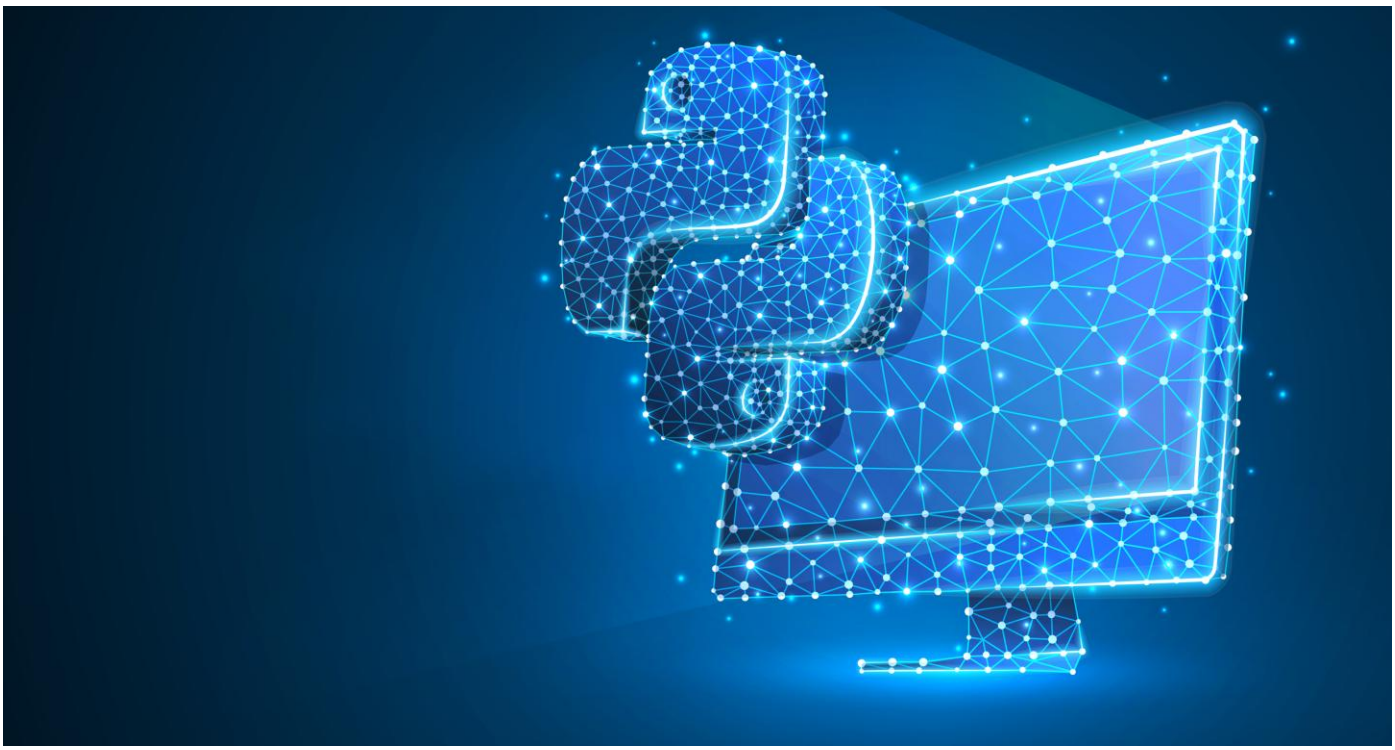
NetPilot – Network Automation Project Report



Introduction

NetPilot is a complete virtual network lab designed to automate network configuration and simulate real-world enterprise environments using:

- Python scripting

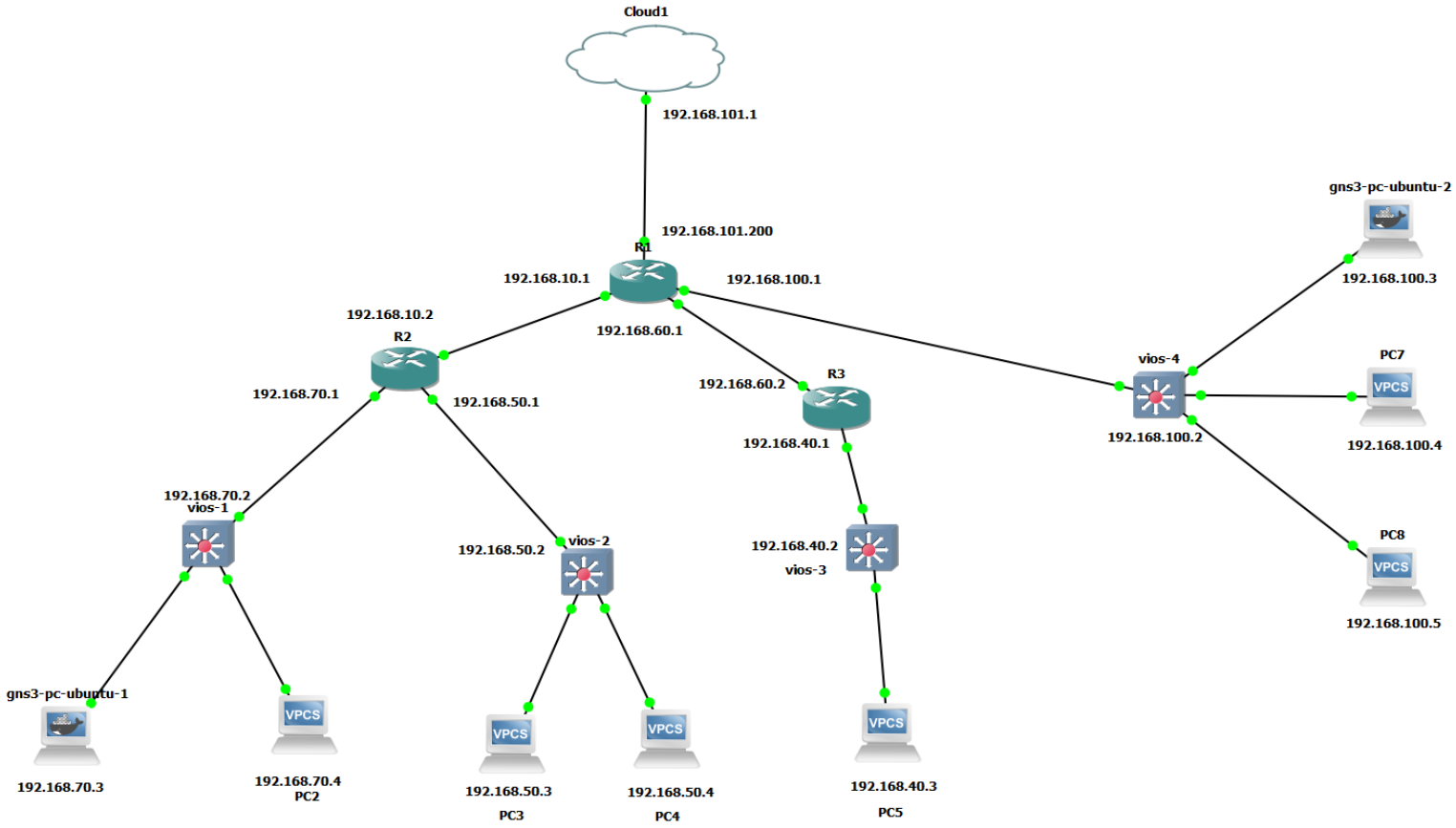
- GNS3 virtual routers

- Docker-based PCs

The goal is to streamline provisioning, configuration, and management through NetOps (Network Automation) concepts.

Project Overview

GNS3 allows full connectivity (SSH and ping) between all network elements as if they were on the same network.



📌 Cloud1 represents the host PC, with IP: 192.168.101.1.

```
PS C:\Users\asabe> ipconfig | Select-String -Context 0,5 "VMware Network Adapter VMnet1"

> Ethernet adapter VMware Network Adapter VMnet1:

     Connection-specific DNS Suffix  . :
     Link-local IPv6 Address . . . . . : fe80::1ccb:82e6:15a0:ffc%16
     IPv4 Address. . . . . . . . . . . : 192.168.101.1
     Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

Static Routing on PC:

```
route -p add 192.168.10.0 mask 255.255.255.0 192.168.101.200

route -p add 192.168.60.0 mask 255.255.255.0 192.168.101.200
```

```
PS C:\Users\asabe> ping 192.168.10.2   #for R2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time=25ms TTL=254
Reply from 192.168.10.2: bytes=32 time=47ms TTL=254
Reply from 192.168.10.2: bytes=32 time=47ms TTL=254
Reply from 192.168.10.2: bytes=32 time=47ms TTL=254

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 25ms, Maximum = 47ms, Average = 41ms
PS C:\Users\asabe> ping 192.168.60.2   #for R3

Pinging 192.168.60.2 with 32 bytes of data:
Reply from 192.168.60.2: bytes=32 time=53ms TTL=254
Reply from 192.168.60.2: bytes=32 time=47ms TTL=254
Reply from 192.168.60.2: bytes=32 time=48ms TTL=254
Reply from 192.168.60.2: bytes=32 time=47ms TTL=254

Ping statistics for 192.168.60.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 47ms, Maximum = 53ms, Average = 48ms
PS C:\Users\asabe>
```

Enabling SSH on Routers

```
en
conf t
hostname R1
ip domain-name gns3.local
crypto key generate rsa   # 1024 bits
username admin privilege 15 secret your_password
interface FastEthernet0/0
ip address 192.168.10.1 255.255.255.0
no shutdown
ip ssh version 2
line vty 0 4
login local
transport input ssh
end
wr
```

Repeat for R2, R3, and SW1-4. Then test connectivity from the PC using ping.

Switch Configuration

```
en
conf t
no ip routing                  # Very important
ip default-gateway 192.168.100.1    # Very important
interface vlan1
```

ip address 192.168.100.2 255.255.255.0
no shutdown
interface gigabitEthernet 0/0
switchport mode access
no shutdown
exit

```
PS C:\Users\asabe> ping 192.168.70.2   #for sw1

Pinging 192.168.70.2 with 32 bytes of data:
Reply from 192.168.70.2: bytes=32 time=62ms TTL=253
Reply from 192.168.70.2: bytes=32 time=95ms TTL=253
Reply from 192.168.70.2: bytes=32 time=63ms TTL=253
Reply from 192.168.70.2: bytes=32 time=63ms TTL=253

Ping statistics for 192.168.70.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 62ms, Maximum = 95ms, Average = 70ms
PS C:\Users\asabe> ping 192.168.50.2   #for sw2

Pinging 192.168.50.2 with 32 bytes of data:
Reply from 192.168.50.2: bytes=32 time=163ms TTL=253
Reply from 192.168.50.2: bytes=32 time=62ms TTL=253
Reply from 192.168.50.2: bytes=32 time=63ms TTL=253
Reply from 192.168.50.2: bytes=32 time=95ms TTL=253

Ping statistics for 192.168.50.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 62ms, Maximum = 163ms, Average = 95ms
PS C:\Users\asabe> ping 192.168.40.2   #for sw3

Pinging 192.168.40.2 with 32 bytes of data:
Reply from 192.168.40.2: bytes=32 time=65ms TTL=253
Reply from 192.168.40.2: bytes=32 time=63ms TTL=253
Reply from 192.168.40.2: bytes=32 time=64ms TTL=253
```

Note: This should be done after configuring OSPF.

Static Routing + OSPF + DHCP Scripting

Static route example:
en
conf t
ip route 192.168.10.0 255.255.255.0 192.168.10.2

```
R1#show running-config | include ip route
ip route 192.168.10.0 255.255.255.0 192.168.10.2
ip route 192.168.60.0 255.255.255.0 192.168.60.2
```

```
R2#show running-config | include ip route
ip route 192.168.60.0 255.255.255.0 192.168.10.1
ip route 192.168.101.0 255.255.255.0 192.168.10.1
```

```
R3#show running-config | include ip route
ip route 192.168.10.0 255.255.255.0 192.168.60.1
ip route 192.168.101.0 255.255.255.0 192.168.60.1
```

OSPF via Python (Netmiko):

- Router configs are stored as YAML files.

- YAML passed to Python script.

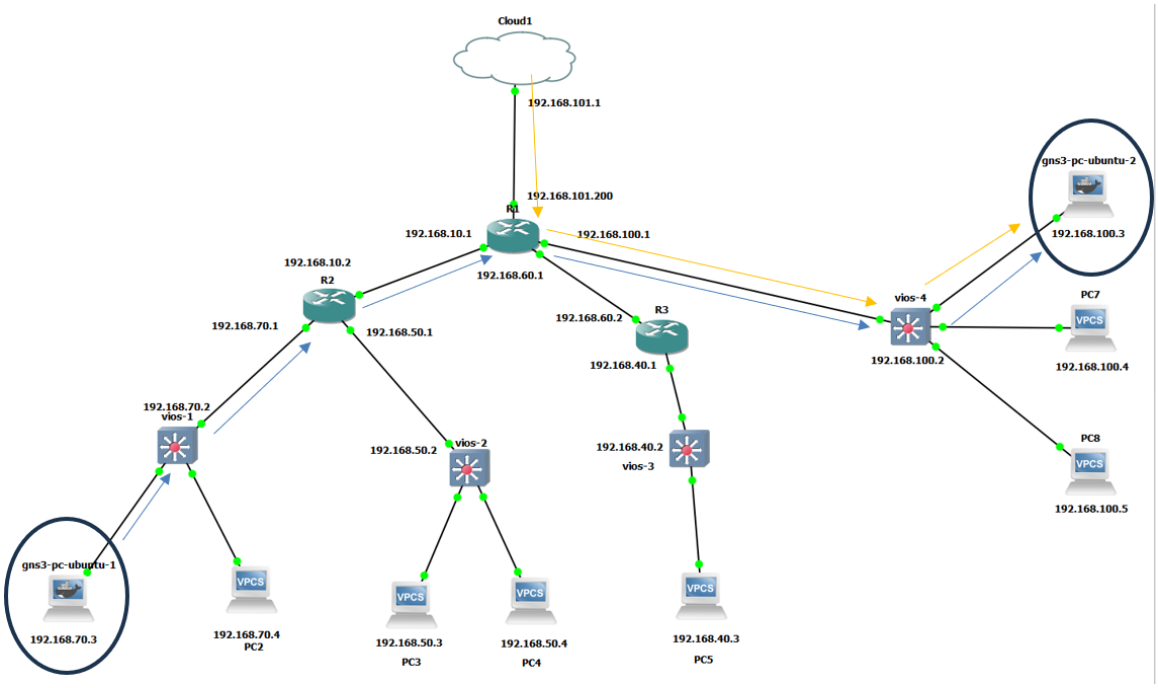
```
R1#show run | section ospf
router ospf 1
 network 10.10.10.1 0.0.0.0 area 0
 network 10.10.10.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
 network 192.168.60.0 0.0.0.255 area 0
 network 192.168.100.0 0.0.0.255 area 0
 network 192.168.101.0 0.0.0.255 area 0
```

- Same applies to DHCP via Paramiko.

```
R1#show running-config | section dhcp
ip dhcp excluded-address 192.168.100.1 192.168.100.10
ip dhcp pool DHCP_POOL
 network 192.168.100.0 255.255.255.0
 default-router 192.168.100.1
 dns-server 8.8.8.8
 lease 0 12
```

Source code: https://github.com/AhmedSabet0/network

Hosting a Web Application



1. Build Docker container:

```
FROM ubuntu:22.04

RUN apt update && apt install -y iproute2 iputils-ping net-tools curl vim apache2 tcpdump isc-dhcp-client telnet netcat sudo systemctl

RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf

CMD ["bash"]
```

2. Build it inside GNS3 VM.

Configure Static IP for gns3-pc-ubuntu-2

Edit /etc/network/interfaces:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
 address 192.168.100.3
 netmask 255.255.255.0
 gateway 192.168.100.1
 dns-nameservers 8.8.8.8 8.8.4.4
```

Start Apache:

```
sudo service apache2 start
```

Configure DHCP for gns3-pc-ubuntu-1

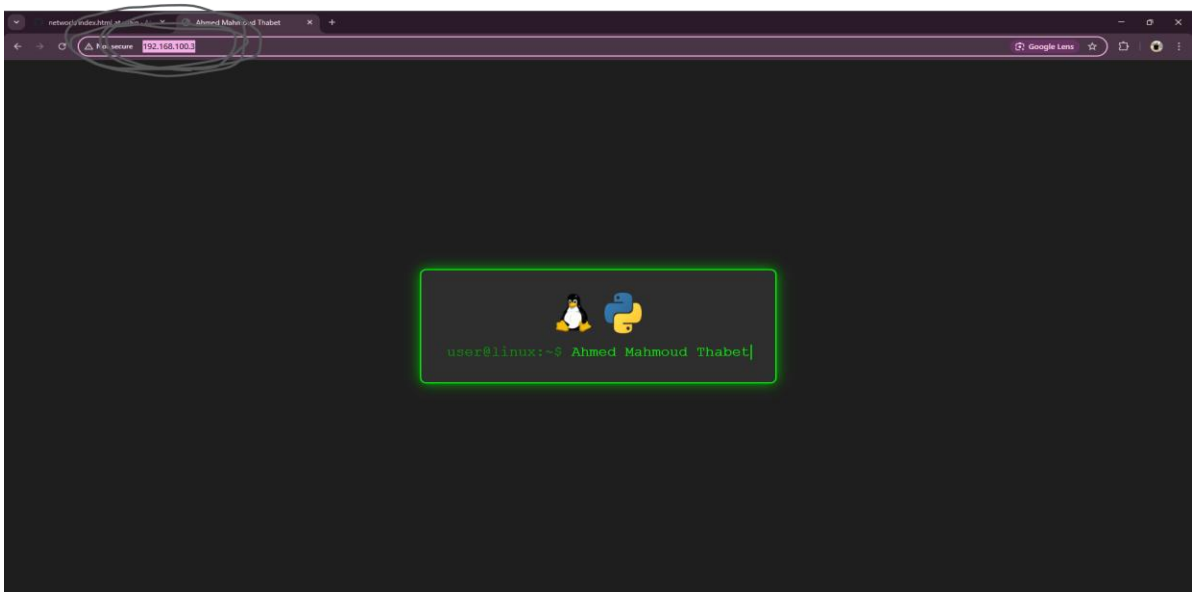
Edit /etc/network/interfaces:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

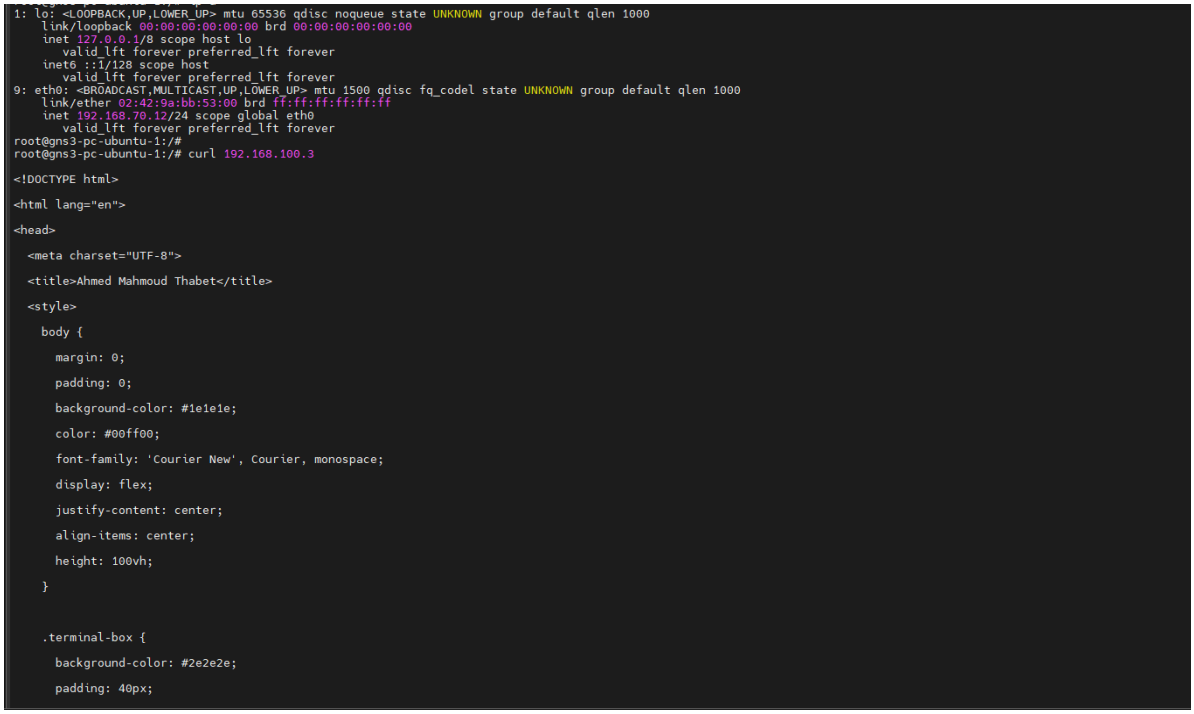
```
Connected to 192.168.101.128.
Entering character mode
Escape character is '^]'.
gns3-pc-ubuntu-1 console is now available... Press RETURN to get started.
ip: RTNETLINK answers: File exists
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 192.168.70.12
udhcpc: lease of 192.168.70.12 obtained, lease time 43200
root@gns3-pc-ubuntu-1:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
9: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 02:42:9a:bb:53:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.70.12/24 scope global eth0
       valid_lft forever preferred_lft forever
root@gns3-pc-ubuntu-1:/#
root@gns3-pc-ubuntu-1:/# ps aux | grep dhclient

root          68  0.0  0.0   3472  1696 pts/0    S+   09:49   0:00 grep --color=auto dhclient
root@gns3-pc-ubuntu-1:/#
root@gns3-pc-ubuntu-1:/#
```

Testing Access

From real PC and gns3-pc-ubuntu-1 (DHCP client):



curl http://192.168.100.3



You should see the HTML content served by Apache.

Problems and how to solve

PC is unable to ping the vIOS switch. Upon investigation, it appears there is a **duplex mismatch** between the switch port and the PC NIC.

interface FastEthernet1/1

duplex auto

speed auto

show interfaces FastEthernet1/0 | include duplex|speed

Disable auto-negotiation and manually set:

Speed: 100 Mbps

Duplex: Full

```
SW1(config)# interface GigabitEthernet0/0
SW1(config-if)# no negotiation auto
SW1(config-if)# speed 100
SW1(config-if)# duplex full
show interfaces GigabitEthernet0/0 | include Duplex|Speed
```

pc cannot ping vios switch

For L2-only switch (vIOS-L2), make sure you have these settings:

```
no ip routing                  # Very important
ip default-gateway 192.168.100.1    # Very important
```

These are mandatory for pinging the switch from a device outside its VLAN or subnet.

Conclusion

This project showcases how NetOps can automate:

- Device provisioning
- Network routing
- Service hosting (web, DHCP)
- Testing connectivity with virtual topologies

Source code: https://github.com/AhmedSabet0/network