Hadoop Distributed File System Plugin

Bacula Systems Documentation

Contents

Contents

The following article aims at presenting the reader with information about the **Bacula Enterprise Hadoop Distributed File System (HDFS) Plugin**. The document briefly describes the target technology of the plugin, and presents its main features.

Through subchapters, more in-depth information can be found about the following topics:

1 Scope

The HDFS Plugin supports any Bacula Supported Operative System based on Linux.

This plugin is available since **Bacula Enterprise 12.4**, and needs to be deployed in a Linux host.

See also:

- Go to *Features*
- Go to *Architecture*
- Go to *Installation*
- Go to *Configuration*
- Go to *Operations*
- Go to *Limitations*

Go back to the *main HDFS Plugin page*.

2 Features

The main feature of **Bacula Enterprise HDFS Plugin** is to offer backup and restore of any file contained in HDFS Clusters in an efficient way. The technology supports Full, Incremental, and Differential backups, and is able to perform backups with automatic snapshot management. Using the HDFS Plugin ensures protection of the information stored in Hadoop environments.

A unique characteristic of the Plugin is the ability to filter information based on date, which may be quite useful for very large systems, where old information may not be of somebody's interest, and/or where having a backup of everything could be problematic.

In order to increase user comfort, a wide range of backup filters have been incorporated. Moreover, a very useful feature of the Plugin is the ability to restore inside the original or a different HDFS filesystem, as well as to any other non-HDFS filesystem.

Also, the Plugin is integrated with Bweb, which guarantees ease of use.

See the detailed list of HDFS Plugin features:

2.1 Backup Features

- Full/Incremental/Differential backups
- Automatic snapshot management
- Backup filters:
  - Exclude directories with a specific name
  - Exclude files with a pattern
  - Include files with a pattern
  - Include files created/modified after a given time

The configuration for HDFS backups is done in a Bacula FileSet configuration file.

During a backup, the Bacula plugin will contact the Hadoop File System to generate a system Snapshot and retrieve Files one by one. During an Incremental or a Differential backup session, the Bacula File Daemon will need to read the difference between two Snapshots to determine which files should be backed up.

2.2 Restore Features

- Restore to local disk
- Restore to the same HDFS instance
- Restore to a different HDFS instance

See also:

- Go to *HDFS Architecture*
- Go to *HDFS Installation*
- Go to *HDFS Configuration*
- Go to *HDFS Operations*
- Go to *HDFS Limitations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

3 Architecture

The Bacula Enterprise HDFS Plugin uses the Hadoop Java API to access its Distributed File System.

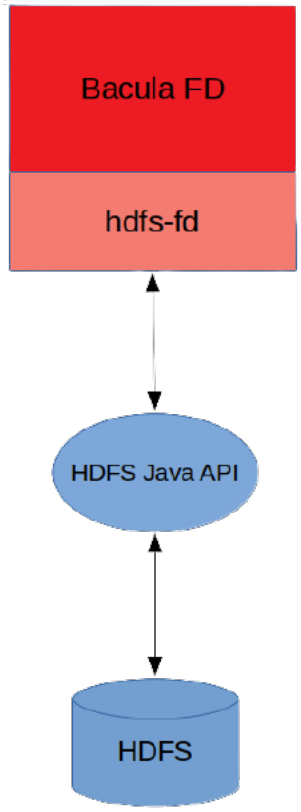

Fig. 1: HDFS Plugin Architecture

See also:

- Go back to *HDFS Features*
- Go to *HDFS Installation*
- Go to *HDFS Configuration*
- Go to *HDFS Operations*
- Go to *HDFS Limitations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

4 Installation

This article describes how to install Bacula Enterprise Hadoop Distributed File System (HDFS) Plugin.

4.1 Prerequisites

- The plugin is based on Java, so the Java version 8 or greater is needed
- Network access from the File Daemon where you install the plugin to the HDFS cluster nodes.

HDFS Installation with BIM

In order to install the HDFS Plugin with BIM, install the File Daemon with BIM and choose to install the HDFS Plugin during the FD installation.

For more details on the plugin installation process with BIM, click here.

See also:

See an alternative way of installing the HDFS Plugin - *HDFS Installation with Package Manager*.

Go back to the *main HDFS Plugin Installation page*.

Go back to the *main HDFS Plugin page*.

HDFS Installation with Package Manager

On the Bacula File Daemon that you want to connect to your HDFS instance, extend the repository file for your package manager to contain a section for the HDFS plugin. For example, in 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 EnterpriseHdfsPlugin]
name=Bacula Enterprise Hdfs Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/hdfs/@version@/rhel7-64/
enabled=1
protect=0
gpgcheck=0
```

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

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

Once the repository is configured for your system, the package *bacula-enterprise-hdfs-plugin* can be installed with `yum install` or `apt-get install`.


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

See also:

See an alternative way of installing the HDFS Plugin - *HDFS Installation with BIM*.

Go back to the *main HDFS Plugin Installation page*.

Go back to the *main HDFS Plugin page*.

See also:

- Go back to *HDFS Features*
- Go back to *HDFS Architecture*
- Go to *HDFS Configuration*
- Go to *HDFS Operations*
- Go to *HDFS Limitations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

5 Configuration

The following chapter presents the information on HDFS Plugin parameters, estimation and backup parameters, and restore parameters.

5.1 Plugin Parameters

The following parameters affect any type of HDFS Plugin Job (Backup, Estimation or Restore).

- `url=<string>` specifies the URL of the HDFS instance. This parameter is mandatory.
- `user=<string>` specifies the User who owns the `root path "/"` in the HDFS instance. Bacula needs to know this user in order to create snapshots in the system. This parameter is mandatory.

5.2 Plugin Estimation and Backup Parameters

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

If none of the optional paramaters `include`, `regexinclude`, `exclude` or `regexexclude` are specified then all files from the Hadoop File System to which the user `bacula` has access will be backed up.

5.3 Plugin Restore Parameters

- `user=<string>` specifies an account where restore will be performed. This parameter is optional. If not set, the `user` parameter from the backup Job will be used.
- `url=<string>` specifies the URL of the HDFS system during a restore. This parameter is optional. If not set, the `url=<string>` parameter from the backup Job will be used.
- `restore_local=<yes or no>` specifies that the files should be restored to a local directory based on the `where=` restore job parameter. This parameter is optional and defaults to no.

FileSet Examples

In the example below, all files inside the path `btest1` will be backed up.

```
FileSet {
 Name = FS_Hdfs
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*"
 }
}
```

In the example below, all files that do not end with `tmp` inside the path `btest1` will be backed up.

```
FileSet {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/* exclude=*tmp"
 }
}
```

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

```
FileSet {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/* regexclude=.*\\.
↪tmp\\Z(?ms)"
 }
}
```

In the example below, all files that end with `.pdf` inside the path `path1` will be backed up.

```
FileSet {
 Name = FS_Hdfs_without_tmp
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/* regexinclude=.*\
↪\.pdf\\Z(?ms)"
 }
}
```

In the example below, all files will be backed up.


```
FileSet {
  Name = FS_Hdfs_everything
  Include {
    Options {
      Compression = LZO
    }
    Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000"
  }
}
```

See also:

- Go back to *HDFS Features*
- Go back to *HDFS Architecture*
- Go back to *HDFS Installation*
- Go to *HDFS Operations*
- Go to *HDFS Limitations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

6 Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise HDFS Plugin**.

6.1 Backup

Assuming that we have the following Job configured in `bacula-dir.conf`:

```
JobDefs {
  Name = BackupJob
  Type = Backup
  Pool = Default
  Storage = File
  Messages = Standard
  Priority = 10
  Client=127.0.0.1-fd
  Write Bootstrap = "/home/hdev/bacula-cloud/regress/working/%n-%f.bsr"
}

Job {
  Name = PluginHdfsTest
  JobDefs = BackupJob
  FileSet= FS_Hdfs
}

FileSet {
 Name = FS_Hdfs
```

(continues on next page)

(continued from previous page)

```
 Include {
  Plugin = "hdfs: user=hadoop URL=hdfs://localhost:9000 include=btest1/*"
 }
}
```

We can run this Job using the bconsole program:

```
run job=PluginHdfsTest
Using Catalog "MyCatalog"
Run Backup job
JobName:  PluginHdfsTest
Level:    Full
Client:   127.0.0.1-fd
FileSet:  TestPluginHdfsSet
Pool:     Default (From Job resource)
Storage:  File (From Job resource)
When:     2020-04-06 12:19:10
Priority: 10
OK to run? (yes/mod/no): yes
Job queued. JobId=1
wait
You have messages.
messages
06-abr 12:29 127.0.0.1-dir JobId 1: Start Backup JobId 1, Job=PluginHdfsTest.2020-04-06_
↪12.29.12_05
06-abr 12:29 127.0.0.1-dir JobId 1: Using Device "FileStorage" to write.
06-abr 12:29 127.0.0.1-sd JobId 1: Wrote label to prelabeled Volume "TestVolume001" on␣
↪File device "FileStorage" (/home/hdev/bacula-cloud/regress/tmp)
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Starting HDFS Plugin Job
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Finished reading HDFS Plugin Params
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Starting backup
06-abr 12:29 127.0.0.1-fd JobId 1: hdfs: Finishing HDFS Plugin Job
06-abr 12:29 127.0.0.1-sd JobId 1: Elapsed time=00:00:01, Transfer rate=3.157 K Bytes/
↪second
06-abr 12:29 127.0.0.1-sd JobId 1: Sending spooled attrs to the Director. Despooling 3,
↪581 bytes ...
06-abr 12:29 127.0.0.1-dir JobId 1: Bacula 127.0.0.1-dir 12.4.0 (20Dec19):
  Build OS:               x86_64-pc-linux-gnu ubuntu 18.04
  JobId:                  1
  Job:                    PluginHdfsTest.2020-04-06_12.29.12_05
  Backup Level:           Full
  Client:                 "127.0.0.1-fd" 12.4.0 (20Dec19) x86_64-pc-linux-gnu,ubuntu,18.
↪04
  FileSet:                "TestPluginHdfsSet" 2020-04-06 12:29:10
  Pool:                   "Default" (From Job resource)
  Catalog:                "MyCatalog" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         06-abr-2020 12:29:12
  Start time:             06-abr-2020 12:29:14
  End time:               06-abr-2020 12:29:17
  Elapsed time:           3 secs
  Priority:               10
```

(continues on next page)

(continued from previous page)

```
  FD Files Written:       13
  SD Files Written:       13
  FD Bytes Written:       60 (60 B)
  SD Bytes Written:       3,157 (3.157 KB)
  Rate:                   0.0 KB/s
  Software Compression:   None
  Comm Line Compression:  9.8% 1.1:1
  Snapshot/VSS:           no
  Encryption:             no
  Accurate:               no
  Volume name(s):         TestVolume001
  Volume Session Id:      1
  Volume Session Time:    1586186949
  Last Volume Bytes:      4,586 (4.586 KB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK

06-abr 12:29 127.0.0.1-dir JobId 1: Begin pruning Jobs older than 6 months .
06-abr 12:29 127.0.0.1-dir JobId 1: No Jobs found to prune.
06-abr 12:29 127.0.0.1-dir JobId 1: Begin pruning Files.
06-abr 12:29 127.0.0.1-dir JobId 1: No Files found to prune.
06-abr 12:29 127.0.0.1-dir JobId 1: End auto prune.

list files jobid=1
Using Catalog "MyCatalog"
+-------------------------------------+
| filename                            |
+-------------------------------------+
| /@hdfs/btest1/files/A/A0A/bFa5csqF |
| /@hdfs/btest1/files/A/A0A/bEa4csqE |
| /@hdfs/btest1/files/A/A0A/bDa3csqD |
| /@hdfs/btest1/files/A/A0A/bCa2csqC |
| /@hdfs/btest1/files/A/A0A/bBa1csqB |
| /@hdfs/btest1/files/C/aCa2aaaC     |
| /@hdfs/btest1/files/F/aFa5aaaF     |
| /@hdfs/btest1/files/B/aBa1aaaB     |
| /@hdfs/btest1/files/bAa0csqA       |
| /@hdfs/btest1/files/E/aEa4aaaE     |
| /@hdfs/btest1/files/A/aAa0aaaA     |
| /@hdfs/btest1/files/D/aDa3aaaD     |
+-------------------------------------+

+-------+----------------+---------------------+------+-------+----------+----------+----
↪-------+
| jobid | name           | starttime           | type | level | jobfiles | jobbytes |␣
↪jobstatus |
+-------+----------------+---------------------+------+-------+----------+----------+----
↪-------+
|     1 | PluginHdfsTest | 2020-04-06 12:29:14 | B    | F     |       13 |       60 | T ␣
↪        |
```

(continues on next page)

(continued from previous page)

```
+-------+---------------+---------------------+------+-------+----------+----------+----
↪-------+
quit
```

See also:

- Go to *Restore*
- Go to *Useful Commands*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

6.2 Restore

To restore the backup performed in the last section into the same HDFS System, inside the directory `bacula-restores` we also use `bconsole`:

```
restore jobid=1 where=/bacula-restores all done yes
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
12 files inserted into the tree and marked for extraction.
Bootstrap records written to /home/hdev/bacula-cloud/regress/working/127.0.0.1-dir.
↪restore.1.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    TestVolume001             File                      FileStorage

Volumes marked with "*" are in the Autochanger.


12 files selected to be restored.

Automatically selected Client: 127.0.0.1-fd
Using Catalog "MyCatalog"
Job queued. JobId=5
wait
You have messages.
messages
06-abr 12:29 127.0.0.1-dir JobId 5: Start Restore Job RestoreFiles.2020-04-06_12.29.39_16
06-abr 12:29 127.0.0.1-dir JobId 5: Restoring files from JobId(s) 1
06-abr 12:29 127.0.0.1-dir JobId 5: Using Device "FileStorage" to read.
06-abr 12:29 127.0.0.1-sd JobId 5: Ready to read from volume "TestVolume001" on File␣
↪device "FileStorage" (/home/hdev/bacula-cloud/regress/tmp).
06-abr 12:29 127.0.0.1-sd JobId 5: Forward spacing Volume "TestVolume001" to addr=239
06-abr 12:29 127.0.0.1-sd JobId 5: Elapsed time=00:00:01, Transfer rate=2.142 K Bytes/
↪second
```

(continues on next page)

(continued from previous page)

```
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Starting HDFS Plugin Job
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Finished reading HDFS Plugin Params
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Starting restore
06-abr 12:29 127.0.0.1-fd JobId 5: hdfs: Finishing HDFS Plugin Job
06-abr 12:29 127.0.0.1-dir JobId 5: Bacula 127.0.0.1-dir 12.4.0 (20Dec19):
  Build OS:               x86_64-pc-linux-gnu ubuntu 18.04
  JobId:                  5
  Job:                    RestoreFiles.2020-04-06_12.29.39_16
  Restore Client:         127.0.0.1-fd
  Where:                  /bacula-restores
  Replace:                Always
  Start time:             06-abr-2020 12:29:41
  End time:               06-abr-2020 12:29:45
  Elapsed time:           4 secs
  Files Expected:         12
  Files Restored:         12
  Bytes Restored:         60 (60 B)
  Rate:                   0.0 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK

06-abr 12:29 127.0.0.1-dir JobId 5: Begin pruning Jobs older than 6 months .
06-abr 12:29 127.0.0.1-dir JobId 5: No Jobs found to prune.
06-abr 12:29 127.0.0.1-dir JobId 5: Begin pruning Files.
06-abr 12:29 127.0.0.1-dir JobId 5: No Files found to prune.
06-abr 12:29 127.0.0.1-dir JobId 5: End auto prune.
```

See also:

- Go back to *Backup*
- Go to *Useful Commands*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

6.3 Useful Commands

List files inside a directory

```
# hadoop fs -ls /user/hadoop/file1
```

Upload a file

```
# hadoop fs -put /local-files/file1.txt /hdfs-path
```

Upload many files

```
# hadoop fs -put /local-files /hdfs-path
```

Download many files


```
# hadoop fs -get /hdfs-path /local-path
```

For a complete set of commands and options, refer to the Hadoop documentation:

- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html
- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsSnapshots.html

See also:

- Go back to *Backup*
- Go back to *Restore*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

See also:

- Go back to *HDFS Features*
- Go back to *HDFS Architecture*
- Go back to *HDFS Installation*
- Go back to *HDFS Configuration*
- Go to *HDFS Limitations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

7 Limitations

The following article presents limitations of HDFS Plugin.

- The HDFS plugin requires snapshot enabled in the HDFS file system, available from Hadoop 3.3.1. If you cannot enable snapshot, please use fuse and mount the HDFS locally on the system running the FD and the SD.
- The creation time of a file cannot be backed up.
- Empty directories and directory attributes cannot be backed up.
- Bacula's Accurate backup mode is not supported. You will receive a warning message if it is applied.
- The current implementation of the plugin cannot backup ACL and Extended Attributes.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

See also:

- Go back to *HDFS Features*
- Go back to *HDFS Architecture*
- Go back to *HDFS Installation*
- Go back to *HDFS Configuration*
- Go back to *HDFS Operations*

Go back to *the main HDFS Plugin page*.

Go back to the main Dedicated Backup Solution page.

