1 HDFS

This document shows information about different HDFS configurations, upgrades and some basic commands.

1.1 Upgrade HDFS

This section shows the steps to upgrade HDFS to a new version, or change any configuration in a secure way.

To facilitate the compression of this guide, I am going to use an example of 2 namenodes:

Namenode1: Active node

Namenode2: Standby node

1.1.1 All Namenodes

1. Prepare the namenodes for the upgrade and create a rollback image in case the HDFS cluster doesn't woks correctly after upgrade we can go back.

```
$ hdfs dfsadmin -rollingUpgrade prepare
```

2. Check the rollback image state:

```
$ hdfs dfsadmin -rollingUpgrade query
```

1.1.2 Namenode 2

1. We are going to start the upgrade with the "Standby" namenode:
2. Stop namenode.

```
$ hadoop-daemon.sh stop namenode
```

3. Update HDFS version or the configuration.
4. Run Namenode2 on standby mode using "-rollingUpgrade started":

```
$ hdfs namenode -rollingUpgrade started
```

1.1.3 Namenode 1

1. Start the failover from Namenode1 to Namenode2 (nn2 is going to be active and nn1 is going to be in standby)

```
$ hdfs haadmin -failover namenode1 namenode2
```

2. Check the failover

```
$ hdfs haadmin -getServiceState namenode1
$ hdfs haadmin -getServiceState namenode2
```

3. Stop namenode.

```
$ hadoop-daemon.sh stop namenode
```

4. Update HDFS version or the configuration
5. Run Namenode1 on standby mode using “-rollingUpgrade started”:

```
$ hdfs namenode -rollingUpgrade started
```

1.1.4 All Datanodes

The datanodes upgrade will be done one by one or in small groups, to avoid losing data.

1. Shutdown datanodes on upgrade mode.

```
$ hdfs dfsadmin -shutdownDatanode datanode1:50020 upgrade
```

2. Check the datanode is turned off.

```
$ hdfs dfsadmin -getDatanodeInfo datanode1:50020
```

3. Update HDFS version or the configuration.

1.1.5 Namenodes (only in one)

1. Terminate the rolling upgrade

```
$ Ctrl +C
$ hdfs dfsadmin -rollingUpgrade finalize
```

2. Start namenodes

```
$ hadoop-daemon.sh start namenode
```

1.1.6 Datanodes

1. Run datanodes

```
$ hadoop-daemon.sh start datanode
```

1.2 HDFS basic commands

This section shows some basic command to interact with HDFS terminal.

- Hadoop help

```
$ hadoop fs
```

- Copy the file “ficheroLocal.txt” to the HDFS system where the file is going to be called “ficheroHDFS.txt”.

```
$ hadoop fs -put ficheroLocal.txt ficheroHDFS.txt
```

- Copy a file from HDFS to the local file system.

```
$ hadoop fs -get ficheroHDFS.txt ficheroLocal.txt
```

- List the content of the HDFS “pruebaCarpeta” directory.

```
$ hadoop fs –ls pruebaCarpeta
```

- Show the content of an HDFS file.

```
$ hadoop fs -cat ficheroHDFS.txt
```

- Crear un directorio en HDFS

```
$ hadoop fs -mkdir Directorio
```

- Borrar un directorio y todo su contenido:

```
$ hadoop fs -rm -r Directorio
```

1.3 Scale HDFS

This section explained how to add new agents to the HDFS cluster that is actually deployed and running, without having any downtime.

Here it is explains how to scale HDFS when it's running in local servers and how to scale when it's running in the cloud "AWS".

1.3.1 Scale HDFS locally

1. The cluster nodes and the new nodes we are going to add, they have to know each other by hostname (in case it is actually configured, go to the step 2).
   - Deploy **hostnames** playbook**.**

```
$ make hostnames
```

2. Besides knowing each other, they also have to have SSH Access without password each other (in case it is actually configured, go to the step 3).
   - Deploy **ssh-keys** playbook

```
$ make ssh-keys
```

3. Scale HDFS cluster.
   - Add the new IPs to the "inventory" of the **hdfs** playbook (keeping the actual cluster nodes IPs).
   - Deploy **hdfs** playbook.

```
$ make hdfs
```

1.3.2 Scale HDFS on AWS cloud with Terraform and Ansible

1. The first thing we have to do, is edit file "variables.tf":
   - Private_slaves_count: (Number of private slaves you want to have, if they are 2, and you want 3 more, you have to put 5).

2. After updating the number of slaves we want to add to the cluster, just run again the "setup-cluster.sh" script and it will automatically configure and add the new agents to the HDFS cluster. ("Check the script and delete all the playbooks you don't want to scale, by default it scale all the platform").