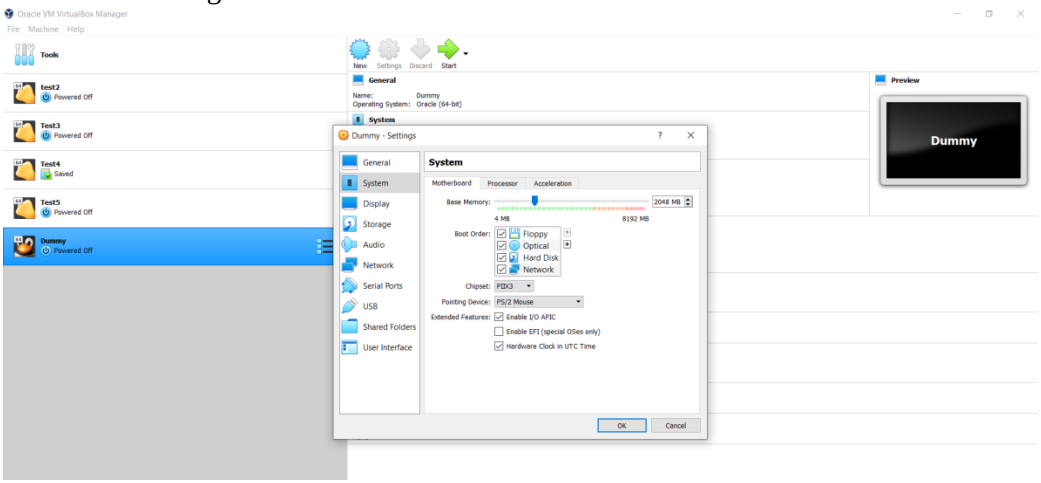
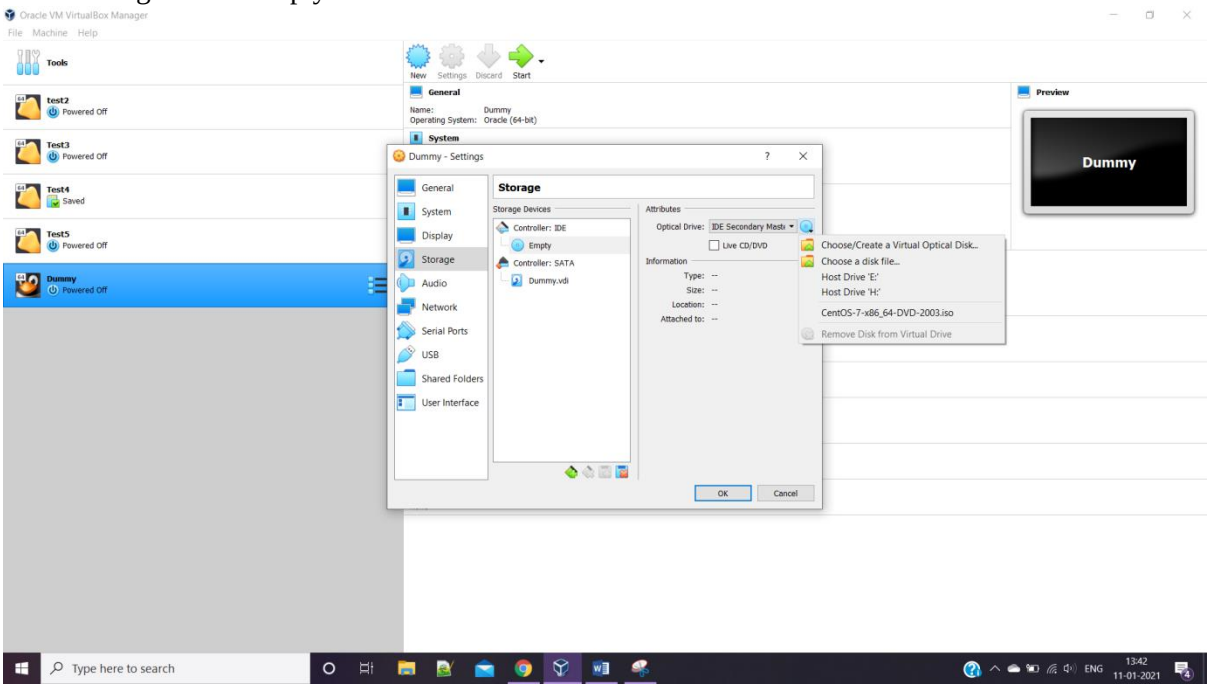
Setup CentOS machine using Oracle VM Virtualbox.

- Download Oracle VM Virtualbox for windows and install.
- Download CentOS 7 (4.5GB file approx)

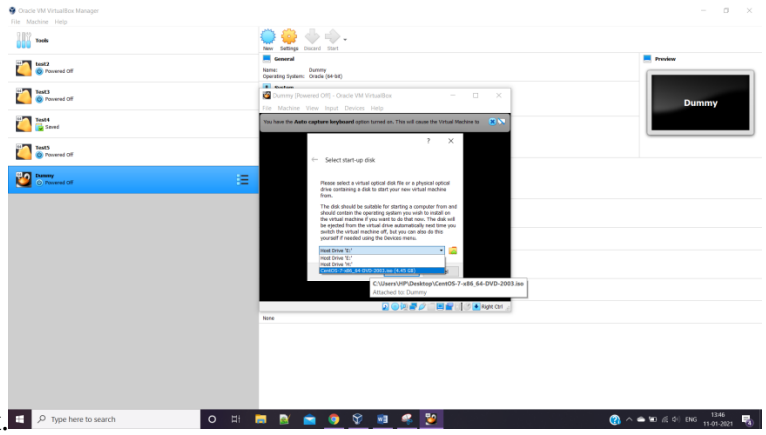
1. Launch Oracle VM Virtualbox.
2. Click New. Enter name for your machine , Type:Linux and Version: Other Linux 64 bit (if ‘Other Linux 64 bit’ is not in the options then enable your virtualisation by restarting your system)
3. Click Next
4. Give Memory Size (RAM in MB) recommended size: >=2048
5. Click Next
6. Select Create a Virtual hard disk now and click Create
7. Select VDI and click next
8. Select dynamically allocated and click next
9. Give Virtual hard disk size recommended size: >=32GB and click Create
10. A virtual Machine Created.
11. Now Click Settings



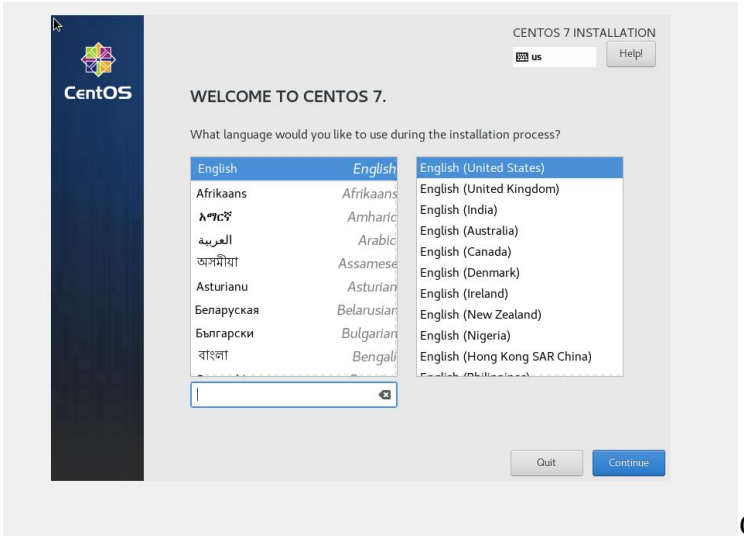
12. Click Storage. Click Empty. Select Centos



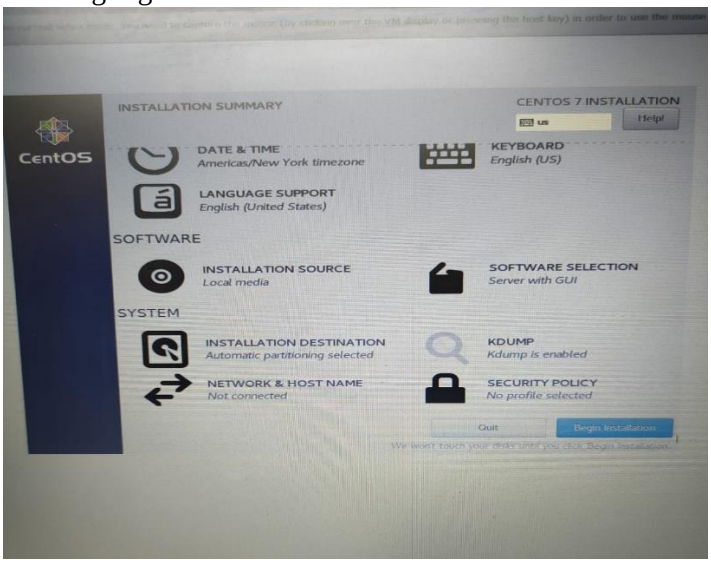
13. Click Network. Select Bridge Adapter in Place of NAT.
14. Click Ok
15. Now Start the Machine.



16. Select Centos and click start.
17. Press Enter and wait for few seconds.

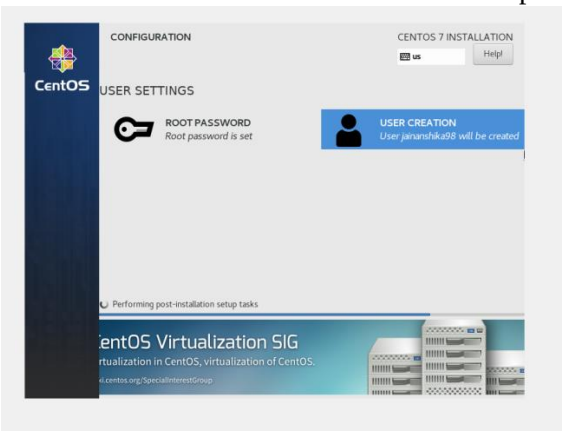


18. Continue
19. Click Software Selection and select Server with GUI. If any Other Option is showing Warning Sign then Click on it and then Click Done.



20. Click Begin Installation.

21. Set Root Password and create user and user password. Click Done. And Wait it will take time.



22. Then Click Reboot.
23. Click on battery icon and click wired off -> connect
24. Right click->open terminal
25. Type su – to login with root and enter root password

NOTE: Download PUTTY for convenience.

Once done, type **ipaddr** in your VM, and copy the ip in PUTTY and click open.

JAVA_PATH Setup

1. sudo yum install java-1.8.0-openjdk-devel

   Update jdk

2. Check java path and version installed on your machine.
   sudo update-alternatives --config java

3. Set JAVA_Home in etc/profile
   Cmd: vi /etc/profile (open etc/profile file)
   Press i to edit file
   Add this line in /etc/profile file
   export JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.275.b01-0.el7_9.x86_64"
   (java file name will be different)
   Press esc key and :wq! to save changes and exit.
4. Now open .bashrc file and add java path there
   Cmd: vi .bashrc
   Add these in .bashrc file
   export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.275.b01-0.el7_9.x86_64

   export PATH=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.275.b01-0.el7_9.x86_64/bin:$PATH
   Press esc key and :wq! to save changes and exit.

   Then

Cmd: source .bashrc (For applying all these changes to the current Terminal, execute the source command)

5. Check java version
   Cmd : java –version

Install Hadoop: Setting up a Single Node Hadoop Cluster.

Download the Hadoop 2.7.3 Package.

***Command*:** wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

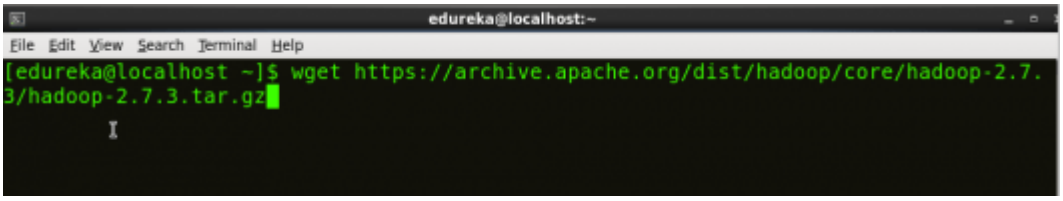


Fig: Hadoop Installation – Downloading Hadoop

Step 4: Extract the Hadoop tar File.

Command: tar -xvf hadoop-2.7.3.tar.gz

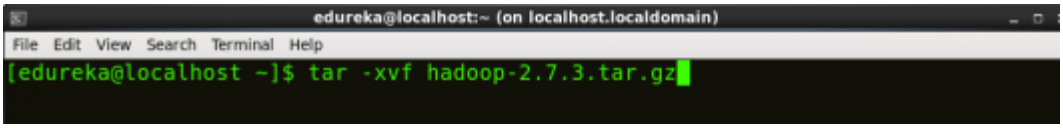


Fig: Hadoop Installation – Extracting Hadoop Files

In order to check your Hadoop path type ls command in the root.

If you can find hadoop-2.7.3 there, as shown below:

then your Hadoop path will be:

/root/hadoop-2.7.3

- ✓ In step 5, after writing the command vi.bashrc
  You need to copy these lines there:

  To copy first press esc key followed by i (as done earlier)

  export HADOOP_HOME=/root/hadoop-2.7.3
  export HADOOP_CONF_DIR=/root/hadoop-2.7.3/etc/hadoop
  export HADOOP_HDFS_HOME=/root/hadoop-2.7.3
  export HADOOP_MAPRED_HOME=/root/hadoop-2.7.3
  export HADOOP_YARN_HOME=/root/hadoop-2.7.3
  export HADOOP_COMMON_HOME=/root/hadoop-2.7.3
  export PATH=/root/hadoop-2.7.3/bin:$PATH

NOTE: copy the HADOOP_HOME & PATH only if you find the Hadoop in the root, as shown above.

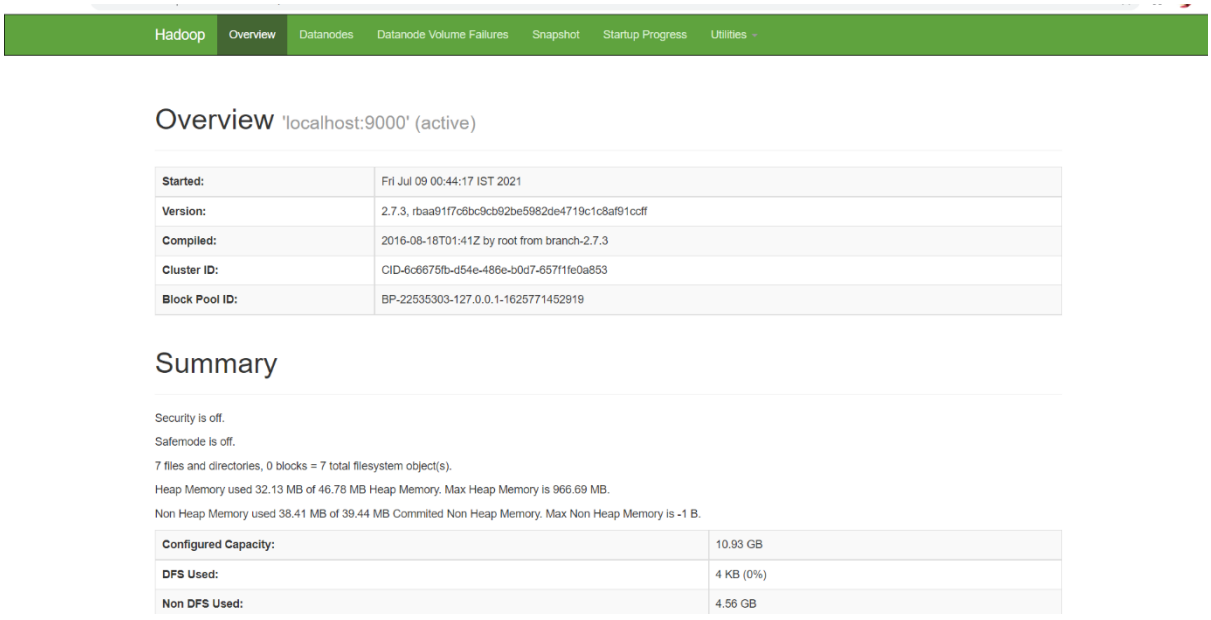
- ✓ For further steps you can follow the document.

- ✓ Once you are done till Step 14,

  Type **systemctl stop firewalld** in PUTTY

  Now go to your browser, and type the ip address followed by :50070
  Example,
  198.162.0.10:50070

  Below shown screen will appear:



Install Putty:

After installation of putty, enter the ip addrss of virtual machnine and then you can work on putty for your virtual machine using theses commands:

cd

cd hadoop-2.7.3

start-dfs.sh - Starts the Hadoop DFS daemons, the namenode and datanodes. Use this before

start-mapred.shstop-dfs.sh - Stops the Hadoop DFS daemons.

start-mapred.sh - Starts the Hadoop Map/Reduce daemons, the jobtracker and tasktrackers.

stop-mapred.sh - Stops the Hadoop Map/Reduce daemons.

start-all.sh - Starts all Hadoop daemons, the namenode, datanodes, the jobtracker and tasktrackers. Deprecated; use start-dfs.sh then start-mapred.sh

stop-all.sh - Stops all Hadoop daemons. Deprecated; use stop-mapred.sh then stop-dfs.sh

enter jsp

hive

select * from Dataset;(to print data of the table)

show tables;

drop table table_name;(to deleate the table)

To create a table in hive the command is:

```
CREATE EXTERNAL TABLE IF NOT EXISTS Dataset
(SNO INT,
MOVIENAME STRING,
YEAR INT,
GENRE STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/dataset';
```

Qeries to run:

```
Select GENRE from Dataset where Genre like’%Comedy’;
Select * from Dataset where YEAR between 1990 AND 1995;
Select * from Dataset where YEAR between 1995 AND 2000;
Select * from Dataset where YEAR between 2000 AND 2005;
Select * from Dataset where YEAR between 2005 AND 2010;
Select GENRE from Dataset where Genre like’%Romance’;
Select GENRE from Dataset where Genre like’%Action’;
Select GENRE from Dataset where Genre like’%Social’;
Select GENRE from Dataset where Genre like’%Thriller’;
Select GENRE from Dataset where Genre like’%Family’;
```

Select GENRE from Dataset where Genre like’%Horror’;

CREATE TABLE One(MOVIENAME STRING);

INSERT INTO TABLE One Select MOVIENAME from Dataset where YEAR between 1990 AND 1995;

select * from One;

CREATE TABLE One(MOVIENAME STRING);

INSERT INTO TABLE Two Select MOVIENAME from Dataset where YEAR between 1995 AND 2000;

select * from Two;

CREATE TABLE Three(MOVIENAME STRING);

INSERT INTO TABLE Three Select MOVIENAME from Dataset where YEAR between 2000 AND 2005;

select * from Three;

CREATE TABLE Four(MOVIENAME STRING);

INSERT INTO TABLE Four Select MOVIENAME from Dataset where YEAR between 2005 AND 2010;

select * from Five;

CREATE TABLE Countcomedy(MOVIENAME STRING);

INSERT INTO TABLE Countcomedy Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Comedy’;

select * from Countcomedy;

CREATE TABLE Countaction(MOVIENAME STRING);

INSERT INTO TABLE Countaction Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Action’;

select * from Countaction;

CREATE TABLE Countfamily(MOVIENAME STRING);

INSERT INTO TABLE Countfamily Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Family’;

select * from Countfamily;

CREATE TABLE Countsocial(MOVIENAME STRING);

INSERT INTO TABLE Countsocial Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Social’;

select * from Countsocial;

CREATE TABLE Countromance(MOVIENAME STRING);

INSERT INTO TABLE Countromance Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Romance’;

select * from CountRomance;

CREATE TABLE Counthorror(MOVIENAME STRING);

INSERT INTO TABLE Counthorror Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Horror’;

select * from Counthorror;

CREATE TABLE Countthriller(MOVIENAME STRING);

INSERT INTO TABLE Countthriller Select COUNT(MOVIENAME) from Dataset where GENRE like ‘%Thriller’;

select * from Countthriller;