OpenStack Plugin

Bacula Systems Documentation

Contents

Contents

> Note
>
> You can download this article as a PDF

Enterprise

> Bacula Enterprise Only
>
> This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

> **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.

1 Features

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

2 Best Practices

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

2.1 Openstack Instances and Snapshot Consistency

Openstack Nova usually relies on QEMU to manage virtual machines. QEMU provides a mechanism to communicate with an agent installed on the instance, allowing the instance to take specific actions before a snapshot is created.

This communication happens via a virtual device added to the instance when OpenStack Nova detects that the image being used contains the property `hw_qemu_guest_agent=yes`.

Verifying that `hw_qemu_guest_agent` is activated:

- In Horizon: the `Custom Properties` of the image should display `hw_qemu_guest_agent yes`.

- On CLI the command: `openstack image show -f value -c properties <name or uuid>` should display `'hw_qemu_guest_agent': 'yes'`

Note

It is possible to update the properties of an already existing image.

- In Horizon: Go to Images, select the relevant image, and choose Update Image Metadata. Under `libvirt Driver Options for Images`, set `hw_qemu_guest_agent` to `yes`.
- On CLI: `openstack image set --property hw_qemu_guest_agent=yes <image name or uuid>`

To ensure that snapshots are consistent, you can create a custom hook file for freeze/thaw actions at `/etc/qemu-ga/fsfreeze-hook.d/test_hook.sh`

After taking a snapshot, verify that the hook is executed for both freeze and thaw events.

Here is an example of such a hook script:

Listing 1: Test hook

```
#!/bin/bash

case $1 in
freeze)
echo "I'm frozen" > /tmp/freeze
;;
thaw)
echo "I'm thawed" >> /tmp/freeze
;;
*)
exit 1
;;
esac
```

It is then easy to check `/tmp/freeze` and confirm that the instance has been frozen/thawed.

Another way to verify this is by examining the QEMU guest agent logs, as shown in the example below.

```
tail /var/log/qemu-ga/qga-fsfreeze-hook.log
Mon Jan 01 12:00:00 PM EST 2025: Executing /etc/qemu-ga/fsfreeze-hook.d/test-
↪hook.sh freeze
Mon Jan 01 12:00:01 PM EST 2025: /etc/qemu-ga/fsfreeze-hook.sh finished␣
↪successfully
Mon Jan 01 12:00:02 PM EST 2025: Executing /etc/qemu-ga/fsfreeze-hook.d/test-
↪hook.sh thaw
Mon Jan 01 12:00:03 PM EST 2025: /etc/qemu-ga/fsfreeze-hook.sh finished␣
↪successfully
```

2.2 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.

3 Backup and Restore Strategies

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

This article presents information regarding backup and restore strategies of the OpenStack Plugin.

3.1 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.

3.2 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.

3.3 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.

3.4 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-
↪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   |
+-------------------------------------------------------------------------------
↪------------------------------+
```

4 Installation

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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 ␣
```

(continues on next page)

(continued from previous page)

```
↪ | 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                   |
+----------------------------------+-----------+-------------+--------------
↪-+---------+-----------+------------------------------------------------+
```

4.1 OpenStack Installation with BIM

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

4.2 OpenStack Installation with Package Manager

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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
```

5 Configuration

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

5.1 Automatic Object Integration

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

5.2 General Parameters

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

The following OpenStack Plugin parameters impact Backup and Restore Jobs as well as the Query bconsole commands.

Table 1: 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/openstack.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.

5.3 Backup Parameters

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

5.4 Restore Parameters

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

Table 2: OpenStack Restore Parameters

Parameter	Values	Default	Description
proxy_ser	<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_USEI	<String		Specifies the username to access the OpenStack API.
OS_PASS	<String		Specifies the password to access the OpenStack API
OS_AUT	<String		Specifies the endpoint to access the OpenStack API
OS_USEI	<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.

5.5 Fileset Examples

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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"
```

(continues on next page)

(continued from previous page)

```
    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.

6 Operations

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

The following article describes details regarding backup, ingestion, restore with Bacula Enterprise OpenStack Plugin.

6.1 Backup

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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"
}

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.
```

(continues on next page)

(continued from previous page)

```
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
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):
```

(continues on next page)

(continued from previous page)

```
 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
 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
      }
      # 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
```

(continues on next page)

(continued from previous page)

```
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
  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)
```

(continues on next page)

(continued from previous page)

```
 Non-fatal FD errors:    0
 SD Errors:              0
 FD termination status:  OK
 SD termination status:  OK
 Termination:            Backup OK
```

6.2 Restore

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.
```

(continues on next page)

(continued from previous page)

```
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
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
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
```

(continues on next page)

(continued from previous page)

```
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
↪"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
```

(continues on next page)

(continued from previous page)

```
  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)
===========================================================================

    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
```

(continues on next page)

(continued from previous page)

```
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"
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-
```

(continues on next page)

(continued from previous page)

```
↪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:
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
```

(continues on next page)

(continued from previous page)

```
-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
```

6.3 bconsole Query Commands

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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
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`.

7 Limitations

Enterprise

Bacula Enterprise Only

This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

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.

