**Pre-Requisite:** - virtual box and ubuntu

>> su root

**Step1:** **download java (JDK is important for running ubuntu machine).**

1). Write this command on terminal (VM):

>> sudo apt-get install openjdk-8-jre-headless

If it shows, username not in the sudoers file.the incident will be reported follow this:-

>> su

>> apt install sudo

>> usermod -aG sudo username (of the VM)

>> reboot

Then, start from :-

>> sudo apt-get install openjdk-8-jre-headless

2). To check if it is downloaded or not:

>> java -version

3). To check if it is in correct location:

>> files/other location/Computer/usr/lib/jvm (Check manually in File Directory)

**step2: copy paste the link in browser of ubuntu machine (VM).**

<https://hadoop.apache.org/releases.html>

version to download - **3.2.4 - binary**

**Note - a tar.gz file will be downloaded in download folder, cut it from download folder and paste it on home folder after that extract it in home folder.**

**Creating of SSH (Secured Shell)** [SSH is important for configuring Hadoop because it consists of Master-slave relationship, there would be a lot of internal communication, so we want to setup a password less communication to make it efficient and save time to authenticate it again and again.]

step 3: terminal command

>> sudo apt-get install ssh

step 4: terminal command

>> ssh-keygen -t rsa -P ""

**[It will generate SSH certificates, RSA is an Algorithm for Public Cryptography. So, we generate public-private RSA key pair.]**

It will ask to give a File Location: [Just Press Enter]

step5: terminal command

>> cat /home/souvik/.ssh/id\_rsa.pub >> /home/souvik/.ssh/authorized\_keys

**[This command adds/appends public key (.pub) to authorized keys]**

**Note – Rename souvik with your Virtual Machine (VM) Username.**

**Setup Configuration Files**

step6: terminal command

>>gedit ~/.bashrc **[ bashrc is a central repository where we have to give the path of all the software in our linux.]**

1). A new file(repository) will open with this command.

*-- We need to add the Hadoop environment variables at the end of the repository. --*

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP\_INSTALL=/home/souvik/hadoop-3.2.4

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

export PATH=$PATH:$HADOOP\_INSTALL/sbin6.cat /home/souvik/.ssh/id\_rsa.pub >> /home/souvik/.ssh/authorized\_keys **[Change souvik with your VM’s Name.]**

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

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

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

step 7: Go to this location –

files/home/hadoop3.2.4/hadoop/etc/hadoop-env.sh

make this particular change on line no 36 after removing # symbol

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

step 8: terminal command:

>> sudo mkdir -p /home/souvik/hadoop-3.2.4/tmp [Rename Souvik with your VM’s Name]

**The Core-site file contains the configuration properties that Hadoop uses when it’s starting up.**

Step 9: Go to this location –

files/home/hadoop3.2.4/hadoop/etc/core-site.xml [Between <configuration> </configuration>]

**Modify the configuration: -**

<property>

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

<value>/home/shivani/hadoop-3.2.4/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>

**Here, we are creating a Temporary Directory in core-site because whenever the Hadoop is starting up, it must store some files in temporary space so that we should be having some space ready for Hadoop.**

**Now, we check whether Hadoop has been recognized or not: -**

>> cd $HADOOP\_INSTALL [ As, we wrote this in bashrc file, export HADOOP\_INSTALL=/home/souvik/hadoop-3.2.4]

**We give permission to this folder to avoid hindrance afterwards: -**

>> sudo chmod 777 /home/souvik/hadoop-3.2.4/tmp [ 777 is for top level privilege to the tmp]

Step9: Go to this location –

files/home/hadoop3.2.4/hadoop/etc/ mapred-site.xml [Between <configuration> </configuration>] [ In Hadoop, the algorithm which is working behind is MapReduce. So, we need to overwrite its configuration.]

**Modify the configuration: -**

<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

<description>The host and port that the MapReduce job tracker runs

at. If "local", then jobs are run in-process as a single map

and reduce task.

</description>

</property>

**Next, we need to configure HDFS (Hadoop Distributed file system).**

**Note: >**> cd ~ [It will take you back to the home location]

Step10: command on terminal

>> sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode

>> sudo mkdir -p /usr/local/hadoop\_store/hdfs/datanode

**[We are creating these two directories using mkdir with location and -p is creating sub-directories in one go.]**

**After that, we give privilege to this folder,**

>>sudo chmod 777/usr/local/hadoop\_store/hdfs/namenode

>> sudo chmod 777/usr/local/hadoop\_store/hdfs/datanode

Go to this location –

files/home/hadoop3.2.4/hadoop/etc/hdfs-site.xml xml [Between <configuration> </configuration>]

**Modify the configuration: -**

<property>

<name>dfs.replication</name>

<value>1</value>

</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>

**This are concept of replication factor that the data is going to be copied to the multiple location, we are keeping replication factor value as 1 because we are working in the single node installation, and we are providing the value to NameNode and dataNode directory respectively.**

**Now, We need to format the newly created file systems, 1 local and another 1 hdfs.**

Step11: terminal command

>> hadoop namenode -format

>> start-all.sh [ We are opening the daemons the background processes of Hadoop, i.e., JobTracker and TaskTracker for MapReduce and NameNode, DataNode and secondary Namenode for HDFS]

>> jps