

Hadoop 2.x Single Node Setup

Note:

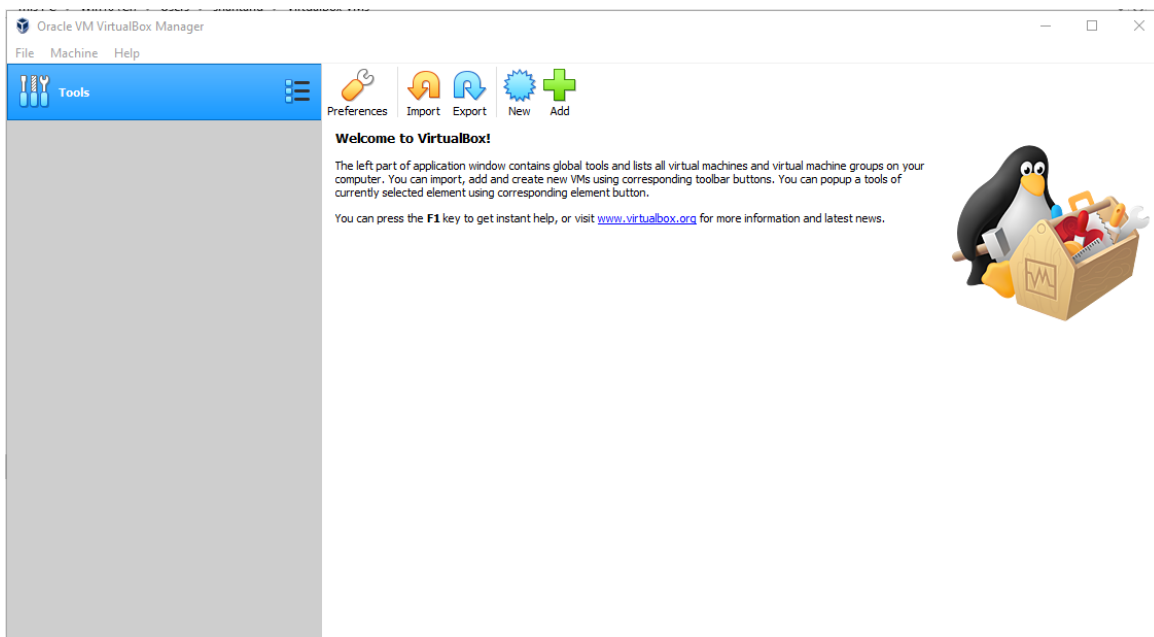
1. Make sure you have copied the bundle as `c:\hadoop-soft`

LAB 1: Install VirtualBox Software and Import the VM

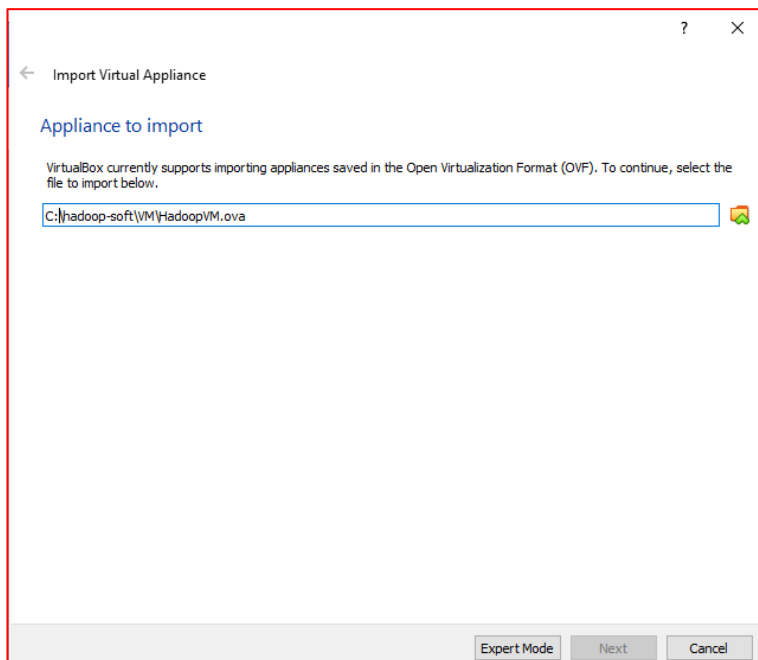
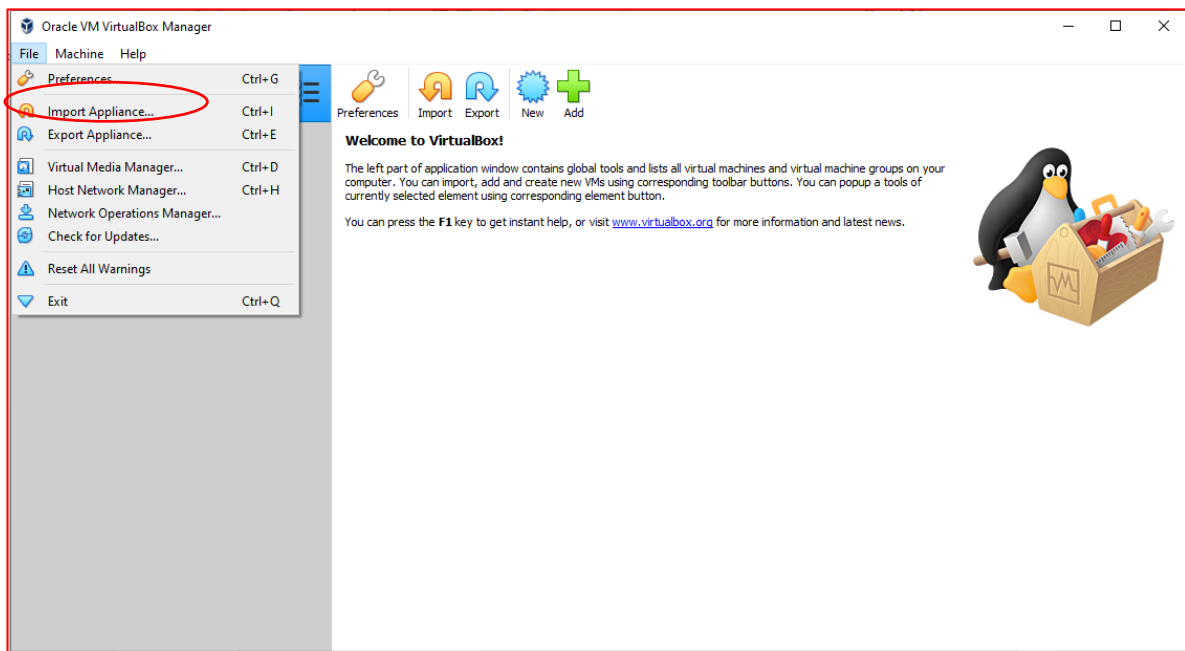
- To install VirtualBox it is necessary to enable virtualisation in your computer (Verify and Enable it)
- Double click “VirtualBox-6.xxx.exe” from `c:\hadoop-soft\VM` folder and follow the On Screen instructions
- Once the installation is complete then start the VirtualBox Manager. See the following Screen



Double click on  It starts VirtualBox Manager as below

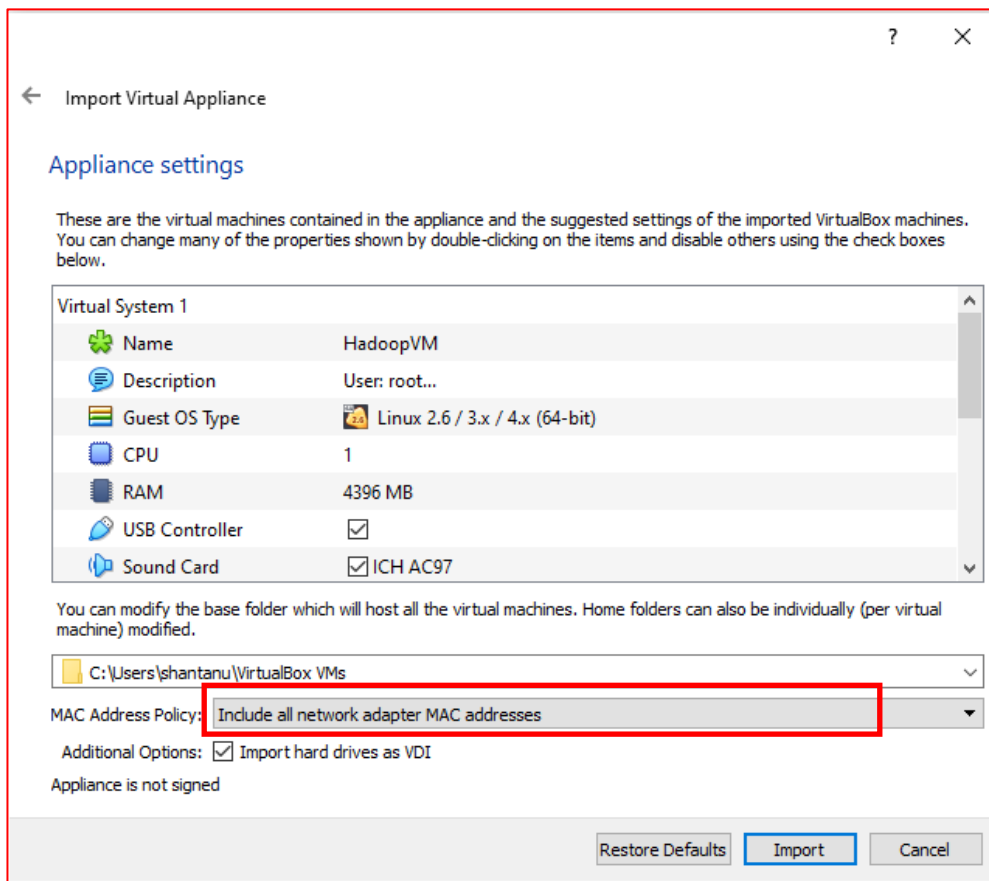


Click on **File>Import Appliance** and navigate to ‘`hadoop-soft/VM`’ folder and select the ‘.ova’ file and click on **Next** button

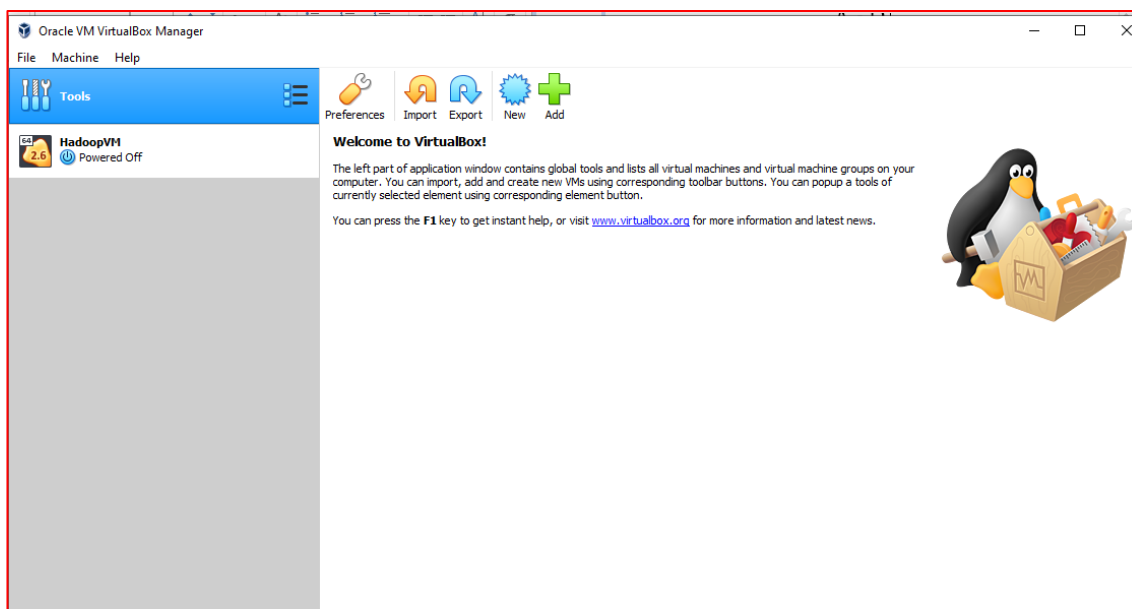


In the Next Screen Change the name of the Linux VM to “HadoopVM” and then click on

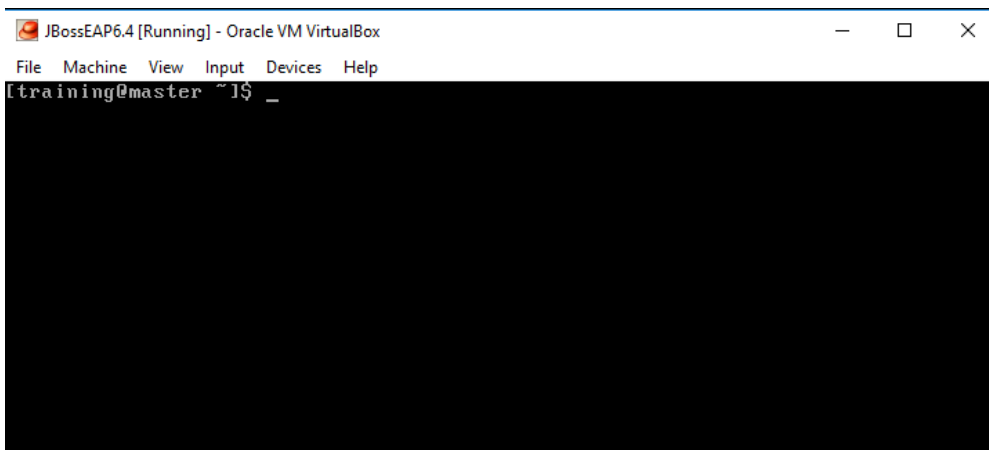
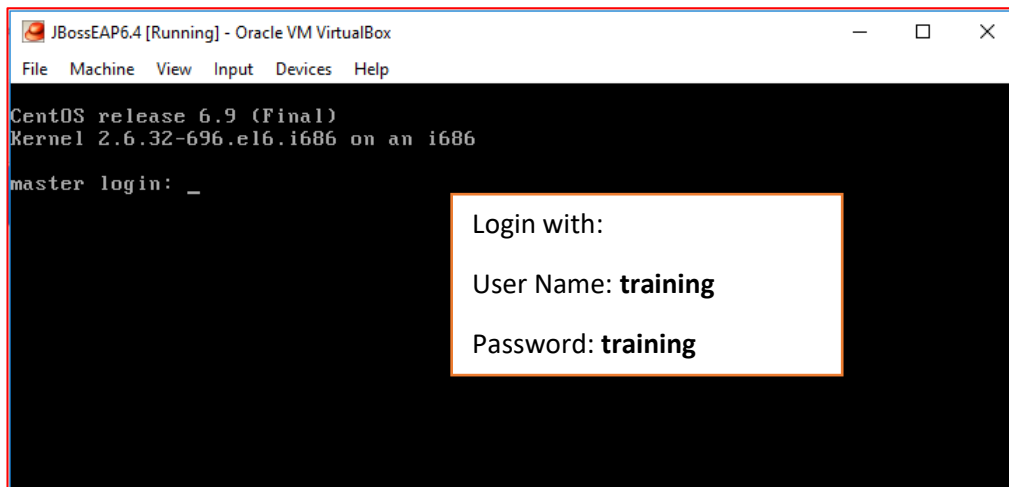
Import



Once it is imported your screen should look like the following:



Start the Virtual machine by clicking the start button. After the VM Starts, it looks like the following screen.



\$ifconfig

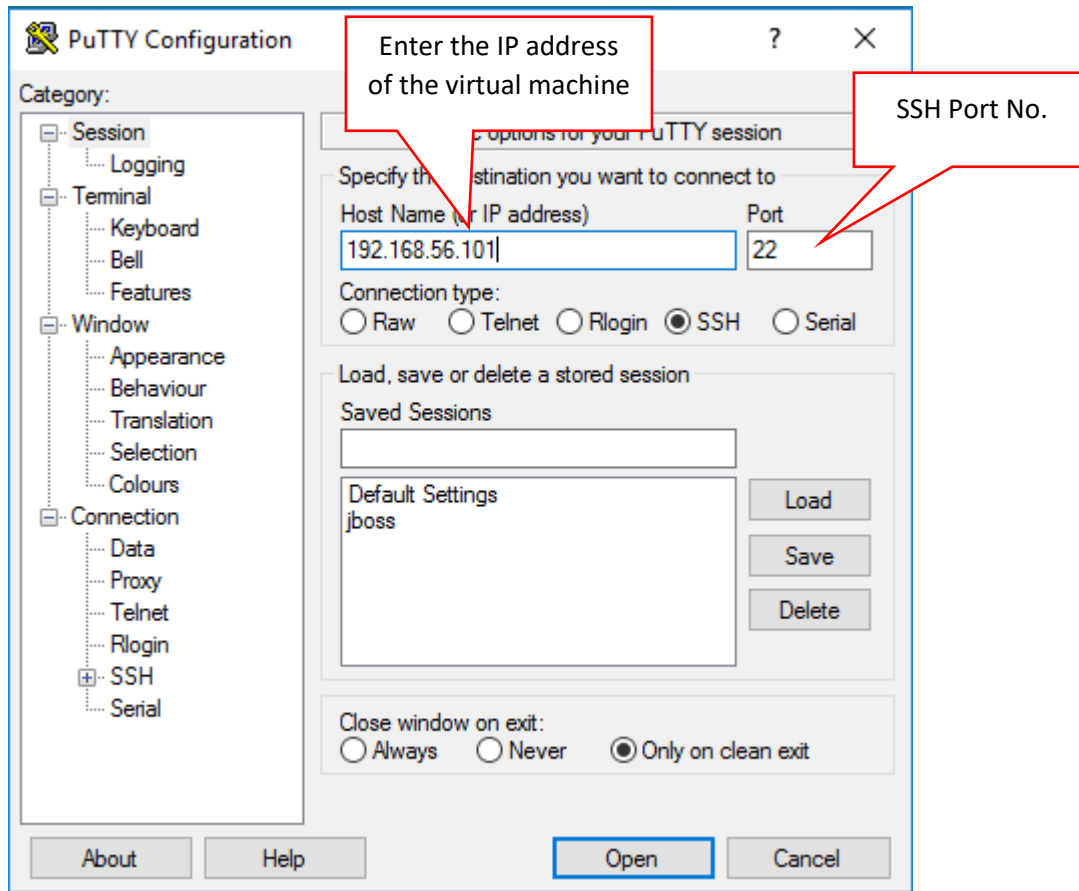
```
[training@master ~]$  
[training@master ~]$ ifconfig  
eth1      Link encap:Ethernet  HWaddr 08:00:27:4A:77:41  
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe4a:7741/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:3268 (3.1 KiB)  TX bytes:1272 (1.2 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

IP Address for the VM Now is 192.168.56.101

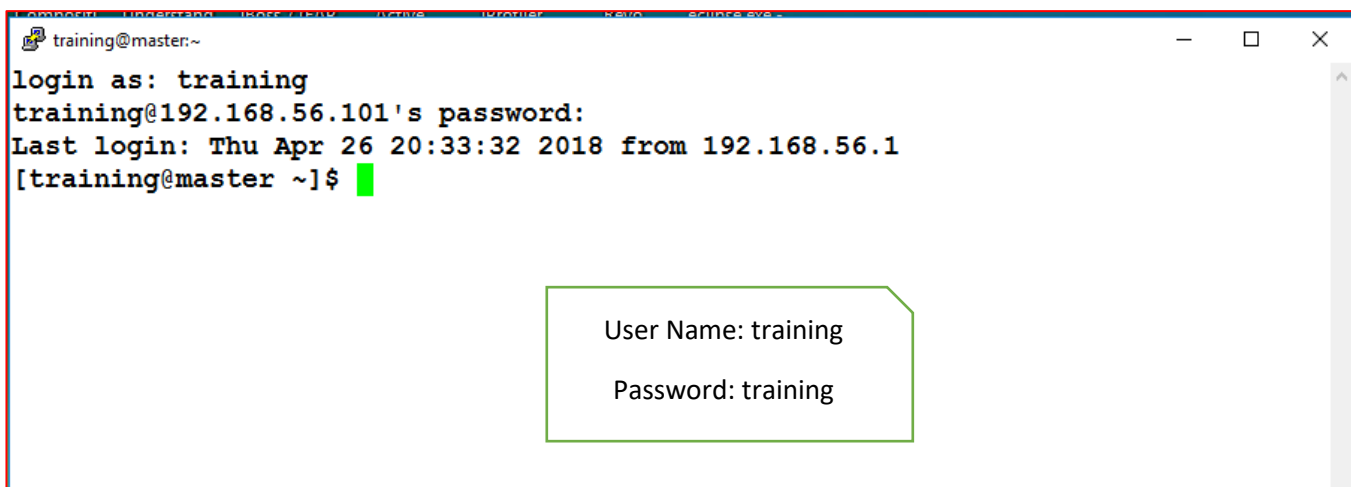
LAB2: Connect the VM using Putty and WinSCP

1. Putty Connection:

Double click on putty.exe. Putty opens and the following screen appears:



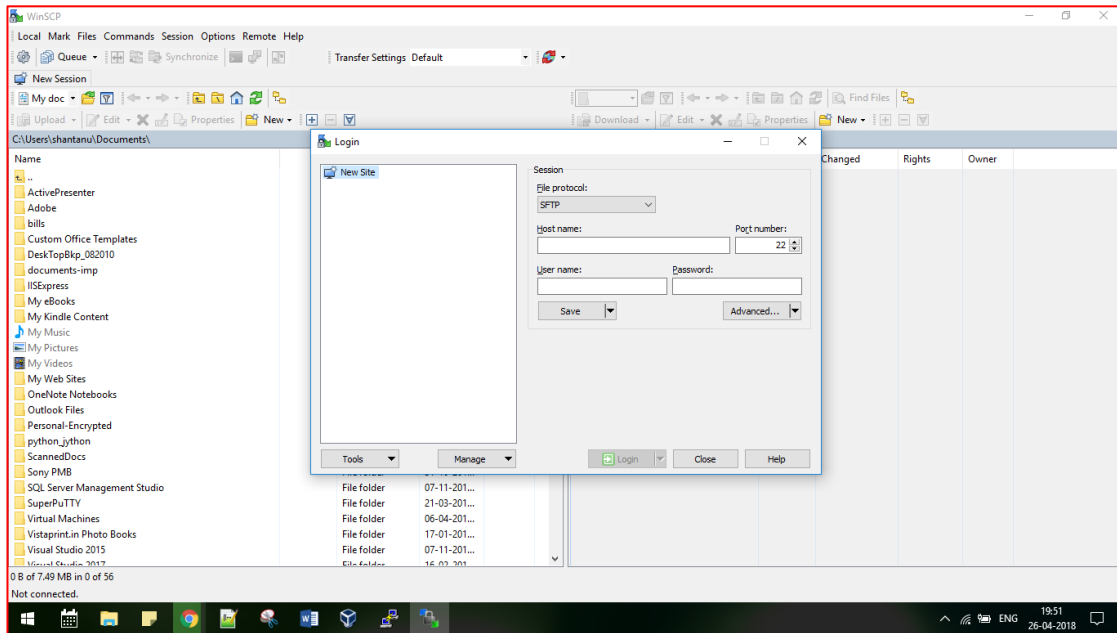
Provide the IP address and SSH port and press “Open”



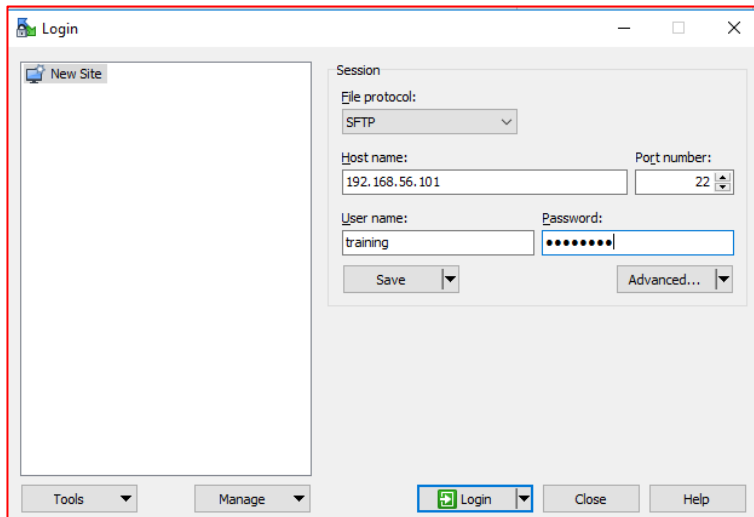
YOU ARE NOW CONNECTED TO THE VM USING PUTTY

2. WinSCP Connection: (Install WinSCP, if required and then do the following)

Double Click on WinSCP and it opens as follows:

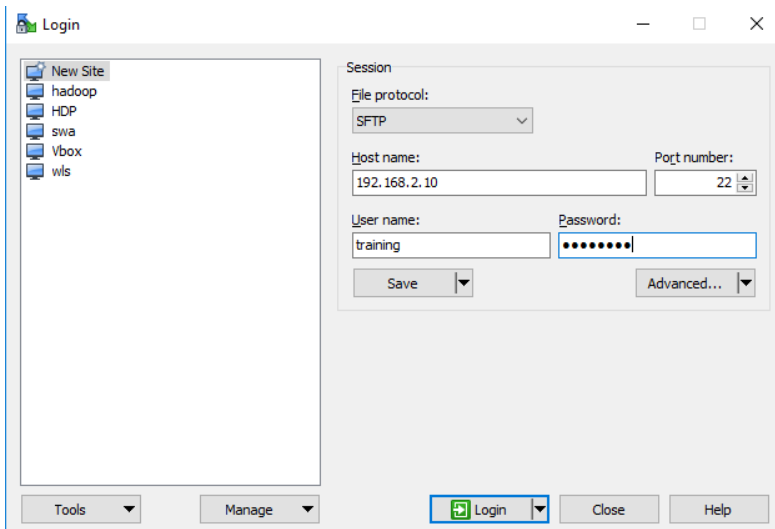


Provide IP Address, username and password and click on save. Save the configuration as JBossEAp64.



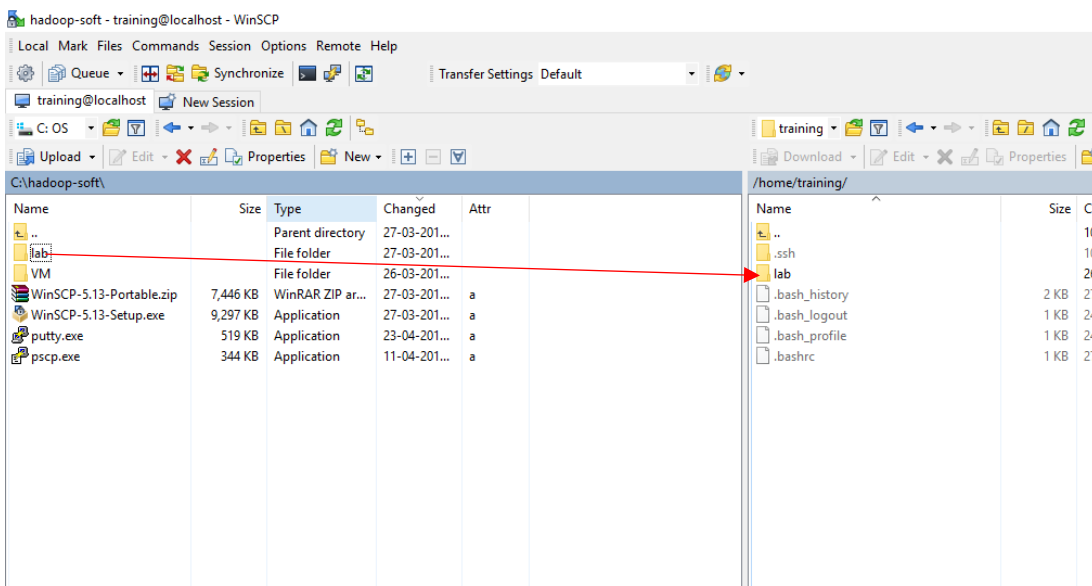
1. Use **winscp** from host machine to copy training bundle to Linux VM e.g.

(Note: Skip to “Install Java” Step if you have already transferred the training bundle to your linux VM.)



User Name: training

Password: training



Now your softwares are available in Linux OS path **'/home/training/lab'**

LAB3: Install Java

1. Open a terminal on Linux OS or Use Putty Terminal

2. Now downloaded jdk is available in \$HOME/lab/install in your guest OS. Use the **Linux terminal** to execute the following commands. Please edit the Java installer file name if required in the following commands.

```
$ sudo mkdir /usr/java
```

```
$ sudo tar zxvf /home/training/lab/install/jdk-7u55-linux-x64.tar.gz -C /usr/java/
```

```
$ sudo ln -s /usr/java/jdk1.7.0_55 /usr/java/latest
```

3. set JAVA_HOME and PATH as given below

```
$ sudo vi /etc/profile.d/javaenv.sh
```

Make the following entry and save the file

```
JAVA_HOME=/usr/java/latest
```

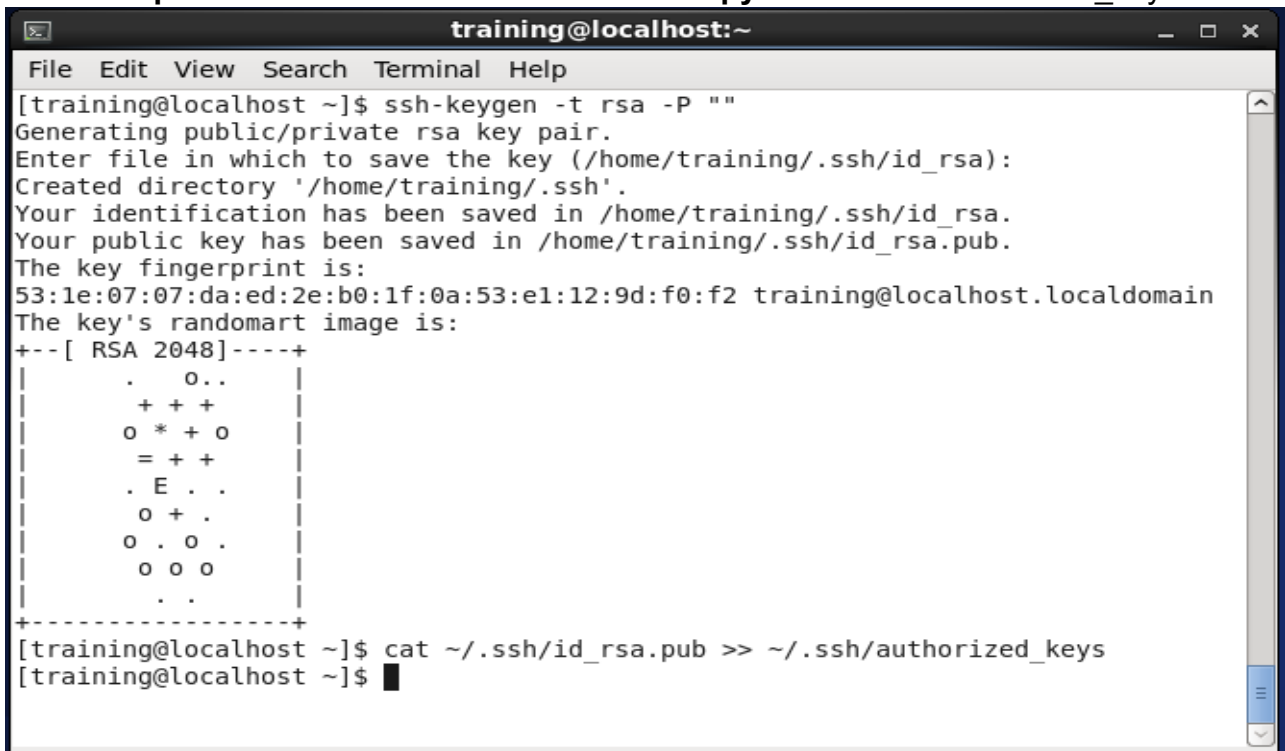
```
PATH=$JAVA_HOME/bin:$PATH
```

```
export JAVA_HOME PATH
```

4. Execute the following command to load javaenv.sh file

```
$source /etc/profile.d/javaenv.sh
```

2. Generate password less SSH certificate and copy it to “~/.ssh/authorized_keys”



```
training@localhost:~  
File Edit View Search Terminal Help  
[training@localhost ~]$ ssh-keygen -t rsa -P ""  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/training/.ssh/id_rsa):  
Created directory '/home/training/.ssh'.  
Your identification has been saved in /home/training/.ssh/id_rsa.  
Your public key has been saved in /home/training/.ssh/id_rsa.pub.  
The key fingerprint is:  
53:1e:07:07:da:ed:2e:b0:1f:0a:53:e1:12:9d:f0:f2 training@localhost.localdomain  
The key's randomart image is:  
+--[ RSA 2048 ]-----+  
|          .  o..      |  
|        + + +      |  
|       o * + o      |  
|      = + +       |  
|     . E . .       |  
|      o + .        |  
|     o . o .       |  
|      o o o        |  
|          . .       |  
+-----+  
[training@localhost ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
[training@localhost ~]$
```

Depending on your version of SSH you might also have to do the following changes:

```
$ chmod 0640 ~/.ssh/id_rsa.pub
```

```
$ chmod 0640 ~/.ssh/authorized_keys
```


LAB4: Install hadoop as follows:

1. **hadoop-2.7.5.tar.gz** is available in Linux VM's /home/training/lab/install folder.

execute the following commands from Linux terminal.

```
$sudo mkdir /u01
```

```
$sudo chown training:training /u01
```

```
$ tar zxvf /home/training/lab/install/hadoop-2.7.5.tar.gz -C /u01/
```

```
$ ln -s /u01/hadoop-2.7.5 /u01/hadoop
```

2. Modify the **\$HOME/.bashrc** file of training to include the following lines

```
# User specific aliases and functions
```

```
export JAVA_LIBRARY_PATH=/u01/hadoop/lib/native
```

```
export HADOOP_PREFIX=/u01/hadoop
```

```
export PATH=$PATH:$HADOOP_PREFIX/bin:$HADOOP_PREFIX/sbin
```

```
export HADOOP_MAPRED_HOME=$HADOOP_PREFIX
```

```
export HADOOP_COMMON_HOME=$HADOOP_PREFIX
```

```
export HADOOP_HDFS_HOME=$HADOOP_PREFIX
```

```
export YARN_HOME=$HADOOP_PREFIX
```

3. **Update JAVA_HOME in \$HADOOP_PREFIX/etc/hadoop/hadoop-env.sh file**

```
export JAVA_HOME=/usr/java/latest
```

4. Create Hadoop data directories

```
$ mkdir -p /u01/hadoop-work/data
```

```
$ mkdir -p /u01/hadoop-work/name
```

```
$ mkdir -p /u01/hadoop-work/tmp
```

5. Configure **\$HADOOP_PREFIX/etc/hadoop/core-site.xml** as follows:

```
<configuration>
```

```
<property>
```

```
<name>fs.defaultFS</name>
```

```
<value>hdfs://localhost:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/u01/hadoop-work/tmp</value>
</property>
</configuration>
```

6. Configure \$HADOOP_PREFIX/etc/hadoop/hdfs-site.xml as follows:

```
<configuration>
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
<property>
    <name>dfs.name.dir</name>
    <value>file:///u01/hadoop-work/name</value>
</property>
<property>
    <name>dfs.data.dir</name>
    <value> file:///u01/hadoop-work/data</value>
</property>
</configuration>
```

7. Create and update \$HADOOP_INSTALL/etc/hadoop/mapred-site.xml as follows:

```
$ mv mapred-site.xml.template mapred-site.xml
$ vi mapred-site.xml
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
```

```
</property>
```

```
</configuration>
```

8. Configure \$HADOOP_PREFIX/etc/hadoop/yarn-site.xml as follows:

```
<configuration>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.hostname</name>
```

```
    <value>localhost</value>
```

```
  </property>
```

```
  <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>
```

```
</configuration>
```

9. Format Namenode:

```
$ hdfs namenode -format
```

10. Start Hadoop Services

```
$start-dfs.sh
```

```
$start-yarn.sh
```

11. Verify the running Daemons

\$jps

You should get the following if hadoop system starts successfully [pids would be different]:

1898 ResourceManager

2248 Jps

1397 NameNode

1517 DataNode

1995 NodeManager

1759 SecondaryNameNode

12. Run hadoop pi estimation sample program to test the setup:

\$hadoop jar \$HADOOP_PREFIX/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.5.jar pi 2 5

Note: in some SSH versions do the following

chmod 700 \$HOME/.ssh

chmod 640 \$HOME/.ssh/authorized_keys

Note: SSH Permissions

drwx-----	2	training	training	4096	Jan 5 09:10	.ssh	(700)
-rw-r-----	1	training	training	412	Jan 5 09:10	authorized_keys	(640)
-rw-----	1	training	training	1675	Jan 5 09:09	id_rsa	(600)
-rw-r-----	1	training	training	412	Jan 5 09:09	id_rsa.pub	(640)
-rw-r--r--	1	training	training	1184	Jan 5 09:43	known_hosts	(644)