FINAL CASE STUDY

Submitted by:

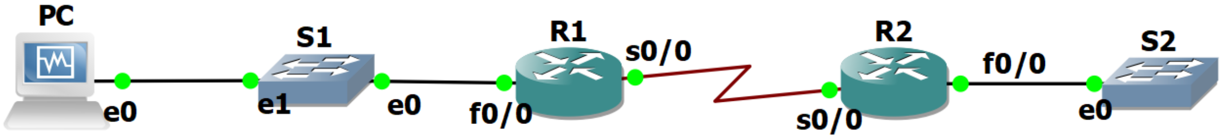
Elain S. Perol

CPE41S3

Submitted to:

Engr. Alonica Villanueva

Network Automation using Ansible



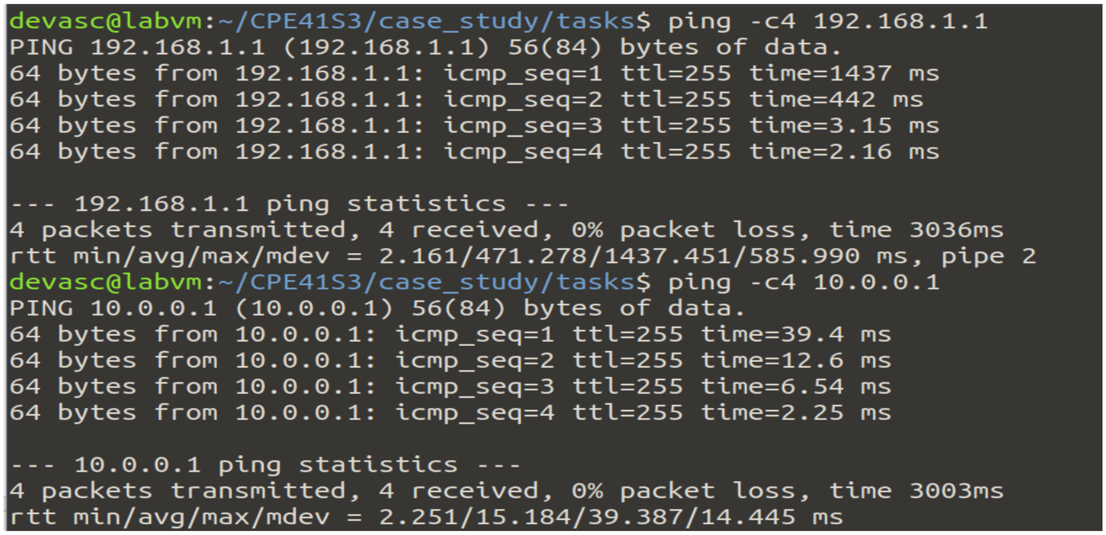
Objectives

- Configure ACL using Ansible Playbook
- Configure Single Area OSPF using Ansible Playbook
- Create backup files using Ansible Playbook
- Getting IPv4 and IP Route Information using Ansible Playbook
- Create Ansible Playbook

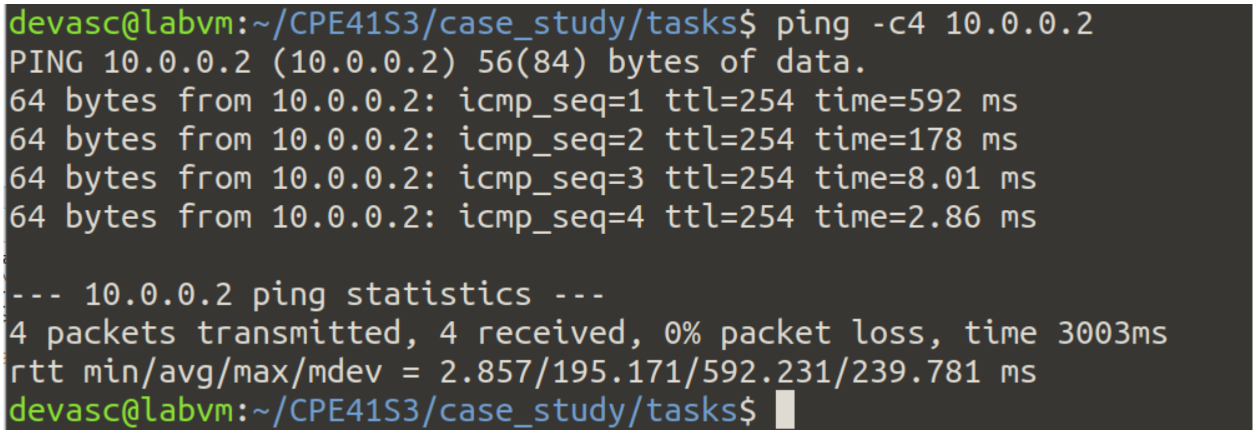
Resources

- 1 PC with capability of running GNS3 and VirtualBox
- VirtualBox
- DEVASC Virtual Machine
- GNS3
- GNS3 VM

1. **Check the connection between the PC and R1 and R2. Ping R1 and R2 in PC.**

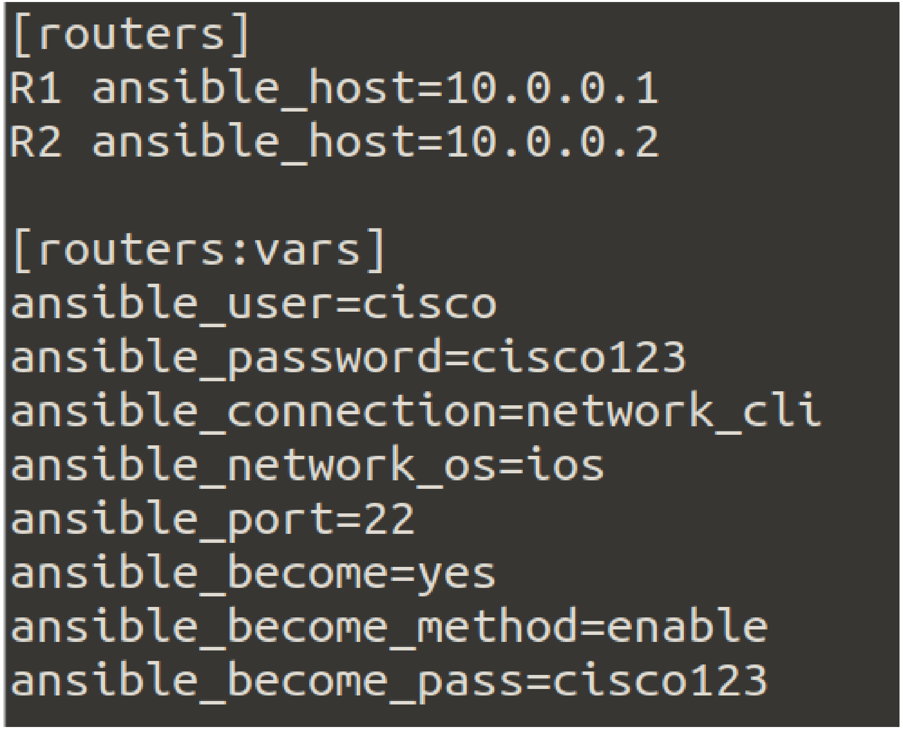


```
devasc@labvm:~/CPE41S3/case_study/tasks$ ping -c4 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=1437 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=442 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=255 time=3.15 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=255 time=2.16 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3036ms
rtt min/avg/max/mdev = 2.161/471.278/1437.451/585.990 ms, pipe 2
devasc@labvm:~/CPE41S3/case_study/tasks$ ping -c4 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=255 time=39.4 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=255 time=12.6 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=255 time=6.54 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=255 time=2.25 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 2.251/15.184/39.387/14.445 ms
```



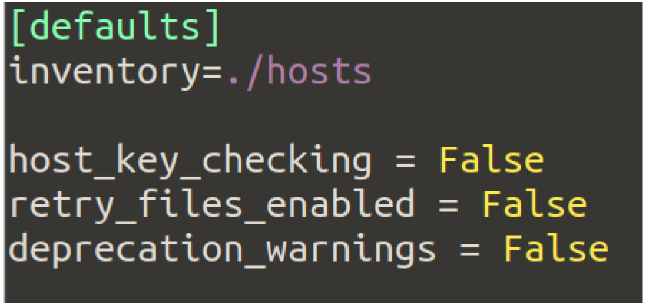
```
devasc@labvm:~/CPE41S3/case_study/tasks$ ping -c4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=254 time=592 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=254 time=178 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=254 time=8.01 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=254 time=2.86 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 2.857/195.171/592.231/239.781 ms
devasc@labvm:~/CPE41S3/case_study/tasks$
```

2. **Create new folder with the name of your choice. Make a hosts file and input the following:**



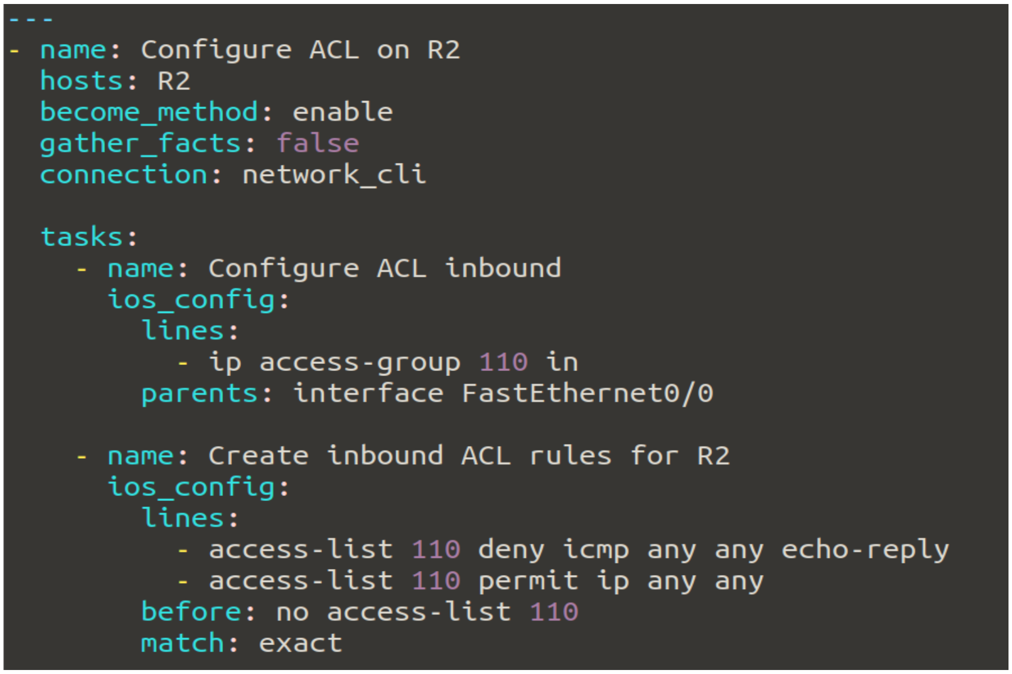
```
[routers]
R1 ansible_host=10.0.0.1
R2 ansible_host=10.0.0.2

[routers:vars]
ansible_user=cisco
ansible_password=cisco123
ansible_connection=network_cli
ansible_network_os=ios
ansible_port=22
ansible_become=yes
ansible_become_method=enable
ansible_become_pass=cisco123
```

3. **Create a new file named “ansible.cfg” and type the following:**



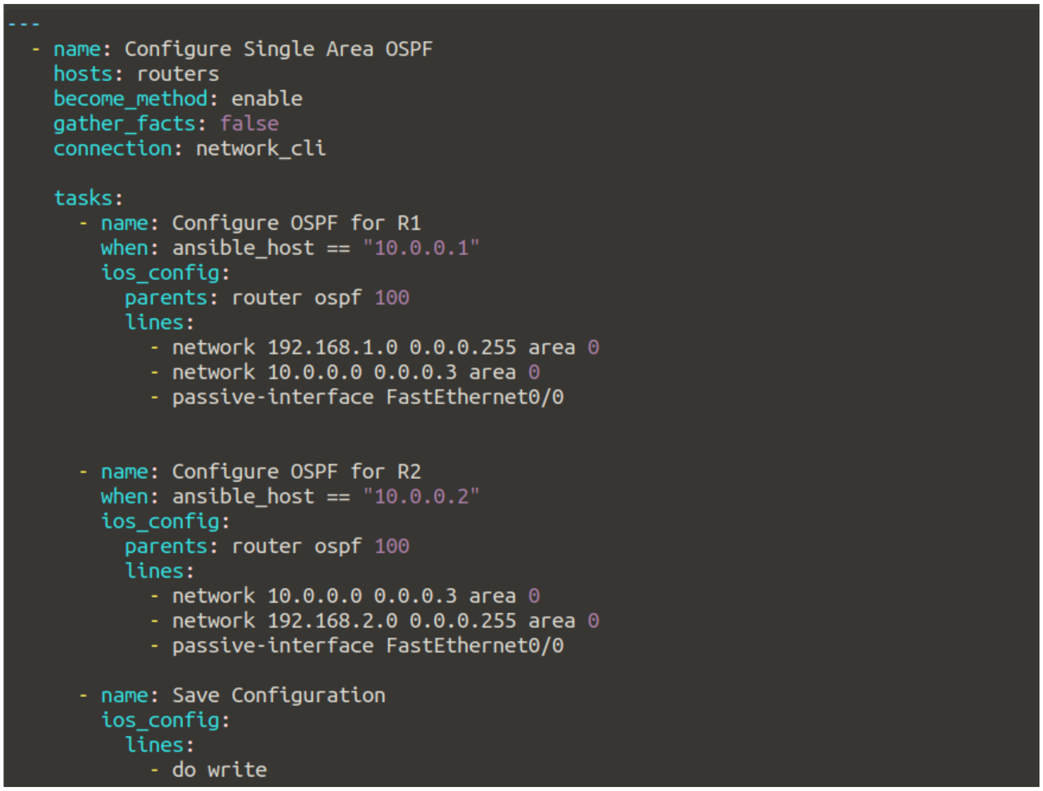
```
[defaults]
inventory=./hosts

host_key_checking = False
retry_files_enabled = False
deprecation_warnings = False
```

4. Create a new folder named “tasks” and inside the tasks folder, create a playbook for ACL configuration then type the following:



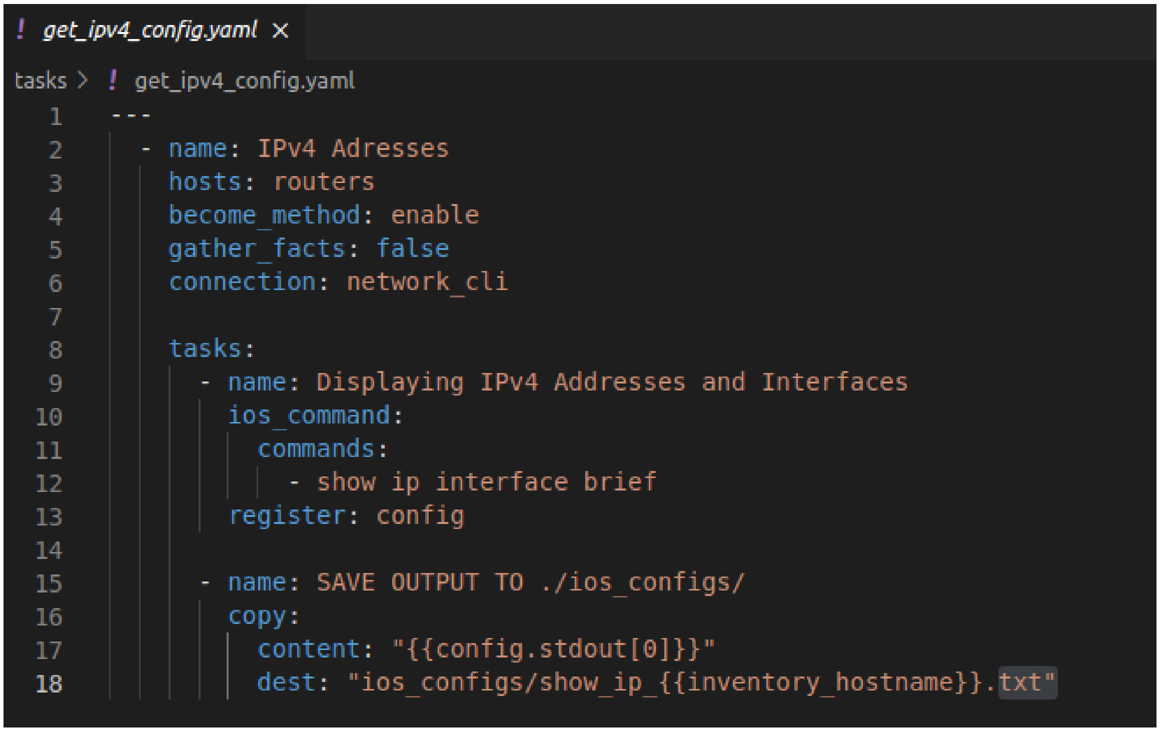
```
---
- name: Configure ACL on R2
  hosts: R2
  become_method: enable
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Configure ACL inbound
      ios_config:
        lines:
          - ip access-group 110 in
        parents: interface FastEthernet0/0

    - name: Create inbound ACL rules for R2
      ios_config:
        lines:
          - access-list 110 deny icmp any any echo-reply
          - access-list 110 permit ip any any
        before: no access-list 110
        match: exact
```

5. Create a new playbook for OSPF Configuration with the following codes.

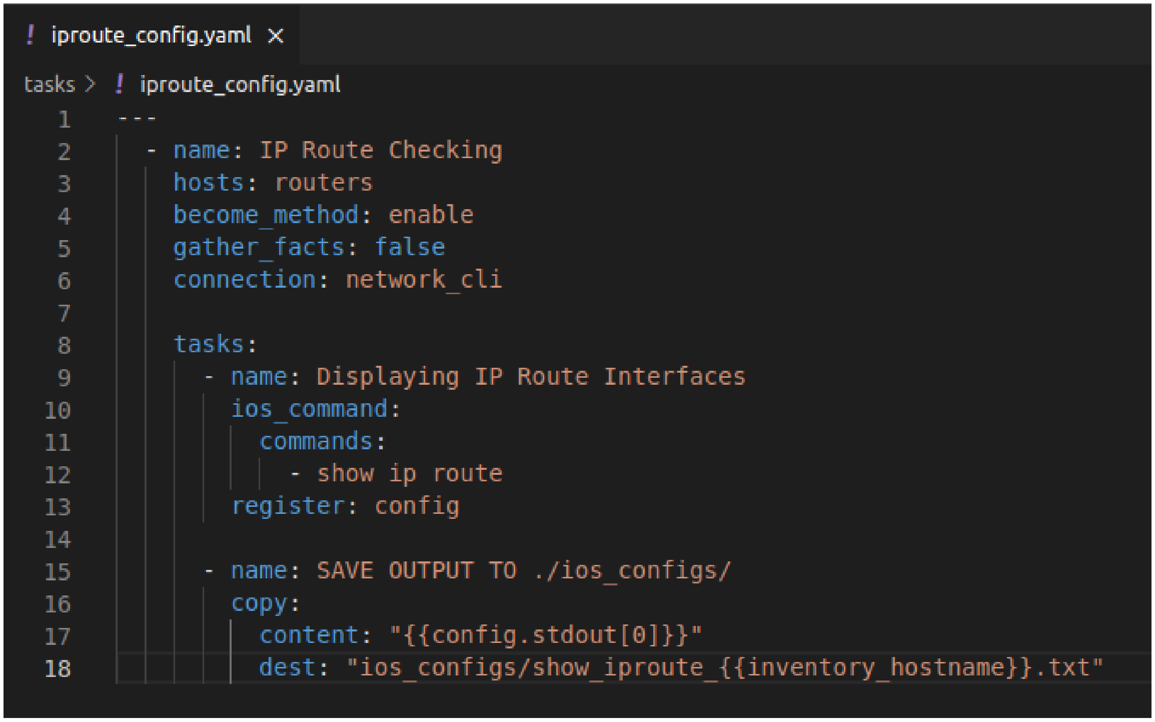


```
---
 - name: Configure Single Area OSPF
   hosts: routers
   become_method: enable
   gather_facts: false
   connection: network_cli

   tasks:
    - name: Configure OSPF for R1
      when: ansible_host == "10.0.0.1"
      ios_config:
        parents: router ospf 100
        lines:
          - network 192.168.1.0 0.0.0.255 area 0
          - network 10.0.0.0 0.0.0.3 area 0
          - passive-interface FastEthernet0/0


    - name: Configure OSPF for R2
      when: ansible_host == "10.0.0.2"
      ios_config:
        parents: router ospf 100
        lines:
          - network 10.0.0.0 0.0.0.3 area 0
          - network 192.168.2.0 0.0.0.255 area 0
          - passive-interface FastEthernet0/0

    - name: Save Configuration
      ios_config:
        lines:
          - do write
```

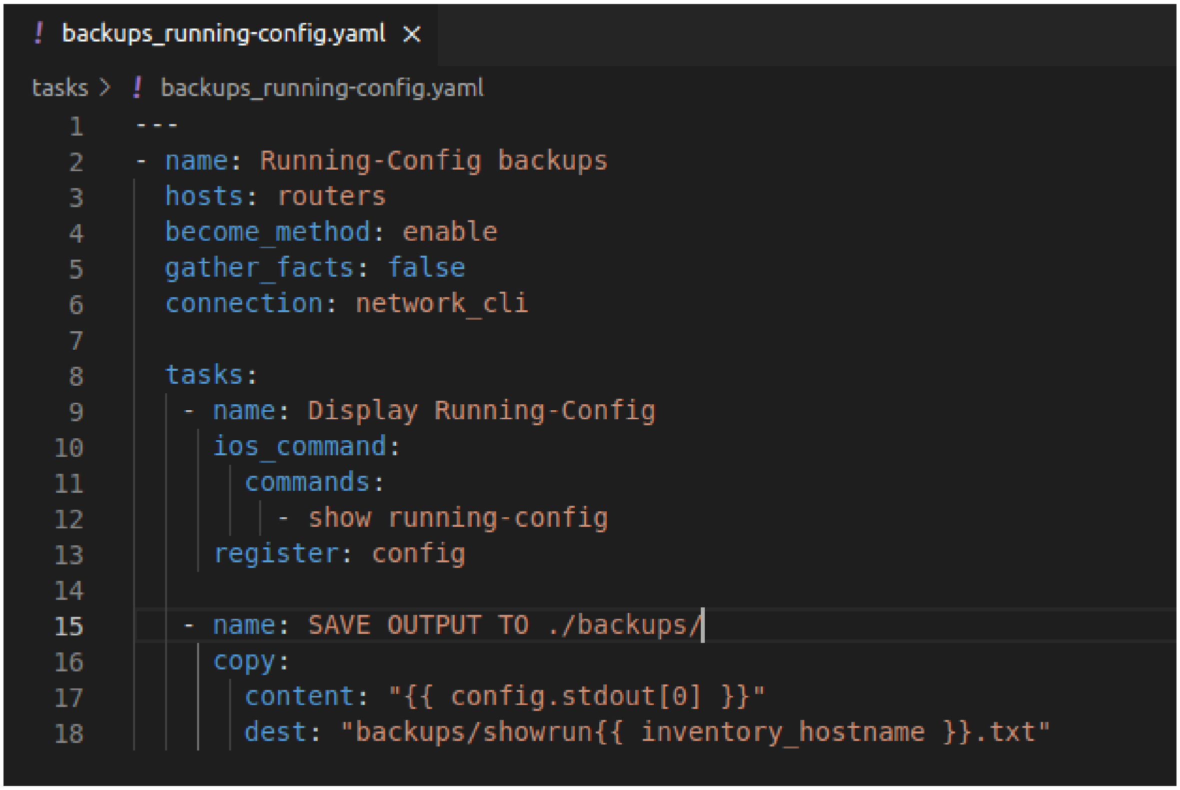
6. **Outside the tasks folder, create backups and ios_config folder, this is where the outputs of the playbooks tasks will be saved. Go back to tasks folder and create playbooks for getting IPv4 addresses and interface, IP Route, running-configuration, and for saving the outputs to the made folders. Using the following codes:**



```yaml
---
- name: IPv4 Adresses
  hosts: routers
  become_method: enable
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Displaying IPv4 Addresses and Interfaces
      ios_command:
        commands:
          - show ip interface brief
      register: config

    - name: SAVE OUTPUT TO ./ios_configs/
      copy:
        content: "{{config.stdout[0]}}"
        dest: "ios_configs/show_ip_{{inventory_hostname}}.txt"
```

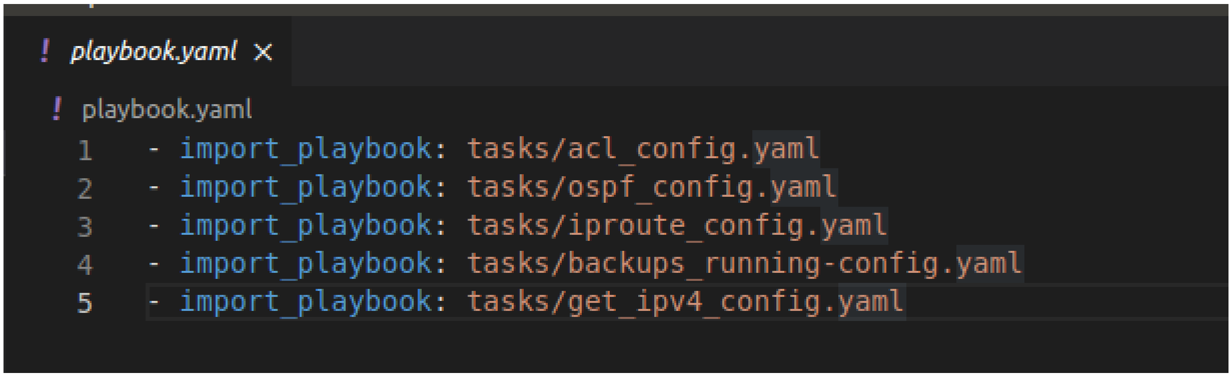


```yaml
---
- name: IP Route Checking
  hosts: routers
  become_method: enable
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Displaying IP Route Interfaces
      ios_command:
        commands:
          - show ip route
      register: config

    - name: SAVE OUTPUT TO ./ios_configs/
      copy:
        content: "{{config.stdout[0]}}"
        dest: "ios_configs/show_iproute_{{inventory_hostname}}.txt"
```


backups_running-config.yaml
tasks > backups_running-config.yaml

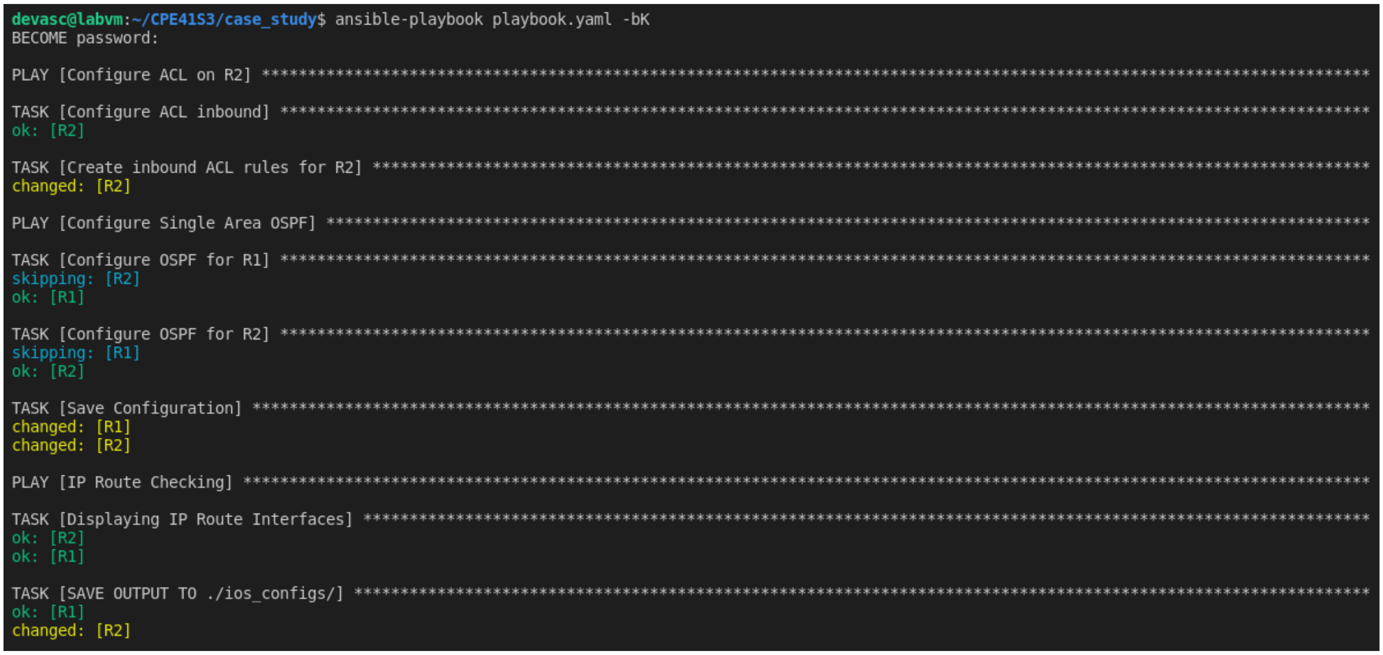
```yaml
---
- name: Running-Config backups
  hosts: routers
  become_method: enable
  gather_facts: false
  connection: network_cli

  tasks:
   - name: Display Running-Config
     ios_command:
       commands:
         - show running-config
     register: config

   - name: SAVE OUTPUT TO ./backups/
     copy:
       content: "{{ config.stdout[0] }}"
       dest: "backups/showrun{{ inventory_hostname }}.txt"
```

7. **Outside the tasks folder create a playbook that will run all the playbooks made inside the tasks folder. Using the following codes:**



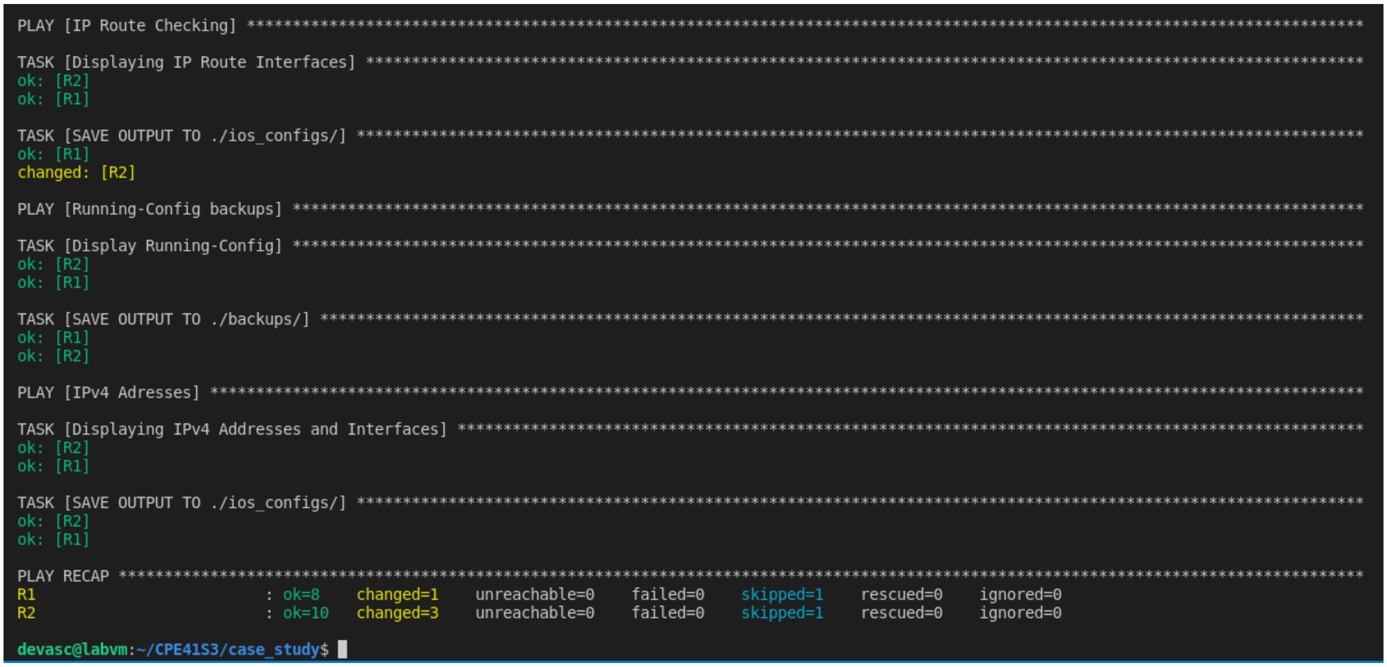
```yaml
- import_playbook: tasks/acl_config.yaml
- import_playbook: tasks/ospf_config.yaml
- import_playbook: tasks/iproute_config.yaml
- import_playbook: tasks/backups_running-config.yaml
- import_playbook: tasks/get_ipv4_config.yaml
```

8. **Open the terminal and run the command ansible-playbook playbook.yaml -bK. It will ask for become password, used the ansible_become_pass in from the hosts file, and wait for the playbook to run.**



```
devasc@labvm:~/CPE41S3/case_study$ ansible-playbook playbook.yaml -bK
BECOME password:

PLAY [Configure ACL on R2] *****************************************************************************************************

TASK [Configure ACL inbound] ***************************************************************************************************
ok: [R2]

TASK [Create inbound ACL rules for R2] *****************************************************************************************
changed: [R2]

PLAY [Configure Single Area OSPF] **********************************************************************************************

TASK [Configure OSPF for R1] ***************************************************************************************************
skipping: [R2]
ok: [R1]

TASK [Configure OSPF for R2] ***************************************************************************************************
skipping: [R1]
ok: [R2]

TASK [Save Configuration] ******************************************************************************************************
changed: [R1]
changed: [R2]

PLAY [IP Route Checking] *******************************************************************************************************

TASK [Displaying IP Route Interfaces] ******************************************************************************************
ok: [R2]
ok: [R1]

TASK [SAVE OUTPUT TO ./ios_configs/] *******************************************************************************************
ok: [R1]
changed: [R2]
```

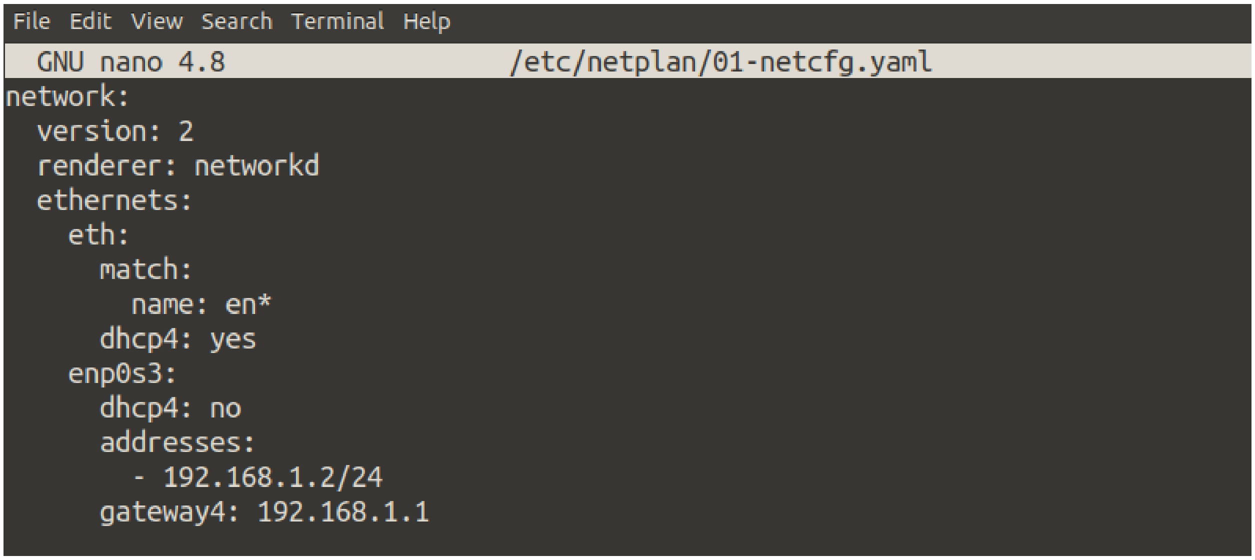
Output of ACL and OSPF Configuration Tasks



```
PLAY [IP Route Checking] *******************************************************************************************************

TASK [Displaying IP Route Interfaces] ******************************************************************************************
ok: [R2]
ok: [R1]

TASK [SAVE OUTPUT TO ./ios_configs/] *******************************************************************************************
ok: [R1]
changed: [R2]

PLAY [Running-Config backups] **************************************************************************************************

TASK [Display Running-Config] **************************************************************************************************
ok: [R2]
ok: [R1]

TASK [SAVE OUTPUT TO ./backups/] ***********************************************************************************************
ok: [R1]
ok: [R2]

PLAY [IPv4 Adresses] ***********************************************************************************************************

TASK [Displaying IPv4 Addresses and Interfaces] ********************************************************************************
ok: [R2]
ok: [R1]

TASK [SAVE OUTPUT TO ./ios_configs/] *******************************************************************************************
ok: [R2]
ok: [R1]

PLAY RECAP *********************************************************************************************************************
R1                         : ok=8    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
R2                         : ok=10   changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

devasc@labvm:~/CPE41S3/case_study$ 
```

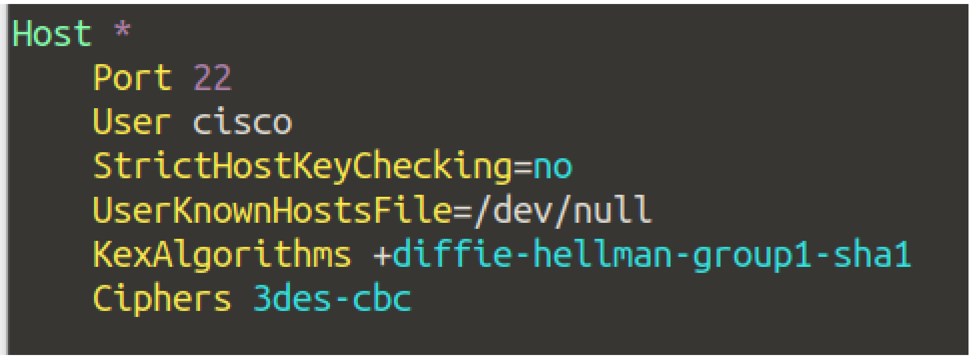
Output of getting IPv4 addresses and interfaces, IP route, running configuration and the OVERALL result of the tasks that we run using ansible playbook.

Net Plan Configuration:



```
  GNU nano 4.8                    /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    eth:
      match:
        name: en*
      dhcp4: yes
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.2/24
      gateway4: 192.168.1.1
```

~/.ssh/config



```
Host *
     Port 22
     User cisco
     StrictHostKeyChecking=no
     UserKnownHostsFile=/dev/null
     KexAlgorithms +diffie-hellman-group1-sha1
     Ciphers 3des-cbc
```

LIST OF PLAYBOOKS

- acl_config.yaml
- backups_running-config.yaml
- get_ipv4_config.yaml
- iproute_config.yaml
- ospf_config.yaml
- playbook.yaml (**master playbook**)

GitHub Repository: https://github.com/eperol-tip/Final-Case-Study/tree/master
Video Presentation Link: Video

I affirm that I have not given or received any unauthorized help on this case study, and that this work is my own.