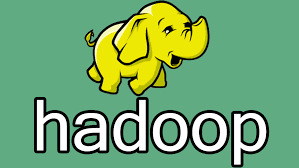
**Hadoop Implementation Steps on Ubuntu 16.-04/18.04 Linux**



**(COMPUTER SCIENCE AND ENGINEERING)**

**BY**

**ADITYA BHARDWAJ**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PEC**

**SECTOR – 12, CHANDIGARH, INDIA**

**2019**

**Step 1 – Prerequsities**

Before beginning the installation run login shell as the sudo user and update the current packages installed. Lets my ubuntu host name is server3

sudo apt update

### **OpenJDK 8**

Java 8 is the current Long Term Support version and is still widely supported, though public maintenance ends in January 2019. To install OpenJDK 8, execute the following command:

[root@server3](mailto:root@server3): sudo apt install openjdk-8-jdk

Verify that this is installed with

[root@server3](mailto:root@server3): java -version

You'll see output like this:

Output

openjdk version "1.8.0\_162"

OpenJDK Runtime Environment (build 1.8.0\_162-8u162-b12-1-b12)

OpenJDK 64-Bit Server VM (build 25.162-b12, mixed mode)

You have successfully installed Java 11 on Ubuntu 16.04 LTS system.

[root@server3](mailto:root@server3):readlink -f /usr/bin/java | sed "s:bin/java::"

[root@server3](mailto:root@server3):sudo gedit /etc/environment

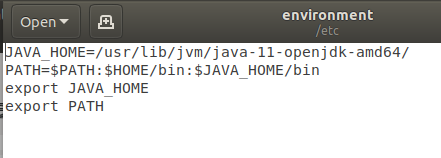
Following configuration are done in environment file

JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64/

PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin

export JAVA\_HOME

export PATH



Verify that the environment variable is set:

[root@server3](mailto:root@server3): echo $JAVA\_HOME

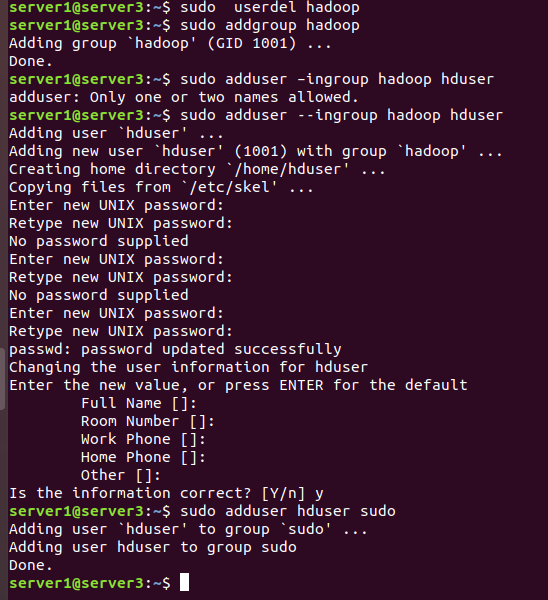
## Step 2 – Create User for Haddop

Hit CTRL+ALT+T to get started. We will install Hadoop from the terminal. For new Linux users, things might get confusing while installing different programs and managing them from the same login. If you are one of them, we have a solution. Let’s create a new dedicated Hadoop user. Whenever you want to use [Hadoop](https://www.digitalvidya.com/blog/understanding-hadoop-ecosystem-concept-and-applications/), just use the separate login. Simple.

$ sudo addgroup hadoop

$ sudo adduser –ingroup hadoop hduser

Note: You just enter Unix user name pwd and for other Just hit enter and press ‘y’ at the end.

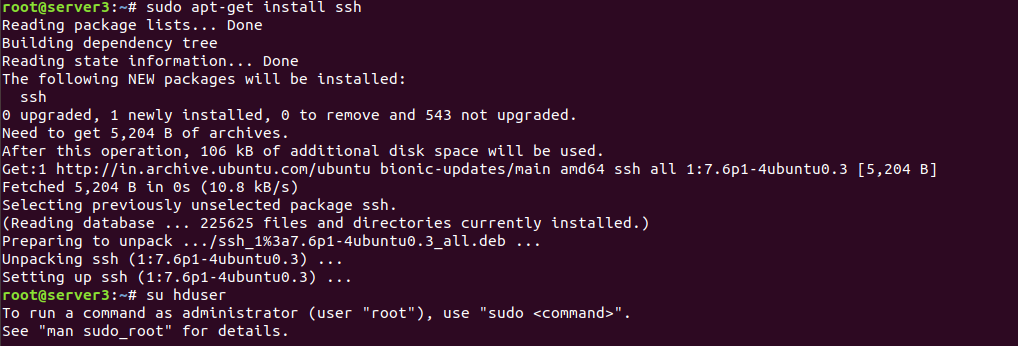


Add Hadoop user to sudo group (Basically, grant it all permissions)

server1@server3: sudo adduser hduser sudo

#### Install SSH

[root@server3](mailto:root@server3): sudo apt-get install ssh



Passwordless entry for localhost using SSH

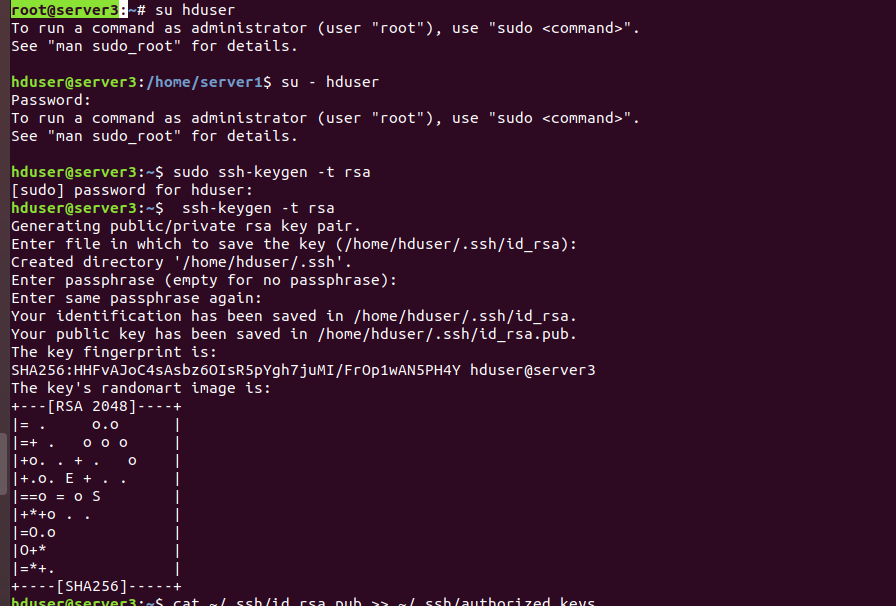
[root@server3](mailto:root@server3): su -hduser

[hduser@server3](mailto:hduser@server3): sudo ssh-keygen -t rsa

Note: When ask for file name or location, leave it blank.

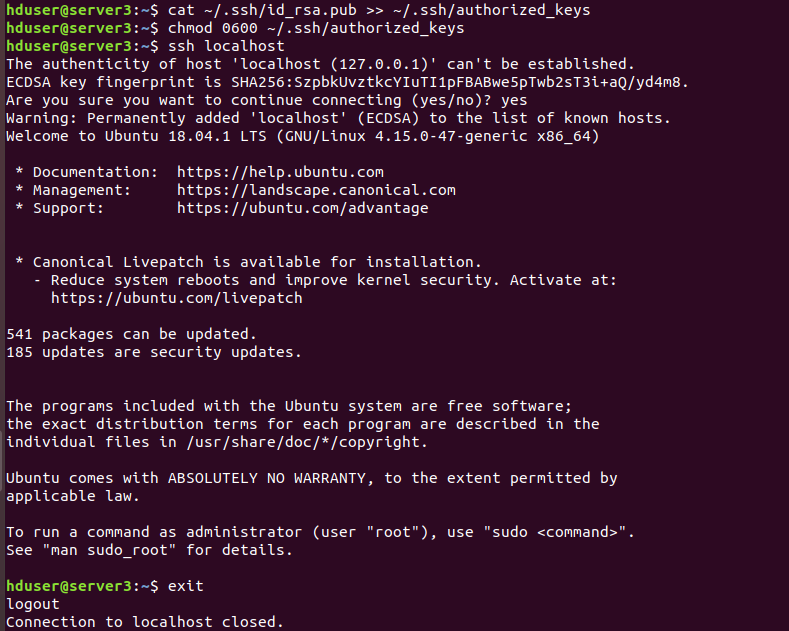
[hduser@server3](mailto:hduser@server3): cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

[hduser@server3](mailto:hduser@server3): chmod 0600 ~/.ssh/authorized\_keys

Figure: SSH Key generation

Check if ssh works,

$ **ssh localhost**

Figure: hduser permission

Once we are logged in localhost, exit from this session using following command.

$ exit

## Step 3 – Download Hadoop Source Archive

In this step, download hadoop 3.1 source archive file using below command. You can also select alternate [download mirror](https://www.apache.org/dyn/closer.cgi/hadoop/common/) for increasing download speed.

cd ~

[server1@server3](mailto:server1@server3): wget http://www-eu.apache.org/dist/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz

[server1@server3](mailto:server1@server3): tar xzf hadoop-3.1.2.tar.gz

#### 3.2 Hadoop Configuration

Make a directory called hadoop from the hduser and move the folder ‘hadoop-3.1.2’ to this directory

[server1@server3](mailto:server1@server3): sudo mkdir -p /usr/local/hadoop

[server1@server3](mailto:server1@server3): cd hadoop-3.1.2/

[server1@server3](mailto:server1@server3): sudo mv \* /usr/local/hadoop

[server1@server3](mailto:server1@server3): sudo chown -R hduser:hadoop /usr/local/hadoop

## STEP 4 – Setting up Configuration files

We will change content of following files in order to complete hadoop installation.

1. ~/.bashrc
2. hadoop-env.sh
3. core-site.xml
4. hdfs-site.xml
5. yarn-site.xml

Details:

* **hadoop-env.sh** – This file contains some environment variable settings used by Hadoop. You can use these to affect some aspects of Hadoop daemon behavior, such as where log files are stored, the maximum amount of heap used etc. The only variable you should need to change at this point is in this file is JAVA\_HOME, which specifies the path to the Java 1.7.x installation used by Hadoop.
* **core-site.xml** –  key property fs.default.name – for namenode configuration for e.g hdfs://*namenode*/. **Namenode** is the node which stores the filesystem metadata i.e. which file maps to what block locations and which blocks are stored on which datanode
* **hdfs-site.xml** – key property – dfs.replication – by default 3
* mapred-site.xml  – key property  mapred.job.tracker for jobtracker configuration for e.g *jobtracker*:8021
* yarn-site.xml: resource management

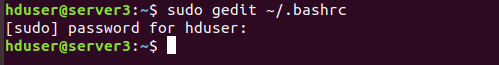
#### 4.1 ~/.bashrc

If you don’t know the path where java is installed, first run the following command to locate it

[root@server3](mailto:root@server3):readlink -f /usr/bin/java | sed "s:bin/java::"

Now open the ~/.bashrc file

hduser@server3:~$ sudo gedit ~/.bashrc



#HADOOP VARIABLES START

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export HADOOP\_HOME=/usr/local/hadoop

export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"

#HADOOP VARIABLES END

### 

Update .bashrc file to apply changes

$source ~/.bashrc

#### **4.2 hadoop-env.sh**

We need to tell Hadoop the path where java is installed. That’s what we will do in this file, specify the path for JAVA\_HOME variable.

Open the file,

hduser@server3:~$ sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh

Now, the first variable in file will be JAVA\_HOME variable, change the value of that variable to

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

#### **4.3 core-site.xml**

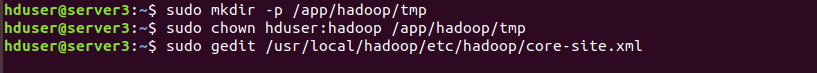
Create temporary directory

[hduser@server3](mailto:hduser@server3) :~$ sudo mkdir -p /app/hadoop/tmp

[hduser@server3](mailto:hduser@server3) :~$ sudo chown hduser:hadoop /app/hadoop/tmp

open the file

[hduser@server3](mailto:hduser@server3) :~$ sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml

Append the following between configuration tags. Same as below.

<property>

<name>hadoop.tmp.dir</name>

<value>/app/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

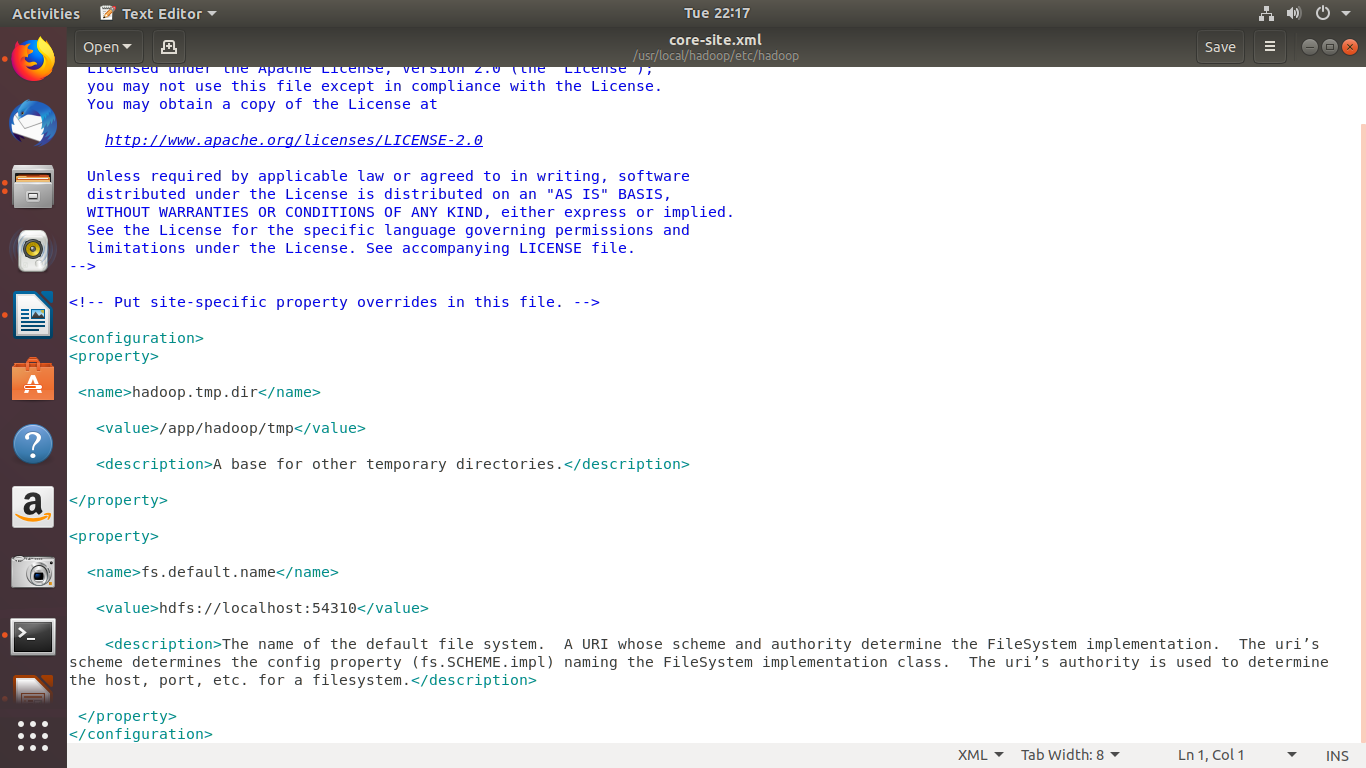
<property>

<name>fs.default.name</name>

<value>hdfs://localhost:54310</value>

<description>The name of the default file system. A URI whose scheme and authority determine the FileSystem implementation. The uri’s scheme determines the config property (fs.SCHEME.impl) naming the FileSystem implementation class. The uri’s authority is used to determine the host, port, etc. for a filesystem.</description>

</property>



#### **4.4 hdfs-site.xml**

Mainly there are two directories,

1. Name Node
2. Data Node

Make directories

[**hduser@server3**](mailto:hduser@server3) **sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode**

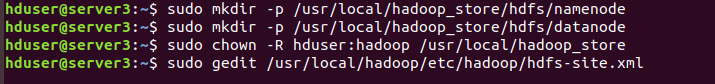
[**hduser@server3**](mailto:hduser@server3) **sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode**

[**hduser@server3**](mailto:hduser@server3) **sudo mkdir -p /usr/local/hadoop\_store/hdfs/datanode**

[**hduser@server3**](mailto:hduser@server3) **sudo chown -R hduser:hadoop /usr/local/hadoop\_store**

Open the file,

[**hduser@server3**](mailto:hduser@server3) **sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml**



Change the content between configuration tags shown as below.

<property>

<name>dfs.replication</name>

<value>1</value>

<description>Default block replication.The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.

</description>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop\_store/hdfs/namenode</value>

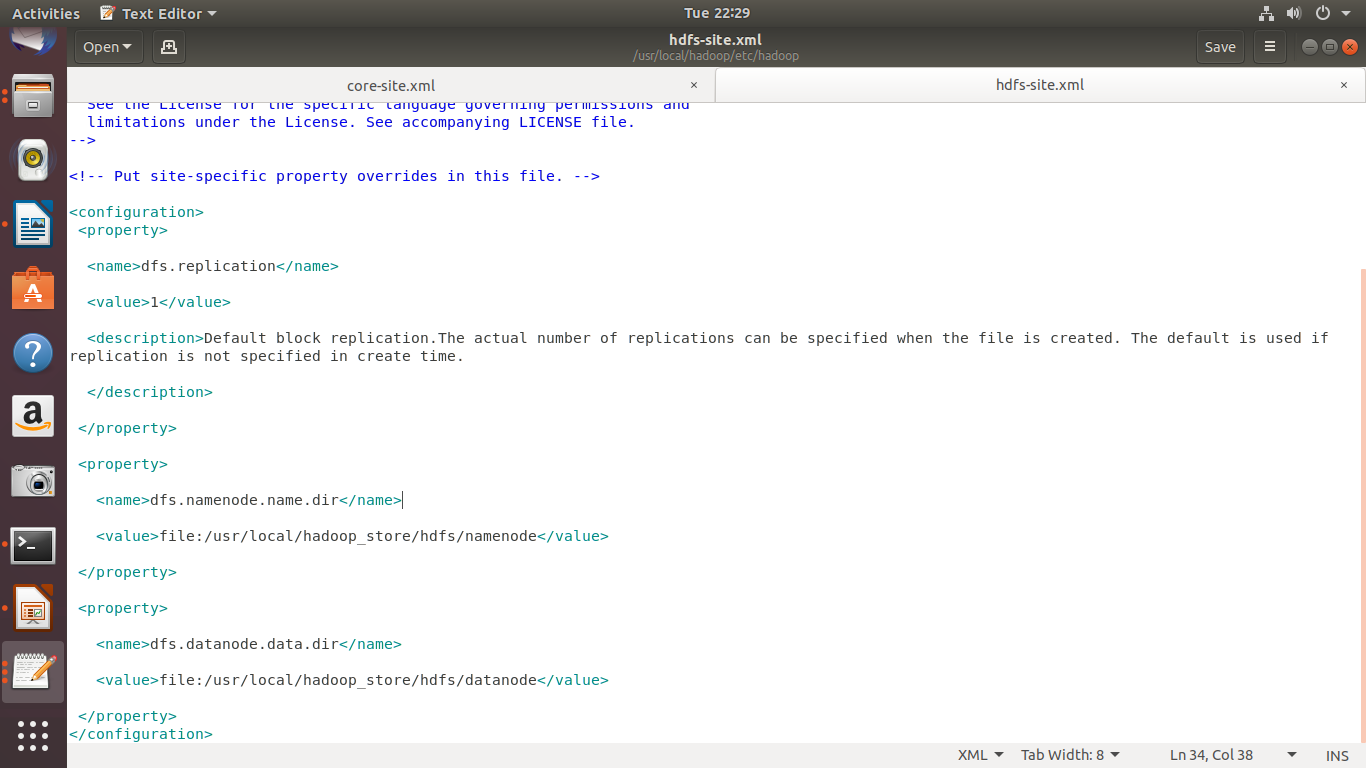
</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop\_store/hdfs/datanode</value>

</property>



#### **4.5 yarn-site.xml**

Open the file,

[**hduser@server3**](mailto:hduser@server3) **:~$ sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml**

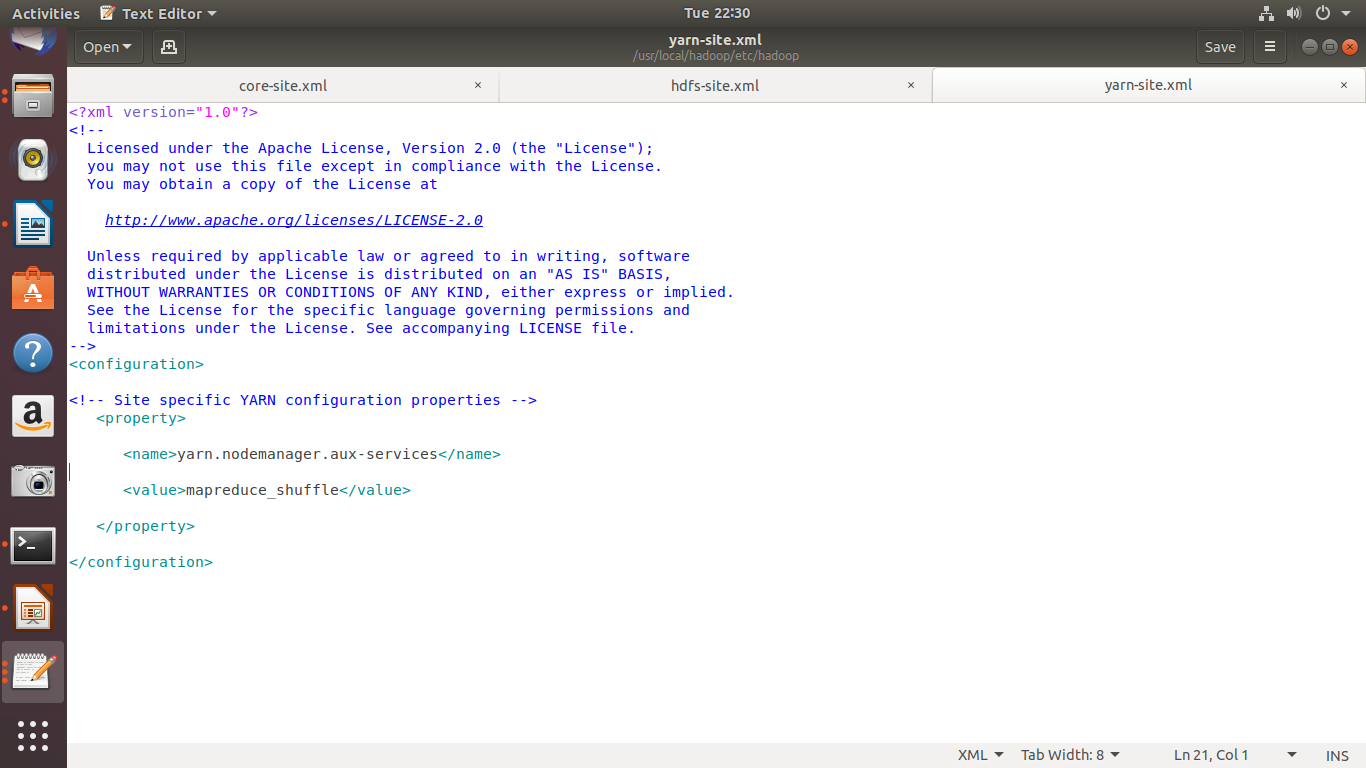
Just like the other two, add the content to configuration tags.

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

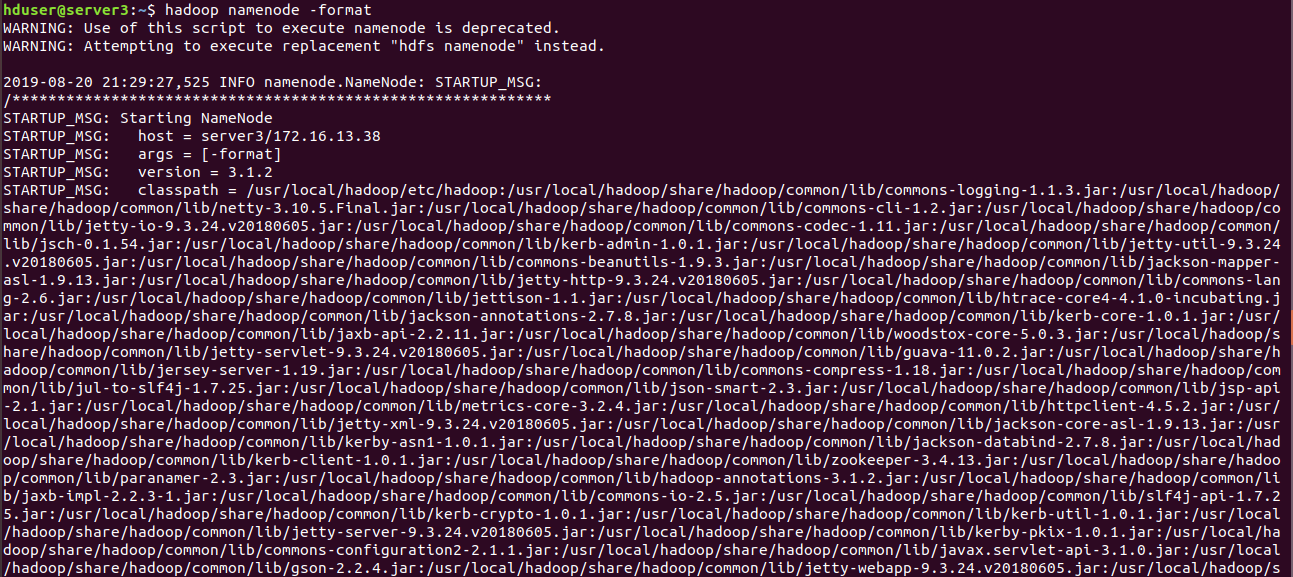
</property>

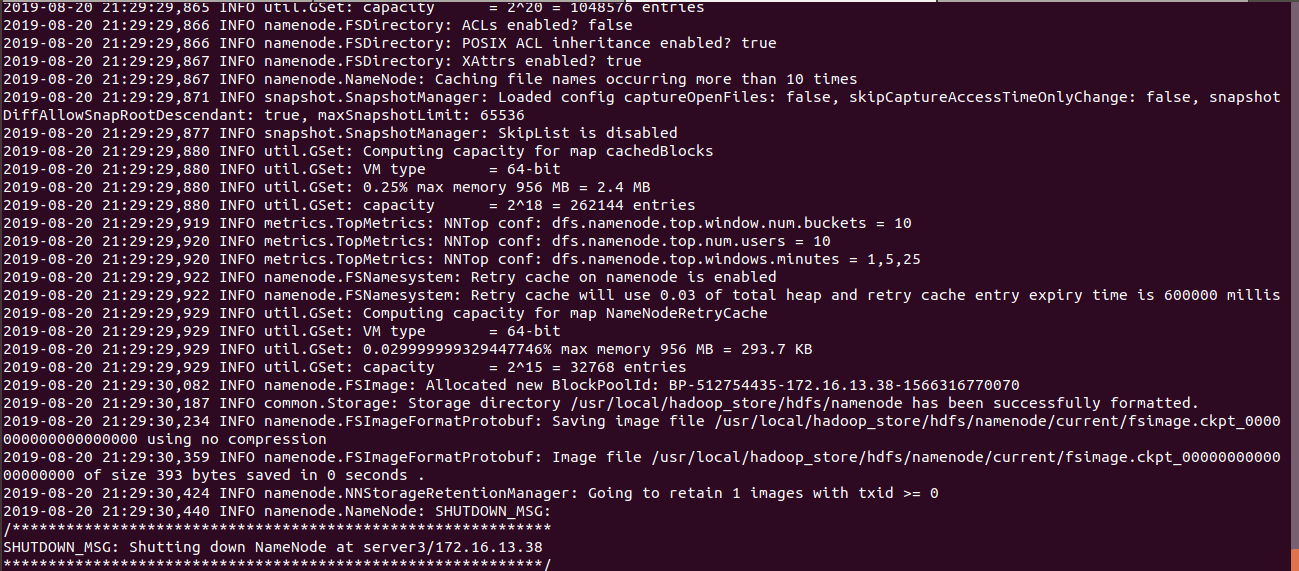


### **STEP 5- Format Hadoop file system**

Hadoop installation is now done. All we have to do is change format the name-nodes before using it.

[hduser@server3](mailto:hduser@server3) :~$ hadoop namenode -format



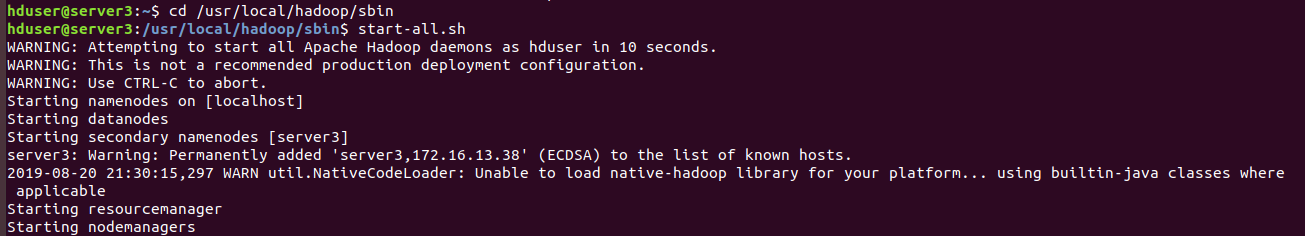


### **STEP 6- Start Hadoop daemons**

Now that hadoop installation is complete and name-nodes are formatted, we can start hadoop by going to following directory.

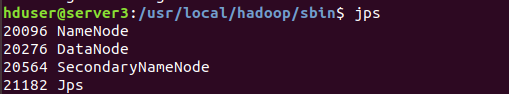
**$ cd /usr/local/hadoop/sbin**

**$ start-all.sh**



Just check if all daemons are properly started using the following command:

**$ jps**



### **STEP 7 – IF you want to Stop Hadoop daemons**

Step 7 of hadoop installation is when you need to stop Hadoop and all its modules.

**$ stop-all.sh**

Appreciate yourself because you’ve done it. You have completed all the Hadoop installation steps and Hadoop is now ready to run the first program.

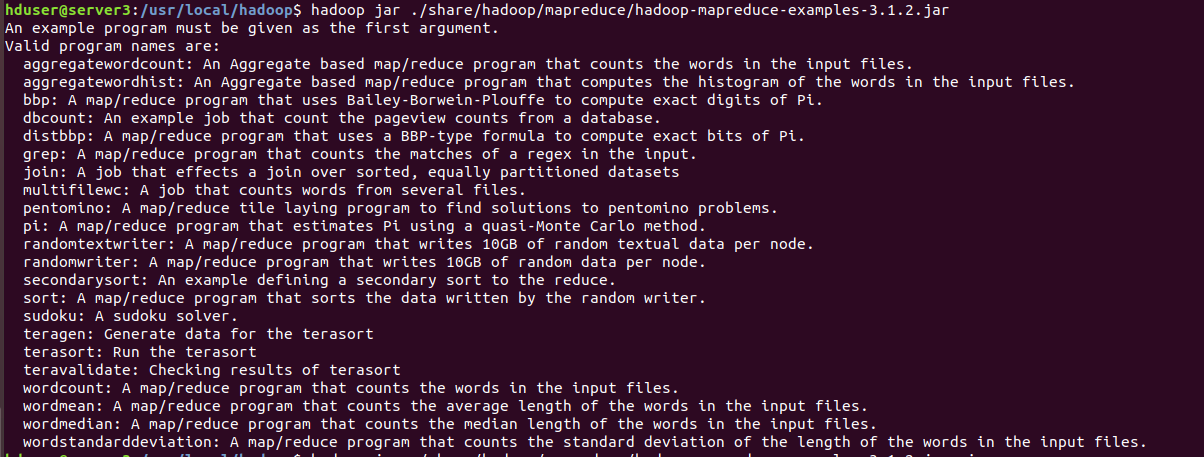
### 

### **Let’s run MapReduce job on our entirely fresh Hadoop cluster setup**

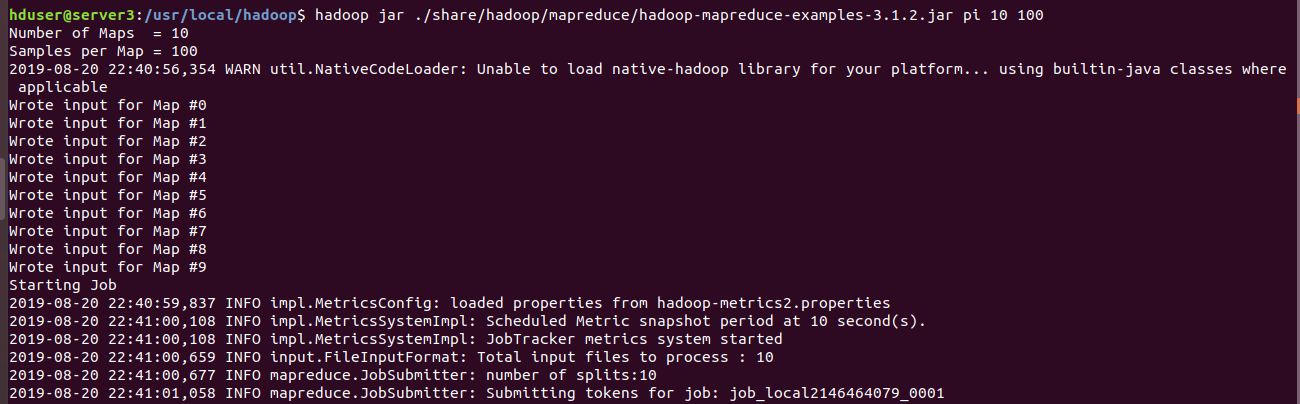
Go to the following directory

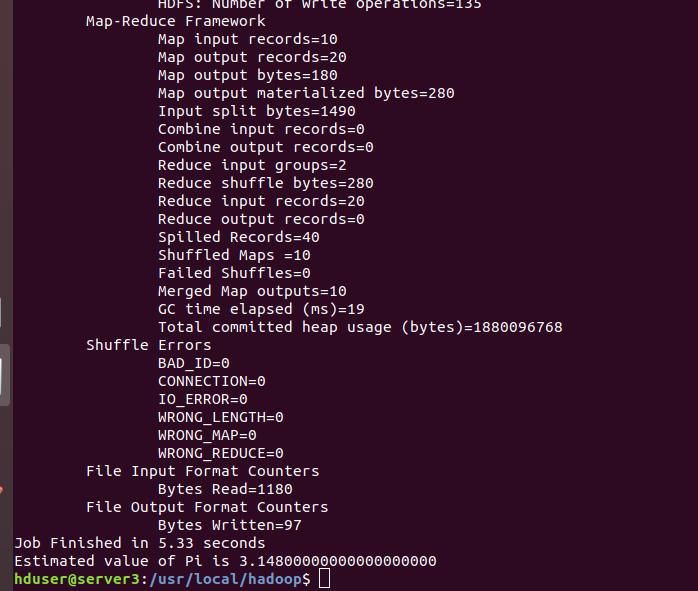
**$ cd /usr/local/hadoop**

Run the following command



[**hduser@server3**](mailto:hduser@server3) **:/usr/local/hadoop$ hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar pi 10 100**





|  |  |
| --- | --- |
| userdel hadoop | Command to delete hadoop user name |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |