

**SECURED-T**

**Installation Guide**

**Version 2.0**

**[03 August 2017]**

**Contents**

**LIST OF ILLUSTRATIONS iii**

**Overview of Secured-T 1**

**Pre-requisites 1**

**Setting up VMware Workstation & Virtual Machine 1**

**Install The Prerequisites Of Secured-T 7**

**Mininet guide 10**

**Create the Mininet topology with Miniedit 10**

**Editing the Mininet Topology file for Secured-T System 12**

**RYU Controller Guide 16**

**Download The Required Ryu Controller Scripts 16**

**DNS Proxy Guide 17**

**Creating and Configuring the DNS Proxy 17**

**Tomcat Installation and Configuration 18**

**Prepare a tomcat management account 19**

**Setting Up mod\_JK 19**

**MySQL Workbench Installation guide 20**

**Adding Connection On MySQL Workbench 20**

**Create Database Tables 23**

**Deployment guide 24**

**Create a shell script 24**

**Deploying websites 25**

#### **LIST OF ILLUSTRATIONS**

**Figures**

Figure 1: Ubuntu ISO 16.04.2 LTS Download Page

Figure 2: Create A New Virtual Machine

Figure 3: Type Of Configuration

Figure 4: Install From Location

Figure 5: Specify Disk Capacity

Figure 6: Ready To Create Virtual Machine

Figure 7: Add A New Network Adaptor

Figure 8: Add A Custom Network (VMnet0)

Figure 9: Edit Settings For NAT

Figure 10: Check Network Adaptor Configuration

Figure 11: Compare MAC Address to Device

Figure 12: Set IP Address Of Bridged Mode

Figure 13: Take Snapshot

Figure 14: Extract Mininet Archive

Figure 15: Edit Interfaces In ISC-DHCP-Server (Before)

Figure 16: Edit Interfaces In ISC-DHCP-Server (After)

Figure 17: Comment Lines In DHCPD Configuration (Before)

Figure 18: Comment Lines In DHCPD Configuration (After)

Figure 19: Uncomment A Line In DHCPD Configuration (Before)

Figure 20: Uncomment A Line In DHCPD Configuration (After)

Figure 21: Add Lines In DHCPD Configuration

Figure 22: Checking IP Address And Subnet Of NAT And Bridged

Figure 23: Topology

Figure 24: MiniEdit Properties

Figure 25: Add External Interface Eth0

Figure 26: Import OVSSwitch (Before)

Figure 27: Import OVSSwitch (After)

Figure 28: Define Switch (Before)

Figure 29: Define Switch (After)

Figure 30: Change To Ryu Controller (Before)

Figure 31: Change To Ryu Controller (After)

Figure 32: Edit The Switches (Before)

Figure 33: Edit The Switches (After)

Figure 34: Edit The Host (Before)

Figure 35: Edit The Host (After)

Figure 36: Edit The Controllers (Before)

Figure 37: Edit The Controllers (After)

Figure 38: Configure Switches and Hosts (Before)

Figure 39: Configure Switches and Hosts (After)

Figure 40: Closed Mininet Settings (Before)

Figure 41: Closed Mininet Settings (After)

Figure 42: Disable DNS Cache (Before)

Figure 43: Disable DNS Cache (After)

Figure 44: Edit System Name And Path

Figure 45: Change Port Number Of Tomcat

Figure 46: Edit Tomcat Configuration File

Figure 47: Remove Connector

Figure 48: Define A Real Worker With Properties

Figure 49: JkWorkersFile Properties

Figure 50: Configure Apache2 Web Server

Figure 51: Users and Privileges

Figure 52: Database Account Credentials

Figure 53: Grant Rights To Perform All Tasks

Figure 54: Add Database Connection

Figure 55: Add Schema

Figure 56: Create Schema

Figure 57: Set Schema As Default Schema

Figure 58: T\_ANNOUNCEMENT

Figure 59: T\_ASSESSMENT

Figure 60: T\_LECTURER\_INFO

Figure 61: T\_MODULE

Figure 62: T\_STUDENT\_INFO

Figure 63: T\_WHITELIST\_DOMAIN

Figure 64: Script To Run All Four Programs

Figure 65: All Four programs running

Figure 66: Tomcat8 Default Page

Figure 67: Tomcat8 Login

Figure 68: Deploy War File

Figure 69: Webapp Manager

**Tables**

Table 1: Software/Program And The Corresponding Filename

#### **Overview of Secured-T**

Secured-T provides a secured and controlled test environment which allows the students to use their personal notebook during an assessment to access online resources while preventing them from cheating.

#### **Pre-requisites**

* VMware Workstation
* Mininet
* Python IDLE 2.7
* RYU Controller
* DNS Proxy
* MySQL Workbench
* Apache Tomcat 8
* Apache2 Web Server
* Netbeans IDE 8.2
* Mars Eclipse

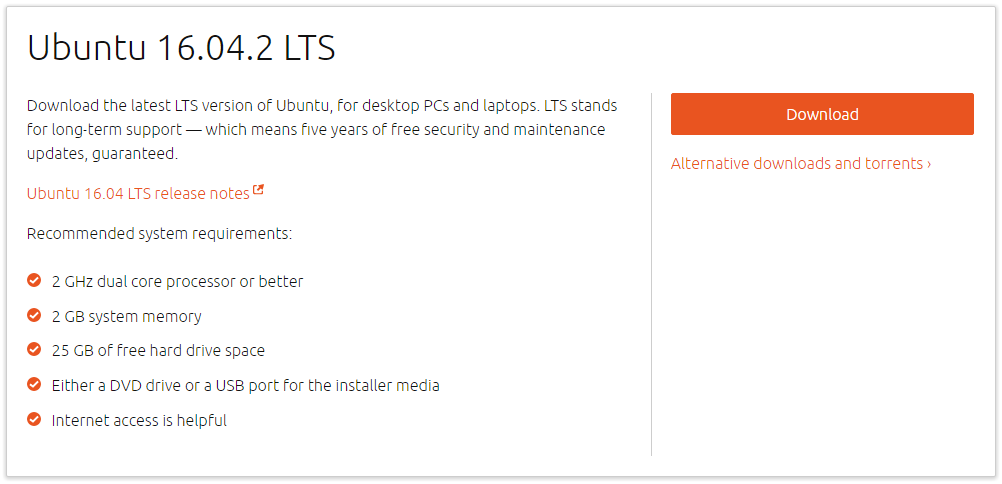
#### **Setting up VMware Workstation & Virtual Machine**

1. Download VMware Workstation Pro and install it.

(<http://www.vmware.com/ap/products/workstation/workstation-evaluation.html>)

1. Download Ubuntu ISO 16.04.2 LTS file. This will be used as the Virtual Machine base.

(<https://www.ubuntu.com/download/desktop>)



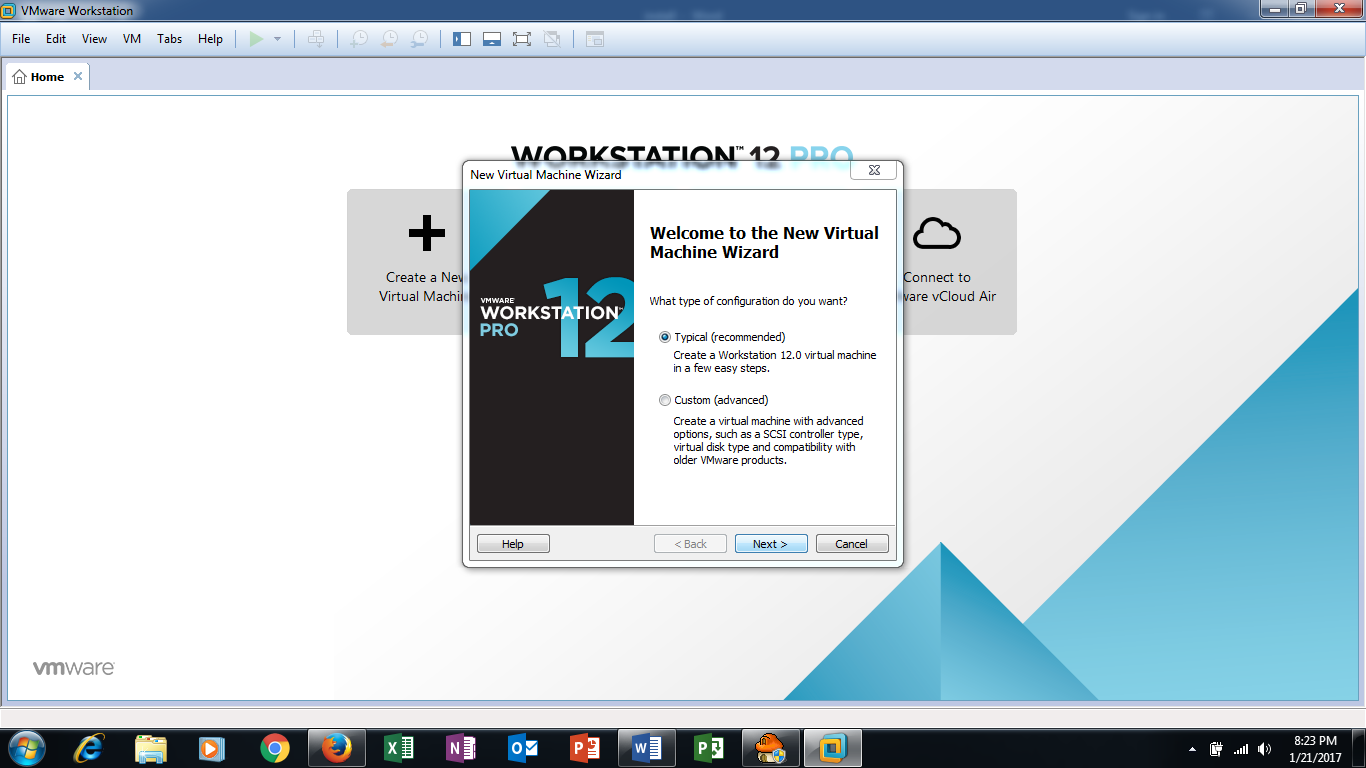
**Figure 1: Ubuntu ISO 16.04.2 LTS Download Page**

1. Launch VMware Workstation after downloading the ISO file.
2. Select “**Create a New Virtual Machine”**



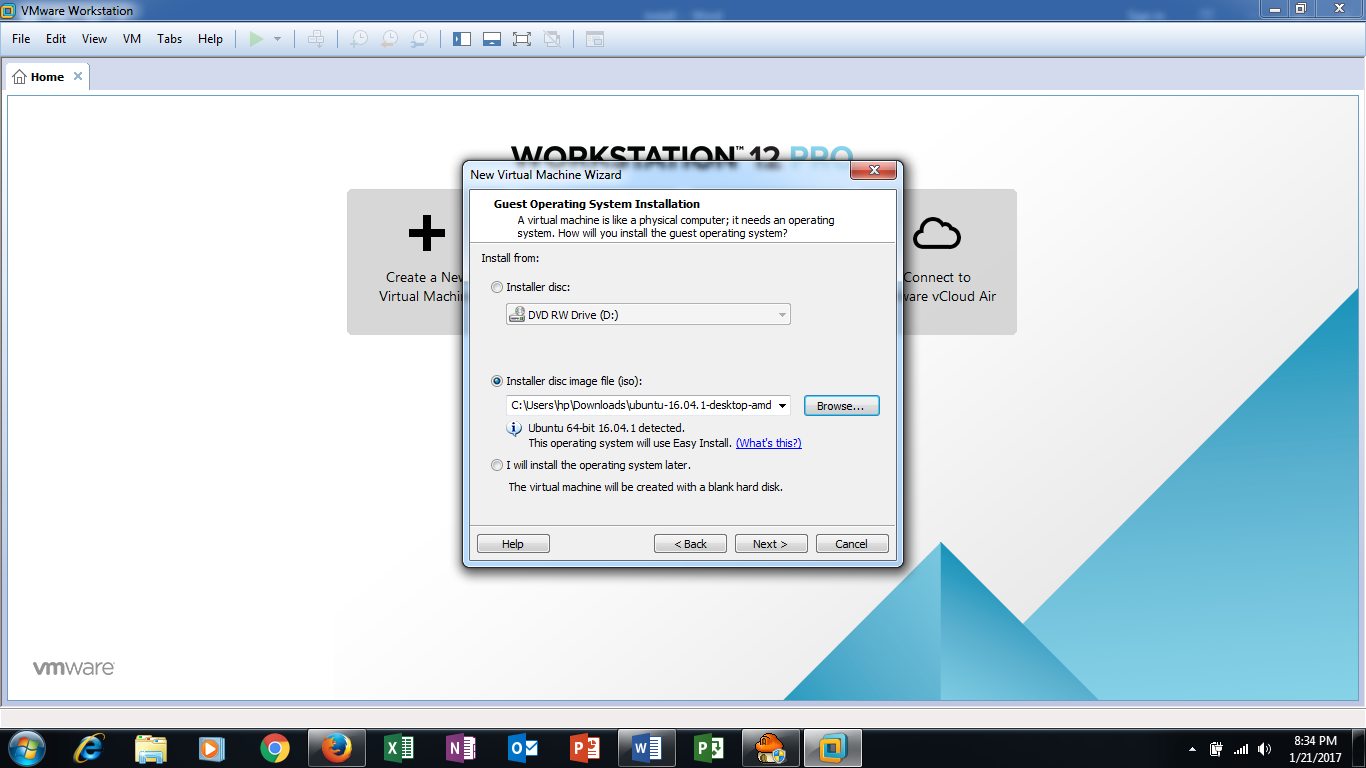
**Figure 2: Create A New Virtual Machine**

1. Select the “**Typical**” Option.



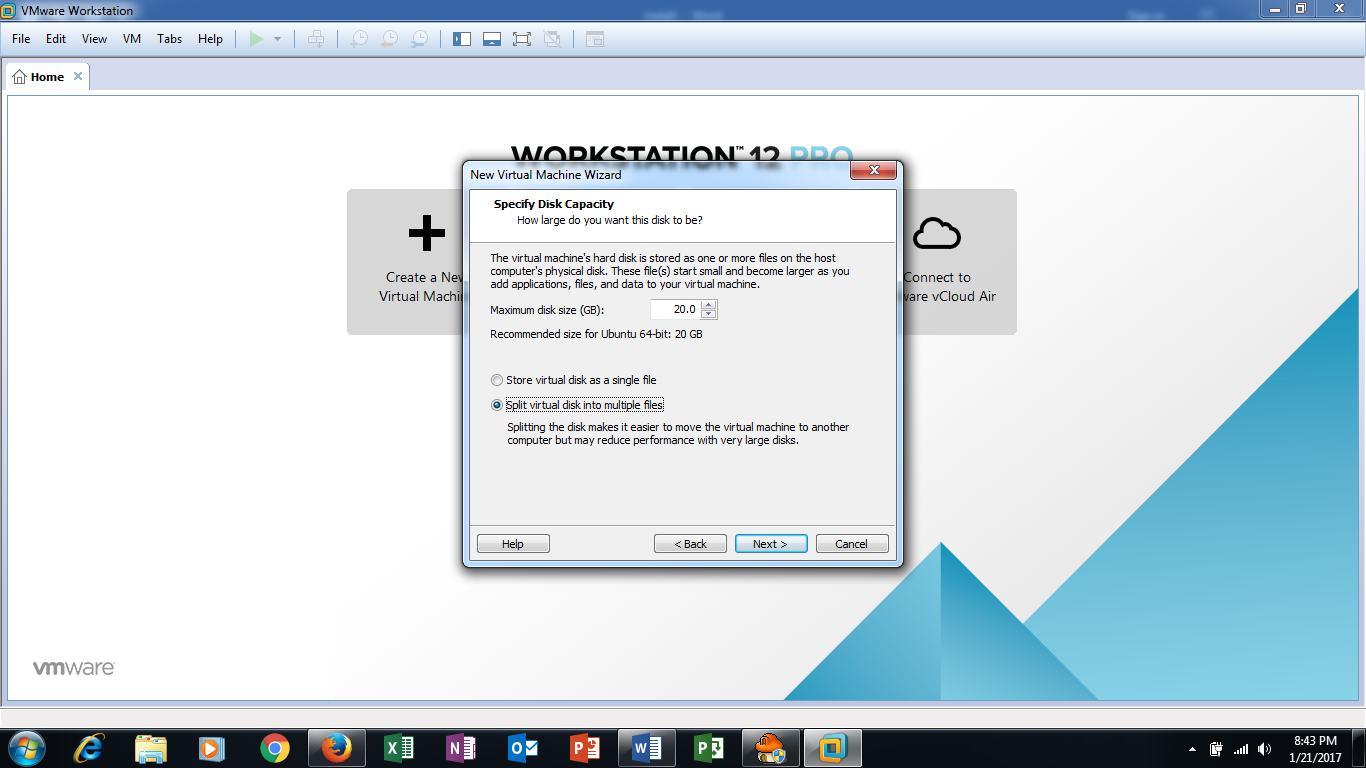
**Figure 3: Type Of Configuration**

1. Select “**Installer disc image file (iso)**” and browse for the Ubuntu ISO file downloaded earlier.



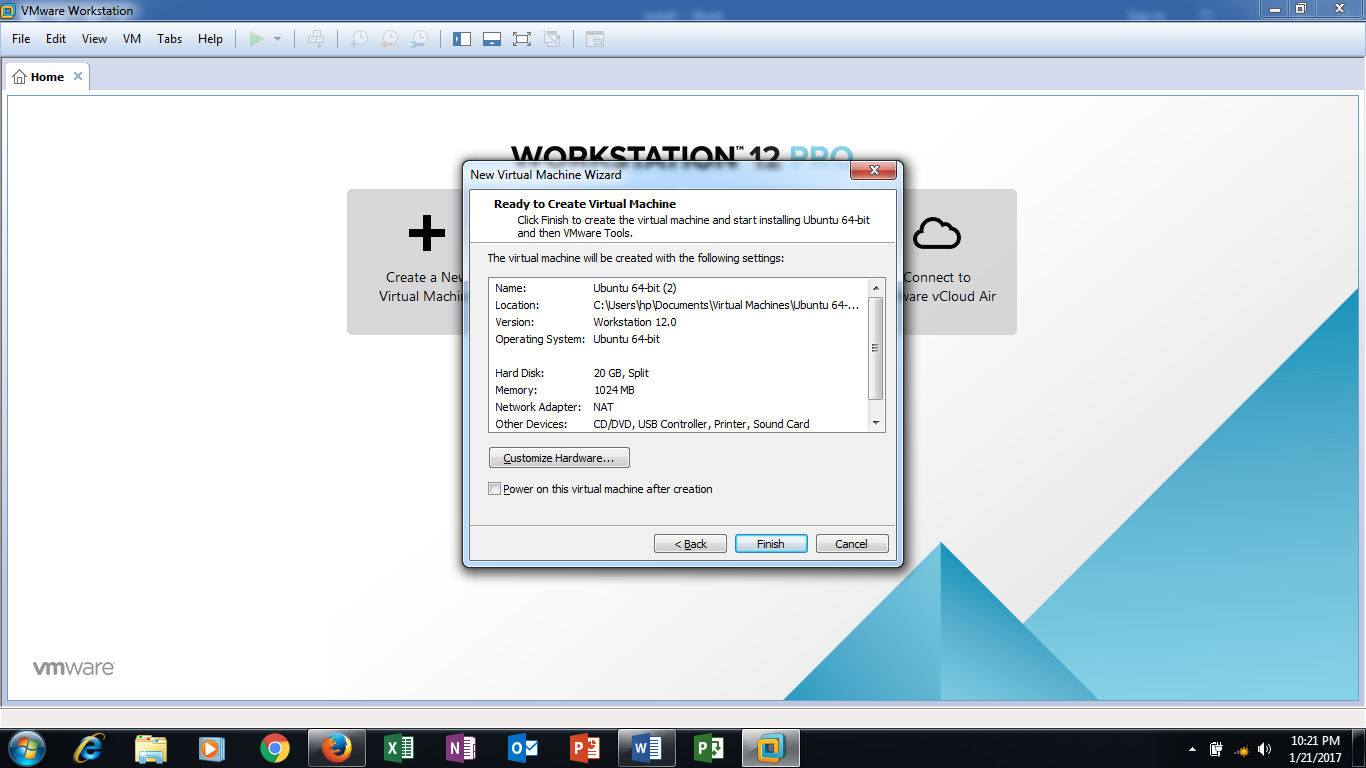
**Figure 4: Install From Location**

1. Give the Virtual Machine a name, username and password. The username and password would be used for your account. Take note that username should only contain lowercase letters (a-z) and/or numbers (0-9).
2. Leave the Virtual machine name and location alone unless otherwise specified.
3. Leave the maximum disc size at **20.0 GB** and let it to **Split virtual disc into multiple files**.



**Figure 5: Specify Disk Capacity**

1. Uncheck “**Power on this virtual machine after creation**” and click **Customize Hardware**.



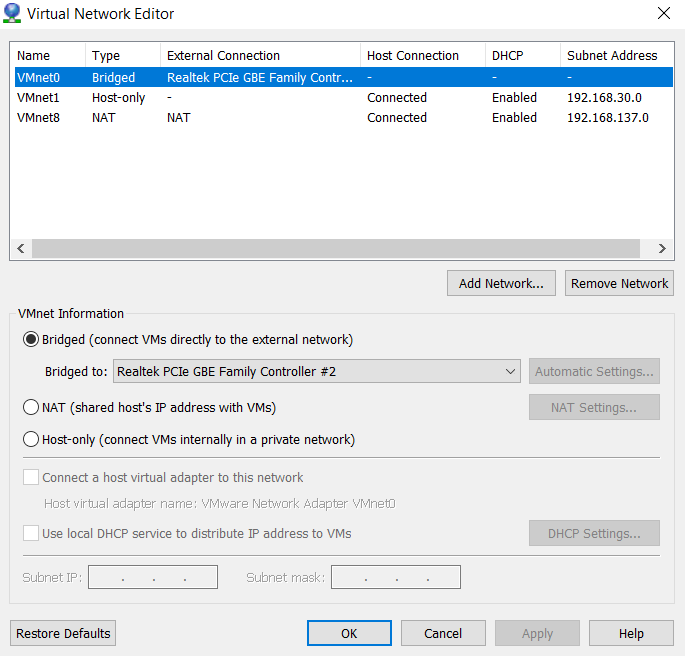
**Figure 6: Ready To Create Virtual Machine**

1. Add a new **Network Adapter** and set the memory to **2GB**. Finish the creating of the Virtual Machine.



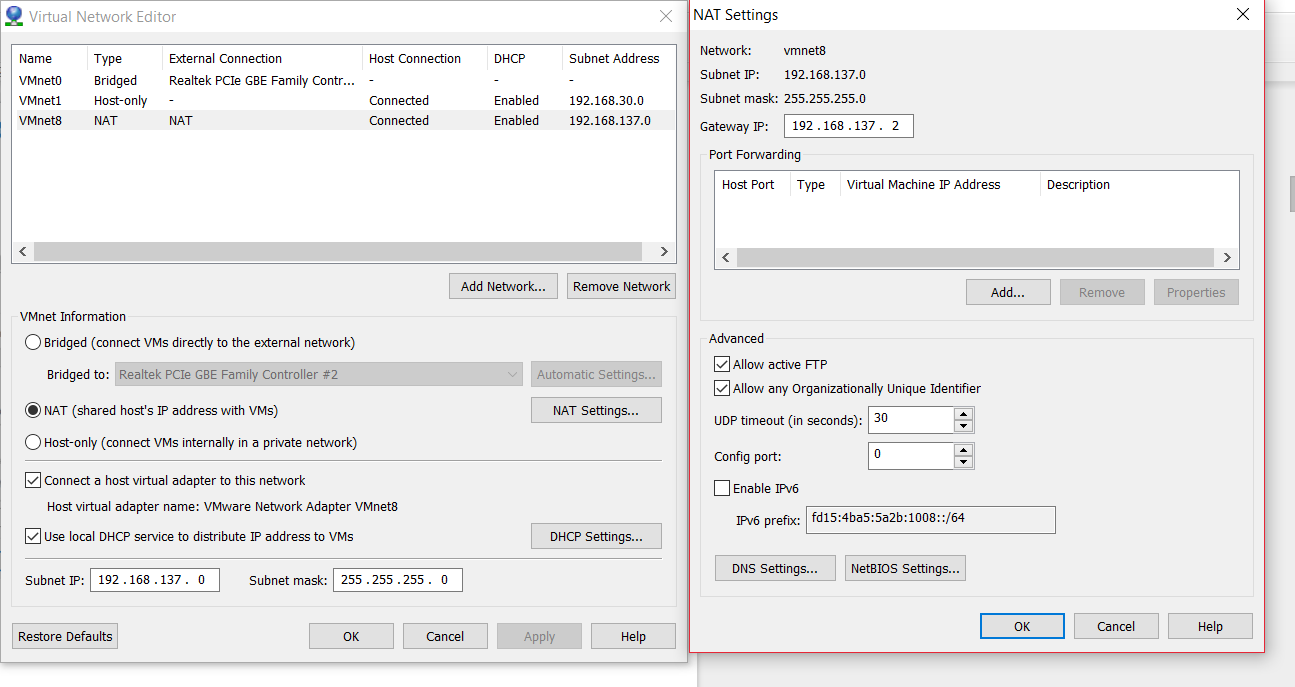
**Figure 7: Add A New Network Adaptor**

1. Change the hardware setting of the Virtual Machine. Set the first Network Adaptor to be **VMnet8** with type set as **NAT** and the second Network Adaptor **VMnet0** to be in **Bridged** for vm bridge mode (recommended) or **Host Only** for windows bridge mode.
   1. Go to **Edit > Virtual Network Editor**.
   2. Add a custom network, **VMnet0**, with type set as **Bridged**. This custom network will be used to connect to the router.
   3. Select your base machine’s ethernet port that is going to be connected to the router in the bridged to drop down box



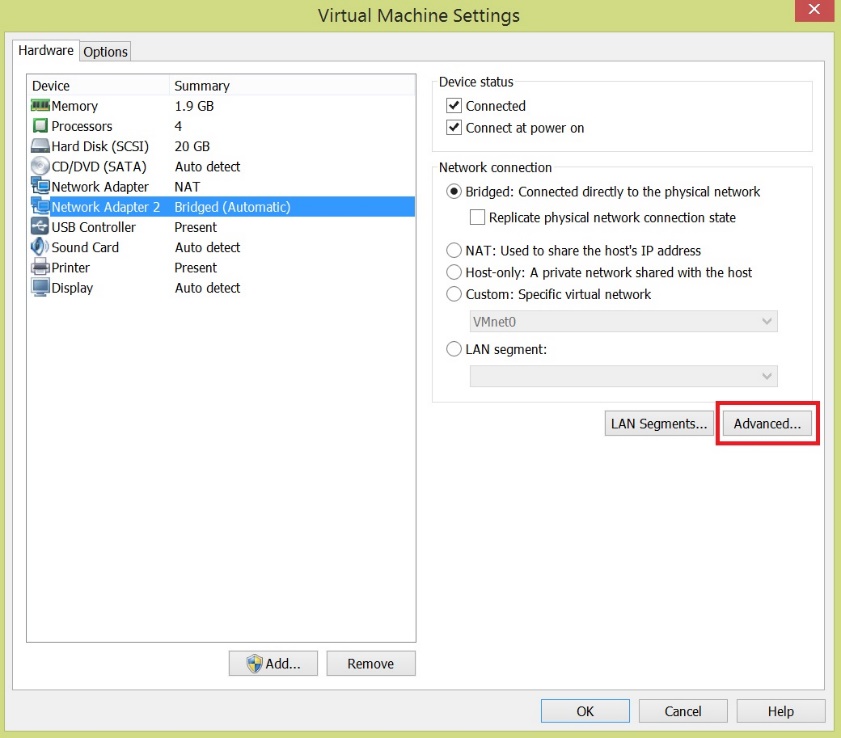
**Figure 8: Add A Custom Network (VMnet0)**

* 1. Edit the settings for NAT like the image below



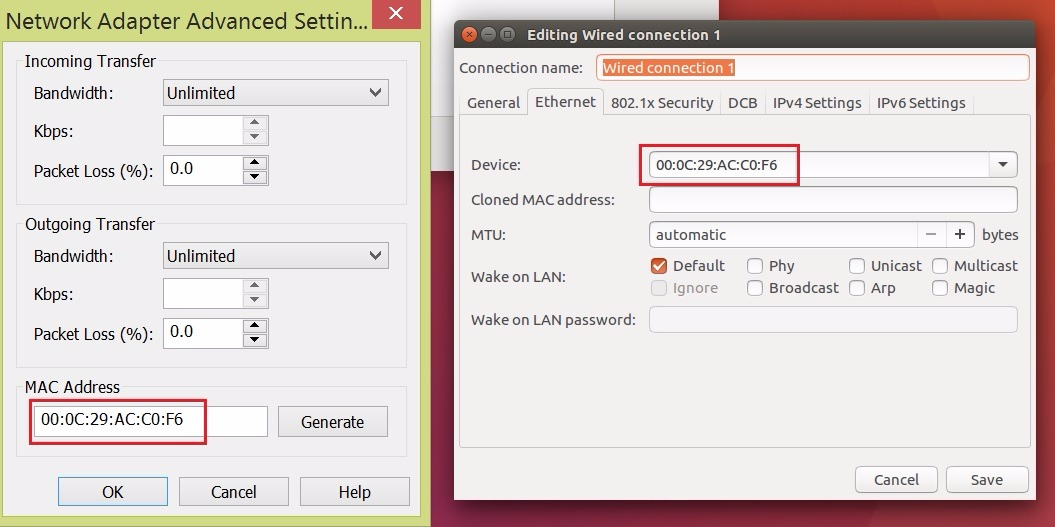
**Figure 9: Edit Settings For NAT**

* 1. The Second Network Adaptor must be set to use a static IP address. To determine which network adapter is configured for NAT and which is for Bridged mode is easy.
  2. Go to **Edit > Virtual Network Editor**.
  3. Under Network Adapter 2, go to Advanced.



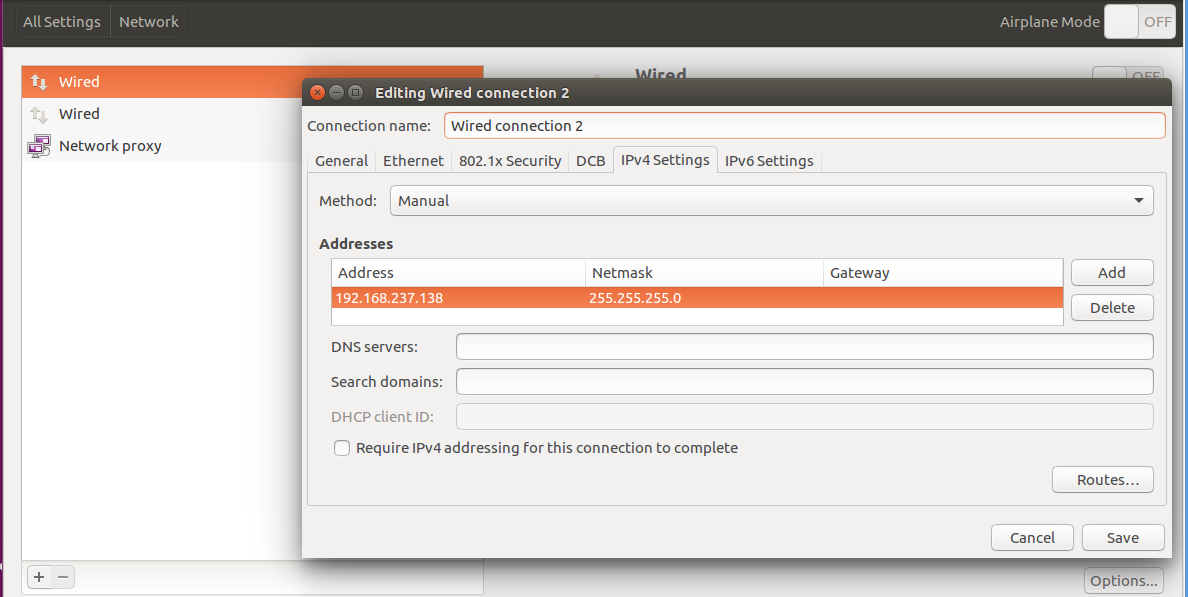
**Figure 10: Check Network Adaptor Configuration**

* 1. Compare the MAC Address in the Network Adaptor Advanced Settings with the wired connection’s device found in system settings menu.



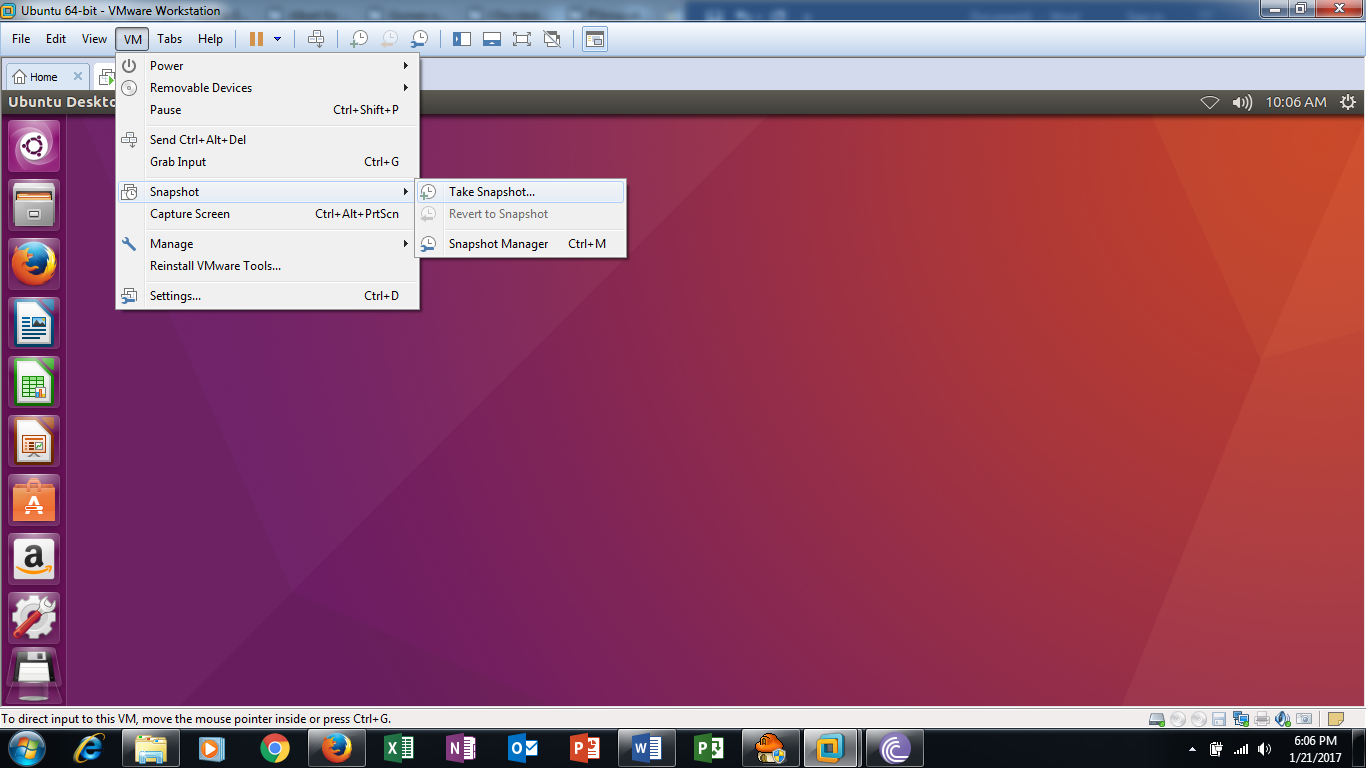
**Figure 11: Compare MAC Address to Device**

* 1. After finding out which wired connection is the Bridged mode, set the IP address of the Bridged mode to: ***192.168.237.138***.
  2. For this interface, you do not need to set the DNS nor the gateway.

.

**Figure 12: Set IP Address Of Bridged Mode**

1. Power on the Virtual Machine and let it fully install.
2. Login as the user account you have created at (g) and take a snapshot. A snapshot is used as a backup to fall back on if an irremovable error were to occur.



**Figure 13: Take Snapshot**

1. After the snapshot has finished backing up, restart the Virtual Machine.

#### **Install The Prerequisites Of Secured-T**

1. Open a terminal.
2. Make a submission folder for students to be able to upload their assessment to the server.
   1. Command: ***mkdir submission***
3. Install any updates that Ubuntu may had missed out during the installation. Use the following commands one-by-one.
   1. Command: ***sudo apt-get update***

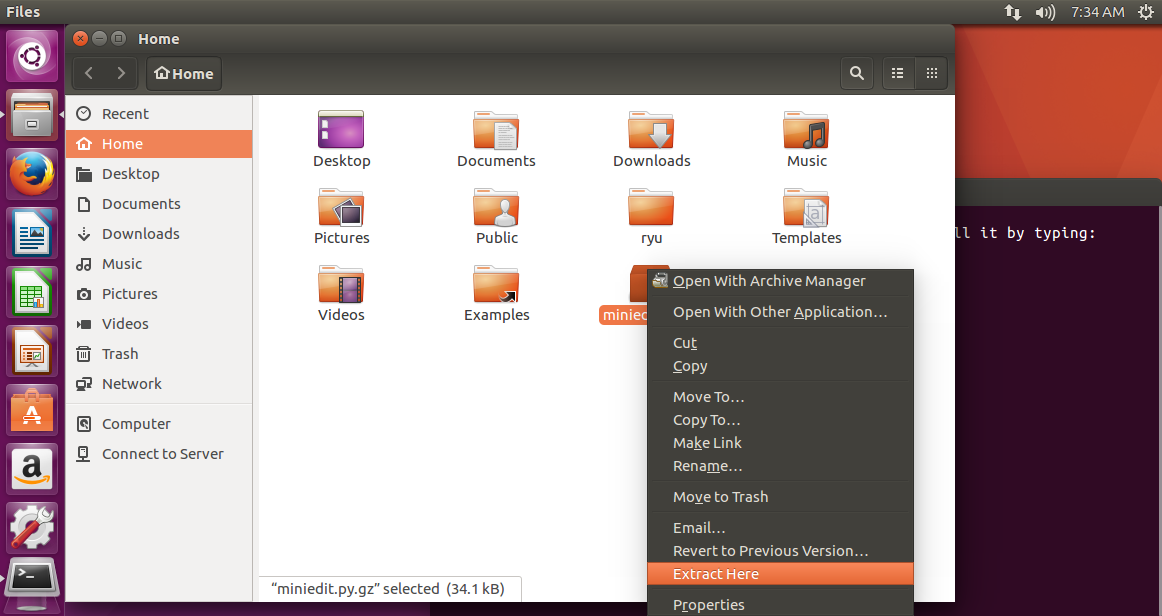
***sudo apt-get upgrade***

***sudo apt-get dist-upgrade***

1. Install pip and git. They are needed to install certain programs easier than apt-get.
   1. Command: ***sudo apt-get install python-pip***

***sudo apt-get install git***

1. Install Mininet. This is used to craft a topology for bridging between the NAT and bridged network.
   1. Command: ***sudo apt-get install mininet***
2. Move an archived file to the home. This archived file is used to easily craft the topology used for the Mininet.
   1. Command: ***cp /usr/share/doc/mininet/examples/miniedit.py.gz ./***
3. Open file explorer and extract the archive in the home tab.



**Figure 14: Extract Mininet Archive**

1. Install python 2.7 IDLE editor.
   1. Command: ***sudo apt-get install idle***
2. Install the Ryu library. The controller for switches runs on Ryu for the setup. Ensure you are in the home directory when running the commands.
   1. Command: ***pip install ryu***

***sudo pip install ryu faucet***

***sudo git clone*** [***https://github.com/osrg/ryu.git***](https://github.com/osrg/ryu.git)

1. Install the dnslib library and dpkt. These are required as we will implement our own DNS server in our system.
   1. Command: ***pip install dnslib***

***pip install dpkt***

1. Install the pymysql library. This is required as it will connect DNS to mysql database.
   1. Command: ***sudo pip install pymysql***
2. Install the following as they are prerequisites for running python programs.
   1. Command: ***sudo apt-get install python-tk***

***sudo pip install datetime***

1. Install the Apache2 web server. This is used to host the default web page.
   1. Command: ***sudo apt-get install apache2***
2. Install the DHCP server. This is used to provide DHCP addresses to student notebooks.
   1. Command: ***sudo apt-get install isc-dhcp-server***
3. Get the interface name of the host-only network with *ifconfig*. For this guide, it is ***eth1\****. Edit the ***isc-dhcp-server*** file in ***/etc/default***.
   1. Command: ***sudo gedit /etc/default/isc-dhcp-server***

**BEFORE**



**Figure 15: Edit Interfaces In ISC-DHCP-Server (Before)**

**AFTER**

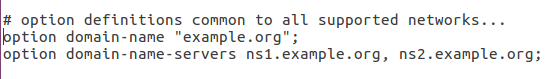


**Figure 16: Edit Interfaces In ISC-DHCP-Server (After)**

***\*****You need to identify the correct network interface name using ifconfig, it may be ens34.*

1. Edit the ***dhcpd.conf*** in ***/etc/dhcp***. For this guide, assume that the bridged subnet is **192.168.237.0/24**.
   1. Command: ***sudo gedit /etc/dhcp/dhcpd.conf***
   2. Comment out the domain name and domain name server option

**BEFORE**



**Figure 17: Comment Lines In DHCPD Configuration (Before)**

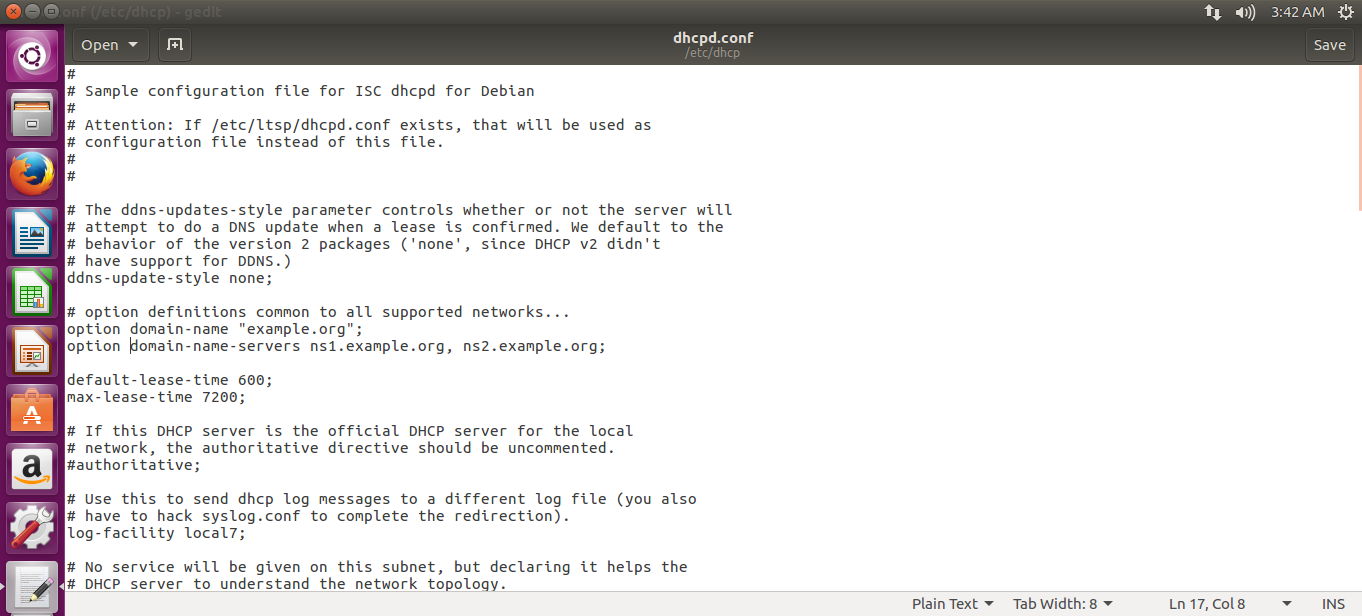
**AFTER**



**Figure 18: Comment Lines In DHCPD Configuration (After)**

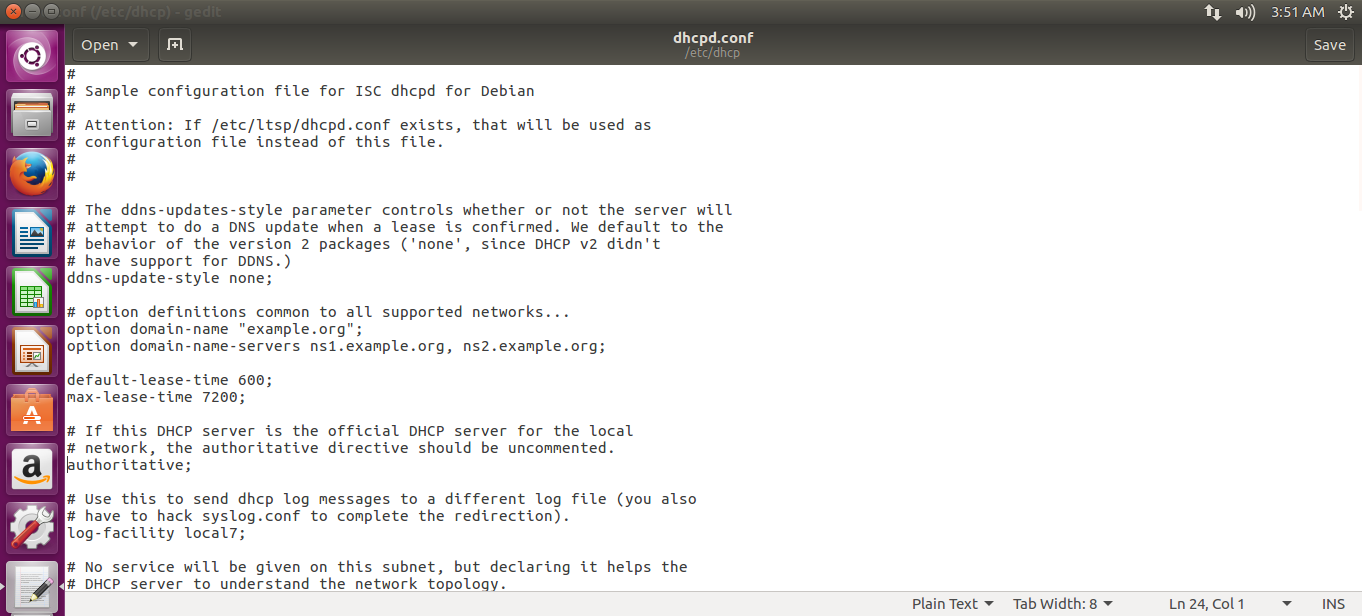
* 1. Uncomment the last line: authoritative

**BEFORE**



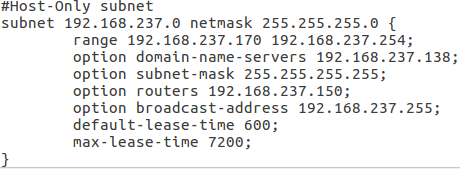
**Figure 19: Uncomment A Line In DHCPD Configuration (Before)**

**AFTER**



**Figure 20: Uncomment A Line In DHCPD Configuration (After)**

* 1. Add the following lines at the end of the file.



**Figure 21: Add Lines In DHCPD Configuration**

* 1. The subnet of each IP in the file should be changed accordingly, however everything else must be the same.\*

\*The domain-name-server setting should be the Bridged network, in this case: **192.168.237.138**

1. After configuration is done, start up the DHCP service.
   1. Command: ***sudo service isc-dhcp-server start***
2. Use the following command to check if the isc-dhcp-server is up.
   1. Command:***sudo service –status-all***
3. At this point, you may connect another Virtual Machine in the same notebook (via host-only mode ) or Physical System (via bridge mode) to test your dhcp server. Please note that your gateway – <your subnet>.150 is not set up yet.

#### 

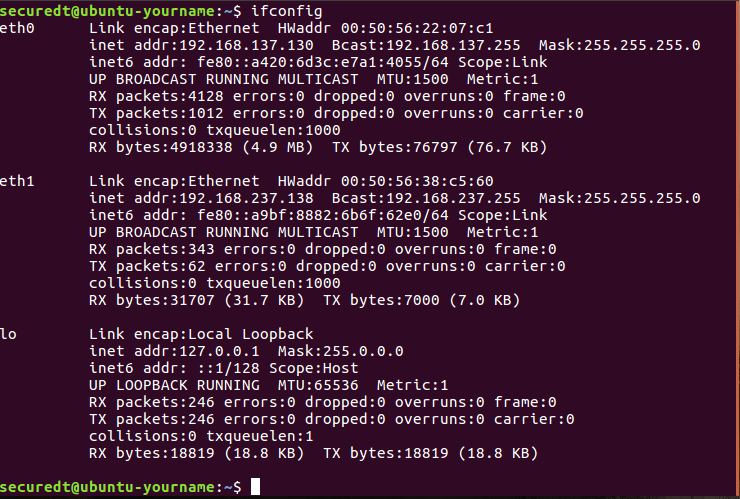
#### **Mininet guide**

Overview:

The purpose of this guide is to guide the user to set up and configure Mininet. Refer to Install The Prerequisites of Secured-T part (e) to install the Mininet before moving on.

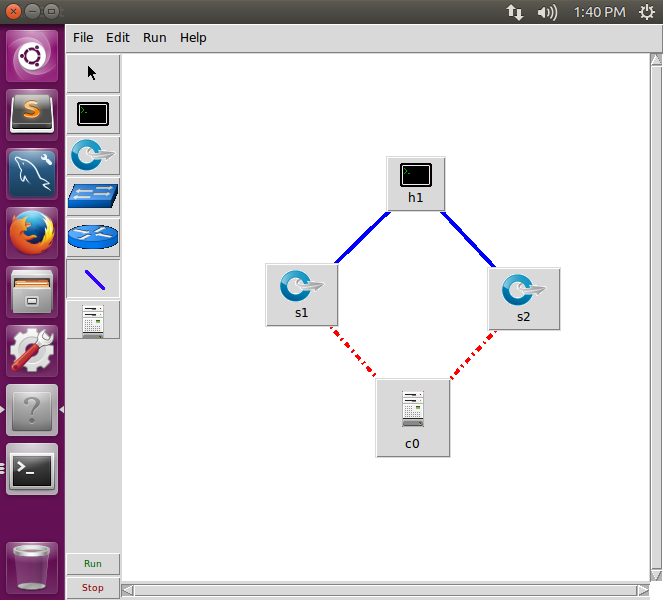
#### **Create the Mininet topology with Miniedit**

1. Change the permission of **miniedit.py** so that it could be executed.
   1. Command: ***sudo chmod +x miniedit.py***
2. Find out the network interfaces present. If Setting up VMware Workstation & Virtual Machine is followed, there should be 2 network interfaces (excluding lo).
   1. Command: ***ifconfig***
   2. The following will be shown. Take down the IP addresses and subnets of the interfaces that is NAT and Bridged.
3. In this case: eth0 is NAT while eth1 is Bridged



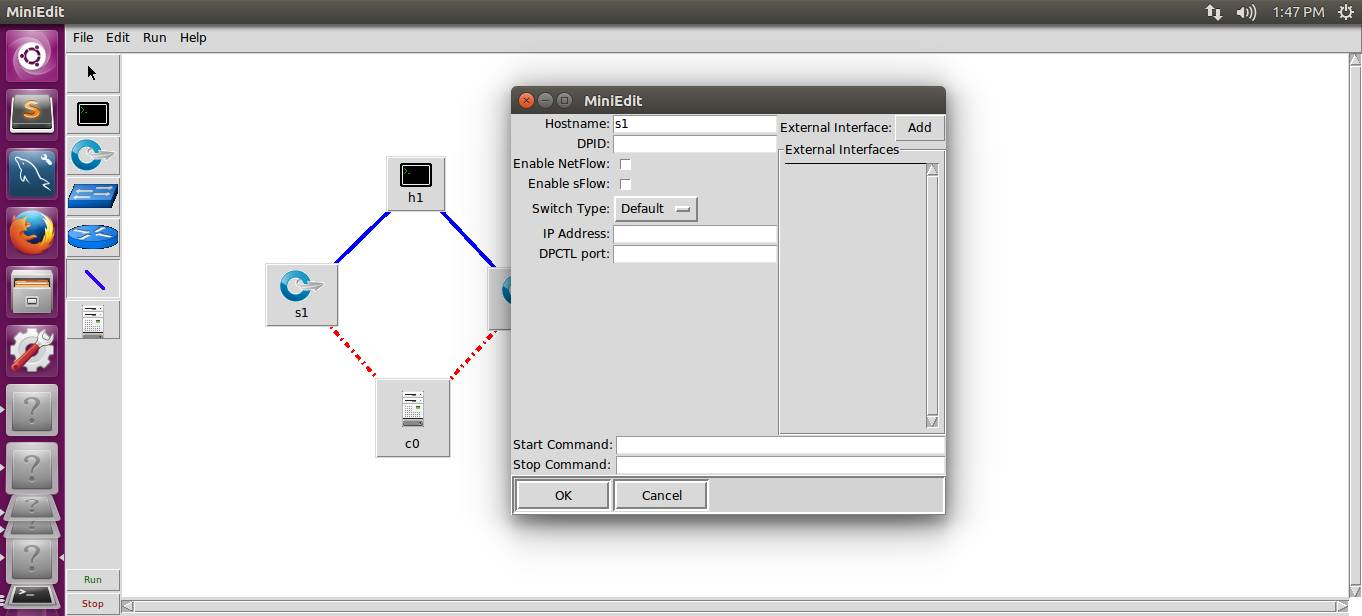
**Figure 22: Checking IP Address And Subnet Of NAT And Bridged**

1. Start miniedit.py and it will open a GUI to create the Mininet topology.
   1. Command: ***./miniedit.py***
2. Use the Miniedit GUI and make the following topology.



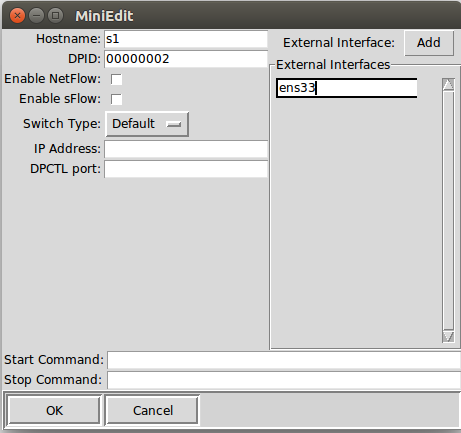
**Figure 23: Topology**

1. This will be the default topology used and will be configured for Secured-T system.
2. Right Click and hold on **s1** and open **Properties.** The following diagram will be shown.



**Figure 24: MiniEdit Properties**

1. Under **DPID**, type in 00000002 and press **Add** beside external interface. Under the new blank, type in **ens33**. This will bind **s1** to the network interface **ens33**, for reasons explained later.



**Figure 25: Add External Interface Eth0**

1. Repeat the steps (g) and (h) for **s2**. However, this time, set the **DPID** as 0000000A and the network interface as **ens34**.
2. Go to **File > Export Level 2 Script**. This will create the basic Mininet topology to configure. Save it under any name you want. For this guide, this file would be referred as **sectestTopo.py**.

#### **Editing the Mininet Topology file for Secured-T System**

1. Change the permission of **sectestTopo.py** to be able to be executed.
   1. Command: ***chmod +x sectestTopo.py***
2. Use an editor to configure the **sectestTopo.py** file.
   1. Command: ***gedit sectestTopo.py***
3. Multiple changes need to be made for the topology to work.
4. Import OVSSwitch from mininet.node

***BEFORE***



**Figure 26: Import OVSSwitch (Before)**

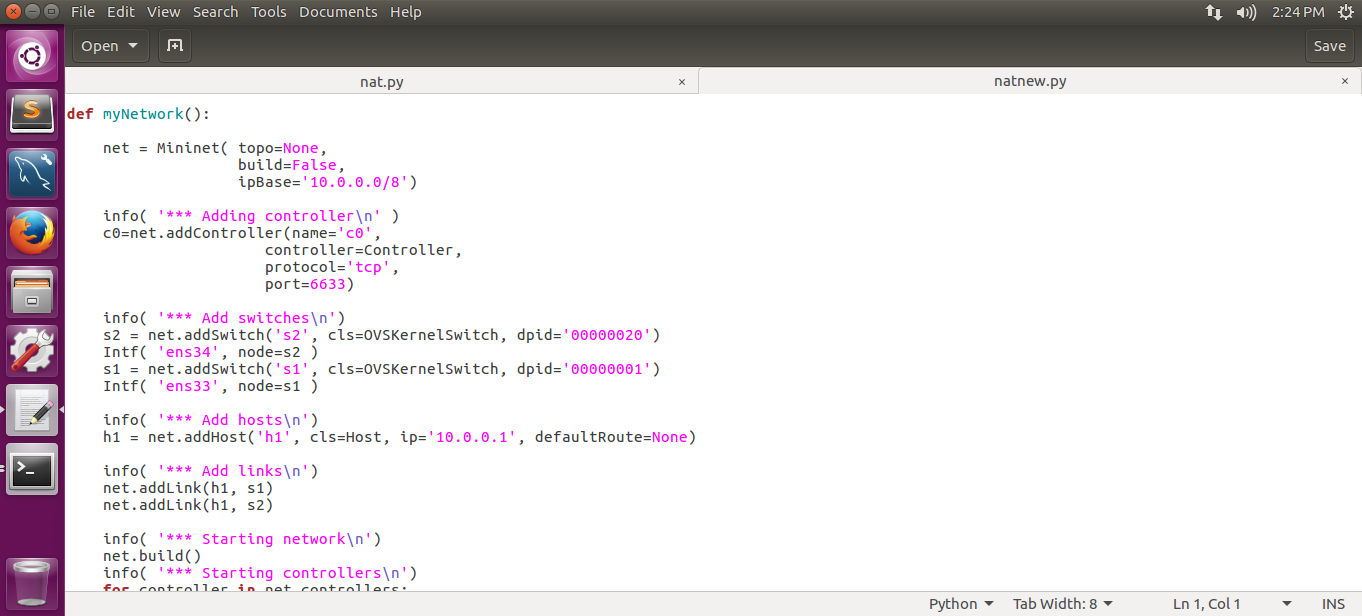
***AFTER***



**Figure 27: Import OVSSwitch (After)**

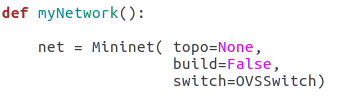
1. Define the switch so that Mininet will know that all switches in this topology is an Open Virtual Switch (OVS).
2. The ipBase is removed as there are 2 different subnets in the topology which could not be used as a base.

***BEFORE***



**Figure 28: Define Switch (Before)**

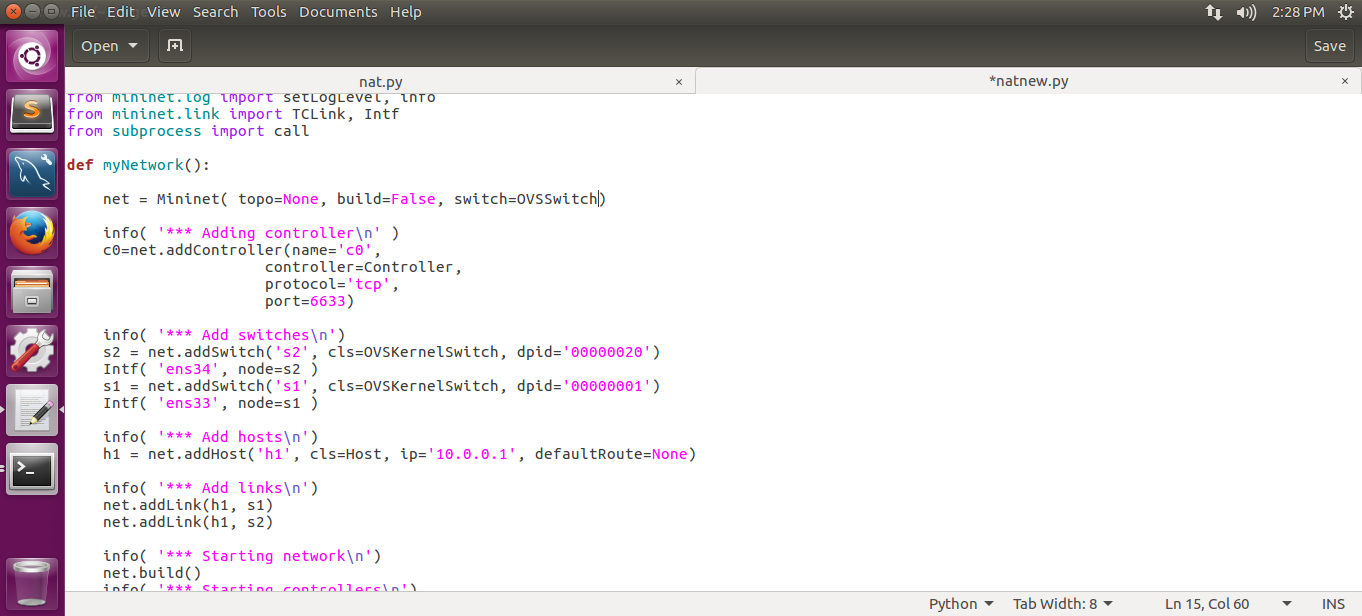
***AFTER***



**Figure 29: Define Switch (After)**

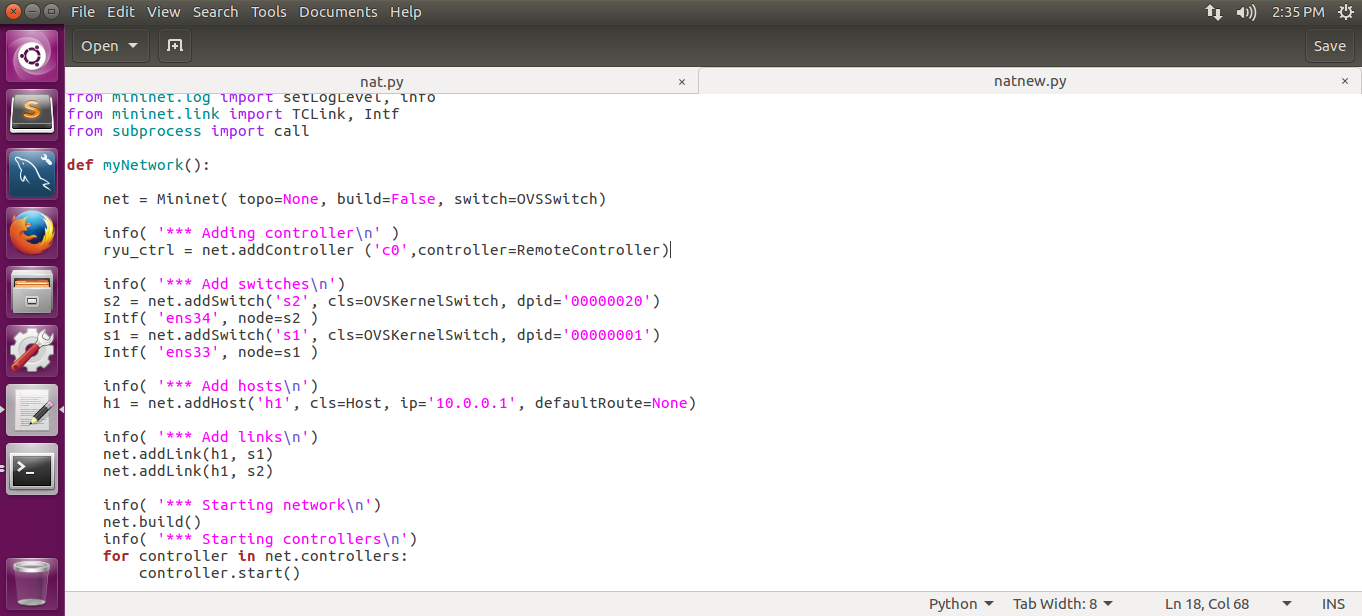
1. Instead of using the controller provided by Mininet, the controller would be an external controller from Ryu. For more information, refer to the Ryu Controller Guide.

***BEFORE***



**Figure 30: Change To Ryu Controller (Before)**

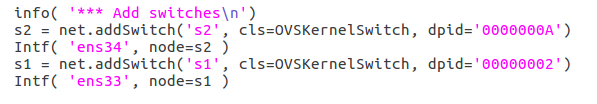
***AFTER***



**Figure 31: Change To Ryu Controller (After)**

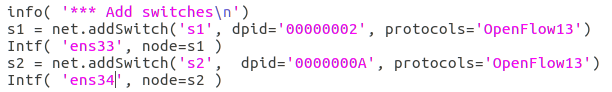
1. Swap the lines of **s1** and **s2** to avoid confusion since the switch is already defined as an OpenvSwitch, the cls is redundant.’
2. Include protocol openflow 13 at the end as well

***BEFORE***



**Figure 32: Edit The Switches (Before)**

***AFTER***



**Figure 33: Edit The Switches (After)**

1. Remove cls, ip and defaultRoute.
2. The characteristics of h1 will be defined in the later part of the program.

***BEFORE***



**Figure 34: Edit The Host (Before)**

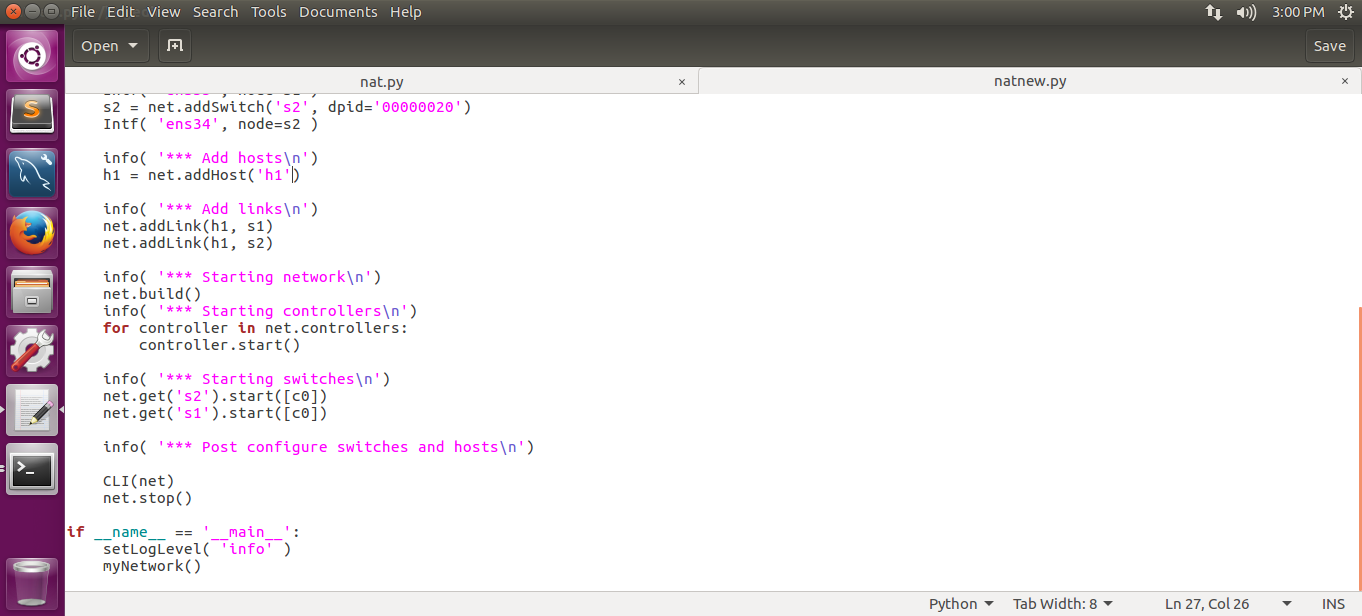
***AFTER***



**Figure 35: Edit The Host (After)**

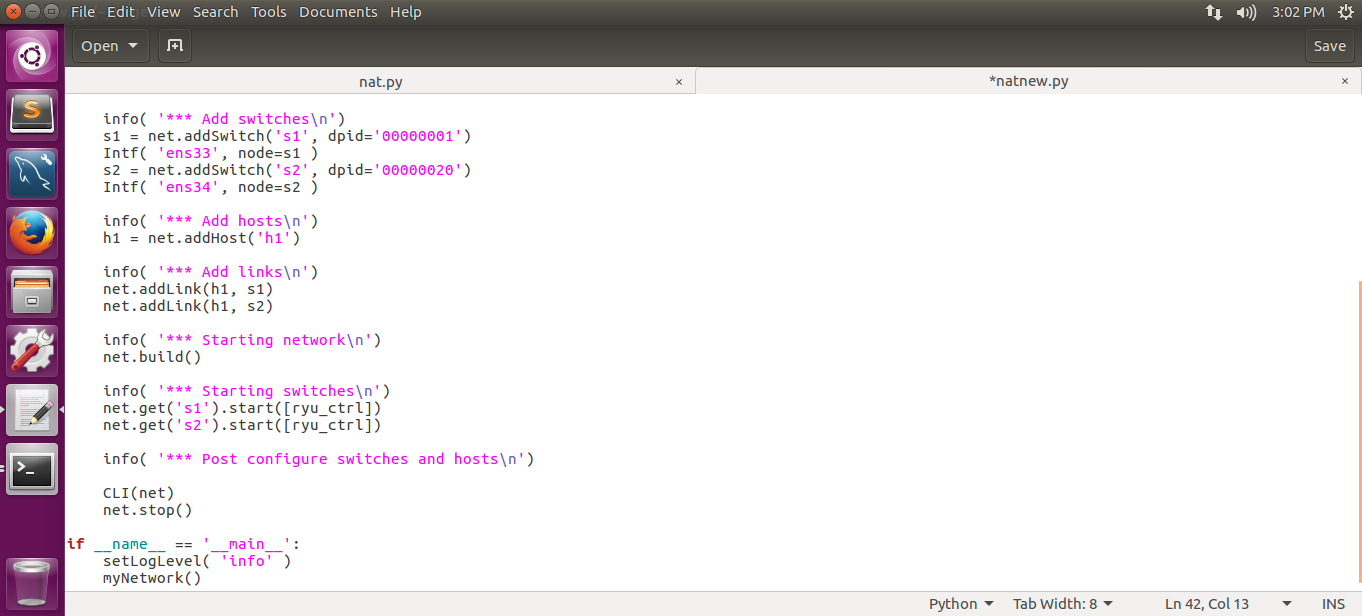
1. Since the controller used is an external controller, starting the controller is unnecessary. Since the controller is named **ryu\_ctrl** and not **c0**, adjust accordingly.

***BEFORE***



**Figure 36: Edit The Controllers (Before)**

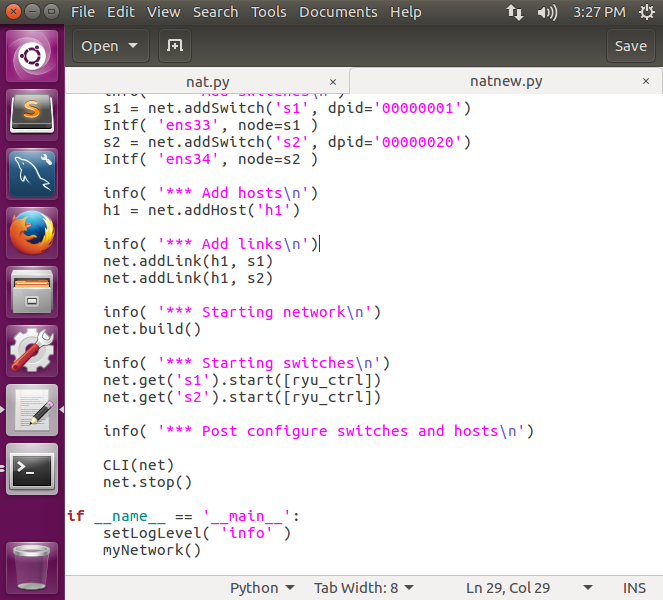
***AFTER***



**Figure 37: Edit The Controllers (After)**

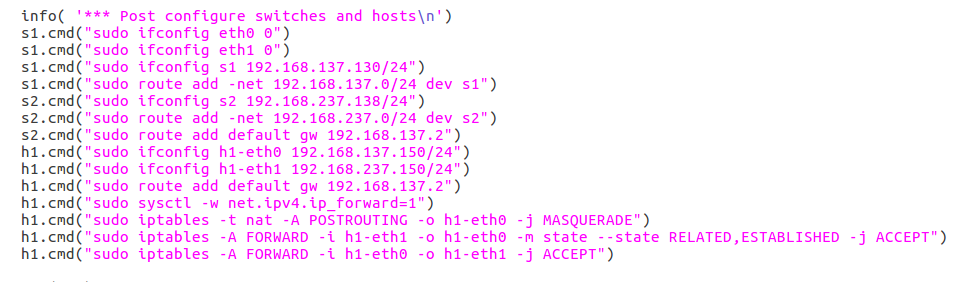
1. Take note of the IP address of the two network interfaces of the Virtual Machine. At this point onwards, you can always use eth0 and eth1 to refer to the virtual interfaces. They are set by the mininet, and the names are not affected by the real interface name.
   1. Remove the IP of both the Network Interfaces and give it to the two switches so that all the traffic that comes from the internet or the bridged side goes through the switches instead of the Network Interfaces.
   2. Set the default route of subnets so that Mininet knows which switch to send packets to if it belongs to a certain subnet.
   3. Set the network interfaces of **h1**.
   4. The default route for all packets that do not belong to either of the subnet is sent to the DHCP of the Virtual Machine.
   5. The last 4 lines is needed for the packets to flow between the two subnets. This allows any computer from Bridged subnet to access the Internet if it has the permission to do so.

***BEFORE***



**Figure 38: Configure Switches and Hosts (Before)**

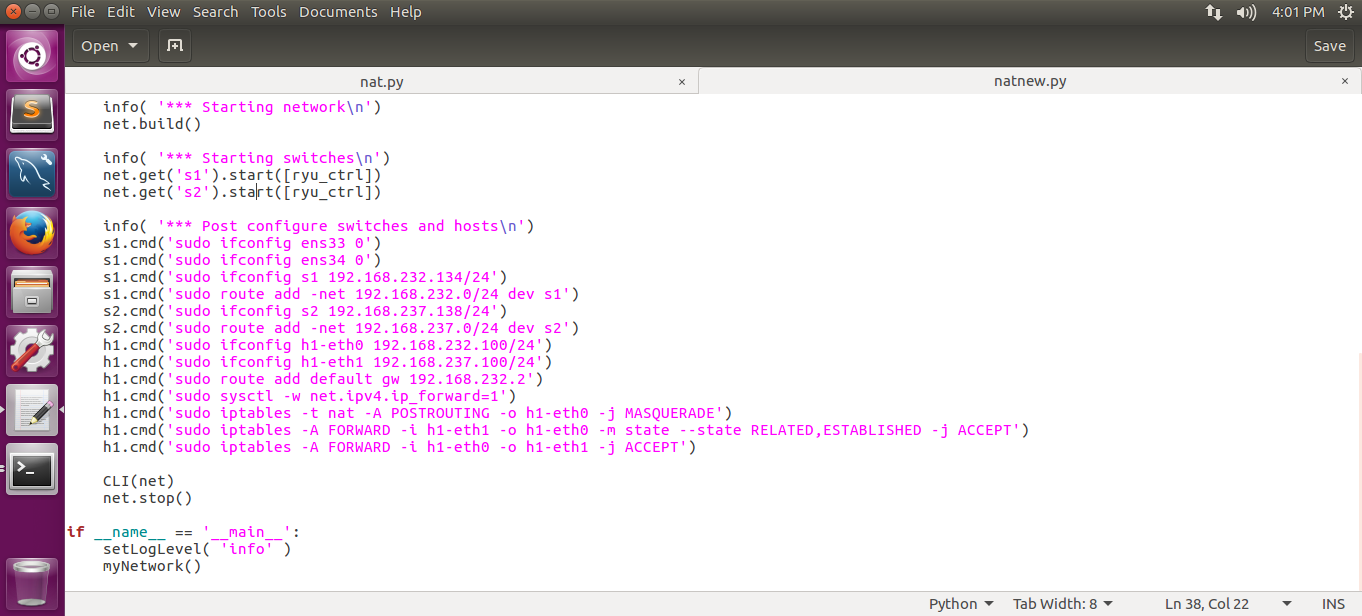
***AFTER***



**Figure 39: Configure Switches and Hosts (After)**

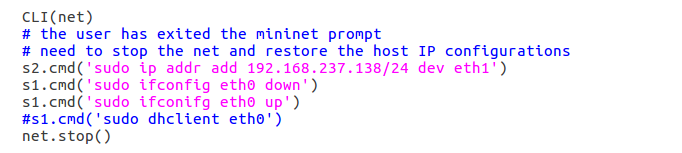
1. This modification will give the IP back to the network interfaces once the Mininet is closed.

***BEFORE***



**Figure 40: Closed Mininet Settings (Before)**

***AFTER***



**Figure 41: Closed Mininet Settings (After)**

1. Save the file with CTRL + S and close the editor.
2. The Mininet Topology is now in a deployable state for Secured-T. To execute the file that was set up, simply type in the following command.
   1. Command: ***./sectestTopo.py***

#### **RYU Controller Guide**

Overview:

The purpose of this guide is to guide the user to set up Ryu Controller. Refer to Install The Prerequisites Of Secured-T part (i) for the installation of the Ryu Controller before moving on.

#### **Download The Required Ryu Controller Scripts**

1. Download 3 scripts named ***sim\_switch.py***, ***sim\_switch\_rest.py*** and ***my\_db.py*** from Secured-T github and store under ***ryu/ryu/app***.
2. Download a script named ***my\_switch.py*** from Secured-T github and store under the home folder.
3. Download script ***simple\_switch.sh*** from Secured-T github and store under **ryu**. It will run the ryu controller.
4. Change the permission of ***sim\_switch\_rest.py***. This would be the file executed for the Ryu Controller.
   1. Command: ***sudo chmod +x sim\_switch\_rest.py***

#### **DNS Proxy Guide**

Overview:

The purpose of this guide is to setup a DNS proxy. This DNS is used to only allow access to domains that are in the whitelist domain table in the database. This guide would assume that part (j) of Install The Prerequisites Of Secured-T has been done.

1. Disable the Ubuntu build-in local dnsmasq feature (DNS cache).
   1. Ubuntu has start a local DNSCache service to speed up the DNS lookup, and it occupies the port 53. We need to disable it before we can host a new DNS service.

**BEFORE**



**Figure 42: Disable DNS Cache (Before)**

**AFTER**



**Figure 43: Disable DNS Cache (After)**

1. Restart the NetworkManager service.
   1. Command: ***sudo service network-manager restart***

#### **Creating and Configuring the DNS Proxy**

1. The deployable version of the file can be found in Secured-T github with the name ***dnsProxy.py***. As our system name and path may be different, you will need to configure the script to work. For this guide, the home directory is ***/home***.
2. Download script ***dnsProxy.py*** from Secured-T github and put it into the home folder.
3. Change the permissions of the file.
   1. Command: ***sudo chmod 777 dnsProxy.py***
4. Edit the file.
   1. Command: ***gedit dnsProxy.py***
5. Edit the following line to your system name and path.



**Figure 44: Edit System Name And Path**

1. Save and exit the file. The file is now in a deployable state.

**Tomcat Installation and Configuration**

1. Switch to root account.
   1. Command: ***sudo -s***

1. Update and upgrade apt-get.
   1. Command: ***apt-get update && apt-get upgrade***

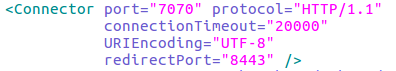
1. Switch to a normal user.

1. Install Apache Tomcat 8.
   1. Command: ***sudo apt-get install tomcat8***

* 1. For development:
     1. Command: **sudo apt-get install tomcat8-docs tomcat8-examples tomcat8-admin**
  2. For production:
     1. Command: **sudo apt-get install tomcat8-admin**

1. Start the Tomcat service.
   1. Command: **systemctl start tomcat8**
2. You can verify if Tomcat is running at the localhost by browsing to ***http://localhost:8080***.
3. In order to run both Tomcat and Ryu Controller, both must not be using the same port number. Since Ryu Controller is using port 8080, we will change Tomcat’s default port to something else.
   1. Command: ***sudo gedit /var/lib/tomcat8/conf/server.xml***

1. Change the port number to (e.g. 7070):



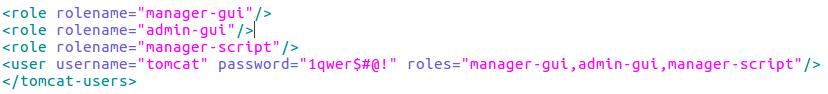
**Figure 45: Change Port Number Of Tomcat**

1. Restart Tomcat service and verify if Tomcat is running at the localhost
   1. Command: **systemctl restart tomcat8**

1. Browse to ***http://localhost:7070*.**

#### **Prepare a tomcat management account**

1. Update the following xml file.
   1. Command: **sudo gedit /var/lib/tomcat8/conf/tomcat-users.xml**
2. With the following tags (change your "username" and "password" accordingly), add this codes to the end of the file.



**Figure 46: Edit Tomcat Configuration File**

1. Restart Tomcat service and verify the manager page.
   1. Command: **systemctl restart tomcat8**

1. Browse to ***http://localhost:7070/manager.***

1. Enter your username and password set in the tomcat-users.xml file.

#### **Setting Up mod\_JK**

Overview:

The purpose of setting up mod\_JK is to bridge apache2 port 80 access to the web applications in tomcat8 so that it won’t be necessary to insert the port number in the url when going to the website.

1. Edit Tomcat server.xml file
   1. Command: ***sudo gedit /usr/share/tomcat8/conf/server.xml***

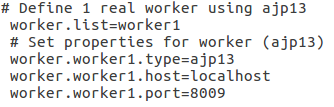
1. Uncomment the following line:



**Figure 47: Remove Connector**

1. Create a workers.properties file.
   1. Command: ***sudo gedit /etc/apache2/workers.properties***

1. Insert the following lines to define a real worker and set properties for it.

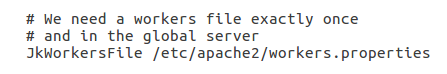


**Figure 48: Define A Real Worker With Properties**

1. Install mod-jk.
   1. Command: ***sudo apt-get install libapache2-mod-jk***

1. Edit jk.conf file.
   1. Command: ***sudo gedit /etc/apache2/mods-available/jk.conf***

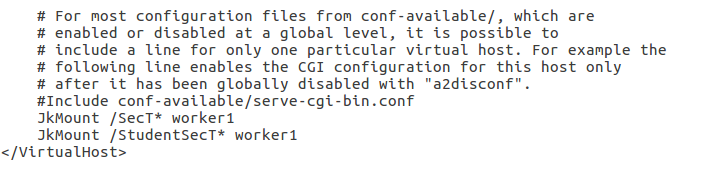
1. Change the JkWorkersFile properties in jk.conf.



**Figure 49: JkWorkersFile Properties**

1. Finally, to configure the URL of Apache2 to pass through the Tomcat8.
   1. Command:***sudo gedit /etc/apache2/sites-enabled/000-default.conf***

1. Add the following lines in 000-default.conf.



**Figure 50: Configure Apache2 Web Server**

* 1. The trailing \* is a wild card to enable all sub folder access.
  2. You may add in multiple JkMount statement if needed.

1. Restart the servers.
   1. Command: ***sudo service tomcat8 restart***

***sudo service apache2 restart***

1. Check the functionality by browsing to ***http://localhost/SecT*** and ***http://localhost/StudentSecT***

#### **MySQL Workbench Installation guide**

Overview:

The purpose of this guide is to install and set up MySQL Workbench for the Secured-T system to store all the details.

1. First, launch a terminal, type the following command to install MySQL Workbench.
   1. Command: ***sudo apt-get install mysql-workbench***

***sudo apt-get install mysql-server\****

***sudo mysql\_secure\_installation***

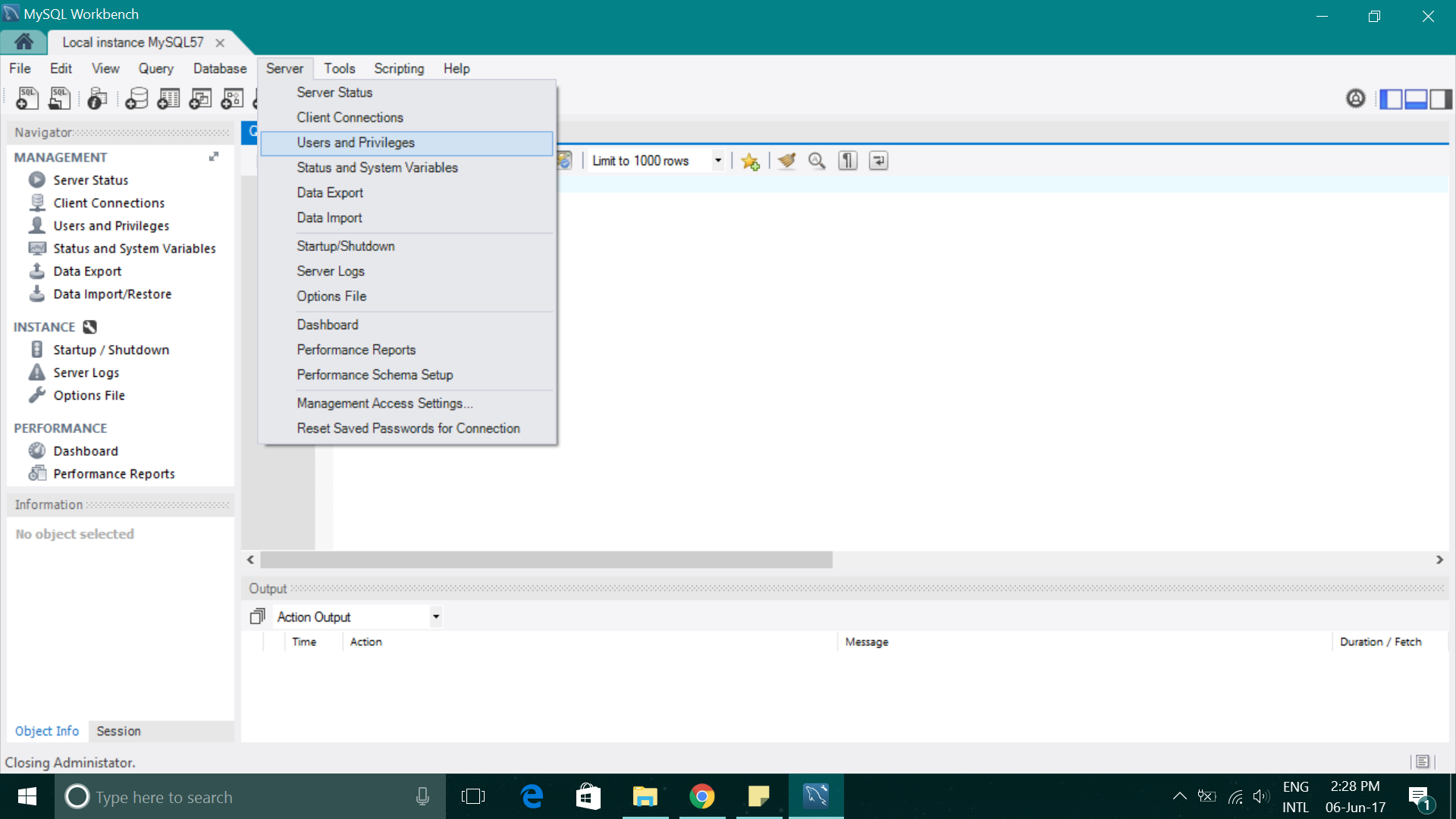
***sudo apt-get install mysql-client***

* 1. Remember the password you set for the root account as it will be needed later

1. Once the installation is done, start/enable the service.
   1. Command: **sudo service mysql start**

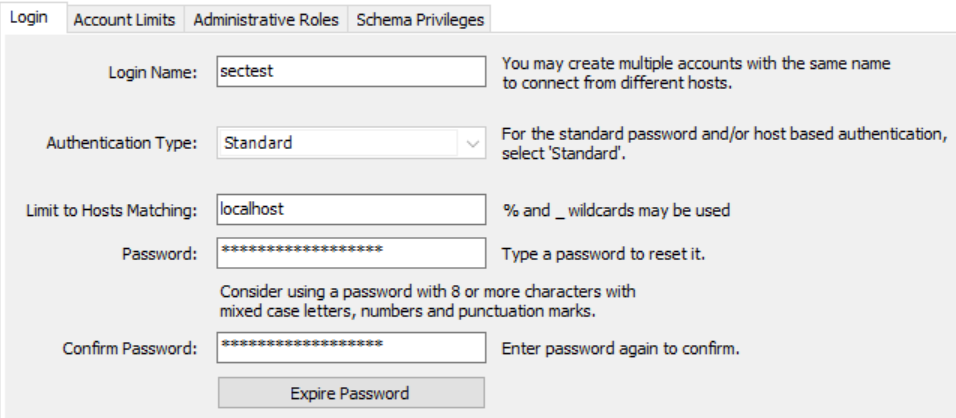
#### **Adding Connection On MySQL Workbench**

1. After installation, there will be a default root user account. Connect to the root account with the password set during installation.
2. Go to Server > Users and Privileges



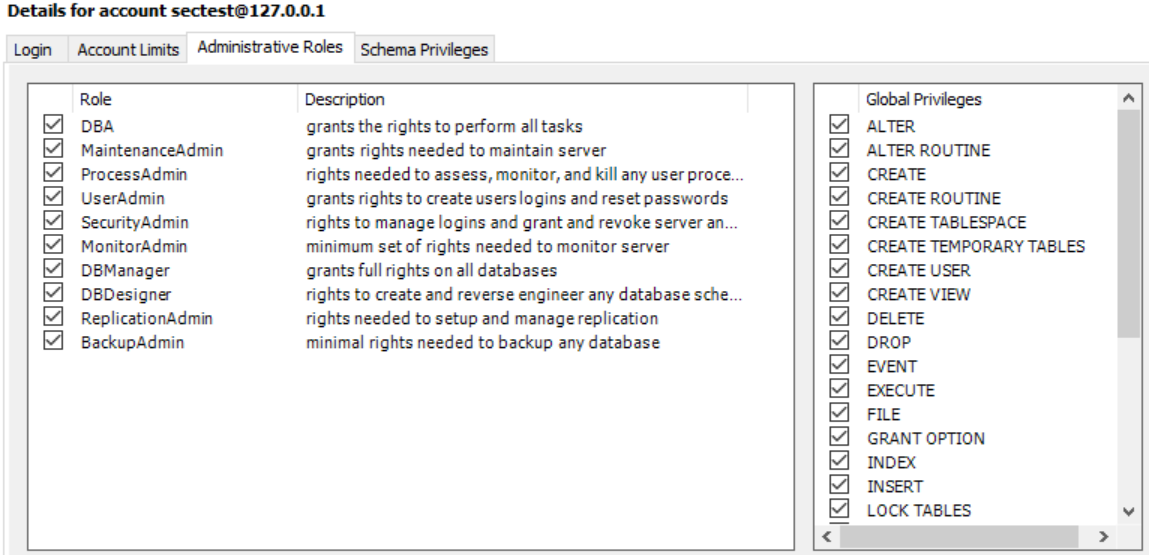
**Figure 51: Users and Privileges**

1. Click Add account and enter the details as follows:



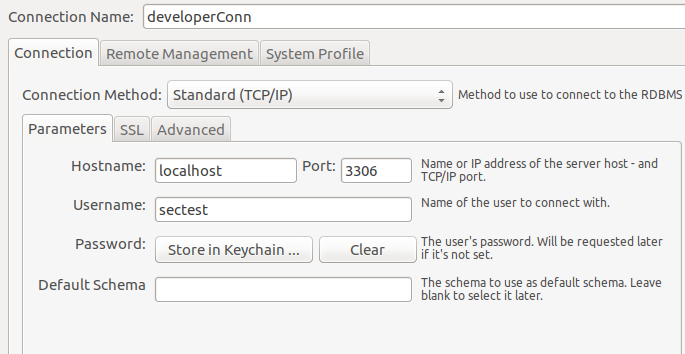
**Figure 52: Database Account Credentials**

1. The password will change it’s length automatically so no need to worry about it.
2. Then, go to the Administrative Roles and tick the role DBA.



**Figure 53: Grant Rights To Perform All Tasks**

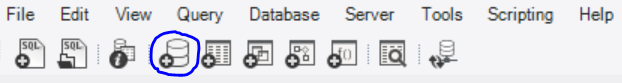
1. Click Apply.
2. Then, go back to the Home Page of MySQL Workbench.
3. Add a connection and enter the details as follows:



**Figure 54: Add Database Connection**

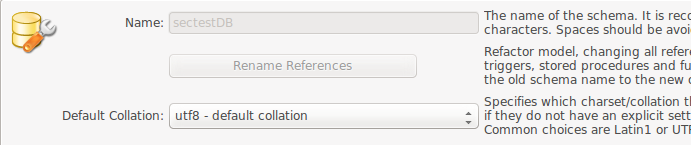
1. The username must be the same as the account Login Name. Leave the default schema field.

1. Click OK.
2. Connect to the new connection using the password set when adding the account.
3. Next, create a new schema by clicking the icon:



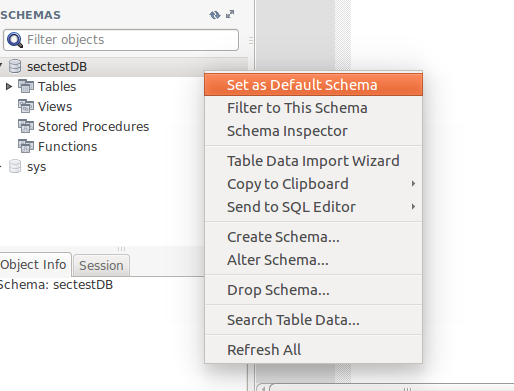
**Figure 55: Add Schema**

1. Enter the details as follows:



**Figure 56: Create Schema**

1. Next, set the schema created to the default schema.

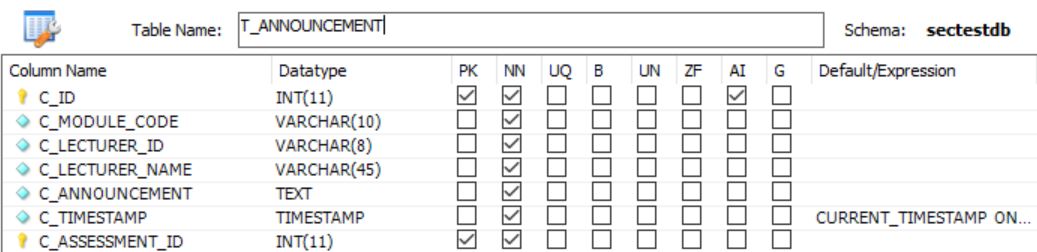


**Figure 57: Set Schema As Default Schema**

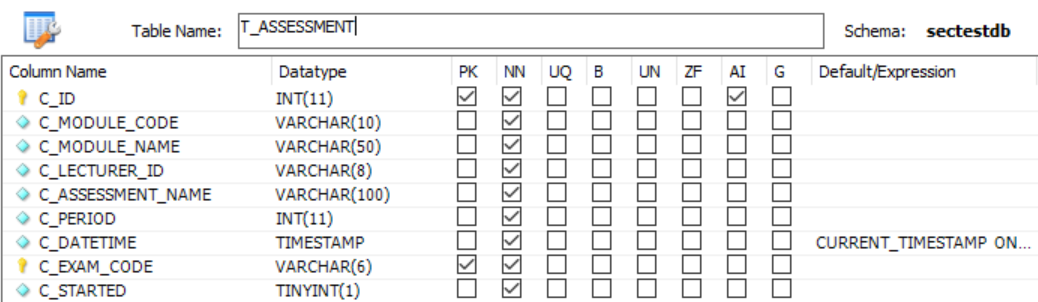
1. Now, you should have a custom user account with connection and a schema within it.

#### **Create Database Tables**

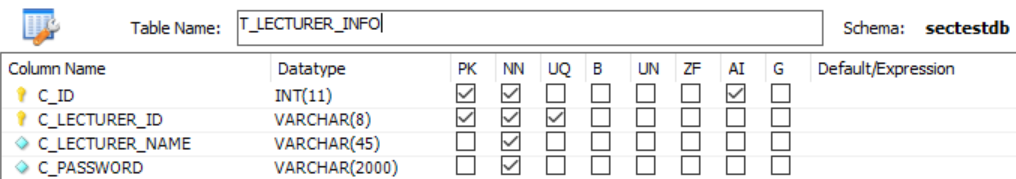
1. Create tables as shown in the figures below.



**Figure 58: T\_ANNOUNCEMENT**



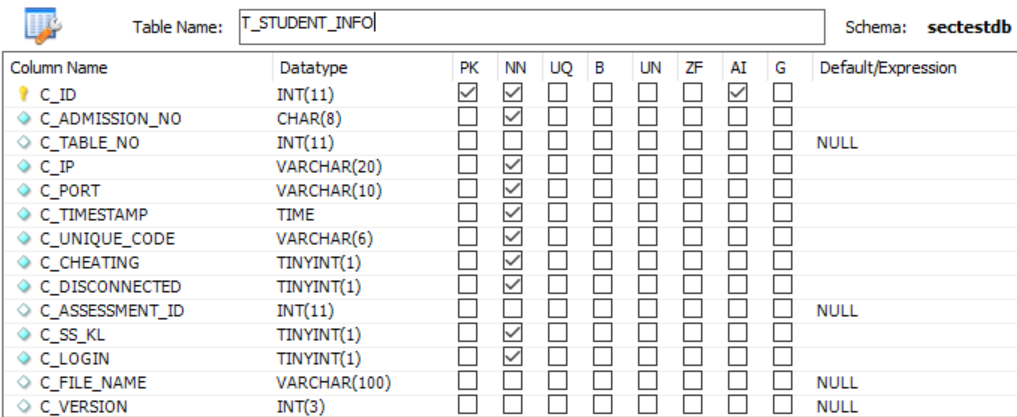
**Figure 59: T\_ASSESSMENT**



**Figure 60: T\_LECTURER\_INFO**



**Figure 61: T\_MODULE**



**Figure 62: T\_STUDENT\_INFO**



**Figure 63: T\_WHITELIST\_DOMAIN**

#### **Deployment guide**

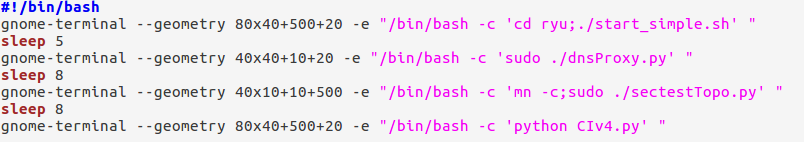
Overview:

The purpose of this guide is start up the system for use.

1. After completing all of the installation and configuration stated, do a system reboot
   1. Command: ***sudo reboot***
2. There is a batch attached called ***routeadd.bat****.* Download on the base machine and run it on administrative mode.

#### **Create a shell script**

1. There is a script called ***startAll.sh*** in Secured-T github. Download the script and place it in your home directory.
2. startAll.sh is a script that will open up terminals and run all of the required programs separately.

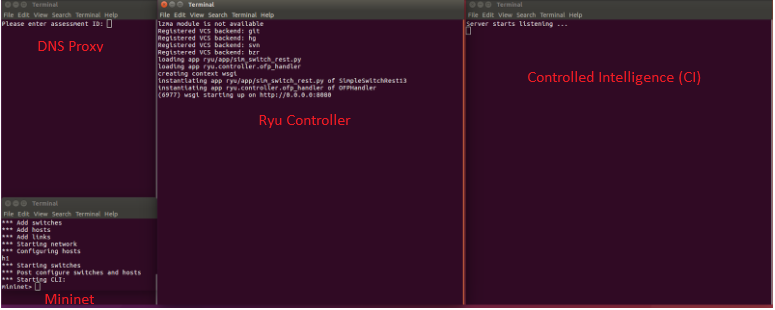


**Figure 64: Script To Run All Four Programs**

1. Now you should have four terminals being displayed on your screen as shown in the table below:

**Table 1: Software/Program And The Corresponding Filename**

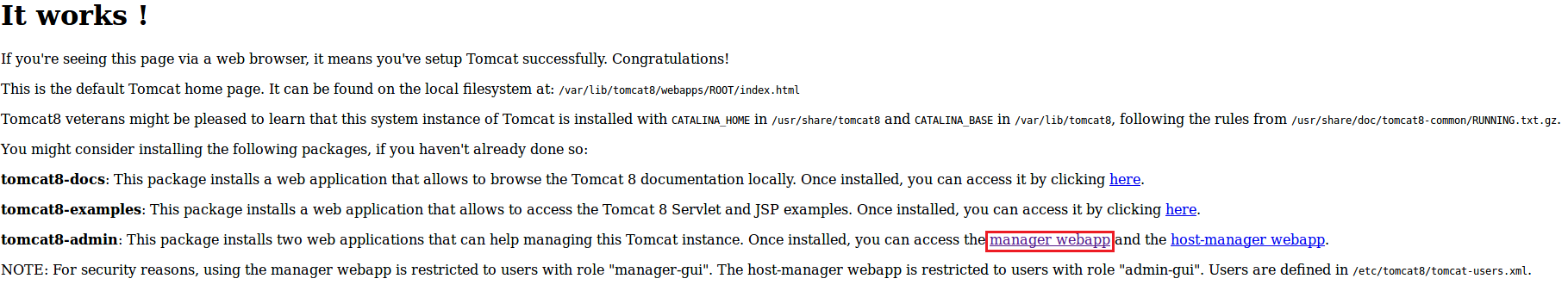
|  |  |
| --- | --- |
| **Software/Program** | **Filename** |
| Ryu Controller | start\_simple.sh |
| Mininet | sectestTopo.py |
| DNS Proxy Server | dnsProxy.py |
| Controlled Intelligence | CIv4.py |



**Figure 65: All Four programs running**

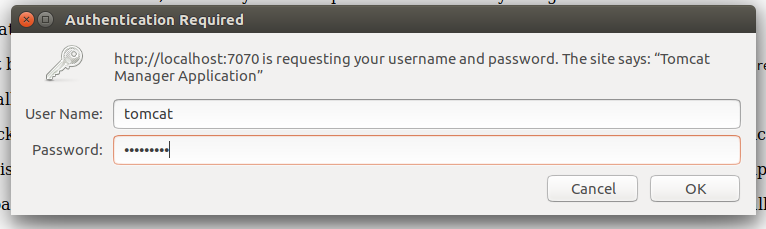
#### **Deploying websites**

1. Download **SectT.war** and **StudentSecT.war** from Secured-T github. These files still need to be deployed on your tomcat. Store the files in the home directory.
2. Go to [***http://localhost***](http://localhost:7070)***:7070***
3. Click manager webapp



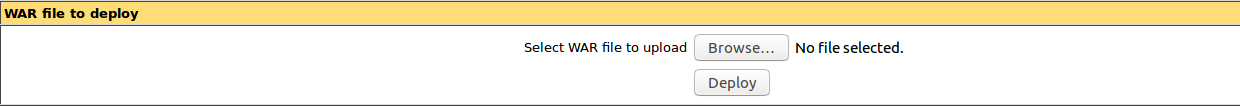
**Figure 66: Tomcat8 Default Page**

1. Key in the username and password of your tomcat account created previously.



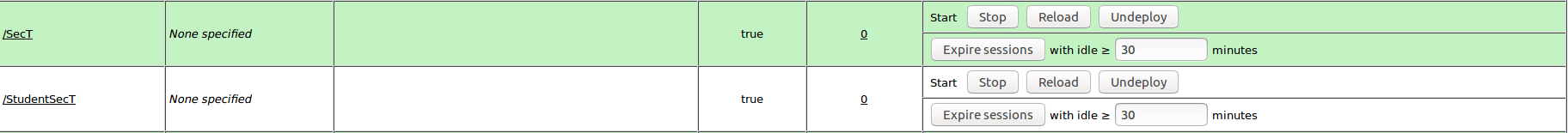
**Figure 67: Tomcat8 Login**

1. You will be directed to tomcat web application manager page, scroll down to ‘Deploy’ and then to ‘War file to deploy’ under it.



**Figure 68: Deploy War File**

1. Select your war file and click deploy.
2. Repeat for the second war file.
3. Now your files will be under Applications and your web applications are up and running



**Figure 69: Webapp Manager**

**The End**