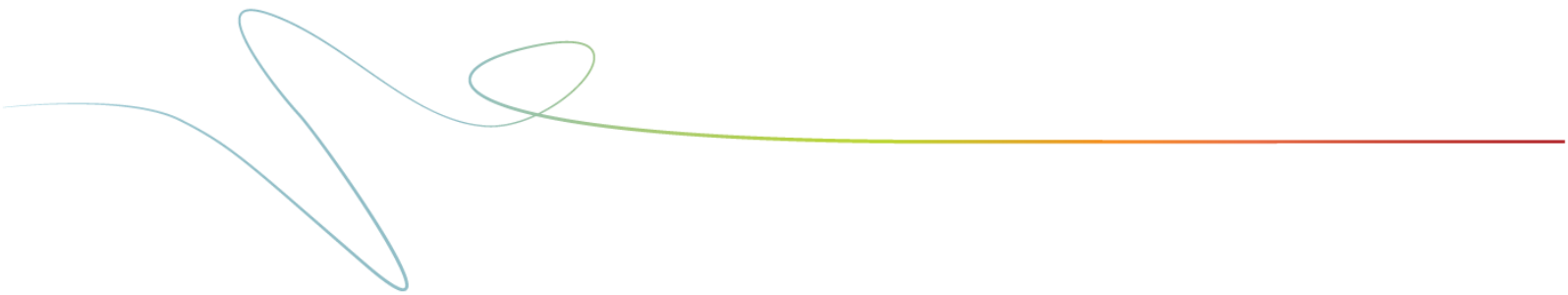
VERITAS The truth in information.

Protecting HDFS File System and Hive Metastore Database using Veritas NetBackup

Solution Paper

NetBackup Version 8.1.00 and later



Contents

About this document

This document provides information of the NetBackup solution for protecting HDFS File System and Hive Metastore database.

The document contains instructions for deployments using NetBackup version 8.1 or later.

Protecting HDFS File System and Hive metastore database

HDFS File System and Hive metastore deployment can be protected using Veritas NetBackup Hadoop Plug-in (Parallel Streaming Framework Plug-in).

BigData policy is used along with **ApplicationType** as **hadoop** to protect Hadoop data.

Hive metastore can be created in the following databases:

- PostgreSQL
- MySQL
- Oracle
- Maria DB

Note: This document discusses the solution with PostgreSQL for Cloudera and My SQL for Hortonworks as examples.

Protection strategy

The protection strategy comprises of the following stages:

1. Stop the Hive database or services.
2. Export Hive metastore database dump in flat file and copy it to the HDFS file system.
3. Take the HDFS file system backup using the PSF Hadoop plug-in.

Restore strategy

The restore comprises of the following stages:

1. Stop the Hive database or services.
2. Restore Hive metastore database dump file from the backup copy to HDFS file system and then copy from HDFS file system to local file system.
3. Import the Hive metastore database dump from the flat file system that is recovered above.

Supported configuration

The following configurations are supported:

1. Hortonworks Data Platform (HDP®) 2.6.5
    a. Hadoop 2.7.3
    b. Hive 1.2.1000
2. Cloudera 5.8.5
    a. Hadoop 2.6.0-cdh5.8.5
    b. Hive 1.1.0-cdh5.8.5

Hive Metastore backup and restore for Cloudera Hadoop deployment

To back up a PostgreSQL database, the procedure is same for embedded or external database.

To back up a Hive metastore with PostgreSQL databases

1. Log on to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from:
   `/etc/cloudera-scm-server/db.properties`
   `com.cloudera.cmf.db.name=scm`
   `com.cloudera.cmf.db.user=scm`
   `com.cloudera.cmf.db.password=NnYfWIjlbk`
3. Run the following command as root using the parameters from the preceding step:
   `# pg_dump -h hostname -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +%Y%m%d)`
4. Enter the password from the `com.cloudera.cmf.db.password` property in step 2.
5. Run the following command on the local host as the **role user**, to back up a database that is created for one of the roles that are described in Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server,
   `# pg_dump -h hostname -p 7432 -U roleuser > /tmp/roledb`
6. Enter the password that is specified when the database was created.
7. Copy the database dump file to Hadoop files system (HDFS) to include in the backup.

To restore a Hive metastore with PostgreSQL databases

1. Log on to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from:
   `/etc/cloudera-scm-server/db.properties`
   `com.cloudera.cmf.db.name=scm`
   `com.cloudera.cmf.db.user=scm`

```
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root, to drop or create new hive database.

```
psql -c "drop database <dbname>;"

To set/alter password of DB
postgres=# \password hive
Enter new password:
Enter it again:

postgres=# alter database hive owner to hive;
ALTER DATABASE
To list the newly create database
postgres=# \list
```

4. Run the following command, to restore DB dump

```
# psql -h <hostname> -p  <Port> -U <hive-db-user> -d <hive-DB-Name>
-f <path of db dump file to be imported>
e.g.
# psql -h localhost -p  7432 -U hive -d hive -f
/tmp/all.20181001.sql
```

5. Enter the password that is specified when the database was created.
6. Start the Hive database or services.

Hive Metastore backup and restore for Hortonworks Hadoop deployment

For Hortonworks Hadoop deployment with MySQL, consider the following

- Back up the Hive Metastore in MySQL periodically.
- Restore Hive metastore from the backup in the new MySQL database.
- Move the Hive services to a new node.

Note: One can drop the existing hive database and recreate on same node. Next, import the database dump into newly created hive database and start the hive services.

To back up a Hive metastore with MySQL databases

1. Stop the Hive service.
   This prevents metadata updates during the backup and avoids data inconsistency.
   In Ambari, go to **Services > Hive > Service Actions > Stop**.
   The MySQL, HiveMetastore, HiveServer2, and WebHCat stop and the services are temporarily unavailable during this process.
2. Dump Hive Metastore database (MySQL) to a local file.
   By default, Hive Metastore, HiveServer2, and MySQL Server are deployed on the same node.
   a. Identify the node that is running MySQL.
      In Ambari, go to **Services > Hive > Summary > MySQL Server**.
   b. Log on to the node as **root**. The default database name is **hive**.
   c. Determine the database.
      In Ambari, go to **Services > HiveConfigs > Advanced > Hive Metastore > Database Name**.
      MySQL
      `# mysqldump hive > /tmp/hive_backup.sql`
   d. Verify if the dump is successful. Run the following command:
      `cat /tmp/hive_backup.sql`
      The raw content is displayed.
   e. Store the file on a different node, or system with a timestamp.
3. After the backup is completed, start the Hive services again.

   In Ambari, go to **Services > Hive > Service Actions > Start**.

   All the Hive services, including MySQL, HiveMetastore, HiveServer2 and WebHCat are started.

To restore a Hive metastore with MySQL databases

1. Move MySQL server to a new node.
   a. In Ambari, go to **Services > Hive > Service Actions > Move MySQL Server**.
      Select the host and follow the wizard until you reach the manual command step.
   b. For restoring a database, you do not need to run **mysqldump** on the previous system.
   c. Upload the **hive_backup.sql** to the node where MySQL server is running.
      i. Log on to the new MySQL server node as **root**.
      ii. Launch **mysql**.
      iii. Execute `CREATE DATABASE hive` command.
      iv. Exit mysql with `exit` command.
      v. Import the database using the following command:
         `mysql hive < /tmp/hive_backup.sql`
      vi. Proceed to complete the steps in the wizard.
2. Move Hive Metastore server to a new node.

a. In Ambari, go to **Services > Hive > Service Actions > Move Hive Metastore**. Select the host and follow the wizard until you reach the manual command step. Before deploying, ensure that you have run the following command on the Ambari server.
```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql
```
Replace **/path/to/mysql** with the actual path to the mysql connector JAR file.
On RHEL, the default path is **/usr/share/java/mysql-connector-java.jar**. If **mysql-connector-java.jar** does not exist on the Ambari server node, install it by running the command:
```
yum install mysql-connector-java*
```
If the Ambari server is already set up with the jdbc-driver location, ignore this step.

b. Restart all required services that are marked by the Ambari UI. Moving these servers may have affected some HDFS configurations.

3. Move HiveServer2 server to a new node

a. In Ambari, go to **Services > Hive > Service Actions > Move Hive Metastore2**. Select the host and follow the wizard until you reach the manual command step. Before deploying, ensure that you have run the following command on the Ambari server.
```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql
```
Replace **/path/to/mysql** with the actual path to the mysql connector JAR file.
On RHEL, the default path is **/usr/share/java/mysql-connector-java.jar**. If **mysql-connector-java.jar** does not exist on the Ambari server node, install it by running the command:
```
yum install mysql-connector-java*
```
If the Ambari server is already set up with the jdbc-driver location, ignore this step.

b. Restart all required services that are marked by the Ambari UI. Moving these servers may have affected some HDFS configurations.

4. Move WebHCat to a new node

a. In Ambari, go to **Services > Hive > Service Actions > Move WebHCat Server**. Select the host and follow the wizard until you reach the manual command step. Before deploying, ensure that you have run the following command on the Ambari server.
```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql
```
Replace **/path/to/mysql** with the actual path to the mysql connector JAR file.
On RHEL, the default path is **/usr/share/java/mysql-connector-java.jar**. If **mysql-connector-java.jar** does not exist on the Ambari server node, install it by running the command:
```
yum install mysql-connector-java*
```
If the Ambari server is already set up with the jdbc-driver location, ignore this step.

b. Restart all required services that are marked by the Ambari UI. Moving these servers may have affected some HDFS configurations.

References

Refer to following documentation for additional information:

- Documentation for PostgresSQL
- Documentation for MySQL
- Veritas NetBackup for Hadoop Administrator's Guide