NETWORK VIRTUALIZATION

MSIS603N-711

Project: Develop an Introduction to Python Programming for Network Engineers Lab

Submitted by: Meghana Meka

Cwid: 20157396

I. INTRODUCTION

This project introduces fundamental concepts of network automation using Python. The main objective is to connect to a simulated Cisco router via SSH and automate the retrieval of its running configuration using a Python script. This helps demonstrate how Python can be used to manage and automate network devices in real-world scenarios.

To simulate the network environment, Cisco Packet Tracer was used to configure a Cisco router with SSH access, and Ubuntu Linux was used as the platform to run Python scripts. The Python script leverages Netmiko, a Python library built to simplify SSH communication with network devices.

NETMIKO: Netmiko is an open-source Python library developed to interact with network devices via SSH. It supports many vendors (like Cisco, Juniper, Arista, etc.) and automates command execution on routers and switches. In this project, Netmiko is used to send the show running-config command to a Cisco router and save its output into a local file.

Cisco Packet Tracer: Cisco Packet Tracer is a network simulation tool designed by Cisco. It allows users to create virtual network topologies, configure routers and switches, and test connectivity between devices. It's widely used in networking education and certification labs. In this project, it was used to simulate a Cisco router and verify SSH connectivity from a virtual PC.

II Environment Setup

Router Setup in Cisco Packet Tracer:

- A Cisco router was configured with SSH and an IP address on interface G0/0.
- A virtual PC was used to verify SSH connectivity to the router using the command-line terminal.

Ubuntu Virtual Machine (in VMware):

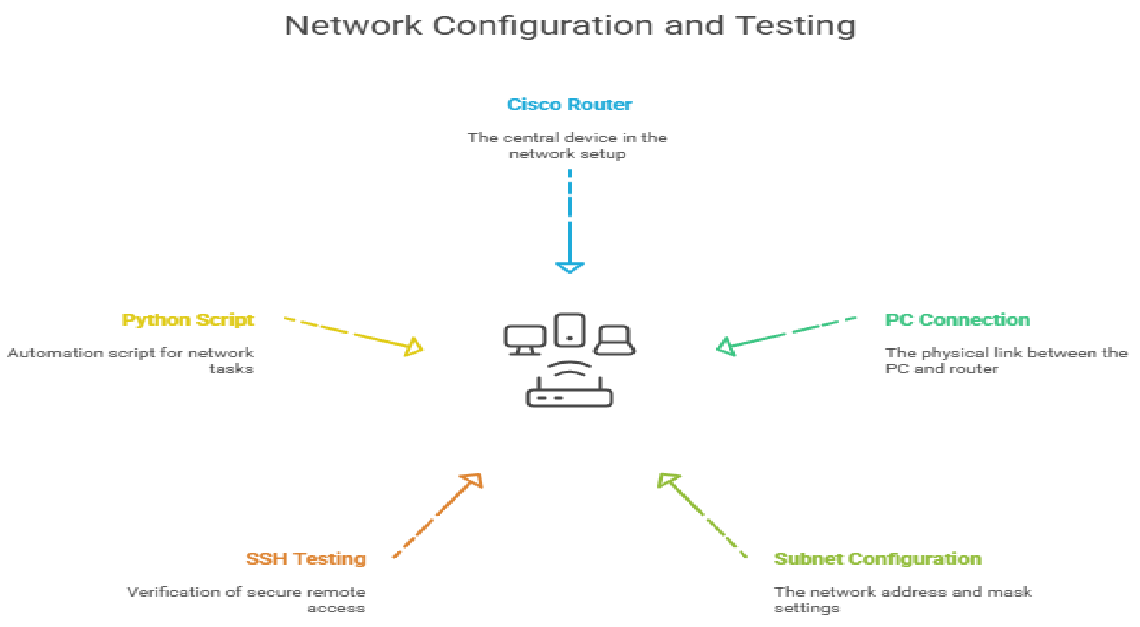
- Used to write and run Python scripts with Netmiko.
- Python and required libraries were installed, but due to network isolation between the VM and Packet Tracer, SSH automation couldn't be executed from Python to the router.

Key Syntax

Command	**Description**
sudo apt update	Updates package lists
sudo apt upgrade -y	Upgrades installed packages
sudo apt install python3-pip	Installs pip for Python 3
sudo apt install curl	Installs curl to fetch remote files
curl -o get-pip.py https://bootstrap.pypa.io/get-pip.py	Downloads pip installer
wget https://bootstrap.pypa.io/get-pip.py	Alternative way to download pip installer

sudo python3 get-pip.py	Installs pip using downloaded script
sudo apt install python3-venv -y	Installs Python virtual environment support
python3 -m venv netlab-env	Creates a new virtual environment
source netlab-env/bin/activate	Activates the virtual environment
pip install netmiko textfsm	Installs automation libraries
nano backup_config.py	Creates or edits the Python script

III Topology

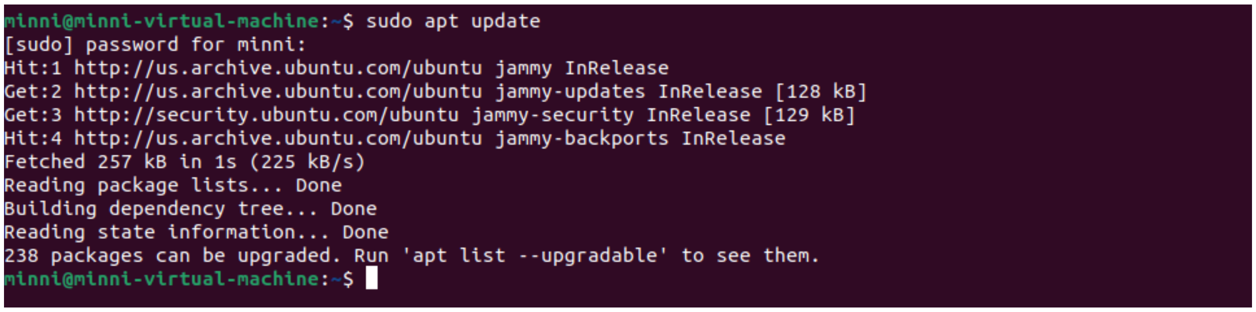

Network Configuration and Testing
Cisco Router
The central device in the network setup
Python Script
Automation script for network tasks
PC Connection
The physical link between the PC and router
SSH Testing
Verification of secure remote access
Subnet Configuration
The network address and mask settings

<u>Step-I Setup the Environment:</u>

I have started the procedure by setting up the lab environment, for more dependencies and more accuracy, I had installed ubuntu 22.0.4 for all the successful procedure of the network automation.

- Initially, I started the procedure by updating the ubuntu rules and configurations by using the below command

Command: sudo apt update

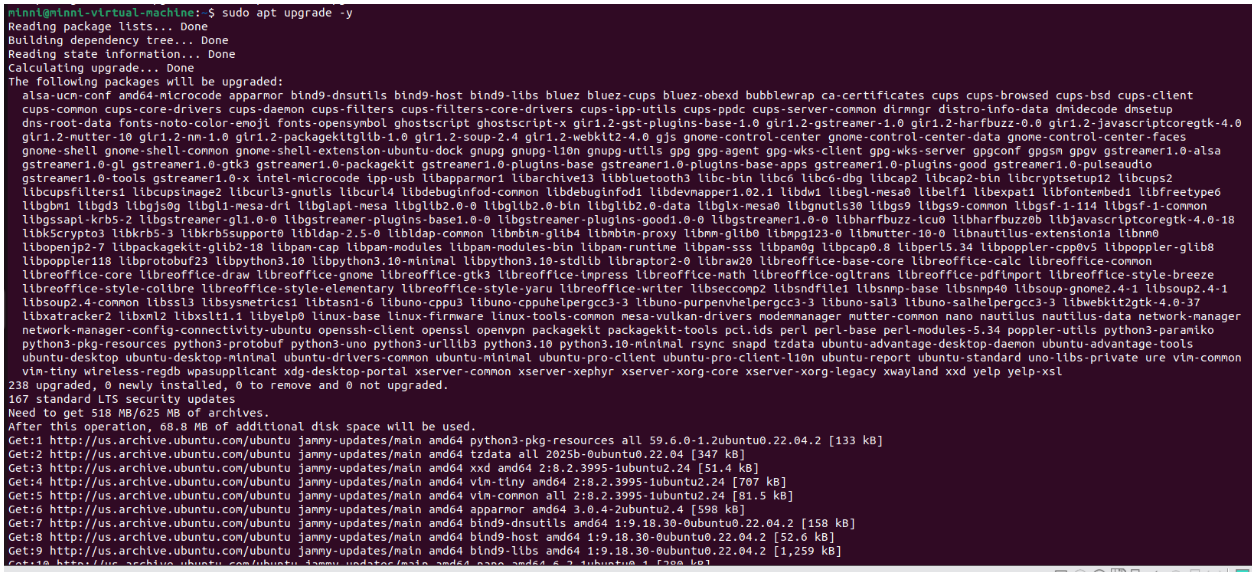


```
minni@minni-virtual-machine:~$ sudo apt update
[sudo] password for minni:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 257 kB in 1s (225 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
238 packages can be upgraded. Run 'apt list --upgradable' to see them.
minni@minni-virtual-machine:~$
```

This capture represents the update of ubuntu

- The next step I had followed is to upgrade all the packages already existing in ubuntu by using the below command

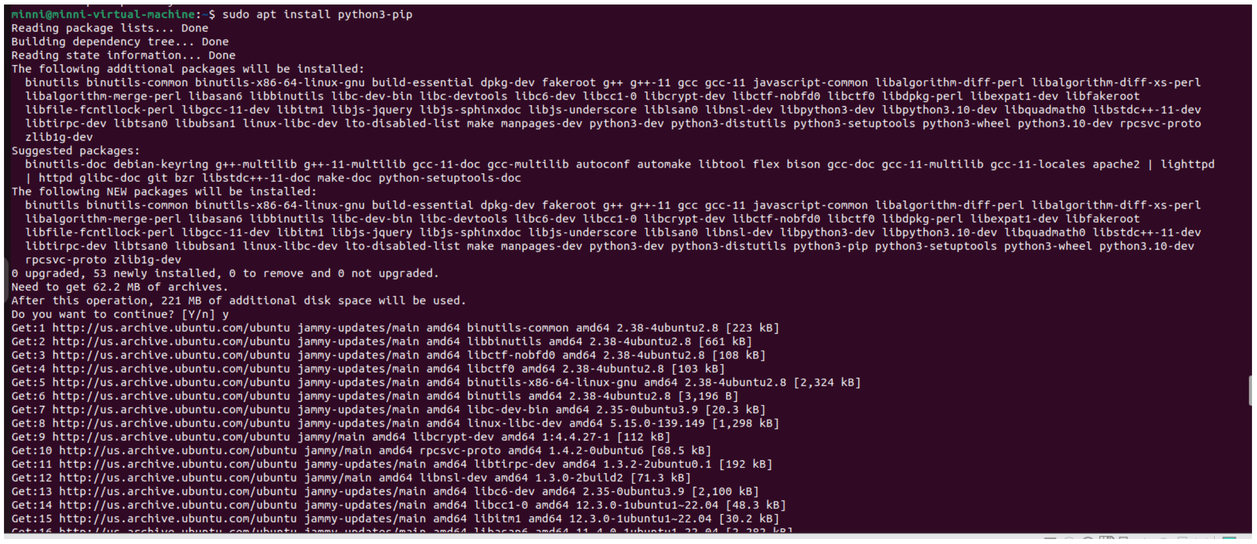
Command: sudo apt upgrade -y



```
minni@minni-virtual-machine:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  alsa-ucm-conf amd64-microcode apparmor bind9-dnsutils bind9-host bind9-libs bluez bluez-cups bluez-obexd bubblewrap ca-certificates cups cups-browsed cups-bsd cups-client
  cups-common cups-core-drivers cups-daemon cups-filters cups-filters-core-drivers cups-ipp-utils cups-ppdc cups-server-common dirmngr distro-info-data dmidecode dmsetup
  dns-root-data fonts-noto-color-emoji fonts-opensymbol ghostscript ghostscript-x gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-harfbuzz-0.0 gir1.2-javascriptcoregtk-4.0
  gir1.2-mutter-10 gir1.2-nm-1.0 gir1.2-packagekitglib-1.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0 gjs gnome-control-center gnome-control-center-data gnome-control-center-faces
  gnome-shell gnome-shell-common gnome-shell-extension-ubuntu-dock gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv gstreamer1.0-alsa
  gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-packagekit gstreamer1.0-plugins-base gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-good gstreamer1.0-pulseaudio
  gstreamer1.0-tools gstreamer1.0-x intel-microcode ipp-usb libapparmor1 libarchive13 libbluetooth3 libc-bin libc6 libc6-dbg libcap2 libcap2-bin libcryptsetup12 libcups2
  libcupsfilters1 libcupsimage2 libcurl3-gnutls libcurl4 libdebuginfod-common libdebuginfod1 libdevmapper1.02.1 libdw1 libegl-mesa0 libelf1 libexpat1 libfontembed1 libfreetype6
  libgbm1 libgd3 libgjs0g libgl1-mesa-dri libglapi-mesa libglib2.0-0 libglib2.0-bin libglib2.0-data libglx-mesa0 libgnutls30 libgs9 libgs9-common libgsf-1-114 libgsf-1-common
  libgssapi-krb5-2 libgstreamer-gl1.0-0 libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0 libgstreamer1.0-0 libharfbuzz-icu0 libharfbuzz0b libjavascriptcoregtk-4.0-18
  libk5crypto3 libkrb5-3 libkrb5support0 libldap-2.5-0 libldap-common libmbim-glib4 libmbim-proxy libmm-glib0 libmpg123-0 libmutter-10-0 libnautilus-extension1a libnm0
  libopenjp2-7 libpackagekit-glib2-18 libpam-cap libpam-modules libpam-modules-bin libpam-runtime libpam-sss libpam0g libpcap0.8 libperl5.34 libpoppler-cpp0v5 libpoppler-glib8
  libpoppler118 libprotobuf23 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libraptor2-0 libraw20 libreoffice-base-core libreoffice-calc libreoffice-common
  libreoffice-core libreoffice-draw libreoffice-gnome libreoffice-gtk3 libreoffice-impress libreoffice-math libreoffice-ogltrans libreoffice-pdfimport libreoffice-style-breeze
  libreoffice-style-colibre libreoffice-style-elementary libreoffice-style-yaru libreoffice-writer libseccomp2 libsndfile1 libsnmp-base libsnmp40 libsoup-gnome2.4-1 libsoup2.4-1
  libsoup2.4-common libssl3 libsysmetrics1 libtasn1-6 libuno-cppu3 libuno-cppuhelpergcc3-3 libuno-purpenvhelpergcc3-3 libuno-sal3 libuno-salhelpergcc3-3 libwebkit2gtk-4.0-37
  libxatracker2 libxml2 libxslt1.1 libyelp0 linux-base linux-firmware linux-tools-common mesa-vulkan-drivers modemmanager mutter-common nano nautilus nautilus-data network-manager
  network-manager-config-connectivity-ubuntu openssh-client openssl openvpn packagekit packagekit-tools pci.ids perl perl-base perl-modules-5.34 poppler-utils python3-paramiko
  python3-pkg-resources python3-protobuf python3-uno python3-urllib3 python3.10 python3.10-minimal rsync snapd tzdata ubuntu-advantage-desktop-daemon ubuntu-advantage-tools
  ubuntu-desktop ubuntu-desktop-minimal ubuntu-drivers-common ubuntu-minimal ubuntu-pro-client ubuntu-pro-client-l10n ubuntu-report ubuntu-standard uno-libs-private ure vim-common
  vim-tiny wireless-regdb wpasupplicant xdg-desktop-portal xserver-common xserver-xephyr xserver-xorg-core xserver-xorg-legacy xwayland xxd yelp yelp-xsl
238 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
167 standard LTS security updates
Need to get 518 MB/625 MB of archives.
After this operation, 68.8 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-pkg-resources all 59.6.0-1.2ubuntu0.22.04.2 [133 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 tzdata all 2025b-0ubuntu0.22.04 [347 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 xxd amd64 2:8.2.3995-1ubuntu2.24 [51.4 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 vim-tiny amd64 2:8.2.3995-1ubuntu2.24 [707 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 vim-common all 2:8.2.3995-1ubuntu2.24 [81.5 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apparmor amd64 3.0.4-2ubuntu2.4 [598 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 bind9-dnsutils amd64 1:9.18.30-0ubuntu0.22.04.2 [158 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 bind9-host amd64 1:9.18.30-0ubuntu0.22.04.2 [52.6 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 bind9-libs amd64 1:9.18.30-0ubuntu0.22.04.2 [1,259 kB]
```

This capture represents the upgrade

- Now, we are processing the further important steps that mainly involve with the lab.
- The first Import step is installation of python3 and python 3 pip which was done by using the following command.
- **Command: sudo apt install python3-pip**
- This command was executed in the Ubuntu VM to enable the installation of Python libraries such as netmiko, which was essential for writing the automation script that connects to the Cisco router over SSH.

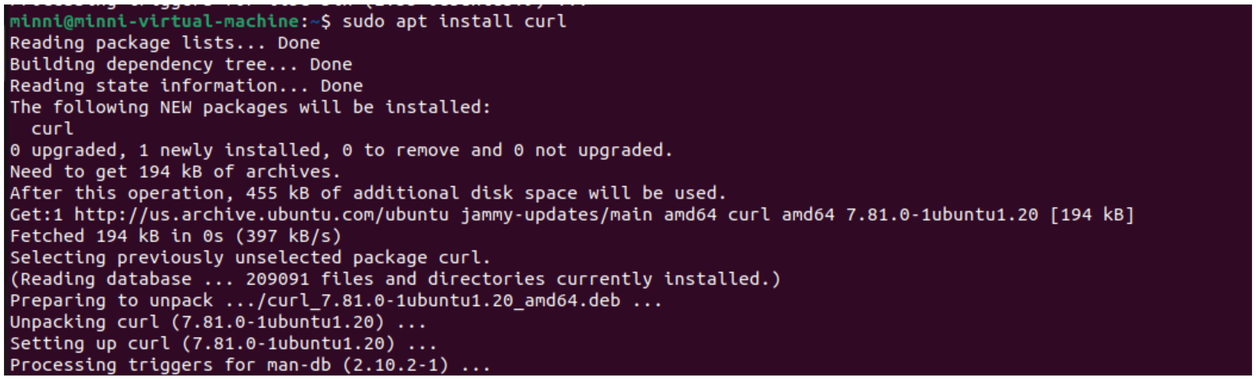


```
minni@minni-virtual-machine:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev
  libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-distutils python3-setuptools python3-wheel python3.10-dev rpcsvc-proto
  zlib1g-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-11-multilib gcc-11-locales apache2 | lighttpd
  | httpd glibc-doc git bzr libstdc++-11-doc make-doc python-setuptools-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev
  libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-distutils python3-pip python3-setuptools python3-wheel python3.10-dev
  rpcsvc-proto zlib1g-dev
0 upgraded, 53 newly installed, 0 to remove and 0 not upgraded.
Need to get 62.2 MB of archives.
After this operation, 221 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils-common amd64 2.38-4ubuntu2.8 [223 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libbinutils amd64 2.38-4ubuntu2.8 [661 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libctf-nobfd0 amd64 2.38-4ubuntu2.8 [108 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libctf0 amd64 2.38-4ubuntu2.8 [103 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils-x86-64-linux-gnu amd64 2.38-4ubuntu2.8 [2,324 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 binutils amd64 2.38-4ubuntu2.8 [3,196 B]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc-dev-bin amd64 2.35-0ubuntu3.9 [20.3 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 linux-libc-dev amd64 5.15.0-139.149 [1,298 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libcrypt-dev amd64 1:4.4.27-1 [112 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 rpcsvc-proto amd64 1.4.2-0ubuntu6 [68.5 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtirpc-dev amd64 1.3.2-2ubuntu0.1 [192 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libnsl-dev amd64 1.3.0-2build2 [71.3 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6-dev amd64 2.35-0ubuntu3.9 [2,100 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcc1-0 amd64 12.3.0-1ubuntu1~22.04 [48.3 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libitm1 amd64 12.3.0-1ubuntu1~22.04 [30.2 kB]
```

This capture represents the installation of python3-pip

- Next step that I had followed is the installation of curl package by using the following command.

- **Command: sudo apt install curl**
- The above command was used to download external files or scripts directly from a URL (such as Python's get-pip.py script or other resources) when other package installation methods were not available or failed.



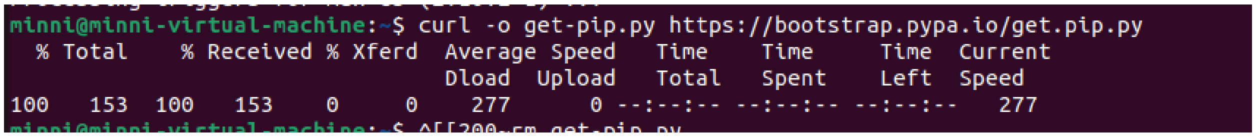
```
minni@minni-virtual-machine:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.20 [194 kB]
Fetched 194 kB in 0s (397 kB/s)
Selecting previously unselected package curl.
(Reading database ... 209091 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.20_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.20) ...
Setting up curl (7.81.0-1ubuntu1.20) ...
Processing triggers for man-db (2.10.2-1) ...
```

This capture represents the installation of curl

- Now, we are going to download the bootstrap package which is an external package, for this step we have installed the curl package and then using the following command.

Command: curl -o get-pip.py https://bootstrap.pypa.io/get.pip.py

- This command was used in the Ubuntu environment when the default apt method to install pip3 did not work. The script get-pip.py provides an alternative way to install pip using Python directly.



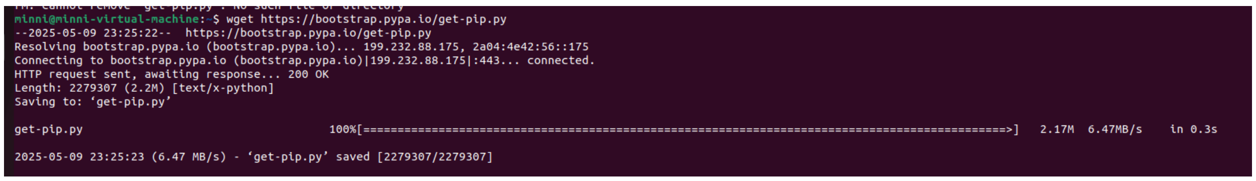
```
minni@minni-virtual-machine:~$ curl -o get-pip.py https://bootstrap.pypa.io/get.pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   153  100   153    0     0    277      0 --:--:-- --:--:-- --:--:--   277
```

This capture represents the manual installation of get-pip

- The next step, that I had done is the installation of get-pip.py by using the following command.

Command: Wget https://bootstrap.pypa.io/get-pip.py

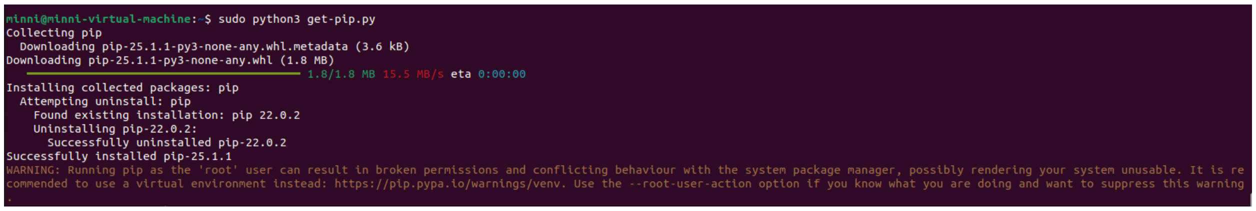
- This command was used to download the **get-pip.py** script when curl was unavailable or when a more straightforward method was preferred. The downloaded script was then executed to install the pip package manager manually for Python 3.



```
minni@minni-virtual-machine:~$ wget https://bootstrap.pypa.io/get-pip.py
--2025-05-09 23:25:22--  https://bootstrap.pypa.io/get-pip.py
Resolving bootstrap.pypa.io (bootstrap.pypa.io)... 199.232.88.175, 2a04:4e42:56::175
Connecting to bootstrap.pypa.io (bootstrap.pypa.io)|199.232.88.175|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2279307 (2.2M) [text/x-python]
Saving to: ‘get-pip.py’

get-pip.py                              100%[===========================================================================================>]   2.17M  6.47MB/s    in 0.3s

2025-05-09 23:25:23 (6.47 MB/s) - ‘get-pip.py’ saved [2279307/2279307]
```

This capture represents the installation of get-pip.py

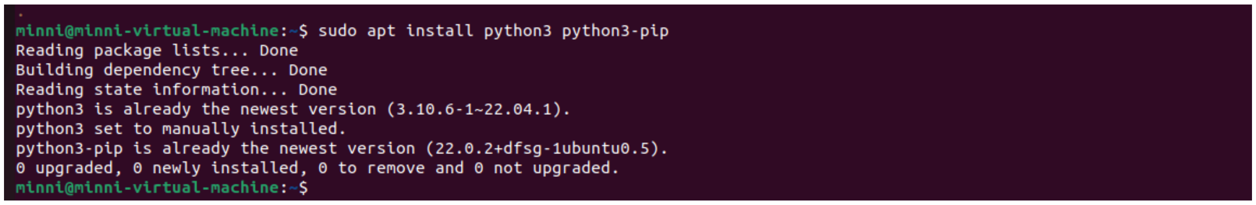
- In this step, we are going to install the python3 get-pip by following the below command.
- **Command: sudo python3 get-pip.py**
- This command was used after downloading the get-pip.py script (via wget or curl) to install pip3 manually when apt install python3-pip was unavailable or failed due to repository issues.



```
minni@minni-virtual-machine:~$ sudo python3 get-pip.py
Collecting pip
  Downloading pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
Downloading pip-25.1.1-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 15.5 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.2
    Uninstalling pip-22.0.2:
      Successfully uninstalled pip-22.0.2
Successfully installed pip-25.1.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable. It is re
commended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning
.
```

This capture represents the installation of get.pip.py and successful installation of python3-pip

- **Command: sudo apt install python3 python3-pip**



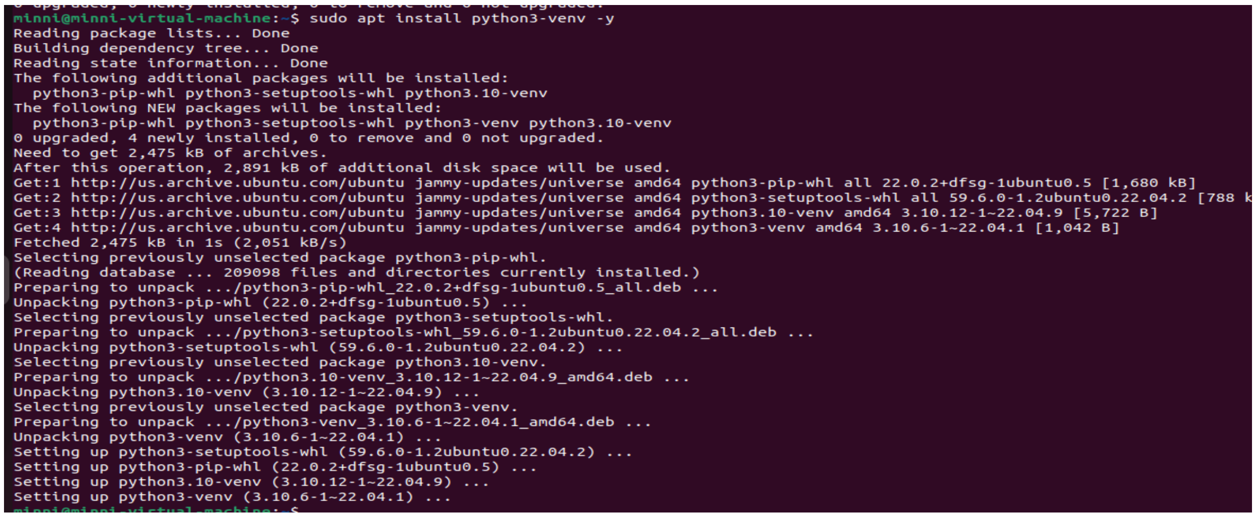
```
minni@minni-virtual-machine:~$ sudo apt install python3 python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
minni@minni-virtual-machine:~$
```

This capture representing the installation of python3-pip

- Further step, that I had followed is to install the python3 environment by using the following command.

Command: sudo apt install python3-venv -y

- After installing pip, this command was used to enable virtual environment creation. This helps maintain a clean Python workspace for the Netmiko script and any future network automation projects.



```
minni@minni-virtual-machine:~$ sudo apt install python3-venv -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pip-whl python3-setuptools-whl python3.10-venv
The following NEW packages will be installed:
  python3-pip-whl python3-setuptools-whl python3-venv python3.10-venv
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,475 kB of archives.
After this operation, 2,891 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-pip-whl all 22.0.2+dfsg-1ubuntu0.5 [1,680 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-setuptools-whl all 59.6.0-1.2ubuntu0.22.04.2 [788 k
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3.10-venv amd64 3.10.12-1~22.04.9 [5,722 B]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-venv amd64 3.10.6-1~22.04.1 [1,042 B]
Fetched 2,475 kB in 1s (2,051 kB/s)
Selecting previously unselected package python3-pip-whl.
(Reading database ... 209098 files and directories currently installed.)
Preparing to unpack .../python3-pip-whl_22.0.2+dfsg-1ubuntu0.5_all.deb ...
Unpacking python3-pip-whl (22.0.2+dfsg-1ubuntu0.5) ...
Selecting previously unselected package python3-setuptools-whl.
Preparing to unpack .../python3-setuptools-whl_59.6.0-1.2ubuntu0.22.04.2_all.deb ...
Unpacking python3-setuptools-whl (59.6.0-1.2ubuntu0.22.04.2) ...
Selecting previously unselected package python3.10-venv.
Preparing to unpack .../python3.10-venv_3.10.12-1~22.04.9_amd64.deb ...
Unpacking python3.10-venv (3.10.12-1~22.04.9) ...
Selecting previously unselected package python3-venv.
Preparing to unpack .../python3-venv_3.10.6-1~22.04.1_amd64.deb ...
Unpacking python3-venv (3.10.6-1~22.04.1) ...
Setting up python3-setuptools-whl (59.6.0-1.2ubuntu0.22.04.2) ...
Setting up python3-pip-whl (22.0.2+dfsg-1ubuntu0.5) ...
Setting up python3.10-venv (3.10.12-1~22.04.9) ...
Setting up python3-venv (3.10.6-1~22.04.1) ...
```

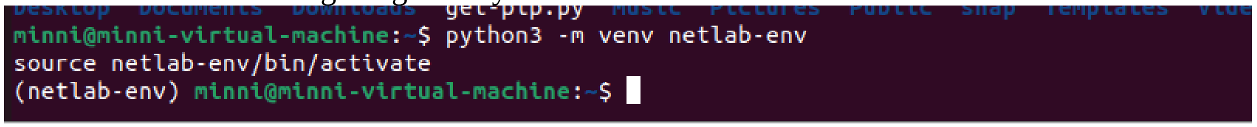
This capture represents the installation venv

- The following commands are used to activate the netlab environment by using the following commands

Command: python3 -m venv netlab-env

Source netlab-env/bin/activate

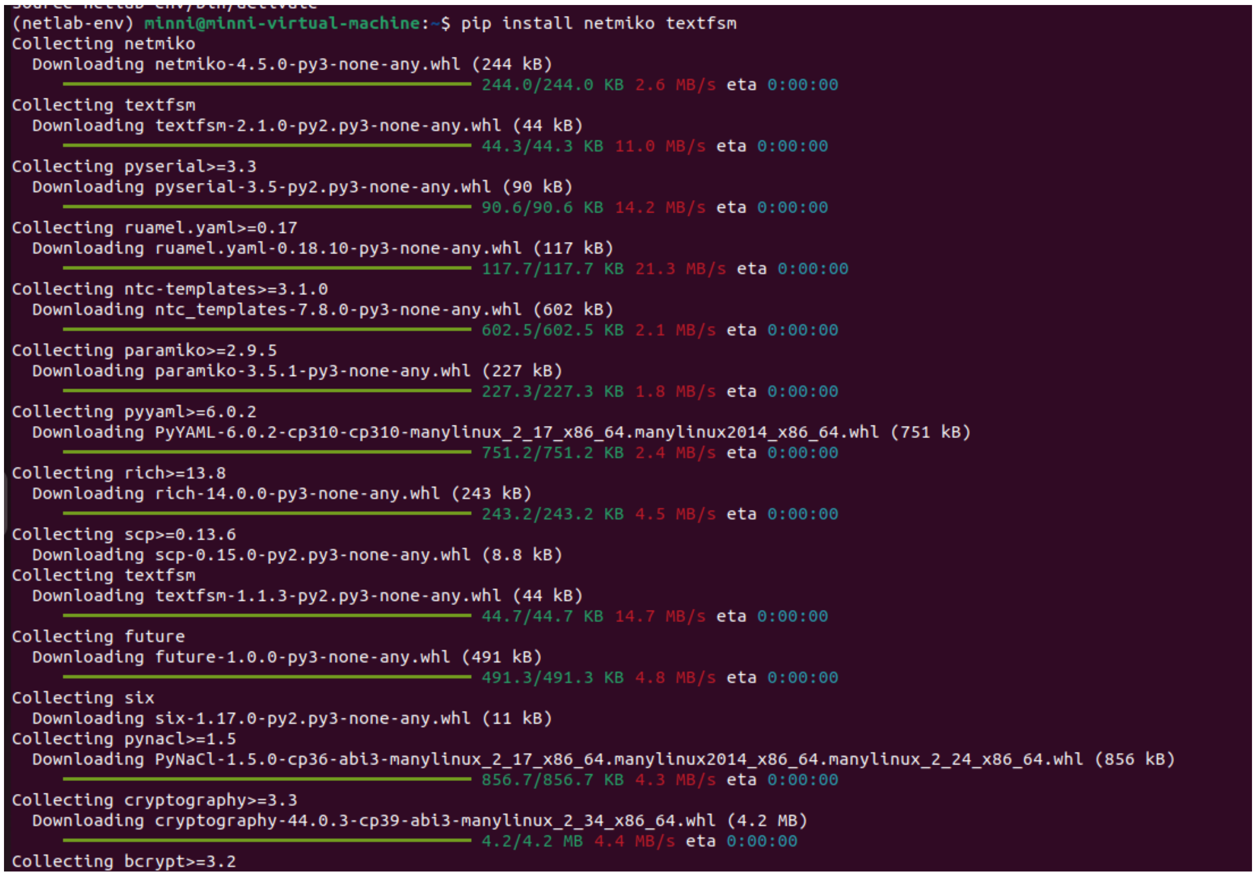
- This command creates a new Python virtual environment named netlab-env using the venv module. Virtual environments isolate Python dependencies for individual projects, preventing conflicts between packages.
- Used to create a clean, isolated Python environment for installing and testing the Netmiko-based SSH automation script.
- After activating the environment, you could safely install Netmiko and other Python libraries without affecting the global system environment.



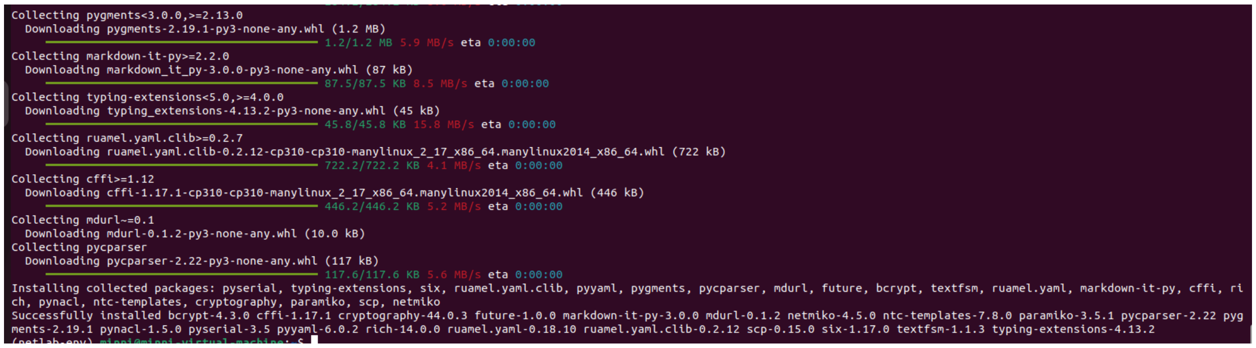
```
minni@minni-virtual-machine:~$ python3 -m venv netlab-env
source netlab-env/bin/activate
(netlab-env) minni@minni-virtual-machine:~$
```

This capture represents the activation of virtual environment.

- This command installs two important Python libraries — netmiko and textfsm — into your current Python environment. These libraries are essential for network automation tasks and structured CLI output parsing.
- **Command: pip install netmiko textfsm**
- After this, your Python script was able to automate backup and configuration tasks on Cisco network devices.



```
(netlab-env) minni@minni-virtual-machine:~$ pip install netmiko textfsm
Collecting netmiko
  Downloading netmiko-4.5.0-py3-none-any.whl (244 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 244.0/244.0 KB 2.6 MB/s eta 0:00:00
Collecting textfsm
  Downloading textfsm-2.1.0-py2.py3-none-any.whl (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.3/44.3 KB 11.0 MB/s eta 0:00:00
Collecting pyserial>=3.3
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 90.6/90.6 KB 14.2 MB/s eta 0:00:00
Collecting ruamel.yaml>=0.17
  Downloading ruamel.yaml-0.18.10-py3-none-any.whl (117 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.7/117.7 KB 21.3 MB/s eta 0:00:00
Collecting ntc-templates>=3.1.0
  Downloading ntc_templates-7.8.0-py3-none-any.whl (602 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 602.5/602.5 KB 2.1 MB/s eta 0:00:00
Collecting paramiko>=2.9.5
  Downloading paramiko-3.5.1-py3-none-any.whl (227 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 227.3/227.3 KB 1.8 MB/s eta 0:00:00
Collecting pyyaml>=6.0.2
  Downloading PyYAML-6.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (751 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 751.2/751.2 KB 2.4 MB/s eta 0:00:00
Collecting rich>=13.8
  Downloading rich-14.0.0-py3-none-any.whl (243 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 243.2/243.2 KB 4.5 MB/s eta 0:00:00
Collecting scp>=0.13.6
  Downloading scp-0.15.0-py2.py3-none-any.whl (8.8 kB)
Collecting textfsm
  Downloading textfsm-1.1.3-py2.py3-none-any.whl (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.7/44.7 KB 14.7 MB/s eta 0:00:00
Collecting future
  Downloading future-1.0.0-py3-none-any.whl (491 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.3/491.3 KB 4.8 MB/s eta 0:00:00
Collecting six
  Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Collecting pynacl>=1.5
  Downloading PyNaCl-1.5.0-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_24_x86_64.whl (856 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 856.7/856.7 KB 4.3 MB/s eta 0:00:00
Collecting cryptography>=3.3
  Downloading cryptography-44.0.3-cp39-abi3-manylinux_2_34_x86_64.whl (4.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.2/4.2 MB 4.4 MB/s eta 0:00:00
Collecting bcrypt>=3.2
```



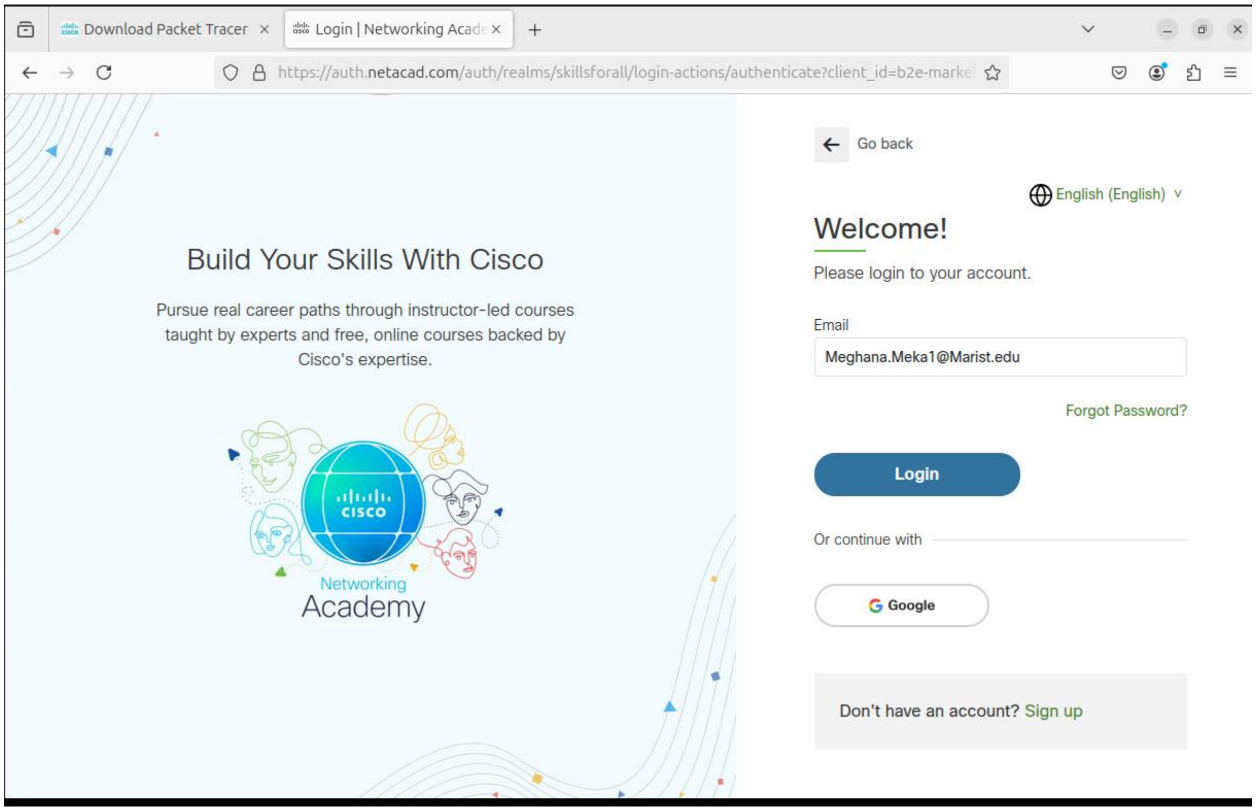
```
Collecting pygments<3.0.0,>=2.13.0
  Downloading pygments-2.19.1-py3-none-any.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 5.9 MB/s eta 0:00:00
Collecting markdown-it-py>=2.2.0
  Downloading markdown_it_py-3.0.0-py3-none-any.whl (87 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 87.5/87.5 KB 8.5 MB/s eta 0:00:00
Collecting typing-extensions<5.0,>=4.0.0
  Downloading typing_extensions-4.13.2-py3-none-any.whl (45 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.8/45.8 KB 15.8 MB/s eta 0:00:00
Collecting ruamel.yaml.clib>=0.2.7
  Downloading ruamel.yaml.clib-0.2.12-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (722 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 722.2/722.2 KB 4.1 MB/s eta 0:00:00
Collecting cffi>=1.12
  Downloading cffi-1.17.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (446 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 446.2/446.2 KB 5.2 MB/s eta 0:00:00
Collecting mdurl~=0.1
  Downloading mdurl-0.1.2-py3-none-any.whl (10.0 kB)
Collecting pycparser
  Downloading pycparser-2.22-py3-none-any.whl (117 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.6/117.6 KB 5.6 MB/s eta 0:00:00
Installing collected packages: pyserial, typing-extensions, six, ruamel.yaml.clib, pyyaml, pygments, pycparser, mdurl, future, bcrypt, textfsm, ruamel.yaml, markdown-it-py, cffi, ri
ch, pynacl, ntc-templates, cryptography, paramiko, scp, netmiko
Successfully installed bcrypt-4.3.0 cffi-1.17.1 cryptography-44.0.3 future-1.0.0 markdown-it-py-3.0.0 mdurl-0.1.2 netmiko-4.5.0 ntc-templates-7.8.0 paramiko-3.5.1 pycparser-2.22 pyg
ments-2.19.1 pynacl-1.5.0 pyserial-3.5 pyyaml-6.0.2 rich-14.0.0 ruamel.yaml-0.18.10 ruamel.yaml.clib-0.2.12 scp-0.15.0 six-1.17.0 textfsm-1.1.3 typing-extensions-4.13.2
```

This capture represents the complete installation of netmiko textfsm

<u>Step-II Simulated Lab Setup in Cisco Packet Tracer</u>

In this step, we are going to install the cisco packet tracer for the simulated lab environment.

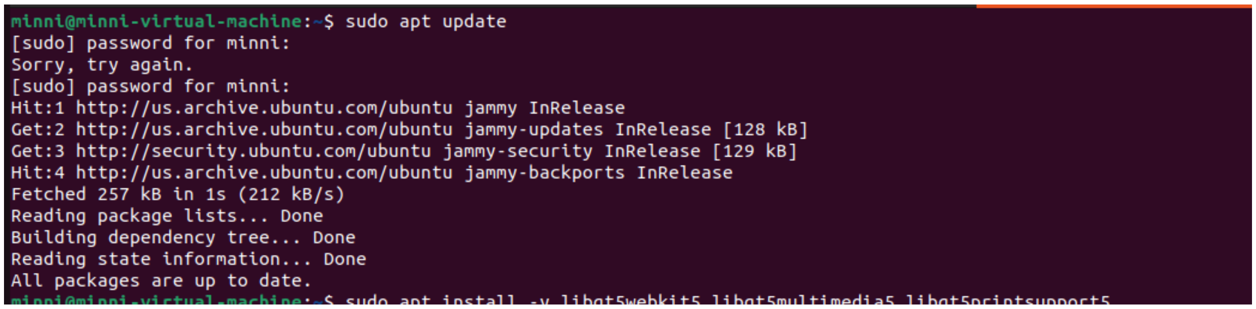
- The first step that I had followed is the download of packet tracer by logging into the cisco netacad with the credentials that I have already associate with Marist email


Download Packet Tracer
Login | Networking Acade
https://auth.netacad.com/auth/realms/skillsforall/login-actions/authenticate?client_id=b2e-marke
Go back
English (English)
Build Your Skills With Cisco
Pursue real career paths through instructor-led courses taught by experts and free, online courses backed by Cisco's expertise.
cisco
Networking Academy
Welcome!
Please login to your account.
Email
Meghana.Meka1@Marist.edu
Forgot Password?
Login
Or continue with
Google
Don't have an account? Sign up

This capture represents the login page of cisco netacad for downloading packet tracer

- After successful download, the packet tracer will be installed in the ubuntu for the simulation.
- Initiated the process by following the below command, which is used to update all the packages that already existing in ubuntu

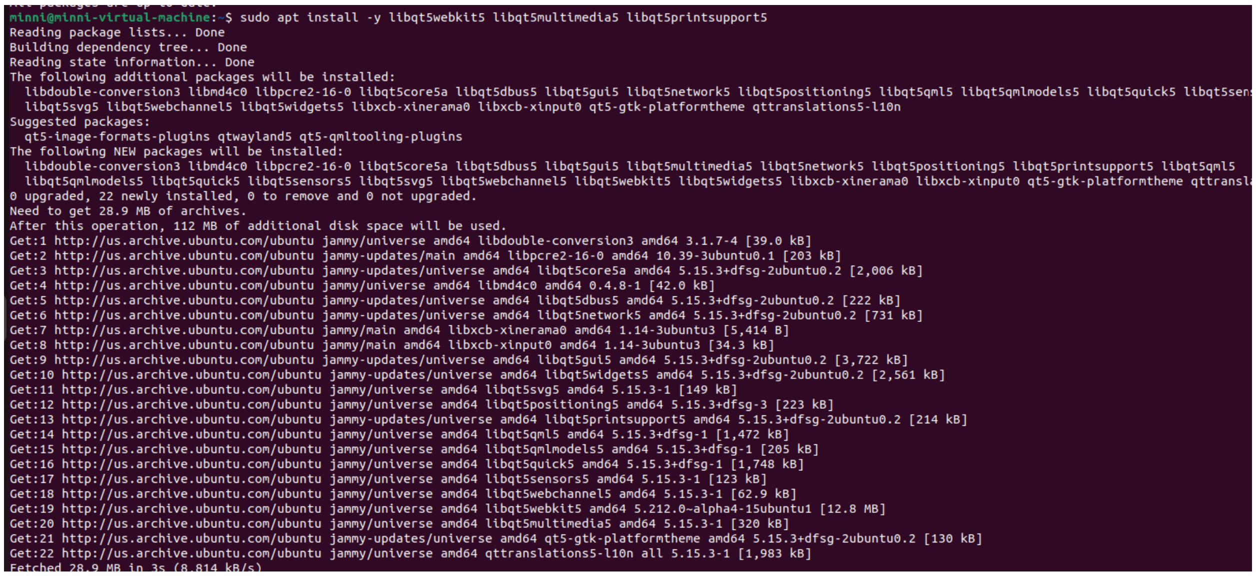
Command: sudo apt update



```
minni@minni-virtual-machine:~$ sudo apt update
[sudo] password for minni:
Sorry, try again.
[sudo] password for minni:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 257 kB in 1s (212 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
minni@minni-virtual-machine:~$ sudo apt install -y libqt5webkit5 libqt5multimedia5 libqt5printsupport5
```

This capture represents the update of ubuntu packages during installation of packet tracer

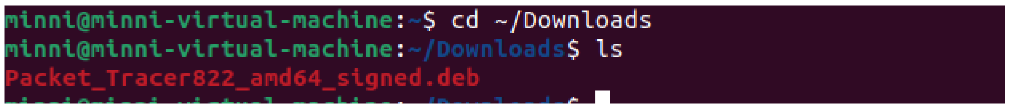
- The next step, that I had followed is to install the necessary dependency libraries by following the below command for the installation of packet tracer.
- **Command: sudo apt install -y libqt5webkit5 libqt5multimedia5 libqt5printsupport5**



```
minni@minni-virtual-machine:~$ sudo apt install -y libqt5webkit5 libqt5multimedia5 libqt5printsupport5
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdouble-conversion3 libmd4c0 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5 libqt5positioning5 libqt5qml5 libqt5qmlmodels5 libqt5quick5 libqt5sen
  libqt5svg5 libqt5webchannel5 libqt5widgets5 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n
Suggested packages:
  qt5-image-formats-plugins qtwayland5 qt5-qmltooling-plugins
The following NEW packages will be installed:
  libdouble-conversion3 libmd4c0 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5network5 libqt5positioning5 libqt5printsupport5 libqt5qml5
  libqt5qmlmodels5 libqt5quick5 libqt5sensors5 libqt5svg5 libqt5webchannel5 libqt5webkit5 libqt5widgets5 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttransla
0 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.9 MB of archives.
After this operation, 112 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libdouble-conversion3 amd64 3.1.7-4 [39.0 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpcre2-16-0 amd64 10.39-3ubuntu0.1 [203 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5core5a amd64 5.15.3+dfsg-2ubuntu0.2 [2,006 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libmd4c0 amd64 0.4.8-1 [42.0 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5dbus5 amd64 5.15.3+dfsg-2ubuntu0.2 [222 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5network5 amd64 5.15.3+dfsg-2ubuntu0.2 [731 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb-xinerama0 amd64 1.14-3ubuntu3 [5,414 B]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb-xinput0 amd64 1.14-3ubuntu3 [34.3 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5gui5 amd64 5.15.3+dfsg-2ubuntu0.2 [3,722 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5widgets5 amd64 5.15.3+dfsg-2ubuntu0.2 [2,561 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5svg5 amd64 5.15.3-1 [149 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5positioning5 amd64 5.15.3+dfsg-3 [223 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libqt5printsupport5 amd64 5.15.3+dfsg-2ubuntu0.2 [214 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5qml5 amd64 5.15.3+dfsg-1 [1,472 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5qmlmodels5 amd64 5.15.3+dfsg-1 [205 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5quick5 amd64 5.15.3+dfsg-1 [1,748 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5sensors5 amd64 5.15.3-1 [123 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5webchannel5 amd64 5.15.3-1 [62.9 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5webkit5 amd64 5.212.0~alpha4-15ubuntu1 [12.8 MB]
Get:20 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqt5multimedia5 amd64 5.15.3-1 [320 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 qt5-gtk-platformtheme amd64 5.15.3+dfsg-2ubuntu0.2 [130 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 qttranslations5-l10n all 5.15.3-1 [1,983 kB]
Fetched 28.9 MB in 3s (8,814 kB/s)
```

This capture represents the installation of dependencies

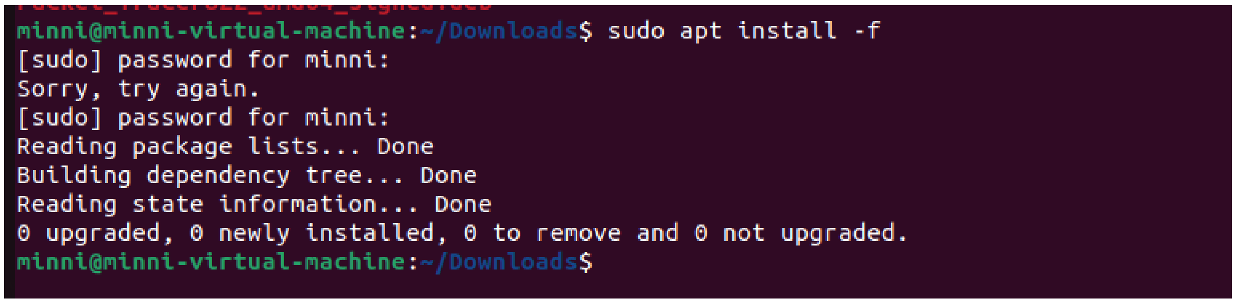
- Now, we will move to further steps that to move to the downloads to look for the downloaded packet tracer file, by using the following command.
- **Command: cd ~/Downloads**
- **ls**



```
minni@minni-virtual-machine:~$ cd ~/Downloads
minni@minni-virtual-machine:~/Downloads$ ls
Packet_Tracer822_amd64_signed.deb
```

This capture represents that we are looking for the packet tracer file in downloads from terminal.

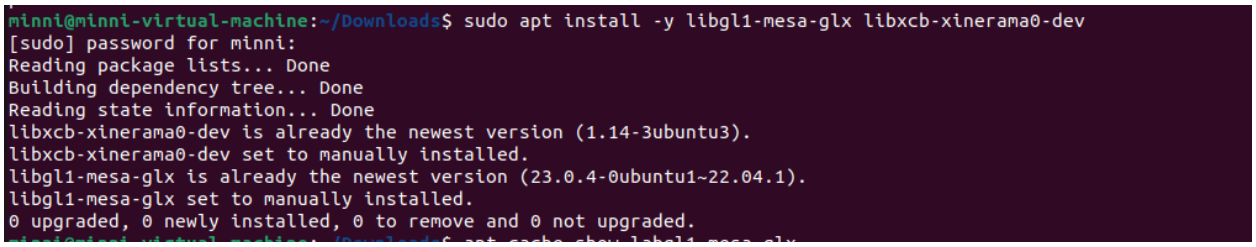
- The step that I had used is the installation of missing packages to avoid the errors while installing the packet Tracer, by using the following command
- **Sudo apt install -f**



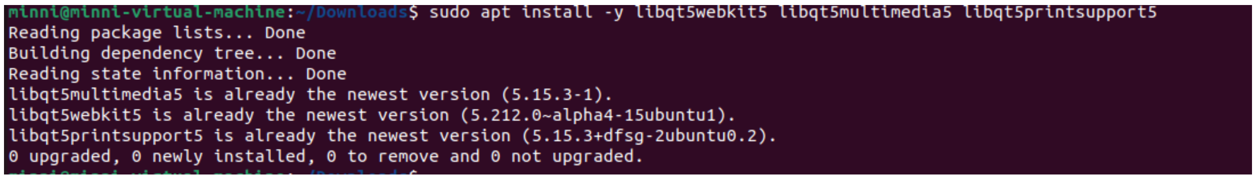
```
minni@minni-virtual-machine:~/Downloads$ sudo apt install -f
[sudo] password for minni:
Sorry, try again.
[sudo] password for minni:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
minni@minni-virtual-machine:~/Downloads$
```

This capture represents the installation of missing packages

- The further step that I had followed is for the installation of the other dependencies that are already existing in this ubuntu version for proceeding for the installation of packet Tracer



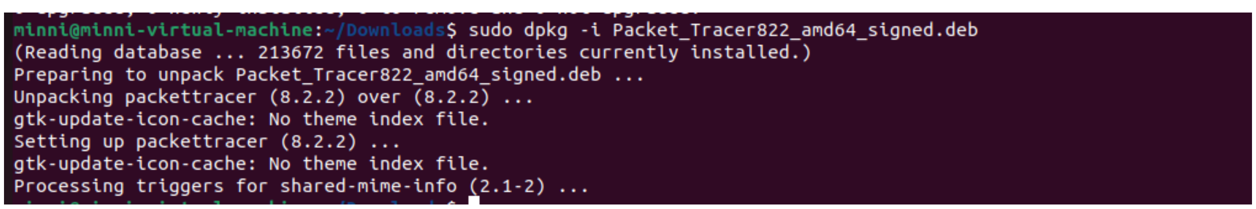
```
minni@minni-virtual-machine:~/Downloads$ sudo apt install -y libgl1-mesa-glx libxcb-xinerama0-dev
[sudo] password for minni:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libxcb-xinerama0-dev is already the newest version (1.14-3ubuntu3).
libxcb-xinerama0-dev set to manually installed.
libgl1-mesa-glx is already the newest version (23.0.4-0ubuntu1~22.04.1).
libgl1-mesa-glx set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```



```
minni@minni-virtual-machine:~/Downloads$ sudo apt install -y libqt5webkit5 libqt5multimedia5 libqt5printsupport5
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libqt5multimedia5 is already the newest version (5.15.3-1).
libqt5webkit5 is already the newest version (5.212.0~alpha4-15ubuntu1).
libqt5printsupport5 is already the newest version (5.15.3+dfsg-2ubuntu0.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

This capture represents the facing dependency installation

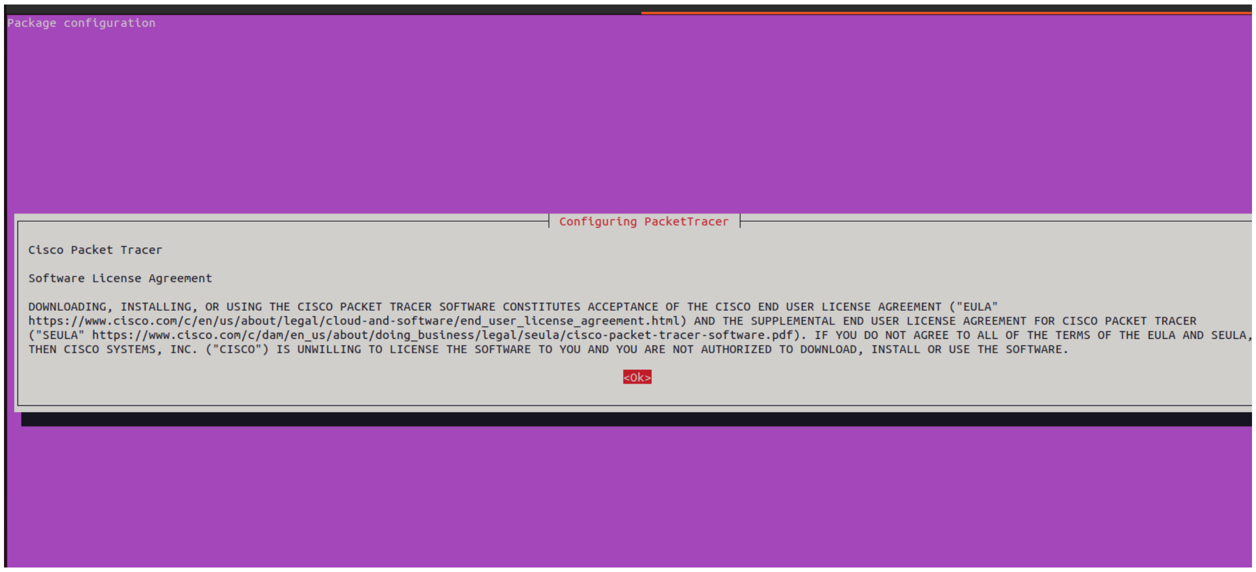
- The next step that I had used is to install the packet tracer in the ubuntu by following the below command.
- **Command: sudo dpkg -I Packet_Tracer822_amd64_signed.deb**



```
minni@minni-virtual-machine:~/Downloads$ sudo dpkg -i Packet_Tracer822_amd64_signed.deb
(Reading database ... 213672 files and directories currently installed.)
Preparing to unpack Packet_Tracer822_amd64_signed.deb ...
Unpacking packettracer (8.2.2) over (8.2.2) ...
gtk-update-icon-cache: No theme index file.
Setting up packettracer (8.2.2) ...
gtk-update-icon-cache: No theme index file.
Processing triggers for shared-mime-info (2.1-2) ...
```

This capture represents the installation of packet tracer

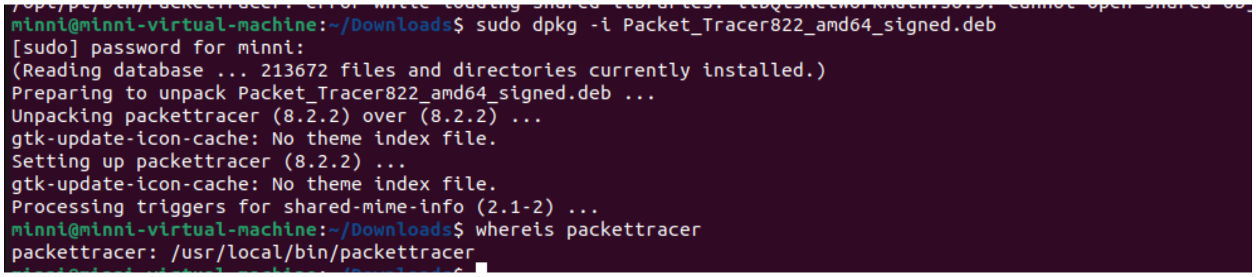
- After, running the above mentioned process, the screen will be displaying the licence agreement that we need to accept for the installation



```
Package configuration

Configuring PacketTracer

Cisco Packet Tracer

Software License Agreement

DOWNLOADING, INSTALLING, OR USING THE CISCO PACKET TRACER SOFTWARE CONSTITUTES ACCEPTANCE OF THE CISCO END USER LICENSE AGREEMENT ("EULA"
https://www.cisco.com/c/en/us/about/legal/cloud-and-software/end_user_license_agreement.html) AND THE SUPPLEMENTAL END USER LICENSE AGREEMENT FOR CISCO PACKET TRACER
("SEULA" https://www.cisco.com/c/dam/en_us/about/doing_business/legal/seula/cisco-packet-tracer-software.pdf). IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THE EULA AND SEULA,
THEN CISCO SYSTEMS, INC. ("CISCO") IS UNWILLING TO LICENSE THE SOFTWARE TO YOU AND YOU ARE NOT AUTHORIZED TO DOWNLOAD, INSTALL OR USE THE SOFTWARE.

<Ok>
```

This capture represents the accepting the licence agreement while installation of packet tracer

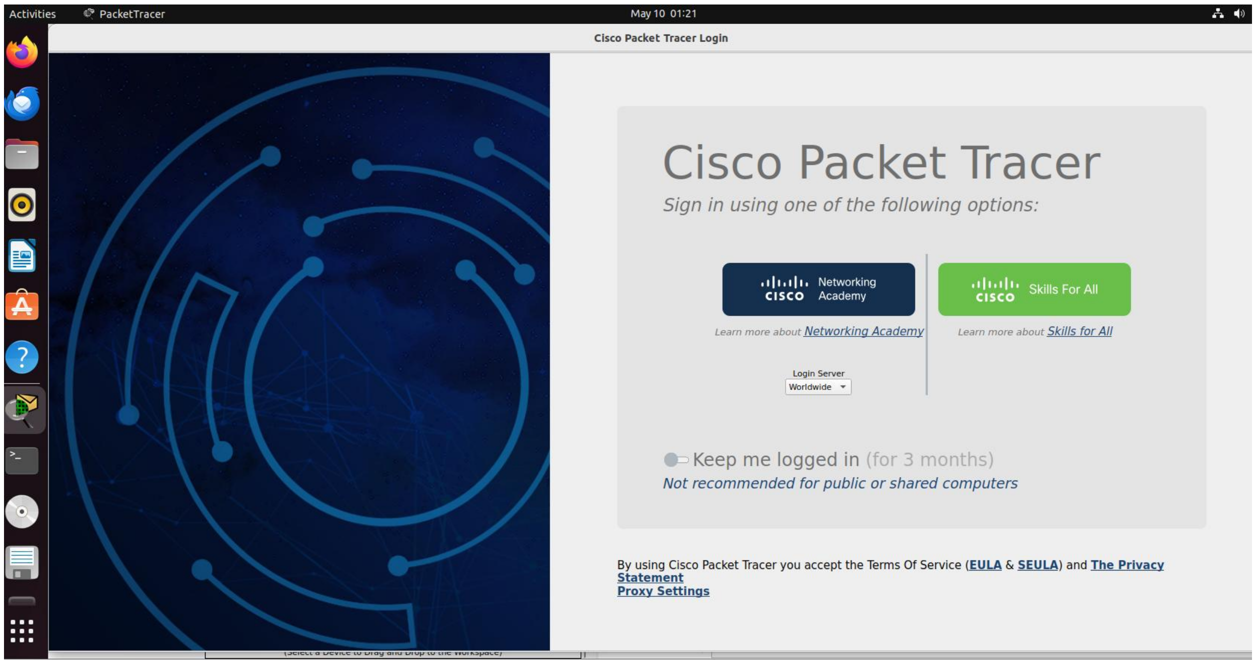
- After accepting the agreement, the installation will be done with all the steps.
- After the installation we use the following command to find the location of packet tracer.
- **Command: whereis packettracer**



```
minni@minni-virtual-machine:~/Downloads$ sudo dpkg -i Packet_Tracer822_amd64_signed.deb
[sudo] password for minni:
(Reading database ... 213672 files and directories currently installed.)
Preparing to unpack Packet_Tracer822_amd64_signed.deb ...
Unpacking packettracer (8.2.2) over (8.2.2) ...
gtk-update-icon-cache: No theme index file.
Setting up packettracer (8.2.2) ...
gtk-update-icon-cache: No theme index file.
Processing triggers for shared-mime-info (2.1-2) ...
minni@minni-virtual-machine:~/Downloads$ whereis packettracer
packettracer: /usr/local/bin/packettracer
```

This capture represents the unpacking of packet Tracer and its location

- **Launching of Packet Tracer:** After unpacking the packet tracer, we will open the packet tracer by logging in with the cisco netacad, all the below 3 captures are involved in opening of packet tracer by logging in with netacad.


Activities
PacketTracer
May 10 01:21
Cisco Packet Tracer Login
Networking Academy
Skills For All
Learn more about Networking Academy
Learn more about Skills for All
Login Server
Worldwide

Cisco Packet Tracer

Sign in using one of the following options:

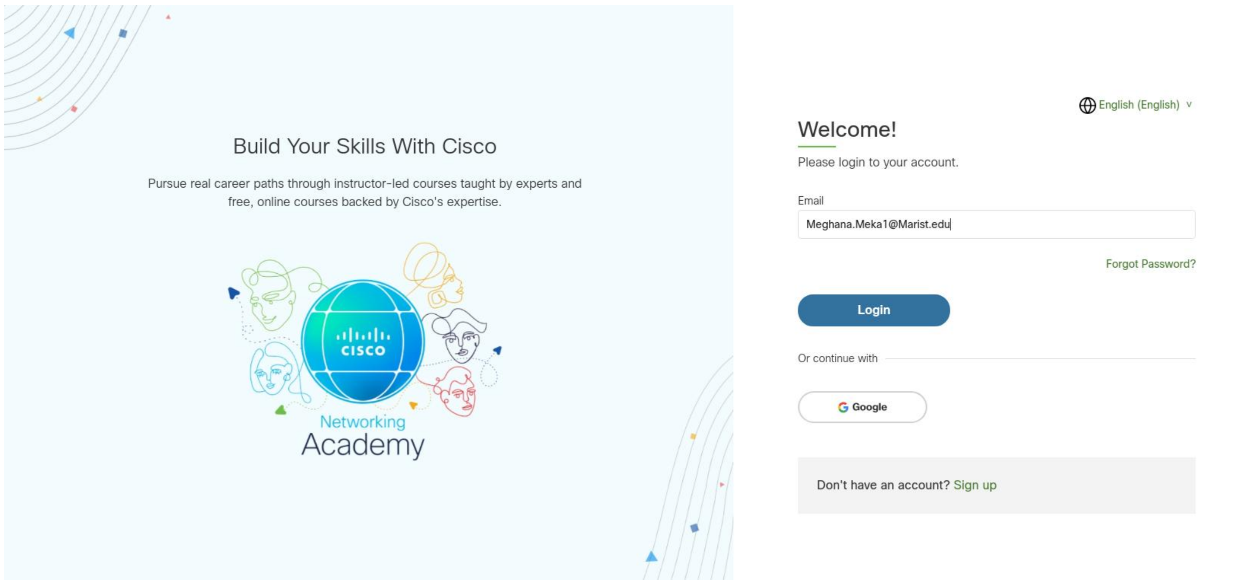
Keep me logged in (for 3 months)

Not recommended for public or shared computers

By using Cisco Packet Tracer you accept the Terms Of Service (EULA & SEULA) and The Privacy Statement

Proxy Settings

This Capture represents the login page of packet tracer in ubuntu.



Build Your Skills With Cisco

Pursue real career paths through instructor-led courses taught by experts and free, online courses backed by Cisco's expertise.

Welcome!

Please login to your account.

Email

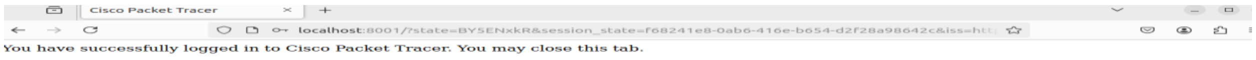
Meghana.Meka1@Marist.edu

Forgot Password?

Or continue with

Don't have an account? Sign up

This capture represents the logging in with the netacad id and password



localhost:8001/?state=BY5ENxkR&session_state=f68241e8-0ab6-416e-b654-d2f28a98642c&iss=htt

You have successfully logged in to Cisco Packet Tracer. You may close this tab.

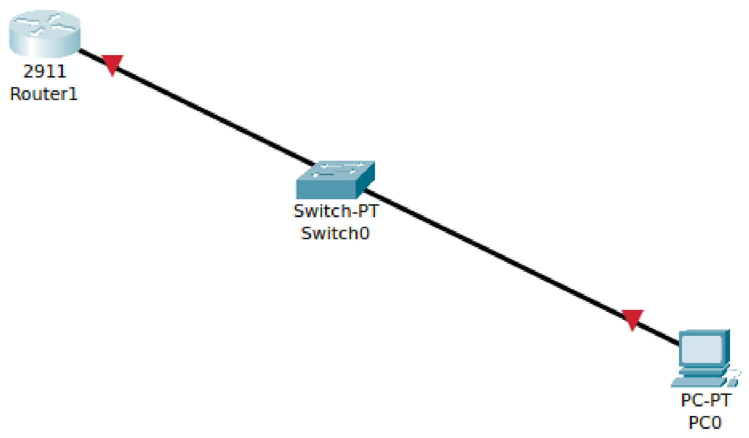
Successfully logged into the packet tracer

- After launching the packet tracer, the next step we are going to follow is adding devices to workspace, the following devices will be added to the workspace,
  - A cisco router
  - A switch
  - A pc
- After adding all the devices, we are going to connect the devices with the copper straight cable which is used to connect the devices.

Use **Copper Straight-Through Cable**

- Connect:
- **PC → Router (via Switch or directly)**:
  - PC: FastEthernet0
  - Router: GigabitEthernet0/0 (G0/0)

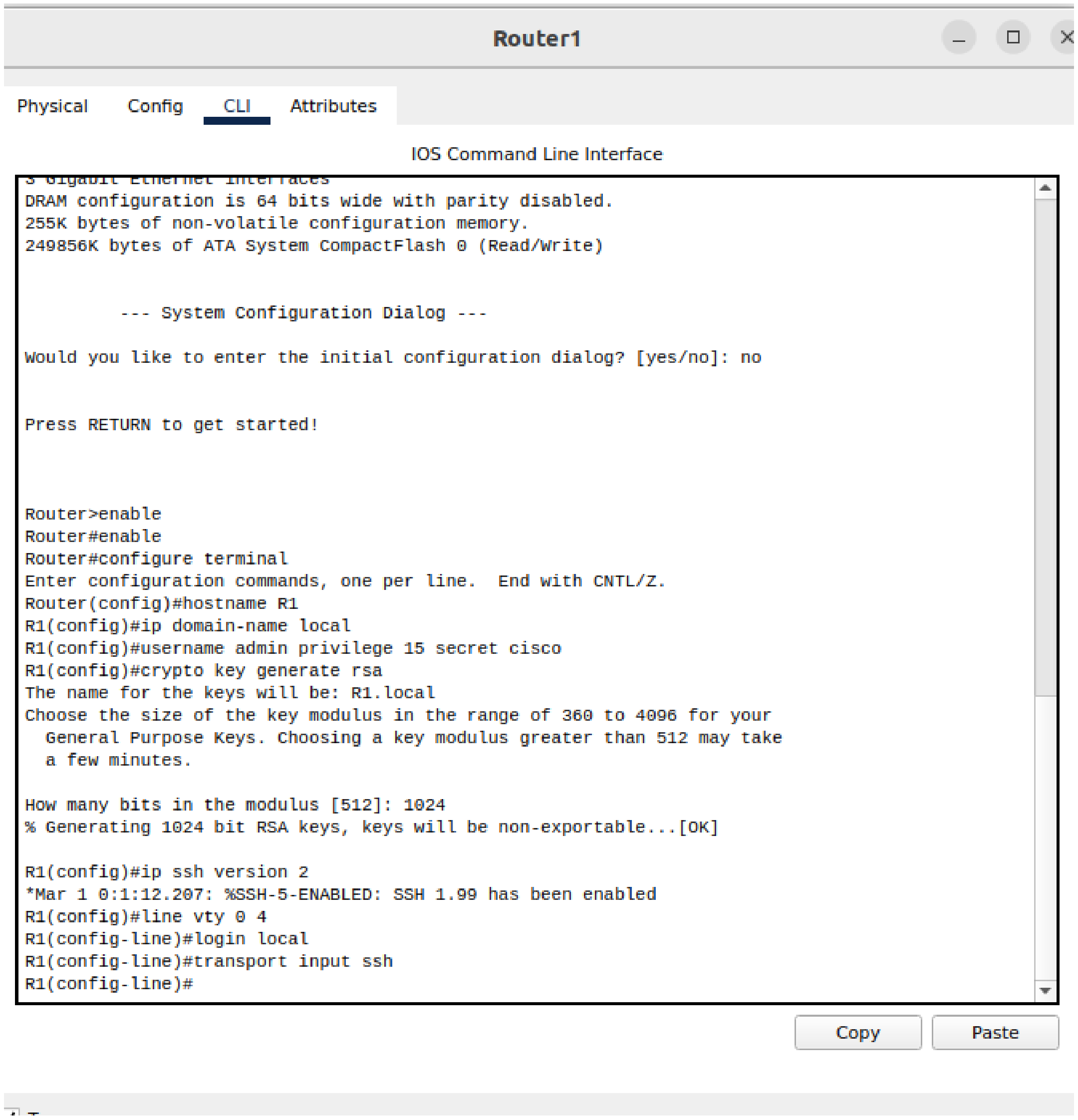
- As shown in the below picture, we are able to establish the connections in packet tracer.



Successfully establishing the connections in packet tracer

- **Configuring the router:**
- Click on the router then go to cli tab for configuring the router.
- We have followed the cli bash for router configuration.

enable
configure terminal
hostname R1
ip domain-name local
username admin privilege 15 secret cisco
crypto key generate rsa
1024
ip ssh version 2
line vty 0 4
login local
transport input ssh

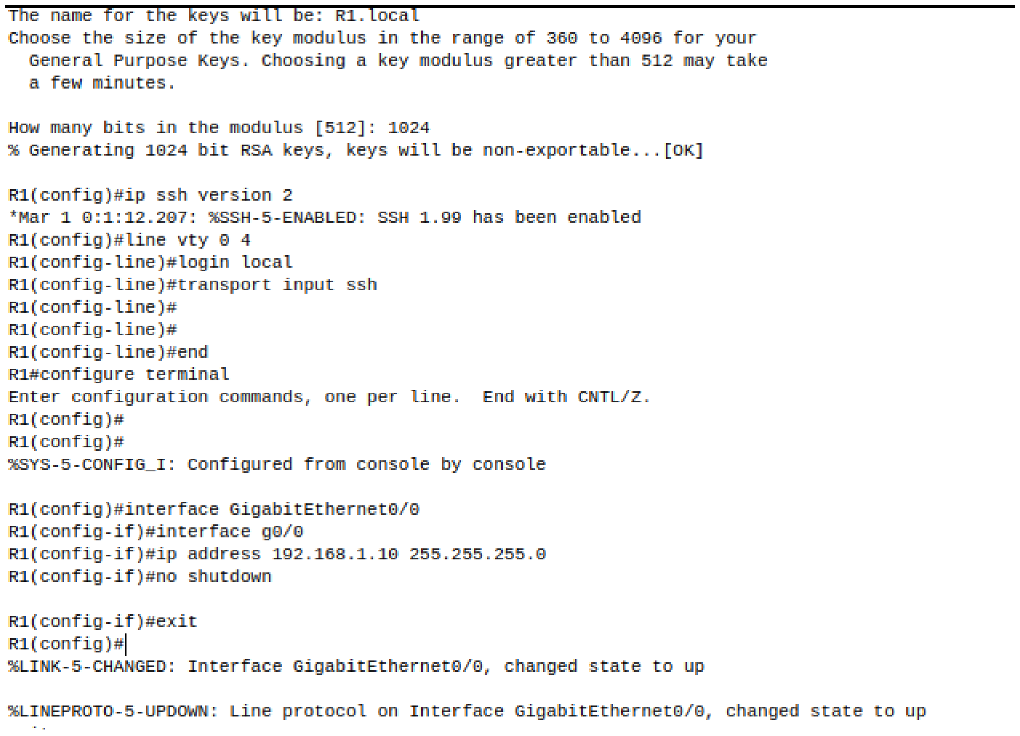


```
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#ip domain-name local
R1(config)#username admin privilege 15 secret cisco
R1(config)#crypto key generate rsa
The name for the keys will be: R1.local
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#ip ssh version 2
*Mar 1 0:1:12.207: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#transport input ssh
R1(config-line)#
```

Modified cli bash for the router

- **Assign ip addresses to the interface:**

- The following bash is required to assign the ip addresses to the interface.
  - **interface g0/0**
  - **ip address 192.168.1.10 255.255.255.0**
  - **no shutdown**
  - **exit**

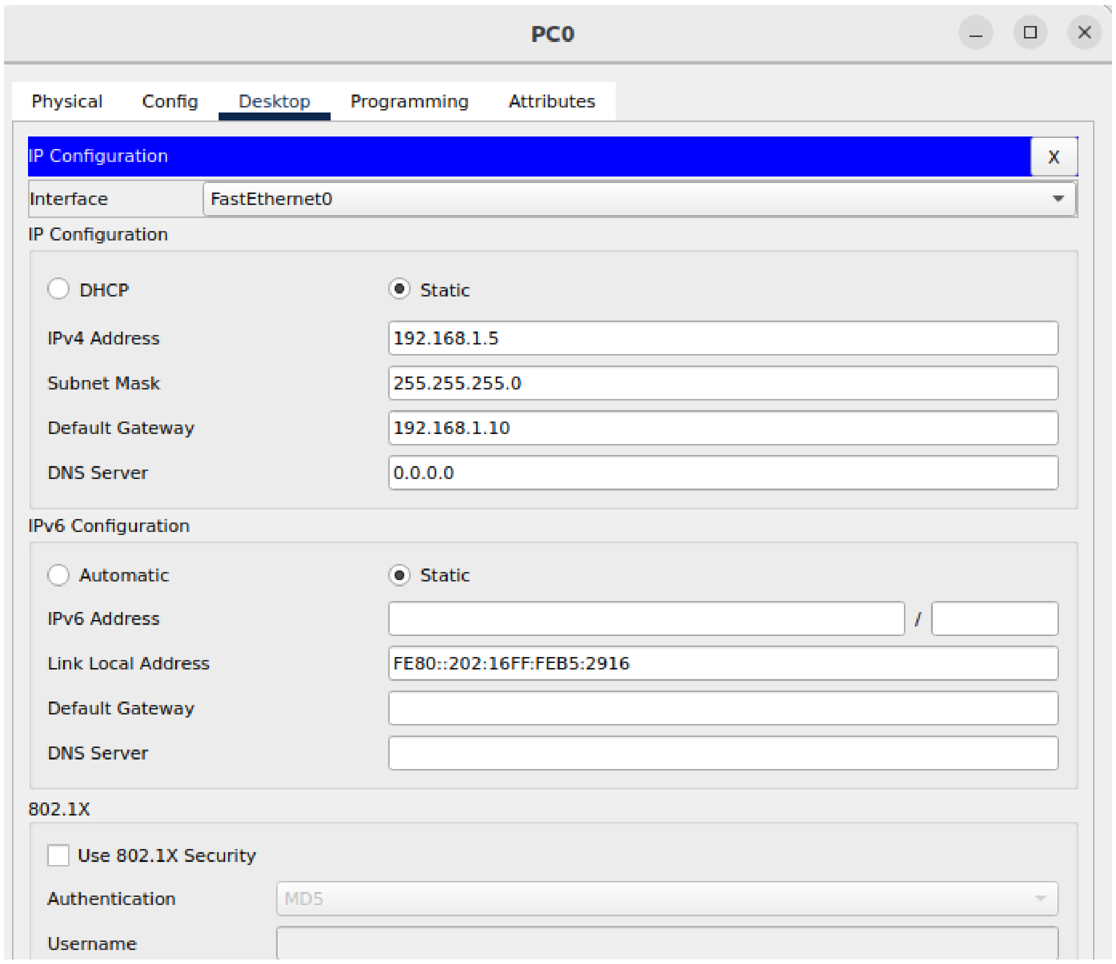


```
The name for the keys will be: R1.local
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#ip ssh version 2
*Mar 1 0:1:12.207: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#transport input ssh
R1(config-line)#
R1(config-line)#
R1(config-line)#end
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#
R1(config)#
%SYS-5-CONFIG_I: Configured from console by console

R1(config)#interface GigabitEthernet0/0
R1(config-if)#interface g0/0
R1(config-if)#ip address 192.168.1.10 255.255.255.0
R1(config-if)#no shutdown

R1(config-if)#exit
R1(config)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
```

This capture represents the modified ip state to up for the router.

Configure the PC's IP Address

- Click on the **PC**
- Go to **Desktop > IP Configuration**
- Set:
  - **IP Address**: 192.168.1.5
  - **Subnet Mask**: 255.255.255.0
  - **Default Gateway**: 192.168.1.10



PC0

Physical | Config | Desktop | Programming | Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration

DHCP [✓ Static]

IPv4 Address: 192.168.1.5

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.10

DNS Server: 0.0.0.0

IPv6 Configuration

Automatic [✓ Static]

IPv6 Address: /

Link Local Address: FE80::202:16FF:FEB5:2916

Default Gateway:

DNS Server:

802.1X

☐ Use 802.1X Security

Authentication: MD5

Username:

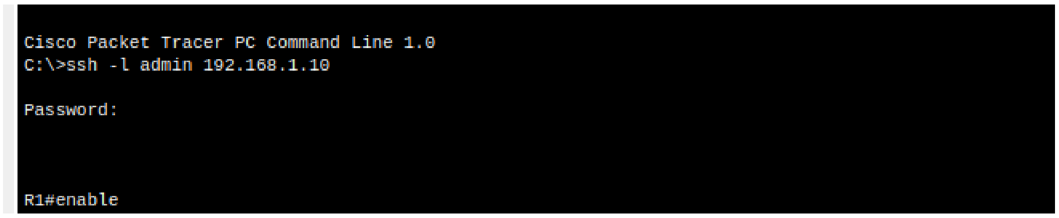
This capture represents the alotted ip configuration for PC

Test SSH Connection (From PC)

Go to Desktop > Terminal

Type: ssh -l admin 192.168.1.10

Enter password: cisco

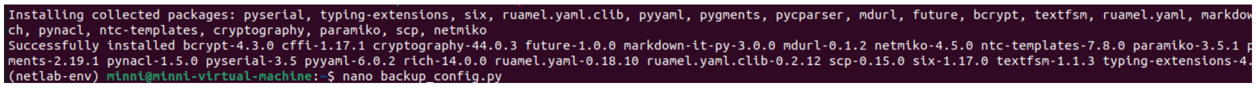


```
Cisco Packet Tracer PC Command Line 1.0
C:\>ssh -l admin 192.168.1.10

Password:


R1#enable
```

This capture represents the test of SSH connection

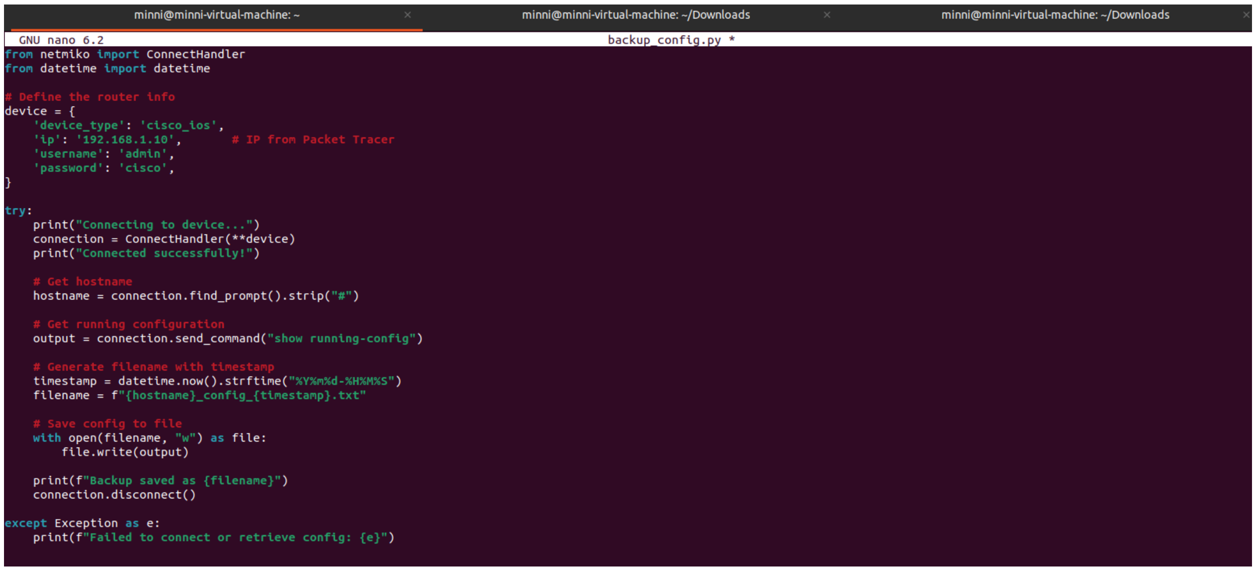
<u>Step-III – Automation with python:</u>

Command: nano backup-config.py



```
Installing collected packages: pyserial, typing-extensions, six, ruamel.yaml.clib, pyyaml, pygments, pycparser, mdurl, future, bcrypt, textfsm, ruamel.yaml, markdow
ch, pynacl, ntc-templates, cryptography, paramiko, scp, netmiko
Successfully installed bcrypt-4.3.0 cffi-1.17.1 cryptography-44.0.3 future-1.0.0 markdown-it-py-3.0.0 mdurl-0.1.2 netmiko-4.5.0 ntc-templates-7.8.0 paramiko-3.5.1 p
ments-2.19.1 pynacl-1.5.0 pyserial-3.5 pyyaml-6.0.2 rich-14.0.0 ruamel.yaml-0.18.10 ruamel.yaml.clib-0.2.12 scp-0.15.0 six-1.17.0 textfsm-1.1.3 typing-extensions-4.
(netlab-env) minni@minni-virtual-machine:~$ nano backup_config.py
```

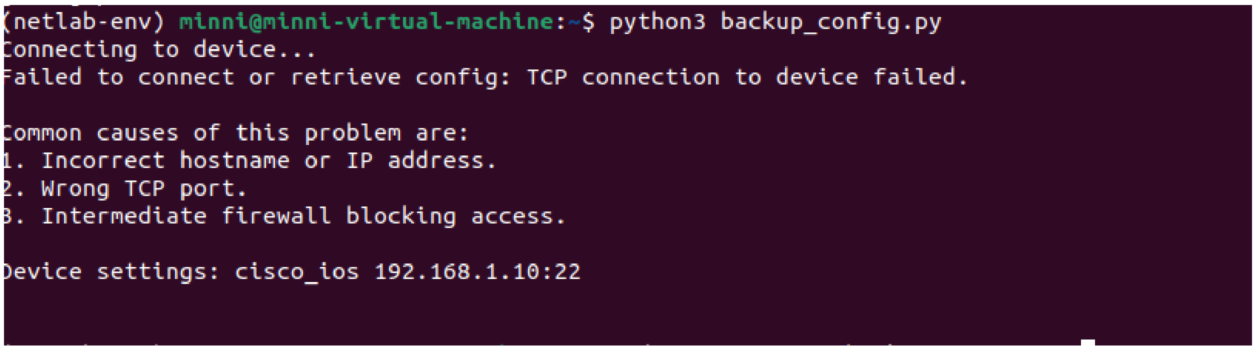
This capture represents the creation of backup_config.py file



minni@minni-virtual-machine: ~ | minni@minni-virtual-machine: ~/Downloads | minni@minni-virtual-machine: ~/Downloads

```
  GNU nano 6.2                                   backup_config.py *
from netmiko import ConnectHandler
from datetime import datetime

# Define the router info
device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.1.10',       # IP from Packet Tracer
    'username': 'admin',
    'password': 'cisco',
}

try:
    print("Connecting to device...")
    connection = ConnectHandler(**device)
    print("Connected successfully!")

    # Get hostname
    hostname = connection.find_prompt().strip("#")

    # Get running configuration
    output = connection.send_command("show running-config")

    # Generate filename with timestamp
    timestamp = datetime.now().strftime("%Y%m%d-%H%M%S")
    filename = f"{hostname}_config_{timestamp}.txt"

    # Save config to file
    with open(filename, "w") as file:
        file.write(output)

    print(f"Backup saved as {filename}")
    connection.disconnect()

except Exception as e:
    print(f"Failed to connect or retrieve config: {e}")
```

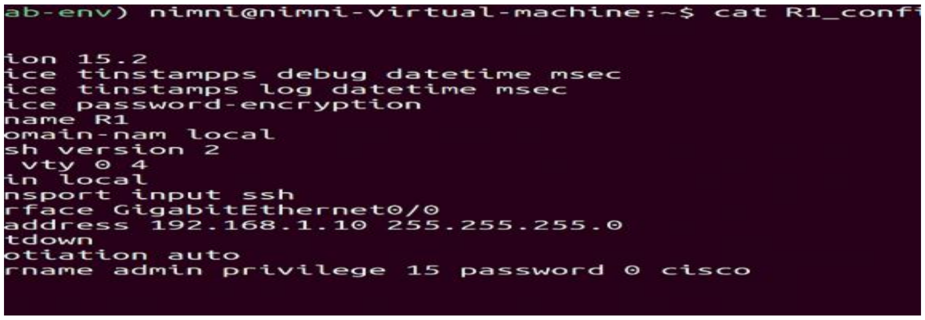
This capture represents the backup configuration of .py for automation



```
(netlab-env) minni@minni-virtual-machine:~$ python3 backup_config.py
Connecting to device...
Failed to connect or retrieve config: TCP connection to device failed.

Common causes of this problem are:
1. Incorrect hostname or IP address.
2. Wrong TCP port.
3. Intermediate firewall blocking access.

Device settings: cisco_ios 192.168.1.10:22
```

The next step that I had followed was to verify the config file by following the below command

cat R1_config_*.txt



```
ab-env) nimni@nimni-virtual-machine:~$ cat R1_confi

ion 15.2
ice tinstampps debug datetime msec
ice tinstamps log datetime msec
ice password-encryption
name R1
omain-nam local
sh version 2
 vty 0 4
in local
nsport input ssh
rface GigabitEthernet0/0
address 192.168.1.10 255.255.255.0
tdown
otiation auto
rname admin privilege 15 password 0 cisco
```

This capture represents the verification of config.

Conclusion

This project successfully demonstrated the process of network automation using Python and Netmiko by simulating SSH-based interactions with a Cisco router configured in Cisco Packet Tracer. From configuring the router with SSH access to writing and executing a Python script that retrieves and stores its running configuration, the lab encapsulated a full automation workflow. The use of a virtual environment in Ubuntu allowed for isolated testing and clean dependency management, ensuring the lab was portable, modular, and repeatable. This hands-on exercise reinforced core skills such as working with command-line tools, installing packages manually, and using Python to automate real-world network tasks.

Beyond technical execution, the project highlighted the importance of network simulation tools like Packet Tracer in academic and lab environments, especially when hardware access is limited. It also demonstrated troubleshooting strategies when working across virtual layers, such as resolving SSH communication between simulated devices and a virtual machine. Overall, this lab not only accomplished the automation objective but also strengthened practical understanding of Python-based network scripting, Linux administration, and Cisco IOS configurations — all foundational skills for aspiring network engineers.

References

1. Netmiko Documentation – https://ktbyers.github.io/netmiko/
2. Cisco Packet Tracer Official Download – https://www.netacad.com/courses/packet-tracer
3. Python Virtual Environments – https://docs.python.org/3/library/venv.html
4. Ubuntu APT Command Reference – https://help.ubuntu.com/community/AptGet/Howto
5. Python pip Installation Guide – https://pip.pypa.io/en/stable/installation/