**STEPS TO INSTALL HADOOP ON UBUNTU**

**1. Update and Upgrade Ubuntu**

*sudo apt-get update*

*sudo apt-get upgrade*

**2. Install the JDK**

*sudo apt install default-jdk*

**Check java version:**

*java -version*

**3. Add a user Hadoop**

*sudo adduser Hadoop*

**After creating the user, switch to the user.**

*su –hadoop*

**4. Install open-SSH and Generate SSH Key pair:**

*Do not enter the passphrase when generating keys.*

*apt install openssh-server openssh-client -y*

*ssh-keygen*

Enter your directory as: /your\_home/.ssh/id\_rsa

\*\* Replace your home as : /home/XYZ where XYZ is your user

**5. Put the public keys to Authorized keys**

*cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys*

**6. Give necessary permissions to SSH folder and authorized key file.**

*chmod 700 ~/.ssh*

*chmod 600 ~/.ssh/authorized\_keys*

**7. Allow firewall for SSH access.**

*sudo ufw allow ssh*

**8. Download Hadoop (Binary):**

*You can go to the* [*Apache Hadoop Site*](https://hadoop.apache.org/releases.html)*, and download the latest stable binary version from site.*

*Copy the link to the binary file and use the command below:*

*wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz*

**9. Untar (Unzip) the downloaded file.**

*tar –xvzf hadoop-3.4.0.tar.gz*

**10. Now we move the extracted Hadoop directory to the installation folder.**

*sudo mv hadoop-3.4.0 /usr/local/Hadoop*

**11. Create a directory for storing the system logs for Hadoop**

*sudo mkdir /usr/local/hadoop/logs*

**12. Change the ownership of Hadoop Directory.**

*sudo chown -R hadoop:hadoop /usr/local/Hadoop*

**13. Configuring the environment variables for Hadoop.**

*sudo vi ~/.bashrc*

***ADD the following lines at the bottom of this file:***   
export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

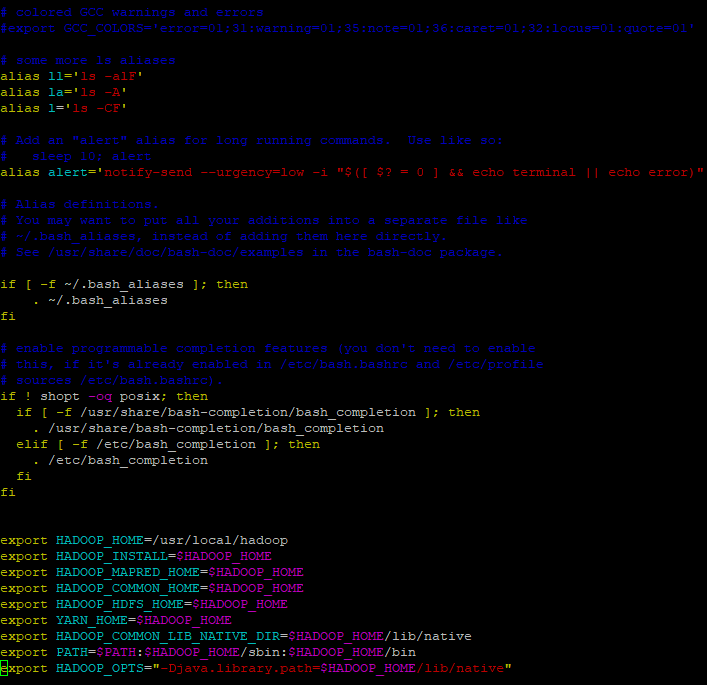
export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

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

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

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



*Save and close the file.*

**14. Make the variables effective**

*source ~/.bashrc*

**15. Find path to Java Compiler**

*which javac*

**16. Determine the open JDK location by using the location from the output of above command.**

*readlink -f /usr/bin/javac*

**17. Copy the path found in the output of above command (upto open-jdk directory) and open the file:**

*vi $HADOOP\_HOME/etc/hadoop/hadoop-env.shop/hadoop-env.sh*

*and write the below lines at the bottom:*

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

export HADOOP\_CLASSPATH+=" $HADOOP\_HOME/lib/\*.jar"

*## Remember your JAVA\_HOME variable should be the one you copied(upto open-jdk directory as shown above)*

*SAVE AND CLOSE THE FILE.*

**18. Navigate to Hadoop lib directory**

*cd /usr/local/hadoop/lib*

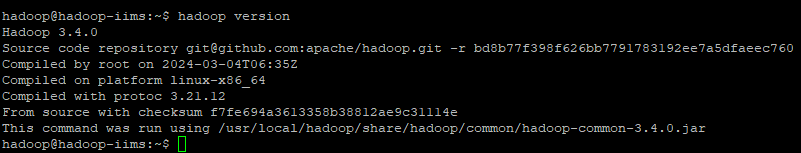
**19. Use the wget command to download the Javax activation file:**

*sudo wget* [*https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar*](https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar)

**20. Verify the Hadoop installation**

*hadoop version*

You shall get output as:



**21. Make necessary configuration settings to specify the URL of Namenode.**

*sudo vi $HADOOP\_HOME/etc/hadoop/core-site.xml*

*Add the following lines inside configuration part:*

<configuration>

<property>

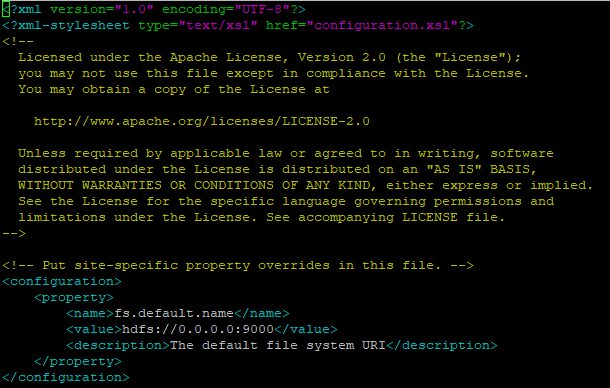
<name>fs.default.name</name>

<value>hdfs://0.0.0.0:9000</value>

<description>The default file system URI</description>

</property>

</configuration>



**22. Create a directory for storing node metadata and change the ownership to Hadoop**

*sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}*

*sudo chown -R hadoop:hadoop /home/hadoop/hdfs*

**23. Edit the hdfs-site.xml configuration file to define the location for storing node metadata and the replication factor**

*sudo vi $HADOOP\_HOME/etc/hadoop/hdfs-site.xml and add the following lines:*

*Add the following lines inside configuration part:*

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:///home/hadoop/hdfs/namenode</value>

</property>

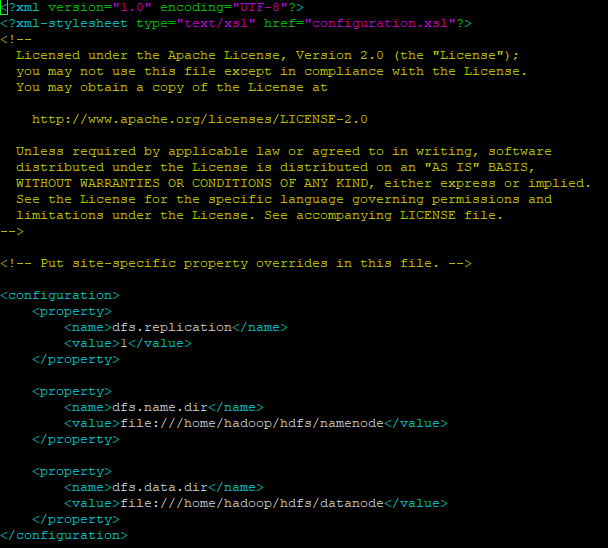
<property>

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

<value>file:///home/hadoop/hdfs/datanode</value>

</property>

</configuration>



**24. Edit the mapred-site.xml configuration file to define MapReduce values.**

*Add the following lines inside configuration part:*

sudo vi $HADOOP\_HOME/etc/hadoop/mapred-site.xml

And add the following lines:

<configuration>

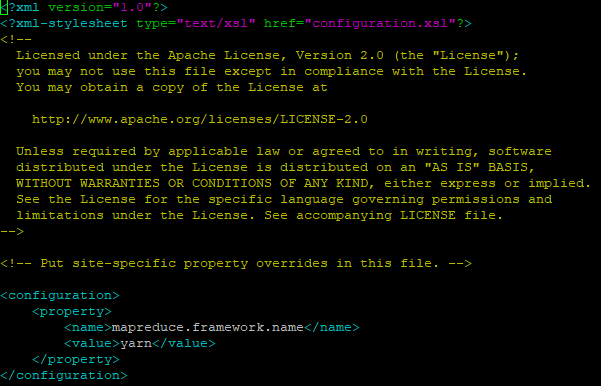
<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>



**25. Edit the yarn-site.xml configuration file and define YARN-related settings.**

*sudo vi $HADOOP\_HOME/etc/hadoop/yarn-site.xml*

*And add the following lines:*

<configuration>

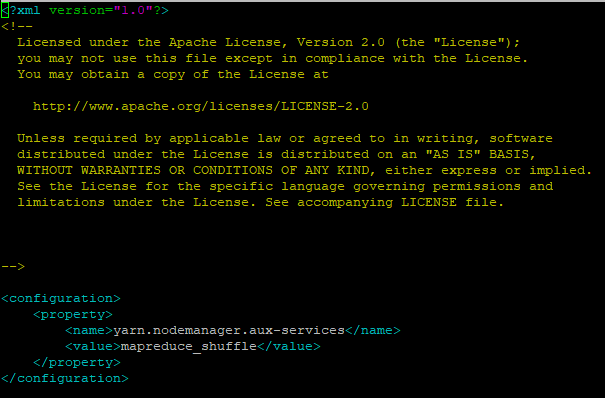
<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>



**26. Validate Hadoop Configuration. This command wipes out all existing data in the Namenode, effectively setting up a new HDFS file system. As a result, it also deletes the file system metadata, including files, directories, and block locations.**

*hdfs namenode –format*

**27. Start the Namenode and Datanode:**

*start-dfs.sh*

THE OUTPUT SHOULD LOOK SOMEWHAT LIKE THIS:

Starting namenode on [namenode\_host]... started

Starting secondarynamenode on [secondarynamenode\_host]... started

Starting datanode on [datanode\_host1]... started

... (similar messages for all datanodes)

**28. Start YARN**

*start-yarn.sh*

You should get something like:

starting yarn daemons

starting resourcemanager, logging to /path/to/yarn-resourcemanager.log ... started

starting nodemanager on [nodemanager\_host1]... started

... (similar messages for all nodemanagers)

**29. Verify all the running components**

*jps*

You should get something like:

3214 SecondaryNameNode

4320 Jps

3854 Resourcemanager

3456 DataNode

4084 NodeManager

3274 NameNode

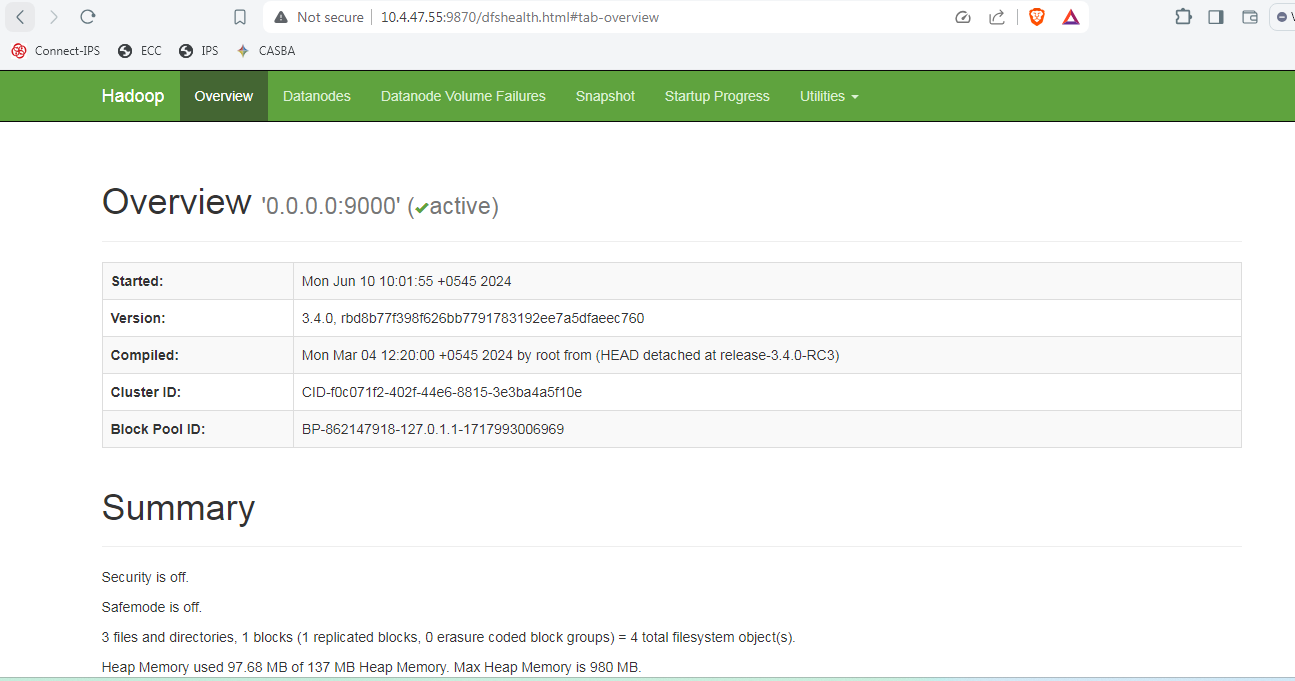
**30. Access HDFS on command line**

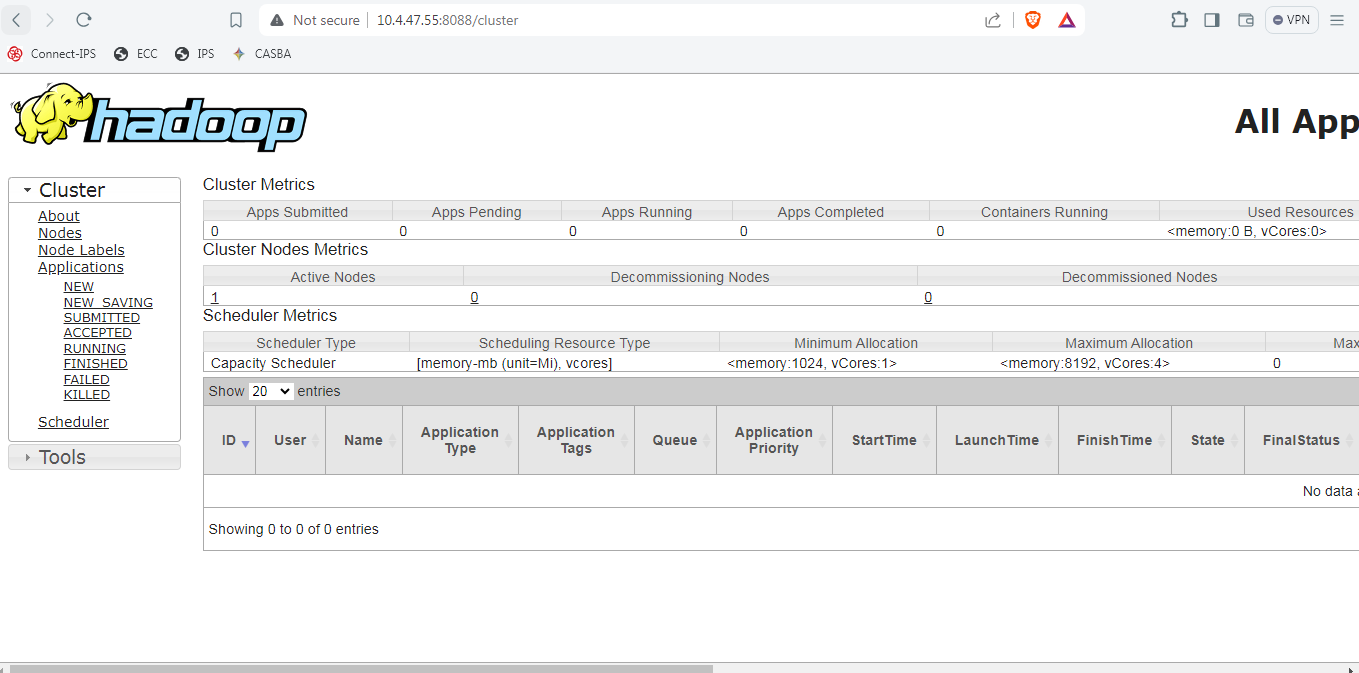
a. *ssh localhost*

b. Start by:

*hdfs dfs –mkdir /xyz*

**31. You can open a Web browser to access Hadoop’s NameNode (http://localhost:9870) and ResourceManager (http://localhost:8088) interfaces**

****

****

**\*\*END OF MANUAL\*\***

**TO RUN MAPREDUCE JOBS:**

edit the **mapred-site.xml** file:

Add following properties:

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

<property>

<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

Now for all the above 3 properties, replace the path /usr/local/Hadoop with your Hadoop Home Address.

Now add the 5th property :

1. <property> <name>mapreduce.application.classpath</name> <value></value> </property>

The **<value>** element must be filled with content after executing below commands (up to yarn/\*) in your terminal:

1. export HADOOP\_CLASSPATH=$(hadoop classpath)
2. echo $HADOOP\_CLASSPATH

My output for above command is: /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/\*:/usr/local/hadoop/share/hadoop/common/\*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/hadoop/hdfs/lib/\*:/usr/local/hadoop/share/hadoop/hdfs/\*:/usr/local/hadoop/share/hadoop/mapreduce/\*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop/yarn/lib/\*:/usr/local/hadoop/share/hadoop/yarn/\*:/usr/local/hadoop/lib/javax.activation-api-1.2.0.jar

**My final mapred-sire.xml is:**

**<configuration>**

**<property>**

**<name>mapreduce.framework.name</name>**

**<value>yarn</value>**

**</property>**

**<property>**

**<name>yarn.app.mapreduce.am.env</name>**

**<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>**

**</property>**

**<property>**

**<name>mapreduce.map.env</name>**

**<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>**

**</property>**

**<property>**

**<name>mapreduce.reduce.env</name>**

**<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>**

**</property>**

**<property>**

**<name>mapreduce.application.classpath</name>**

**<value>/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/\*:/usr/local/hadoop/share/hadoop/common/\*:/usr/local/hadoop/share/hadoop/hdfs:/usr/local/hadoop/share/hadoop/hdfs/lib/\*:/usr/local/hadoop/share/hadoop/hdfs/\*:/usr/local/hadoop/share/hadoop/mapreduce/\*:/usr/local/hadoop/share/hadoop/yarn:/usr/local/hadoop/share/hadoop/yarn/lib/\*:/usr/local/hadoop/share/hadoop/yarn/\*</value>**

**</property>**

**</configuration>**

**SAMPLE COMMAND:**

**hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar wordcount /abhi/mysales.txt /output/part\_1**