HADOOP INSTALLATION 3.0.3 ON LINUX 18.04

July 29,2019

Aishwarya Verma
2018MSBDA002
Central University of Rajasthan

Contents:

1. Installing JDK 8

1.1 Open terminal and type the following command to add Oracle's PPA,

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
```

1.2 Go to this link and download the java-8
https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

1.3 A .tar.gz file will get downloaded and extract that folder there itself.

1.4 By changing the mode of /opt directly we create a folder java and put the extracted folder from above in that folder.

```
$ sudo chmod a+rwx /opt
```

1.5 Open terminal and go to home directory.

```
$ cd
```

1.6 Now open the .bashrc file in your preferred text editor (I am using gedit).

```
$ gedit .bashrc
```

1.7 Add the following lines to the bottom of your .bashrc file.

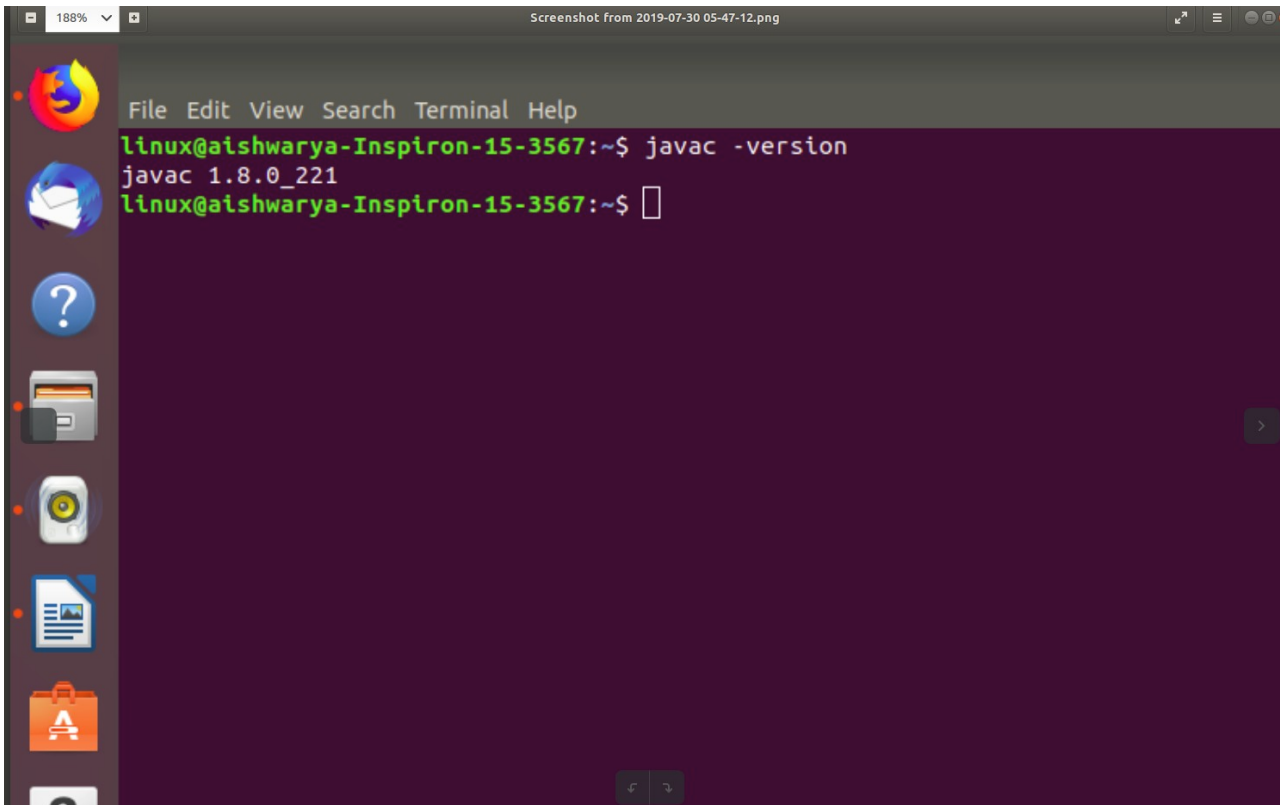
```
export PATH=$PATH:/opt/java/jdk1.8.0_221/bin
export JAVA_HOME=/opt/java/jdk1.8.0_221
```

1.8 Compile the .bashrc to make your changes permanent.

$ source .bashrc

1.9 Check if the installation is working type the following command.

$ javac -version



2. Setting up passphraseless SSH

2.1 If you have not installed SSH software you will need to install it.

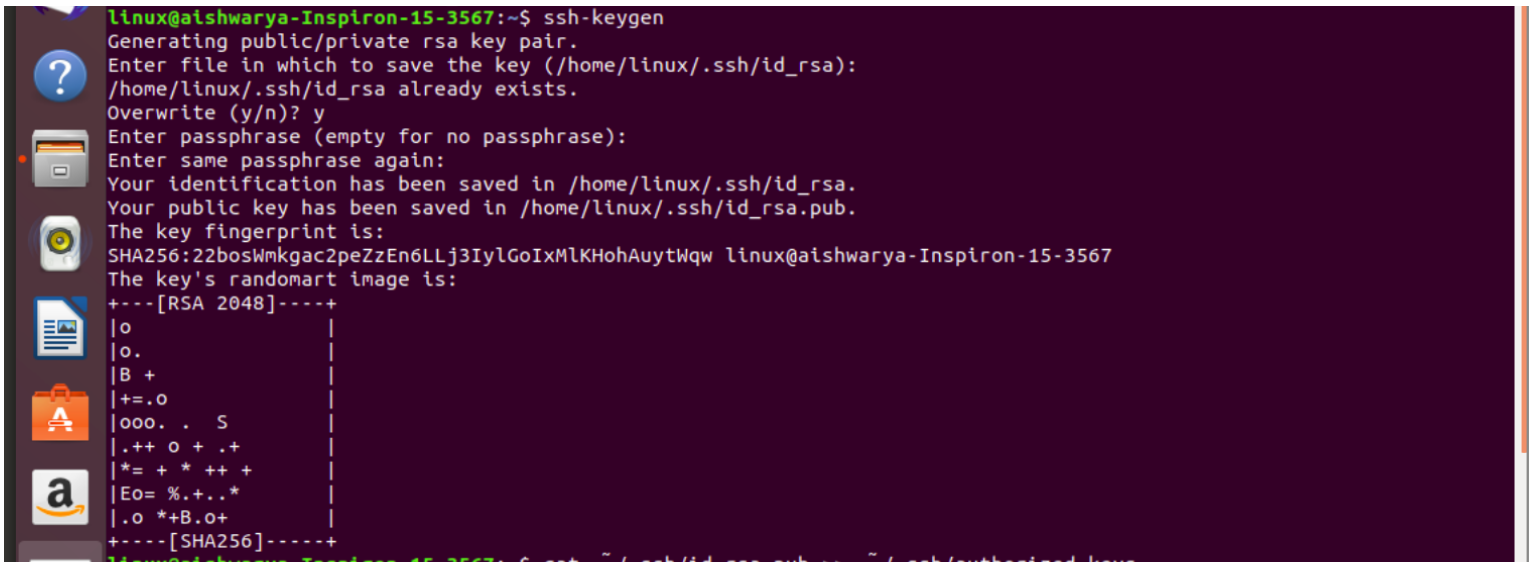
$ sudo apt-get install ssh

$ sudo apt-get install pdsh

2.2 Now in order to ssh to localhost without a passphrase (Empty password), execute the following commands:

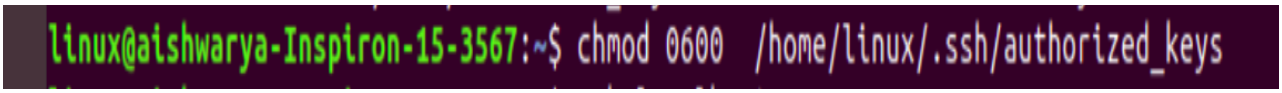
$ ssh-keygen

(Press ENTER whenever required without giving any passphrase)



$ cat /[path]/.ssh/id_rsa.pub >> /[path]/.ssh/authorized_keys

$ chmod 0600 /[path]/.ssh/authorized_keys



(Remark: To see the path of .ssh directory. Go to Home and click the option to show hidden files. You will then able to see the .ssh directory.)

3. Downloading and Installing Hadoop 3.0.3

3.1 Type the following command to download Hadoop 3.0.3 binary tar file to your machine. It will be downloaded to your home directory.

$ wget http://www-eu.apache.org/dist/hadoop/common/hadoop-3.0.3.tar.gz

3.2 Unpack the downloaded tar file to the home directory and change the mode of /usr/*local* directory *to* read and write by the following command .

$ tar -xvf hadoop-3.0.3.tar.gz -C /home/linux/Downloads

$ sudo chmod a+rwx /usr/local

3.3 Now a folder named hadoop-3.0.3 will appear in home directory . Change its name to hadoop and cut the folder and paste it into /usr/local directory.

3.4 Now it's time to set the Hadoop specific environment variable. open .bashrc for editing.

$ cd

$ gedit .bashrc

3.5 Add the following lines to your .bashrc.

export PATH=$PATH:/opt/java/jdk1.8.0_221/bin

export JAVA_HOME=/opt/java/jdk1.8.0_221

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop

```
export
HADOOP_MAPRED_HOME=/usr/local/hadoop
export
HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_HDFS_HOME=/usr/local/hadoop
export YARN_HOME=/usr/local/hadoop
export PATH=$PATH:/usr/local/hadoop
export PATH=$PATH:/usr/local/hadoop/bin
export PATH=$PATH:/usr/local/hadoop/sbin
```

3.6 Compile your .bashrc to make your changes permanent.

```
$ source .bashrc
```

3.7 To check if it worked out type the following command.

```
$ hadoop version
```

Your output will be something like this.

4. HDFS and Yarn Configuration

We have all the Hadoop binaries on our system but that's not it yet. In order to get Hadoop up and running we have to specify certain properties and configurations. Which is easily done with a little
bit of XML hacking.

Change to hadoop configuration directory

$ cd /usr/local/hadoop/etc/hadoop

4.1 Edit hadoop-env.sh

4.1.1 Open hadoop-env.sh for editing.

$ gedit hadoop-env.sh

4.1.2 Add the following line to point Hadoop installation towards your JDK (Version may change).

export JAVA_HOME=/opt/java/jdk1.8.0_221

4.2 Edit core-site.xml

This file informs Hadoop daemon where NameNode runs in the cluster.

4.2.1 Open core-site.xml for editing.

$ gedit core-site.xml

4.2.2 Add the following properties in between the <configuration> and </configuration> tags.

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

4.3 Edit hdfs-site.xml

This file contains configuration settings of various HDFS daemons (i.e. NameNode, DataNode,Secondary NameNode). It also includes the replication factor and block size of HDFS.

4.3.1 Open hdfs-site.xml for editing.

```
$ gedit hdfs-site.xml
```

4.3.2 Add the following properties in between the <configuration> and </configuration> tags.

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

4.4 Edit mapred-site.xml

This file contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process,
etc.

If mapred-site.xml file is not available. So, we have to create the mapred-site.xml fileusing mapred-site.xml template.

4.4.1 Open mapred-site.xml for editing.

```
$ gedit mapred-site.xml
```

4.4.2 Add the following properties in between the <configuration> and </configuration> tags.

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>mapreduce.application.classpath</name>
<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
</property>
</configuration>
```

4.5 Edit yarn-site.xml

This file contains configuration settings of ResourceManager and NodeManager like application

memory management size ,the operation needed on program and algorithm.

4.5.1 Open yarn-site.xml for editing.

$ gedit yarn-site.xml

4.5.2 Add the following properties in between the <configuration> and </configuration> tags.

<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>

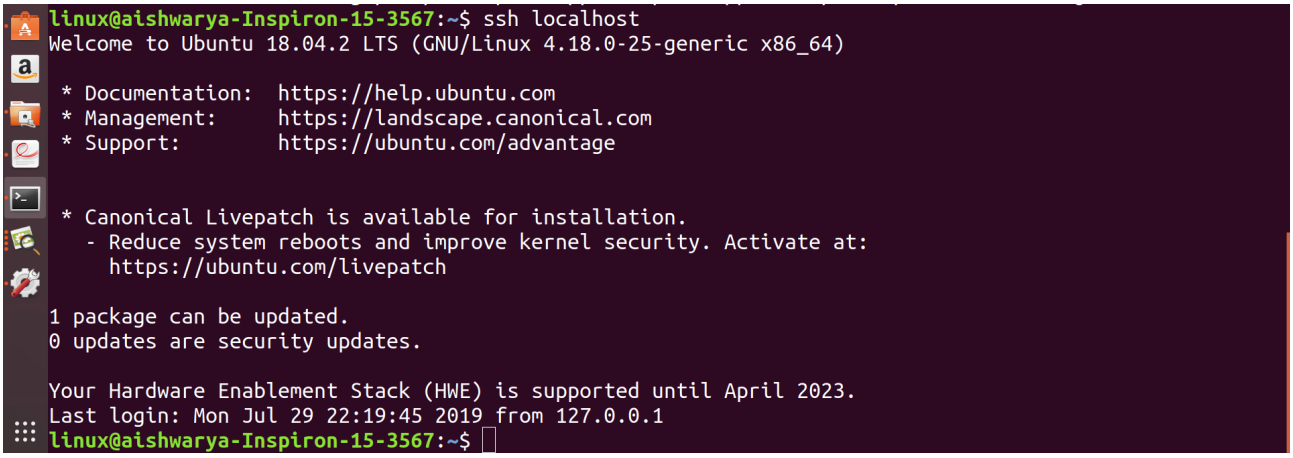
5. Running Hadoop Services

Now since we have added all the the Hadoop binaries and shell script which are present in /usr/local/hadoop/bin and in /usr/local/hadoop/sbin to our PATH. We can directly run those binaries and scripts from terminal by just typing in their names.

5.1 Firstly we need to SSH to localhost

$ ssh localhost

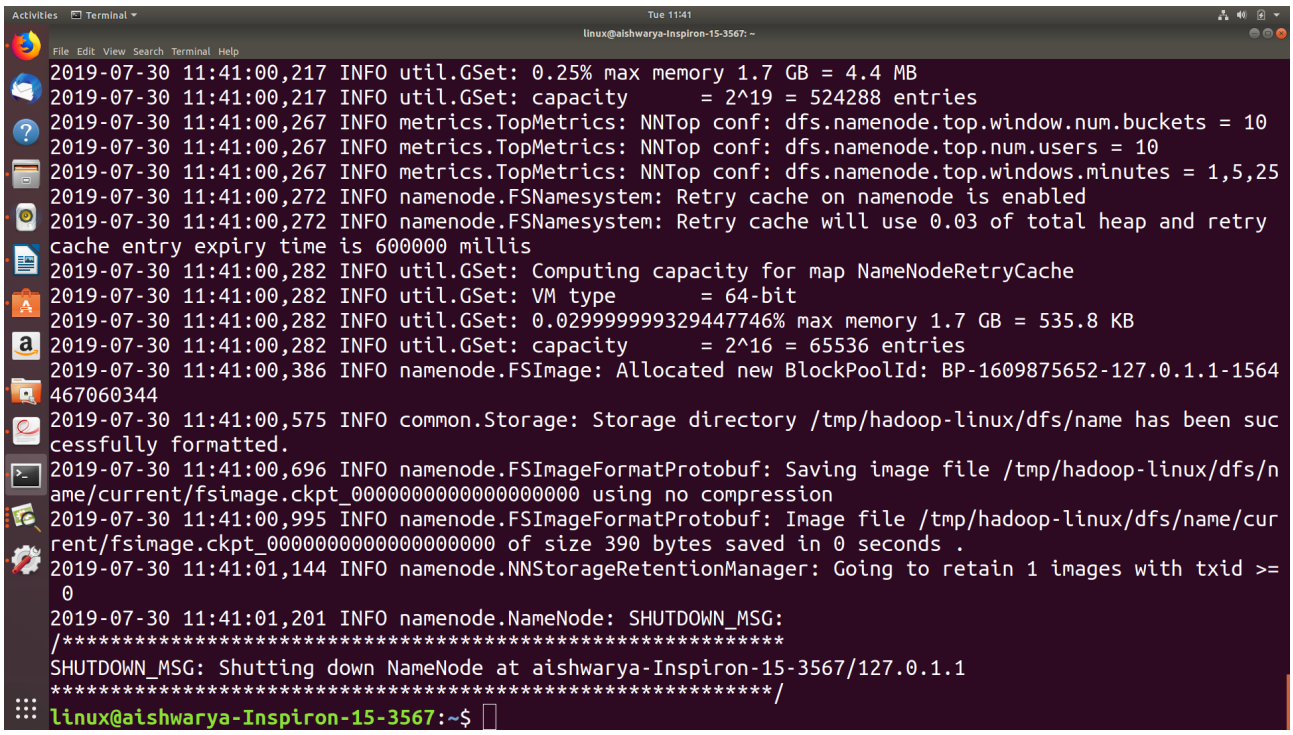
You will see output like below. You are taken to the BASH terminal of your machine but this time commands will run via a SSH connection to localhost (or 127.0.0.1).



5.2 Before running our cluster for the first time we need to format our namenode

$ hadoop namenode -format

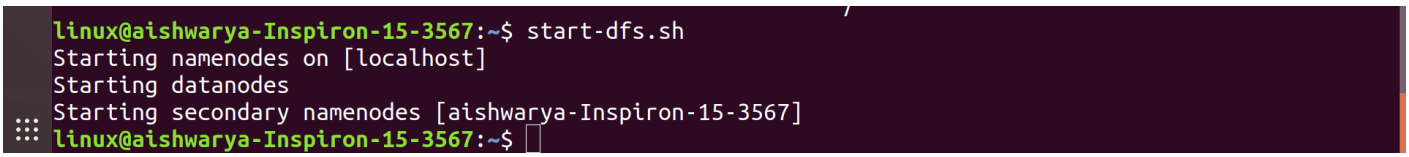
You will receive the below output:



5.3 Now to start the HDFS file system with its Namenodes and Datanodes enter the following command.

$ start-dfs.sh

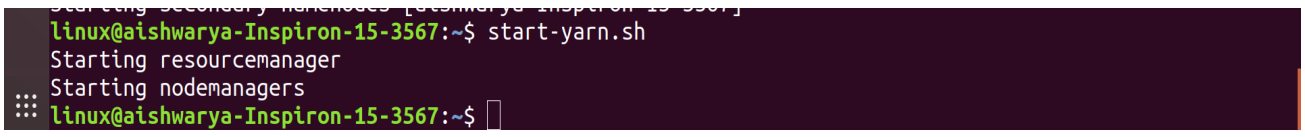
You will receive the below output:



5.4 Now to start the Yarn resource manager type.

$ start-yarn.sh

You will receive the below output:



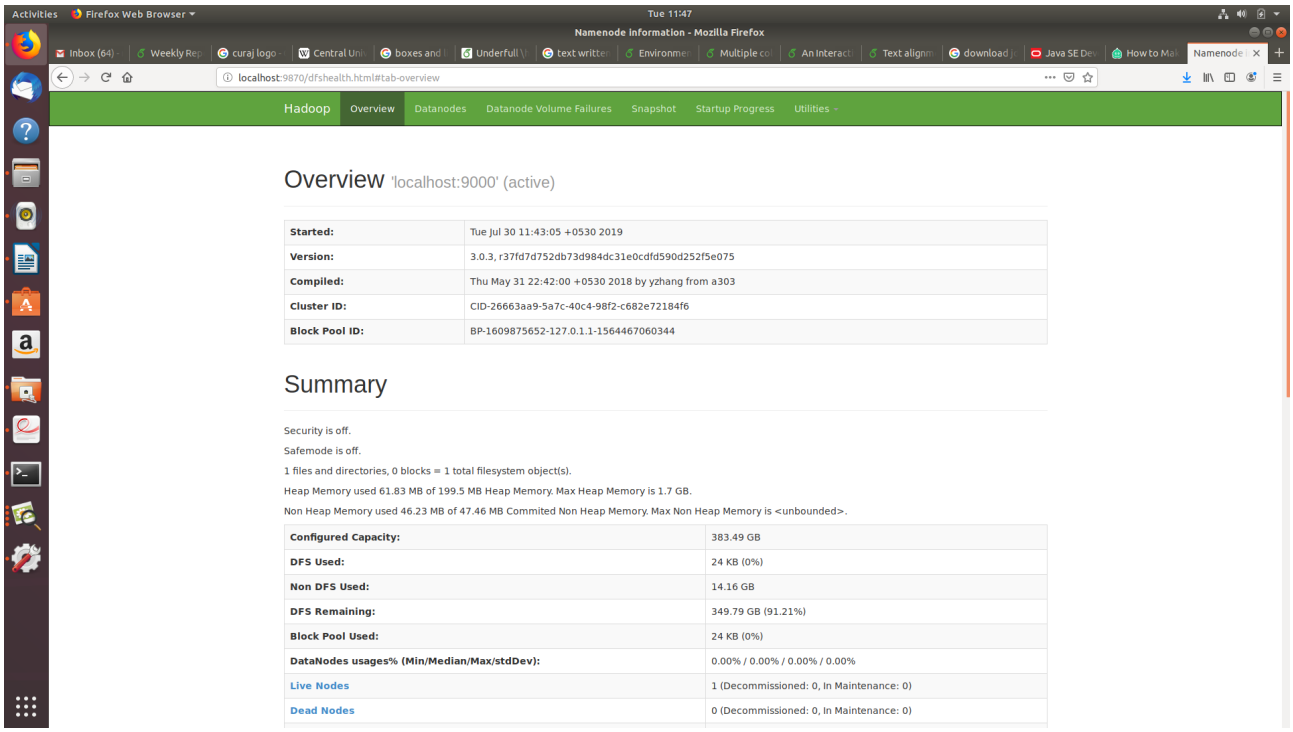
5.5 Using jps command we can see all the running services.

$ jps

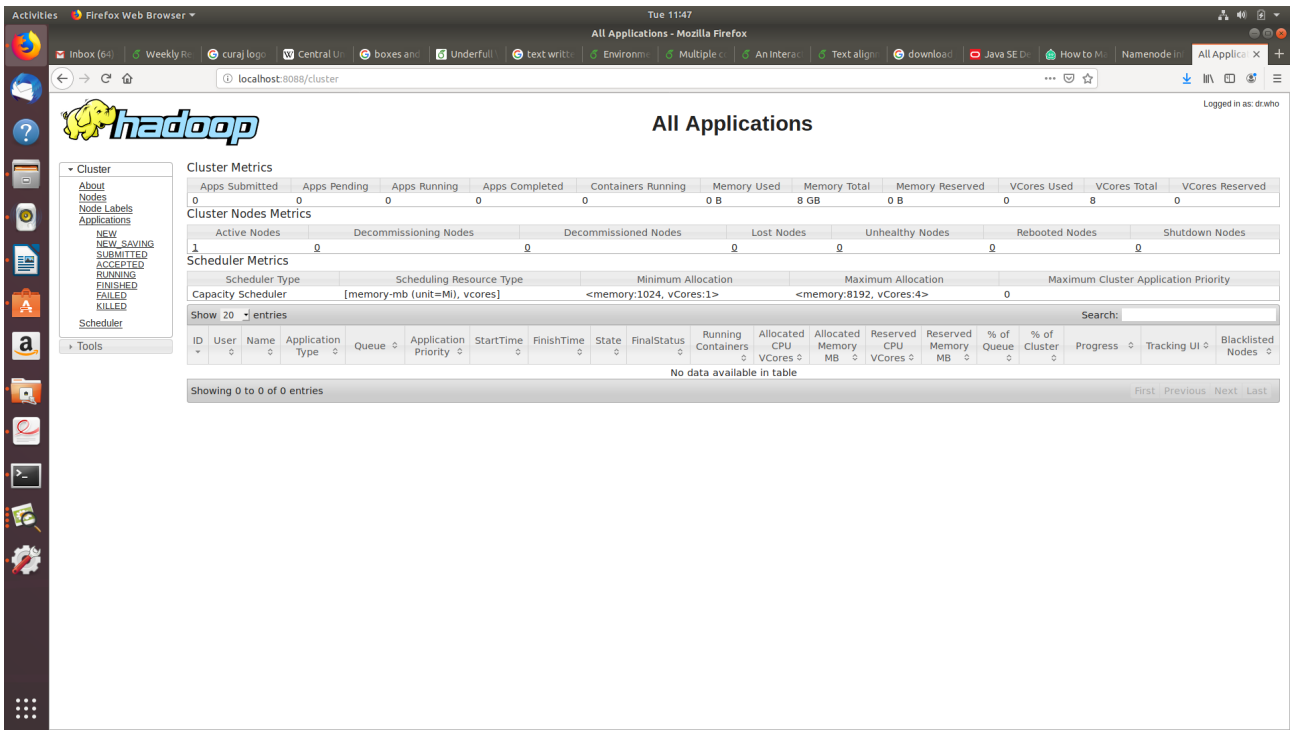
You will receive the below output:

```
linux@aishwarya-Inspiron-15-3567:~$ jps
9669 Jps
8375 NameNode
8551 DataNode
9273 NodeManager
8812 SecondaryNameNode
9102 ResourceManager
linux@aishwarya-Inspiron-15-3567:~$
```

5.6 Open **http://localhost:9870** in browser to see the Namenode interface.
You will receive the below output:



5.7 Open **http://localhost:8088** in browser to see the Cluster interface.
You will receive the below output:


Activities
Firefox Web Browser
Tue 11:47
All Applications - Mozilla Firefox
localhost:8088/cluster
Logged in as: dr.who
All Applications
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted
Apps Pending
Apps Running
Apps Completed
Containers Running
Memory Used
Memory Total
Memory Reserved
VCores Used
VCores Total
VCores Reserved
0 0 0 0 0 0 B 8 GB 0 B 0 8 0
Cluster Nodes Metrics
Active Nodes
Decommissioning Nodes
Decommissioned Nodes
Lost Nodes
Unhealthy Nodes
Rebooted Nodes
Shutdown Nodes
1 0 0 0 0 0 0
Scheduler Metrics
Scheduler Type
Scheduling Resource Type
Minimum Allocation
Maximum Allocation
Maximum Cluster Application Priority
Capacity Scheduler
[memory-mb (unit=Mi), vcores]
<memory:1024, vCores:1>
<memory:8192, vCores:4>
0
Show 20 entries
Search:
ID
User
Name
Application Type
Queue
Application Priority
StartTime
FinishTime
State
FinalStatus
Running Containers
Allocated CPU VCores
Allocated Memory MB
Reserved CPU VCores
Reserved Memory MB
% of Queue
% of Cluster
Progress
Tracking UI
Blacklisted Nodes
No data available in table
Showing 0 to 0 of 0 entries
First
Previous
Next
Last

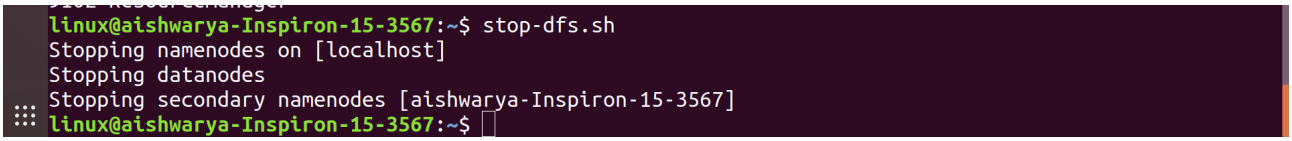
6. Conclusion and Stopping

Congratulations !, Your Linux(ubuntu) 18.04Machine has a fully functional single Node Hadoop 3.0.3
cluster up and running.

6.1 Now,To start the HDFS file system enter the following command.

$ stop-dfs.sh

You will receive the below output:



6.2 To stop the Yarn resource manager type.

$ stop-yarn.sh

You will receive the below output: