

LOUIS ADRIAN MENDOZA

Lab – Final Case Study (Network Automation)

Objectives

- Part 1: Launch the DEVASC VM & GNS3
- Part 2: Create Network Topology
- Part 3: Configure the basic configuration of the network
- Part 4: Utilize Ansible for network automation (3 Network Topics)
- Part 5: Use PyATS and Genie to test connectivity

Background / Scenario

In this lab, I'm going to create a network topology for a school using GNS3. I would also be going to use Ansible for network automation. The school wants to have three networks for the computer lab, school office and for the management.

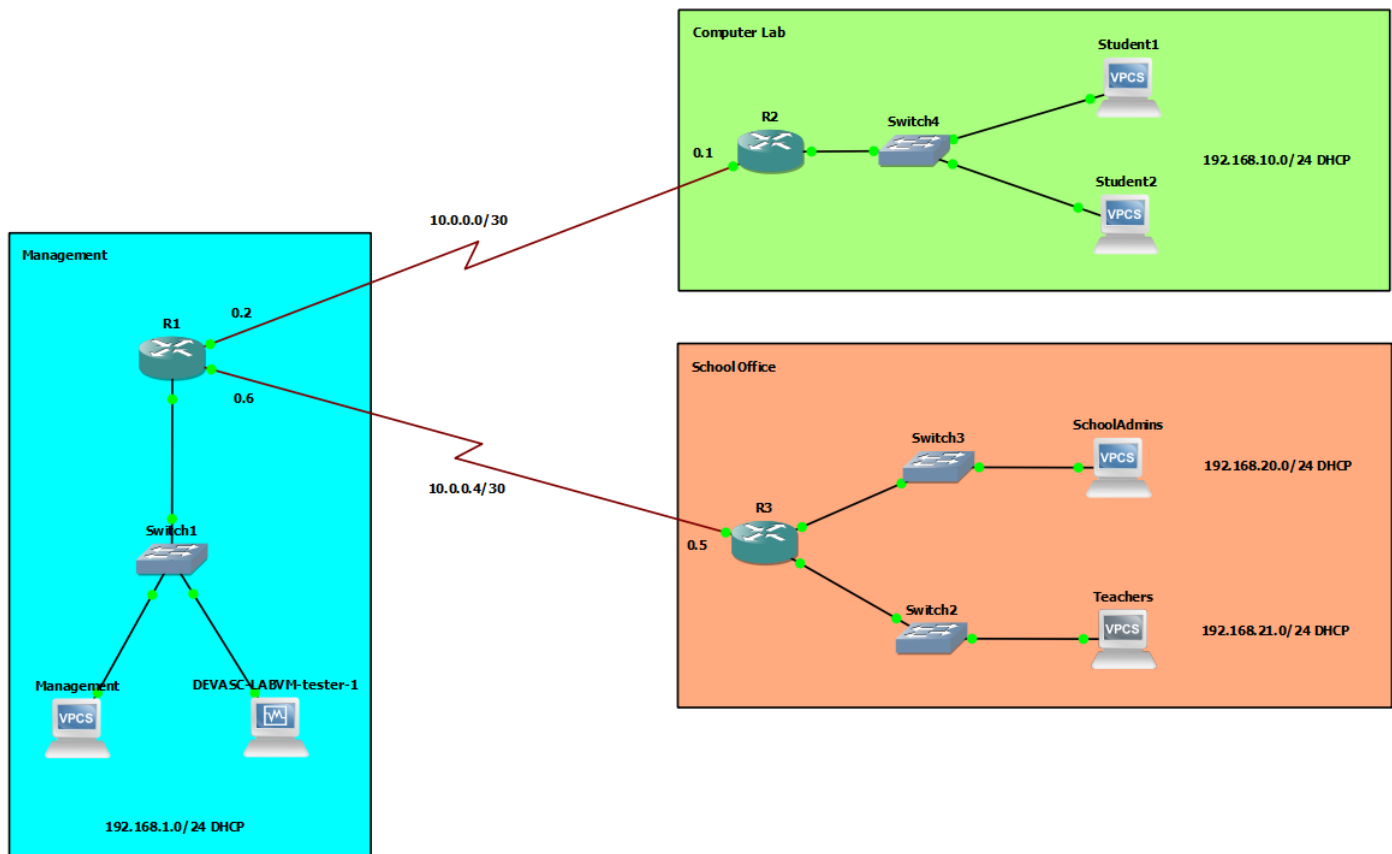


Figure 1: School Network Topology

The figure shows the network topology that the school wanted to be accomplished. There are three networks: Management for the network automation and management of school, Computer lab for the student's educational computers, and School office for the teachers and school admins.

Table 1: Addressing table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	F0/0	192.168.1.1	255.255.255.0	N/A
	S2/0	10.0.0.6	255.255.255.252	
	S2/1	10.0.0.2	255.255.255.252	
R2	F0/0	192.168.10.1	255.255.255.0	N/A
	S2/0	10.0.0.1	255.255.255.252	
R3	F0/0	192.168.20.1	255.255.255.0	N/A
	F0/1	192.168.21.1	255.255.255.0	
	S2/0	10.0.0.5	255.255.255.252	
PC-DEVASC	E0	192.168.1.2	255.255.255.0	192.168.1.1
PC-Management	E0	(DHCP) 192.168.1.0/24	255.255.255.0	192.168.1.1
PC-Students	E0	(DHCP) 192.168.10.0/24	255.255.255.0	192.168.10.1
PC-School Admins	E0	(DHCP) 192.168.20.0/24	255.255.255.0	192.168.20.1
PC-Teachers	E0	(DHCP) 192.168.21.0/24	255.255.255.0	192.168.21.1

The table shows the planned addressing table for the network topology and the configuration. There are four networks that will have a DHCP for their network, the management, students, school admins and for the teachers.

The three network topics that I tackle on the network automation are the following:

- Configuration of DHCP pools for the 4 networks
- Configuration of OSPF in the routers
- Application of Extended ACL for the network.

Required Resources

- GNS3 network simulator and GNS3 VM
- Virtual Box or VMWare
- DEVASC Virtual Machine
- Ansible
- PyATS

Instructions

Part 1: Launch the DEVASC VM & GNS3

Launch the DEVASC VM, GNS3 network simulator, and GNS3 VM, that are needed for the case study.

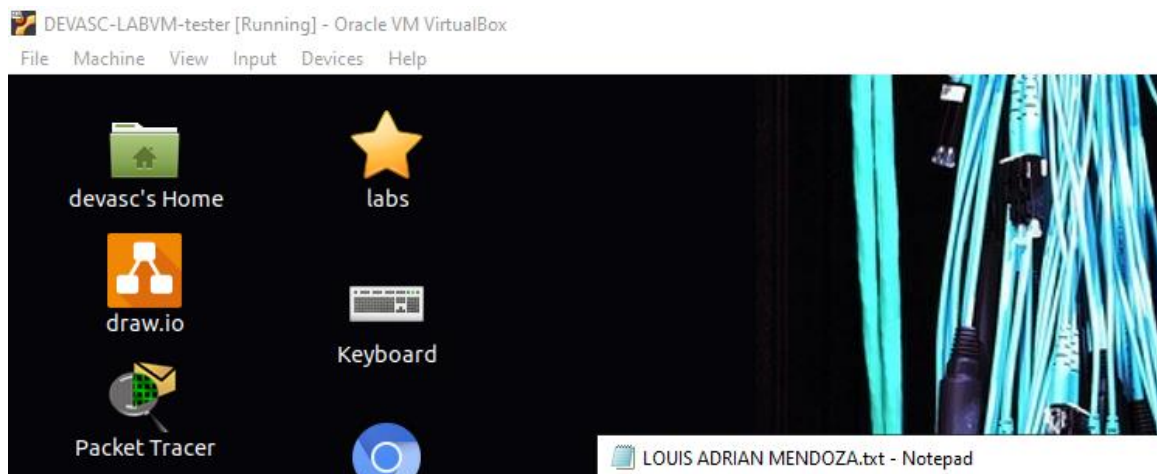


Figure 2: DEVASC Machine

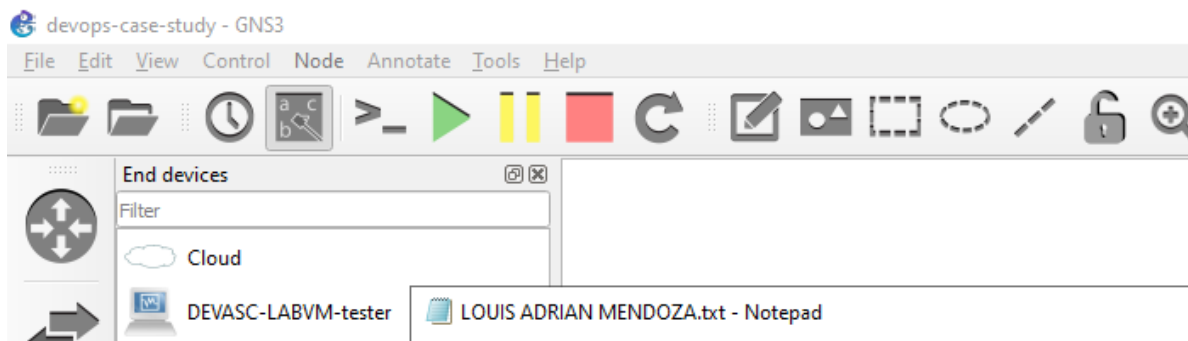


Figure 2: GNS3 Network Simulator

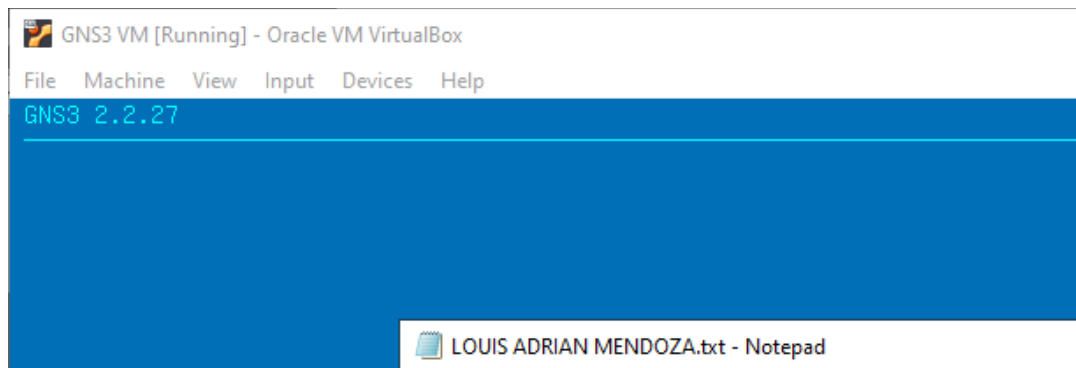


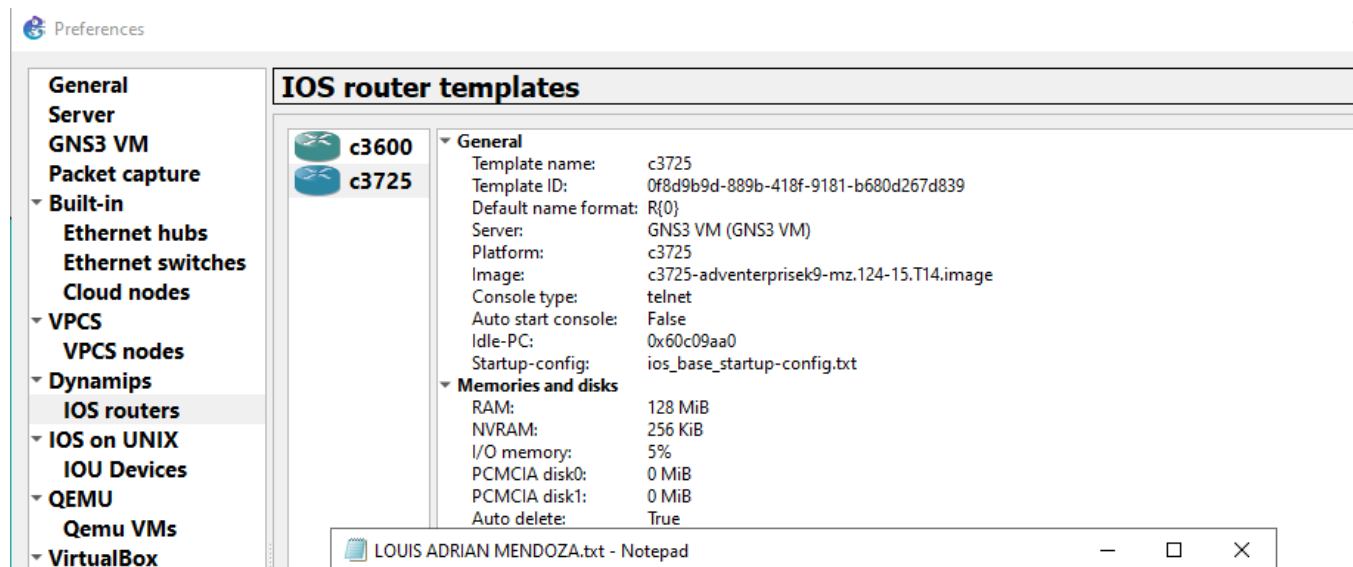
Figure 3: GNS3 VM

Part 2: Create Network Topology

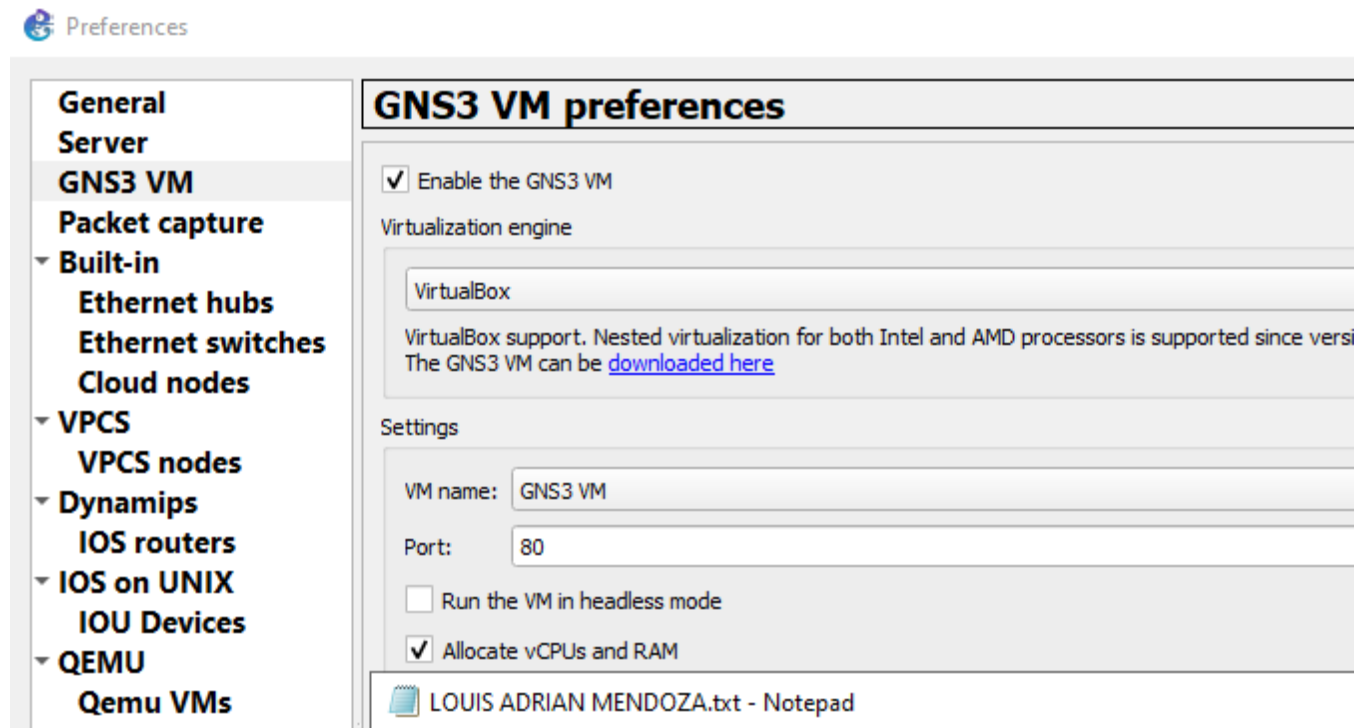
In this Part, we are going to create the network topology using the GNS3 machine and the GNS3 network simulator.

Step 1: Add Devices on the GNS3

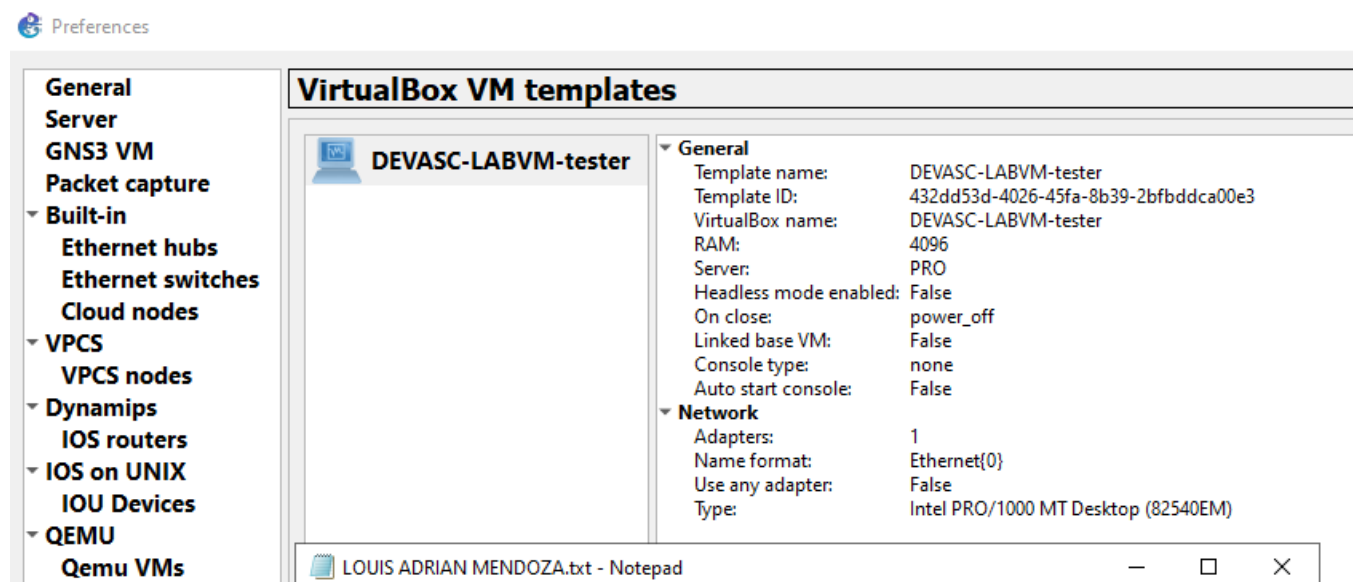
- a. Open preferences and add devices on IOS routers templates



- b. Enable the GNS3 VM

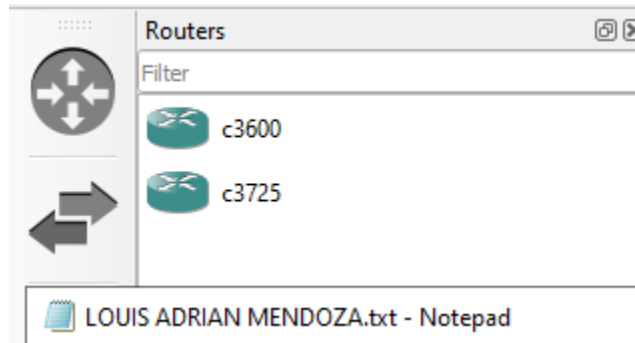


- c. Add the DEVASC machine to the virtualbox VM templates

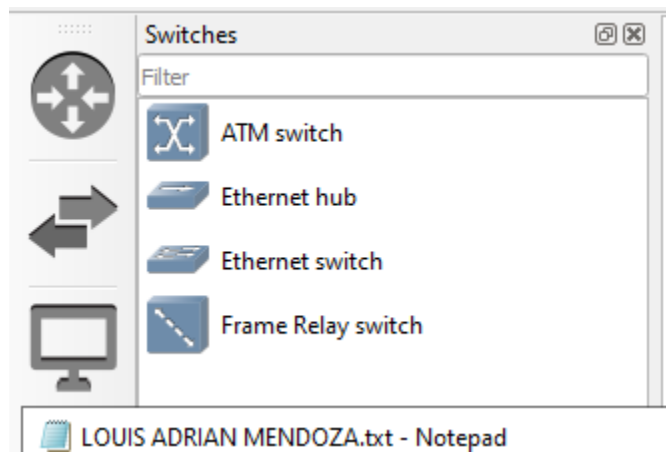


Step 2: Create the topology

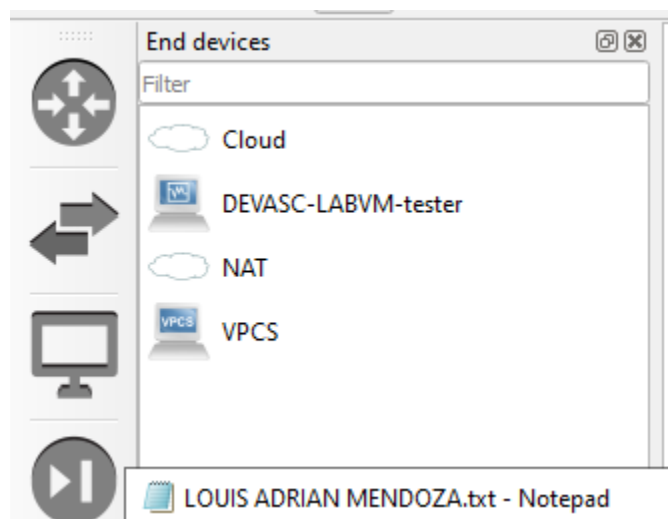
- a. Add 3 routers to the topology from the routers tab.



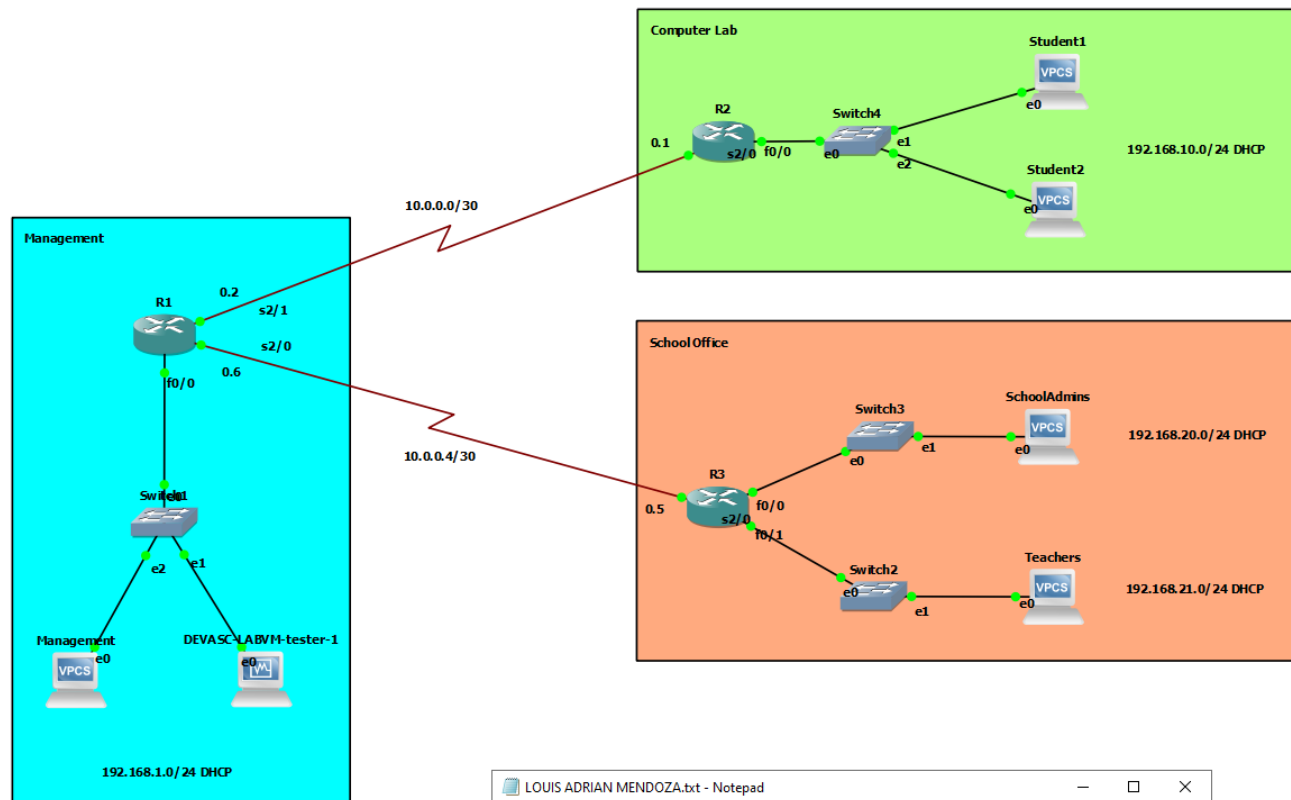
- b. Add 4 ethernet switches to the topology from the switch tab.



- c. Add DEVASC machine and Virtual PCs to the topology from the end devices tab.



- d. Connect devices using link devices tab and create the topology based of the Figure 1 showcase above.



Part 3: Configure the basic configuration of the network

Utilizing the knowledge from past lessons of Computer Networks 1,2 and 3, I'm able to apply and configure the basic configuration needed for the network

- Enter the Basic config for the R1.

R1

```
enable password cisco123
ip domain name www.abc.com
username cisco password 0 cisco123

crypto key generate rsa
1024

ip ssh version 2

int f0/0
```

```
ip address 192.168.1.1 255.255.255.0
no shut

int s2/0
ip address 10.0.0.6 255.255.255.252
no shut

int s2/1
ip address 10.0.0.2 255.255.255.252
no shut

router ospf 10
log-adjacency-changes
network 10.0.0.0 0.0.0.3 area 0
network 10.0.0.4 0.0.0.3 area 0
network 192.168.1.0 0.0.0.255 area 0
default-information originate

line vty 0 15
login local
transport input ssh
```

b. Enter the Basic config for the R2.

```
R2

enable password cisco123
ip domain name www.abc.com
username cisco password 0 cisco123

crypto key generate rsa
1024

ip ssh version 2

int f0/0
ip address 192.168.10.1 255.255.255.0
no shut

int s2/0
```



```
ip address 10.0.0.1 255.255.255.252
no shut

router ospf 10
log-adjacency-changes
network 10.0.0.0 0.0.0.3 area 0
network 192.168.10.0 0.0.0.255 area 0

line vty 0 15
login local
transport input ssh
```

- c. Enter the Basic config for the R3

R3
<pre>enable password cisco123 ip domain name www.abc.com username cisco password 0 cisco123 crypto key generate rsa 1024 ip ssh version 2 int f0/0 ip address 192.168.20.1 255.255.255.0 no shut int f0/1 ip address 192.168.21.1 255.255.255.0 no shut int s2/0 ip address 10.0.0.5 255.255.255.252 no shut router ospf 10 log-adjacency-changes network 10.0.0.4 0.0.0.3 area 0</pre>

```
network 192.168.20.0 0.0.0.255 area 0
```

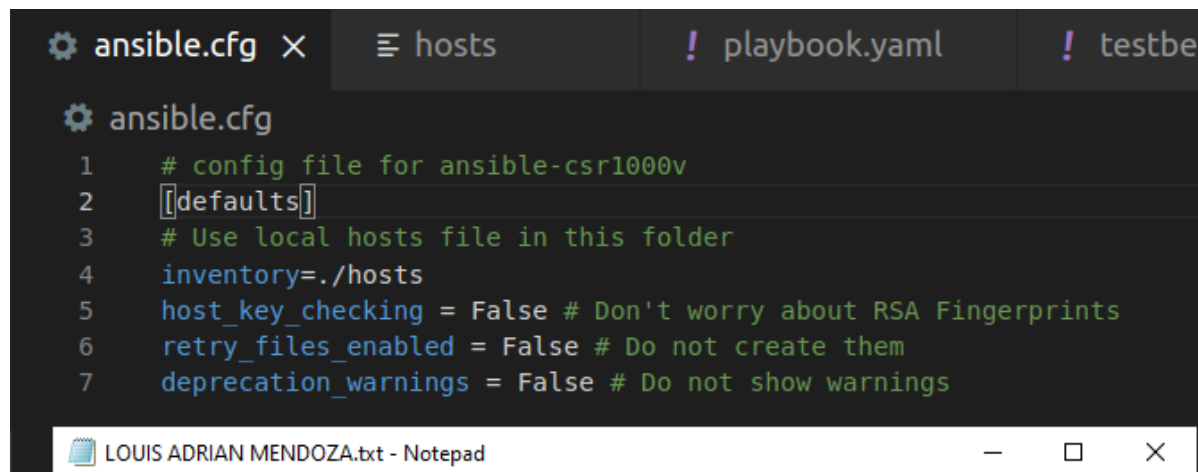
```
line vty 0 15
login local
transport input ssh
```

Part 4: Utilize Ansible for network automation (Three Network Topics)

Ansible was utilized for the case study for the network automation of the school. This application makes automation more accessible and easier for the network configurations.

Step 1: Create hosts and ansible.cfg for the playbook

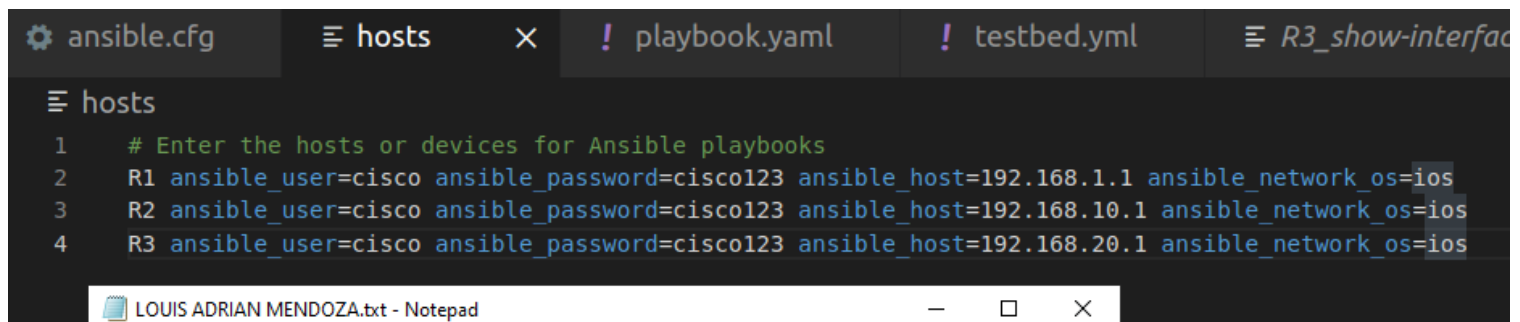
- a. Create the config file for the ansible in VS code studio.



The screenshot shows the VS Code editor with the `ansible.cfg` file open. The file contains the following configuration:

```
1 # config file for ansible-csr1000v
2 [defaults]
3 # Use local hosts file in this folder
4 inventory=./hosts
5 host_key_checking = False # Don't worry about RSA Fingerprints
6 retry_files_enabled = False # Do not create them
7 deprecation_warnings = False # Do not show warnings
```

- b. Create another file for the hosts and their specific information. This is important as it will be the information that the ansible will use for its hosts or devices.

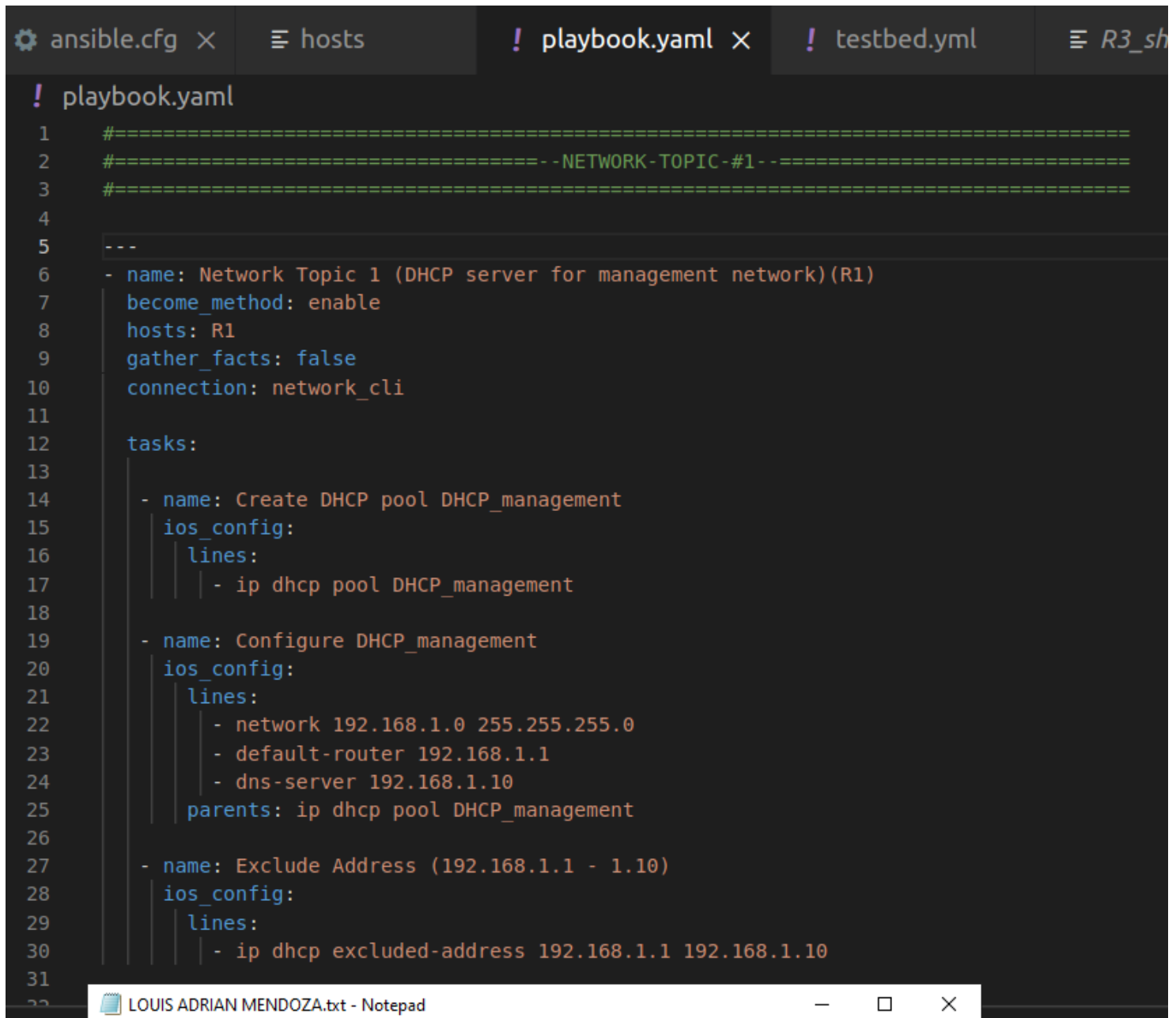


The screenshot shows the VS Code editor with the `hosts` file open. The file contains the following configuration:

```
1 # Enter the hosts or devices for Ansible playbooks
2 R1 ansible_user=cisco ansible_password=cisco123 ansible_host=192.168.1.1 ansible_network_os=ios
3 R2 ansible_user=cisco ansible_password=cisco123 ansible_host=192.168.10.1 ansible_network_os=ios
4 R3 ansible_user=cisco ansible_password=cisco123 ansible_host=192.168.20.1 ansible_network_os=ios
```

Step 2: Create the Playbook for the three network topics

Network Topic 1: Configuration of DHCP pools for the 4 networks



```
! playbook.yaml
1  #=====
2  #----- --NETWORK-TOPIC-#1-- -----
3  #=====
4
5  ---
6  - name: Network Topic 1 (DHCP server for management network)(R1)
7    become_method: enable
8    hosts: R1
9    gather_facts: false
10   connection: network_cli
11
12   tasks:
13
14     - name: Create DHCP pool DHCP_management
15       ios_config:
16         lines:
17           - ip dhcp pool DHCP_management
18
19     - name: Configure DHCP_management
20       ios_config:
21         lines:
22           - network 192.168.1.0 255.255.255.0
23           - default-router 192.168.1.1
24           - dns-server 192.168.1.10
25         parents: ip dhcp pool DHCP_management
26
27     - name: Exclude Address (192.168.1.1 - 1.10)
28       ios_config:
29         lines:
30           - ip dhcp excluded-address 192.168.1.1 192.168.1.10
31
32
```


Configuration of DHCP for the management network in R1

```
- name: Network Topic 1 (DHCP server for computer lab network)(R2)
  become_method: enable
  hosts: R2
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Create DHCP pool DHCP_students
      ios_config:
        lines:
          - ip dhcp pool DHCP_students

    - name: Configure DHCP_students
      ios_config:
        lines:
          - network 192.168.10.0 255.255.255.0
          - default-router 192.168.10.1
          - dns-server 192.168.10.10
        parents: ip dhcp pool DHCP_students

    - name: Exclude Address (192.168.10.1 - 10.10)
      ios_config:
        lines:
          - ip dhcp excluded-address 192.168.10.1 192.168.10.10
```

 LOUIS ADRIAN MENDOZA.txt - Notepad

— □ ×

Configuration of DHCP for the computer lab network in R2

```
- name: Network Topic 1 (DHCP server for school main office network)(R3)
  become_method: enable
  hosts: R3
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Create 2 DHCP pool (DHCP_teachers & DHCP_schooladmins)
      ios_config:
        lines:
          - ip dhcp pool DHCP_teachers
          - ip dhcp pool DHCP_schooladmins

    - name: Configure DHCP_schooladmins
      ios_config:
        lines:
          - network 192.168.20.0 255.255.255.0
          - default-router 192.168.20.1
          - dns-server 192.168.20.10
        parents: ip dhcp pool DHCP_schooladmins

    - name: Configure DHCP_teachers
      ios_config:
        lines:
          - network 192.168.21.0 255.255.255.0
          - default-router 192.168.21.1
          - dns-server 192.168.21.10
        parents: ip dhcp pool DHCP_teachers

    - name: Exclude Address (192.168.20.1 - 20.10) & (192.168.21.1 - 21.10)
      ios_config:
        lines:
          - ip dhcp excluded-address 192.168.20.1 192.168.20.10
          - ip dhcp excluded-address 192.168.21.1 192.168.21.10
```

LOUIS ADRIAN MENDOZA.txt - Notepad

— □ ×

Configuration of DHCP for the 2 networks (School Admin & Teachers) in R3

Network Topic 2: Configuration of OSPF in the routers

```
#=====
#----- -- NETWORK-TOPIC-#2 -- -----
#=====

- name: Network Topic 2 (Configuration of OSPF)(R1)
  become_method: enable
  hosts: R1
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Create ospf 10 (R1)
      ios_config:
        lines:
          - router ospf 10

    - name: Configure router ospf 10 on R1
      ios_config:
        lines:
          - router-id 1.1.1.1
          - log-adjacency-changes
          - network 10.0.0.0 0.0.0.3 area 0
          - network 10.0.0.4 0.0.0.3 area 0
          - network 192.168.1.0 0.0.0.255 area 0
          - default-information originate
        parents: router ospf 10
```


LOUIS ADRIAN MENDOZA.txt - Notepad

Configuration of OSPF on R1

```
- name: Network Topic 2 (Configuration of OSPF)(R2)
  become_method: enable
  hosts: R2
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Create ospf 10 (R2)
      ios_config:
        lines:
          - router ospf 10

    - name: Configure router ospf 10 on R2
      ios_config:
        lines:
          - router-id 2.2.2.2
          - log-adjacency-changes
          - network 10.0.0.0 0.0.0.3 area 0
          - network 192.168.10.0 0.0.0.255 area 0
        parents: router ospf 10
```


 LOUIS ADRIAN MENDOZA.txt - Notepad

Configuration of OSPF on R2

```
- name: Network Topic 2 (Configuration of OSPF)(R3)
  become_method: enable
  hosts: R3
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Create ospf 10 (R3)
      ios_config:
        lines:
          - router ospf 10

    - name: Configure router ospf 10 on R3
      ios_config:
        lines:
          - router-id 3.3.3.3
          - log-adjacency-changes
          - network 10.0.0.4 0.0.0.3 area 0
          - network 192.168.20.0 0.0.0.255 area 0
          - network 192.168.21.0 0.0.0.255 area 0
        parents: router ospf 10
```

 LOUIS ADRIAN MENDOZA.txt - Notepad

Configuration of OSPF on R3

Network Topic 3: Application of Extended ACL for the network

```
#=====
#----- - NETWORK-TOPIC-#3 - -----
#=====

- name: Network Topic 3 (Extended ACL configuration)
  become_method: enable
  hosts: R2
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Configure Extended ACL on R2
      ios_config:
        lines:
          - access-list 101 deny ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
          - access-list 101 permit ip any any

    - name: Add ACL 101 inbound on int f0/0
      ios_config:
        lines:
          - ip access-group 101 in
        parents: interface FastEthernet0/0
```

LOUIS ADRIAN MENDOZA.txt - Notepad

Application of Extended ACL to deny the Students network to the School Admin Network

Ansible Playbook Output

```
devasc@labvm:~/FINAL-CASE-STUDY$ ansible-playbook playbook.yaml -Kb
BECOME password:

PLAY [Network Topic 1 (DHCP server for management network)(R1)] *****

TASK [Create DHCP pool DHCP_management] *****
changed: [R1]

TASK [Configure DHCP_management] *****
changed: [R1]

TASK [Exclude Address (192.168.1.1 - 1.10)] *****
changed: [R1]

PLAY [Network Topic 1 (DHCP server for computer lab network)(R2)] *****

TASK [Create DHCP pool DHCP_students] *****
changed: [R2]

TASK [Configure DHCP_students] *****
changed: [R2]

TASK [Exclude Address (192.168.10.1 - 10.10)] *****

PLAY [Network Topic 1 (DHCP server for computer lab network)(R2)] *****

TASK [Create DHCP pool DHCP_students] *****
changed: [R2]

TASK [Configure DHCP_students] *****
changed: [R2]

TASK [Exclude Address (192.168.10.1 - 10.10)] *****
changed: [R2]

PLAY [Network Topic 1 (DHCP server for school main office network)(R3)] *****

TASK [Create 2 DHCP pool (DHCP_teachers & DHCP_schooladmins)] *****
changed: [R3]

TASK [Configure DHCP_schooladmins] *****
changed: [R3]

TASK [Configure DHCP_teachers] *****
changed: [R3]

TASK [Exclude Address (192.168.20.1 - 20.10) & (192.168.21.1 - 21.10)] *****
```

```

TASK [Exclude Address (192.168.20.1 - 20.10) & (192.168.21.1 - 21.10)] *****
changed: [R3]

PLAY [Network Topic 2 (Configuration of OSPF)(R1)] *****

TASK [Create ospf 10 (R1)] *****
ok: [R1]

TASK [Configure router ospf 10 on R1] *****
changed: [R1]

PLAY [Network Topic 2 (Configuration of OSPF)(R2)] *****

TASK [Create ospf 10 (R2)] *****
ok: [R2]

TASK [Configure router ospf 10 on R2] *****
changed: [R2]

PLAY [Network Topic 2 (Configuration of OSPF)(R3)] *****

TASK [Create ospf 10 (R3)] *****

PLAY [Network Topic 2 (Configuration of OSPF)(R3)] *****

TASK [Create ospf 10 (R3)] *****
ok: [R3]

TASK [Configure router ospf 10 on R3] *****
changed: [R3]

PLAY [Network Topic 3 (Extended ACL configuration)] *****

TASK [Configure Extended ACL on R2] *****
changed: [R2]

TASK [Add ACL 101 inbound on int f0/0] *****
changed: [R2]

PLAY RECAP *****
R1      : ok=5I  changed=4  unreachable=0  failed=0  skipped=0
R2      : ok=7  changed=6  unreachable=0  failed=0  skipped=0
R3      : ok=6  changed=5  unreachable=0  failed=0  skipped=0

devasc@labvm:~/FINAL-CASE-STUDY$

```

Part 5: Use PyATS and Genie to test connectivity

PyATS and Genie was used for testing the configuration of the network automation we did from the the ansible. We are also able to test the connectivity of the network through PyATS and Genie.

- a. Create a testbed.yml for the devices that will be used for testing.

```
devices:
  R1:
    connections:
      cli:
        ip: 192.168.1.1
        protocol: ssh
    credentials:
      default:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
        username: cisco
      enable:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
    os: ios
    type: ios
  R2:
    connections:
      cli:
        ip: 192.168.10.1
        protocol: ssh
    credentials:
      default:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
        username: cisco
      enable:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
    os: ios
    type: ios
```

LOUIS ADRIAN MENDOZA.txt - Notepad

```
R3:
  connections:
    cli:
      ip: 192.168.20.1
      protocol: ssh
  credentials:
    default:
      password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
      username: cisco
    enable:
      password: '%ENC{w5PDos0Uw5fDosKQwpbCmA==}'
  os: ios
  type: ios
```

LOUIS ADRIAN MENDOZA.txt - Notepad

- b. The following commands was used for testing the network configuration and connectivity:
- `genie parse "show ip ospf" --testbed-file yaml/testbed.yml --devices R1 R2 R3 --output test-ospf`
 - `genie parse "show access-lists" --testbed-file yaml/testbed.yml --devices R1 R2 R3 --output test-acl`
 - `genie parse "show interfaces accounting" --testbed-file yaml/testbed.yml --devices R1 R2 R3 --output test-dhcp`

```
devasc@labvm:~/FINAL-CASE-STUDY/pyats$ genie parse "show ip ospf" --testbed-file yaml/testbed.yml --devices R1 R2 R3 --output tes
t-ospf
Using the default YAML encoding key since no key was specified in configuration.
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN `pyats secret keygen` AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALU
ES.
100%|██████████████████████████████████████████████████████████████████████████████| 1/1 [00:02<00:00, 2.04s/it]
=====+
| Genie Parse Summary for R1 |
=====+
| Connected to R1 |
| - Log: test-ospf/connection_R1.txt |
|-----+
| Parsed command 'show ip ospf' |
| - Parsed structure: test-ospf/R1_show-ip-ospf_parsed.txt |
| - Device Console: test-ospf/R1_show-ip-ospf_console.txt |
|-----+
100%|██████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 2.14it/s]
=====+
| Genie Parse Summary for R2 |
=====+
| Connected to R2 |
| - Log: test-ospf/connection_R2.txt |
|-----+
| Parsed command 'show ip ospf' |
| - Parsed structure: test-ospf/R2_show-ip-ospf_parsed.txt |
| - Device Console: test-ospf/R2_show-ip-ospf_console.txt |
|-----+
```

```
| - Parsed structure: test-ospf/R2_show-ip-ospf_parsed.txt
| - Device Console: test-ospf/R2_show-ip-ospf_console.txt
|-----|
100%| 1/1 [00:00<00:00, 2.29it/s]
+=====+
| Genie Parse Summary for R3
+=====+
| Connected to R3
| - Log: test-ospf/connection_R3.txt
|-----|
| Parsed command 'show ip ospf'
| - Parsed structure: test-ospf/R3_show-ip-ospf_parsed.txt
| - Device Console: test-ospf/R3_show-ip-ospf_console.txt
|-----|
```

```
devasc@labvm:~/FINAL-CASE-STUDY/pyats$ genie parse "show access-lists" --testbed-file yaml/testbed.yml --devices R1 R2 R3 --output test-acl
```

Using the default YAML encoding key since no key was specified in configuration.
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN `pyats secret keygen` AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALUES.

```
100%|██████████████████████████████████████████████████████████████████████████████| 1/1 [00:02<00:00, 2.19s/it]
```

```
+=====+
| Genie Parse Summary for R1 |
+=====+
```

```
| Connected to R1 |
| - Log: test-acl/connection_R1.txt |
|-----|
| Parsed command 'show access-lists' but it returned empty |
| - Device Console: test-acl/R1_show-access-lists_console.txt |
|-----|
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 2.04it/s]
```

```
+=====+
| Genie Parse Summary for R2 |
+=====+
```

```
| Connected to R2 |
| - Log: test-acl/connection_R2.txt |
|-----|
| Parsed command 'show access-lists' |
| - Parsed structure: test-acl/R2_show-access-lists_parsed.txt |
| - Device Console: test-acl/R2_show-access-lists_console.txt |
|-----|
```

```
| - Parsed structure: test-acl/R2_show-access-lists_parsed.txt
| - Device Console: test-acl/R2_show-access-lists_console.txt
|-----|-----| 1/1 [00:00<00:00, 2.31it/s]
+=====+
| Genie Parse Summary for R3
+=====+
| Connected to R3
| - Log: test-acl/connection_R3.txt
|-----|-----|
| Parsed command 'show access-lists' but it returned empty
| - Device Console: test-acl/R3_show-access-lists_console.txt
|-----|-----|

devasc@labvm:~/FINAL-CASE-STUDY/pyats$ genie parse "show interfaces accounting" --testbed-file yaml/testbed.yml --devices R1 R2 R
3 --output test-dhcp
Using the default YAML encoding key since no key was specified in configuration.
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN 'pyats secret keygen' AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALU
ES.
100%|-----| 1/1 [00:01<00:00, 1.84s/it]
+=====+
| Genie Parse Summary for R1
+=====+
| Connected to R1
| - Log: test-dhcp/connection_R1.txt
|-----|-----|
| Parsed command 'show interfaces accounting'
| - Parsed structure: test-dhcp/R1_show-interfaces-accounting_parsed.txt
| - Device Console: test-dhcp/R1_show-interfaces-accounting_console.txt
|-----|-----|
100%|-----| 1/1 [00:00<00:00, 1.78it/s]
+=====+
| Genie Parse Summary for R2
+=====+
| Connected to R2
| - Log: test-dhcp/connection_R2.txt
|-----|-----|
| Parsed command 'show interfaces accounting'
| - Parsed structure: test-dhcp/R2_show-interfaces-accounting_parsed.txt
| - Device Console: test-dhcp/R2_show-interfaces-accounting_console.txt
|-----|-----|
| - Parsed structure: test-dhcp/R2_show-interfaces-accounting_parsed.txt
| - Device Console: test-dhcp/R2_show-interfaces-accounting_console.txt
|-----|-----|
100%|-----| 1/1 [00:00<00:00, 1.65it/s]
+=====+
| Genie Parse Summary for R3
+=====+
| Connected to R3
| - Log: test-dhcp/connection_R3.txt
|-----|-----|
| Parsed command 'show interfaces accounting'
| - Parsed structure: test-dhcp/R3_show-interfaces-accounting_parsed.txt
| - Device Console: test-dhcp/R3_show-interfaces-accounting_console.txt
|-----|-----|
```

Conclusion

This case study really challenged my skills as a computer engineer, as I recall and apply all my knowledge from the past lesson on Computer Networks 1-3 and System administration. GNS3 acts as a virtual network for the scenario of the case study, and it was able to connect devices and acts as a real network for the scenario. Application of the Ansible for network automation makes things easier for the network engineer, as we can create playbooks that can run and automate task for setting up or configuring a network device or even a host machine. Lastly for the PyATS and Genie, it was a great tool for testing the configuration and connectivity of the network. This activity was great for learning new things related to automation and computer networks, it is also a great activity to test and apply skills learned from the past lessons.

GITHUB LINK:

<https://github.com/lamendoza-tip/FINAL-CASE-STUDY>

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