



# Swift Plugin

**Bacula Systems Documentation**

---

# Contents

<b>1</b>	<b>Executive summary</b>	<b>2</b>
<b>2</b>	<b>Swift Object Storage</b>	<b>2</b>
2.1	Architecture . . . . .	2
2.2	Features . . . . .	2
2.3	Installation of the Plugin . . . . .	3
2.4	Configuration . . . . .	4
2.5	Restore Examples . . . . .	6
2.6	Useful Swift Commands . . . . .	8
2.7	Limitations . . . . .	9

# Contents

---

<ul style="list-style-type: none"><li>• <i>Executive summary</i></li><li>• <i>Swift Object Storage</i></li></ul>
--

## 1 Executive summary

This document is intended to provide insight into the considerations and processes required to successfully implement a Swift Object Storage backup technique using Bacula Enterprise.

---

**Note:** This functionality is available as of Bacula Enterprise version 8.10.

---

## 2 Swift Object Storage

### 2.1 Architecture

The Bacula Enterprise Swift plugin uses the python-swiftclient library to access the Swift Object Storage daemon.

### 2.2 Features

The configuration for Swift container backups is done in a Bacula FileSet configuration file using parameters to the Swift plugin command line.

During a backup, the Bacula plugin will contact the Swift system to retrieve Objects one by one. During an incremental or a differential backup session, the Bacula File Daemon will need to list all objects and retrieve their attributes to determine if an object must be included in the job.

If found, any container or object metadata are saved during backup.

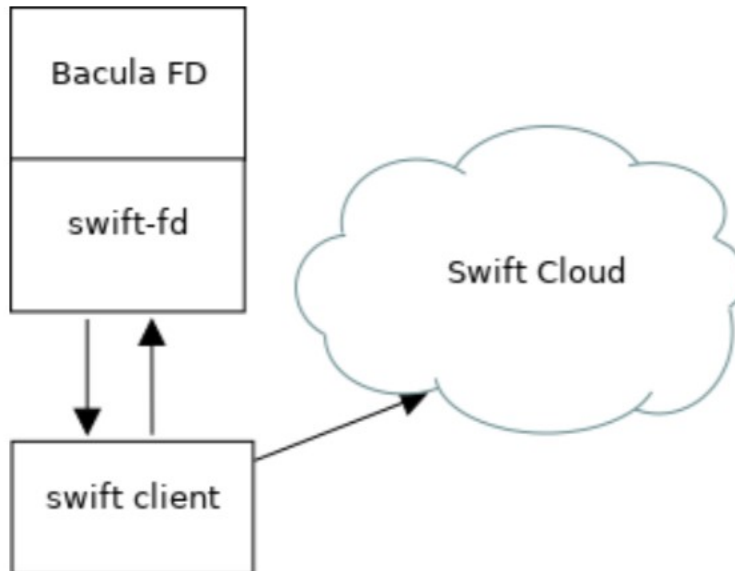


Fig. 1: Swift Plugin Architecture

At restore time, the Bacula Swift plugin will restore the Swift metadata, and the user will be able to restore Swift objects to:

- The original Swift container
- An alternate Swift container/system
- A local directory

Large objects will be restored automatically using the DLO technique.

Swift metadata for containers are automatically backed up with Objects.

## 2.3 Installation of the Plugin

On the Bacula File Daemon that you want to connect to your Swift Object Storage, extend the repository file for your package manager to contain a section for the Swift plugin. For example, in Redhat/CentOS 7, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseSwiftPlugin]
name=Bacula Enterprise Swift Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/swift/@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

On Centos7, python3 and the Swift client library are not available via the standard repositories. The EPEL repository provides python34 and the python-swiftclient package must be installed via the python34-pip package from this repository.

```
# yum install epel-release
# yum install python34 python34-pip
# pip3.4 install python-swiftclient
```

On Redhat7, the package is available in the repository “rhel-7-server-openstack-11-tools-rpms”.

On Debian Jessie, /etc/apt/sources.list.d/bacula.list:

```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@debs/bin/@version@/jessie-64/
↪jessie main
deb https://www.baculasystems.com/dl/@customer-string@debs/swift/@version@/jessie-64/
↪jessie swift
```

Once the repository is configured for your system, perform a yum update or apt-get update, then the package bacula-enterprise-swift-plugin can be installed with yum install or apt-get install.

```
# yum install bacula-enterprise-swift-plugin
or
# apt-get update
# apt-get install bacula-enterprise-swift-plugin
```

If you prefer to manually install the packages, you may download them directly from your download area, and use one of the low level package manager tools (rpm or dpkg) to do the plugin installation.

## 2.4 Configuration

### Plugin Parameters

The following parameters effect any type of Swift plugin Job (Backup, Estimation or Restore).

- `user=<string>` specifies the username to access the Swift system. This parameter is mandatory.
- `password=<string>` specifies the password to access the Swift system. The password or the `passfile` parameter is mandatory.
- `passfile=<string>` specifies a file local to the File Daemon that contains the password for the user. Only the first line of the file will be read. The password or the `passfile` parameter is mandatory.
- `url=<string>` specifies the URL of the Swift system. This parameter is mandatory.
- `abort_on_error=<0 or 1>`, or just `abort_on_error` specifies whether or not the Backend should abort it's execution if a fatal error occurs during Backup, Estimation or Restore. This parameter is optional. The default value is 0.
- `debug=<0 or 1>`, or just `debug` specifies how much information should be logged to the Backend File Log. This parameter is optional. The default value is 0.
- `insecure=<0 or 1>`, or just `insecure` specifies whether or not the SSL certificate should be verified. This parameter is optional. The default value is 0.

## Plugin Estimation and Backup Parameters

- `include=<string>` specifies which containers and/or objects should be backed up from the Swift Storage System. This parameter is optional. There may be more than one `include` parameter.
- `regexinclude=<regex>` specifies, using a Regular Expression, which containers and/or objects should be backed up from the Swift Storage System. This parameter is optional. There may be more than one `regexinclude` parameter.
- `exclude=<string>` specifies which containers and/or objects should NOT be backed up from the Swift Storage System. This parameter is optional. There may be more than one `exclude` parameter.
- `regexexclude=<regex>` specifies, using a Regular Expression, which containers and/or objects should NOT be backed up from the Swift Storage System. This parameter is optional. There may be more than one `regexexclude` parameter.

If none of the parameters `include`, `regexinclude`, `exclude` or `regexexclude` are specified, all containers and objects from the Swift Storage System for an account (user parameter) will be backed up.

## Plugin Restore Parameters

- `user=<string>` specifies an account where restore will be performed. This parameter is optional. If not set, the user parameter from the backup Job will be used.
- `password=<string>` specifies the password to access the Swift system during restore. This parameter is optional. If not set, the `password=<string>` parameter from the backup job will be used.
- `url=<string>` specifies the URL of the Swift system during a restore. This parameter is optional. If not set, the `url=<string>` parameter from the backup Job will be used.
- `be_object_segment_size=<size>` specifies, in bytes, the size of the segments of a DLO object. The default is 5MB. This parameter is optional.
- `restore_local=<yes or no>` specifies that the files should be restored to a local directory based on the `where=` restore job parameter. This parameter is optional.

## FileSet Examples

In the example below, all objects inside the container `container1` will be backed up.

```
FileSet {
  Name = FS_Swift
  Include {
    Plugin = "swift: user=r1 password=t1 URL=http://swift:8080/auth/v1.0 \
            include=container1/*"
  }
}
```

In the example below, all objects that do not end with `tmp` inside the container `container1` will be backed up.

```
FileSet {
  Name = FS_Swift_without_tmp
  Include {
    Plugin = "swift: user=root password=test1 URL=http://swift/auth/v1.0 \
            include=container1/* exclude=*tmp"
  }
}
```

This example is the same as the exclude one above, but using `regexexclude` instead:

```
FileSet {
  Name = FS_Swift_without_tmp
  Include {
    Plugin = "swift: include=container1/* regexexclude=.*\\.tmp\\Z(?ms)" URL=http://swift/
    ↪auth/v1.0 user=r1 password=p1"
  }
}
```

In the example below, all objects that end with `.pdf` inside the container `container1` will be backed up.

```
FileSet {
  Name = FS_Swift_without_tmp
  Include {
    Plugin = "swift: user=r1 password=t1 URL=http://swift/auth/v1.0 \
    include=container1/* regexinclude=.*\\.pdf\\Z(?ms)"
  }
}
```

In the example below, all objects of all containers will be backed up.

```
FileSet {
  Name = FS_Swift_everything
  Include {
    Options {
      Compression = LZ0
    }
    Plugin = "swift: user=r1 password=t1 URL=http://swift:8080/auth/v1.0"
  }
}
```

In the example below, all objects of all containers will be backed up. The password of the swift user `root` will be taken from the file `/etc/swift.txt` file.

```
FileSet {
  Name = FS_Swift_passfile
  Include {
    Plugin = "swift: user=r1 passfile=/etc/swift.txt URL=http://swift/auth/v1.0"
  }
}
```

## 2.5 Restore Examples

### Restore to a Subdirectory Inside the Container

If the objects were selected inside `container1`, the following restore command will relocate objects under `/tmp`. For example: `/container1/home/file.png` will be restored as `/container1/tmp/home/file.png`.

```
* restore where=/container1/tmp
```

## Restore to an Alternate Container

If the objects were selected inside `container1`, the following restore command will relocate them under `container2`. ex: `/container1/home/file.png` will be restored as `/container2/home/file.png`.

```
* restore where=container2
```

## Restore to a Local Directory

You may restore the files to your local filesystem by setting the `restore_local` option to `yes` in the `Plugin Restore Options` menu. You must specify the `where=` option.

```
* restore where=/home/user/my_restored_files
```

After the selection of the files you want to restore, modify the plugin option, select option 5 `restore_local`, and set it to 'yes'

```
Run Restore job
JobName:          RestoreFiles
...
Plugin Options:  *None*
OK to run? (yes/mod/no): mod
Parameters to modify:
  1: Level
  ...
 13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : swift: user=test:tester password=testing
URL=http://localhost:8080/auth/v1.0 include=data1/*

Plugin Restore Options
user:             *None*                (*None*)
password:         *None*                (*None*)
url:              *None*                (*None*)
be_object_segment_size: *None*          (5 MiB)
restore_local:    *None*                (No)
insecure:         *None*                (No)
Use above plugin configuration? (yes/mod/no): mod
You have the following choices:
  1: user (Restore user name)
  2: password (Restore user password)
  3: url (Destination URL for restore)
  4: be_object_segment_size (DLO max segment size)
  5: restore_local (Restore as local file)
  6: insecure (Accept self-signed certificate)
Select parameter to modify (1-6): 5
Please enter a value for restore_local: yes
Plugin Restore Options
user:             *None*                (*None*)
password:         *None*                (*None*)
url:              *None*                (*None*)
be_object_segment_size: *None*          (5 MiB)
restore_local:    yes                   (No)
```

(continues on next page)

(continued from previous page)

```
insecure:          *None*          (No)
Use above plugin configuration? (yes/mod/no): yes
Run Restore job
JobName:          RestoreFiles
...
Plugin Options:   User specified
OK to run? (yes/mod/no): yes
```

## 2.6 Useful Swift Commands

You may define environment variables to simplify Swift command calls.

```
# export ST_AUTH=http://swift.lan:8080/auth/v1.0
# export ST_KEY=testing
# export ST_USER=test:tester
```

Show information about a Swift account:

```
# swift stat
```

List containers

```
# swift list
```

List objects inside a container

```
# swift list container1
```

Upload a file

```
# swift upload container1 afile
```

Set a metadata for container or object

```
# swift post -m Bacula:Pass container1
# swift post -m Bacula:Pass container1 object1
```

Set an ACL data for container

```
# swift post --read-acl=www container1
# swift post --write-acl=www container1
```

For more information, please refer to: <https://docs.openstack.org/python-swiftclient/latest/cli/index.html#swift-usage>



## 2.7 Limitations

- Objects stored as SLO objects on the Swift System will be restored as DLO objects.
- The modification time of an object cannot be restored. This is a Swift limitation.
- The creation time of an object is always equal to the modification time. This is a Swift limitation.
- Bacula's Accurate backup mode is not supported. You will receive a warning message in this case.
- The regexwhere feature is not supported
- It is recommended to not set a high value for the Plugin Restore Parameter `be_object_segment_size` to limit the memory consumption.