Network Management and Automation

Lab 7

DevOps-Ansible, Automation, and Networking

University of Colorado Boulder

Network Engineering Program

Professor Levi Perigo, Ph.D.

# Summary

Automation is an important component to achieve new data center deployments in short time intervals. Data centers require similar type devices to be configured multiple numbers of times and automation eliminates human intervention process of logging into devices using a console for configuration purpose.

# Objectives

* Create Ansible playbooks using a Python script based on your network topology and the router configuration requirements.
* Create Jinja2 templates based on your final router configuration requirements.
* Create a final router configuration file using Ansible and a Jinja2 template.
* Learn to use Ansible, Jinja2, and Python to automate network deployments and device configurations.

Problem Statement:

In your previous labs, you worked on automating the complete network in your data center using Netmiko and you have also worked using Ansible for configuration and package management. In this lab, you will automate new routers deployment in your data center network using Python, Ansible playbooks, and Jinja2 templates.

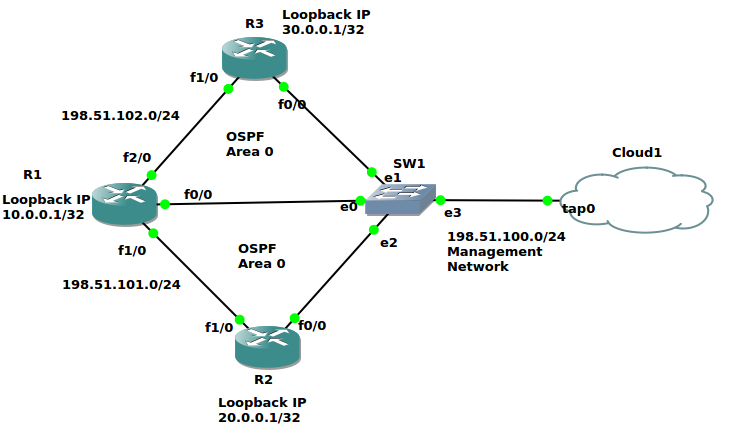


Figure 1

Guidelines:

Create the above-mentioned topology in GNS3 on the NetMan VM. Management network and SSH should be configured/enabled on R1, R2, and R3. Do not configure anything else manually.

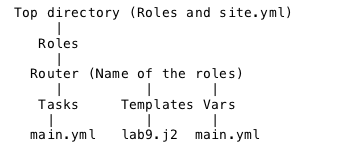
Ansible and Jinja2 Help:

1. https://pynet.twb-tech.com/blog/ansible/ansible-cfg-template.html

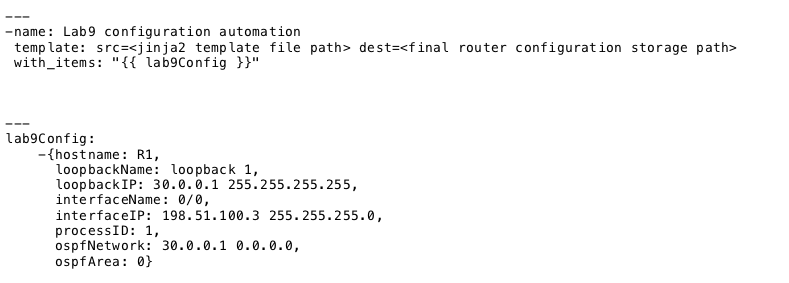
Ansible Example:

Use Ansible Roles directory as the main directory containing Tasks, Templates, and Vars as subdirectories to deploy your automation. Refer to the above Ansible automation reference link for guidance. The below Ansible playbook examples are only for your reference and your Ansible playbooks generated using Python will not be exactly similar to these, they will contain additional elements according to the configuration requirements.

The following is the directory structure required. Follow the above reference link for more information.

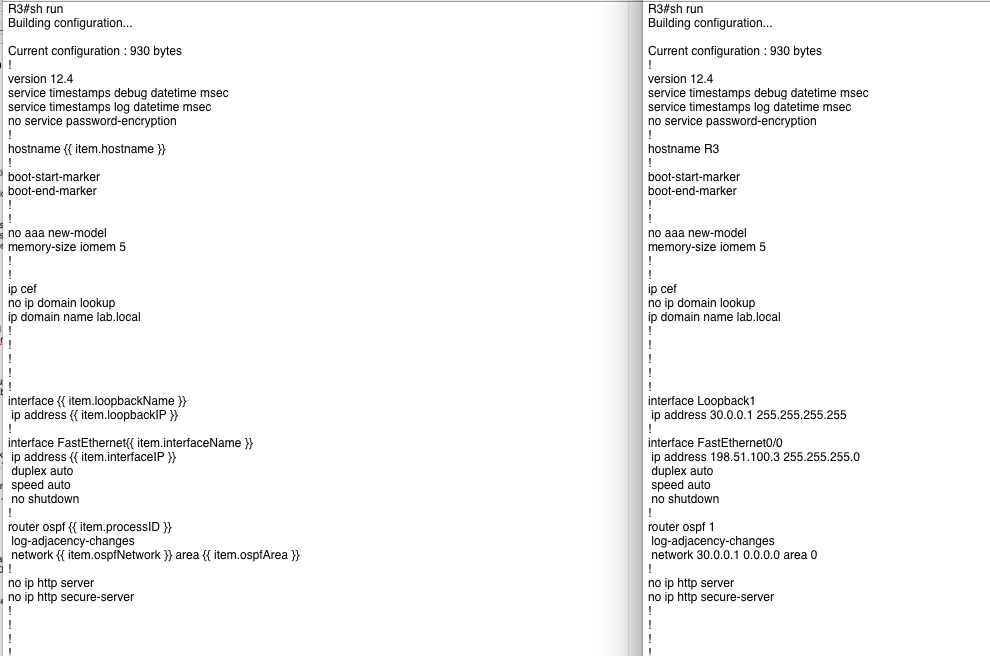


The following are two separate example YAML files. When the first file is executed, it will call the second variables containing file (created using Python). Follow the above reference link for more information.



Jinja2 Example:

The below Jinja2 template example (lab9.j2) is only for your reference and your Jinja2 template will not be exactly similar to this, it will consist of additional elements according to the configuration requirements.



Lab configuration requirements:

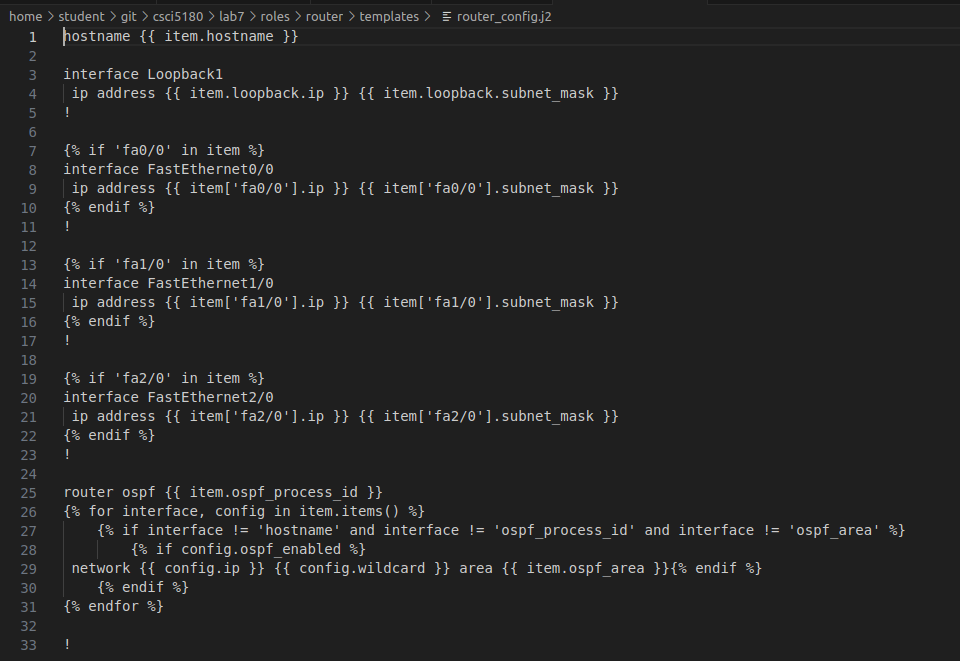
| Hostname | Interface Type | Interface Name | IP/Subnet | OSPF Enabled | OSPF Process ID | OSPF Area |
| --- | --- | --- | --- | --- | --- | --- |
| R1 | Loopback | 1 | 10.0.0.1/32 | Yes | 1 | 0 |
| R1 | FastEthernet | 0/0 | 198.51.100.3/24 | Yes | 1 | 0 |
| R1 | FastEthernet | 1/0 | 198.51.101.3/24 | Yes | 1 | 0 |
| R1 | FastEthernet | 2/0 | 198.51.102.3/24 | Yes | 1 | 0 |
| R2 | Loopback | 1 | 20.0.0.1/32 | Yes | 2 | 0 |
| R2 | FastEthernet | 0/0 | 198.51.100.4/24 | Yes | 2 | 0 |
| R2 | FastEthernet | 1/0 | 198.51.101.4/24 | Yes | 2 | 0 |
| R3 | Loopback | 1 | 30.0.0.1/32 | Yes | 3 | 0 |
| R3 | FastEthernet | 0/0 | 198.51.100.5/24 | Yes | 3 | 0 |
| R3 | FastEthernet | 1/0 | 198.51.102.5/24 | Yes | 3 | 0 |

Objective 1: Using the given topology, create a Jinja2 template for the Cisco routers (R1, R2, and R3). Do not configure the Cisco routers in GNS3 in the NetMan VM manually. **[50 Points]**

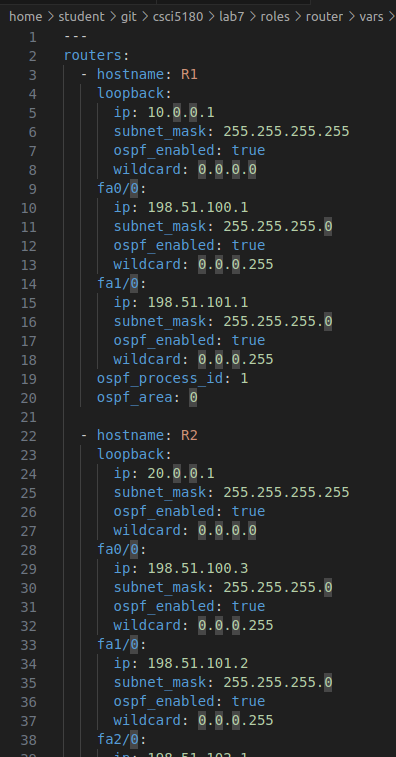
I have to following directories in my current working directory:

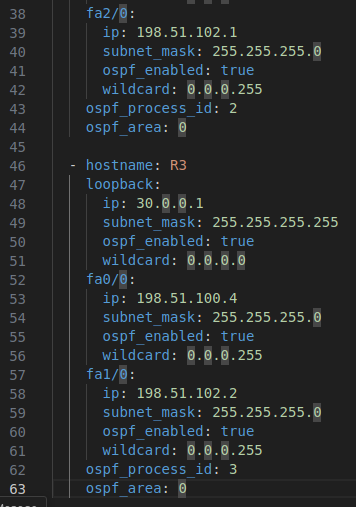
| ./roles ./roles/router ./roles/router/templates ./roles/router/tasks ./roles/router/vars ./cfgs |
| --- |

I created a **router\_config.j2** file under **roles/router/templates**:



Which picks up the variables from **roles/router/vars/main.yaml** (*This is generated by my python script as described in Objective 2*):

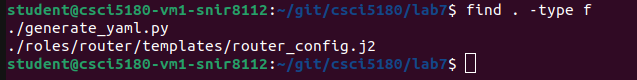


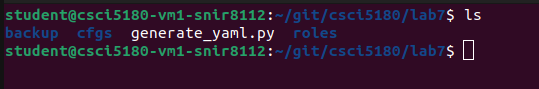


Objective 2: Using the configuration requirements (CSV), create an Ansible playbook in YAML format with the help of a Python script. Follow the Ansible “Roles” directory framework for this objective. This Ansible playbook and Jinja2 templates for R1, R2, and R3 will be used to create the actual router configuration files to configure the above topology in GNS3 in your NetMan VM. **[50 Points]**

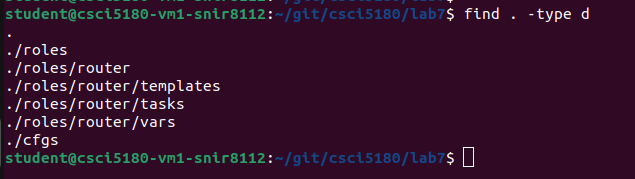
**2.1 Screenshots of the output:**

As of now, I only have the jinja2 template (*roles/router/templates/router\_config.j2*) and my python script in the folder (*screenshot of the python code in* ***Point******2.2***):



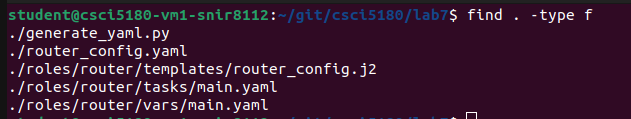


Here’s my directory structure:

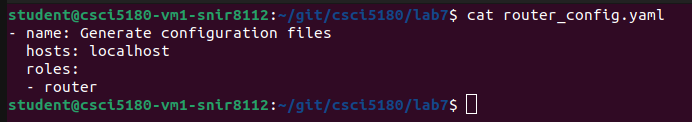


After running my **generate\_yaml.py** python code, it creates 3 YAML files:

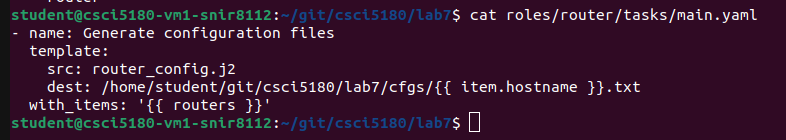
1. **router\_config.yaml** in the current working directory (*which would be used by ansible*),
2. **main.yaml** under **roles/router/tasks** directory which would call the router\_config.j2 Jinja2 template, and
3. **main.yaml** under **roles/router/vars** directory which defines all the variables used by the tasks to call the jinja2 template.



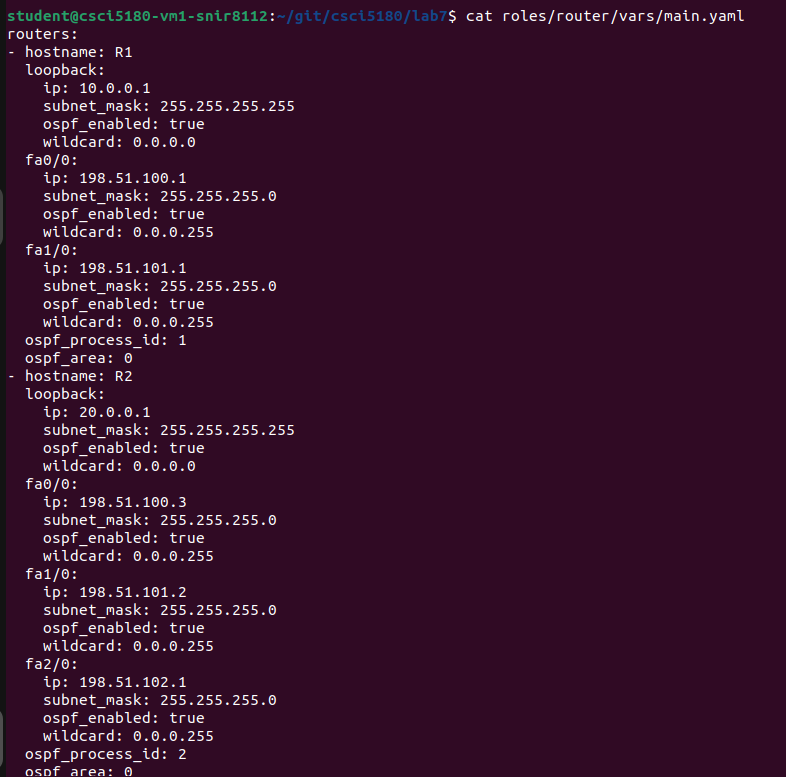
Screenshot of **router\_config.yaml**:

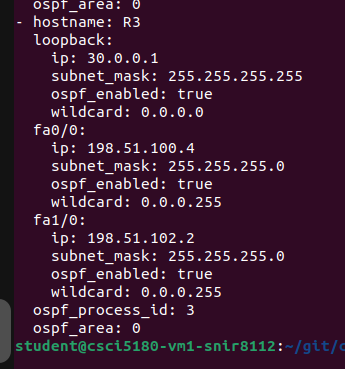


Screenshot of **roles/router/tasks/main.yaml**:

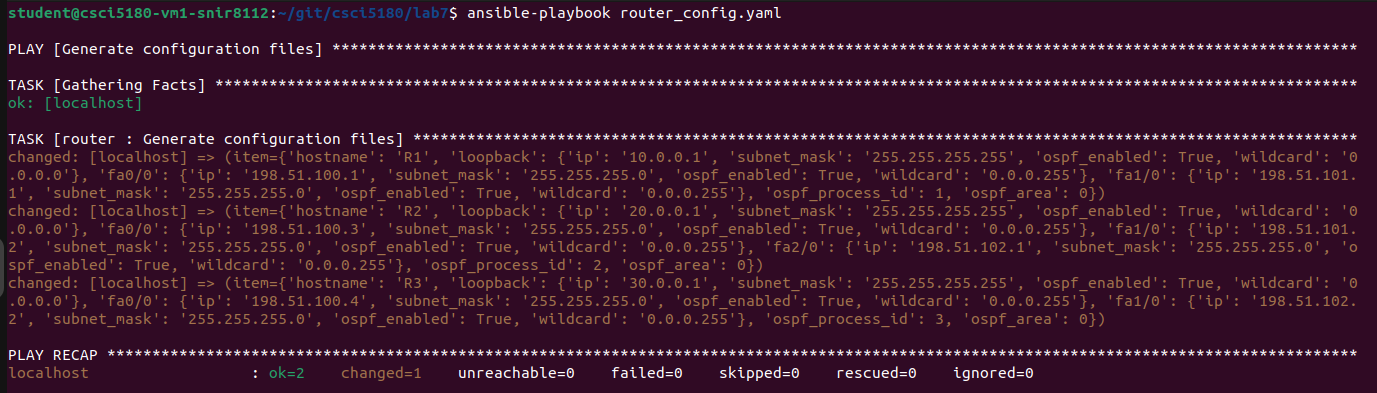


Screenshot of **roles/router/vars/main.yaml**:

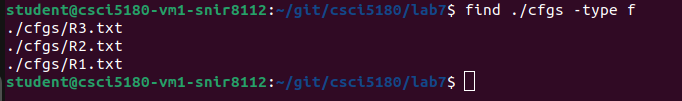




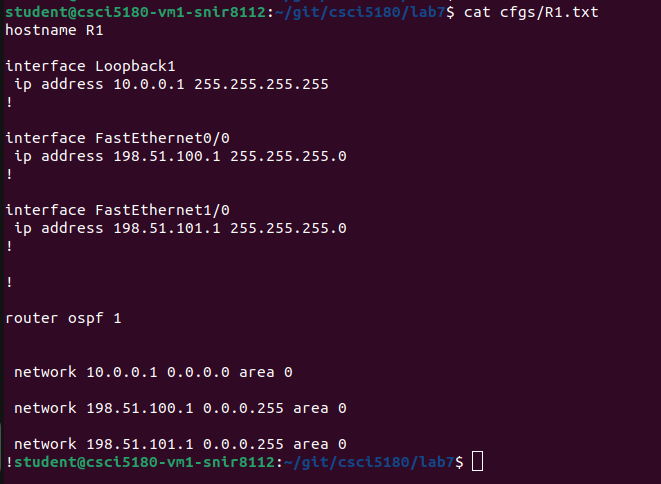
Running the **router\_config.yaml** file through Ansible to generate the configuration for R1, R2 and R3:

****

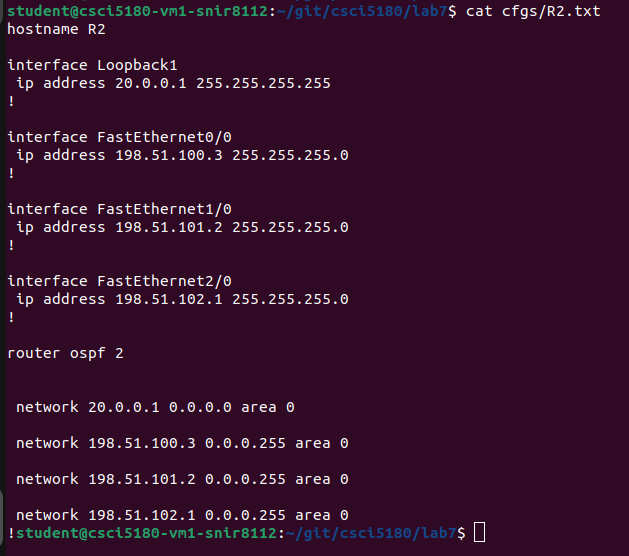
Post this, it generated 3 text files: *R1.txt, R2.txt, R3.txt* which contains the configuration to be pushed to each of these devices.



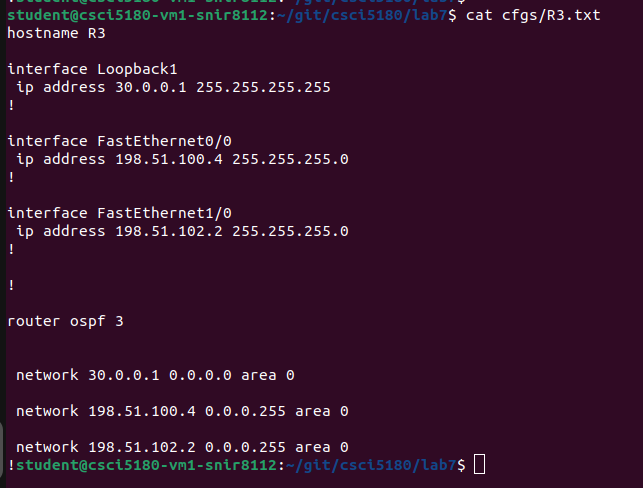
**cfgs/R1.txt**:



**cfgs/R2.txt:**

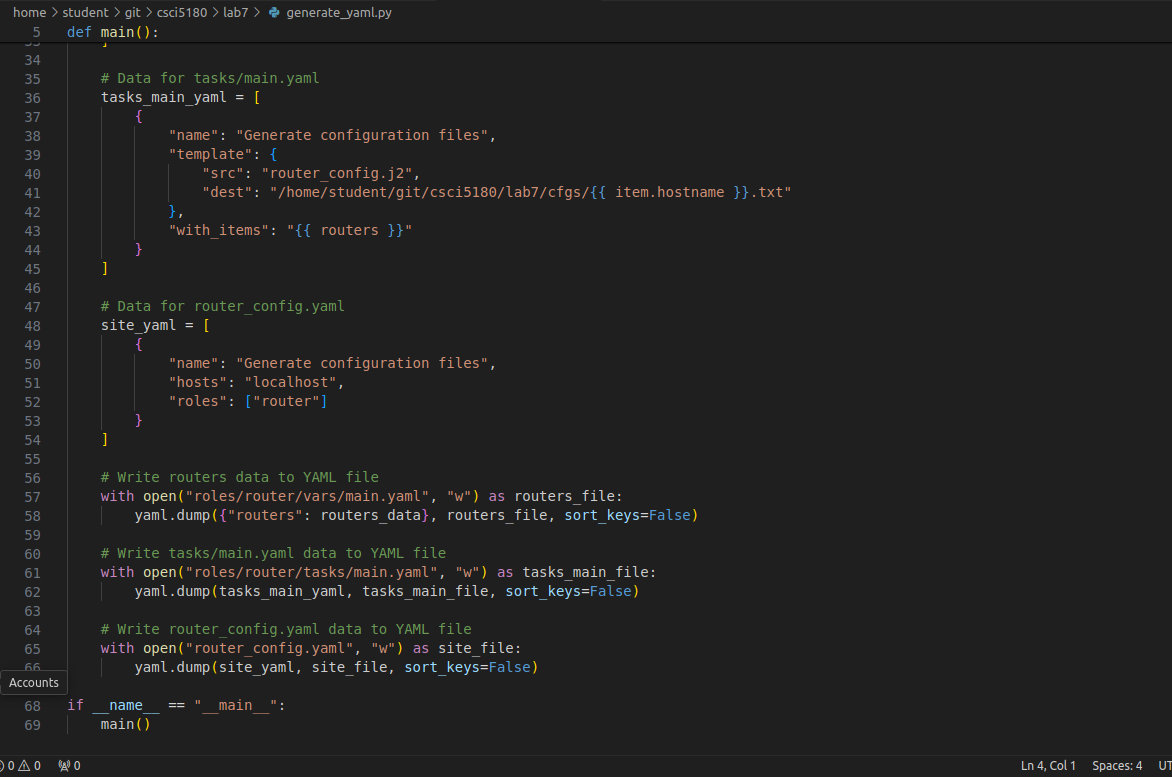


**cfgs/R3.txt:**



**2.2 Screenshot of the generate\_yaml.py python code:**

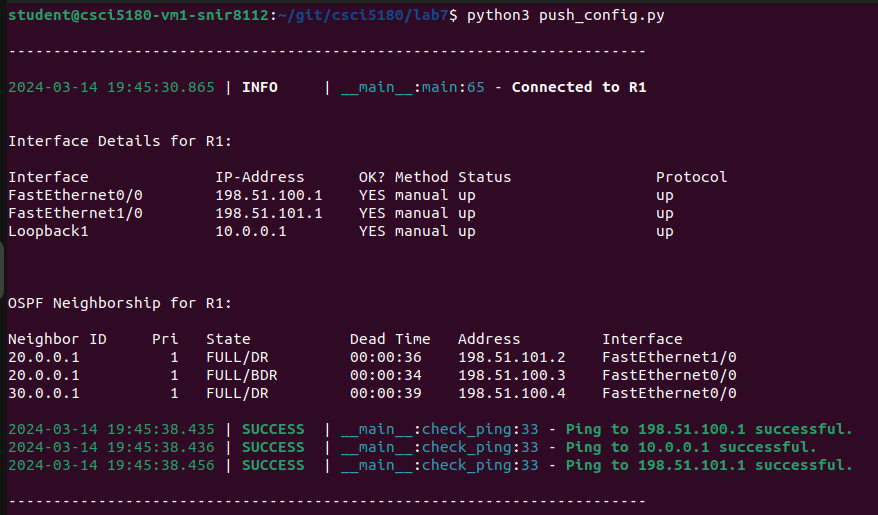




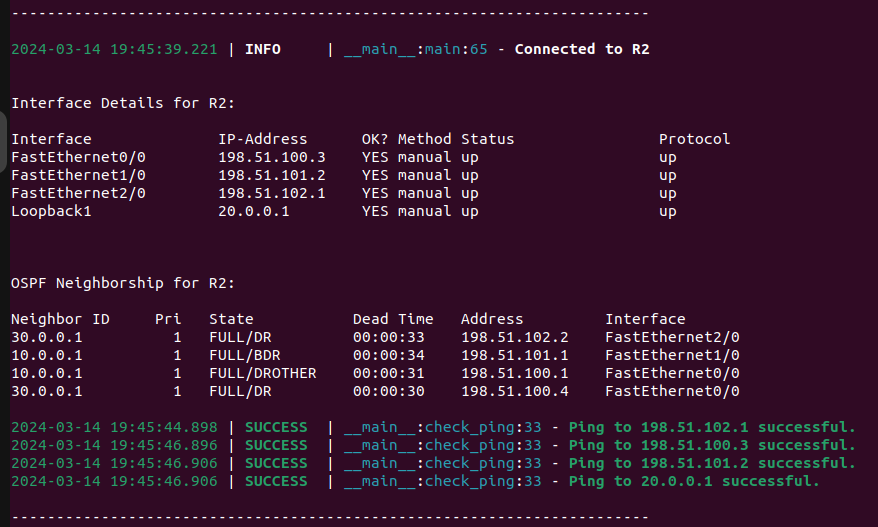
Objective 3: Configure R1, R2, and R3 in GNS3 in your NetMan VM using Netmiko and achieve complete reachability to the real and loopback interfaces of R1, R2, and R3 from your NetMan VM terminal. **[50 Points]**

**Output of my python script**

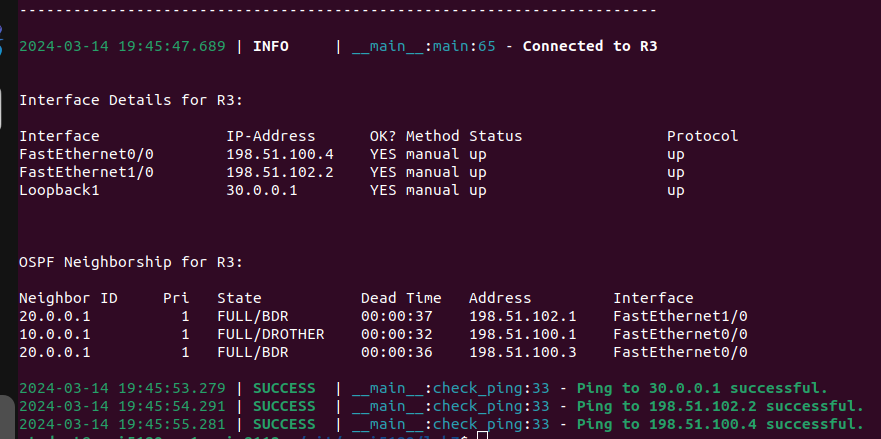
**R1**



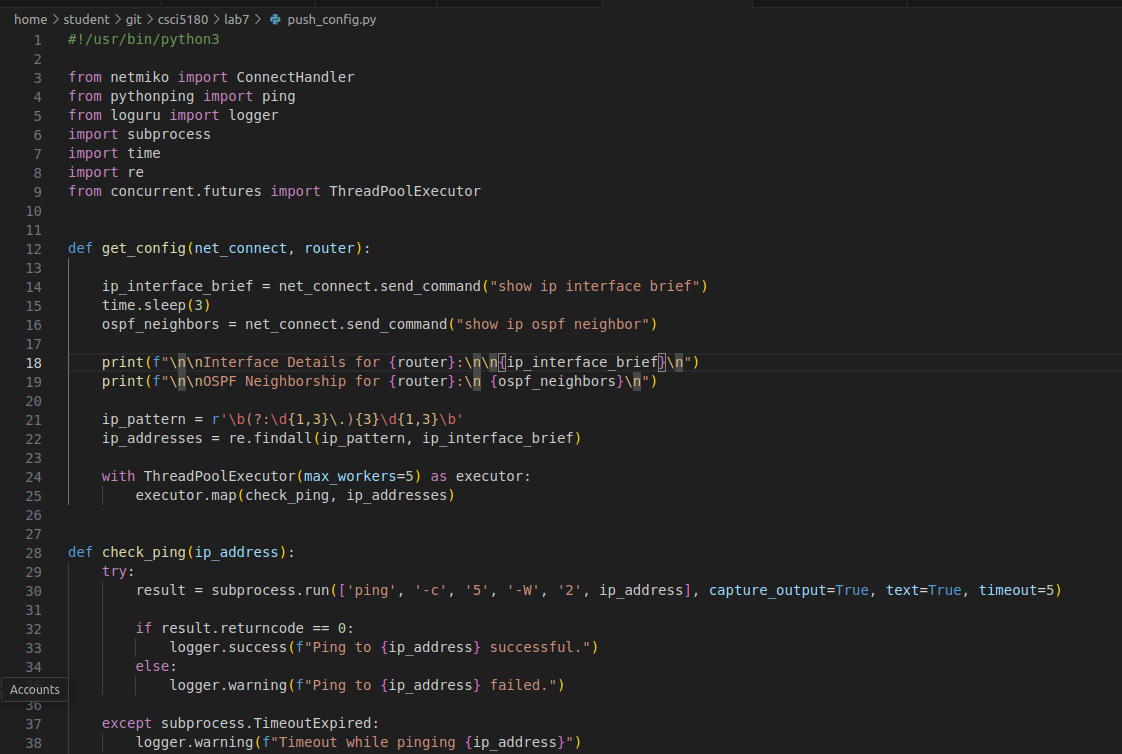
**R2**

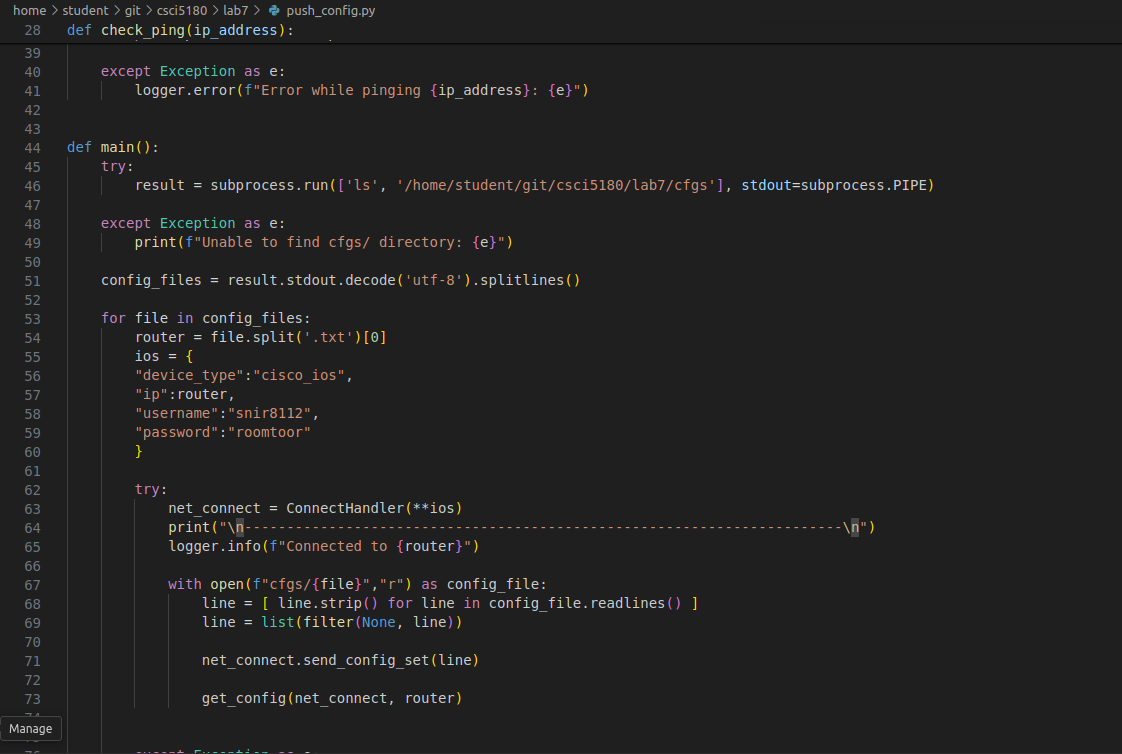


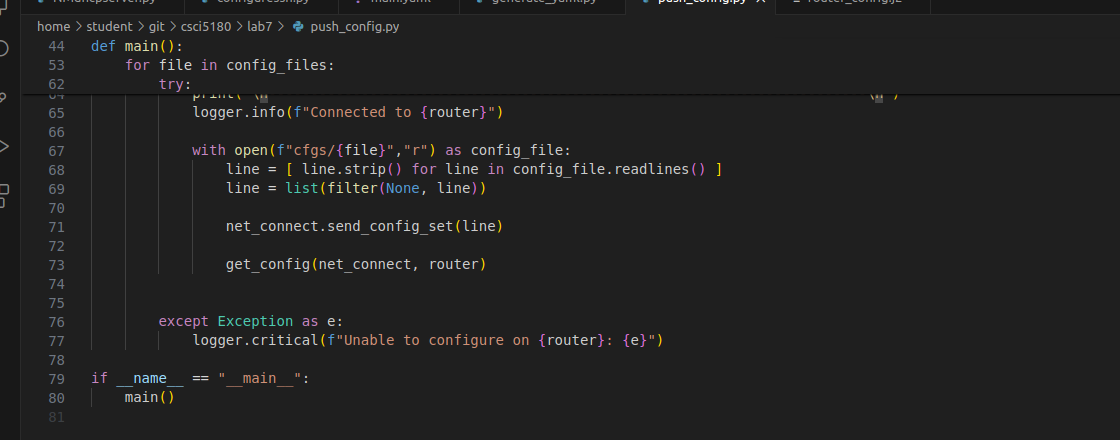
**R3**

****

**Screenshot of push\_config.py code**







Extra Credit I: Configure R1, R2, and R3 by pulling the new configuration file generated in Objective 2 using ZTP. Configure a DHCP server to also act as a TFTP server on your NetMan Ubuntu machine and configure R1, R2, and R3 as DHCP clients to request IP address and the new configuration file from Objective 2. Use below links for reference and also search the Internet for more help and reference. **[25 Points]**

**Use below links for ZTP reference,**

1. <https://www.digitalocean.com/blog/zero-touch-provisioning-how-to-build-a-network-without-touching-anything>
2. <https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-5/configuration_guide/prog/b_165_prog_9500_cg/zero-touch_provisioning.pdf>

Report Questions:

1. After completing this lab, explain ZTP and when can ZTP be implemented in the real world with an example? **[10 points]**

Total Points \_\_\_\_\_\_\_\_\_\_\_\_ / 160 (+25 extra credit)