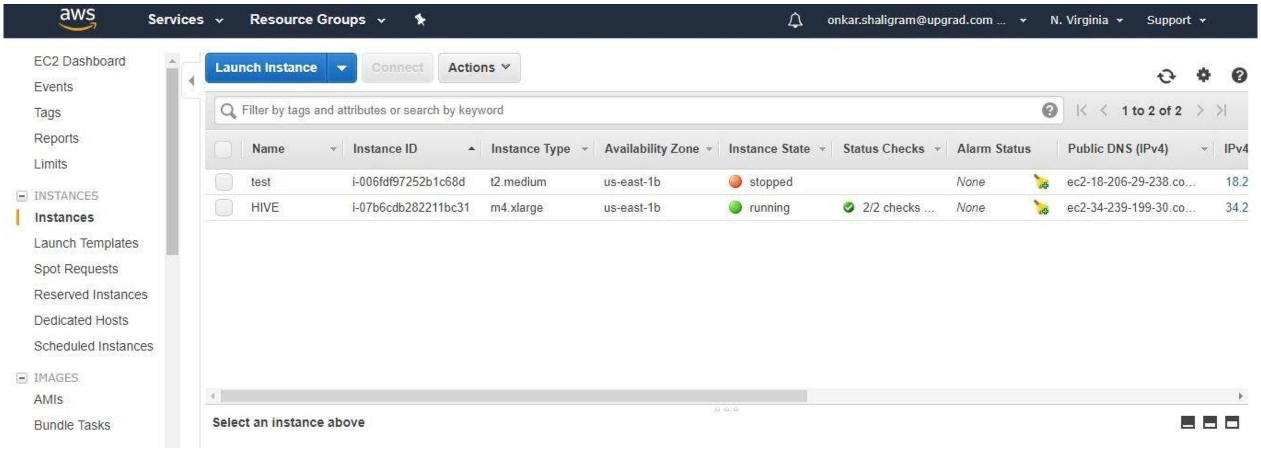
HDFS SHELL COMMANDS ON AMAZON EC2

Step 1: Start your AWS EC2 instance by logging in to your AWS Management Console.



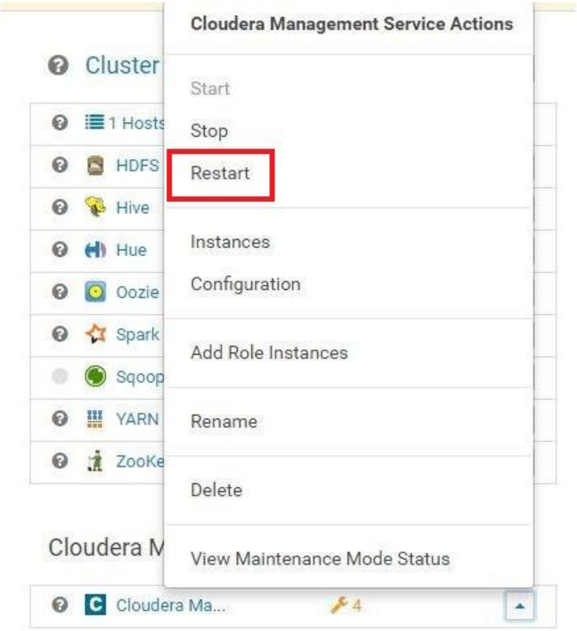
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
test	i-006fdf97252b1c68d	t2.medium	us-east-1b	stopped		None	ec2-18-206-29-238.co...	18.2
HIVE	i-07b6cdb282211bc31	m4.xlarge	us-east-1b	running	2/2 checks ...	None	ec2-34-239-199-30.co...	34.2

Step 2: Make sure that your instance is up and running fully after Step 1. Next, log in to the Cloudera Manager. It is available at the following link:

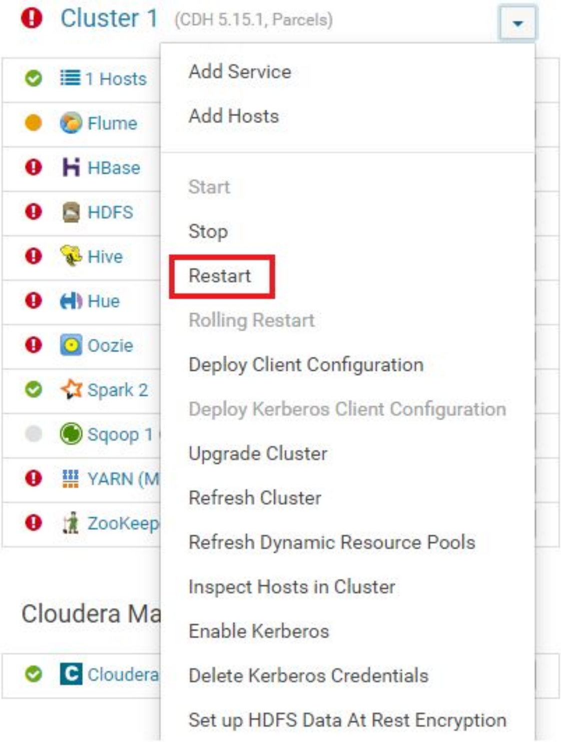
http://<public ip>:7180 [Here, <public ip> will be the ip of your machine such as 34.239.199.30.]

By default, both the username and the password for Cloudera Manager are ‘admin’. (If you have changed them manually, then use the new credentials.)

Step 3a: Restart the Cloudera Management Service



Step 3b: Restart Cluster 1.



Step 4: Connect to AWS EC2 (via PuTTY, etc.). You learnt how to connect to AWS EC2 in the previous modules.

The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS) as well as other file systems that Hadoop supports.

IMPORTANT INSTRUCTIONS

- The following notations have been used throughout the file:

```
[ec2-user@ip-10-0-0-14 ~]$ hadoop command
Output of the command
```

As shown above, the command to be run is written in **bold.** The output of the command is written in *italics*. The [**ec2-user**@ip-10-0-0-14 ~] tells us the user through which the command is to be executed.

- Please be careful with the spaces in the commands.
- If a series of commands is given in a particular order, make sure that you run them in the same order.

NOTE: Before starting with the document below, it is necessary to have created the EC2 instance with Cloudera installed on it and to have connected to it as well. If not
so, kindly go through the Introduction to Cloud and AWS setup module before getting started with this document.

BASIC COMMANDS

- To check the commands that are available in the HDFS, run any of the following commands.

hadoop fs -help or **hadoop dfs -help**

- To read the list of files in the HDFS, use the 'ls' command.

```
[ec2-user@ip-10-0-0-14 ~]$ hadoop fs -ls /
Found two items
drwxrwxrwt   - hdfs supergroup   0 2018-02-09 09:30 /tmp
drwxr-xr-x   - hdfs supergroup   0 2018-02-09 09:30 /user
```

- The ‘sudo -i’ command is used to switch from the ec2-user to the root user. Also the superuser while using hadoop is hdfs. The ‘su -hdfs’ helps us switch from the root user to the hdfs user when in the root user. To switch from the hdfs user to the root user, type ‘exit’.

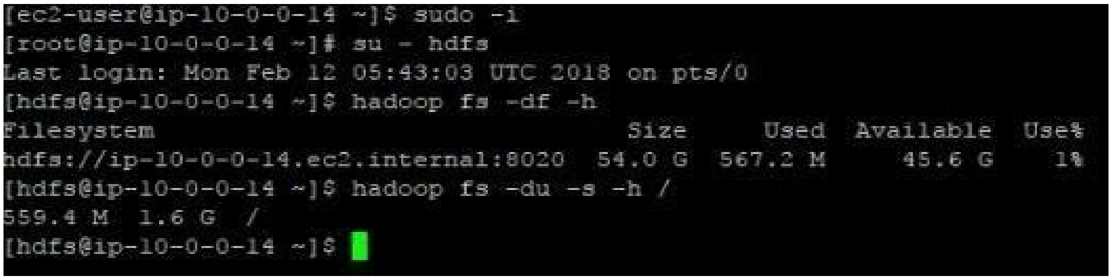
```
[ec2-user@ip-10-0-0-14 ~]$ sudo -i
[root@ip-10-0-0-14 ~]# su - hdfs
[hdfs@ip-10-0-0-14 ~]$ exit
[root@ip-10-0-0-14 ~]#
```

- **df**: This is a command to check the available space in the HDFS.

  du: This will help you check the space usage of the HDFS.

  Please note that both the commands can be run only from the hdfs user.

```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -df -h
Filesystem Size  Used Available Use%
hdfs://ip-10-0-0-14.ec2.internal:8020  54.0 G  567.2 M 45.6 G     1%
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -du -s -h /
559.4 M 1.6 G /
```



```
[ec2-user@ip-10-0-0-14 ~]$ sudo -i
[root@ip-10-0-0-14 ~]# su - hdfs
Last login: Mon Feb 12 05:43:03 UTC 2018 on pts/0
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -df -h
Filesystem                                 Size     Used  Available  Use%
hdfs://ip-10-0-0-14.ec2.internal:8020  54.0 G  567.2 M     45.6 G    1%
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -du -s -h /
559.4 M  1.6 G  /
[hdfs@ip-10-0-0-14 ~]$
```

Create a directory inside the HDFS

- The commands used below demonstrate how to create a directory in the HDFS.

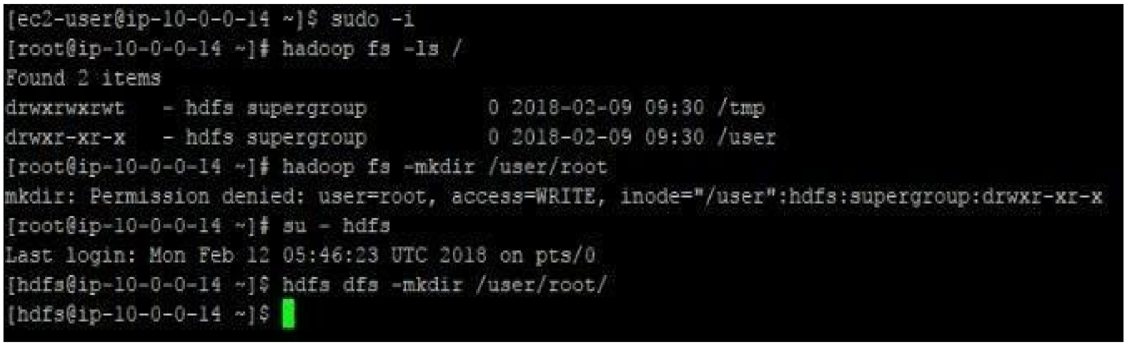
```
[ec2-user@ip-10-0-0-14 ~]$ sudo -i
```

(required to switch from ec2 to root. Not required if already in root)

```
[root@ip-10-0-0-14 ~]# hadoop fs -ls /
Found 2 items
drwxrwxrwt   - hdfs supergroup   0 2018-02-09 09:30 /tmp
drwxr-xr-x   - hdfs supergroup   0 2018-02-09 09:30 /user
[root@ip-10-0-0-14 ~]# hadoop fs -mkdir /user/root
mkdir: Permission denied: user=root, access=WRITE,
inode="/user":hdfs:supergroup:drwxr-xr-x
```

Note: As seen above, trying to create a directory in hadoop using the root user gave us an error. This error occured due to us trying to create a directory inside hdfs using the root user. Please note a directory can be created in hadoop only using the hdfs user. So now, switch to the hdfs user. Please note there is a space between **-** and **hdfs** in the command used below**.**

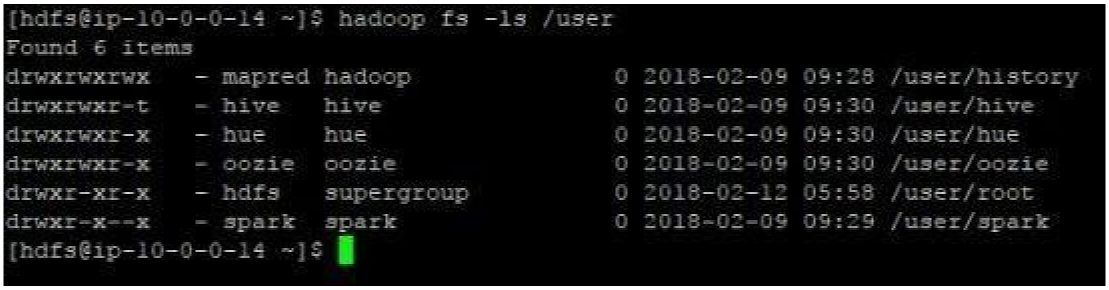
```
[root@ip-10-0-0-14 ~]# su - hdfs
Last login: Mon Feb 12 05:46:23 UTC 2018 on pts/0
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -mkdir /user/root/
```



```
[ec2-user@ip-10-0-0-14 ~]$ sudo -i
[root@ip-10-0-0-14 ~]# hadoop fs -ls /
Found 2 items
drwxrwxrwt   - hdfs supergroup          0 2018-02-09 09:30 /tmp
drwxr-xr-x   - hdfs supergroup          0 2018-02-09 09:30 /user
[root@ip-10-0-0-14 ~]# hadoop fs -mkdir /user/root
mkdir: Permission denied: user=root, access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x
[root@ip-10-0-0-14 ~]# su - hdfs
Last login: Mon Feb 12 05:46:23 UTC 2018 on pts/0
[hdfs@ip-10-0-0-14 ~]$ hdfs dfs -mkdir /user/root/
[hdfs@ip-10-0-0-14 ~]$
```

- You can verify the directory created by running the command shown below.

```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -ls /user
Found 6 items
drwxrwxrwx   - mapred hadoop     0 2018-02-09 09:28 /user/history
drwxrwxr-t   - hive   hive  0 2018-02-09 09:30 /user/hive
drwxrwxr-x   - hue    hue   0 2018-02-09 09:30 /user/hue
drwxrwxr-x   - oozie  oozie 0 2018-02-09 09:30 /user/oozie
drwxr-xr-x   - hdfs   supergroup 0 2018-02-12 05:58 /user/root
drwxr-x--x   - spark  spark 0 2018-02-09 09:29 /user/spark
```



```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -ls /user
Found 6 items
drwxrwxrwx   - mapred hadoop              0 2018-02-09 09:28 /user/history
drwxrwxr-t   - hive   hive                0 2018-02-09 09:30 /user/hive
drwxrwxr-x   - hue    hue                 0 2018-02-09 09:30 /user/hue
drwxrwxr-x   - oozie  oozie               0 2018-02-09 09:30 /user/oozie
drwxr-xr-x   - hdfs   supergroup          0 2018-02-12 05:58 /user/root
drwxr-x--x   - spark  spark               0 2018-02-09 09:29 /user/spark
[hdfs@ip-10-0-0-14 ~]$
```

Now, as seen above, the owner of the directory created is **hdfs** (underlined above). To send a file from any user to hdfs, the owner of the directory inside hdfs should be changed to the user sending the file. For example: If you have to send a file from the root user to a directory inside hdfs, the owner of that particular directory inside hdfs should be changed to root.

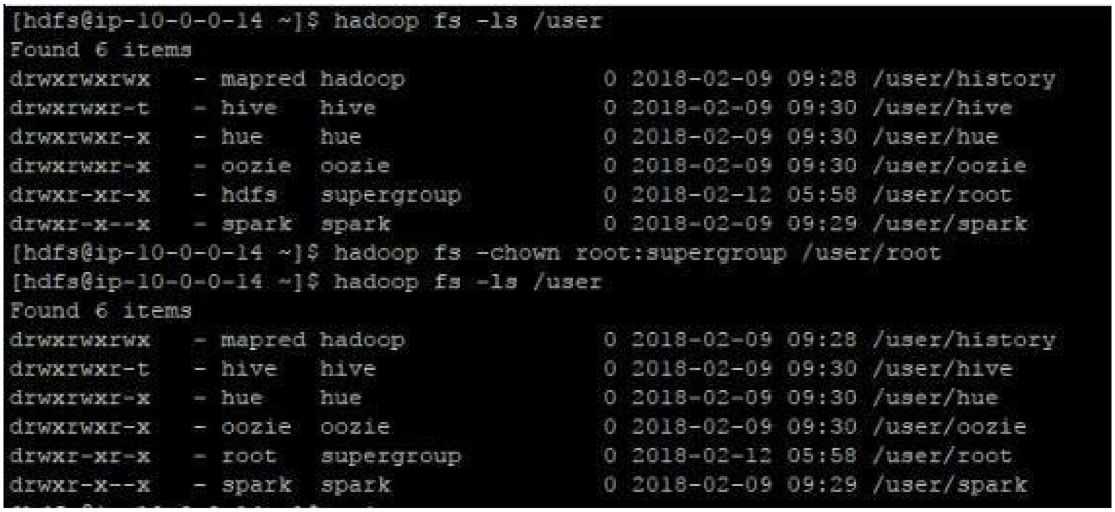
- To change the owner of the directory created from hdfs to root, run the following command:

```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -chown root:supergroup /user/root
```

- You can verify the same using the command shown below.

```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -ls /user
Found 6 items
drwxrwxrwx   - mapred hadoop     0 2018-02-09 09:28 /user/history
drwxrwxr-t   - hive   hive  0 2018-02-09 09:30 /user/hive
drwxrwxr-x   - hue    hue   0 2018-02-09 09:30 /user/hue
drwxrwxr-x   - oozie  oozie 0 2018-02-09 09:30 /user/oozie
drwxr-xr-x   - root   supergroup 0 2018-02-12 05:58 /user/root
drwxr-x--x   - spark  spark 0 2018-02-09 09:29 /user/spark
```

- You can see that the owner has changed from **hdfs** to **root**.



```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -ls /user
Found 6 items
drwxrwxrwx   - mapred hadoop              0 2018-02-09 09:28 /user/history
drwxrwxr-t   - hive   hive                0 2018-02-09 09:30 /user/hive
drwxrwxr-x   - hue    hue                 0 2018-02-09 09:30 /user/hue
drwxrwxr-x   - oozie  oozie               0 2018-02-09 09:30 /user/oozie
drwxr-xr-x   - hdfs   supergroup          0 2018-02-12 05:58 /user/root
drwxr-x--x   - spark  spark               0 2018-02-09 09:29 /user/spark
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -chown root:supergroup /user/root
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -ls /user
Found 6 items
drwxrwxrwx   - mapred hadoop              0 2018-02-09 09:28 /user/history
drwxrwxr-t   - hive   hive                0 2018-02-09 09:30 /user/hive
drwxrwxr-x   - hue    hue                 0 2018-02-09 09:30 /user/hue
drwxrwxr-x   - oozie  oozie               0 2018-02-09 09:30 /user/oozie
drwxr-xr-x   - root   supergroup          0 2018-02-12 05:58 /user/root
drwxr-x--x   - spark  spark               0 2018-02-09 09:29 /user/spark
```

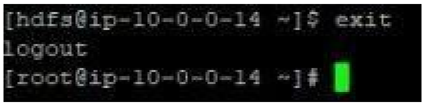
- Create ec2-user in the HDFS

```
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -mkdir -p /user/ec2-user
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -chown -R ec2-user:ec2-user /user/ec2-user
[hdfs@ip-10-0-0-14 ~]$ hadoop fs -chmod -R 777 /user/ec2-user
```

- Note: In these commands, the '-p' argument in the mkdir command means that it will create the parent directories as well if they didn't already exist. Also the '-R' argument in chown and chmod commands mean that it will recursively apply the command on any files or directories present in the directory mentioned in the command.

- Note that whenever you are performing a job using **root** user, then make sure that you are using the root directory i.e., ‘**/user/root**’ in HDFS and similarly if you are performing a job using **ec2-user** then make sure that you are using the ec2-user directory, i.e. ‘**/user/ec2-user**’ in HDFS, for your operations.

- Now, use the ‘exit’ command to shift from the hdfs user to the root user.



```
[hdfs@ip-10-0-0-14 ~]$ exit
logout
[root@ip-10-0-0-14 ~]#
```

Creating a file using a root user

- First, we create a file using the 'cat' command as shown below. After entering the contents of the file, we use 'Ctrl+Z' to save and exit the file.

```
[root@ip-10-0-0-14 ~]# cat > test.txt
```

- You can also use the "**vi test.txt**" command to create a text file using vi if you prefer. Keep in mind that you will have to go to the Input mode by pressing I and then write into the file and then later on press "Esc" to go back to the command mode and then type "wq!" to save and exit vi.

- Now verify whether the file has been created or not using the 'ls' command.

```
[root@ip-10-0-0-14 ~]# ls
test.txt
```

Copy a file in the local file system to the HDFS

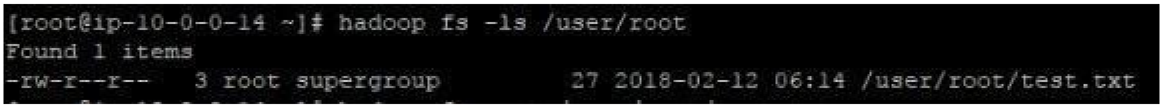
- Now, we will use the 'put' command to copy the file created above from the local file system to the HDFS. The syntax for the put command is:

  hadoop fs -put <src> <destination>

```
[root@ip-10-0-0-14 ~]# hadoop fs -put test.txt /user/root/
```

- We can verify whether the file has been copied as shown below:

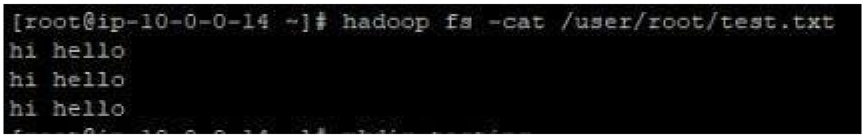
```
[root@ip-10-0-0-14 ~]# hadoop fs -ls /user/root
Found 1 items
-rw-r--r--   3 root supergroup   27 2018-02-12 06:14 /user/root/test.txt
```



```
[root@ip-10-0-0-14 ~]# hadoop fs -ls /user/root
Found 1 items
-rw-r--r--   3 root supergroup          27 2018-02-12 06:14 /user/root/test.txt
```

- Now, check the content of the file, using the 'cat' command.

```
[root@ip-10-0-0-14 ~]# hadoop fs -cat /user/root/test.txt
```

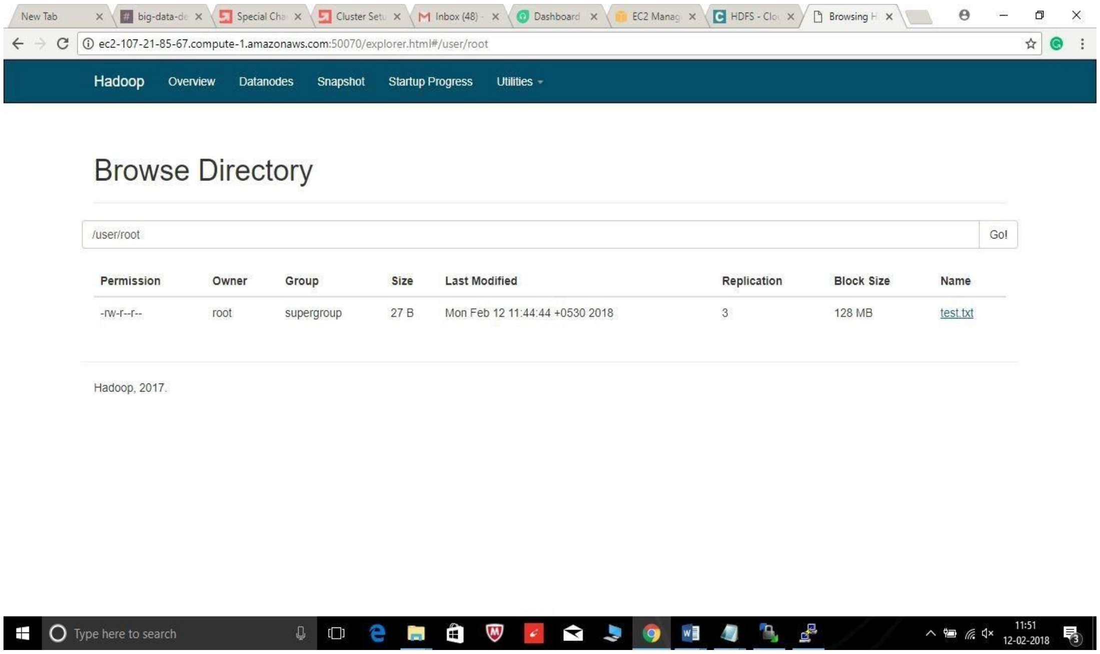


```
[root@ip-10-0-0-14 ~]# hadoop fs -cat /user/root/test.txt
hi hello
hi hello
hi hello
```

- We can also verify the same using the NameNode Browser. To do so, open your browser and add the Public IP address of your instance which can be known from the EC2 dashboard followed by ': 50070' as shown below.

<Public IP address>: 50070

Then click on 'Utilities', followed by "Browse File System". Locate the file to verify whether the file has been copied to the HDFS.



Browse Directory

/user/root Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	27 B	Mon Feb 12 11:44:44 +0530 2018	3	128 MB	test.txt

Hadoop, 2017.

Copy a file to the local file system from the HDFS

- We will now create a new directory in our local file system. Then we will copy a file from the HDFS to this local file system using the get command. Please note that in our case we are using the same file we had copied from the local file system to the HDFS. However, in this case it is copied to a new directory.

  Syntax: **hadoop fs -get <src> <destination>**

- First, we create a new directory named testing using the 'mkdir' command.

```
[root@ip-10-0-0-14 ~]# mkdir testing
```

- Now, we will copy the file from the HDFS to the local system using the 'get' command.

```
[root@ip-10-0-0-14 ~]# hadoop fs -get /user/root/test.txt /root/testing
```

- Now, let us verify the same by navigating to the new directory using the 'cd' command. Then use the 'ls' command and verify whether your file is present or not.

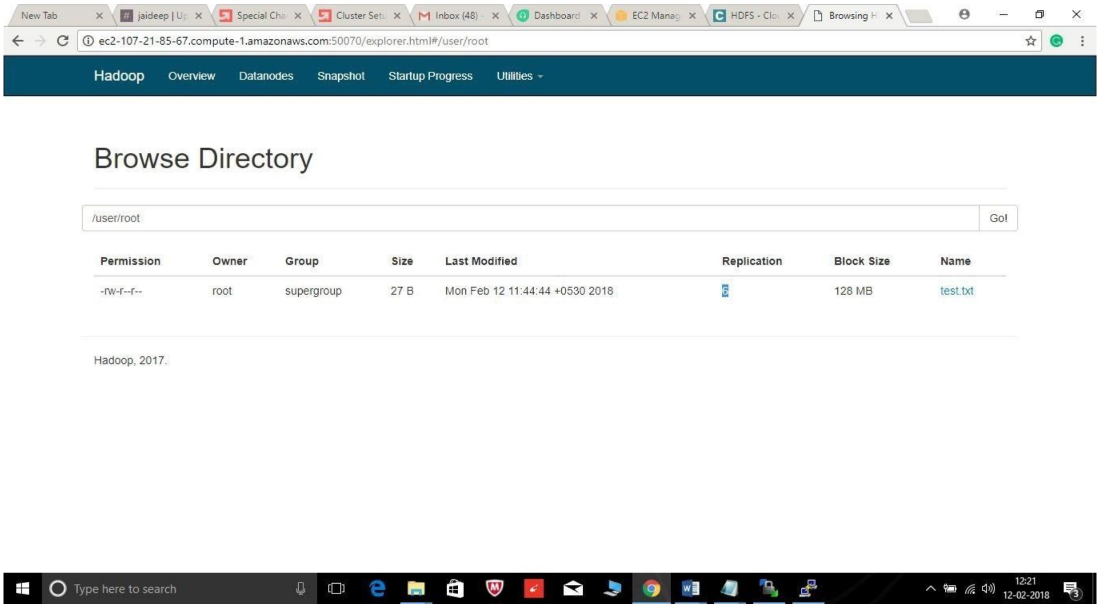
```
[root@ip-10-0-0-14 ~]# cd testing/
[root@ip-10-0-0-14 testing]# ls
test.txt
```

Change the replication factor of a particular file

- Please note that these steps are just for practise and should not be done while doing regular practise
- As we know, the default replication factor in the HDFS is 3. Now, we will use the 'setrep' command to change it to any value desired by us. In this case, we are setting the replication factor of the file test.txt to 6.

```
[root@ip-10-0-0-14 ~]# hadoop fs -setrep 6 /user/root/test.txt
Replication 6 set: /user/root/test.txt
```

- Now, let us verify the same using the NameNode browser. As mentioned earlier, we can access it by using **<Public IP address>: 50070.** Then click on 'Utilities', followed by "Browse File System". Locate the file to verify whether the replication factor has been set to 6.



Browse Directory

/user/root Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	27 B	Mon Feb 12 11:44:44 +0530 2018	6	128 MB	test.txt

Hadoop, 2017.