S3 Plugin

Bacula Systems Documentation

Contents

Contents

1 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.

1.1 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 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

1.2 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*).

1.3 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.

2 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.

2.1 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.

2.2 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.

2.3 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.

3 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:

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.

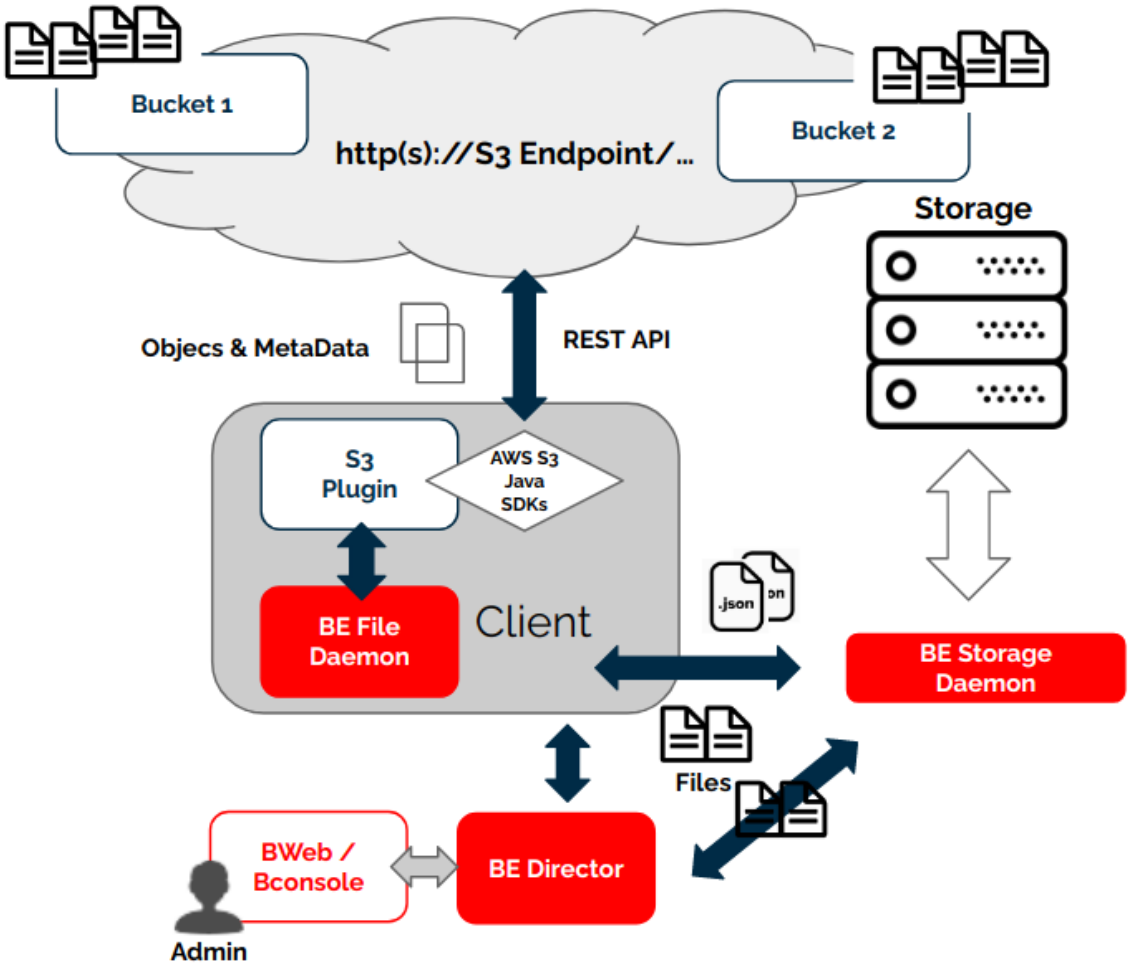

Fig. 1: S3 Plugin Architecture

3.1 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`

4 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).

5 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 1: **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
```

6 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.

6.1 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 2: **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 3: **APT install**

```
apt update
```

Then, the plugin may be installed using:

Listing 4: **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 5: **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.

7 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 6: **Fileset S3**

```
FileSet {
   Name = FS_S3
   Include {
      Options {
        signature = MD5
        ...
      }
      Plugin = "s3: <s3-parameter-1>=<s3-value-1> <s3-parameter-2>=<s3-value-2> ..."
   }
}
```


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 ','.

7.1 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	Required	Default	Values	Example	Description
abort_on_error	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_file	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/etc/s3.settings	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/bacula/working/s3-debug.log	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/bacula/working	An existing path with enough permissions for File Daemon to create any internal plugin file	/mnt/my-vol/	Uses this path to store metadata, plugin internal information, and temporary files

7.2 Advanced common parameters

The following are advanced parameters. They should not be modified in most common use cases:

Option	Required	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_wait	No	120	Positive integer (seconds)	360	Max wait time for FD to answer packet requests
time_max_last_modify_log	No	86400	Positive integer (seconds)	43200	Maximum time to wait to overwrite a debug log that was marked as being used by another process
logging_max_file_size	No	50MB	String size	300MB	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_backup_index	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
split_config_file	No	=	Character	:	Character to be used in config_file parameter as separator for keys and values
opener_queue_timeout_secs	No	1200	Positive integer (seconds)	3600	Timeout when internal object opener queue is full
publisher_queue_timeout_secs	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…

7.3 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	Required	Default	Values	Example	Description
backup_queue_size	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_threads	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_network_retries	No	5	Positive integer (number of retries)	10	Number of retries for the general external retry mechanism
general_network_delay	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/

7.4 S3 Service Parameters

Parameters to connect and control the behavior of the plugin regarding the S3 service:

Option	Required	Default	Values	Example	Description
endpoint	No	https://s3.amazonaws.com/	URL of a S3 endpoint	https://192.168.10.4:9000	Main URL where the S3 service is being served
access_key	Yes		Valid S3 access key to read from or write to buckets	KMN02jCV5Yd[illegible]O	Valid access key to read from or write to buckets
secret_key	Yes		Valid S3 secret key associated to the provided access_key to read from or write to buckets	bTq6FzPbn[illegible]yq	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_path_style	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)
bucket	No		Strings representing existing **buckets** for the given access information (endpoint, keys and region) separated by ‘,’	mybucket1,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
bucket_exclude	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)
bucket_regex_include	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.
bucket_regex_exclude	No		Valid regex	.*Plan	Exclude matching buckets from the selection. Please, only provide list parameters (bucket + bucket_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.
folder	No		Strings representing existing **folders** for the applicable buckets separated by “,”	images, docs	Backup only specified **folders** belonging to the selected buckets
folder_exclude	No		Strings representing existing **folders** for the applicable buckets separated by “,”	personal	Exclude selected **folders** belonging to the selected buckets
folder_regex_include	No		Valid regex	.*Company	Backup matching folders. Please, only provide list parameters (folders + folders_exclude) or regex ones. But do not try to combine them.
folder_regex_exclude	No		Valid regex	.*Plan	Exclude matching folders from the selection. Please, only provide list parameters (folders + folders_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 is excluded.
version_history	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include former versions of every object into the backup process
acl	No	No	0, no, No, false, FALSE, false, off ; 1,	Yes	Backup object ACLs

Note: force_path_style option is available since BE 16.0.7.

7.5 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	Required	Default	Values	Example	Description
destination_bucket	No		Destination bucket name	myrestorebucket	Destination bucket where restore data will be uploaded. If no bucket is set, every selected file will be restored in the original bucket
destination_path	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_storageclass	No		STANDARD, REDUCED_REDUNDANCY, GLACIER, STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, DEEP_ARCHIVE, OUTPOSTS, GLACIER_IR	ONEZONE_IA	Destination storage class to be used for the restore. If none is provided, the original storage class of this object will be used
skip_versions	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_acl	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_hashcheck	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_key	Yes		Valid S3 access key to read from or write to buckets to backup	KMN02jCC5Ysneir0a	Cross-endpoint/bucket restore: Valid access key to write to the bucket to restore
secret_key	Yes		Valid S3 secret key associated to the provided access_key to read from or write to buckets to backup	bTq6FzPbnU9x1jqkl5STRDuk3CPladyS	Cross-endpoint/bucket restore: 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_path_style	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

8 Operations

8.1 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 7: **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 8: **Fileset Example**

```
FileSet {
   Name = FS_MYBUCKET
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: config_file=/opt/bacula/etc/s3.settings bucket=mybucket"
   }
}
```

Listing 9: **Settings file**

```
$ cat /opt/bacula/etc/s3.settings
access_key=XXXXXXXXXXXXXXXXXXX
secret_key=YYYYYYYYYYYYYYYYYYY
region=us-east-1
bucket=mybucket
```

Increase number of threads:

Listing 10: **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 11: **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 12: **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 13: **Fileset Example**

```
FileSet {
   Name = fs-mybucket-startA
   Include {
      Options {
        signature = MD5
      }
      Plugin = "s3: access_key=XXXXXXXXXXXXXXX secret_key=YYYYYYYYYYYYYYYY␣
↪bucket=mybucket folder_regex_include=A.*"
```

(continues on next page)

(continued from previous page)

```
   }
}
```

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 14: **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 15: **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\" "
   }
}
```

8.2 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 16: **Restore Bconsole Session**

```
**restore
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
```

(continues on next page)

(continued from previous page)

```
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.
$ 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.
```

(continues on next page)

(continued from previous page)

```
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
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
```

(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 : s3: region="US-EAST-1" access_key="XXXXXXXXXXXXXXXXXXXXXXXXX"␣
↪secret_key="YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY" 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)
     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)
```

(continues on next page)

(continued from previous page)

```
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.

8.3 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 are some examples:

List S3 Contents

Listing 17: **List example: General information**

```
*.ls plugin="s3:region=\"US-EAST-1\" access_key=\"XXXXXXXXXXXXXXXXXXX\" secret_key=\
↪"YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\" bucket=\"bacbucket\" " 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  /@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
```

(continues on next page)

(continued from previous page)

```
 -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
```

9 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/

10 Troubleshooting

This section lists some scenarios that are known to cause issues and how to solve them.

10.1 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.

