Case Study for DevOps

Network Automation and Programmability

Intended Learning Outcomes:

- Configure and build a network using RIP routing protocol.
- Configure and implement OSPF routing protocol using Ansible in Cisco IOS device
- Configure OSPF routing protocol with NETCONF using python ncclient in Cisco IOS-XE device

Resources:

- GNS3
- Virtual Box
- DEVASC-LABVM virtual machine
- CISCO 3750 IOS Image

Procedures:

Part 1: Network Automation using Ansible in CISCO IOS

1. Build the given topology using GNS3 and 3750 routers

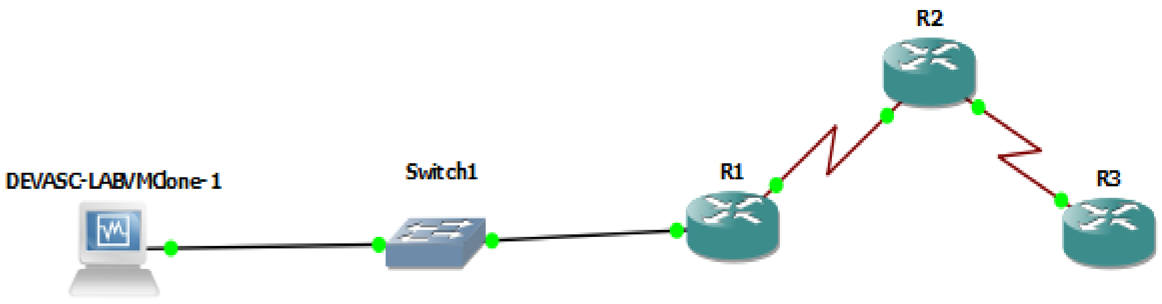
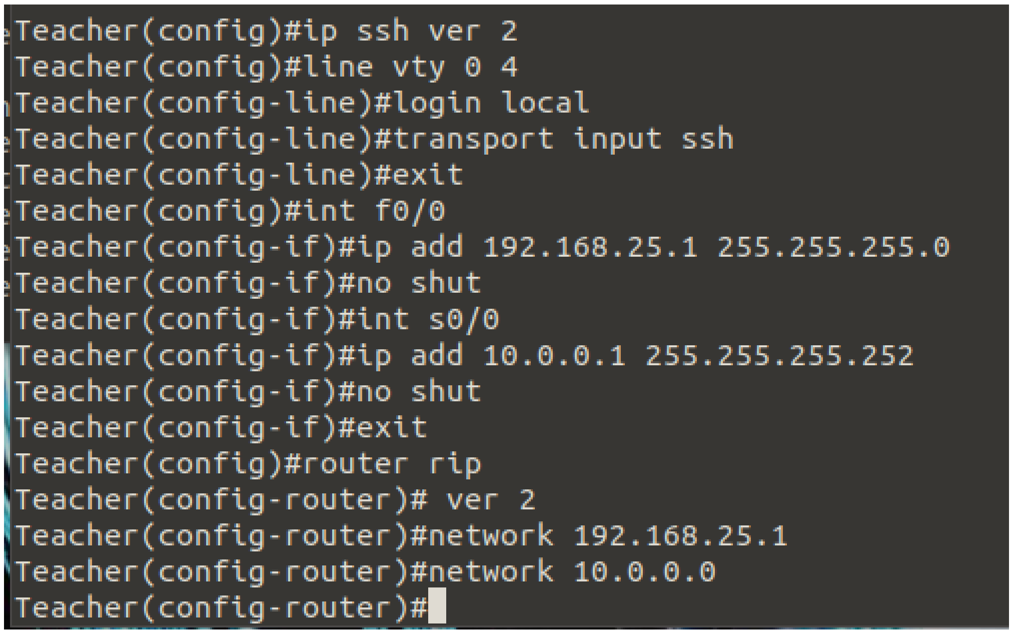


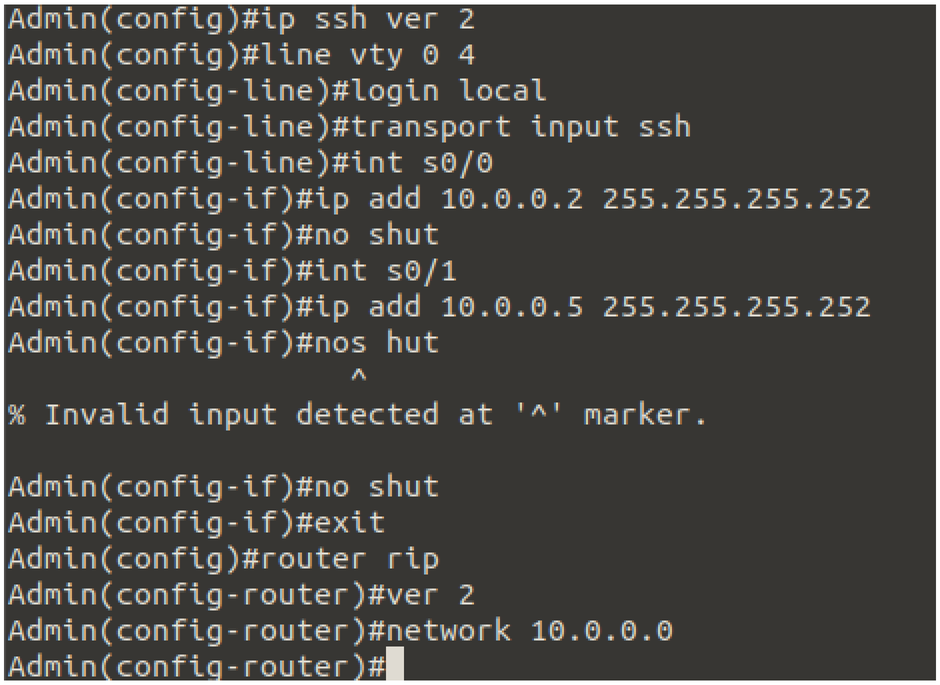
Figure 1. shows the built topology using GNS3 and 3750 routers.

2. Configure the following to the routers:
   a. Remote access using SSH, IP addressing, and RIP version 2 for Router 1(Teacher)



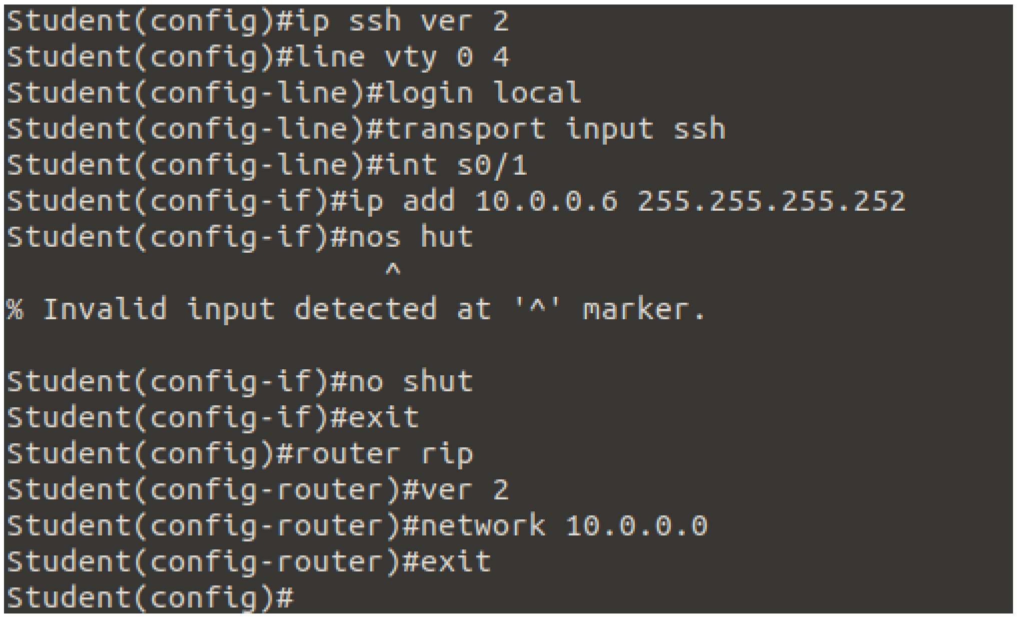
```
Teacher(config)#ip ssh ver 2
Teacher(config)#line vty 0 4
Teacher(config-line)#login local
Teacher(config-line)#transport input ssh
Teacher(config-line)#exit
Teacher(config)#int f0/0
Teacher(config-if)#ip add 192.168.25.1 255.255.255.0
Teacher(config-if)#no shut
Teacher(config-if)#int s0/0
Teacher(config-if)#ip add 10.0.0.1 255.255.255.252
Teacher(config-if)#no shut
Teacher(config-if)#exit
Teacher(config)#router rip
Teacher(config-router)# ver 2
Teacher(config-router)#network 192.168.25.1
Teacher(config-router)#network 10.0.0.0
Teacher(config-router)#
```

b. Remote access using SSH, IP addressing, and RIP version 2 for Router 2(Admin)



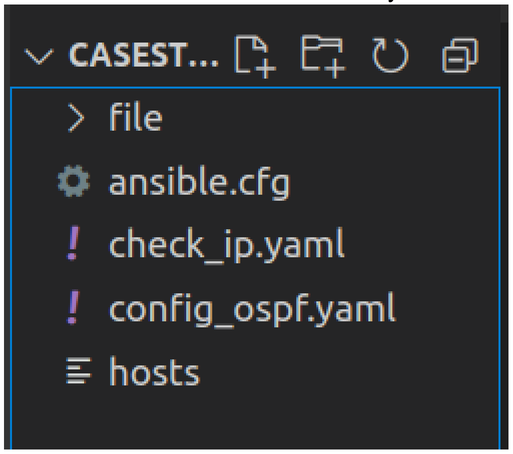
```
Admin(config)#ip ssh ver 2
Admin(config)#line vty 0 4
Admin(config-line)#login local
Admin(config-line)#transport input ssh
Admin(config-line)#int s0/0
Admin(config-if)#ip add 10.0.0.2 255.255.255.252
Admin(config-if)#no shut
Admin(config-if)#int s0/1
Admin(config-if)#ip add 10.0.0.5 255.255.255.252
Admin(config-if)#nos hut
                   ^
% Invalid input detected at '^' marker.

Admin(config-if)#no shut
Admin(config-if)#exit
Admin(config)#router rip
Admin(config-router)#ver 2
Admin(config-router)#network 10.0.0.0
Admin(config-router)#
```

c. Remote access using SSH, IP addressing, and RIP version 2 for Router 3(Student)

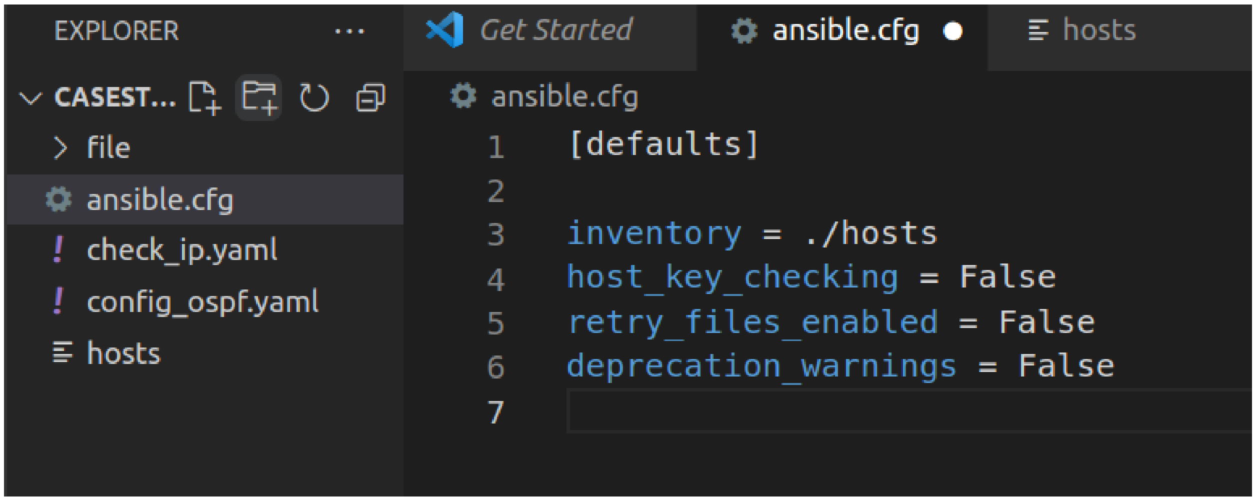


```
Student(config)#ip ssh ver 2
Student(config)#line vty 0 4
Student(config-line)#login local
Student(config-line)#transport input ssh
Student(config-line)#int s0/1
Student(config-if)#ip add 10.0.0.6 255.255.255.252
Student(config-if)#nos hut
                     ^
% Invalid input detected at '^' marker.

Student(config-if)#no shut
Student(config-if)#exit
Student(config)#router rip
Student(config-router)#ver 2
Student(config-router)#network 10.0.0.0
Student(config-router)#exit
Student(config)#
```

3. Click the simulate button in GNS3 to start the network simulation.
4. Create a folder Casestudy in the /labs/devscr/ansible in DEVASC-LABVM
5. Create file folder in the Casestudy folder.

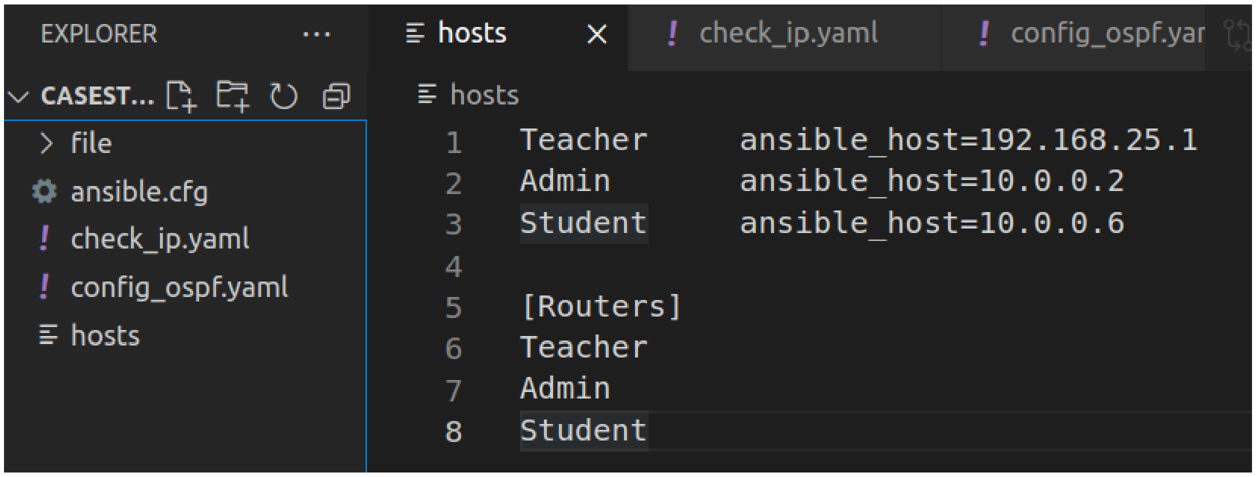


6. Create the following file to configure an OSPF routing protocol with process id of 100 using Ansible
   a. Create ansible.cfg file



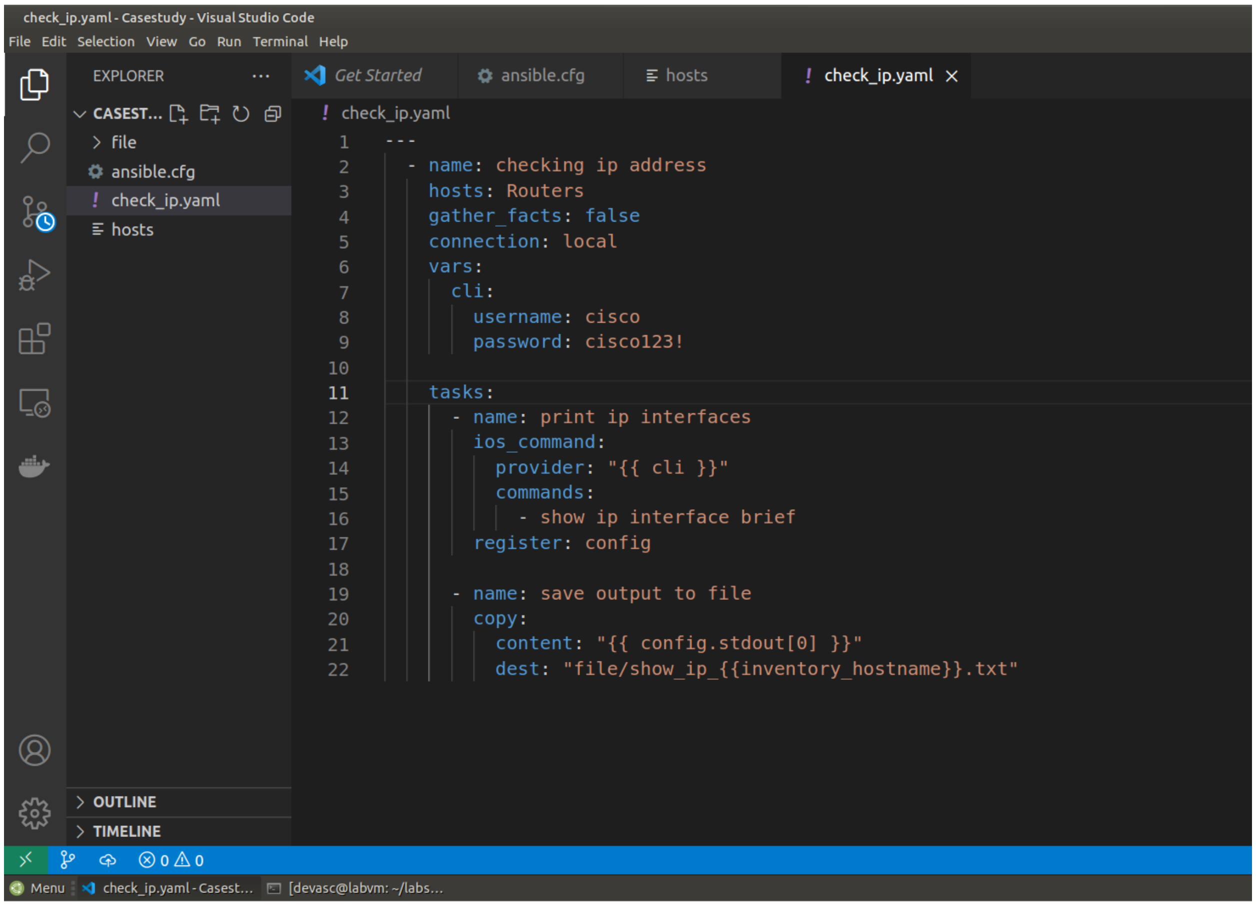
```
[defaults]

inventory = ./hosts
host_key_checking = False
retry_files_enabled = False
deprecation_warnings = False
```

b. Create hosts file



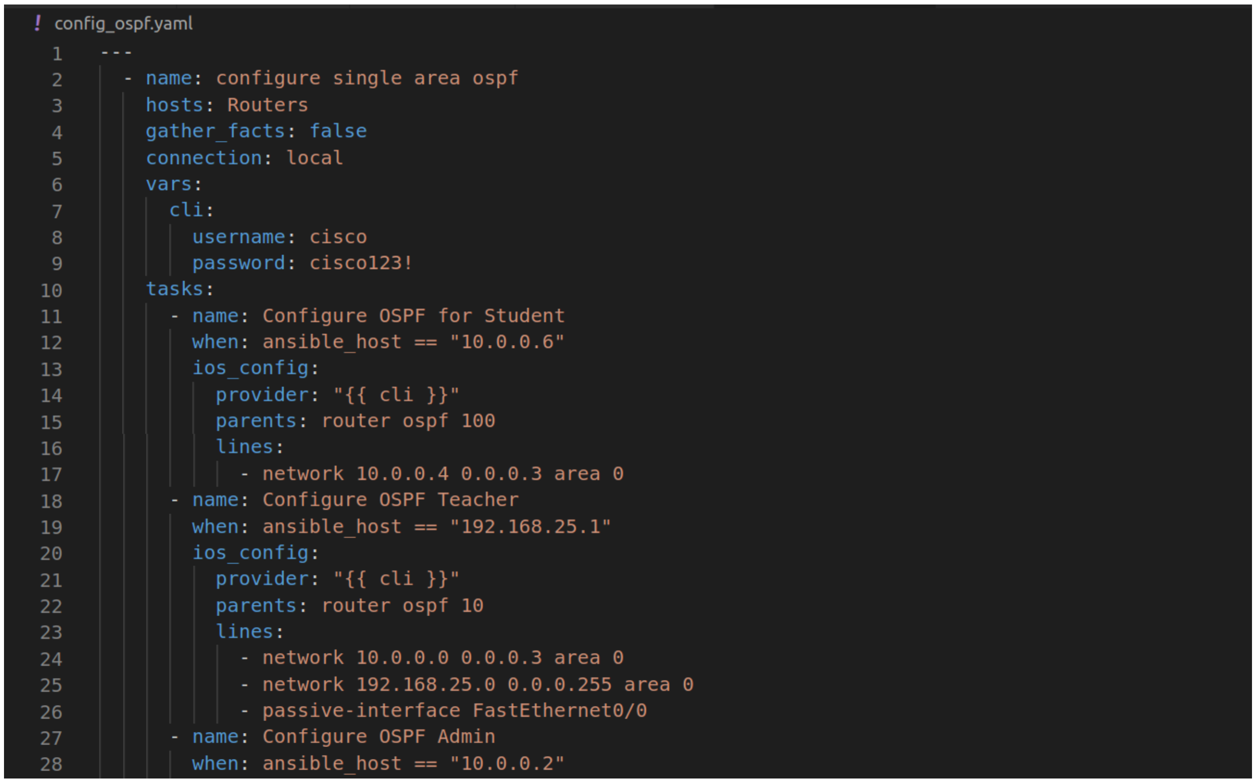
```
hosts
Teacher     ansible_host=192.168.25.1
Admin       ansible_host=10.0.0.2
Student     ansible_host=10.0.0.6

[Routers]
Teacher
Admin
Student
```

c. Create check_ip.yaml



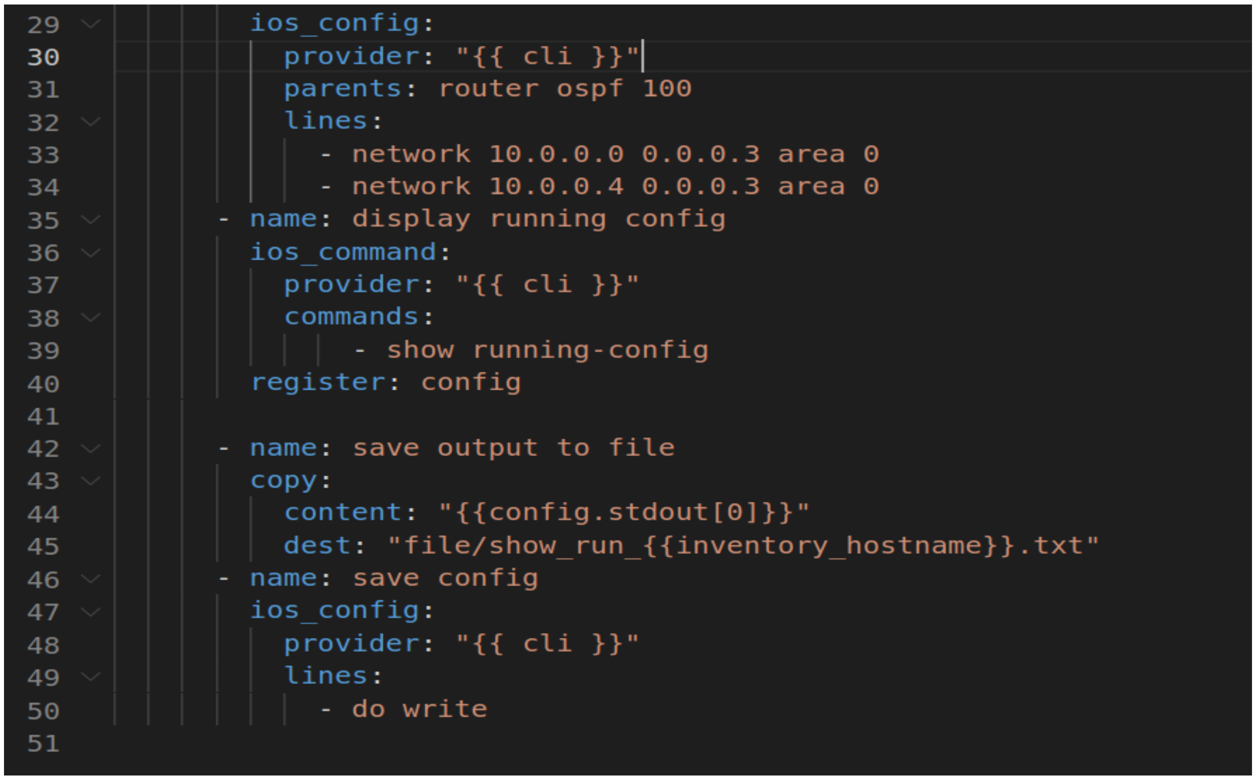
```
---
- name: checking ip address
  hosts: Routers
  gather_facts: false
  connection: local
  vars:
    cli:
      username: cisco
      password: cisco123!

  tasks:
    - name: print ip interfaces
      ios_command:
        provider: "{{ cli }}"
        commands:
          - show ip interface brief
      register: config

    - name: save output to file
      copy:
        content: "{{ config.stdout[0] }}"
        dest: "file/show_ip_{{inventory_hostname}}.txt"
```

d. Create configure_ospf.yaml



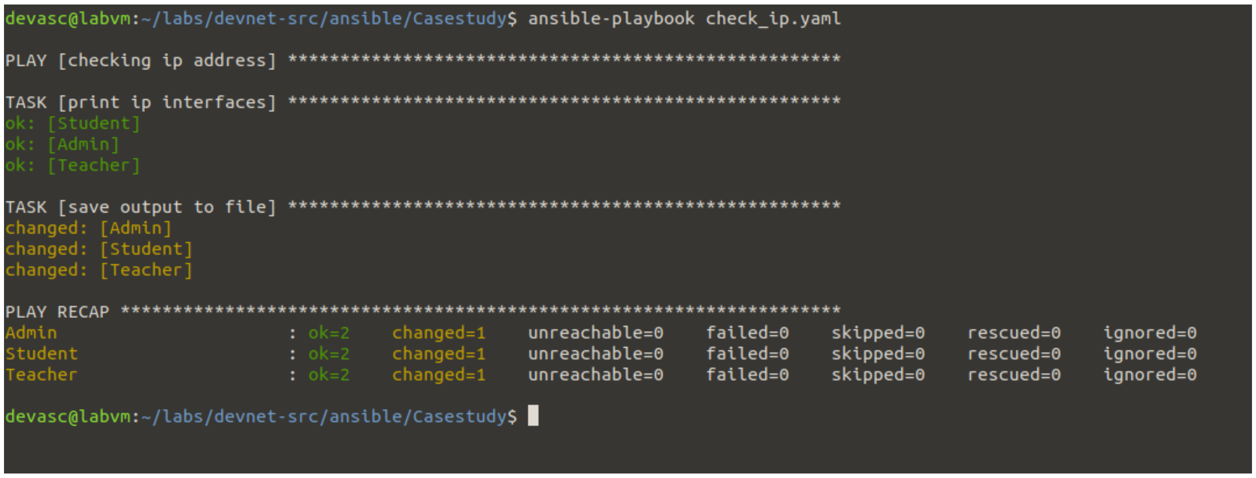
config_ospf.yaml

```yaml
---
- name: configure single area ospf
  hosts: Routers
  gather_facts: false
  connection: local
  vars:
    cli:
      username: cisco
      password: cisco123!
  tasks:
    - name: Configure OSPF for Student
      when: ansible_host == "10.0.0.6"
      ios_config:
        provider: "{{ cli }}"
        parents: router ospf 100
        lines:
          - network 10.0.0.4 0.0.0.3 area 0
    - name: Configure OSPF Teacher
      when: ansible_host == "192.168.25.1"
      ios_config:
        provider: "{{ cli }}"
        parents: router ospf 10
        lines:
          - network 10.0.0.0 0.0.0.3 area 0
          - network 192.168.25.0 0.0.0.255 area 0
          - passive-interface FastEthernet0/0
    - name: Configure OSPF Admin
      when: ansible_host == "10.0.0.2"
```

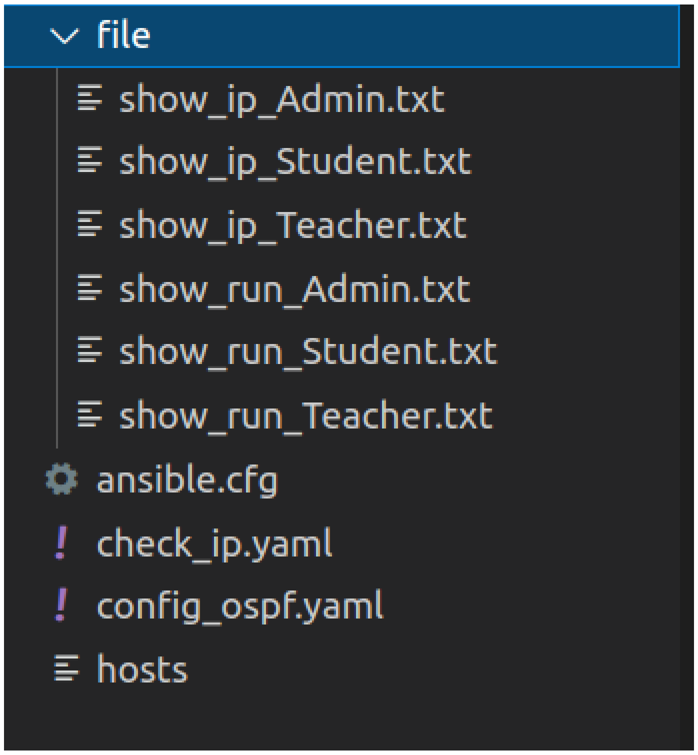


```yaml
      ios_config:
        provider: "{{ cli }}"
        parents: router ospf 100
        lines:
          - network 10.0.0.0 0.0.0.3 area 0
          - network 10.0.0.4 0.0.0.3 area 0
    - name: display running config
      ios_command:
        provider: "{{ cli }}"
        commands:
            - show running-config
      register: config

    - name: save output to file
      copy:
        content: "{{config.stdout[0]}}"
        dest: "file/show_run_{{inventory_hostname}}.txt"
    - name: save config
      ios_config:
        provider: "{{ cli }}"
        lines:
          - do write
```

7. Run the check_ip.yaml and check the output in the file folder.

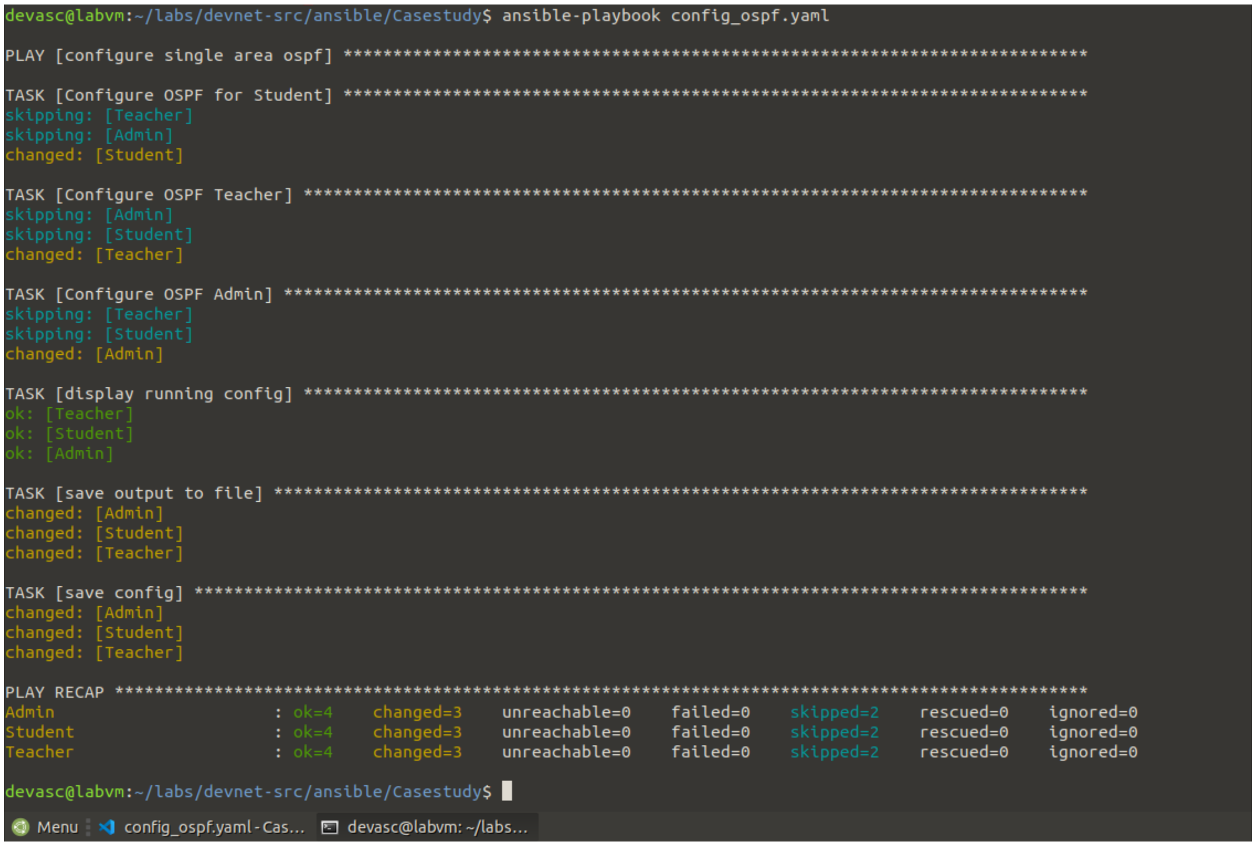


```
devasc@labvm:~/labs/devnet-src/ansible/Casestudy$ ansible-playbook check_ip.yaml

PLAY [checking ip address] *****************************************************

TASK [print ip interfaces] *****************************************************
ok: [Student]
ok: [Admin]
ok: [Teacher]

TASK [save output to file] *****************************************************
changed: [Admin]
changed: [Student]
changed: [Teacher]

PLAY RECAP *********************************************************************
Admin                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Student                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Teacher                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

devasc@labvm:~/labs/devnet-src/ansible/Casestudy$
```

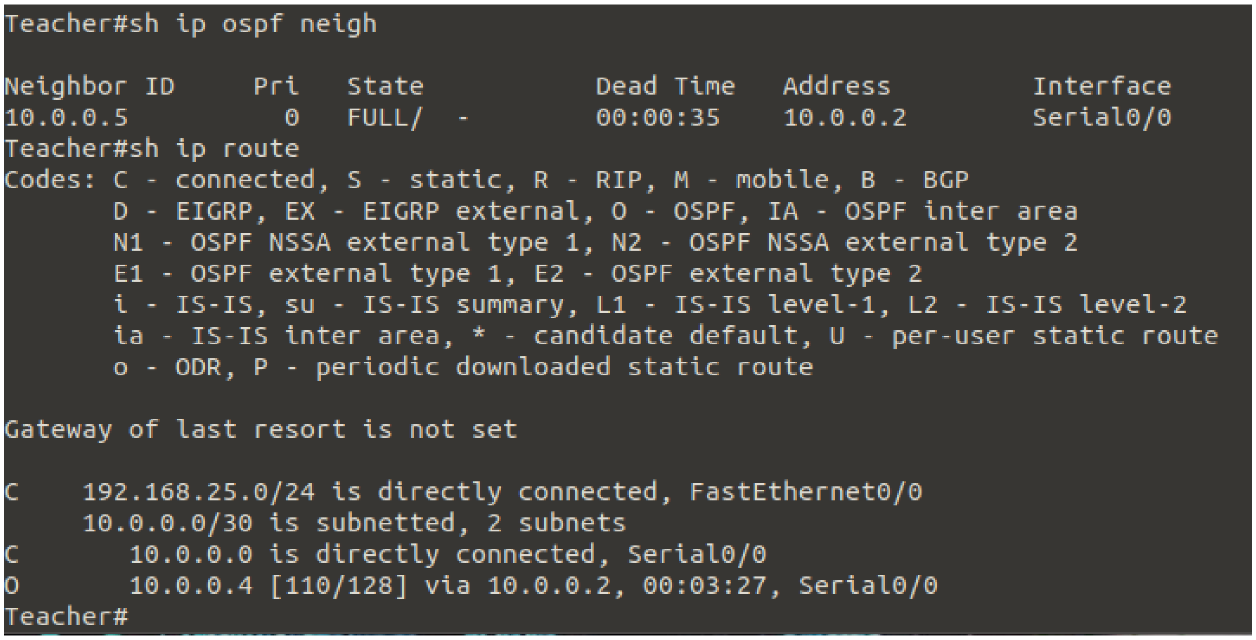


```
file
    show_ip_Admin.txt
    show_ip_Student.txt
    show_ip_Teacher.txt
    show_run_Admin.txt
    show_run_Student.txt
    show_run_Teacher.txt
ansible.cfg
check_ip.yaml
config_ospf.yaml
hosts
```

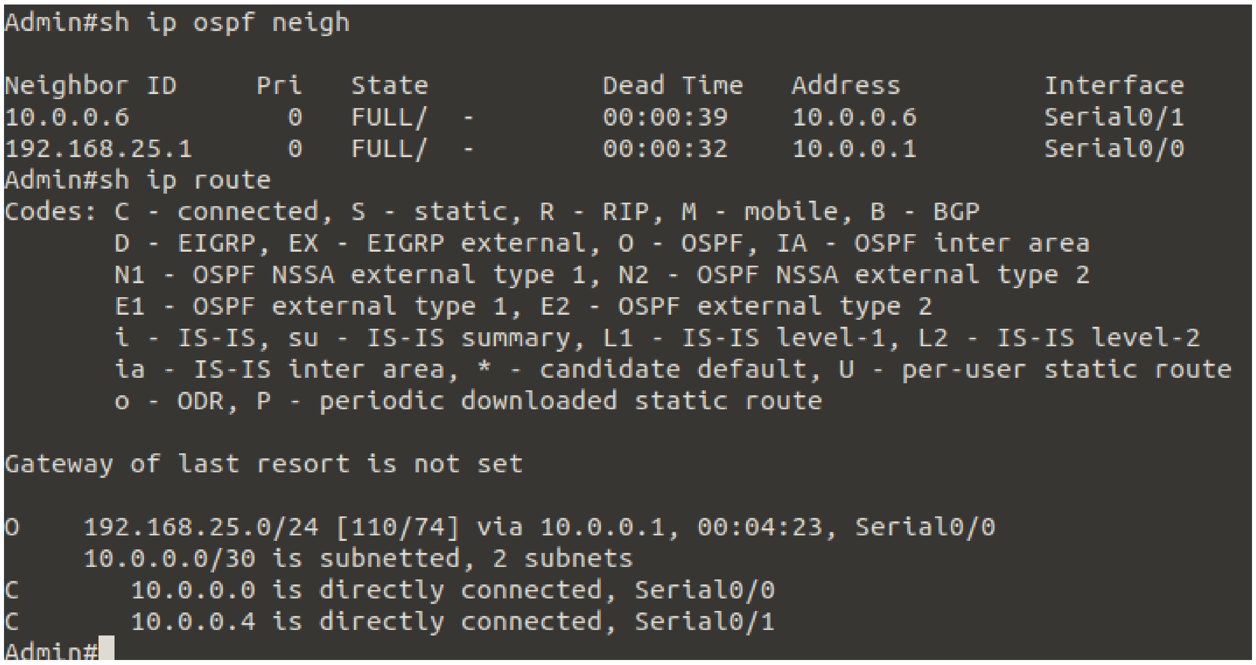
8. Run the configure_ospf.yaml and check the ospf configuration in the Teacher, Admin, Student routers.



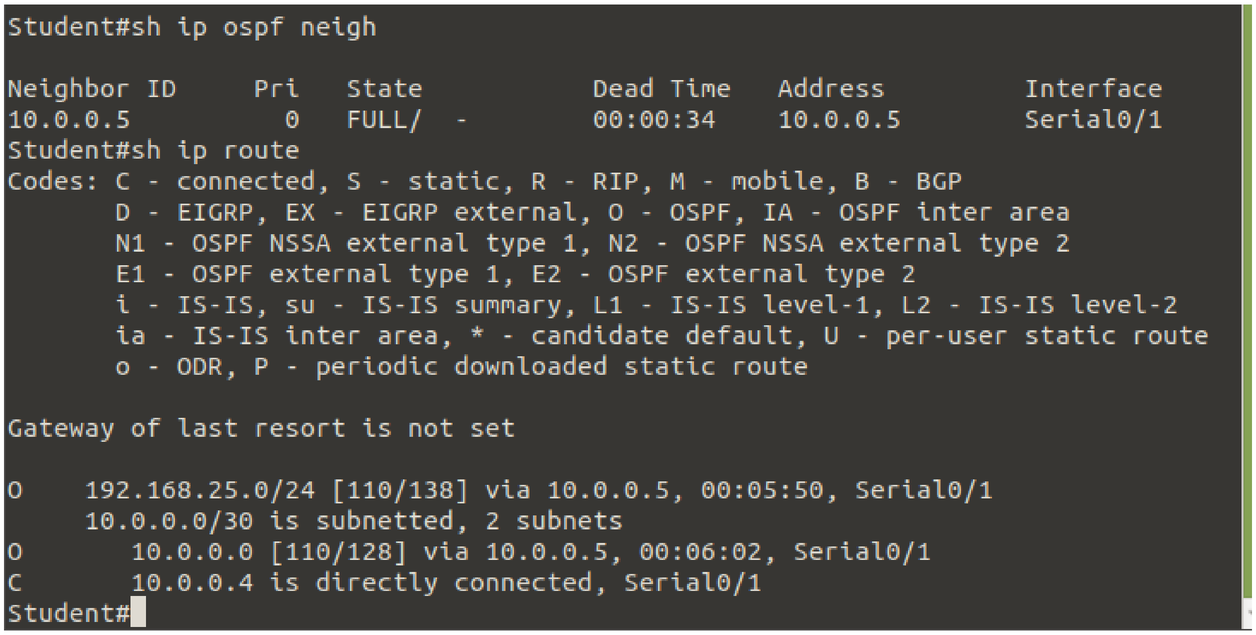
```
devasc@labvm:~/labs/devnet-src/ansible/Casestudy$ ansible-playbook config_ospf.yaml

PLAY [configure single area ospf] **********************************************************************

TASK [Configure OSPF for Student] **********************************************************************
skipping: [Teacher]
skipping: [Admin]
changed: [Student]

TASK [Configure OSPF Teacher] **************************************************************************
skipping: [Admin]
skipping: [Student]
changed: [Teacher]

TASK [Configure OSPF Admin] ****************************************************************************
skipping: [Teacher]
skipping: [Student]
changed: [Admin]

TASK [display running config] **************************************************************************
ok: [Teacher]
ok: [Student]
ok: [Admin]

TASK [save output to file] *****************************************************************************
changed: [Admin]
changed: [Student]
changed: [Teacher]

TASK [save config] *************************************************************************************
changed: [Admin]
changed: [Student]
changed: [Teacher]

PLAY RECAP *********************************************************************************************
Admin                      : ok=4    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
Student                    : ok=4    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
Teacher                    : ok=4    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

devasc@labvm:~/labs/devnet-src/ansible/Casestudy$
```



```
Teacher#sh ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.0.0.5          0   FULL/  -        00:00:35    10.0.0.2        Serial0/0
Teacher#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.25.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, Serial0/0
O       10.0.0.4 [110/128] via 10.0.0.2, 00:03:27, Serial0/0
Teacher#
```



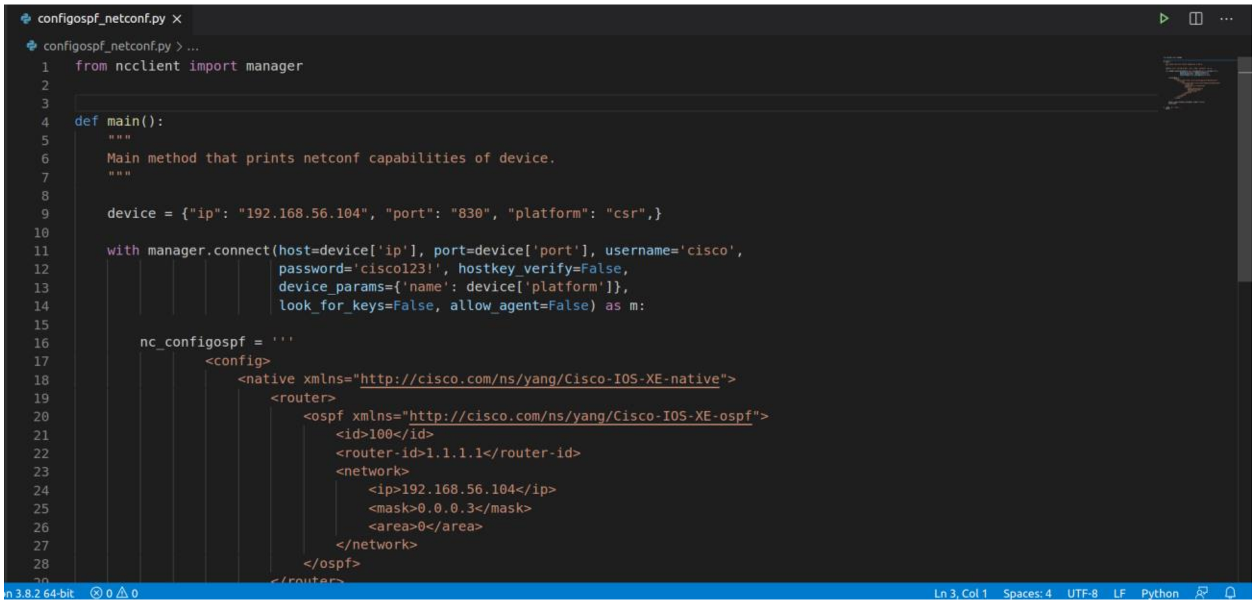
```
Admin#sh ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.0.0.6          0   FULL/  -        00:00:39    10.0.0.6        Serial0/1
192.168.25.1      0   FULL/  -        00:00:32    10.0.0.1        Serial0/0
Admin#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.25.0/24 [110/74] via 10.0.0.1, 00:04:23, Serial0/0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.0.0 is directly connected, Serial0/0
C       10.0.0.4 is directly connected, Serial0/1
Admin#
```



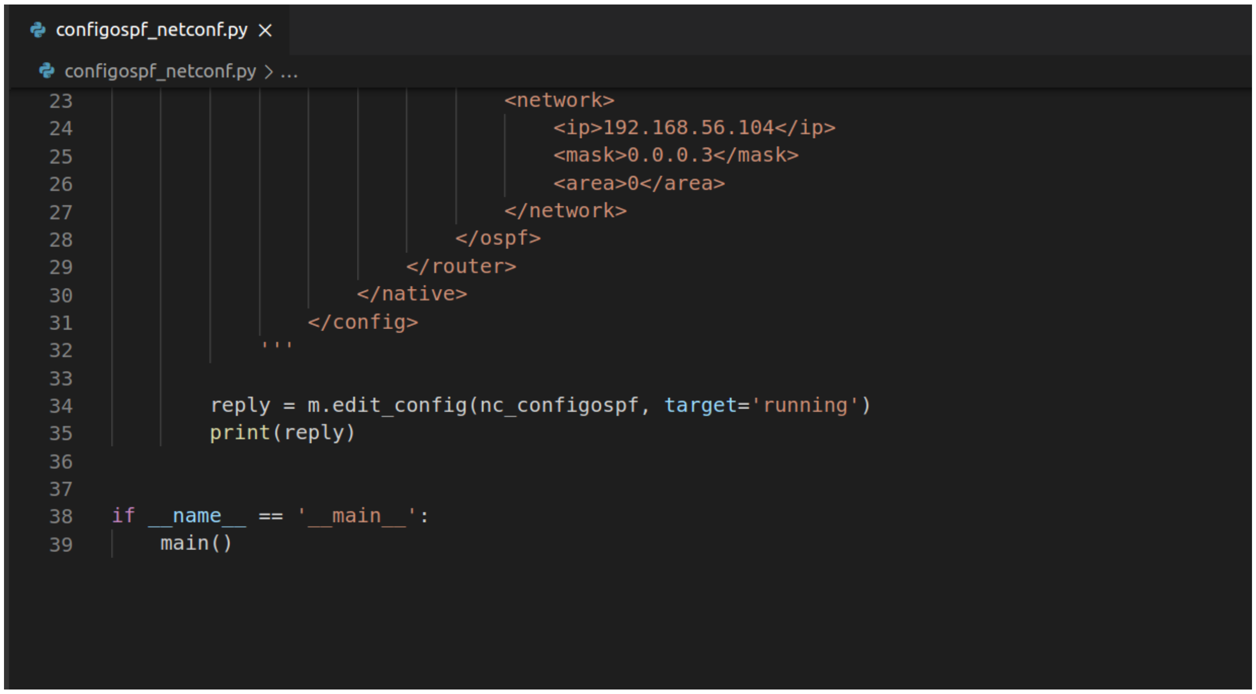
```
Student#sh ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.0.0.5          0   FULL/  -        00:00:34    10.0.0.5        Serial0/1
Student#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.25.0/24 [110/138] via 10.0.0.5, 00:05:50, Serial0/1
     10.0.0.0/30 is subnetted, 2 subnets
O       10.0.0.0 [110/128] via 10.0.0.5, 00:06:02, Serial0/1
C       10.0.0.4 is directly connected, Serial0/1
Student#
```

Part 2: Network Programmability using NETCONF

1. Start DEVASC-LABVM virtual machine
2. Configure an OSPF routing protocol with process id of 100 using NETCONF


configospf_netconf.py
configospf_netconf.py > ...
n 3.8.2 64-bit ⊗ 0 ⚠ 0
Ln 3, Col 1 Spaces: 4 UTF-8 LF Python

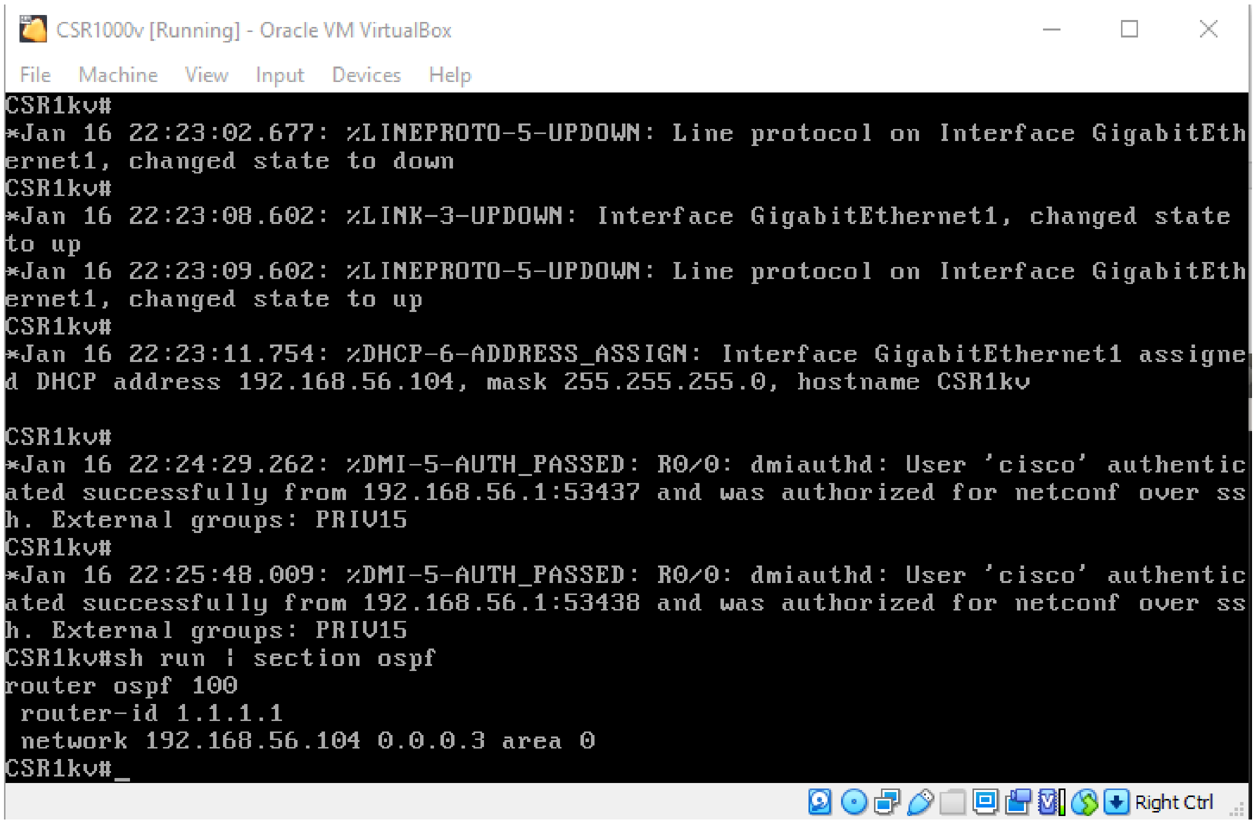
```python
from ncclient import manager


def main():
    """
    Main method that prints netconf capabilities of device.
    """

    device = {"ip": "192.168.56.104", "port": "830", "platform": "csr",}

    with manager.connect(host=device['ip'], port=device['port'], username='cisco',
                         password='cisco123!', hostkey_verify=False,
                         device_params={'name': device['platform']},
                         look_for_keys=False, allow_agent=False) as m:

        nc_configospf = '''
                <config>
                    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
                        <router>
                            <ospf xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-ospf">
                                <id>100</id>
                                <router-id>1.1.1.1</router-id>
                                <network>
                                    <ip>192.168.56.104</ip>
                                    <mask>0.0.0.3</mask>
                                    <area>0</area>
                                </network>
                            </ospf>
```



```python
                                <network>
                                    <ip>192.168.56.104</ip>
                                    <mask>0.0.0.3</mask>
                                    <area>0</area>
                                </network>
                            </ospf>
```

```python
                        </router>
                    </native>
                </config>
            '''

        reply = m.edit_config(nc_configospf, target='running')
        print(reply)


if __name__ == '__main__':
    main()
```

3. Run the file and check the output in the CSR1000v virtual machine
   a. Use sh run | section ospf command to check if the router ospf configure is success.


CSR1000v [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
CSR1kv#
*Jan 16 22:23:02.677: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet1, changed state to down
CSR1kv#
*Jan 16 22:23:08.602: %LINK-3-UPDOWN: Interface GigabitEthernet1, changed state
to up
*Jan 16 22:23:09.602: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEth
ernet1, changed state to up
CSR1kv#
*Jan 16 22:23:11.754: %DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet1 assigne
d DHCP address 192.168.56.104, mask 255.255.255.0, hostname CSR1kv

CSR1kv#
*Jan 16 22:24:29.262: %DMI-5-AUTH_PASSED: R0/0: dmiauthd: User 'cisco' authentic
ated successfully from 192.168.56.1:53437 and was authorized for netconf over ss
h. External groups: PRIV15
CSR1kv#
*Jan 16 22:25:48.009: %DMI-5-AUTH_PASSED: R0/0: dmiauthd: User 'cisco' authentic
ated successfully from 192.168.56.1:53438 and was authorized for netconf over ss
h. External groups: PRIV15
CSR1kv#sh run | section ospf
router ospf 100
 router-id 1.1.1.1
 network 192.168.56.104 0.0.0.3 area 0
CSR1kv#_
```

Prepared by:

Mark Joel P. Enriquez

Technological Institute of the Philippines – Quezon City