```
HADOOP --------------------------------------------------------
Sudo su
Sudo apt-get update


Sudo apt-get install openjdk-8-jdk
Sudo java -version
Sudo update-alternatives –config java
Sudo addgroup Hadoop
Sudo adduser –ingroup Hadoop hduser
Sudo adduser hduser sudo
Sudo apt-get install ssh
Which ssh
Which sshd
Sudo su hduser
cd
Ssh-keygen -t rsa -P “”
Cat $HOME/.ssh/id_rsa/pub >> $HOME/.ssh/authorized_keys
Ssh localhost
Sudo mkdir -p /usr/local/Hadoop
Wget https://archive.apache.org/dist/hadoop/common/hadoop-2.6.5/hadoop2.6.5.tar.gz
tar xvzf Hadoop-2.6.5.tar.gz
cd hadop-2.6.5
sudo mv */usr/local/hadop


sudo chown -R hduser:hadoop /usr/local/hadoop
vim ~/.bashrc
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
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" export
HADOOP_HOME_WARN_SUPPRESS=1

export HADOOP_ROOT_LOGGER="WARN,DRFA"

#HADOOP VARIABLES END
```

```
Source  ~/.bashrc
```

```
Vim /usr/local/Hadoop/hadoop-env.sh
```

```
Export JAVA_HOME = /usr/lib/jvm/java-8-openjdk-amd64
```

```
sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop /app/hadoop/tmp
```

```
vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<property>

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

        <value>/app/hadoop/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>
```

```
cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
<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>
```

```
sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
sudo chown -R hduser:hadoop /usr/local/hadoop_store
```

```
vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<property>
```

```
<name>dfs.replication</name>

<value>1</value>

<description>Default block replication.

The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time. </description>

 </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>
```

```
hadoop namenode –format
```

```
hdfs namenode -format
```

```
start-all.sh
```

```
jps
```

if not working

```
stop-all.sh
```

```
sudo rm -rf /app/hadoop/tmp

sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop /app/hadoop/tmp

sudo chown 750 /app/hadoop/tmp

start-all.sh
```

```
hdfs dfs -mkdir -p input_dir
```

```
cd
vim test.txt
```

```
hdfs dfs -put /home/hduser/test.txt input_dir
hdfs dfs -ls input_dir
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.5.jar wordcount input_dir output
```

```
hdfs dfs -ls output
hdfs dfs cat output/part-r-00000
```

DOCKER

```
sudo apt-get update

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"


sudo apt-get update

apt-cache policy docker-ce

sudo apt-get install -y docker-ce

sudo service docker start

sudo docker run hello-world
```

i) --------------------- running an existing image (Grafana)

```
docker run -d -p 3000:3000 grafana/grafana-enterprise

sudo docker images
```

ii) ---------------------creating a image

```
Mkdir DockerFiles

Cd DockerFiles

Touch Dockerfile

Vim Dockerfile
```

```
FROM ubuntu

MAINTAINER Nishanth VM Nishanth.codes@gmail.com

RUN apt-get update

CMD [ “echo”,”hiiiiii everyone out there”]
```

```
Docker build -t Nishanth:1.1 .

Docker images
```

<<<get id of our image

Docker run <id of image>

Output: CMD

Hosting website ---

In ssh :::

```
Sudo apt-get update
Sudo su
Sudo apt install apache2
```

```
cd..
cd var/www/html
rm index.html
touch index.html
vim index.html
<<<< write html,css, js code here >>>>
```

```
Sudo service apache2 start
```

In browser:::::::::::::::::::::::::

http://publicip/filename.html

Migration (AMI)

Create AMI using right click

Go to AMI

Click add permission

Add user accout ID (our ID)

SAVE

Go to create instance

Go to my amis / shared with me

Create

Reference

Scp command

```
Scp -i “keyname” filename  publicdns
```