Melbourne Institute of Technology

Sydney

MN521

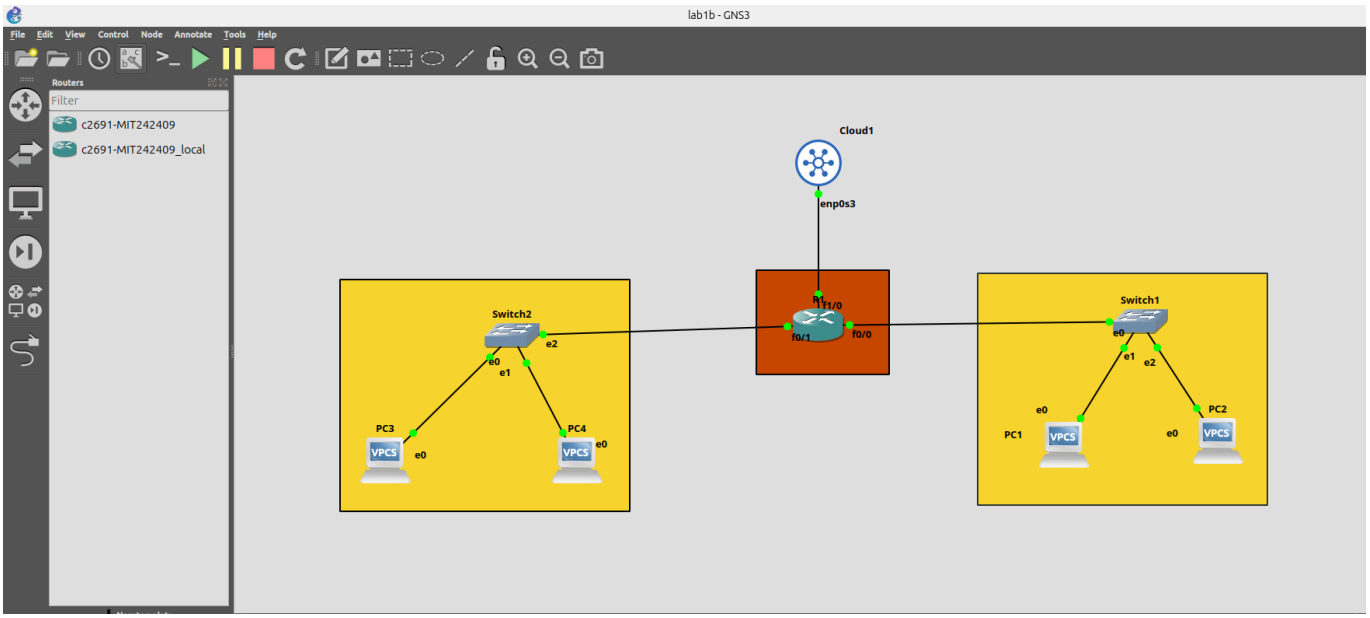
Network Automation

Lab 03

Student Name: Dilip Sapkota

Student ID: MIT242409

1. Network Topology



2. Configuration of Router Interfaces

```
R1#
R1#sh ip int brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            192.168.1.1     YES NVRAM  up                    up

FastEthernet0/1            192.168.2.1     YES NVRAM  up                    up

FastEthernet1/0            10.0.2.16       YES DHCP   up                    up
```

3. Configuring IP on PCs

```
PC1> ip 192.168.1.2/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.2 255.255.255.0 gateway 192.168.1.1
```

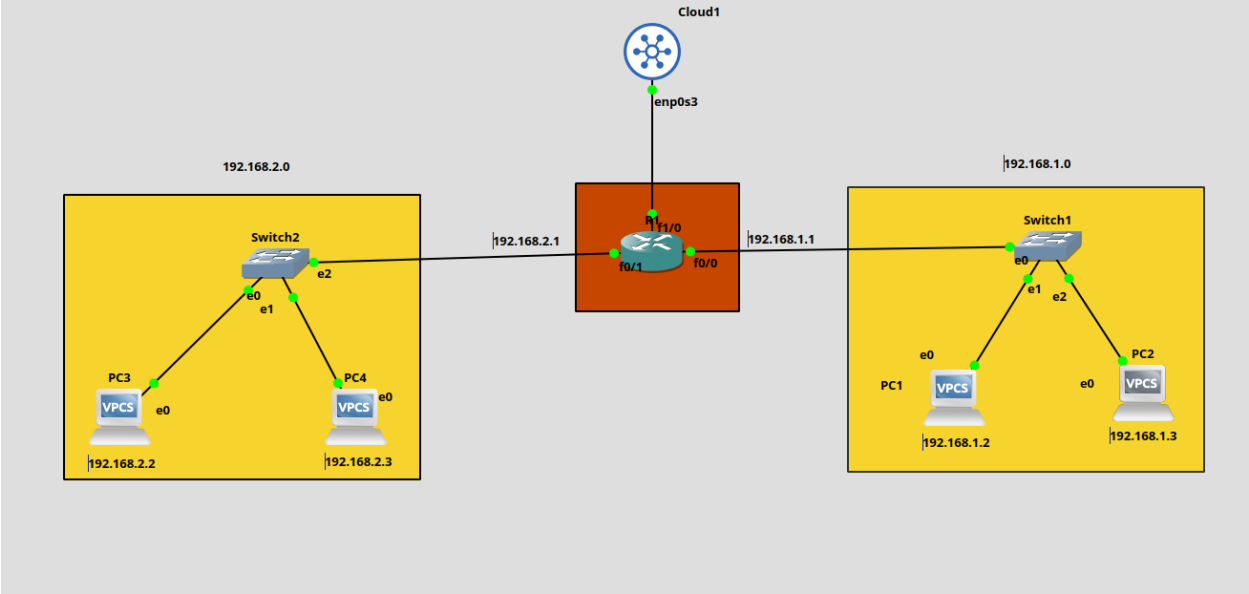
```
PC2> ip 192.168.1.3/24 192.168.1.1
Checking for duplicate address...
PC2 : 192.168.1.3 255.255.255.0 gateway 192.168.1.1
```

```
PC3> ip 192.168.2.2/24 192.168.2.1
Checking for duplicate address...
PC3 : 192.168.2.2 255.255.255.0 gateway 192.168.2.1
```

```
PC4> ip 192.168.2.3/24 192.168.2.1
Checking for duplicate address...
PC4 : 192.168.2.3 255.255.255.0 gateway 192.168.2.1
```

4. Pinging across PCs on Different Subnets

```
192.168.1.2 icmp_seq=1 timeout
84 bytes from 192.168.1.2 icmp_seq=2 ttl=63 time=26.008 ms
84 bytes from 192.168.1.2 icmp_seq=3 ttl=63 time=21.698 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=63 time=25.520 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=63 time=22.722 ms
```



Python Script for configuring interface Fa0/1

```
 GNU nano 7.2                                                    configure_rou
#!/usr/bin/python3

from netmiko import ConnectHandler

def configure_interface(router_ip, username, password, interface, ip_address, subnet_mask):
    # Define device parameters
    device = {
        'device_type': 'cisco_ios',
        'ip':'10.0.60.8',
        'username':'cisco',
        'password':'cisco',
    }

    # Establish an SSH connection to the device
    connection = ConnectHandler(**device)

    # Define the configuration commands
    config_commands = [
        'interface ' + interface,
        'ip address ' + ip_address + ' ' + subnet_mask,
        'no shutdown'
    ]

    # Send the configuration commands to the device
    output = connection.send_config_set(config_commands)

    # Close the SSH connection
    connection.disconnect()

    return output


# Example usage:
output = configure_interface(
    '10.0.60.8',     # Router IP
    'cisco',           # Username
    'cisco',          # Password
    'FastEthernet0/1', # Interface
    '192.168.2.1',   # IP address
    '255.255.255.0'    # Subnet mask
)
```

Executing Python Script to Configure a Router Interface

```
ubuntu@ubuntu2204:~$ python3 configure_router.py
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface FastEthernet0/1
R1(config-if)#ip address 192.168.2.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#end
R1#
```

Router Interface Configured on Interface

```
R1#sh ip int brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            192.168.1.1     YES NVRAM  up                    up

FastEthernet0/1            192.168.2.1     YES NVRAM  up                    up

FastEthernet1/0            10.0.60.8       YES manual up                    up
```