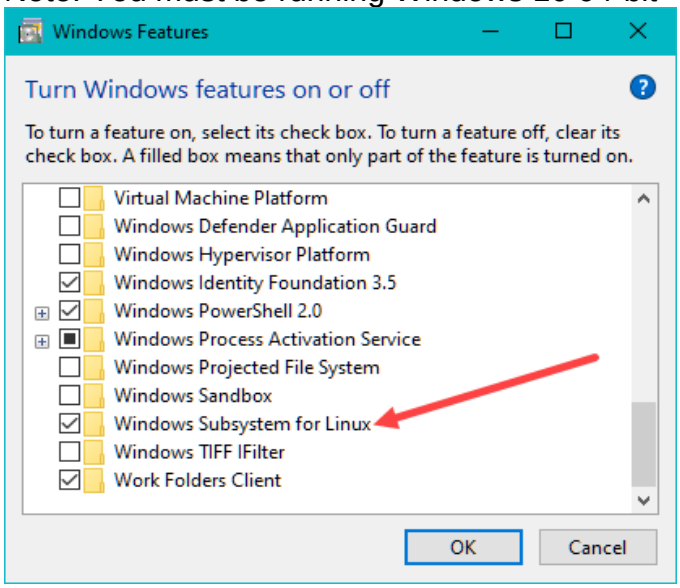
Prerequisites for Network Automation Lab

1. Activate the Windows Subsystem for Linux (WSL) from the Control Panel:

   `Control Panel\All Control Panel Items\Programs and Features > Turn Windows features on or off > Windows Subsystem for Linux`

   Note: You must be running Windows 10 64-bit



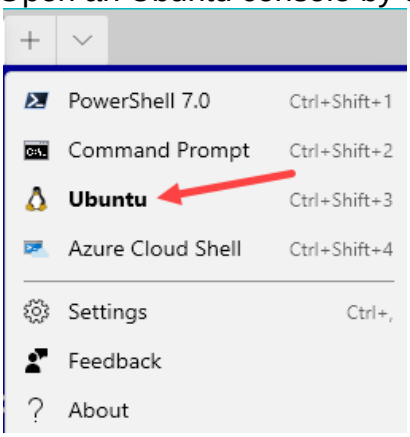
   Restart as needed.

2. Open the Microsoft Store. Download and install the **Windows Terminal** app and **Ubuntu**.

3. After Ubuntu is installed, launch it and configure the username as **cisco** and the password as **cisco**.
4. Close Ubuntu.
5. Open **Windows Terminal**.

   `Start > Windows Terminal or run > wt`

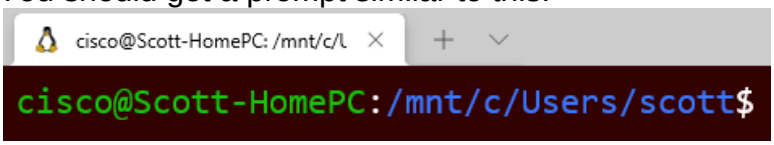
6. Open an Ubuntu console by clicking the dropdown and selecting **Ubuntu**.



   Note: It will take a few moments to install. You may have to restart the Terminal for changes to take effect.

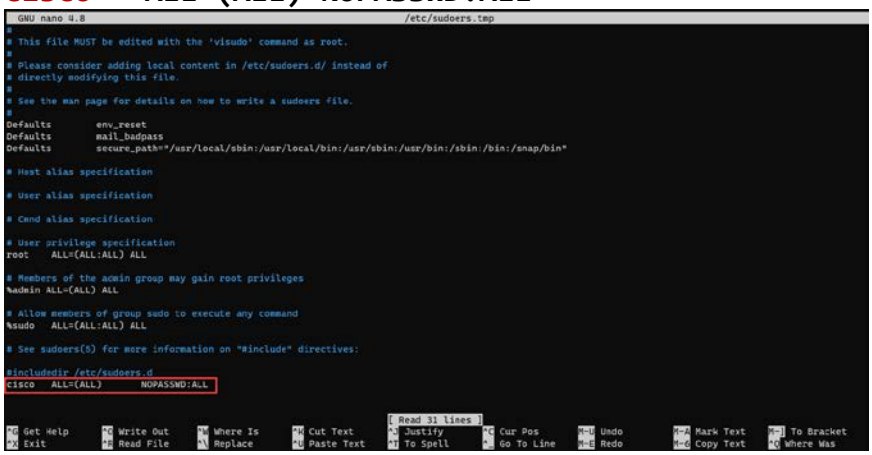
Prerequisites for Network Automation Lab

7. You should get a prompt similar to this:



8. Remove the requirement to enter a password for sudo. Open the /etc/sudoers file and at the end of the file add this line:
   Note: This file is hidden until you add your code.

```
sudo visudo
cisco   ALL=(ALL) NOPASSWD:ALL
```



9. Save, Close, and reopen Windows Terminal.
10. Add user to the sudo group:

```
sudo usermod -aG sudo cisco
```

11. Check the user was added to the group:

```
grep '^sudo' /etc/group
  sudo:x:27:cisco
```

12. Update Ubuntu:

```
sudo apt update && sudo apt upgrade -y
```

13. Check Python and PIP versions:

```
python3 --version
  Python 3.8.10
```

```
pip3 --version
  pip 20.0.2 from /usr/local/lib/python3.8/dist-packages/pip (python 3.8)
```

14. If needed, install Python:

```
sudo apt install python3 -y          (sudo apt reinstall python3)
```

15. If needed, install PiP3:

```
sudo apt install python3-pip -y
```

16. Install Netmiko and Git:

```
sudo apt install python3-netmiko git-all -y
```

17. Install Django, json2html and telnetlib3:

```
pip3 install Django json2html telnetlib3
```

18. You will get a link to the actual lab when the session starts.

VS Code Setup for Python

Visual Studio Code is a free source-code editor developed by Microsoft for Windows, Linux and macOS. Working with Python in Visual Studio Code, using the Microsoft Python extension, is simple, fun, and productive. It leverages all of VS Code's power to provide auto complete and IntelliSense, debugging, and testing, along with the ability to easily switch between Python environments.

Step 1: Setup

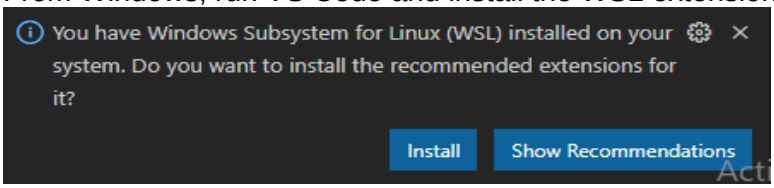
1. To get started, download and install VS Code for your Windows machine:

   https://code.visualstudio.com/download

   Install directly on Ubuntu:
   `sudo snap install --classic code`

   Note: Use the System Installer version for your computer. Do not install the Linux version in WSL.

2. From Windows, run VS Code and install the WSL extensions (Look for this popup):



3. Close and reopen the Ubuntu CLI and run VS Code:

   `code .`

   Note: VS Code will install an extension in Ubuntu.
   Note: The dot (**.**) argument tells VS Code to open the current folder.

Note: If you get a message saying you need to update VS Code and it fails, try removing the Remote – WSL extension:

```
cd ..
cd .vscode/extensions
rm -r ms-vscode-remote.remote-wsl-0.44.2
```

Step 2: Install Extensions

1. Click on the Extensions icon from the left side of the VS Code window.
2. Search extensions to install:

   A. Error Lens
   B. Material Theme
   C. Material Theme Icons
   D. Python
   E. Python Indent
   F. Python Extension Pack
   G. Python for VSCode
   H. Pylance
   I. JSON
   J. YAML
   K. YAML plus JSON
   L. Prettier
   M. Bracket Pair Colorizer 2
   N. Indent-Rainbow

3. Open settings:

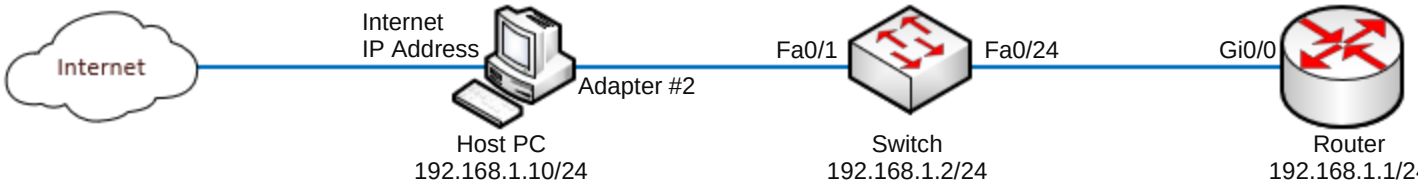
   `Ctrl + ,` or **`command ,`** on Mac

4. Search for these variables:

   A. Check **Editor: Format on Paste**
   B. Check **Editor: Format on Save**
   C. Check **Editor: Format of Type**

Lab Setup

Topology



Equipment

- One Windows 10 PC with two network adapters (one Internet connections; one LAN connection)
- One 2960 Switch running IOS 12.2 or higher with K9 encryption (SSH capable)
- One 1841, 1941, 4000 series or later Router running IOS 12.2 or higher with K9 encryption (SSH capable)
- Three Straight-thru cables
- One Rollover cable for initial configuration

1. You should have two interfaces. One to the internet and one to the practice network. Configure the practice network adaptor with the information below:

   Host PC
   IP address: 192.168.1.10
   SM: 255.255.255.0
   DG: 192.168.1.1

2. From the Ubuntu terminal, check the IP configuration (**ip address**).
3. Each network device needs a basic configuration before starting to gain access over the network. Using a rollover cable and your preferred terminal app (PuTTY, HyperTerminal), configure a basic router and switch with SSH capabilities:
   Note: Only configure these settings (interfaces may vary).

Router

```
enable
configure terminal
hostname R1
enable secret class
username admin password cisco
ip domain name netauto.com
crypto key generate rsa
1024
ip ssh version 2
interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
line con 0
login local
line vty 0
login local
transport input ssh
end
terminal length 0
copy run start
```

Switch

```
enable
configure terminal
hostname S1
enable secret class
username admin password cisco
ip domain-name netauto.com
crypto key generate rsa
1024
ip ssh version 2
interface vlan 1
ip address 192.168.1.2 255.255.255.0
no shutdown
exit
ip default-gateway 192.168.1.1
line con 0
login local
line vty 0
login local
transport input ssh
end
terminal length 0
copy run start
```

4. Be sure you can ping between all devices. Troubleshoot as needed.
5. Add other devices as needed.