S3 Backups

[Introduction](#h.rmxeie4miiwy)

[Environment Setup](#h.y9ln95p98bj)

[Recommended Configuration Step](#h.gc2kx7152pph)

[Virtualenv and AWS Command Line Interface](#h.b1s8zz1i4vor)

[Setup AWS Credentials](#h.gnxrs0w746xm)

[Running AWS CLI Backup Commands Manually](#h.n7i98ql6esn5)

[List Backups](#h.xo1ws8voc6ce)

[Setting up Crontab](#h.nkjg8mqpolda)

[Log Files](#h.ne6ommbn1hfe)

[Application Logic](#h.qdx1mi831seg)

[Backup Component Details](#h.j68k58ewtzod)

[PostgreSQL Database](#h.6bq2kid1j825)

[KerberosKDC](#h.9d37gjumau0k)

[/etc Directory](#h.2dbaxpxum98h)

[NameNode](#h.9bctqxf9clwn)

[NiFi](#h.hhosqcf0d4p1)

[Hue](#h.1fwx71ujb1it)

# Introduction

Backups of critical system level files, Cloudera configurations, and related databases needed to restore Cloudera services will be performed via a script that will run nightly and backup to an Amazon AWS S3 Bucket called s3://YourCompanyDomain-bia-backups

# Environment Setup

Backups to AWS S3 will use AWS Command Line Interface. This is installed in an isolated python environment to prevent general access to the module. Virtualenv is installed, then the awscli module is installed. Either this must all be configured manually, or if the AWS credentials are created manually (recommended) on the server then the backup script will take care of the next steps.

## Recommended Configuration Step

Copy the **/root/.aws** directory from a previously configured system as **root** to the new servers in the environment.

## Virtualenv and AWS Command Line Interface

Virtualenv is used to isolate the python instance and modules from general usage by normal users. The virtual environment for python is configured using the following.

The backups to AWS S3 will use the AWS Command Line Interface. The AWS CLI was installed into the above Virtualenv environment by installing the module after sourcing the isolated python environment.

This is performed as part of the backup script in the situation that it has not already been installed using the following steps.

|  |
| --- |
| virtualenv /opt/backup\_venv  source /opt/backup\_venv/bin/activate  pip install awscli |

The online reference to the AWS CLI can be locate at:

<http://docs.aws.amazon.com/cli/latest/reference/>

The information related to S3 can be found under the **S3** section -

<http://docs.aws.amazon.com/cli/latest/reference/s3/index.html>

## Setup AWS Credentials

These steps presume that the AWS CLI has been installed manually from the previous step (or it has been run from the cronjob and that has failed since the credentials are not in place). Run the **aws configure** command to setup the credentials for accessing AWS CLI. This means that the keys are not needed to be passed on the command line each time the CLI command is run. The configuration files are stored in /root/.aws under the **root** user’s account.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# /opt/backup\_venv/bin/aws configure  AWS Access Key ID [None]: XXXXXXXXXXX  AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXX  Default region name [None]:  Default output format [None]: |

Alternately, the same command can be run as follows on a single line, replacing the AWS keys as shown below:

|  |
| --- |
| echo -e "<AWS\_ACCESS\_KEY\_ID>\n<AWS\_SECRET\_ACCESS\_KEY>\n\n\n" | /opt/backup\_venv/bin/aws configure |

This step has to be completed prior to running the Backup script. If it has not been performed, the PagerDuty error as follows will be generated:

|  |
| --- |
| [ERROR]: AWS CLI Credentials Not Available - Please Run /opt/backup\_venv/bin/aws configure |

Either run the **aws configure** command or copy the **/root/.aws** directory from another node in the same cluster.

# Running AWS CLI Backup Commands Manually

The list of commands related to S3 can be found under the **S3** section -

<http://docs.aws.amazon.com/cli/latest/reference/s3/index.html>

## List Backups

To list the backup environments available:

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **/opt/backup\_venv/bin/aws s3 ls s3://YourCompanyDomain-bia-backups**  PRE cdh-prod-hq/  PRE cdh-rd1-hq/  PRE cdh-staging-hq/ |

To list the backup days available:

|  |
| --- |
| [root@bos-staging-cdh-master1 ~]# **/opt/backup\_venv/bin/aws s3 ls s3://YourCompanyDomain-bia-backups/cdh-rd1-hq/**  PRE 20160318/  PRE 20160321/  PRE 20160322/ |

To list the servers being backed up for a specific day in an environment:

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **/opt/backup\_venv/bin/aws s3 ls s3://YourCompanyDomain-bia-backups/cdh-rd1-hq/20160321/**  PRE bos-rd1-cdh-data1.rd1.hq.YourCompanyDomain.com/  PRE bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com/  PRE bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/ |

To list all the backup files for a specific day we can use the **--recursive** option:

|  |
| --- |
| [root@bos-staging-cdh-master1 ~]# **/opt/backup\_venv/bin/aws s3 ls s3://YourCompanyDomain-bia-backups/cdh-rd1-hq/20160321/ --recursive**  2016-03-22 13:54:14 7268738 cdh-rd1-hq/20160321/bos-rd1-cdh-data1.rd1.hq.YourCompanyDomain.com/etc.tar.gz  2016-03-22 13:54:14 7273704 cdh-rd1-hq/20160321/bos-rd1-cdh-data2.rd1.hq.YourCompanyDomain.com/etc.tar.gz  2016-03-22 13:54:14 7271938 cdh-rd1-hq/20160321/bos-rd1-cdh-data3.rd1.hq.YourCompanyDomain.com/etc.tar.gz  2016-03-22 13:54:14 14306 cdh-rd1-hq/20160321/bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com/Hue.tar.gz  2016-03-22 13:54:14 487766 cdh-rd1-hq/20160321/bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com/NiFi.tar.gz  2016-03-22 13:54:14 7299029 cdh-rd1-hq/20160321/bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com/etc.tar.gz  2016-03-22 13:54:14 14252 cdh-rd1-hq/20160321/bos-rd1-cdh-edge2.rd1.hq.YourCompanyDomain.com/Hue.tar.gz  2016-03-22 13:54:14 7280451 cdh-rd1-hq/20160321/bos-rd1-cdh-edge2.rd1.hq.YourCompanyDomain.com/etc.tar.gz  2016-03-22 13:54:14 18897 cdh-rd1-hq/20160321/bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/KerberosKDC.tar.gz  2016-03-22 13:54:14 2715 cdh-rd1-hq/20160321/bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/KerberosPrincipals.txt  2016-03-22 13:54:14 2959294 cdh-rd1-hq/20160321/bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/NameNode.tar.gz  2016-03-22 13:54:14 40804605 cdh-rd1-hq/20160321/bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/PostgreSQL\_DB\_20160321.sql.gz  2016-03-22 13:54:15 7349632 cdh-rd1-hq/20160321/bos-rd1-cdh-master1.rd1.hq.YourCompanyDomain.com/etc.tar.gz |

# Setting up Crontab

To run the job, we will enable this to run as the **root** user as a cron job. This will run daily at 12:30am. The script is located in **/usr/local/bin/BackupClouderaConfigs.sh**

The cronjob entries for **root** are stored in a file called /root/cronfile until it is loaded into crontab. This is useful to have a backup available. If you need to recreate the file, you can run the following. Always check the current crontab entries first before editing the file to ensure you don’t override someone else's changes. You can check crontab using **crontab -l**

|  |
| --- |
| crontab -l > cronfile |

Below shows the temporary **cronfile**.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **pwd**  /root  [root@bos-rd1-cdh-edge1 ~]# **ls -l cronfile**  -rw-r--r-- 1 root root 68 Mar 21 11:32 cronfile |

The following shows the contents of the temporary **cronfile**, followed by loading this into crontab and then listing what is running in crontab.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **cat cronfile**  30 0 \* \* \* /usr/local/bin/BackupClouderaConfigs.sh > /dev/null 2>&1  [root@bos-rd1-cdh-edge1 ~]# **crontab cronfile**  [root@bos-rd1-cdh-edge1 ~]# **crontab -l**  30 0 \* \* \* /usr/local/bin/BackupClouderaConfigs.sh > /dev/null 2>&1 |

We can validate that the job ran successfully by looking at the log file in the **root** user’s home directory for the script. The log file is **/root/BackupClouderaConfigs.sh.log**. We can check the end of the log file to see that it completed successfully.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# **tail -10 BackupClouderaConfigs.sh.log**  /home/nifi/nifi-0.4.0/../hadoop\_bi/nifi/  /home/nifi/nifi-0.4.0/../hadoop\_bi/nifi/bootstrap.conf  /home/nifi/nifi-0.4.0/../hadoop\_bi/nifi/nifi.properties  /home/nifi/nifi-0.4.0/../hadoop\_bi/nifi/flow.xml.gz  /home/nifi/nifi-0.4.0/../nifi.keytab  CMD=/opt/backup\_venv/bin/aws s3 cp "/tmp/BackupStaging/20160321/NiFi.tar.gz" s3://YourCompanyDomain-bia-backups/20160321/bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com/NiFi.tar.gz  List S3 Backup Destination Files 's3://YourCompanyDomain-bia-backups/20160321/bos-rd1-cdh-edge1.rd1.hq.YourCompanyDomain.com'  Cleaning up Temp Backup Staging Area '/tmp/BackupStaging/20160321'  BackupClouderaConfigs.sh Finished |

# Log Files

As the script is running, a copy of stdout is being logged into **/tmp/<script\_name>.<pid>**

To review the progress of the running job, you can run a **tail -f** on this file to see what is happening. On failure of the job, or completion, the temporary log file is then copied into the user’s home directory and renamed **<script\_name>.log**. For example, to review if the script completed successfully, the last line will have **<script\_name> Finished** as shown below.

|  |
| --- |
| [root@bos-rd1-cdh-edge1 ~]# tail -10 <script\_name>.log  <script\_name> Finished |

Any other condition should have generated an email to log an event into PagerDuty.

# Application Logic

The Application Logic is as follows, as performed as part of the backup process.

* Checks if the user is **root** or exits in error
* Setup AWS CLI
  + Sets up virtualenv under /opt/backup\_venv and installs awscli using pip
* Checks if /root/.aws credentials exist or exits in error
* Creates temp staging area in /tmp/BackupStaging/<YYYYMMDD>
* Backup Core Components if Installed
  + PostgreSQL
  + Kerberos KDC
  + /etc
  + NameNode
  + NiFi
  + Hue
* List S3 Backup Destination
* Cleanup the temp staging area in /tmp BackupStaging/<YYYYMMDD> on error or on completion.

# Backup Component Details

The following are the list of components in the cluster that will be backed up. Within each component, details of what is backed up for each are specified. The components being backed up are as follows:

* PostgreSQL Database
* KerberosKDC
* /etc Directory
* NameNode
* NiFi
* Hue

The same script is run on all nodes in the cluster. Only those components running on each node will be backed up based on the discovery process for each component.

## PostgreSQL Database

The following steps are taken during the PostgreSQL backup:

* PostgreSQL is determined to be installed on a server if the **postgres** user exists in the **/etc/passwd** file on a server.
* Next, a local **/var/lib/pgsql/PostgreSQL\_Backups** directory is created (if it doesn’t exist) to store a local daily backup of the database (for 7 days).
* **pg\_dumpall** is run as the **postgres** user to dump the database into the **/var/lib/pgsql/PostgreSQL\_Backups** directory.
* Any files older than 7 days in /var/lib/pgsql/PostgreSQL\_Backups are removed.
* The currently backup file is then compressed using gzip
* The backup file is uploaded to S3

Backup File Names:

* PostgreSQL\_DB\_<YYYYMMDD>.sql.gz

## KerberosKDC

The following steps are performed during the Kerberos KDC backup:

* Perform an **rpm -qa | grep krb5-server** to determine if the KRB5 Server software is installed on this server
* Check if **/var/kerberos/krb5kdc** directory exists. If not then the backup script will exit with a failure (krb5-server should not be installed if it is not being used).
* gzip/tar **/var/kerberos/krb5kdc** and **/etc/krb5.conf**. Only need to backup /etc/krb5.conf files on the KDC servers as the same copy should be running on all the nodes.
* Copy the KerberosKDC.tar.gz file to S3
* Run **kadmin.local -q "listprincs"** and dump the list of principals from the KDC
* Copy the resulting KerberosPrincipals.txt file to S3

Backup File Names:

* KerberosKDC.tar.gz
* KerberosPrincipals.txt

## /etc Directory

The following steps are run on all servers:

* gzip/tar the /etc/directory
* Upload the etc.tar.gz file to S3

Backup File Names:

* etc.tar.gz

## NameNode

The following steps are performed on the NameNodes:

* Check if the **/storage/cdh/dfs/nn** directory exists to determine if this is the NameNode server.
* gzip/tar the directory excluding any files starting with **edits\_\***. Only complete fsimage files (and other related files) will be backed up for restoring. This is only a daily backup image. The Checkpoint server contains an hourly copy of the fsimage throughout the day.
* Copy the resulting file NameNode.tar.gz to S3

Backup File Names:

* NameNode.tar.gz

## NiFi

The following steps are performed for backing up NiFi since it is not a normal part of the installation of Cloudera.

* Check if the **/home/nifi/nifi-0.4.0** directory exists to determine if NiFi is installed on this server.
* gzip/tar the following directories/files:
  + /home/nifi/nifi-0.4.0/conf
  + /home/nifi/hadoop\_bi
  + /home/nifi/nifi.keytab
* Copy the NiFi.tar.gz file to S3

Backup File Names:

* NiFi.tar.gz

## Hue

The following steps are performed for backing up Hue. This will look for the latest running configuration version and backup this directory.

* Check if there is a valid (and latest) version of the **/run/cloudera-scm-agent/process/\*-hue-HUE\_SERVER** directory exists to determine if Hue is installed on this server.
* gzip/tar the latest /run/cloudera-scm-agent/process/\*-hue-HUE\_SERVER directory.
* Copy the Hue.tar.gz file to S3

Backup File Names:

* Hue.tar.gz