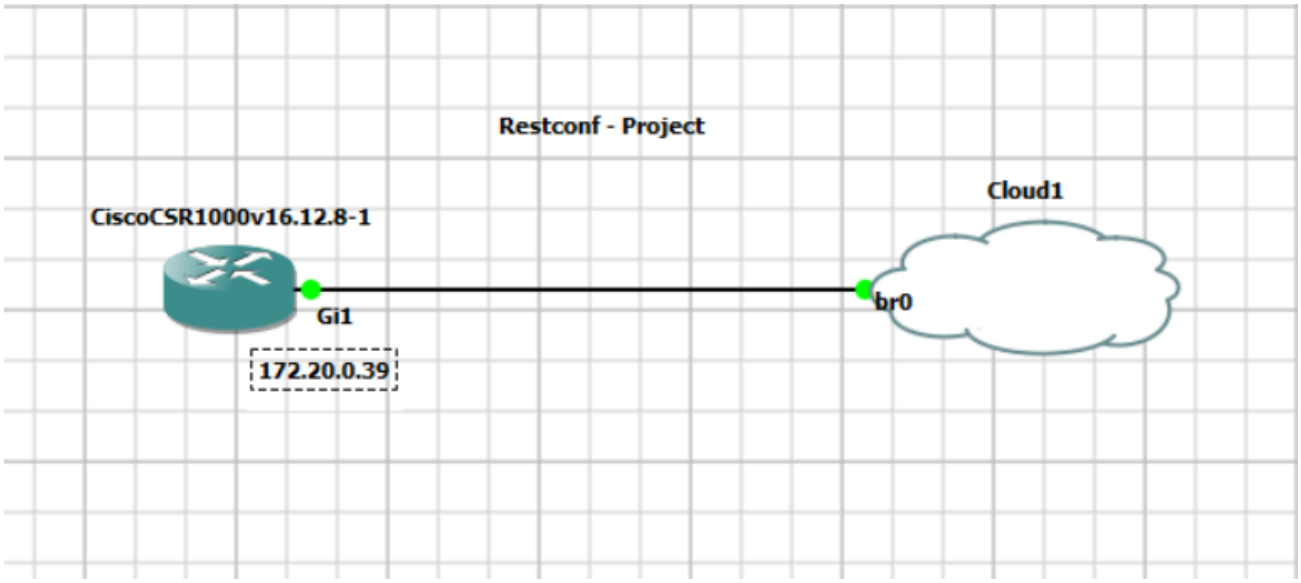
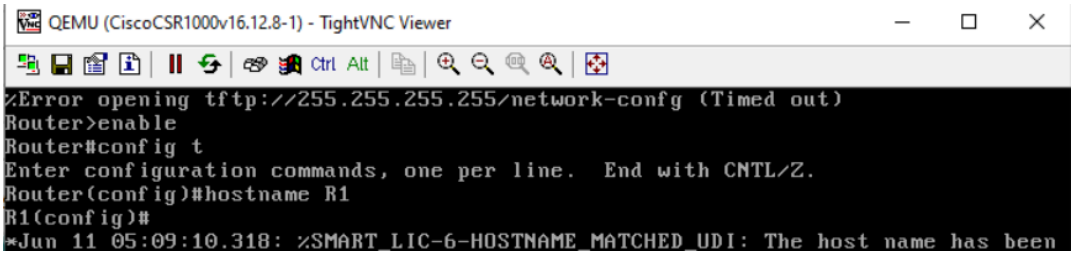
Develop Python scripts using the requests library to perform RESTCONF operations:
GET : Retrieves configuration and operational data.
POST : Create new resources.
PUT : Updates existing resources.
DELETE : Remove resources.

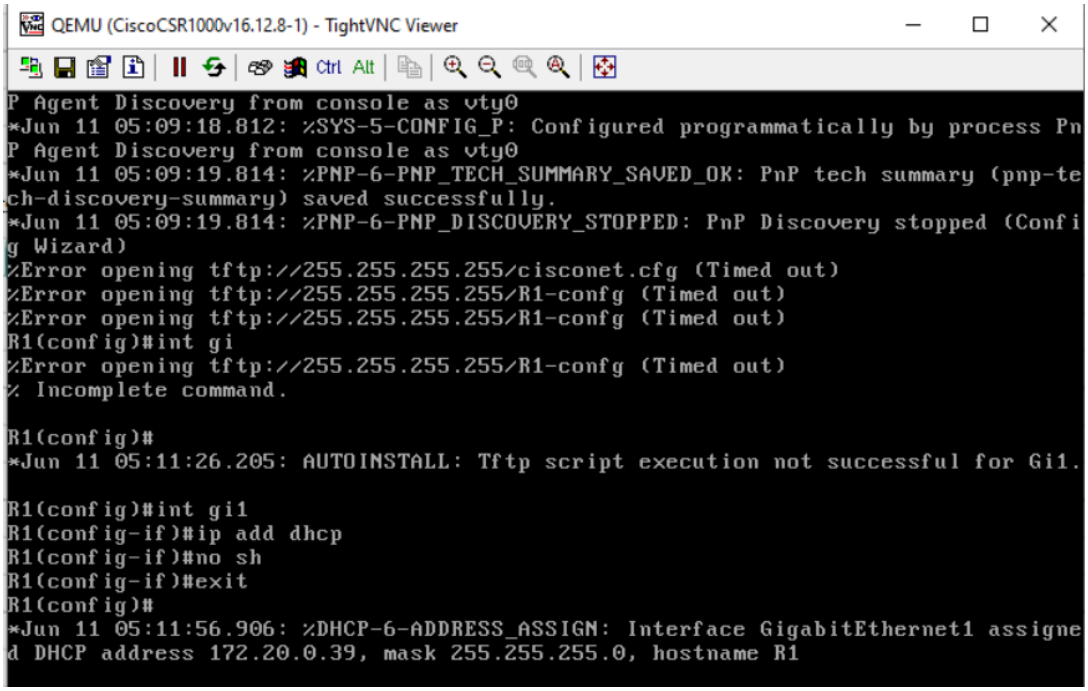
RESCONF Topology



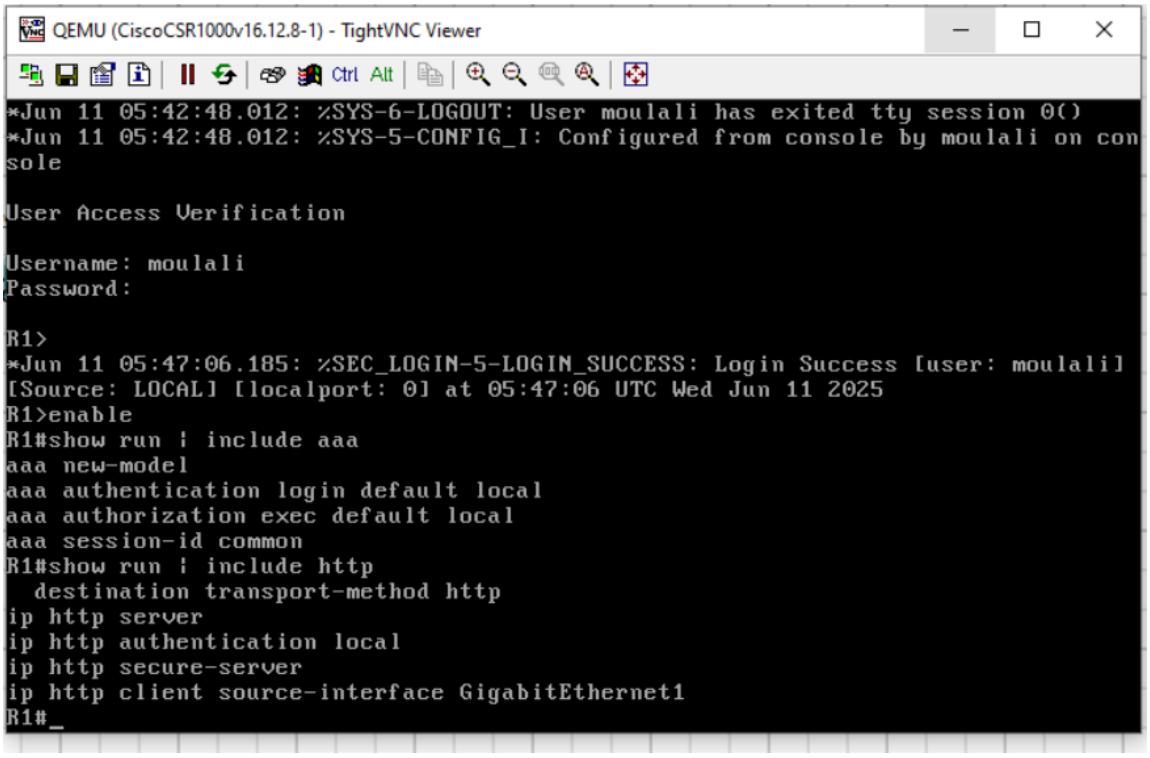
Changing hostname



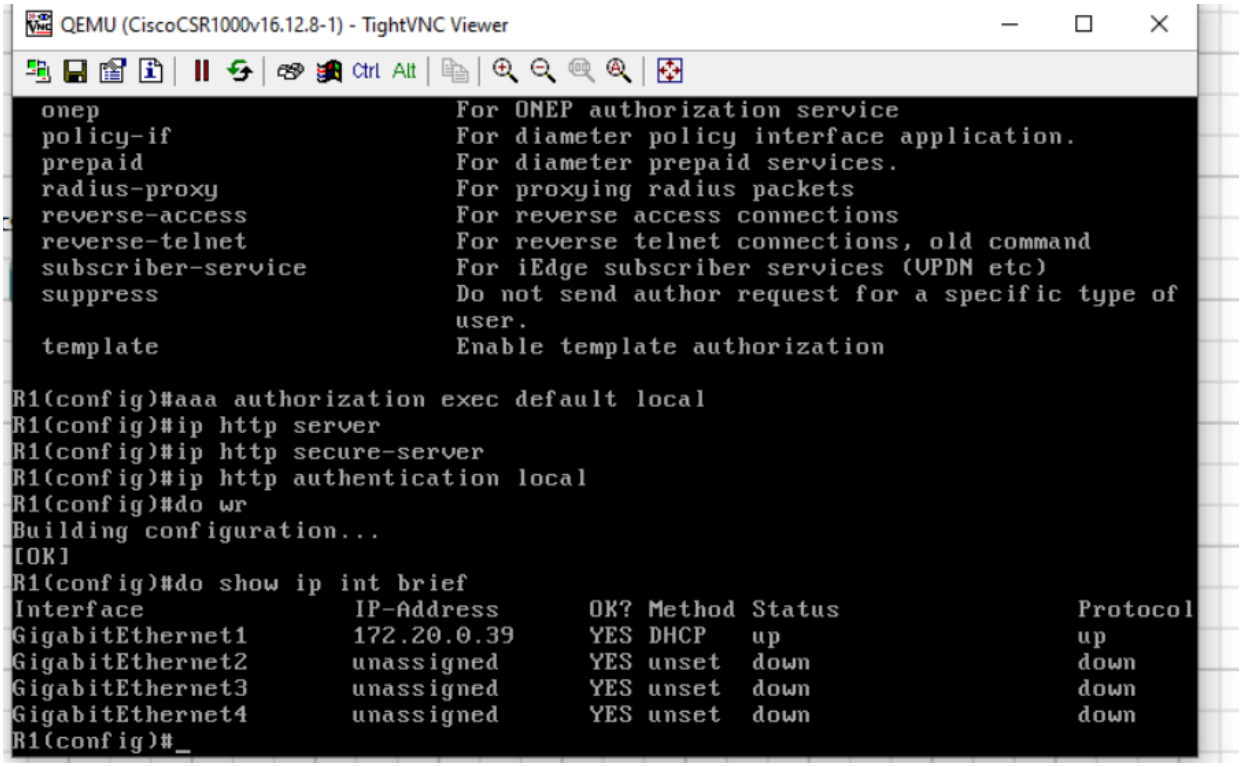
Adding DHCP



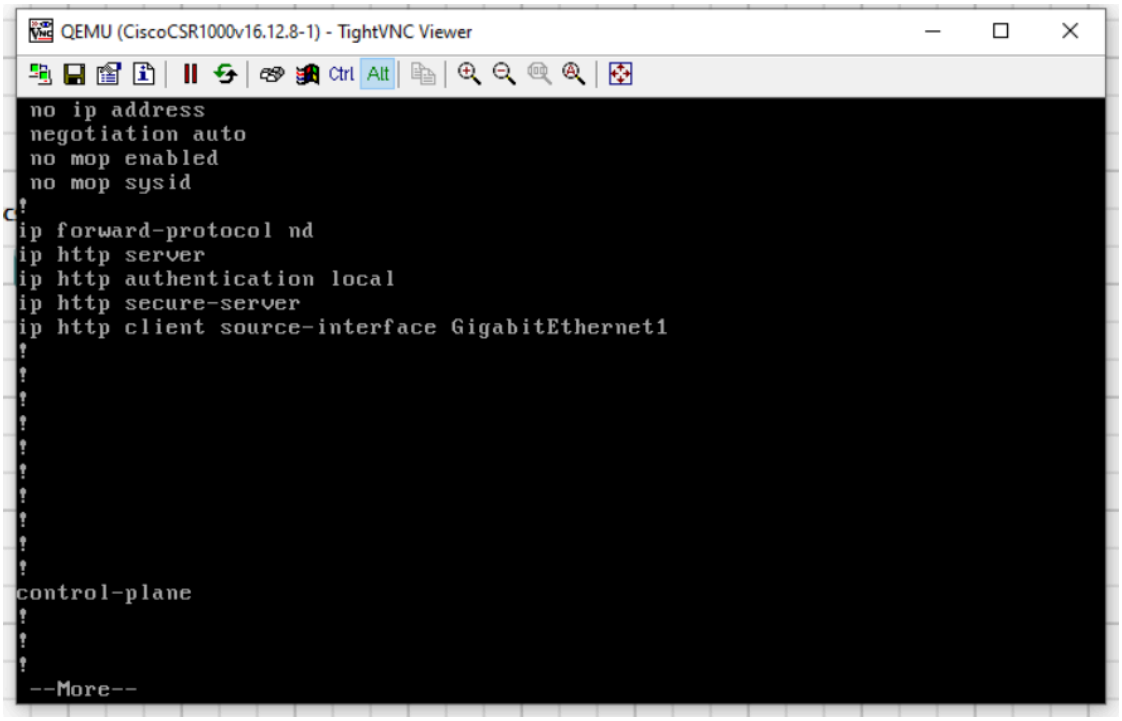
Show run | include aaa and show run | include http



Show ip int brief



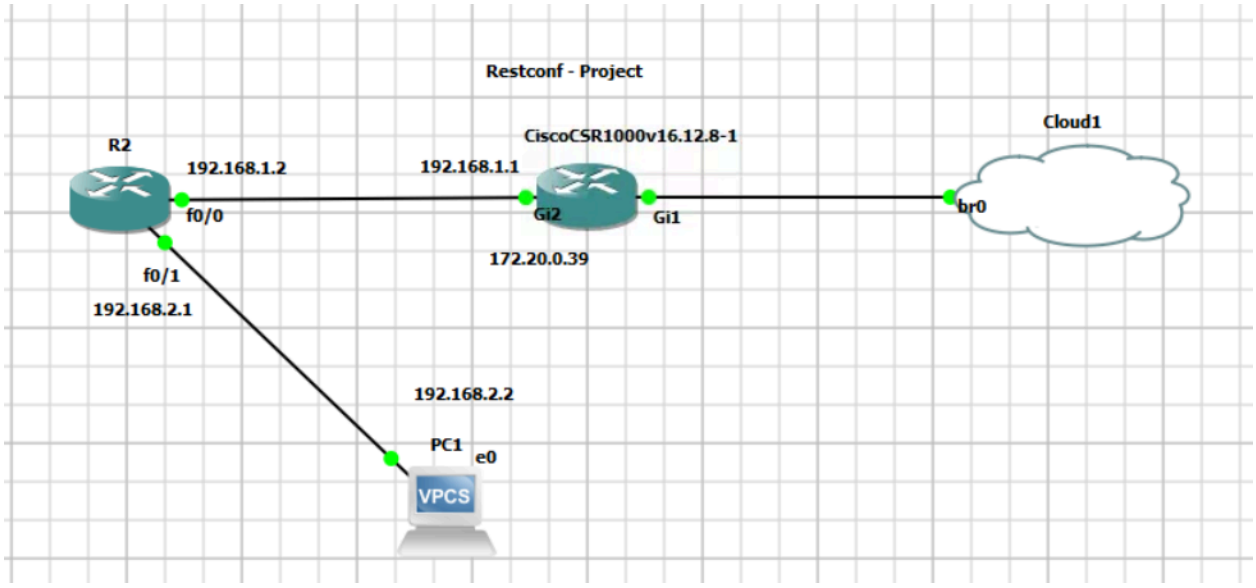
Show run



Adding restconf to the end of configuration

```
User Access Verification

Username: moulali
Password:

R1>en
*Jun 11 10:04:44.465: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: moulali]
[Source: LOCAL] [localport: 0] at 10:04:44 UTC Wed Jun 11 2025
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#restconf
R1(config)#
*Jun 11 10:05:03.183: %PSD_MOD-5-DMI_NOTIFY_RESTCONF_START: R0/0: psd: PSD/DMI:
restconf server has been notified to start
R1(config)#do wr
Building configuration...
_
```

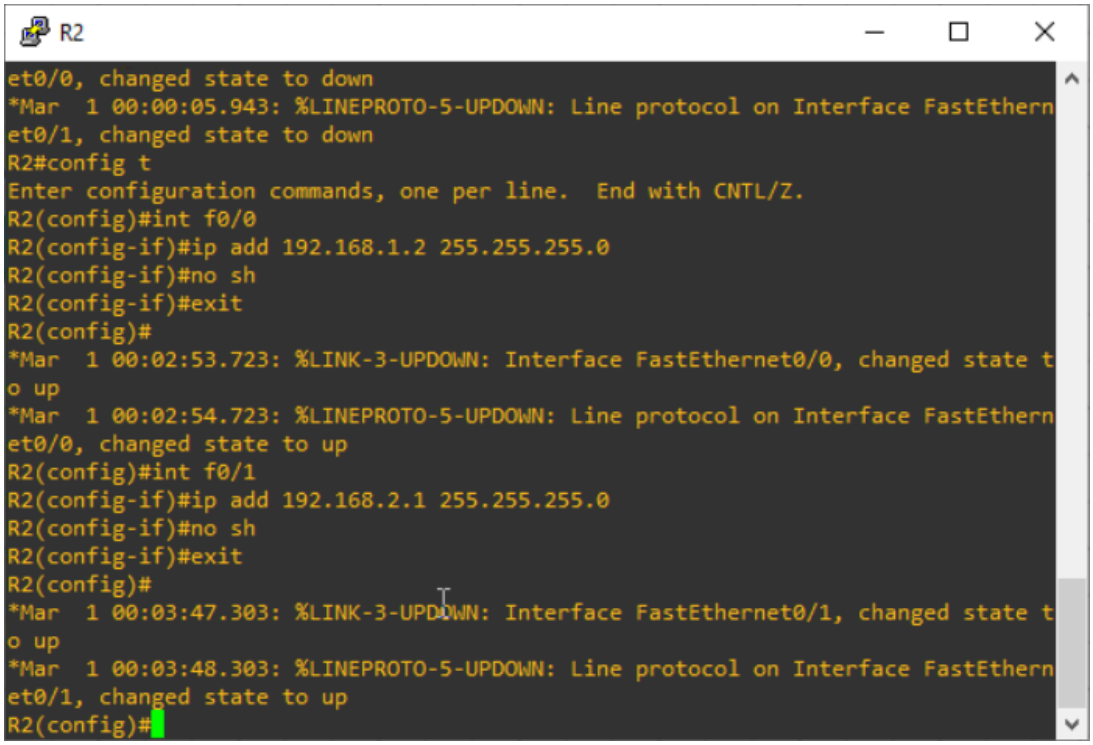
Updated Topology



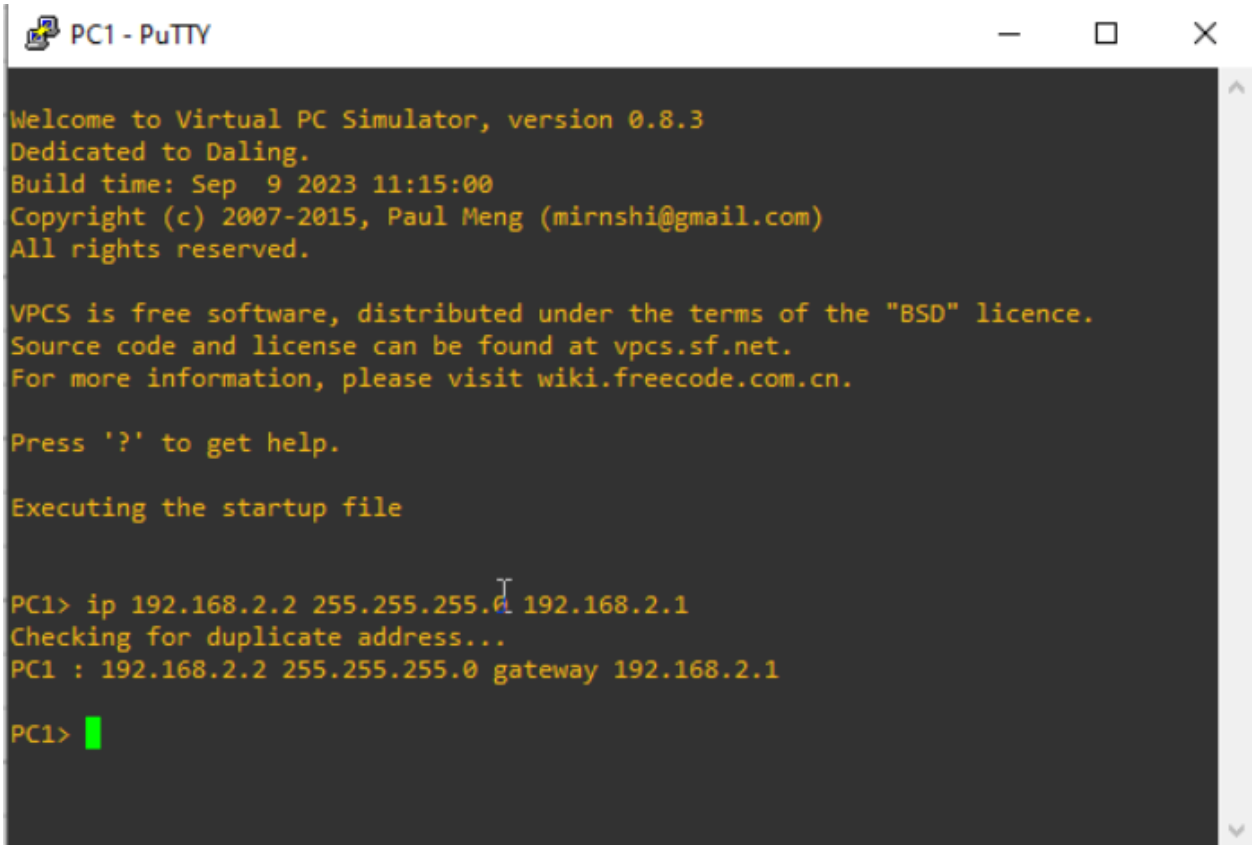
R2 Configurations

```
User Access Verification

Username: moulali
Password:

R1>
*Jun 11 15:32:06.226: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: moulali]
[Source: LOCAL] [localport: 0] at 15:32:06 UTC Wed Jun 11 2025
R1>enable
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int gi2
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#_
```

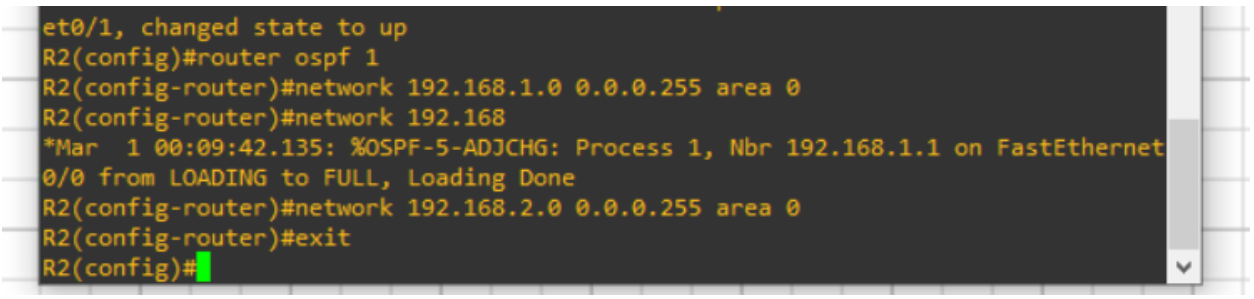
Adding ip address



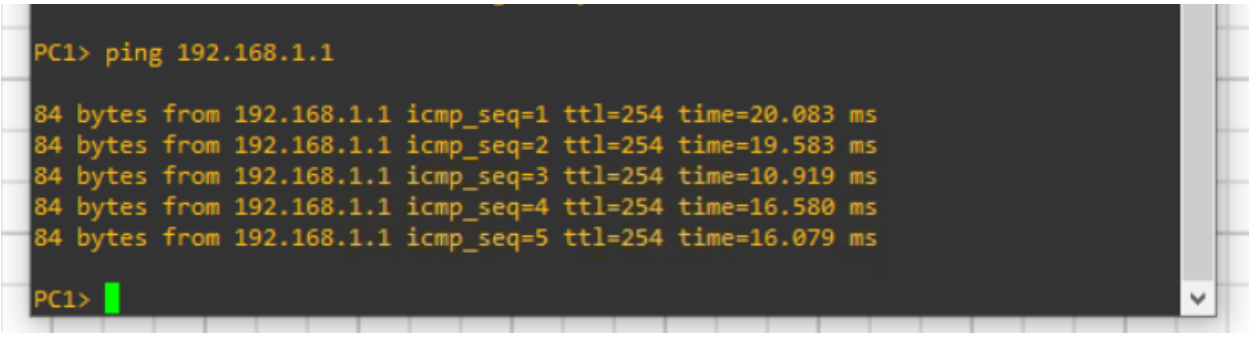
PC ip address



Configuring OSPF

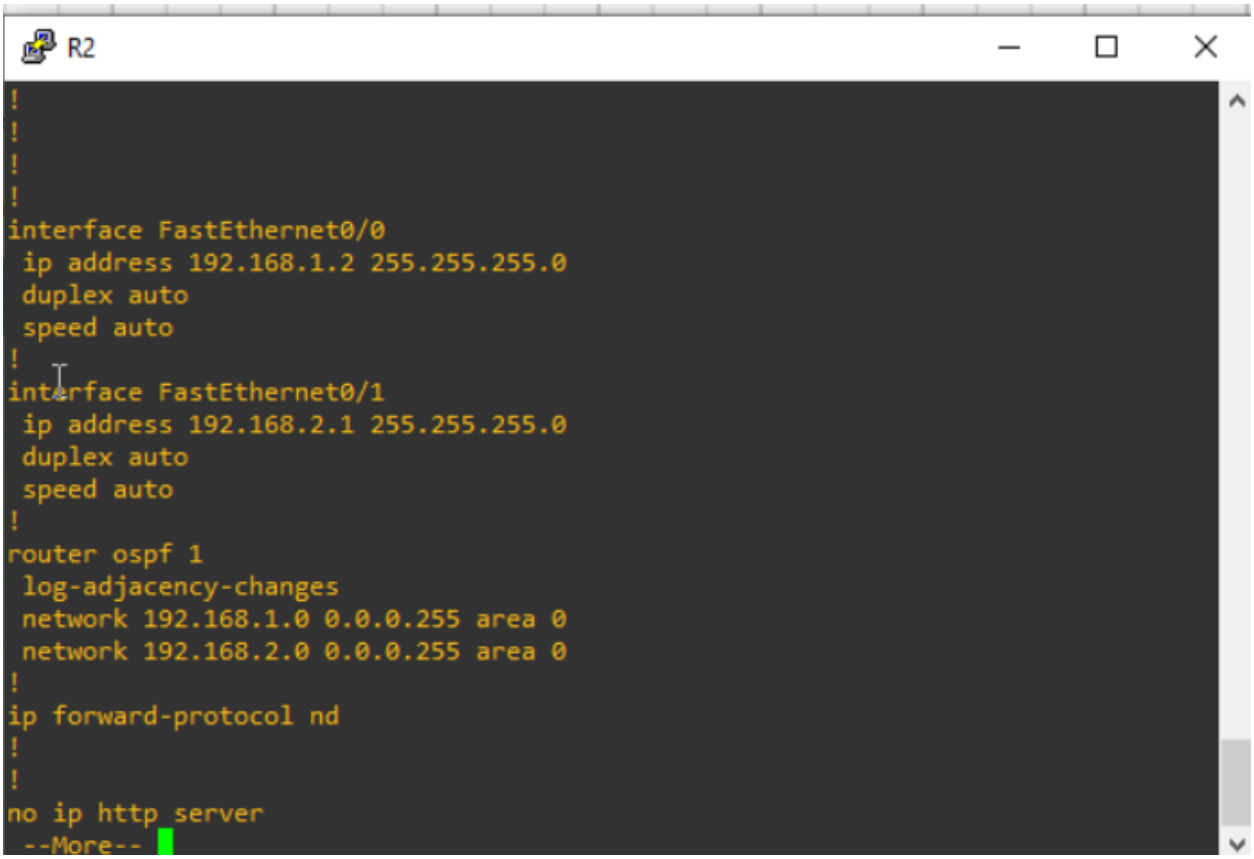


ping



Show ip int brief

Show run

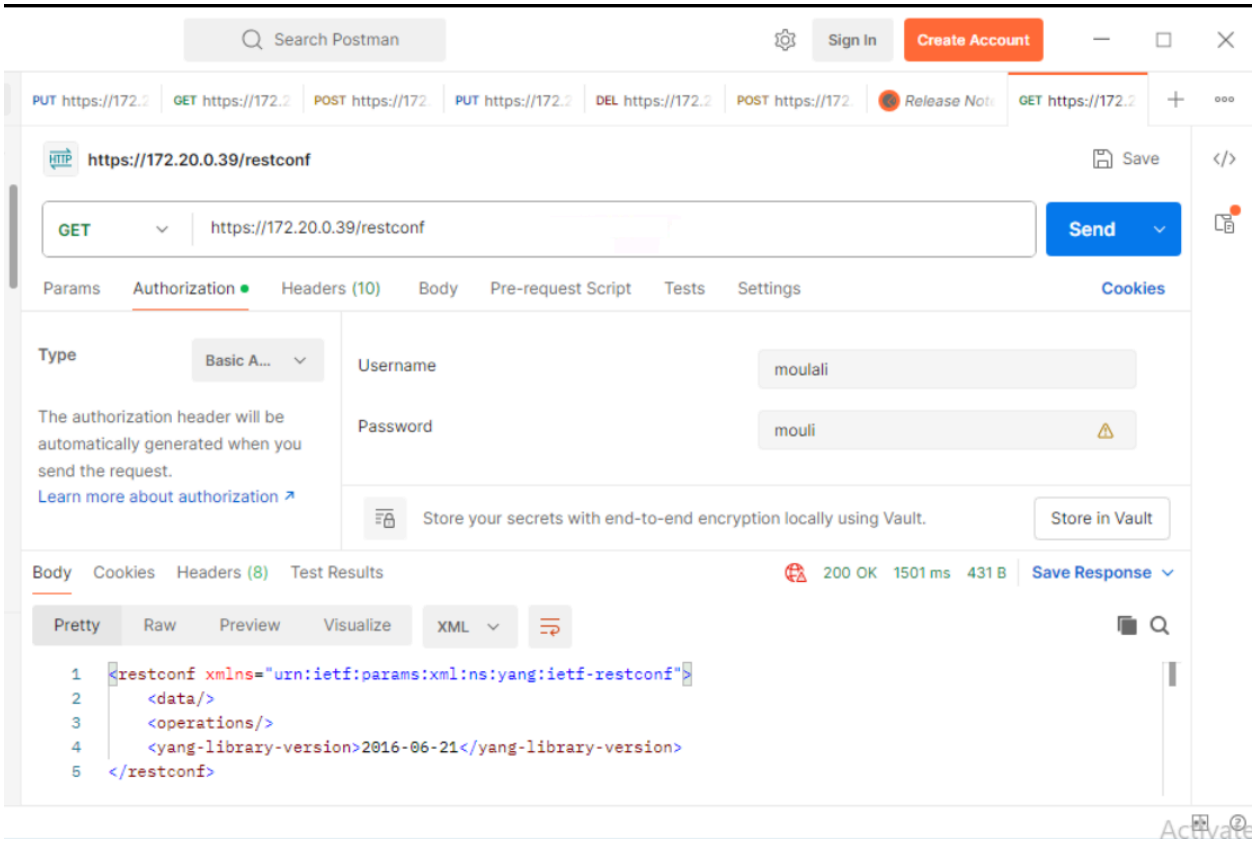


Show ip int brief

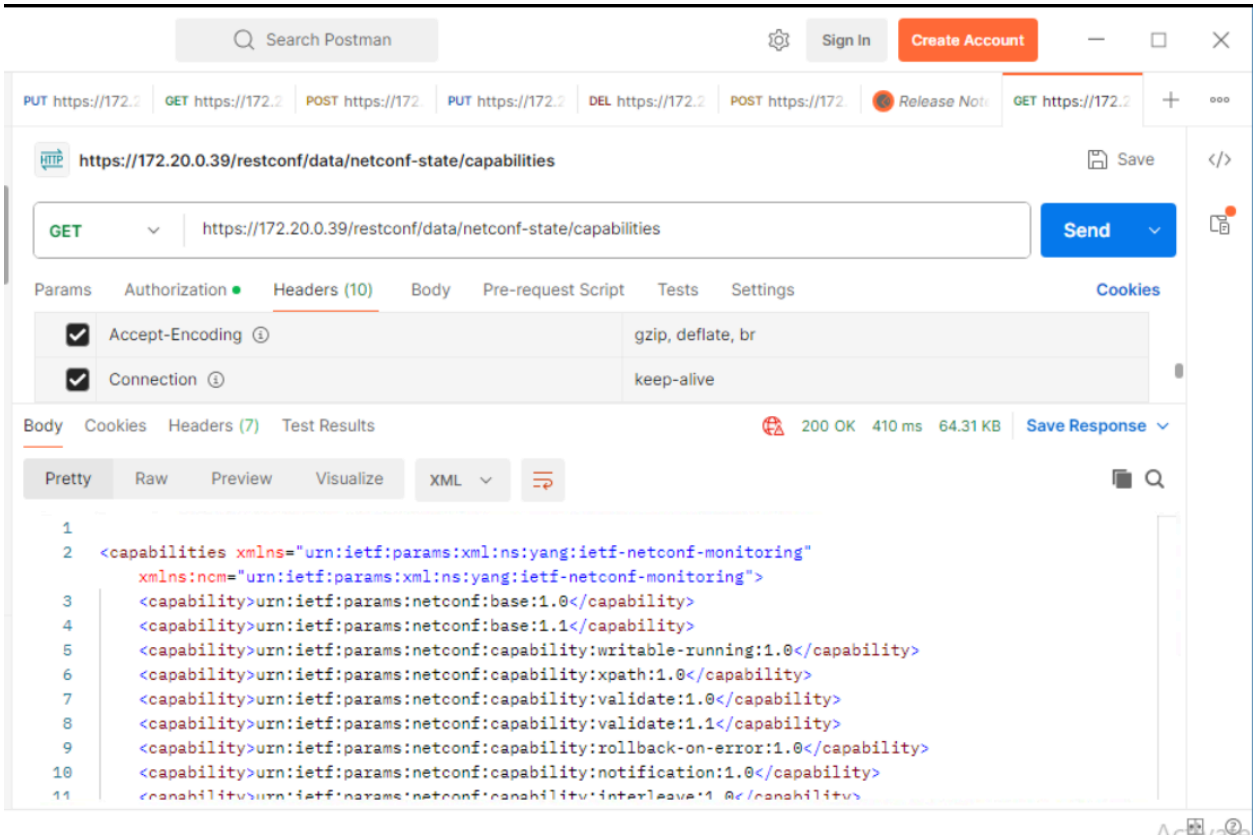
```
R2#show run | include ospf
router ospf 1
R2#show ip int brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            192.168.1.2     YES manual up                    up

FastEthernet0/1            192.168.2.1     YES manual up                    up

R2#
```

Postman

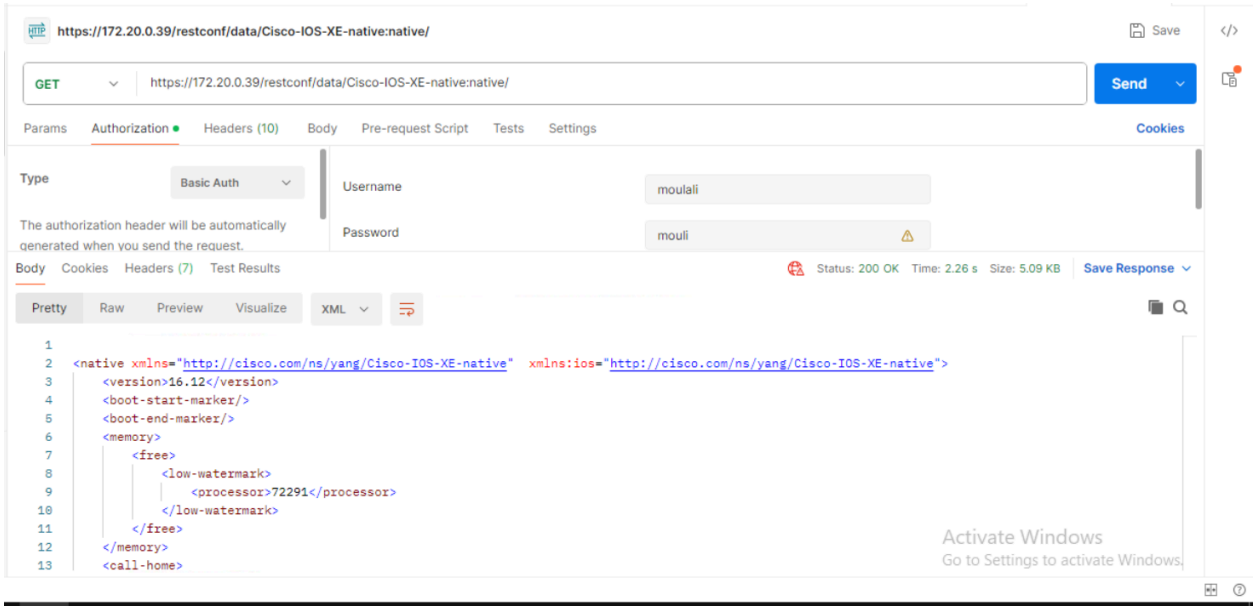
GET



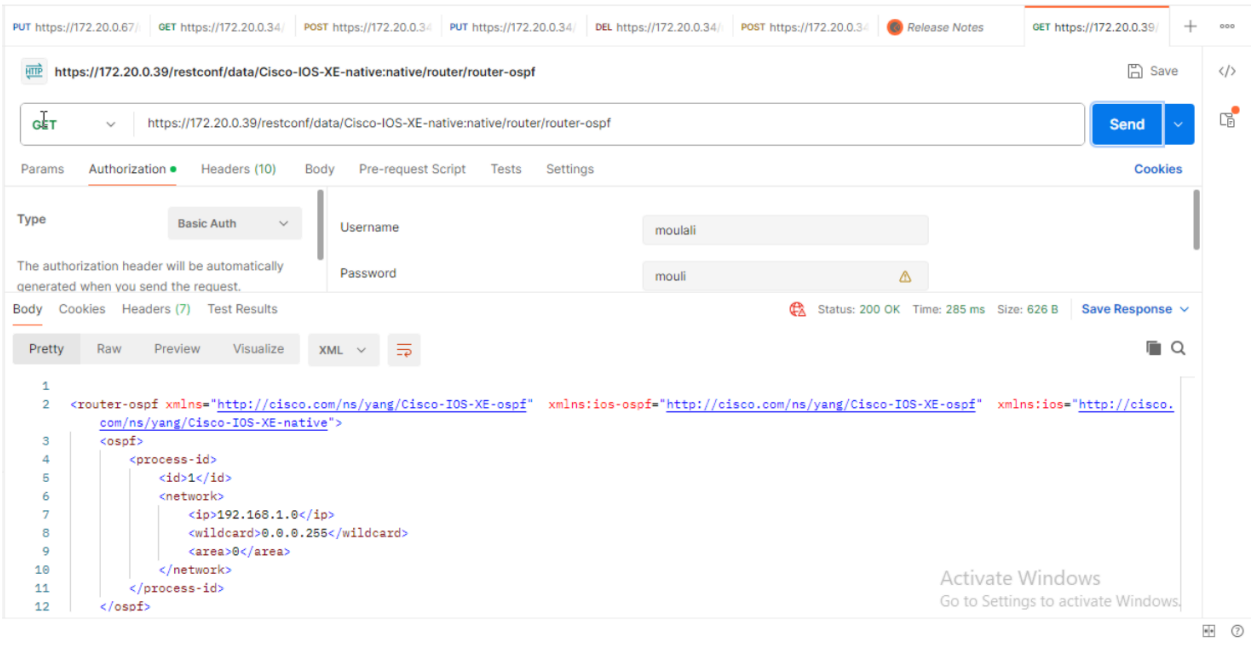
Capabilities



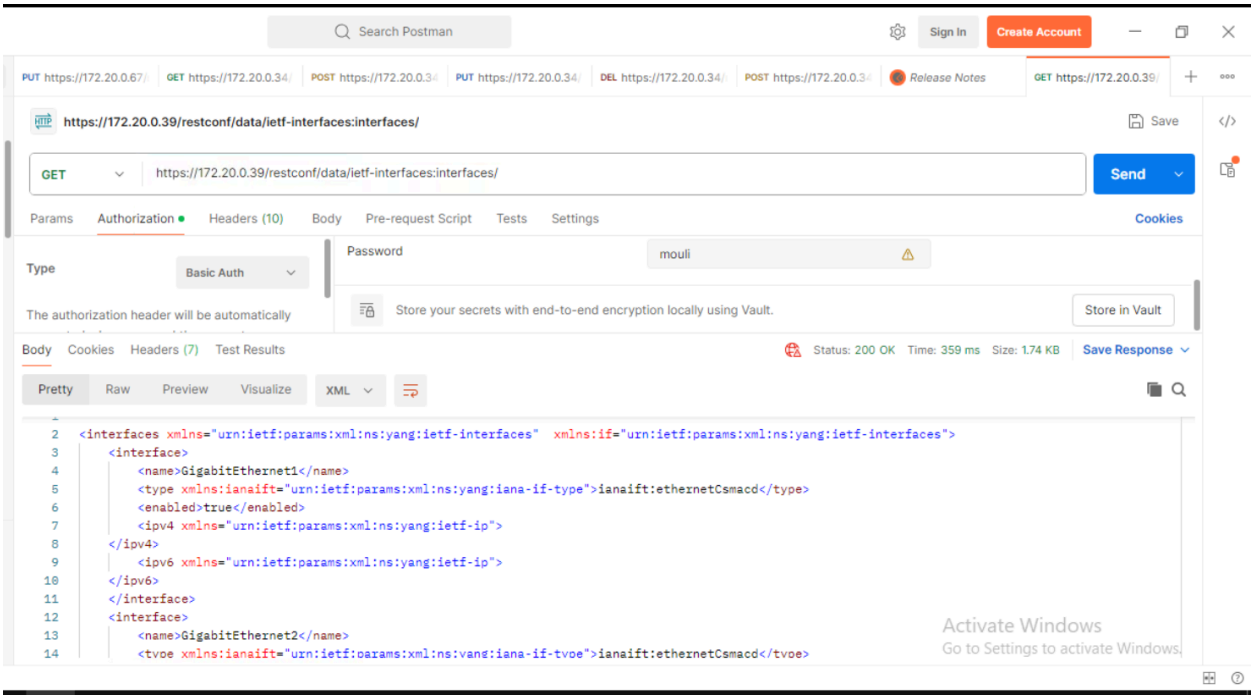
Native config



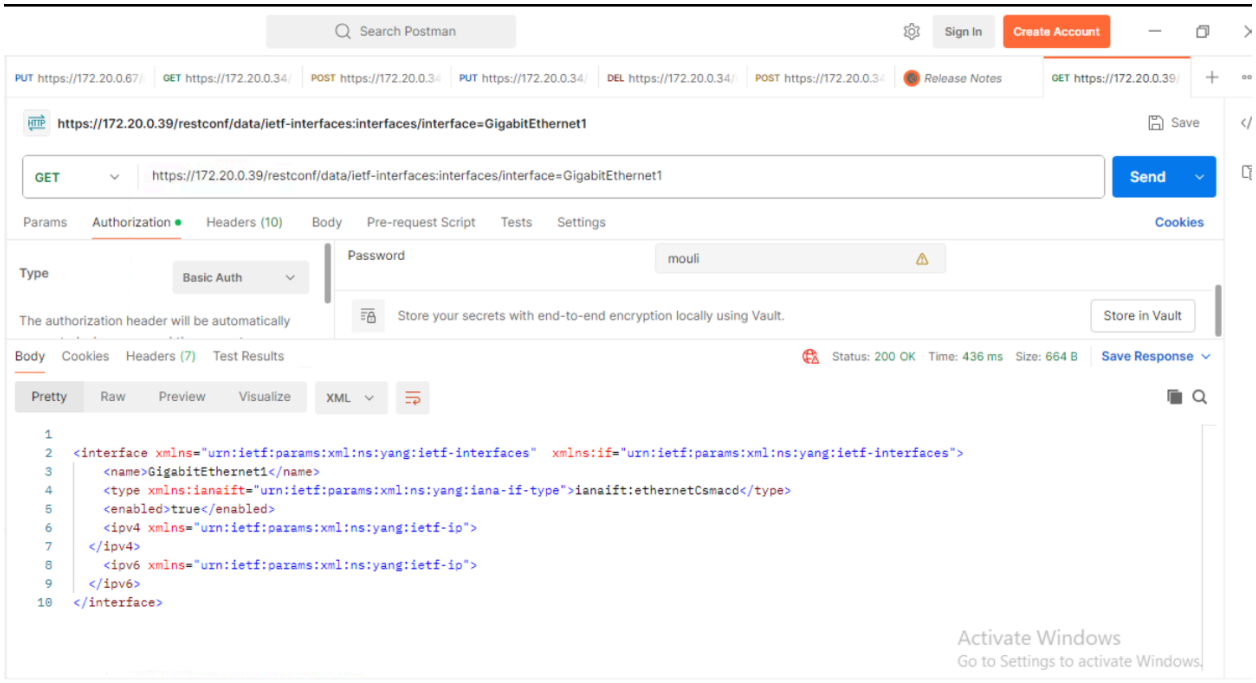
Get OSPF



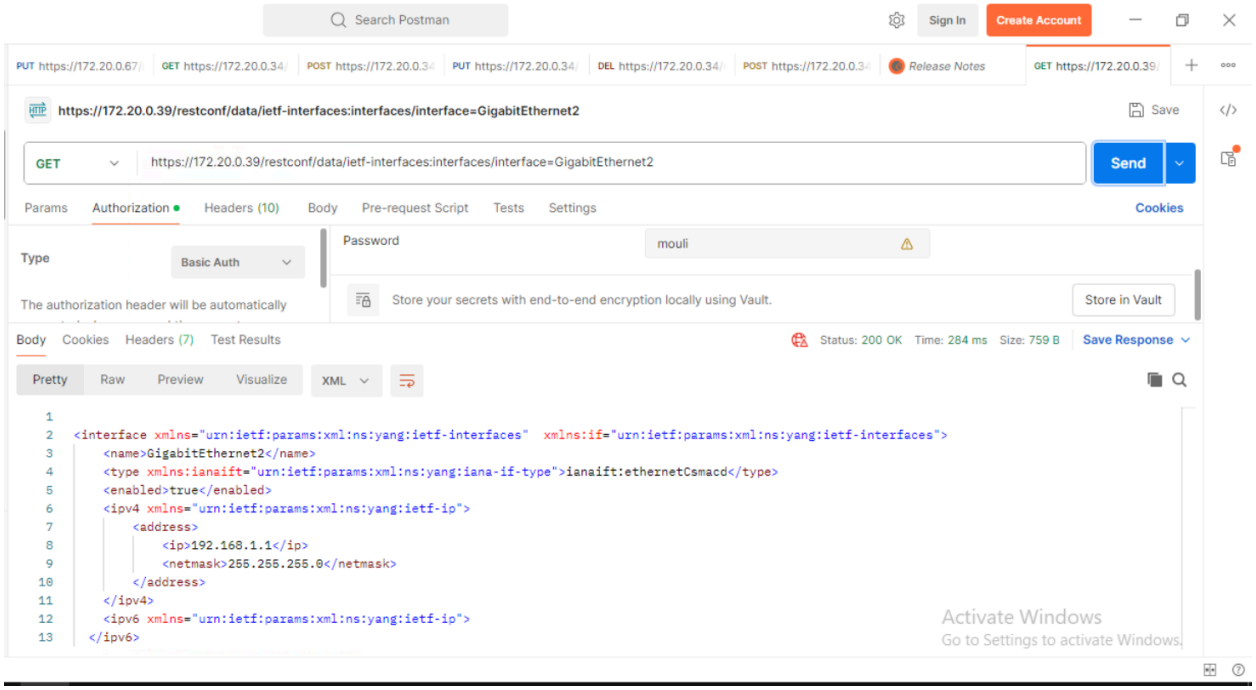
Interfaces



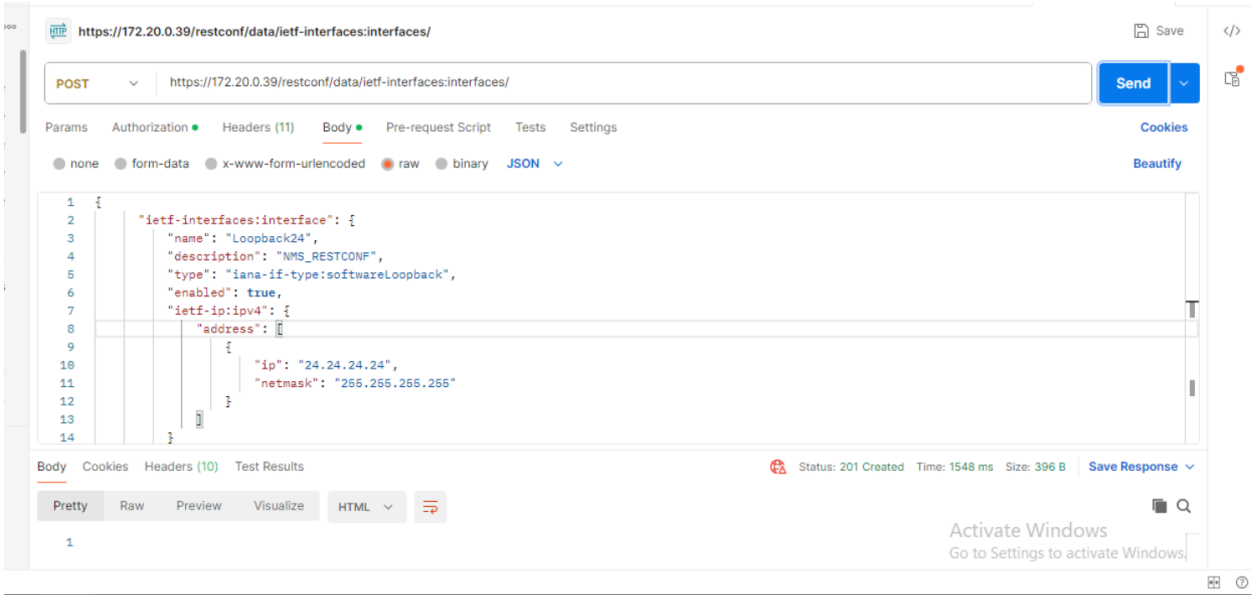
Interface = GigabitEthernet1



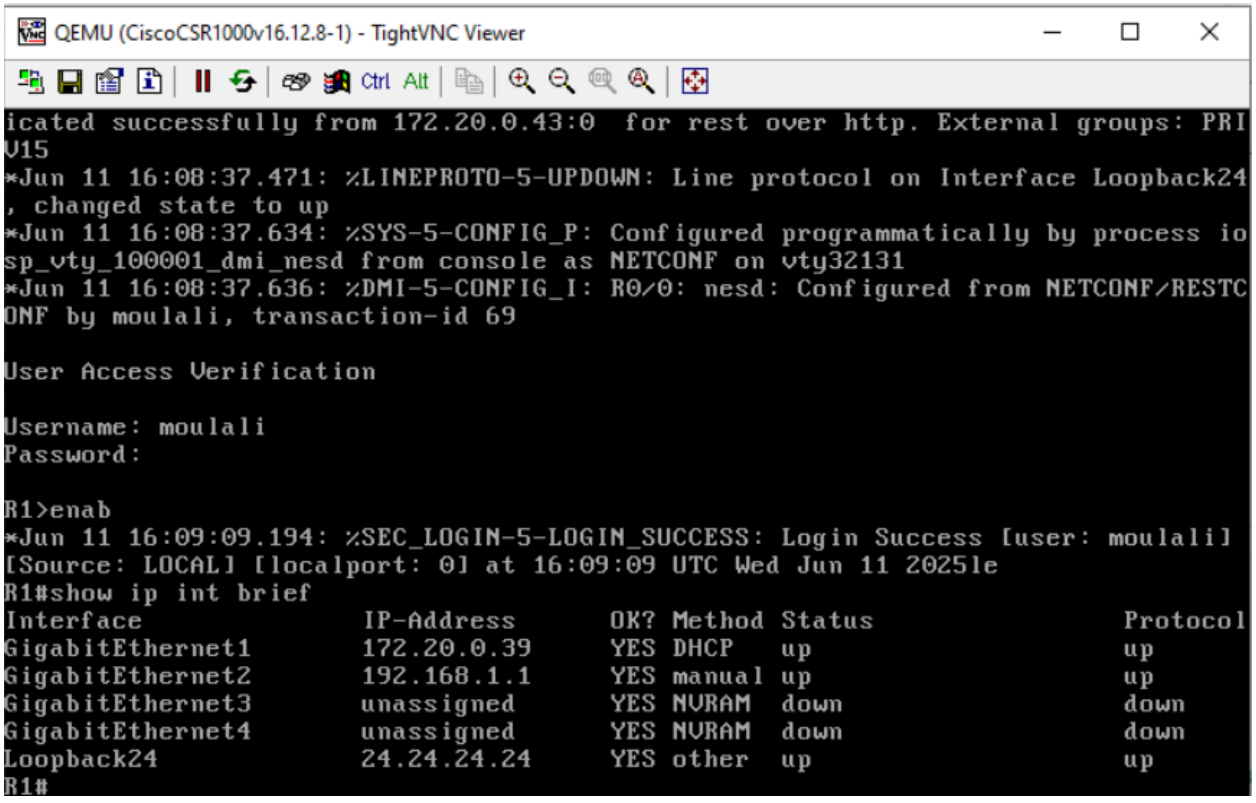
Interface = GigabitEthernet2



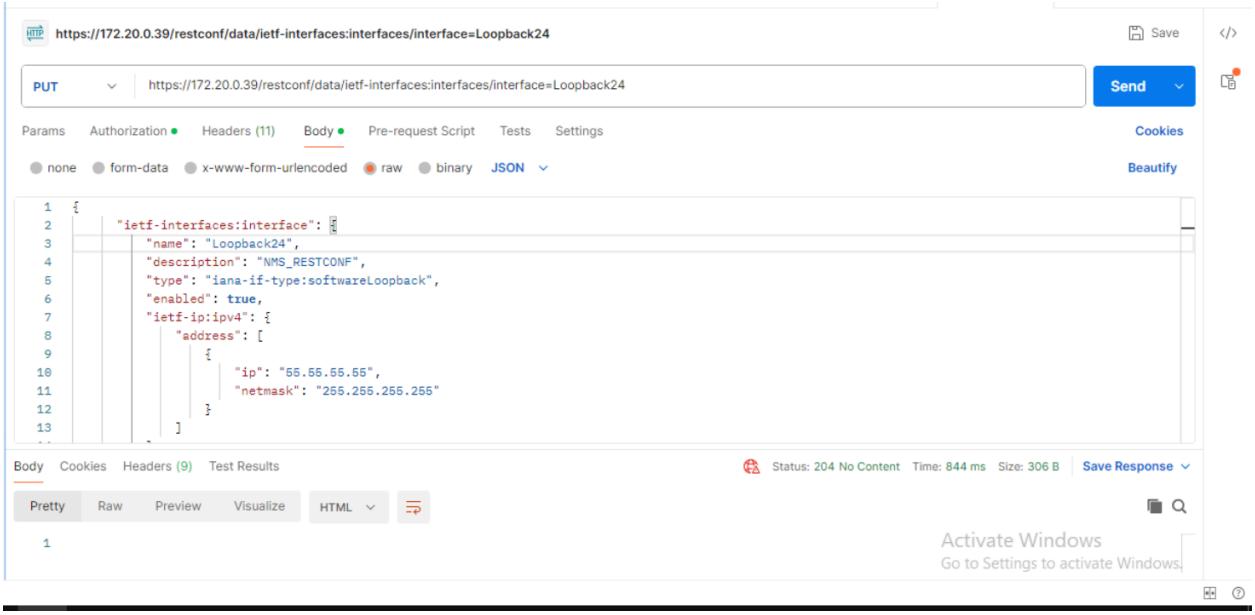
POST - Adding loopback



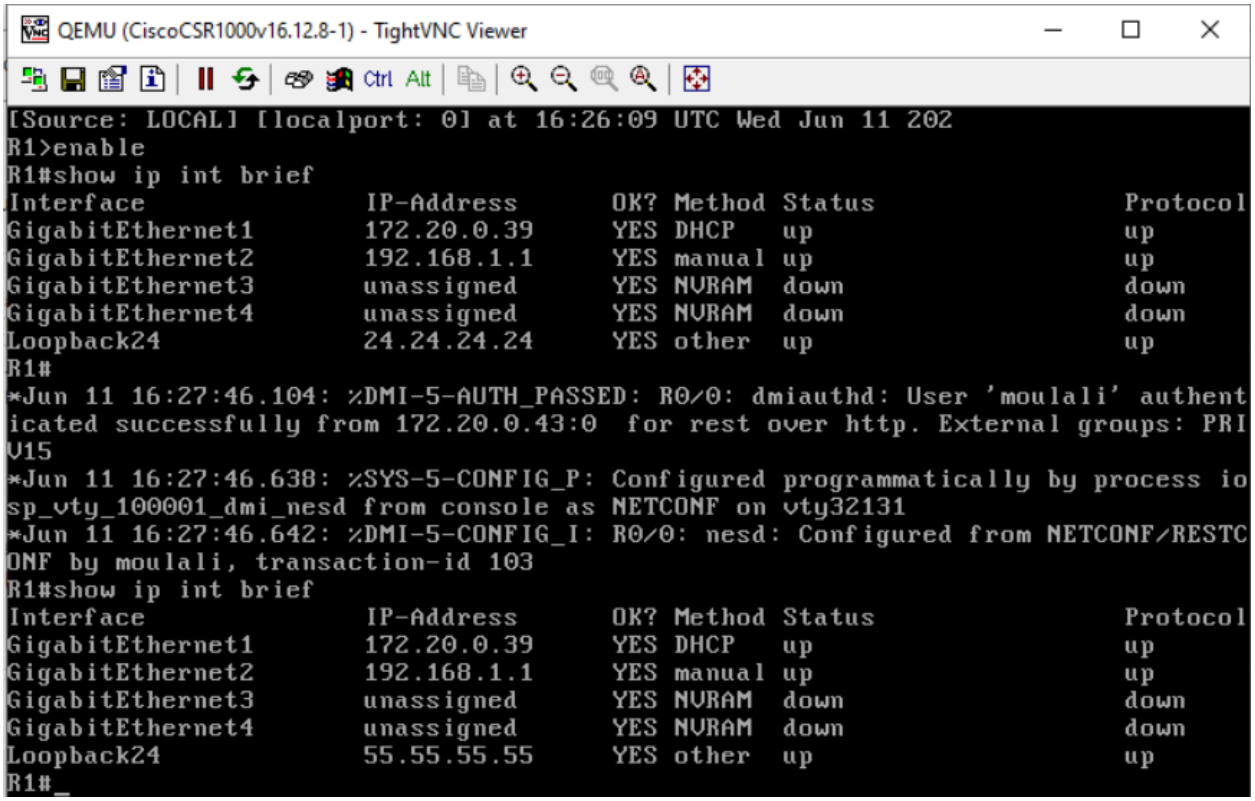
Output



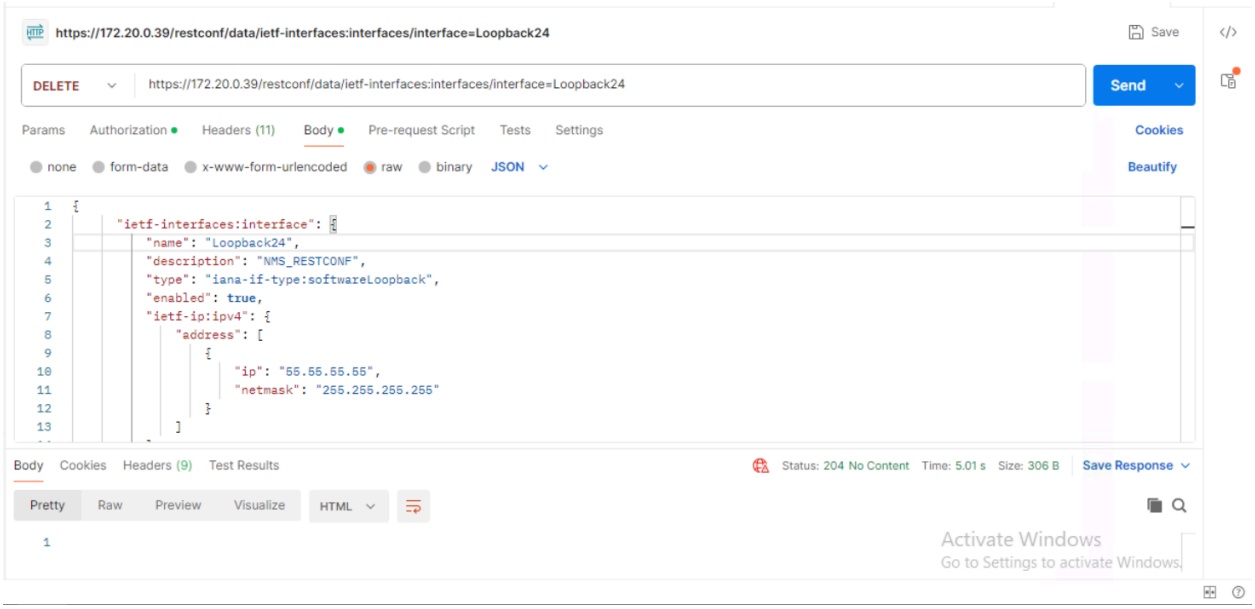
PUT - Overwriting Loopback



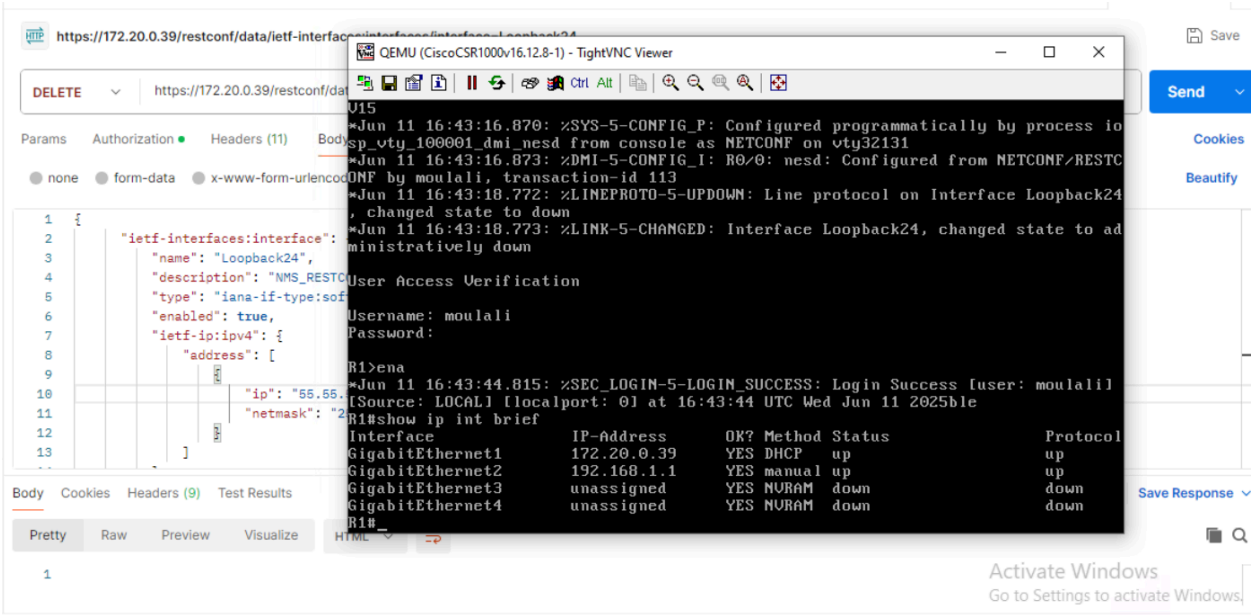
output



Delete
Removing Loopback

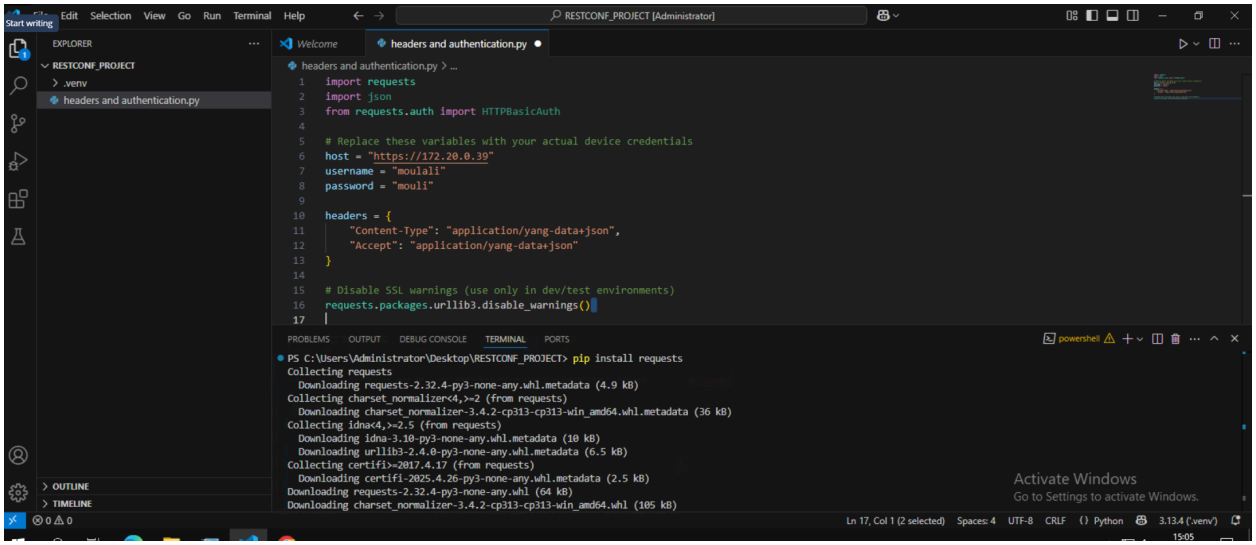


Output

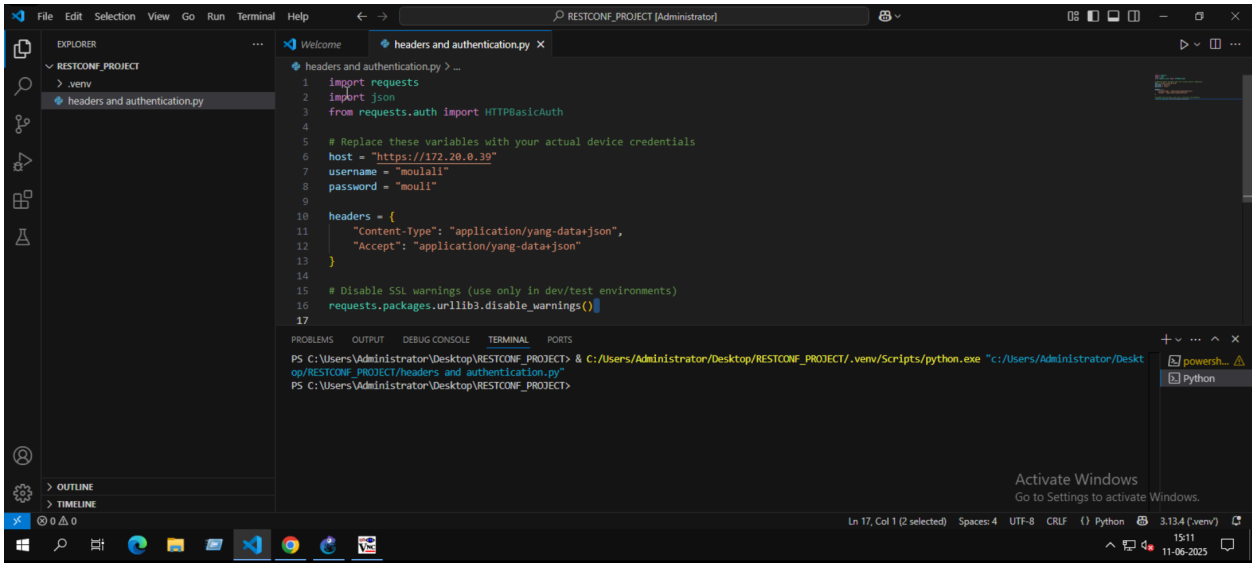


Python scripts

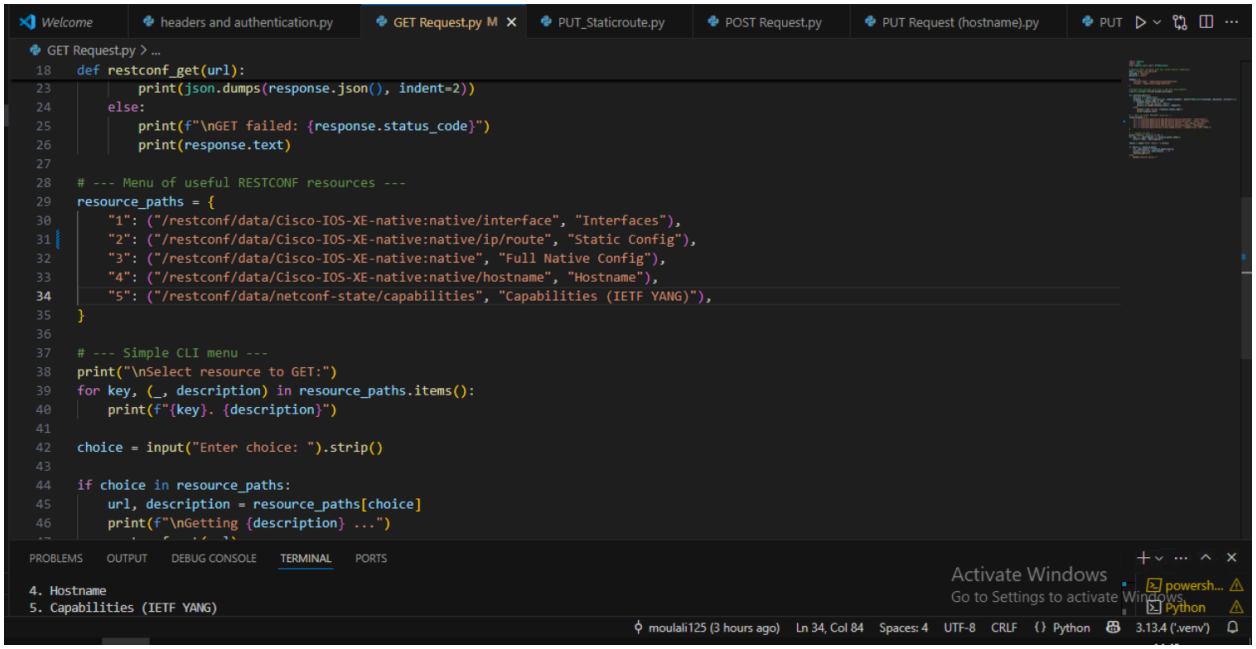
Pip install requests



Headers and authentication file



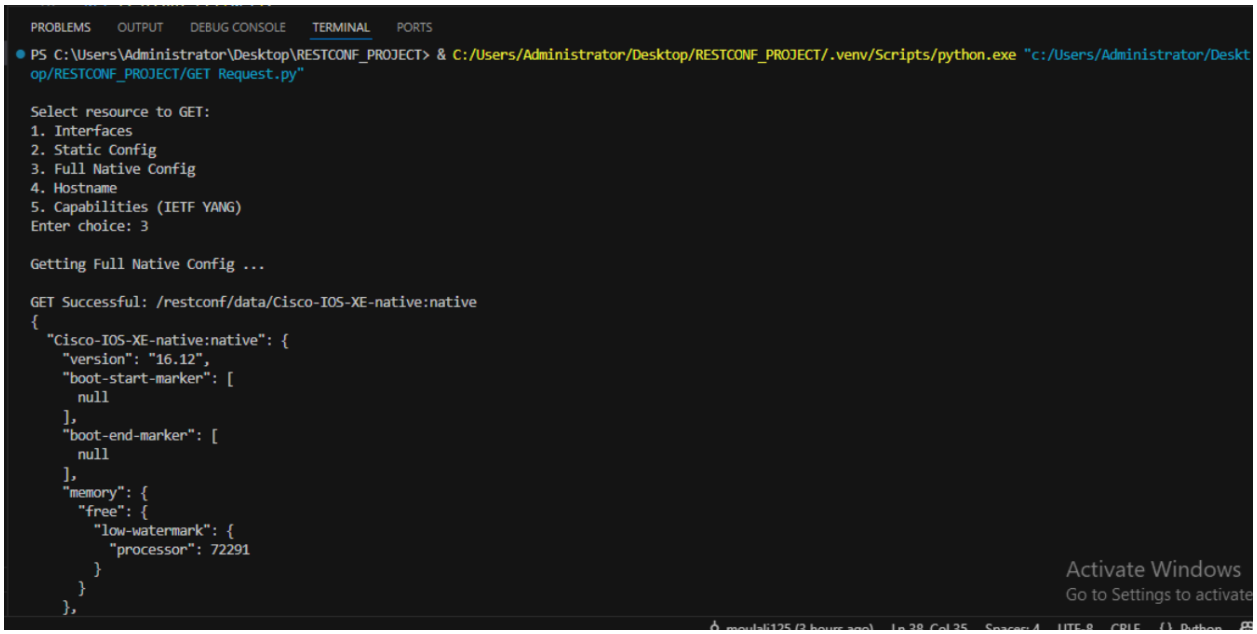
Get Request
Interfaces, Static config, Full native config, Hostname and Capabilities



```python
def restconf_get(url):
        print(json.dumps(response.json(), indent=2))
    else:
        print(f"\nGET failed: {response.status_code}")
        print(response.text)

# --- Menu of useful RESTCONF resources ---
resource_paths = {
    "1": ("/restconf/data/Cisco-IOS-XE-native:native/interface", "Interfaces"),
    "2": ("/restconf/data/Cisco-IOS-XE-native:native/ip/route", "Static Config"),
    "3": ("/restconf/data/Cisco-IOS-XE-native:native", "Full Native Config"),
    "4": ("/restconf/data/Cisco-IOS-XE-native:native/hostname", "Hostname"),
    "5": ("/restconf/data/netconf-state/capabilities", "Capabilities (IETF YANG)"),
}

# --- Simple CLI menu ---
print("\nSelect resource to GET:")
for key, (_, description) in resource_paths.items():
    print(f"{key}. {description}")

choice = input("Enter choice: ").strip()

if choice in resource_paths:
    url, description = resource_paths[choice]
    print(f"\nGetting {description} ...")
```

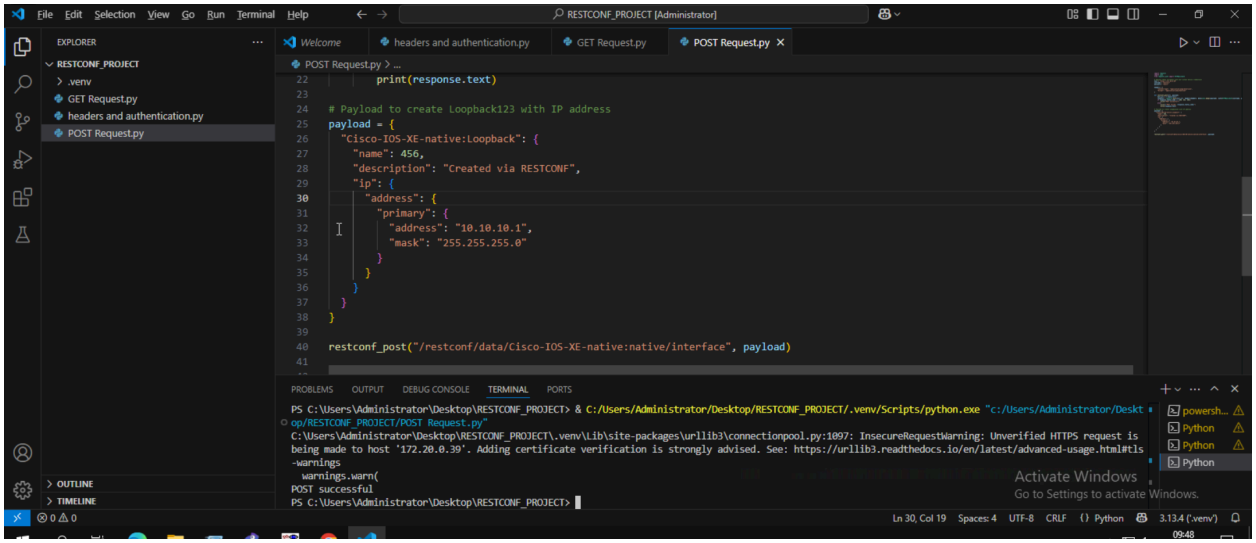
```
4. Hostname
5. Capabilities (IETF YANG)
```

Terminal Output



```
PS C:\Users\Administrator\Desktop\RESTCONF_PROJECT> & C:/Users/Administrator/Desktop/RESTCONF_PROJECT/.venv/Scripts/python.exe "c:/Users/Administrator/Desktop/RESTCONF_PROJECT/GET Request.py"

Select resource to GET:
1. Interfaces
2. Static Config
3. Full Native Config
4. Hostname
5. Capabilities (IETF YANG)
Enter choice: 3

Getting Full Native Config ...

GET Successful: /restconf/data/Cisco-IOS-XE-native:native
{
  "Cisco-IOS-XE-native:native": {
    "version": "16.12",
    "boot-start-marker": [
      null
    ],
    "boot-end-marker": [
      null
    ],
    "memory": {
      "free": {
        "low-watermark": {
          "processor": 72291
        }
      }
    },
```

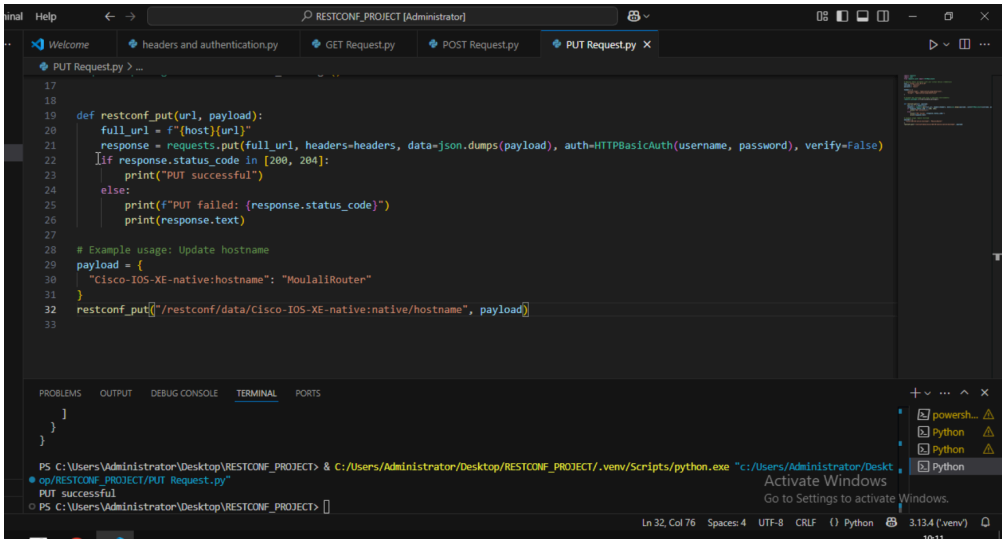
POST Request
Adding Loopback



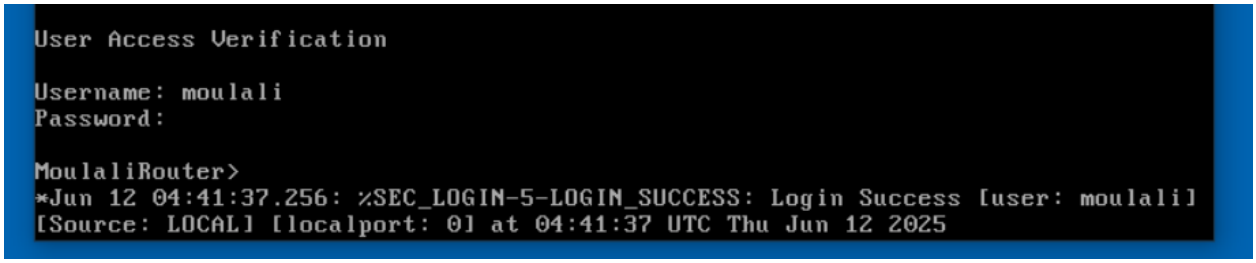
Output

```
MoulaliRouter>
*Jun 12 04:41:37.256: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: moulali]
[Source: LOCAL] [localport: 0] at 04:41:37 UTC Thu Jun 12 2025
MoulaliRouter>show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       172.20.0.39     YES DHCP   up                    up
GigabitEthernet2       192.168.1.1     YES NVRAM  up                    up
GigabitEthernet3       unassigned      YES NVRAM  down                  down
GigabitEthernet4       unassigned      YES NVRAM  down                  down
Loopback123            unassigned      YES unset  up                    up
Loopback456            10.10.10.1      YES other  up                    up
MoulaliRouter>
```

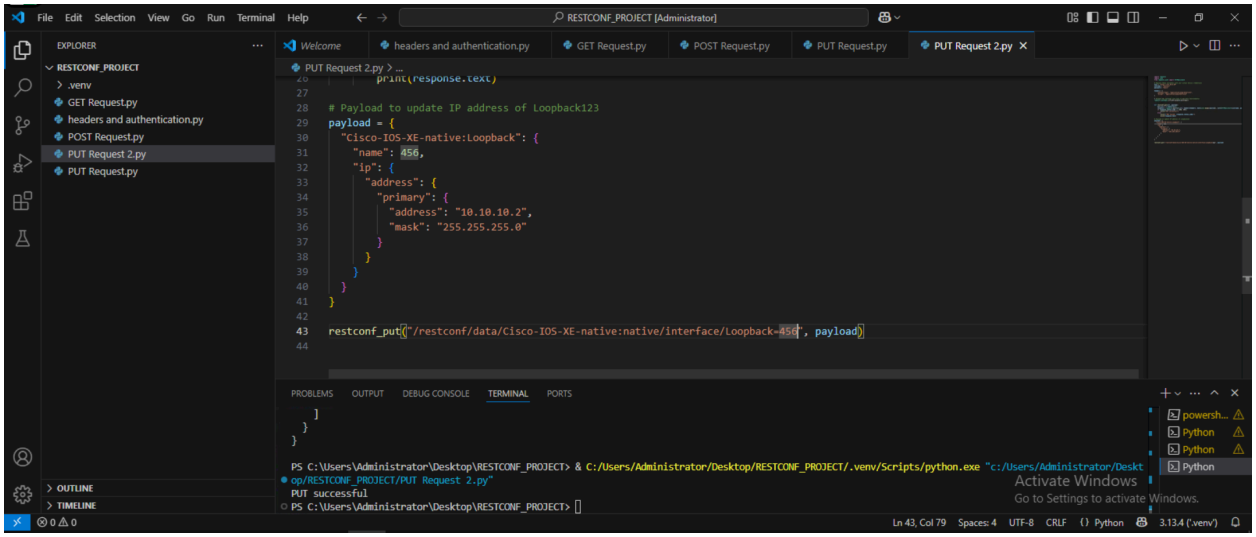
Put
Changing hostname



Output



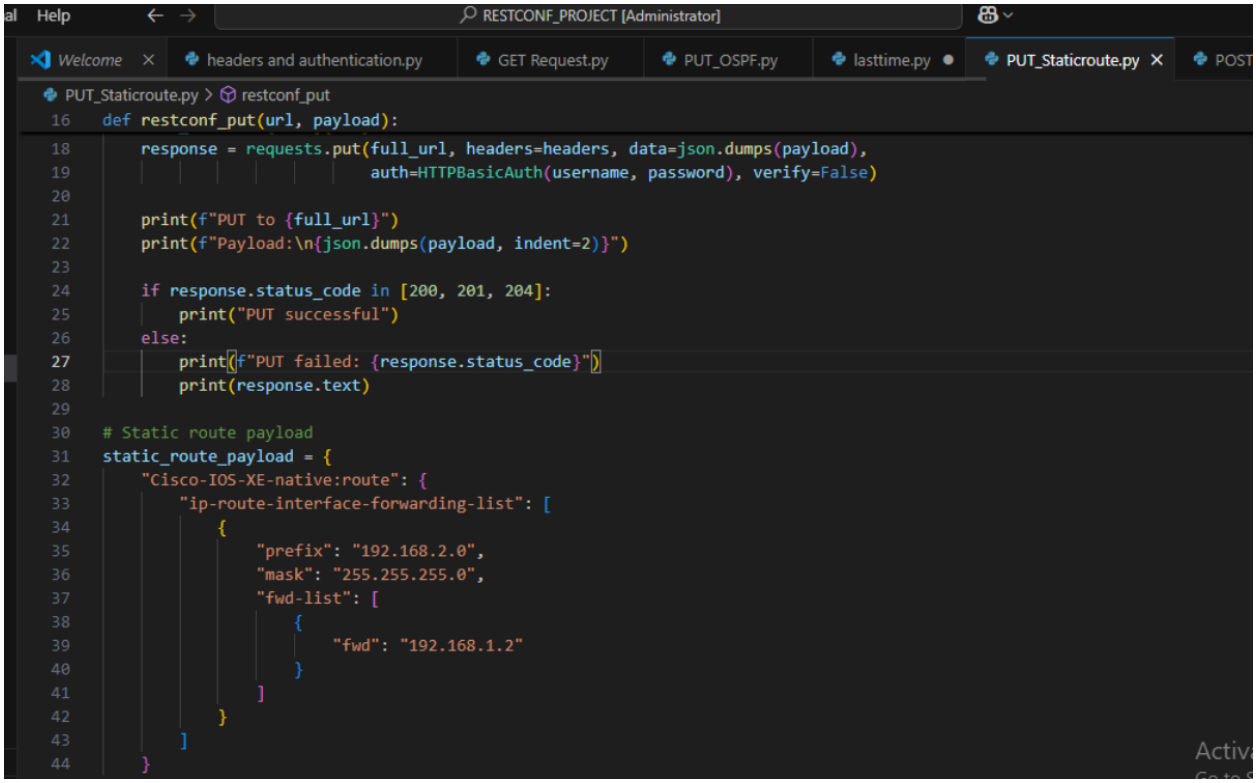
Overwriting loopback interface



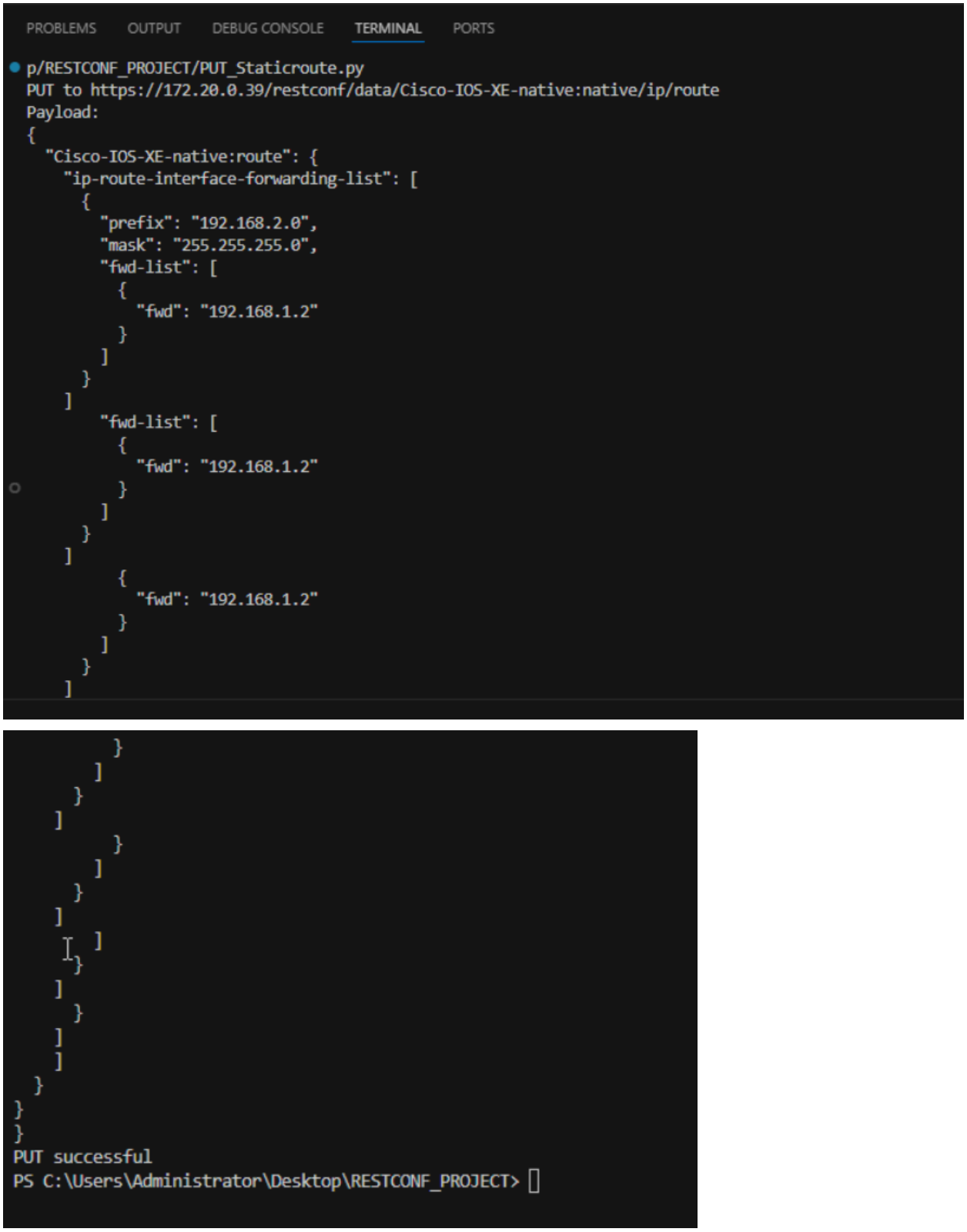
Output

```
MoulaliRouter#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       172.20.0.39     YES DHCP   up                    up
GigabitEthernet2       192.168.1.1     YES NVRAM  up                    up
GigabitEthernet3       unassigned      YES NVRAM  down                  down
GigabitEthernet4       unassigned      YES NVRAM  down                  down
Loopback123            unassigned      YES unset  up                    up
Loopback456            10.10.10.2      YES other  up                    up
MoulaliRouter#
```

Static route configuration



Terminal Output



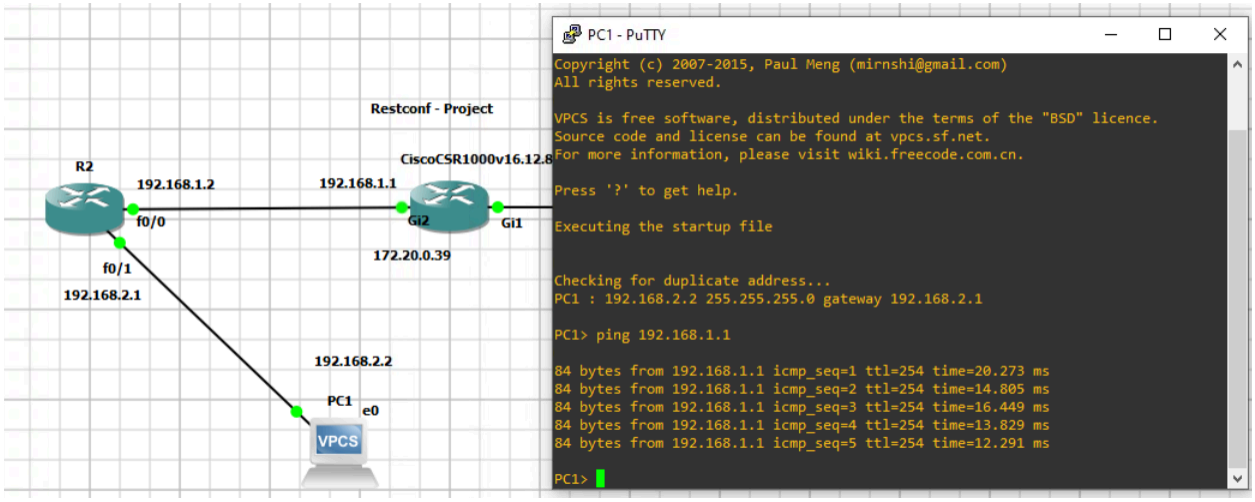
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

p/RESTCONF_PROJECT/PUT_Staticroute.py
PUT to https://172.20.0.39/restconf/data/Cisco-IOS-XE-native:native/ip/route
Payload:
{
  "Cisco-IOS-XE-native:route": {
    "ip-route-interface-forwarding-list": [
      {
        "prefix": "192.168.2.0",
        "mask": "255.255.255.0",
        "fwd-list": [
          {
            "fwd": "192.168.1.2"
          }
        ]
      }
    ]
        "fwd-list": [
          {
            "fwd": "192.168.1.2"
          }
        ]
      }
    ]
          {
            "fwd": "192.168.1.2"
          }
        ]
      }
    ]
          }
        ]
      }
    ]
          }
        ]
      }
    ]
     ]
    }
    ]
      }
    ]
    ]
  }
}
}
PUT successful
PS C:\Users\Administrator\Desktop\RESTCONF_PROJECT>
```

Router output

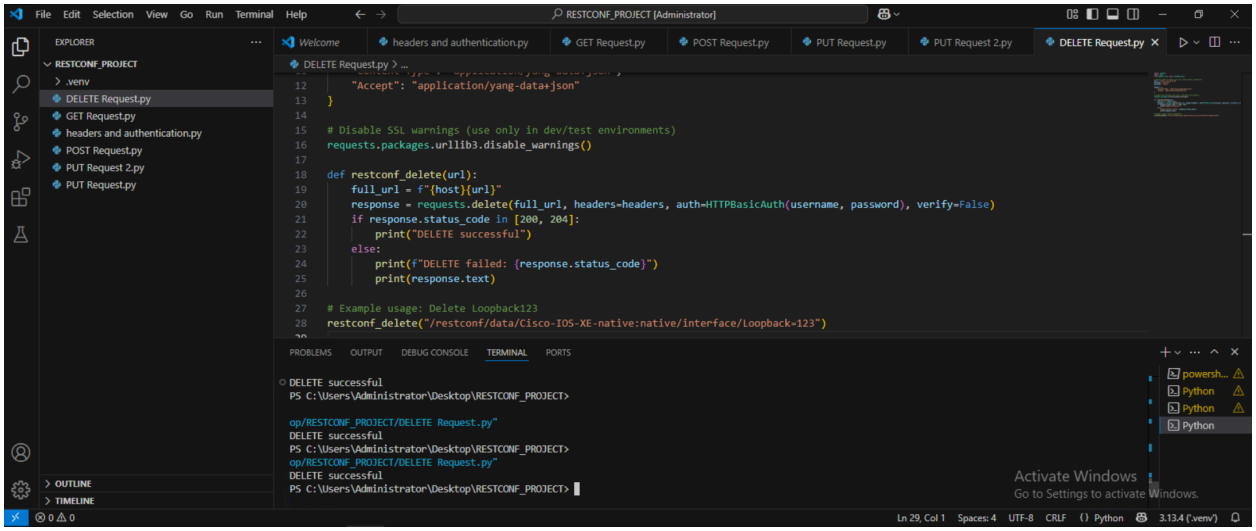
QEMU (CiscoCSR1000v16.12.8-1) - TightVNC Viewer

```
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet1
!
ip route 192.168.2.0 255.255.255.0 192.168.1.2
!
!
!
!
!
!
!
!
!
control-plane
!
!
 --More--
```

ping



Delete



Ouput