Hadoop Installation Steps

https://phoenixnap.com/kb/install-hadoop-ubuntu

https://blog.devgenius.io/install-configure-and-setup-hadoop-in-ubuntu-a3cdd6305a0e

1. Open terminal and copy command as below

```
sudo apt update


sudo apt install openjdk-11-jdk -y

sudo apt-get install jdk-update

java -version
```

2. Type and execute given below command

```
sudo apt install openssh-server openssh-client -y

sudo adduser hadoop

sudo usermod -aG sudo hadoop

su hadoop (to change user in terminal)
```

3. Enable Passwordless SSH for Hadoop User

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 0600 ~/.ssh/authorized_keys

ssh localhost
```

4. Download and Install Hadoop

sudo apt-get install wget (optional)

```
wget
https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

tar xzf hadoop-3.2.1.tar.gz

sudo mv hadoop-3.2.1 /usr/local/hadoop

sudo mkdir /usr/local/hadoop/logs

sudo chown -R hadoop:hadoop /usr/local/hadoop
```

5. **Configure Hadoop**

sudo nano ~/.bashrc (type command in terminal)

Make following changes in the same file and save

```
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”
```

source ~/.bashrc (to save bashrc file)

6. **Configure Java Environmental variables**

```
    which javac

    readlink -f /usr/bin/javac
```

7. **Edit Hadoop-env.sh file**

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_CLASSPATH+=” $HADOOP_HOME/lib/*.jar”

hadoop version
```

8. JAVAX Activation

```
cd /usr/local/hadoop/lib
```

```
sudo wget
https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
```

```
hadoop version
```

9. **Edit core-site.xml**

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://0.0.0.0:9000</value>
<description>The default file system URI</description>
</property>
</configuration>
```

10. **Make a directory for node metadata storage and give it hadoop’s ownership**

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

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

11. **Edit hdfs-site.xml, mapred-site.xml and yarn-site.xml**

hdfs-site.xml

```
<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>
```

mapred-site.xml

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

yarn-site.xml

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

12. **Format the HDFS NameNode and validate the Hadoop configuration.**

sudo su – hadoop **(switch to hadoop user)**

hdfs namenode -format **(format name node)**

13. Launch the Apache Hadoop Cluster

```
start-dfs.sh

start-yarn.sh

jps
```

Example : http://192.168.10.12:9870/

Hadoop Commands

1. jps
2. hadoop fs -usage [command name]
3. hadoop fs -help [command name]
4. hadoop fs -ls
5. hadoop fs -ls g (name of directory)
6. hadoop fs -mkdir **temp**
7. hadoop fs -touchz **f1.txt**
8. hadoop fs -copyFromLocal **f1.txt temp**
9. hadoop fs -cat **temp/f1.txt**
10. hadoop fs -copyToLocal **temp/f1.txt ~/Desktop**
11. hadoop fs -mkdir **temp1**

    hadoop fs -cp **temp/f1.txt temp1/**
12. hadoop fs -mkdir **temp2**

    hadoop fs -mv **temp/f1.txt temp2/**
13. hadoop fs -du temp
14. hadoop fs -dus temp
15. hadoop fs -test -d temp [checks for a given destination]
    echo $?
16. hadoop fs -text -e temp [checks for an exists or not]
    echo $?

17. hadoop fs -text -f temp [checks for a file or not]
    echo $?

18. hadoop fs -ls temp/

19. hadoop fs -text -z temp/f1.txt [checks for a file size zero or not]
    echo $?

20. hadoop fs -moveFromLocal ~/Documents/demo.txt temp/

21. hadoop fs -getmerge -nl temp/f1.txt temp/f2.txt ~/Documents/merge.txt
    cat Documents/merge.txt

22. hadoop fs -appendToFile f1.txt f2.txt f3.txt
    hadoop fs -cat f3.txt

23. hadoop fs -checksum temp/f1.txt   (to check integrity of a file)

24. hadoop fsck - /

25. hadoop fs -count temp/

26. hadoop fs -rm temp/f3.txt

27. hadoop fs -chgrp test temp/f1.txt

28. hadoop fs -stat %b temp/f1.txt [checks file size in bytes]

29. hadoop fs -stat %g temp/f1.txt [checks group name]

30. hadoop fs -stat %r temp/f1.txt   [checks replication factor]

31. hadoop fs -stat %u temp/f1.txt [checks user name]

32. hadoop fs -stat %y temp/f1.txt   [checks for last modified file]

33. hadoop fs -head temp/f1.txt [displays first 1 kb file content]

34. hadoop fs -tail temp/f1.txt [displays last 1 kb file content]

35. **hadoop fs -expunge** [clear trash of HDFS]

36. hadoop fs -chown hadoop:iiit temp/f1.txt [change user of file]

37. hadoop fs -chmod 777 temp/f1.txt [change file access permissions]

38. **hadoop fs -setrep -w 3 temp/f1.txt** [Sets replication factor and waiting for some time period till getting updated replication factor]

39. hadoop fs -truncate -w 100 temp/filename [change file size in HDFS]
    Used for change lengthy log file size

40. hadoop fs -setfattr -n ‘user.ap’/’trusted’/’system’/’security’ -v thisisadummyfile /temp/f1.txt

    set attributes = to add extra information in files

41. hadoop fs -getfattr -d temp/f1.txt

    -d = scanning of set attributes