**Steps to install Hadoop:**

1. Make sure java is installed.

**java -version**

If java is not installed, then type in the following commands:

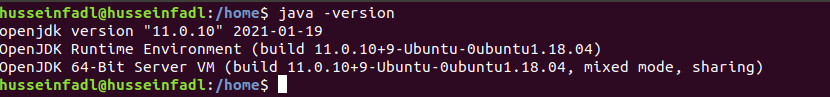
**sudo apt-get install update**

**sudo apt-get update**

**sudo apt-get install default-jdk**

Make sure now java is installed.

**java -version**



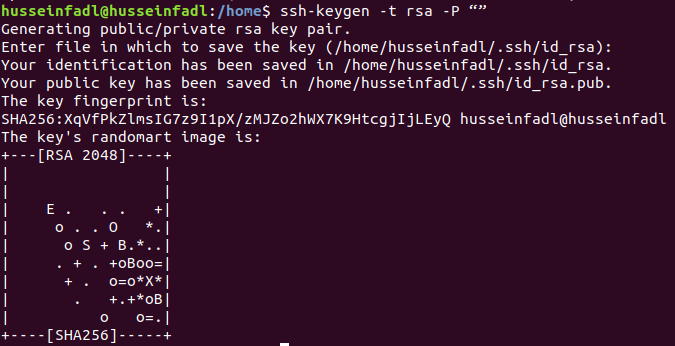
1. Install ssh server

**sudo apt-get install ssh-server**

Generate public/private RSA key pair.

**ssh-keygen -t rsa -P “”**

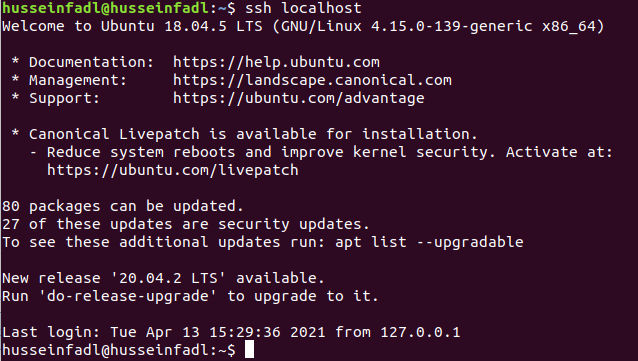
When prompted for the file name to save the key, press Enter (leave it blank).

Type the following commands:

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

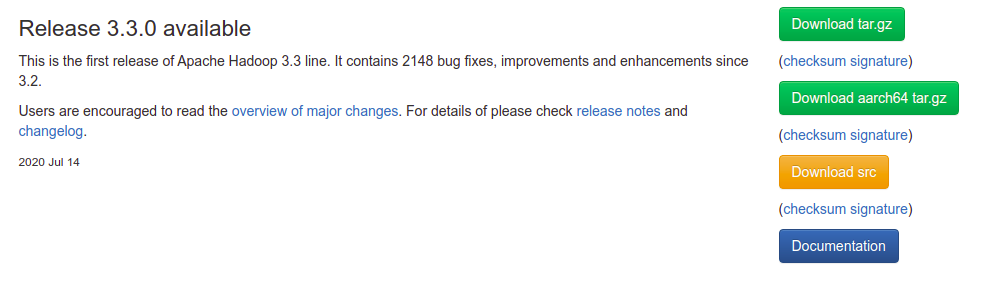
**ssh localhost**

**exit**

****

****

1. Install Hadoop by navigating to the following link and downloading the tar.gz file for Hadoop version 3.3.0 (or a later version if you wish). (478 MB)

<https://hadoop.apache.org/release/3.3.0.html>

1. Once downloaded, open the terminal and cd to the directory where it is downloaded (assume the desktop for example) and extract it as follows:

**cd Desktop**

**sudo tar -xvzf hadoop-3.3.0.tar.gz**

You can now check that there is an extracted file named hadoop-3.3.0 by typing the command “ls” or by visually inspecting the files.

1. Now, we move the extracted file to the location /usr/local/hadoop

**sudo mv hadoop-3.3.0 /usr/local/hadoop**

1. Let’s configure the hadoop system.

Type the following command:

**sudo gedit ~/.bashrc**

At the end of the file, add the following lines: (Note: Replace the java version with the version number you already have. You can navigate to the directory /usr/lib/jvm and check the file name java-xx-openjdk-amd64)

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

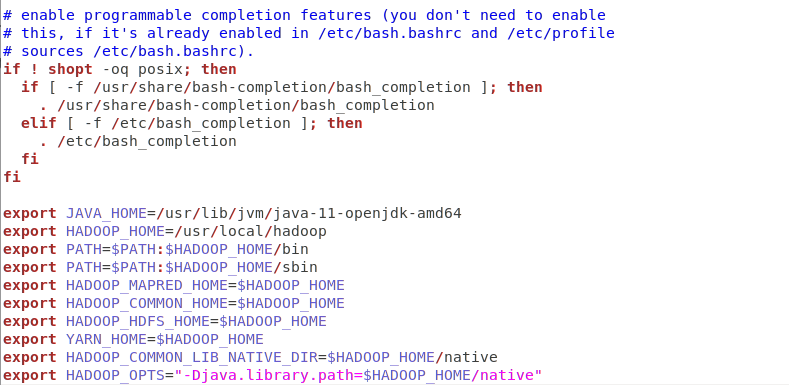
export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

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/native

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

1. Save the file and close it.
2. Now from the terminal, type the following command:

**source ~/.bashrc**

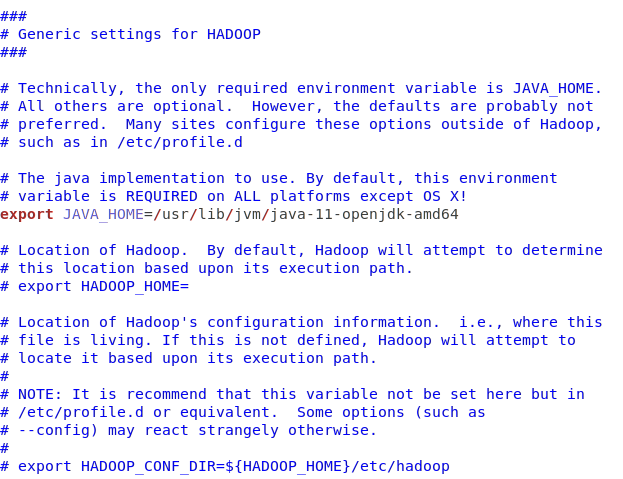
1. We start configuring Hadoop by opening **hadoop-env.sh** as follows:

**sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

Search for the line starting with **export JAVA\_HOME=** and replace it with the following line.

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

Save the file by clicking on “Save” or (Ctrl+S)

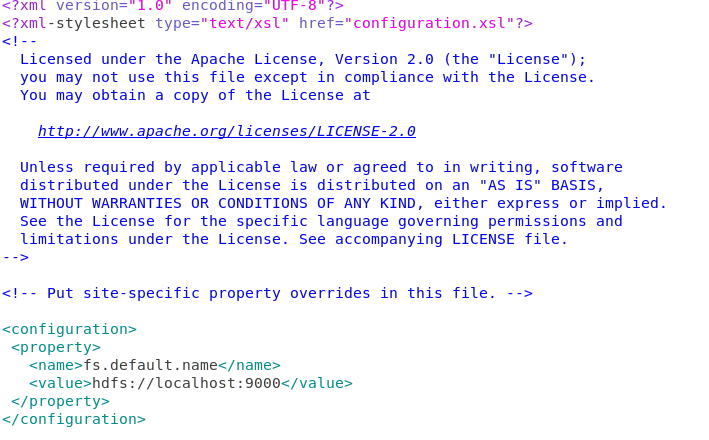
****

1. Open **core-site.xml** as follows:

**sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml**

Add the following lines between the tags <configuration> and </configuration> and save it (Ctrl+S).

**<property>  
 <name>fs.default.name</name>  
 <value>hdfs://localhost:9000</value>  
</property>**



1. Open **hdfs-site.xml** as follows:

**sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

Add the following lines between the tags <configuration> and </configuration> and save it (Ctrl+S).

**<property>**

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

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

**</property>**

**<property>**

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

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

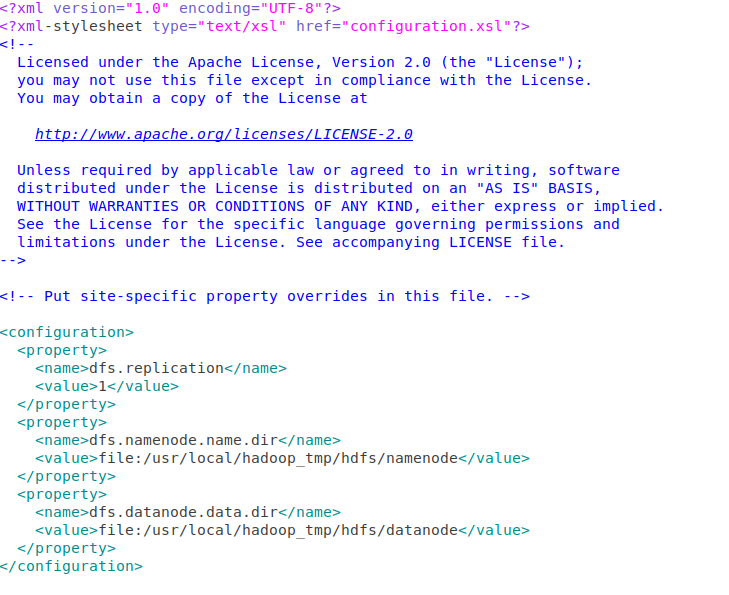
**</property>**

**<property>**

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

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

**</property>**



1. Open **yarn-site.xml** as follows:

**sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml**

Add the following lines between the tags <configuration> and </configuration> and save it (Ctrl+S)

**<property>**

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

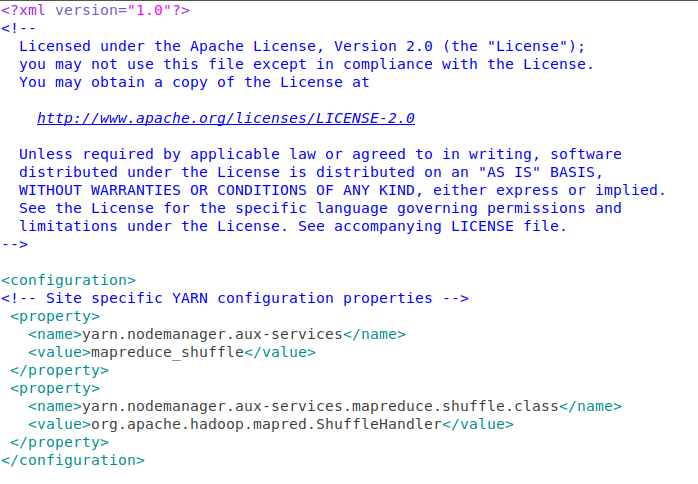
**<value>mapreduce\_shuffle</value>**

**</property>**

**<property>**

**<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>**

**<value>org.apache.hadoop.mapred.ShuffleHandler</value>**

**</property>**

1. Open **mapred-site.xml** as follows:

**sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml**

Add the following lines between the tags <configuration> and </configuration> and save it (Ctrl+S)

**<property>**

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

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

**</property>**

**<property>**

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

**<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>**

**</property>**

**<property>**

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

**<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>**

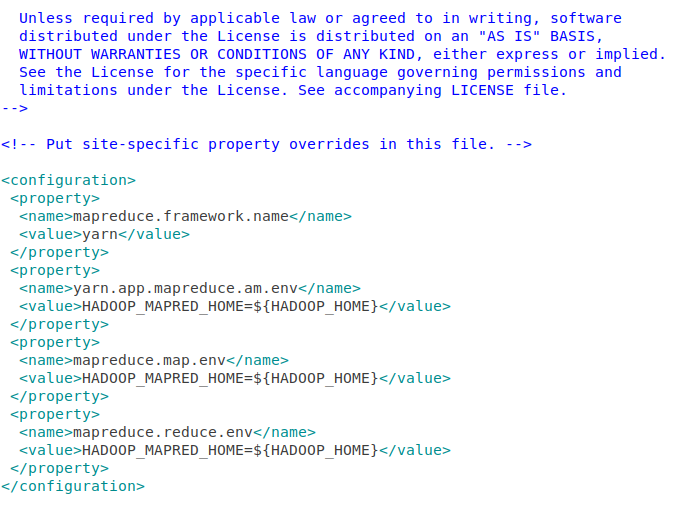
**</property>**

**<property>**

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

**<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>**

**</property>**



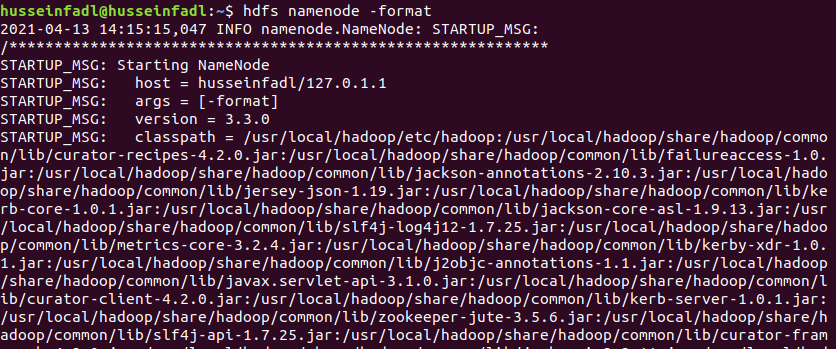
1. Now, run the following commands on the terminal to create a directory for hadoop space, name node and data node.

* **sudo mkdir -p /usr/local/hadoop\_space**
* **sudo mkdir -p /usr/local/hadoop\_space/hdfs/namenode**
* **sudo mkdir -p /usr/local/hadoop\_space/hdfs/datanode**

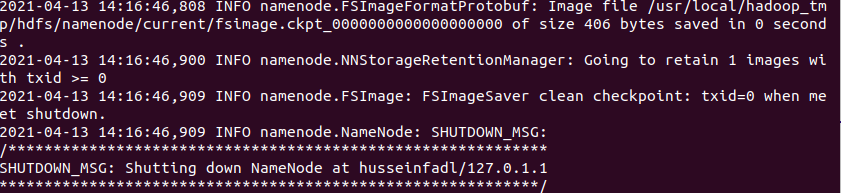
Now we have successfully installed Hadoop.

1. Format the namenode as follows:

**hdfs namenode -format**

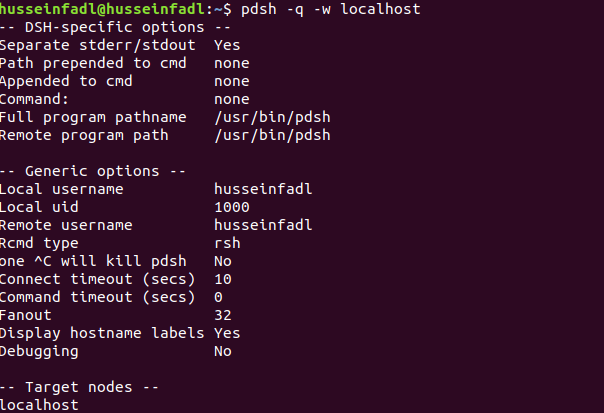
****

This step should end by shutting down the namenode as follows:



1. Before starting the Hadoop Distributed File System (hdfs), we need to make sure that the rcmd type is “ssh” not “rsh” when we type the following command

* **pdsh -q -w localhost**

****

1. If the rcmd type is “rsh” as in the above figure, type the following commands:

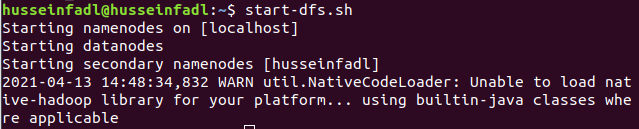
**export PDSH\_RCMD\_TYPE=ssh**

**cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys  
chmod 0600 ~/.ssh/authorized\_keys**

Run Step 16 again to check that the rcmd type is now ssh.

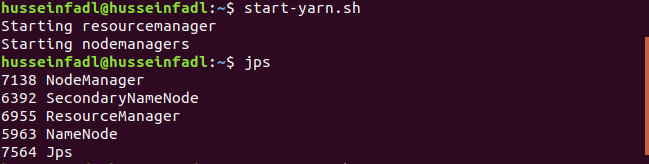
If not, skip that step.

1. Start the HDFS System using the command.

**start-dfs.sh**

1. Start the YARN using the command

**start-yarn.sh**

****

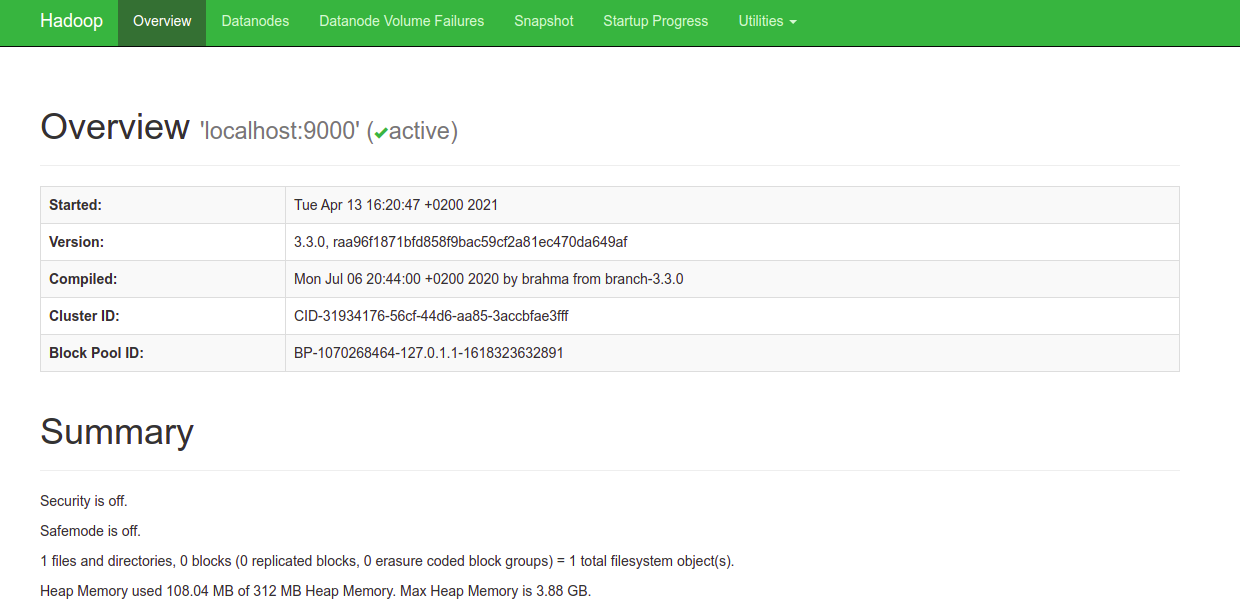
1. Type the following command. You should see an output similar to the one in the following figure.

**jps**

****

Make sure these nodes are listed: (ResourceManager, NameNode, NodeManager, SecondaryNameNode, Jps and DataNode).

1. Go to localhost:9870 from the browser. You should expect the following

****

**Steps to run WordCount Program on Hadoop:**

1. Make sure Hadoop and Java are installed properly

**hadoop version**

**javac -version**

1. Create a directory on the Desktop named Lab and inside it create two folders; one called “Input” and the other called “tutorial\_classes”.

[You can do this step using GUI normally or through terminal commands]

**cd Desktop**

**mkdir Lab**

**mkdir Lab/Input**

**mkdir Lab/tutorial\_classes**

1. Add the file attached with this document “WordCount.java” in the directory Lab
2. Add the file attached with this document “input.txt” in the directory Lab/Input.
3. Type the following command to export the hadoop classpath into bash.

**export HADOOP\_CLASSPATH=$(hadoop classpath)**

Make sure it is now exported.

**echo $HADOOP\_CLASSPATH**

1. It is time to create these directories on HDFS rather than locally. Type the following commands.

**hadoop fs -mkdir /WordCountTutorial**

**hadoop fs -mkdir /WordCountTutorial/Input**

**hadoop fs -put Lab/Input/input.txt /WordCountTutorial/Input**

1. Go to localhost:9870 from the browser, Open “Utilities → Browse File System” and you should see the directories and files we placed in the file system.
2. Then, back to local machine where we will compile the WordCount.java file. Assuming we are currently in the Desktop directory.

**cd Lab**

**javac -classpath $HADOOP\_CLASSPATH -d tutorial\_classes WordCount.java**Put the output files in one jar file (There is a dot at the end)

**jar -cvf WordCount.jar -C tutorial\_classes .**

1. Now, we run the jar file on Hadoop.

**hadoop jar WordCount.jar WordCount /WordCountTutorial/Input /WordCountTutorial/Output**

1. Output the result:

**hadoop dfs -cat /WordCountTutorial/Output/\***

**Requirement:**

Vodafone Egypt is launching a marketing campaign in Ramadan to promote their sales and increase their profit from selling the prepaid recharge cards. These cards are worth 5, 10, 15, 50, and 100 EGP.

The data science team at Vodafone are analyzing the customers’ data which include the customer personal information, the prepaid card they purchased, the timestamp they registered the prepaid amount on their Vodafone accounts, among other information.

The details of the customers are omitted, and you are only provided with a file “in.csv” which includes two columns.

1. Customer ID. (Each ID maps to a certain customer, whose data is hidden for confidentiality).
2. Prepaid Card Amount.

Your task is to generate a report using MapReduce (similar to the WordCount program) showing the total amount of prepaid cards for each customer that they have purchased. For example, if a customer with ID 300 purchased 5 cards with 10, 15, 15, 10, 100, then the report should include that customer ID 300 bought cards with a total amount of 150.



*Disclaimer: Thanks to Vodafone DS team who provided us with this real customer data.*