPUs (Sparc or Power, for example) it may be better to disable this function.

2 Supported Platforms

The following document presents supported platforms.

2.1 Windows Supported Platforms

Microsoft Windows Client and Windows Server supported by Microsoft are fully supported by **Bacula**.

Note: Bacula Director is not supported on Windows.

Bacula Storage Daemon is available for Windows, but only for specific use cases. It is discouraged to run it in production, but only with certain precautions. Consult the Support Team to explain your use case to determine if you enter under such a case.

Unsupported versions of Windows Client and Windows Server are not supported by **Bacula**. That said, we will make a best effort to support them, but we cannot guarantee that new software developed after the support expiration date of each Microsoft OS will be supported.

2.2 Linux Supported Platforms

It is recommended to run Bacula Enterprise Director and Storage Daemons on a recent Linux distribution such as Red Hat, Oracle Linux, AlmaLinux, Rocky Linux, Ubuntu, Debian, or SLES.

Note: Red Hat, Oracle Linux, AlmaLinux, and Rocky Linux are identified as RHEL within the documentation.

The Client (File Daemon) can run on usual platforms running in an enterprise. We provide support for i386 architecture, as well as other architectures like Power 9 or ppc for the Client. We support for the Client Red Hat Enterprise Linux, Oracle Linux, AlmaLinux, Rocky Linux, Ubuntu, Debian and Suse Linux Enterprise Server.

The availability of plugins may be limited to specific platforms. However, comprehensive documentation is provided for each plugin, ensuring that all necessary information is readily available.

2.3 Other Supported Platforms

Bacula Enterprise and Bacula are known to run on a very large variety of platforms.

The Director and the Storage Daemons can be run on Solaris, i386 and SPARC.

The Client or the File Daemon (FD) can be run on the following platforms:

- HP-UX
- macOS
- Solaris
- AIX
- Other if the Bacula Community FD source code can be compiled on it. The Bacula Community FD is compatible with the Director and Storage Daemon Community and Enterprise version.

3 System Requirements

Bacula will require some space for temporary files and logs.

By default, the directory `/opt/bacula/working` is used to store the following things:

- Messages spooled by the Director
- **BWeb** log files
- spooled SD attributes
- DIR, FD, SD trace and dump files
- Plugin log files and other plugin-specific data

It is best practice to have the following partitioning to avoid filling your root space.

It is **recommended** you have:

- at least 10GB+ for `/opt`, we recommend this value as `/opt/bacula/working` is heavily used for caching purpose, plugin logs and single file restore mount points and logs.

- a separated partition for your catalog (typically, at least 150 bytes and up to a multiple of that are used for each object and file kept in the catalog).
- a separate file system backed by dedicated storage devices (striped NVMe or SSDs) for data spooling (if you use spooling to tapes). The spooling configuration (Directive **SpoolSize**) should be adjusted to allow for 80 to 95 percent usage of the storage hardware. Remember that spooled attributes will go by default to `/opt/bacula/working`.

Note: Partitioning your system will guarantee an efficient load on each partition and avoid filling of the root one.

Regarding the space allowed for disk based backup, please configure it to be extensible by using LVM/ZFS or any other mechanism, even by assigning jobs to a dedicated partition and use copy and migration jobs afterwards to keep useful jobs.
