
HADOOP INSTALLATION 3.0.3 ON LINUX 18.04

July 29,2019

Aishwarya Verma
2018MSBDA002
Central University of Rajasthan

Contents:

- Installing JDK 8.....
- Setting up passphraseless SSH.....
- Downloading and Installing Hadoop 3.0.3.....
- HDFS and Yarn Configuration.....
- Running Hadoop Services.....
- Conclusion and Stopping.....

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.

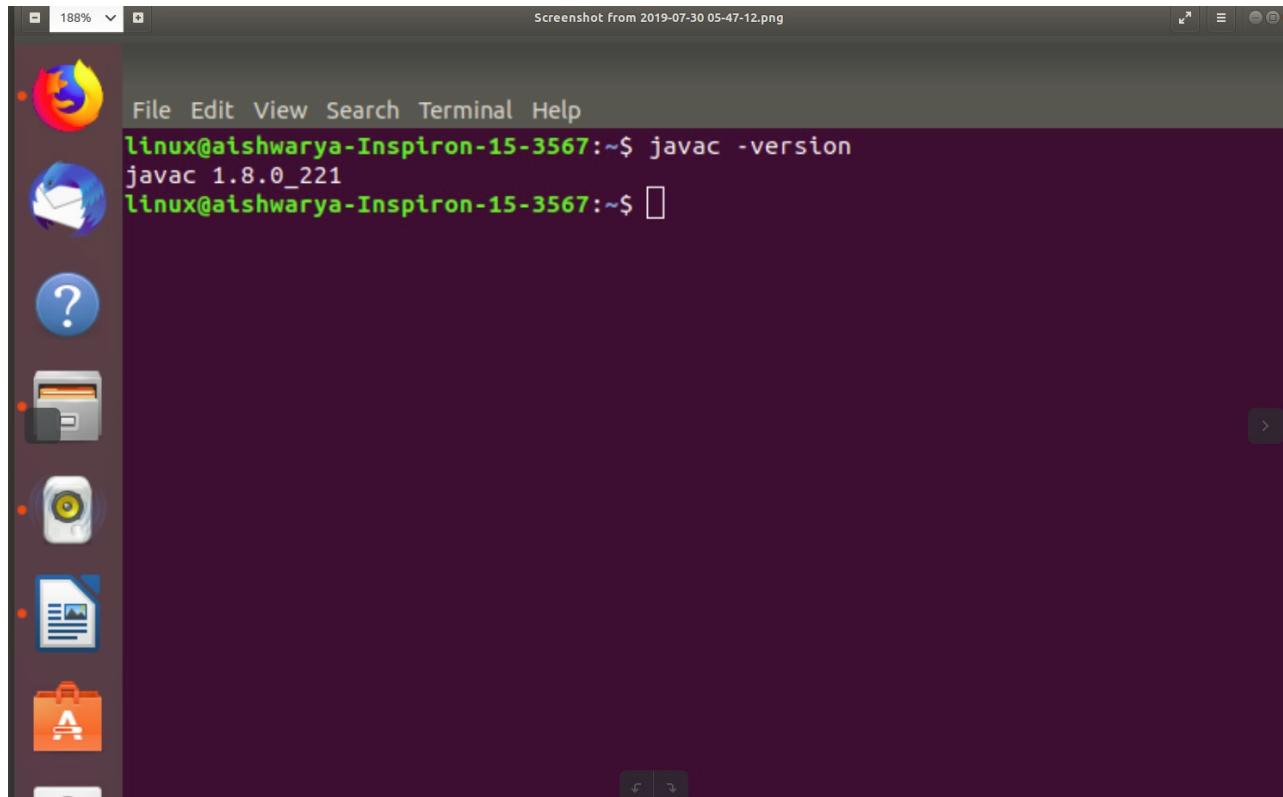
```
export PATH=$PATH:/opt/java/jdk1.8.0_221/bin  
export JAVA_HOME=/opt/java/jdk1.8.0_221
```

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

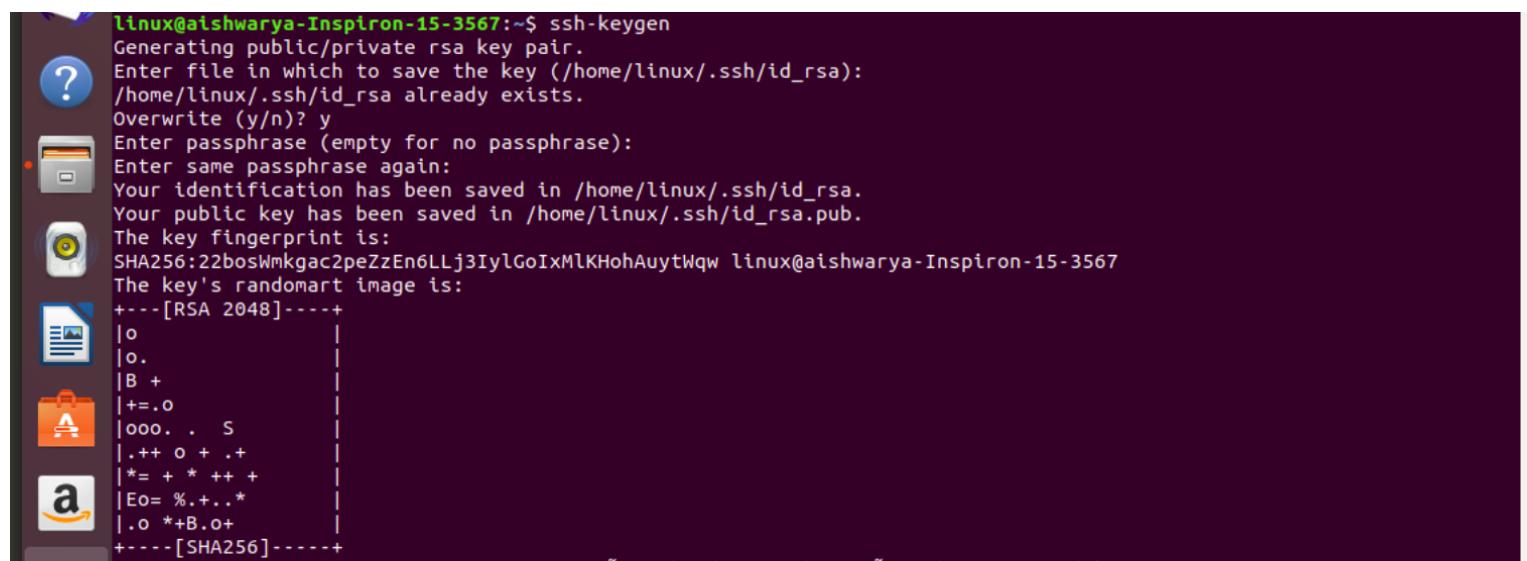
2.1 If you have not installed SSH software you will need to install it.

```
$ 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)



```
linux@aishwarya-Inspiron-15-3567:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/linux/.ssh/id_rsa):  
/home/linux/.ssh/id_rsa already exists.  
Overwrite (y/n)? y  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/linux/.ssh/id_rsa.  
Your public key has been saved in /home/linux/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:22bosWmkac2peZzEn6LLj3IylGoIxMlKHohAuytWqw linux@aishwarya-Inspiron-15-3567  
The key's randomart image is:  
+---[RSA 2048]---  
|o  
|o.  
|B +  
|+=.o  
|ooo.. . S  
|...++ o + .+  
|*= + * ++ +  
|Eo= %...*  
|.o *+B.o+  
+---[SHA256]---
```

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



```
linux@aishwarya-Inspiron-15-3567:~$ cat /home/linux/.ssh/id_rsa.pub >> /home/linux/.ssh/authorized_keys
```

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

```
linux@aishwarya-Inspiron-15-3567:~$ chmod 0600 /home/linux/.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.

```
linux@aishwarya-Inspiron-15-3567:~$ hadoop version  
Hadoop 3.0.3  
Source code repository https://yjzhangal@git-wip-us.apache.org/repos/asf/hadoop.git -r 37fd7d752db73d984dc31e0cdfd590d252f  
5e075  
Compiled by yzhang on 2018-05-31T17:12Z  
Compiled with protoc 2.5.0  
From source with checksum 736cdcefa911261ad56d2d120bf1fa  
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.0.3.jar  
linux@aishwarya-Inspiron-15-3567:~$
```

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 file using 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,HA
DOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPAT
H_PREPEND_DISTCACHE,HADOOP_YARN_HOME,H
ADOOP_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).

```
linux@aishwarya-Inspiron-15-3567:~$ ssh localhost
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

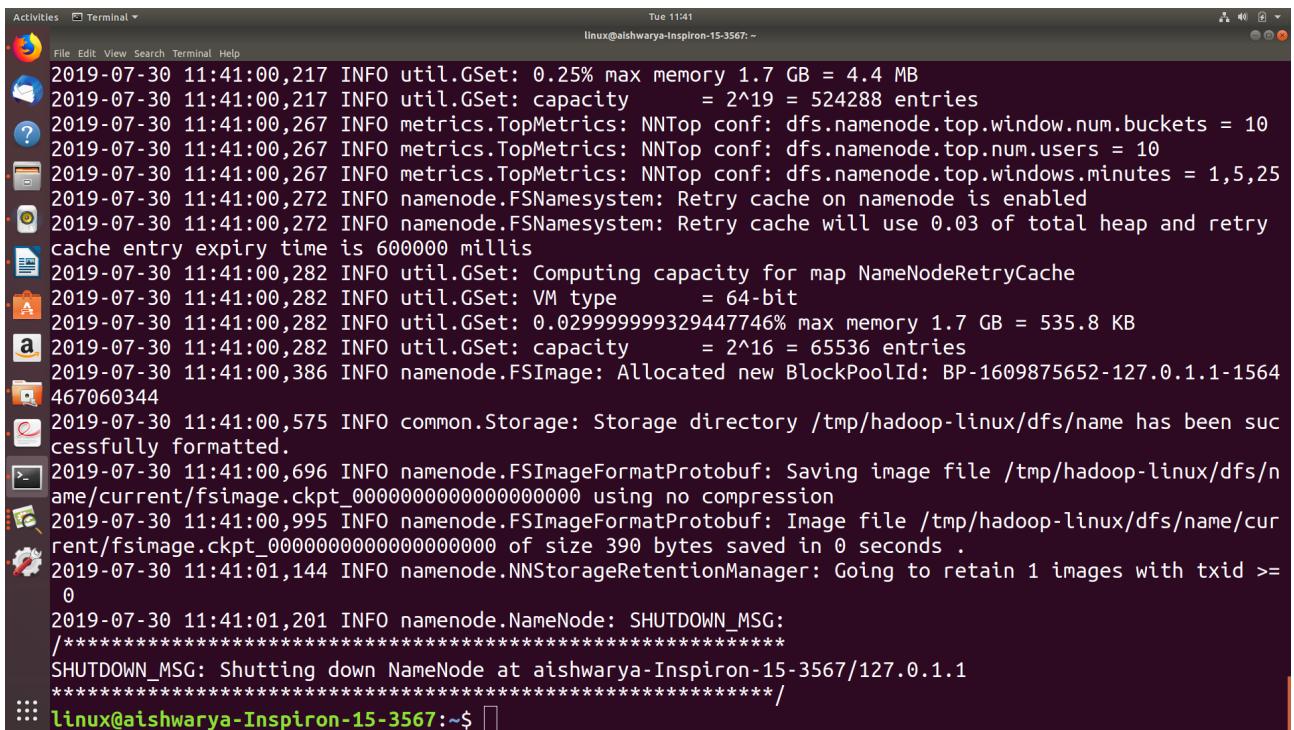
1 package can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Jul 29 22:19:45 2019 from 127.0.0.1
linux@aishwarya-Inspiron-15-3567:~$ 
```

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:



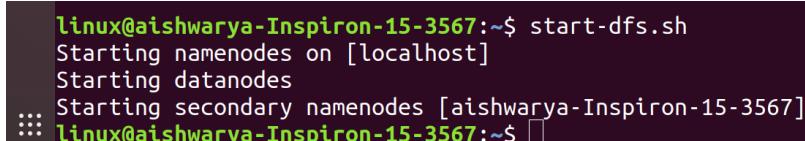
A screenshot of a Linux terminal window titled "Terminal". The window shows a log of HDFS startup messages. The log includes details about NameNode and DataNode initialization, metrics, and file system operations. The terminal window has a dark background with white text and some colored icons for file types.

```
Activities Terminal Tue 11:41  
linux@aishwarya-Inspiron-15-3567:~  
2019-07-30 11:41:00,217 INFO util.GSet: 0.25% max memory 1.7 GB = 4.4 MB  
2019-07-30 11:41:00,217 INFO util.GSet: capacity = 2^19 = 524288 entries  
2019-07-30 11:41:00,267 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10  
2019-07-30 11:41:00,267 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10  
2019-07-30 11:41:00,267 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25  
2019-07-30 11:41:00,272 INFO namenode.FSNamesystem: Retry cache on namenode is enabled  
2019-07-30 11:41:00,272 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry  
cache entry expiry time is 600000 millis  
2019-07-30 11:41:00,282 INFO util.GSet: Computing capacity for map NameNodeRetryCache  
2019-07-30 11:41:00,282 INFO util.GSet: VM type = 64-bit  
2019-07-30 11:41:00,282 INFO util.GSet: 0.029999999329447746% max memory 1.7 GB = 535.8 KB  
2019-07-30 11:41:00,282 INFO util.GSet: capacity = 2^16 = 65536 entries  
2019-07-30 11:41:00,386 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1609875652-127.0.1.1-1564  
467060344  
2019-07-30 11:41:00,575 INFO common.Storage: Storage directory /tmp/hadoop-linux/dfs/name has been suc  
cessfully formatted.  
2019-07-30 11:41:00,696 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-linux/dfs/n  
ame/current/fsimage.ckpt_00000000000000000000 using no compression  
2019-07-30 11:41:00,995 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-linux/dfs/name/cu  
rent/fsimage.ckpt_00000000000000000000 of size 390 bytes saved in 0 seconds .  
2019-07-30 11:41:01,144 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >=  
0  
2019-07-30 11:41:01,201 INFO namenode.NameNode: SHUTDOWN_MSG:  
*****  
SHUTDOWN_MSG: Shutting down NameNode at aishwarya-Inspiron-15-3567/127.0.1.1  
*****  
linux@aishwarya-Inspiron-15-3567:~$
```

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:

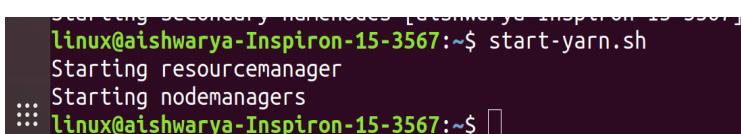


```
linux@aishwarya-Inspiron-15-3567:~$ start-dfs.sh  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [aishwarya-Inspiron-15-3567]  
linux@aishwarya-Inspiron-15-3567:~$
```

5.4 Now to start the Yarn resource manager type.

\$ start-yarn.sh

You will receive the below output:



```
Starting secondary namenodes [aishwarya-Inspiron-15-3567]  
linux@aishwarya-Inspiron-15-3567:~$ start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers  
...  
linux@aishwarya-Inspiron-15-3567:~$
```

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:

The screenshot shows a Firefox browser window with the title "Namenode Information - Mozilla Firefox". The address bar shows the URL "localhost:9870/dfshealth.html#tab-overview". The main content area has a green header bar with tabs: "Hadoop" (selected), "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". Below the header, there are two sections: "Overview" and "Summary".

Overview 'localhost:9000' (active)

Started:	Tue Jul 30 11:43:05 +0530 2019
Version:	3.0.3, r37fd7d752db73d984dc31e0c0cfdf590d252f5e075
Compiled:	Thu May 31 22:42:00 +0530 2018 by yzhang from a303
Cluster ID:	CID-26663aa9-5a7c-40c4-9bf2-c682e72184f6
Block Pool ID:	BP-1609875652-127.0.1.1-1564467060344

Summary

Security is off.
Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 61.83 MB of 199.5 MB Heap Memory. Max Heap Memory is 1.7 GB.

Non Heap Memory used 46.23 MB of 47.46 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	383.49 GB
DFS Used:	24 KB (0%)
Non DFS Used:	14.16 GB
DFS Remaining:	349.79 GB (91.21%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)

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

Logged in as: drwho

The screenshot shows the Hadoop cluster metrics and applications overview. The left sidebar has a tree view with 'Cluster' expanded, showing 'About Nodes', 'Node Labels', 'Applications' (with sub-options: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler), and 'Tools'. The main content area has sections for 'Cluster Metrics' (with tables for Apps Submitted, Apps Pending, Apps Running, Apps Completed, Containers Running, Memory Used, Memory Total, Memory Reserved, VCores Used, VCores Total, VCores Reserved) and 'Scheduler Metrics' (with tables for Active Nodes, Decommissioning Nodes, Decommissioned Nodes, Lost Nodes, Unhealthy Nodes, Rebooted Nodes, Shutdown Nodes). Below these are tables for 'Cluster Nodes Metrics' and 'Scheduler Metrics'. A search bar and pagination controls ('Showing 0 to 0 of 0 entries', 'First', 'Previous', 'Next', 'Last') are at the bottom.

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

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) 18.04 Machine 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:

```
linux@aishwarya-Inspiron-15-3567:~$ stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [aishwarya-Inspiron-15-3567]
::: linux@aishwarya-Inspiron-15-3567:~$ 
```

6.2 To stop the Yarn resource manager type.

\$ stop-yarn.sh

You will receive the below output:

```
linux@aishwarya-Inspiron-15-3567:~$ stop-yarn.sh
Stopping nodemanagers
Stopping resourcemanager
::: linux@aishwarya-Inspiron-15-3567:~$ 
```