

1. WEB SERVICE -

Open Netbeans, new Web Application -> JAVA EE 6

Create a new WebService by right-click on project folder

In the java file, add necessary operations thru : add operation

Build -- Test -- Test WebService -- Deploy

Service is opened in browser. Copy the WSDL URL for when client is created.

As a new project , create new Web Application -> JAVA EE 6

Create a new WebServiceClient and paste the wsdl url when required.

Go to WebServiceReferences , drag and drop the modules into the jsp file.

Basic syntax to create a form to be displayed on browser : (in index.jsp)

```
<form action = "pagename.jsp" method = "GET" target = "_self">
<input type = "text" name = "someName">
...as per need...
<input type = "submit" value = "someValue" name = "someName">
</form>
```

Basic example using "add method" for mainjsp file (Alter jsp file with necessary methods)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ARITHMETIC PAGE</title>
```

```
</head>
<body>
<!-- start web service invocation --%><hr/>
<%
try {
com.cal.example.Calculate_Service service = new com.cal.example.Calculate_Service();
com.cal.example.Calculate port = service.getCalculatePort();
String no1=request.getParameter("num1");
String no2=request.getParameter("num2");
String subty=request.getParameter("submit");
if(subty.equals("ADD"))
{
int num1=Integer.parseInt(no1);
int num2=Integer.parseInt(no2);
int result= port.add(num1, num2);
out.println("ADD Result = "+result);
}
} catch (Exception ex) {
// TODO handle custom exceptions here
}
%>
<!-- end web service invocation --%><hr/>
</body>
</html>
```

Then right click on Client and RUN.

2 . FOR ANY VM EXP., DO THE FOLLOWING TO ENSURE PARTIAL O/P :

CREATE VM : (For desktop iso)

Open VirtualBox and select a vm, go to :

Settings -> General -> Advanced Change Shared Clipboard : Bidirectional ,
Drag'n'Drop : Bidirectional .

Settings -> Storage -> Controller:IDE ... Choose desktop iso (ending with .04)[16.04 preferably]

Settings -> Network -> Adapter1 (Select BRIDGED ADAPTER) .. click on Advanced -> Promiscuous mode .. Select Allow All.

Save the settings and START.

After installing UBUNTU, go to Network Settings and set proxy.

Open System Settings -> Network -> Options -> IPv4 Settings :

Method : Manual

Change the Address, SubnetMask, Gateway, DNS server address as per given values and save changes.

Open Terminal , type :

sudo gedit /etc/resolv.conf

if nameserver is not set to dns server value .. type the address beside the 'nameserver', else just close it.

Open Mozilla Firefox and check if Internet is working **(CHECKPOINT 1)**

TO INSTALL JAVA :

In Terminal , type :

```
sudo apt-get install default-jdk
```

After return of control, type : **java -version** to check if installation was errorfree.

Then, create a java file , say : **xyz.java**

Go to Terminal :

```
javac xyz.java
```

```
java xyz
```

If it works (**CHECKPOINT 2**)

TO INSTALL VM SERVER :

Choose server iso and continue with installation . (Change settings as in desktop iso)

Choose **configure the network** in Main Menu

Configure network manually

Enter IP Address, subnet mask and gateway as given.

Partitioning method : **Guided - use entire disk**

Setup with hostname, username , password, proxy and **no automatic updates.**

Install **OpenSSHServer ...** to select software or deselect software , press 'space' and then press 'Enter' to move to the next step in menu. **[DO NOT press enter before selecting necessary softwares]**

Install GRUB Boot Loader

After server loads, PING ANOTHER VM by using : **ping <ip of another vm>**

eg : ping 10.6.4.155

This ensures connectivity.

3. REMOTE LOGIN :

WITHOUT PASSWORD :

After executing the above steps in server or desktop -

(For desktop , enter the following command to install sshserver and update changes :

```
sudo apt-get install openssh-server
```

```
sudo apt-get update
```

```
)
```

Enter the following commands in order :

```
ssh-keygen -t rsa // press enter when asked for keyphrase and file location
```

```
{ Now, change current directory to that of ssh :
```

```
For example , if it shows /home/vm1/.ssh/id_rsa while generating key,
```

```
do .... cd vm1/.ssh or cd /.ssh or cd /root/.ssh or
```

```
cd ~/.ssh }
```

```
ls //list the contents in directory ( authorized_keys id_rsa  
id_rsa.pub)
```

```
//if anything is not listed, theres some error and google on how to fix
```

```
        chmod 700 id_rsa.pub
cp id_rsa.pub authorized_keys
ssh-copy-id <other vm>@<other ip>    { for eg :    ssh-copy-id vm2@10.6.4.155 }
```

Now vm1 s terminal is logged into vm2

Enter **exit** to return to original vm .

WITH PASSWORD :

(For desktop , enter the following command to install sshserver and update changes :

```
sudo apt-get install openssh-server
```

```
sudo apt-get update
```

)

Enter the following command :

```
ssh <othervm>@<otherip>    { for eg :    ssh vm2@10.6.4.155 }
```

Now vm1 s terminal is logged into vm2

Enter **exit** to return to original vm .

4 . FILE TRANSFER BETWEEN VM :

Execute the steps in order till installation of ssh server.

TO SEND FILE :

Type in the following command in the terminal :

```
scp /home/vm1/<filename> vm2@ipaddress2:<destination_path_of_file>
```

Eg : `scp /home/vm1/f1.txt vm2@10.6.4.155:/home/vm2/`

TO RECEIVE FILE :

Type in the following command in the terminal :

{ If nano editor is not installed, then install it using this command and update :

```
sudo apt-get install nano
```

```
sudo apt-get update
```

}

```
scp vm2@ipaddress2:<source_path_of_file> <newFileName>
```

Eg : `scp vm2@10.6.4.155:/home/vm2/f1.txt f2.txt`

Check for transfer by typing `ls` in terminal or go to the folder and view the file.

5 . FOLDER TRANSFER BETWEEN VM :

First, create a directory using command :

```
mkdir <dirName>
```

Eg : `mkdir myDir`

Inside the directory, create one or more files.

Then , type the following command in the terminal :

`scp -r <source_address> <destPath>`

Eg : `scp -r /home/vm1/myDir vm2@10.6.4.155:/home/vm2/`

Check for transfer by typing `ls` in terminal of vm2 or go and check in the file system .

6. EUCALYPTUS :

Create 2 VMs Configure network as DHCP .. dont use Ethernet.

Install server iso for both

Select Ubuntu Enterprise cloud

Set hostname , leave cloud controller address blank

Select the following : (Press 'space' then press 'enter')

Cloud controller,Walrus storage service,Cluster controller,Storage controller { FOR VM 1 }

Node controller { FOR VM 2 }

Partition disks : select **"Guided-use entries disk and set up LVM"**

Enter size as **10.5 GB**

Enter username , password , clustername

Leave pool of IP addresses blank and then , Install GRUB Loader

Create 3rd VM with desktop iso to act as client (**VM3**)

Install qemu-kvm in VM1

Set a temporary password in VM2 using : `sudo passwd eucalyptus`

Type the command in VM1 :

sudo -u eucalyptus ssh-copy-id -i /var/lib/eucalyptus/.ssh/id_rsa.pub eucalyptus@<ip_vm2>

Remove temporary password in VM2 using : **sudo passwd -d eucalyptus**

In VM3, go to Mozilla Firefox browser and type the following URL : **DONT USE PROXY**

Fehler! Hyperlink-Referenz ungültig. (if it doesnt load , type : Fehler!
Hyperlink-Referenz ungültig. **/#login**)

Username : **admin** Password : **admin**

Give new username, password, email id.

Go to Credentials -> Download Credentials {Download to Downloads}

Then **cd Downloads**

Transfer the file to VM1 using :

scp euca2-admin-x509.zip vm1@ip_vm1:/home/vm1

In VM1 :

Type :

mkdir -p ~/.euca

cd ~/.euca

chmod 0700 ~/.euca

chmod 0600 ~/.euca/*

sudo euca_conf --get-credentials mycreds.zip

unzip mycreds.zip

Check the contents using **ls** command.

In VM3 :

Type :

sudo apt-get update

sudo apt-get install euca2ools

Go to file named **eucarc** under the **X.509** folder downloaded as **certificate credentials** and identify the : **URL , ACCESS KEY , SECRET KEY** .

Now , type the following in the terminal : (its capital 'i' before <accesskey>)

euca-create-volume -U <url> -I <accesskey> -S <secret_key> --size 1 -z <clustername>

euca-describe-volumes -U <url> -I <accesskey> -S <secret_key>

The terminal output indicates private cloud setup and volume creation in client machine.

7 . OPEN NEBULA

Installation - Install 2 desktop iso's or use a single vm and 2 terminals (use as root only for front end node)

Frontend Installation : (**VM1**)

sudo -i

apt-get update

Install packages and dependencies:

apt-get install opennebula opennebula-sunstone nfs-kernel-server

To check if packages were installed: **ls -l /dev/kvm**

Open the file: **gedit /etc/one/sunstone-server.conf**

a. Change the line 'host: 127.0.0.1' to '**host: 0.0.0.0**' (leave untouched if latter is already present)

Restart sunstone server: **/etc/init.d/opennebula-sunstone restart**

Generate keys: **ssh-keygen -t rsa** (click Enter for all subsequent queries)

Copy keys in the following manner:

- a. **cd /root/.ssh**
- b. **chmod 600 id_rsa.pub**
- c. **cp id_rsa.pub authorized_keys**

Create a file in the same directory and put the following contents in it:

- a. **gedit config**

b. Content:

Host* StrictHostKey Checking no

UserKnownHostsFile ./dev/null

Node Installation: (VM2)

Perform: **apt-get update**

Install packages and dependencies:

- a. **apt-get install opennebula-node nfs-common bridge-utils**

Configure server interface (at node):

- a. **cd /etc/network/interfaces.d**

Create a file and put the following contents in it

- a. **gedit eth0.config** (delete any other file starting with 'eth0.config')

b. Content:

auto lo iface lo

inet loopback

auto br0

iface br0 inet static

address <ip>

network 192.168.1.4

netmask <YOUR NETMASK>

broadcast <YOUR BROADCAST>

gateway <YOUR GATEWAY>

bridge_ports eth0

bridge_fd 9

bridge_hello 2

bridge_maxage 12

bridge_stp off

Restart networking: **/etc/init.d/networking restart**

Open Nebula Sunstone Log-In:

You can go to Open Nebula Sunstone's Home Page on your browser using: **'http://localhost:9869'**

Username: **oneadmin** Password: (Unique for each installation)

Found in terminal using:

1. **su – oneadmin**

2. **cat .one/one_auth**

Homepage should get displayed.

8 . CREATION OF VM TEMPLATE – OPEN NEBULA

1. Log-in to Open Nebula Sunstone Home Page.
2. On the left pane, click on Templates Tab and click on VMs

3. To add a VM Template, click on the green + button.
 4. Enter the name, description and other attributes
 5. After selecting all the options above, click the green Create button. You should see the main VM Templates page again with your template updated.
-

9. LIVE MIGRATION OF VM

Install OpenNebula Front-End VM and KVM node VM(Refer EX: 7) [2 VMs with Ubuntu 16.04 desktop image]

After installing OpenNebula front-end and kvm node , the following steps need to be performed in the front-end VM:

1. Initially , list all the hosts,templates and vms using:

```
$ onehost list
```

```
$ onetemplate list
```

```
$ onevm list
```

2. CREATION OF HOSTS:

2.1. The hosts can be created via the command line

```
$ onehost create frontend -i kvm -v kvm -n dummy
```

(or) Through the open nebula web interface : localhost:9869 . Login with the username and password as given in `/var/lib/one/.one/one_auth` file using the command

```
$ sudo gedit /var/lib/one/.one/one_auth
```

2.2 Navigate to Infrastructure -> hosts in the left menu pane in the web interface.

2.3. Click on the '+' add option and specify the hostname and click on "create".

2.4. Similarly create another host and both will be listed.

On clicking the enable button after creating the hosts, the status changes to init.

\$onehost list can be used to list the hosts in the terminal.

3. CREATION OF TEMPLATE:

3.1. Under the virtual resources > template option is selected and in the similar way to hosts , templates are created.

\$onetemplate list will specify the new template created.

4. CREATION OF VM :

4.1. The template created is instantiated to create a virtual machine by clicking on the "instantiate" option in templates.

5. Deploy and migrate the VM- vm1 on host by specifying the vm id and host id.

\$ onevm deploy <vm-id> <host-id>

\$ onevm migrate <vm-id><host-id>

Here, vm1 with id 0 is deployed on host 1 with id 0. host1 appears under the host column.

10 . HADOOP INSTALLATION

Hadoop 3.1 Installation Steps

Install Java 8 and verify that it is working.

```
>> java -version
```

Step 1: Add User hduser with sudo privilege

```
>> adduser hduser
```

```
>> usermod -aG sudo hduser
```

Step 2: Install SSH Server and add the private key to the known lists

```
>> ssh-keygen -t rsa " -f ~/.ssh/id_rsa
```

Which generate public keys using rsa method.

```
>> cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```
>> chmod 0600 ~/.ssh/authorized_keys
```

Check whether you will be able to access to your localhost through ssh 'without a password'.

```
>> sudo apt-get install openssh-server
```

```
>> ssh localhost
```

Step 3: Installing Hadoop 3.1

3.1 Extract and move the hadoop to an installation directory)

```
>> wget www-us.apache.org/dist/hadoop/common/hadoop-3.1.1/hadoop-3.1.1.tar.gz
```

or download manually from

www-us.apache.org/dist/hadoop/common/hadoop-3.1.1/hadoop-3.1.1.tar.gz

```
>> tar xzf hadoop-3.1.1.tar.gz
```

extract zip file

```
>>mv hadoop-3.1.1 /usr/local/hadoop
```

Move extracted file into local disk (Installed)

3.2 Switch local user to hduser

```
>>su hduser
```

3.2 Set the hadoop environment variables

```
>>nano ~/.bashrc
```

Add the following lines to the bashrc file.

```
#HADOOP Variables
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export HADOOP_INSTALL=$HADOOP_HOME
```

```
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/lib/native
```

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

```
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

Save the bashrc file and exit by CTRL+O followed by CTRL+X

Refresh the bashrc file so that our environment variables can be accessed.

```
>>. ~/.bashrc
```

Check the hadoop version

```
>>hadoop version
```

3.3 Change the Hadoop and Related Config

3.3.1 Change the **JAVA_HOME** variable in

\$HADOOP_HOME/etc/hadoop/hadoop-env.sh file.

```
>>sudo nano $HADOOP_HOME/hadoop/hadoop-env.sh
```

set the **JAVA_HOME=**<<Java Installed Directory>>

3.2.2 Change the Hadoop Core config in

```
$sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

tag.

Add the following property tag inside the configuration

```
<property>
```

```
<name>fs.default.name</name>
```

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

```
</property>
```

3.2.3 Add the following properties to hdfs-site.xml

configuration

```
$sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<property>
```

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

```
<value>1</value>
```

```
</property>
```

```
<property>
```

```
<name>dfs.name.dir</name>
```

```
<value>file:///home/hduser/hadoop-store/hdfs/namenode</value>
```

```
</property>
```

```
<property>
```

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

```
<value>file:///home/hduser/hadoop-store/hdfs/datanode</value>
```

```
</property>
```

3.2.4 Add the following properties to the mapred-site.xml

```
>>sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

```
<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_HOME}/share
```

```
e/hadoop/mapreduce/lib/*</value>
```

```
</property>
```

3.2.5 Add the following properties to the yarn-site.xml

```
>>sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

```
<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_HOM
```

```
E</value>
```

```
</property>
```

4. Format and Prepare the HDFS

4.1 Make Sure the data directory mentioned in the hdfs-site.xml is exist and the current user has permission to it.

```
>>sudo chown -R hduser:hduser /usr/local/hadoop
```

```
>>sudo chmod 777 /usr/local/hadoop/logs
```

(make sure current user has permission to it)

>>**hdfs namenode -format**

5. Start the hadoop dfs and yarn process

>>**start-dfs.sh**

>>**start-yarn.sh**

6. Check the hadoop daemons by running the following command

>>**jps// java process**

7. Access the Hadoop DFS and Hadoop YARN Web UI,

-- for NameNode UI

http://localhost:9870/

For -- YARN UI

http://localhost:8042/

WORD COUNT

Step 1: Start Hadoop **start-dfs.sh**

Start yarn **start-yarn.sh**

Step 2 : Access the Hadoop DFS and Hadoop Yarn Web UI **http://localhost:9870/** -- NameNode UI

http://localhost:8042/ -- YARN UI

Step 3: Create a new folder in hadoop **hadoop fs -mkdir /wordcountfolder**

Step 4: Create a sample input text file for the word count.

gedit sampleinput.txt

Step 5: Copy the sample file to the hadoop filesystem **hadoop fs -put sampleinput.txt**

/wordcountfolder

Step 6: Create WordCount.java file **gedit WordCount.java**

Step 7: Copy the WordCount.java to hadoop filesystem

hadoop fs -put WordCount.java /wordcountfolder

Step 8: Run the WordCount.java file

hadoop com.sun.tools.javac.Main WordCount.java

Two new class files will be generated WordCount\$IntSumReducer.class,

WordCount\$TokenizerMapper.class

Step 9: Create a jar file using the class files generated

jar cvf WordCount.jar WordCount*.class

Step 10: Run the jar file with the sample input text

hadoop jar WordCount.jar WordCount /wordcountfolder/sampleinput.txt

/wordcountfolder/output.txt

Step 11: Open the output.txt in the hadoop web UI The part-r file contains the output. Download the file and open it using gedit.

WordCount.java

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;
```

```
import org.apache.hadoop.conf.Configuration;
```

```
import org.apache.hadoop.fs.Path;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper

    extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);

        private Text word = new Text();

        public void map(Object key, Text value, Context context

        ) throws IOException, InterruptedException {

            StringTokenizer itr = new StringTokenizer(value.toString());

            while (itr.hasMoreTokens()) {

                word.set(itr.nextToken());

                context.write(word, one);

            }

        }

        public static class IntSumReducer

        extends Reducer<Text,IntWritable,Text,IntWritable> {

            private IntWritable result = new IntWritable();
```

```

public void reduce(Text key, Iterable<IntWritable> values,
Context context
) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values) {
sum += val.get();
}
result.set(sum);
context.write(key, result);
}
}

public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Word count");
job.setJarByClass(WordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);
}

```

}

}

SampleInput.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Morbi quis commodo odio aenean. Vestibulum mattis ullamcorper velit sed ullamcorper morbi tincidunt. Aliquet eget sit amet tellus cras. Tincidunt lobortis feugiat vivamus at augue eget arcu dictum varius. Mi ipsum faucibus vitae aliquet nec ullamcorper sit amet risus. Adipiscing at in tellus integer feugiat scelerisque. Sed enim ut sem viverra. Quis auctor elit sed vulputate mi sit amet mauris commodo. Nunc congue nisi vitae suscipit tellus mauris. Accumsan tortor posuere ac ut consequat. Eu volutpat odio facilisis mauris sit amet massa vitae tortor. Adipiscing diam donec adipiscing tristique risus. Sit amet mauris commodo quis imperdiet. Sem fringilla ut morbi tincidunt augue interdum. Tellus cras adipiscing enim eu.

Consequat mauris nunc congue nisi vitae suscipit tellus mauris. Facilisi cras fermentum odio eu feugiat pretium. Curabitur gravida arcu ac tortor dignissim convallis aenean et tortor. Faucibus a pellentesque sit amet porttitor eget. Vitae ultricies leo integer malesuada nunc. Commodo ullamcorper a lacus vestibulum sed arcu non. Cras fermentum odio eu feugiat pretium nibh ipsum. Placerat vestibulum lectus mauris ultrices. Netus et malesuada fames ac turpis egestas. Condimentum mattis pellentesque id nibh tortor. Nec ullamcorper sit amet risus. Vitae aliquet nec ullamcorper sit amet risus nullam. Semper auctor neque vitae tempus. Malesuada proin libero nunc consequat. Id leo in vitae turpis.

Ante in nibh mauris cursus mattis molestie. Nibh ipsum consequat nisl vel pretium lectus quam id. Et

malesuada fames ac turpis egestas integer. Venenatis a condimentum vitae sapien pellentesque habitant. Elementum nibh tellus molestie nunc non blandit massa. Nisl purus in mollis nunc. Cursus metus aliquam eleifend mi in nulla posuere sollicitudin. Eu lobortis elementum nibh tellus molestie nunc non. Pellentesque elit eget gravida cum sociis natoque. Lacus viverra vitae congue eu consequat ac. Mattis aliquam faucibus purus in massa tempor.

Felis bibendum ut tristique et egestas quis ipsum suspendisse. Ornare massa eget egestas purus viverra accumsan in nisl. Id aliquet lectus proin nibh nisl condimentum id venenatis. Sed viverra tellus in hac habitasse platea dictumst. Risus pretium quam vulputate dignissim suspendisse in. Elit dui tristique sollicitudin nibh sit amet commodo nulla facilisi. Aliquet porttitor lacus luctus accumsan tortor posuere ac ut consequat. Ultrices sagittis orci a scelerisque. Eget mi proin sed libero enim sed faucibus turpis in. Ac turpis egestas integer eget aliquet nibh.

Tellus pellentesque eu tincidunt tortor aliquam nulla facilisi cras fermentum. Arcu dictum varius dui at consectetur. Viverra tellus in hac habitasse platea dictumst vestibulum. Semper viverra nam libero justo laoreet sit amet cursus. Cras tincidunt lobortis feugiat vivamus at augue. Nisl rhoncus mattis rhoncus urna neque viverra justo. Cum sociis natoque penatibus et magnis dis parturient. Enim blandit volutpat maecenas volutpat blandit aliquam. Et pharetra pharetra massa massa ultricies mi quis hendrerit. Pellentesque pulvinar pellentesque habitant morbi tristique senectus et netus. Ut tellus elementum sagittis vitae et leo dui ut. Amet mattis vulputate enim nulla aliquet. Aliquam purus sit amet luctus venenatis.

Eu tincidunt tortor aliquam nulla facilisi cras fermentum. Purus viverra accumsan in nisl nisi. Aenean et tortor at risus viverra adipiscing at in. Etiam non quam lacus suspendisse faucibus interdum posuere. Amet luctus venenatis lectus magna fringilla. Faucibus et molestie ac feugiat. Etiam sit

amet nisl purus in mollis nunc sed. Sed vulputate odio ut enim blandit volutpat maecenas volutpat blandit. Ac auctor augue mauris augue. Nullam vehicula ipsum a arcu cursus vitae congue mauris. Non sodales neque sodales ut etiam sit amet nisl.

Diam quis enim lobortis scelerisque fermentum dui faucibus in. Euismod quis viverra nibh cras pulvinar mattis nunc sed. Et netus et malesuada fames ac turpis egestas sed tempus. Blandit turpis cursus in hac habitasse platea. Convallis convallis tellus id interdum. Id diam vel quam elementum. Porta non pulvinar neque laoreet. Imperdiet nulla malesuada pellentesque elit eget gravida cum sociis natoque. Habitasse platea dictumst vestibulum rhoncus est pellentesque elit. Lorem ipsum dolor sit amet. Non odio euismod lacinia at. Vitae auctor eu augue ut lectus arcu. Tortor consequat id porta nibh venenatis cras sed. Placerat dui ultricies lacus sed turpis tincidunt id aliquet.

Tortor dignissim convallis aenean et tortor at risus viverra adipiscing. Scelerisque mauris pellentesque pulvinar pellentesque habitant morbi. Sodales ut eu sem integer. Nunc scelerisque viverra mauris in. Eget magna fermentum iaculis eu non. Risus ultricies tristique nulla aliquet enim. Vel facilisis volutpat est velit egestas dui. At auctor urna nunc id cursus metus aliquam eleifend. Purus semper eget dui at tellus at urna condimentum mattis. Arcu dictum varius dui at consectetur lorem donec. Tincidunt nunc pulvinar sapien et ligula ullamcorper. Et odio pellentesque diam volutpat commodo sed egestas egestas fringilla. At elementum eu facilisis sed odio morbi quis commodo. Sed odio morbi quis commodo odio.

Massa massa ultricies mi quis hendrerit dolor magna. Nisl condimentum id venenatis a condimentum vitae sapien pellentesque. Velit dignissim sodales ut eu sem integer. Ornare massa eget egestas purus viverra accumsan in nisl. Commodo ullamcorper a lacus vestibulum sed arcu non. Elementum integer enim neque volutpat. Pulvinar mattis nunc sed blandit libero volutpat. Quis

blandit turpis cursus in hac habitasse platea dictumst. Massa enim nec dui nunc mattis enim ut. Consectetur adipiscing elit ut aliquam purus sit. Proin sagittis nisl rhoncus mattis rhoncus urna neque. Tempus iaculis urna id volutpat lacus. Nunc eget lorem dolor sed viverra. Purus viverra accumsan in nisl. Orci dapibus ultrices in iaculis nunc sed augue lacus viverra. Augue mauris augue neque gravida in fermentum et sollicitudin. Molestie ac feugiat sed lectus vestibulum mattis ullamcorper velit sed.

Felis eget nunc lobortis mattis aliquam. Platea dictumst quisque sagittis purus sit amet volutpat. Non sodales neque sodales ut. Ornare aenean euismod elementum nisi quis eleifend quam adipiscing vitae. Sagittis orci a scelerisque purus semper eget dui. Fermentum leo vel orci porta non. In nibh mauris cursus mattis molestie. Lorem sed risus ultricies tristique nulla aliquet enim tortor at. In hac habitasse platea dictumst vestibulum rhoncus est pellentesque. Mauris pellentesque pulvinar pellentesque habitant morbi tristique. Velit euismod in pellentesque massa placerat dui ultricies lacus sed. Integer feugiat scelerisque varius morbi. Magna etiam tempor orci eu lobortis elementum. Id donec ultrices tincidunt arcu. Massa id neque aliquam vestibulum morbi blandit cursus risus at. Lorem ipsum dolor sit amet consectetur adipiscing elit ut. Massa eget egestas purus viverra.

11 . HADOOP FUSE INSTALLATION

Step 1: Adding the Hadoop fuse repository:

wget http://archive.cloudera.com/cdh5/one-click-install/trusty/amd64/cdh5-repository_1.0_all.deb

Step 2: **sudo dpkg -i cdh5-repository_1.0_all.deb**

Step 3: **sudo apt-get update**

Step 4:Installing Hadoop-hdfs-Fuse

sudo apt-get install hadoop-hdfs-fuse

Step 5:Creating a mount point named Fuse

sudo mkdir -p FUSE

Step 6: Mounting Fuse directory

sudo hadoop-fuse-dfs dfs://localhost:54310 FUSE

Step 7: Display The File System Details

cd FUSE

ls FUSE

Step 8:Unmount HDFS

Type : **umount FUSE** or **umount FUSE**

Step 9: Display File System Details

cd FUSE

ls

12 . GLOBUS TOOLKIT INSTALLATION

1. Install the globus toolkit components as follows:

1.1. **wget http://toolkit.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo_latest_all.deb**

dpkg -i globus-toolkit-repo_latest_all.deb

2. Update using **\$sudo apt-get update**

3. Installing other globus toolkit components:

3.1. myproxy 3.4. globus-gridftp 3.2. myproxy-server 3.5. globus-gram5 3.3. myproxy-admin 3.6.

globus-gsi

```
# apt-get install globus-gridftp globus-gram5 globus-gsi myproxy myproxy-server myproxy-admin
```

3.7. globus-data-management-client 3.8. globus-data-management-server 3.9. globus-data-management-sdk

```
# apt-get install globus-data-management-client globus-data-management-server globus-data-management-sdk
```

3.10. globus-resource-management -server 3.11. globus-resource-management-client 3.12. globus-resource-management-sdk

```
# apt-get install globus-resource-management -server globus-resource-management-client globus-resource-management-sdk
```

3.13. gsi-openssh

```
# apt-get install gsi-openssh
```

13 . GRID FTP

1. Install Virtual box

2. Install a VM with Ubuntu 16.04 desktop (network configurations as done in previous exercises)

3. Check for support of Java (if not, install it as in Ex 3).

4. Install all components of globus toolkit properly as in Ex 12. Later, follow the following steps :

5. Change to root user using **(sudo -i)** and **cd /hom/vm**

```
install -o myproxy -m 664 /etc/grid-security/hostcert.pem /etc/grid-security/myproxy/hostcert.pem
```

```
install -o myproxy -m 664 /etc/grid-security/hostkey.pem /etc/grid-security/myproxy/hostkey.pem
```

6. Edit /etc/myproxy-server.config file to remove comments from the credential repository using

```
nano /etc/myproxy-server.config
```

7. Change usermod as below:

```
#usermod -a -G simpleca myproxy
```

Start myproxy-server service

```
#service myproxy-server start
```

Check status of the server status **#service myproxy-server status**

Check server start on 7512 port by below command

```
#netstat -an | grep 7512
```

8. Execute following commands to know the options for the below commands

```
# man grid-mapfile-add-entry
```

```
# man myproxy-admin-adduser
```

The command to create the myproxy credential for the user is

```
#su - -s /bin/sh myproxy
```

```
$PATH=$PATH:/usr/sbin $myproxy-admin-adduser -c "Gcc LAB" -l root
```

9. User Authorization

create a grid map file entry for this credential, so that the holder of that credential can use it to access globus services

```
# grid-mapfile-add-entry -dn "-----" -l root
```

10. Setting up grid-ftp-start

```
# service globus-gridftp-server start
```

```
# service globus-gridftp-server status
```

11. Check for port 2811

```
netstat -an | grep 2814
```

12. User logon with the passphrase given during step 8.

```
myproxy-logon -s root
```

or

```
myproxy -logon -s *vm1-VirtualBox*
```

13. Transfer file using globus-url-copy command.

```
# globus-url-copy file:///home/.../hello.txt http://localhost:2811/...../Documents
```