JMeter Distributed Test Lab Setup

As part of our lab activities, you are required to run your test scripts using JMeter Distributed Testing. Our lab machines are now equipped with Mint-Linux virtual machines. To proceed, please ensure you sync your virtual machines as follows:

Syncing Virtual Machines

(each team gets three machines, and make sure you use the same lab PC each time).

Note: Use one VM for hosting your app and as a JMeter Master to distribute the test script virtual users among other slave VMs. The other two VMs will be the worker or slave machines.

Create a directory using terminal:
mkdir VBOX

Navigate to the newly created directory:
cd VBOX

Sync the VM using terminal:
rsync -av /home/seng426VMs/vbox/VirtualBox\ VMs/SENG426-Mint .

(Note: make sure to have a space and period in the end, copy/paste is recommended).

Locate the VM files:
Go to /home/VBOX/
You will find SENG426-Mint. Open that folder and double-click on the .vbox file. This will register your VM into VirtualBox.

Start the VM:
(Note: Password is seng426.)
You will find a few things already available on your VM, these are as follows:
  - Required Docker to run your Uranus (UCrypt) app.
  - Visual Studio Code.
  - MySQL Client to manage your database.
  - Uranus App as a sample on desktop.
  - An Angular Sample App.
  - JMeter version 5.6.3 (You are supposed to use this one only).
  - Java JDK (Do not change this; it should be of equal version on all the VMs to run your JMeter).

Setting Up Lab for Distributed Testing

Requirements:

- **JMeter Installed**: Ensure JMeter is installed on all Mint-Linux virtual machines. Use the package manager or download it from the official website. (We have this.)

- **Java Installed**: JMeter requires Java. Ensure that Java is installed and properly configured on all machines. (We have this.)
- **Network Configuration**: All machines must be on the same network and should be able to communicate with each other. (We will do this.)
- **Firewall Settings**: Configure the firewall (using ufw) to allow the necessary ports for JMeter communication (default port 1099 and other configured ports). (We will do this as well.)

Network Configuration:

For two different VMs on two different host machines to communicate, they must be on the same subnet and netmask. When you start the VM, it gets an IP based on the host machine. To check, use the terminal command:
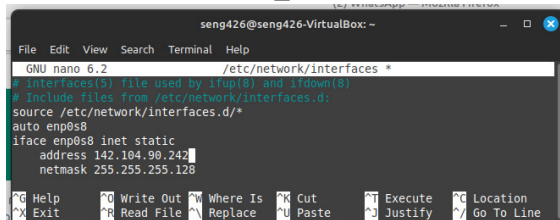
ifconfig

You will see that the enp0s8 inet IP is something like 142.104.90.223. We shouldn't have the same IP on all VMs, so we will set a static IP and NetMask in /etc/network/interfaces. To do that, run this command in terminal:

sudo nano /etc/network/interfaces

Paste the following lines with your new IP address and netmask:

auto enp0s8
iface enp0s8 inet static
*   address   <Your_Static_IP>*
*   netmask   <Your_Netmask>*

(Note: Replace *<Your_Static_IP>*, and *<Your_Netmask>* with the provided IP and netmask.)

To save the changes, hit Ctrl + O, then Enter, and Ctrl + X. Now reboot your VM and check the IP in the terminal using *ifconfig*. You will see a new IP, and you are good to proceed to the next step.

Firewall Settings:

JMeter requires some ports to be open (e.g., 1099, 4455, etc.), but it also uses random ports. To avoid issues, we will turn off the firewall on our VMS. On the Virtual Machine, after you restart it, run this command in the terminal:

sudo ufw disable

This will disable the firewall and allows JMeter instances to communicate across different VMs.

Configure JMeter's Master and Slave Machines
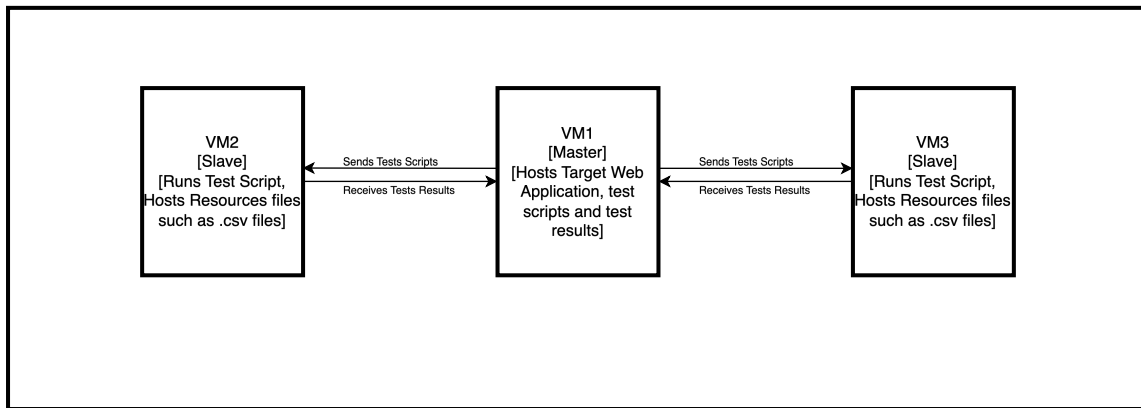
Configure Slave Machines: (Repeat this for Second Slave Machine as well)

Edit jmeter.properties File inside the JMeter/bin Directory:

Inside the file, search for the line *rmi.ssl.disable*, uncomment this line, and set it to true to skip SSL-based Distributed Testing.

Save the file.

Run the JMeter Server on Slave Machine:

Open the terminal in the JMeter/bin directory and run:

sh jmeter-server

If you see an error "Cannot start. seng426-VirtualBox is a loopback address.", solve this temporarily by running:

sh jmeter-server -Djava.rmi.server.hostname=*<Slave_Machine_IP>*

(Note: You will need to repeat this each time you start the jmeter-server on the slave machine.)

Permanent Solution:

Open your hosts file in the terminal:

sudo nano /etc/hosts

Change the addresses of both localhost and seng426-VirtualBox to the IP of your slave machine.

Save the file (Ctrl + O, Enter, and Ctrl + X).

Repeat this step on the Master as well for consistency.

Test the Server:

Run the server on the Slave Machine again:

sh jmeter-server

Configure Master Machines:

Edit jmeter.properties File inside the JMeter/bin Directory**:**
Inside the file, search for *rmi.ssl.disable*, uncomment this line, and set it to true.
Save the file.

Now search for remote_hosts= . and paste your slave machines' IPs separated by commas, for example:
*remote_hosts=142.104.90.190, 142.104.90.191*

```
266
267   # Remote Hosts – comma delimited
268   remote_hosts=142.104.90.190, 142.104.90.191
269   #remote_hosts=localhost:1099,localhost:2010
270
271   # RMI port to be used by the server (must start rmiregistry with same port)
```

Save the file.

Trouble Shooting:

- Now verify that if the Master can access to Slaves and vice versa, via telnet.
- To do that, in a terminal of the Master VM run: *telnet <IP-Address-of-the-VM-you-want-to-access> port number*
  (Note: Jmeter Server must be running. When you run the Jmeter Server on a Slave VM, you will see the IP Address followed by ":PortNumber"), you can use that IPAddress and Port number, (replace : with space) to telnet).
- If the result is "connected" then you may proceed to the next step.
- If not then you probably need to make some changes in all of your VMs' network section before running them. Follow these steps to do that:
- Turn of the VMs and then open its settings, navigate to Network section > Adapter 1 and then Advanced, click multiple time on load reload sign against the MAC Address to get a new virtual MAC Address, copy that MAC Address and then Navigate to Adapter 2 > Advanced and then change Promiscuous Mode to "Allow All" and paste the MAC Address and then click on Ok to save the settings. (Note: Avoid making any other changes.)
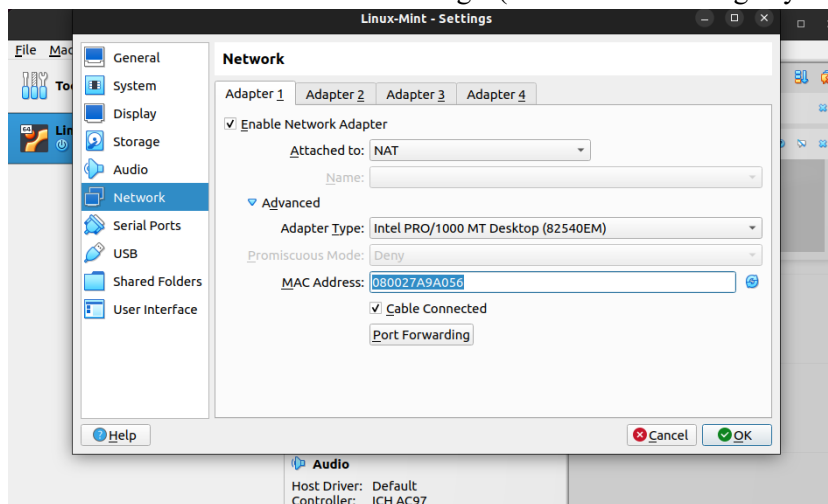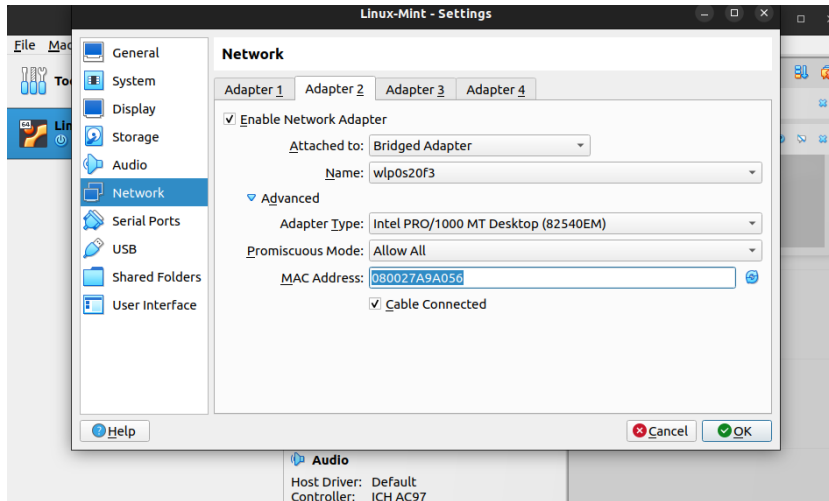
- Repeat this for all the VMs and then restart and verify using telnet again.

Running Test Scripts on Distributed Testing

To run the test scripts on slave machines, you don't need the scripts on the slave machines, but any resources (e.g., processing files) should be present on the slave machines.

There are two ways to run the test scripts: GUI and Non-GUI. For real testing purposes, always use the non-GUI way.

Non-GUI Method:
Run the following command from the Master Machine:
Inside the JMeter/bin folder:

- For running on specific slave machines:

  sh jmeter -n -t [path/to/testplan.jmx] -l [path/to/results.jtl] -R<slave1-IP>, <slave2-IP>

  e.g.: sh jmeter -n -t tests/signUp.jmx -l tests/results.jtl -R142.104.90.190, 142.104.90.191

- For running on all slave machines:

  sh jmeter -n -t [path/to/testplan.jmx] -l [path/to/results.jtl] -r

  e.g: sh jmeter -n -t tests/signUp.jmx -l tests/results.jtl -r

- For closing the server on slave machines after they finish testing.

  Put "-X" in the end:

  sh jmeter -n -t [path/to/testplan.jmx] -l [path/to/results.jtl] -r -X

  e.g.: sh jmeter -n -t tests/signUp.jmx -l tests/results.jtl -r -X

- For having a flashy HTML report.

sh jmeter -n -t *[path/to/testplan.jmx]* -l *[path/to/results.jtl]* -e -o *[path/to/HTML-results]* -r -X

e.g: sh jmeter -n -t tests/signUp.jmx -l tests/results.jtl -e -o tests/results -r -X

(Visit results folder to view the HTML report).

GUI Method:

1. Open JMeter on the Master Machine.

2. Load your test script.

3. Start the test using the remote start option, which will distribute the load across the configured slave machines.

Make sure to verify all machines are properly connected and configurations are set correctly. If you encounter any issues, please refer to the detailed setup guide provided.

IP Address for testing - ELW B203 use only

Make sure every group uses their 3 IP addresses only, double check your group number on Bright Space.

Netmask: 255.255.255.128
Gateway: 142.104.90.129

IP Addresses:

Group	IP Address 1	IP Address 2	IP Address 3
Group1	142.104.90.160	142.104.90.161	142.104.90.162
Group2	142.104.90.165	142.104.90.166	142.104.90.167
Group3	142.104.90.170	142.104.90.171	142.104.90.172
Group4	142.104.90.175	142.104.90.176	142.104.90.177
Group5	142.104.90.180	142.104.90.181	142.104.90.182
Group6	142.104.90.185	142.104.90.186	142.104.90.187
Group7	142.104.90.190	142.104.90.191	142.104.90.192
Group8	142.104.90.195	142.104.90.196	142.104.90.197
Group9	142.104.90.200	142.104.90.201	142.104.90.202
Group10	142.104.90.205	142.104.90.206	142.104.90.207
Group11	142.104.90.210	142.104.90.211	142.104.90.212
Group12	142.104.90.215	142.104.90.216	142.104.90.217
Group13	142.104.90.220	142.104.90.221	142.104.90.222
Group14	142.104.90.225	142.104.90.226	142.104.90.227
Group15	142.104.90.230	142.104.90.231	142.104.90.232
Group16	142.104.90.235	142.104.90.236	142.104.90.237