# **April 28, 2021 LAB\_5:**

## **Installation of Hadoop 3.1.4 in Ubuntu 20.04 LTS**

### Step 1: Install of openjdk-8, vim & ssh(server and client) on terminal

$ sudo apt install openjdk-8-jdk openjdk-8-jre

$ java -version

*// to verify the correct version is installed*

$ sudo apt install vim openssh-server openssh-client

### Step 2: Adding the jdk path to the PATH variable

$ sudo vim ~/.bashrc

*// open ~/.bashrc and go to the last line and add the following*

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

**export PATH=$PATH:$JAVA\_HOME**

*// save the file and exit (Esc then :x)*

$ source ~/.bashrc

*// inform the os about the modification*

*// now type the following commands to verify*

$ echo $JAVA\_HOME

$ echo $PATH

### Step 3: Add a dedicated user for the HADOOP

$ sudo adduser hadoop

$ sudo usermod -aG sudo hadoop

### Step 4: Once the user is added, login to the user “Hadoop” to generate the ssh key for passwordless login ( hadoop@machinename)

$ sudo su - hadoop

$ ssh-keygen -t rsa

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

$ chmod 0600 ~/.ssh/authorized\_keys

*// check the login to localhost using ssh is valid*

$ ssh localhost

*// once the connection is made, logout from ssh*

$ exit

### Step 5: Download the latest binary from Hadoop site “**hadoop-3.1.3.tar.gz**”

$ tar -xvzf hadoop-3.1.3.tar.gz

*// download and decompress the gzip file*

$ mv hadoop-3.1.3 /usr/local/hadoop

*// move the extracted file to the following directory*

### Step 6: Setup the path variables for hadoop

$ sudo vim /etc/profile.d/hadoop\_java.sh

*//open the file and add the following lines to it*

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

**export HADOOP\_HOME=/usr/local/hadoop**

**export HADOOP\_HDFS\_HOME=$HADOOP\_HOME**

**export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME**

**export YARN\_HOME=$HADOOP\_HOME**

**export HADOOP\_COMMON\_HOME=$HADOOP\_HOME**

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

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

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

*// save the file and exit (Esc then :x)*

*// then source the file*

$ source /etc/profile.d/hadoop\_java.sh

*// confirm your hadoop and hdfs version*

$ hadoop version

$ hdfs version

### Step 7: Configuring Hadoop

*// navigate to /usr/local/hadoop/etc/hadoop and type ls*

$ cd /usr/local/hadoop/etc/hadoop

$ hadoop@machine: /usr/local/hadoop/etc/hadoop: ls

*// give the permission for the hadoop folder to hadoop user*

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

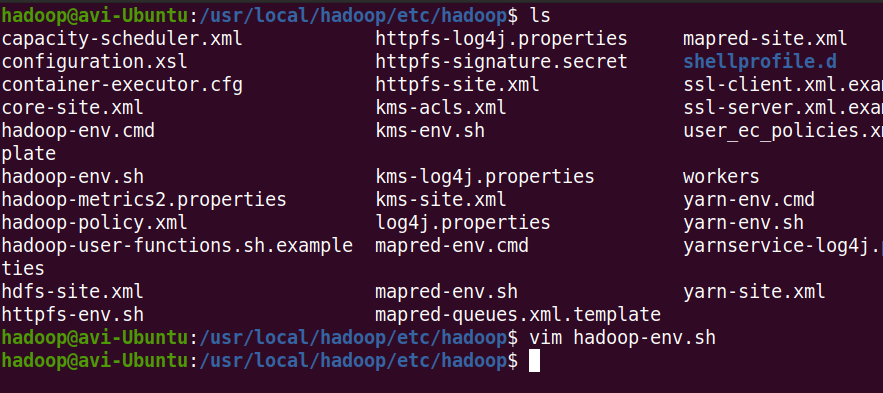
#### Step 7a: Specify JAVA\_HOME in hadoop-env.sh (/usr/local/hadoop/etc/hadoop)

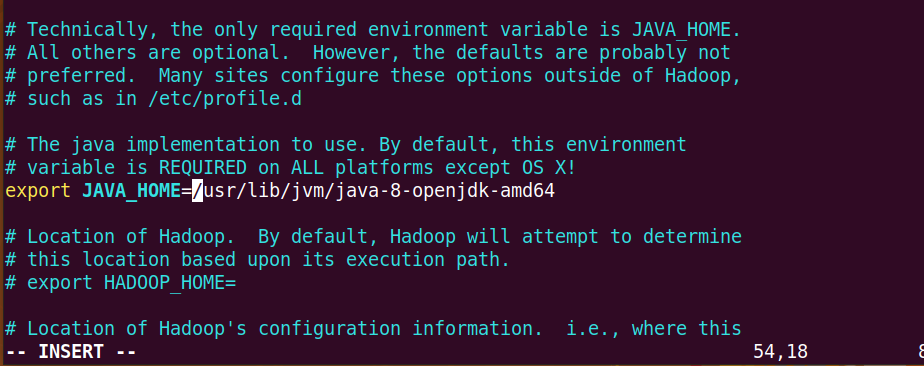
$ vim hadoop-env.sh

*// open the file and add the following line in java implementation (line 54)*

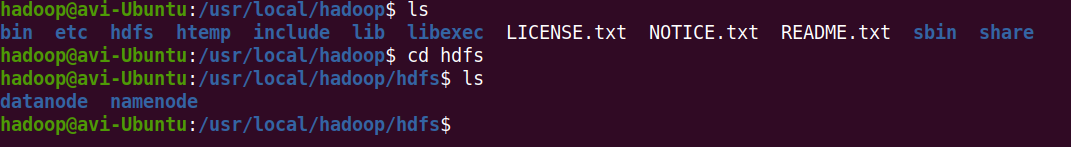
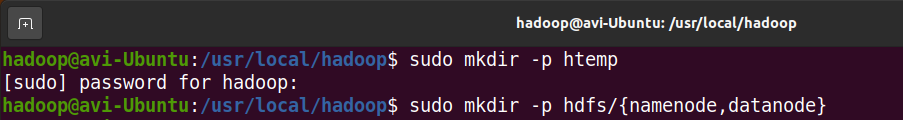
**export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

*// save the file and exit (Esc then :x)*





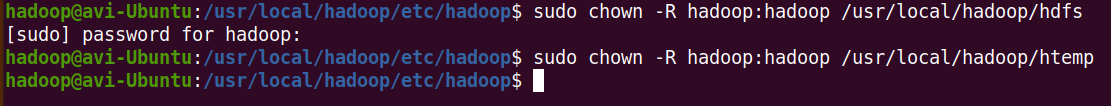
#### Step 7b: Create two directories for hdfs and htemp in /usr/local/hadoop and then create namenode and datanode directories in hdfs



*// give the permission for the hdfs and htemp folder to hadoop user*

$ sudo chown -R hadoop:hadoop /usr/local/hadoop/hdfs

$ sudo chown -R hadoop:hadoop /usr/local/hadoop/htemp



#### Step 7b: Modify core-site.xml to setup web portal for hadoop

*//open core-site.xml and add the following lines to it then save it*

**<configuration>**

**<property>**

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

**<value>hdfs://localhost:9000</value>**

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

**</property>**

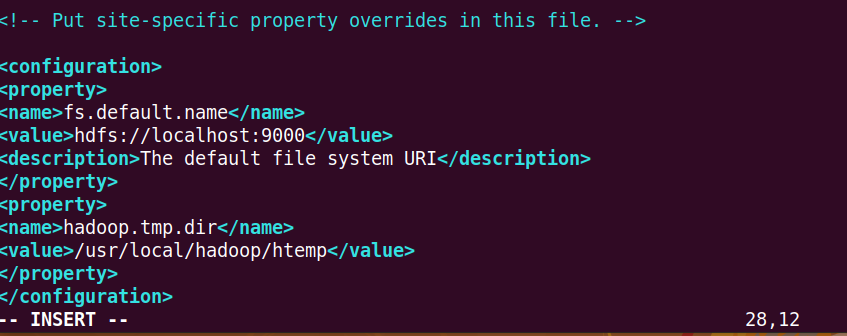
**<property>**

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

**<value>/usr/local/hadoop/htemp</value>**

**</property>**

**</configuration>**



#### Step 7c: Modify hdfs-site.xml and add the following lines inside

*//open hdfs-site.xml and add the following lines to it then save it*

**<configuration>**

**<property>**

**<name>dfs.replication</name>**

**<value>1</value>**

**</property>**

**<property>**

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

**<value>file:/usr/local/hadoop/hdfs/namenode</value>**

**</property>**

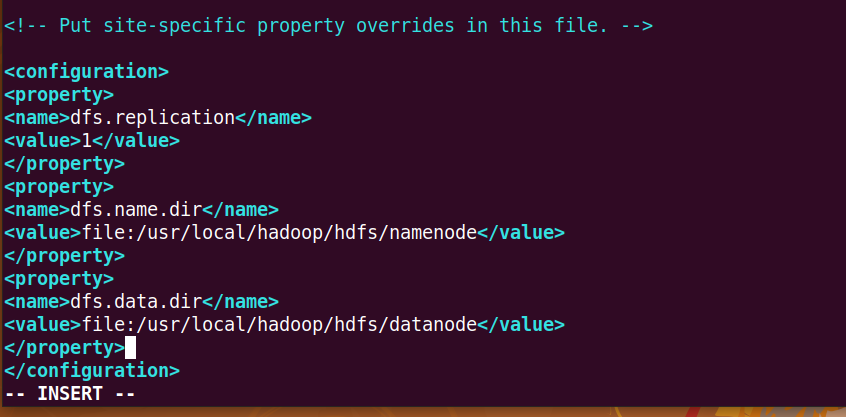
**<property>**

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

**<value>file:/usr/local/hadoop/hdfs/datanode</value>**

**</property>**

**</configuration>**

****

#### Step 7d: Configure the mapreduce framework by editing the mapred-site.xml

*// modify the mapred-site.xml and add the following lines then save it*

**<configuration>**

**<property>**

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

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

**</property>**

**<property>**

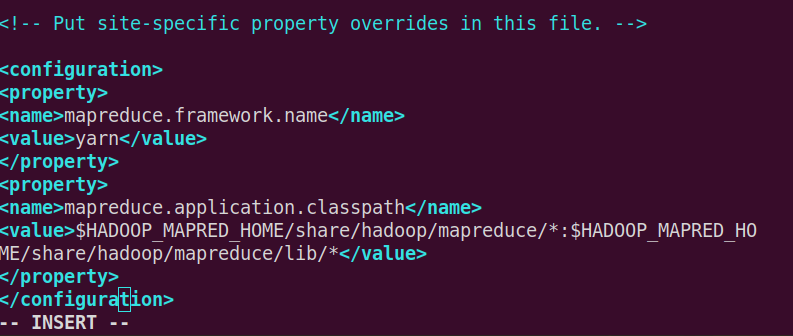
**<name>mapreduce.application.classpath</name>**

**<value>$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/\*:$HADOOP\_MAPRED\_HO**

**ME/share/hadoop/mapreduce/lib/\*</value>**

**</property>**

**</configuration>**

****

#### Step 7e: Configure the YARN resource manager by editing the yarn-site.xml

*// modify the yarn-site.xml and add the following lines then save it*

**<configuration>**

**<property>**

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

**<value>mapreduce\_shuffle</value>**

**</property>**

**<property>**

**<name>yarn.nodemanager.env-whitelist</name>**

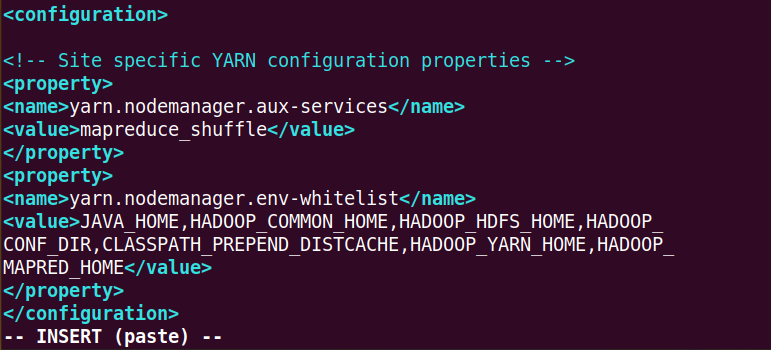
**<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_**

**CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_**

**MAPRED\_HOME</value>**

**</property>**

**</configuration>**

****

### Step 8: Format the namenode using the command

$ hdfs namenode -format

*// test hdfs configuration by opening these files (inside /usr/local/hadoop/sbin/)*

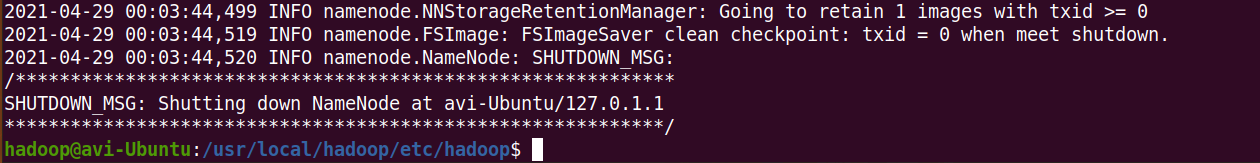
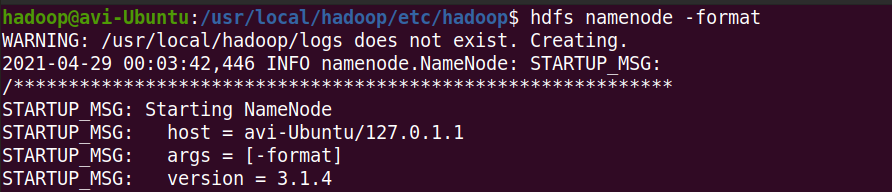
$ ./start-dfs.sh

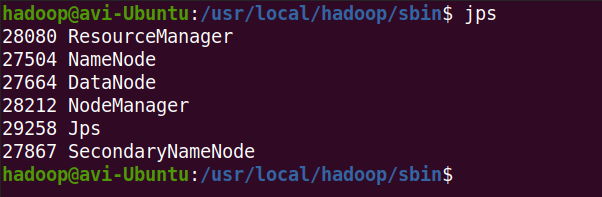
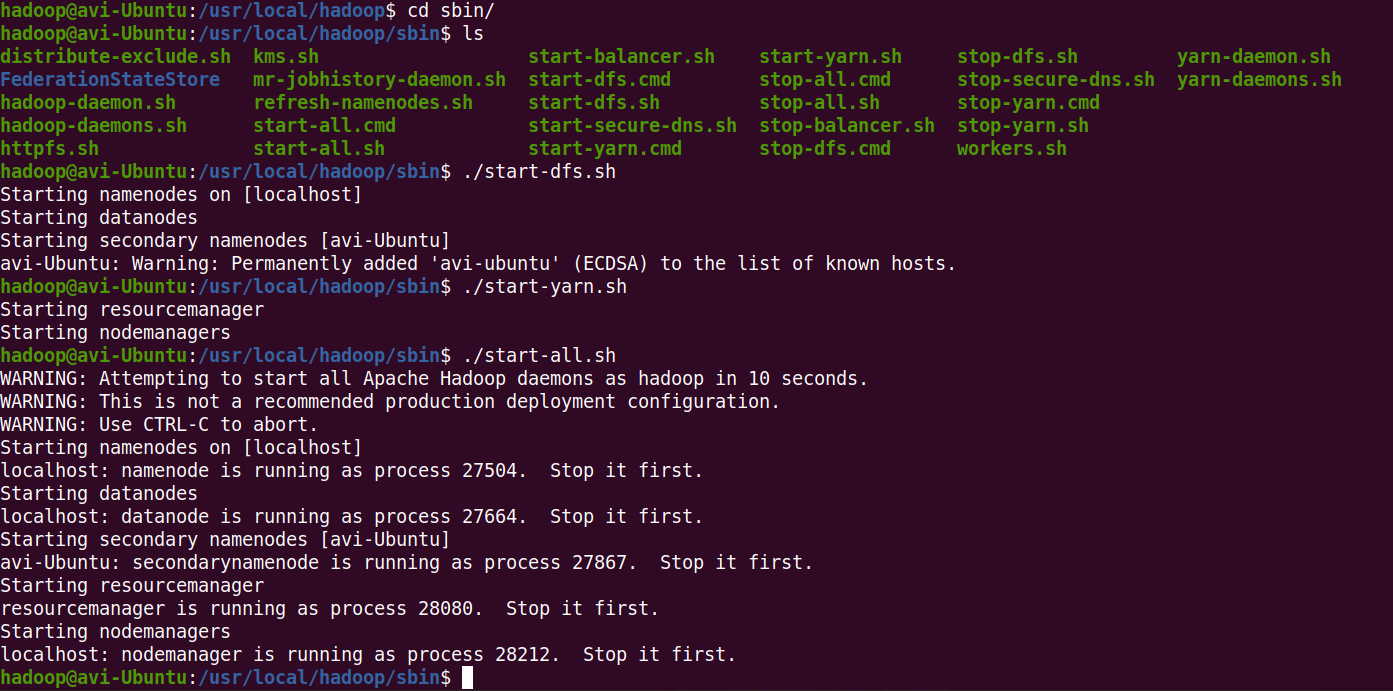
$ ./start-yarn.sh

$ ./start-all.sh

*// check the availability of all the nodes by typing*

$ jps





### 

### 

### 

### 

### 