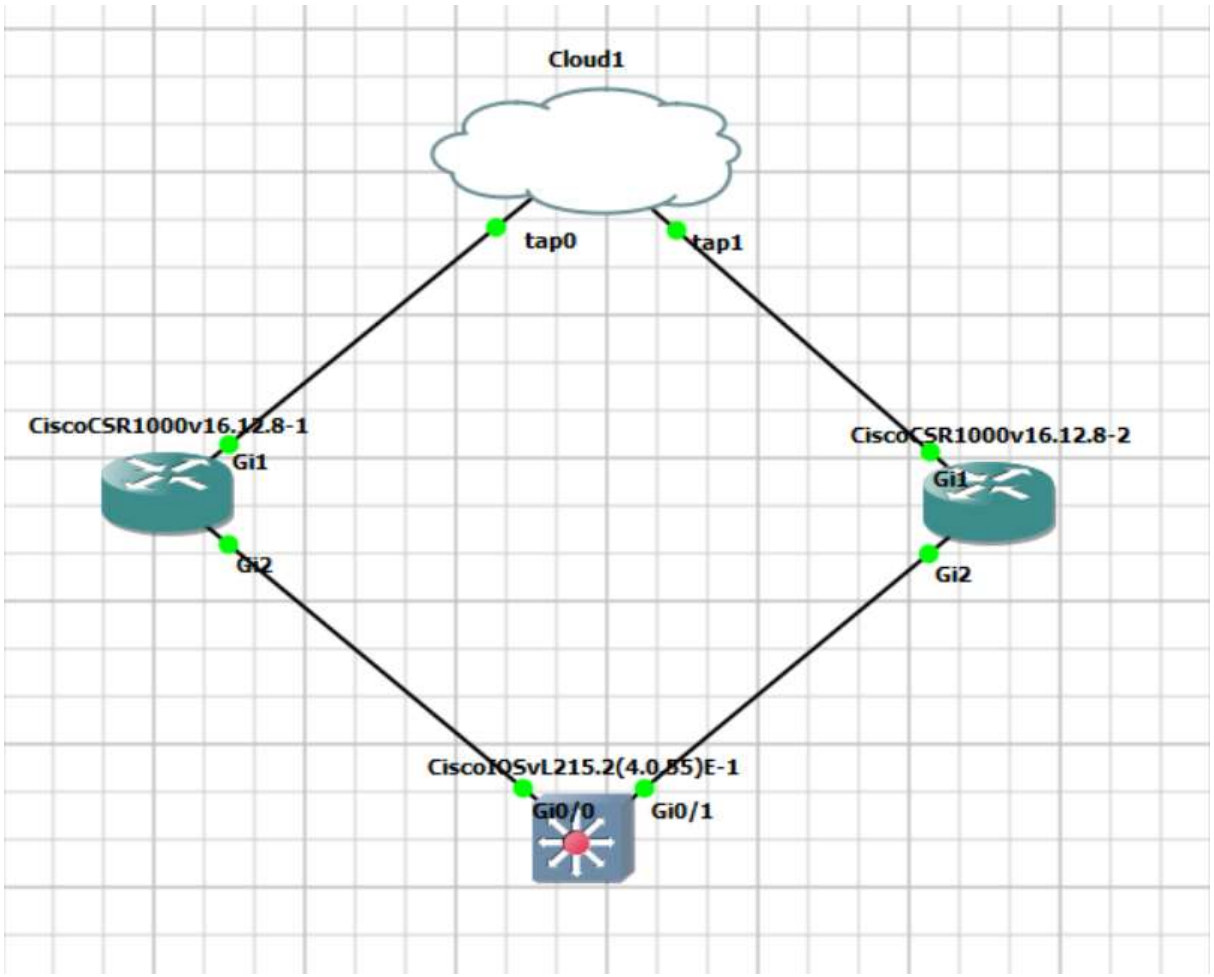
Enable the NETCONF & RESTCONF configuration in router(10-07)

Network Topology:



DHCP will allocate the IP address to the Router:

```
Router>
*Jul 10 13:57:40.193: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [S
ource: LOCAL] [localport: 0] at 13:57:40 UTC Wed Jul 10 2024sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       172.20.0.109    YES DHCP   up                    up
GigabitEthernet2       10.0.0.1        YES manual up                    up
GigabitEthernet3       unassigned      YES unset  down                  down
GigabitEthernet4       unassigned      YES unset  administratively down down
```

Go to Config Mode:

Conf t
user admin privilege 15 secret cisco123
aaa new-model
aaa authentication login default local
aaa authorization exec default local
Netconf-yang

Check platform software yang-management process:

```
Router#sh platform software yang-management process
confd              : Running
mesd               : Running
syncfd             : Running
ncsshd             : Running
mmiauthd           : Running
nginx              : Running
ndbmand            : Running
pubd               : Running

Router#_
```

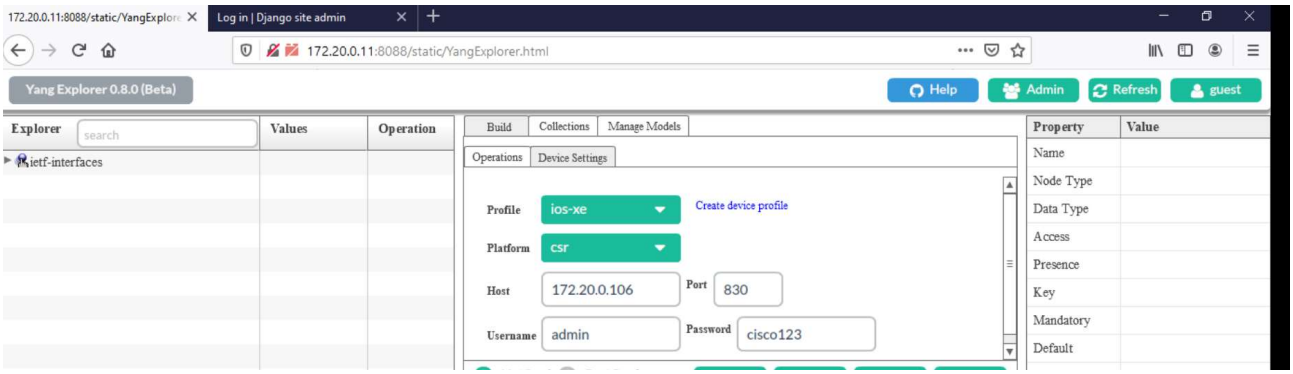
Command prompt and check the yang connectivity:

```
Administrator: Command Prompt - ssh admin@172.20.0.109 -p 830 -s netconf
Microsoft Windows [Version 10.0.19042.1949]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ssh admin@172.20.0.109 -p 830 -s netconf
The authenticity of host '[172.20.0.109]:830 ([172.20.0.109]:830)' can't be established.
RSA key fingerprint is SHA256:tyvhtLLg6cFfpHx9tRJgaCrRqypzuRKDfpQT+703UQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.20.0.109]:830' (RSA) to the list of known hosts.
admin@172.20.0.109's password:

C:\Users\Administrator>ssh admin@172.20.0.109 -p 830 -s netconf
admin@172.20.0.109's password:

C:\Users\Administrator>ssh admin@172.20.0.109 -p 830 -s netconf
admin@172.20.0.109's password:
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</capability>
<capability>urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit&amp;also-supported=report-all-tagge
d</capability>
<capability>urn:ietf:params:netconf:capability:yang-library:1.0?revision=2016-06-21&amp;module-set-id=b1077a37217dd85cbd
```

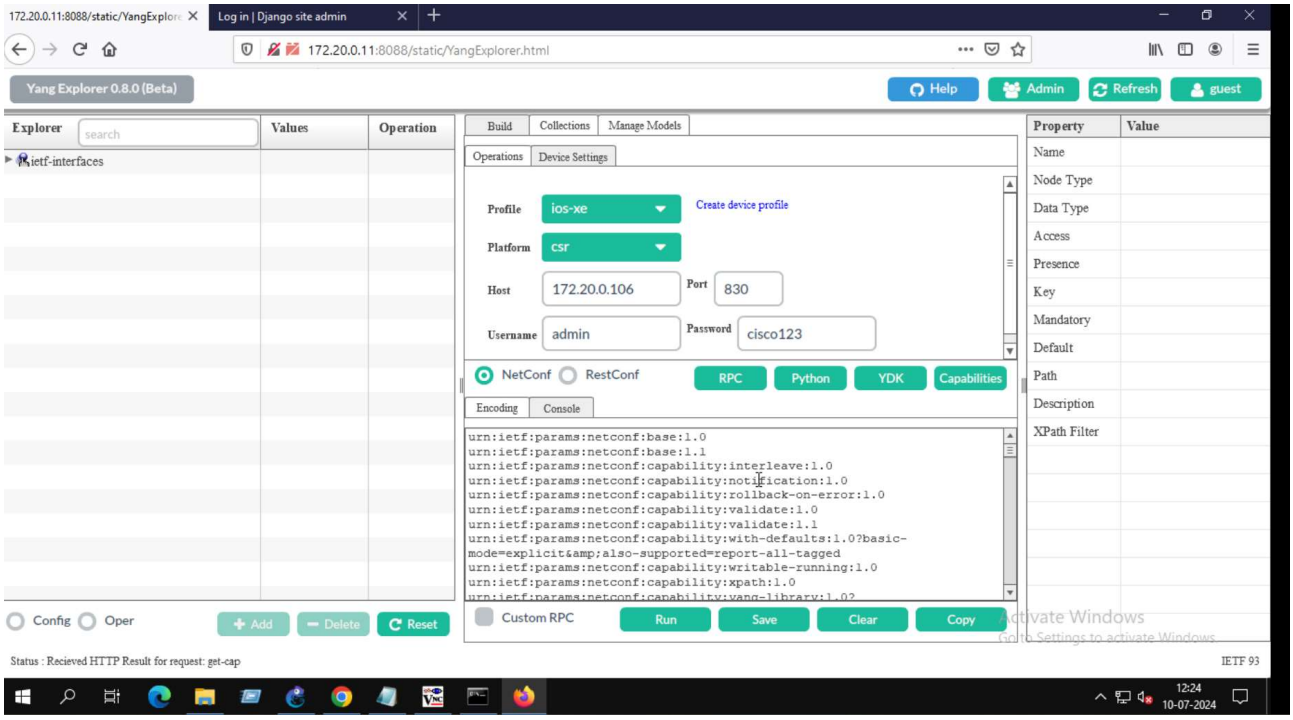
Add the additional Gi2 interface to router and assign the IP 10.0.0.1:

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g2
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#
```

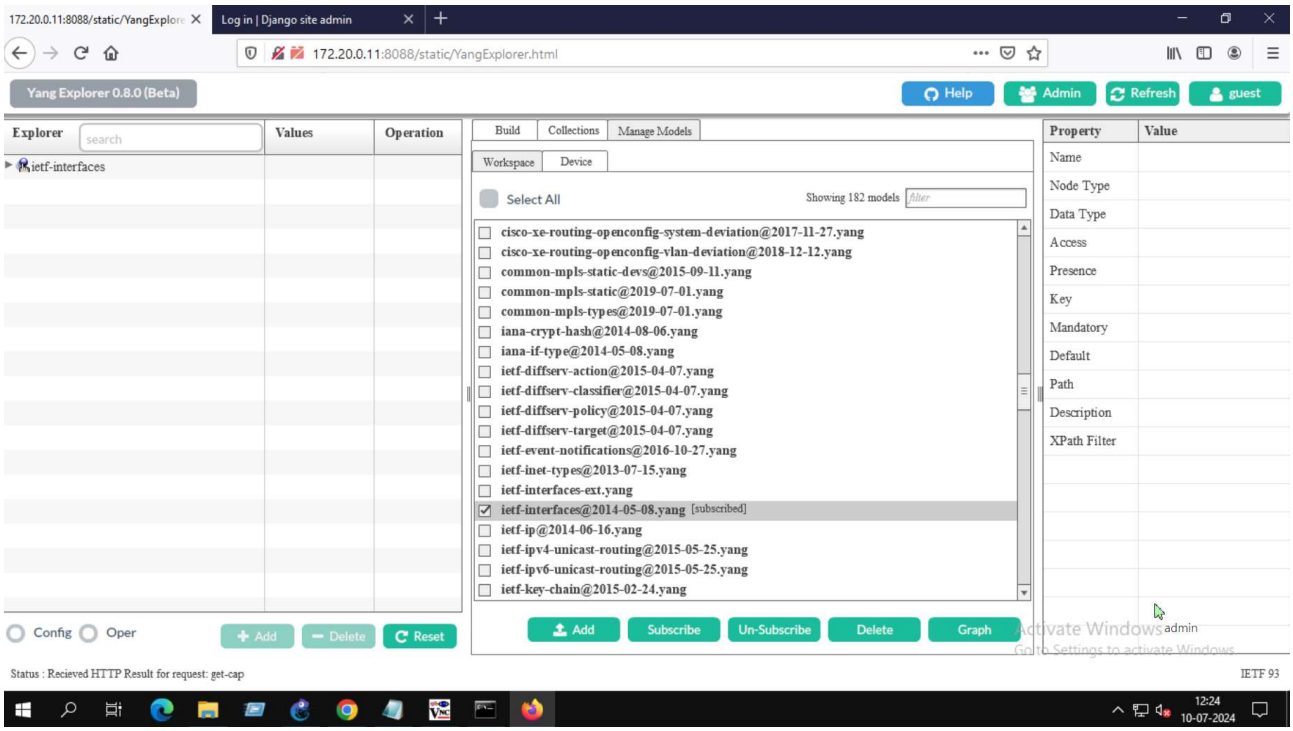
Login to YANG Explorer:



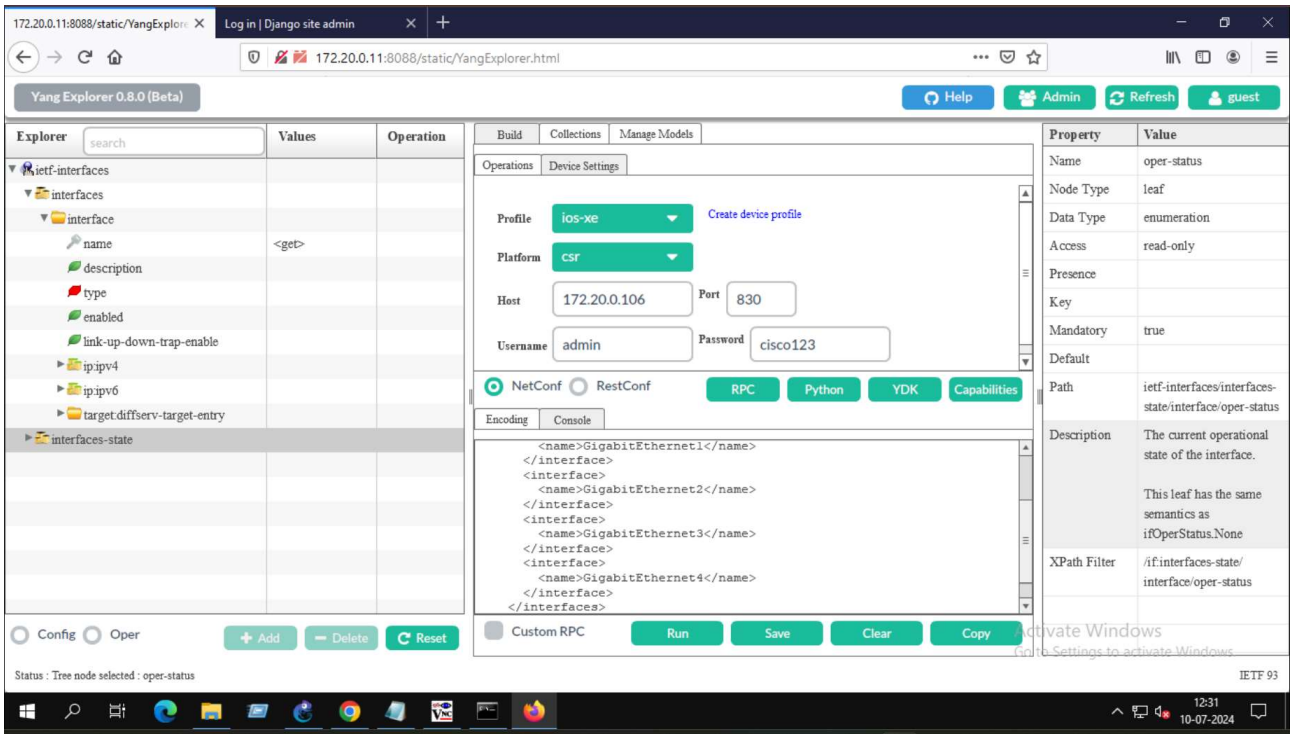
Click on capability, router will exchange the capability.



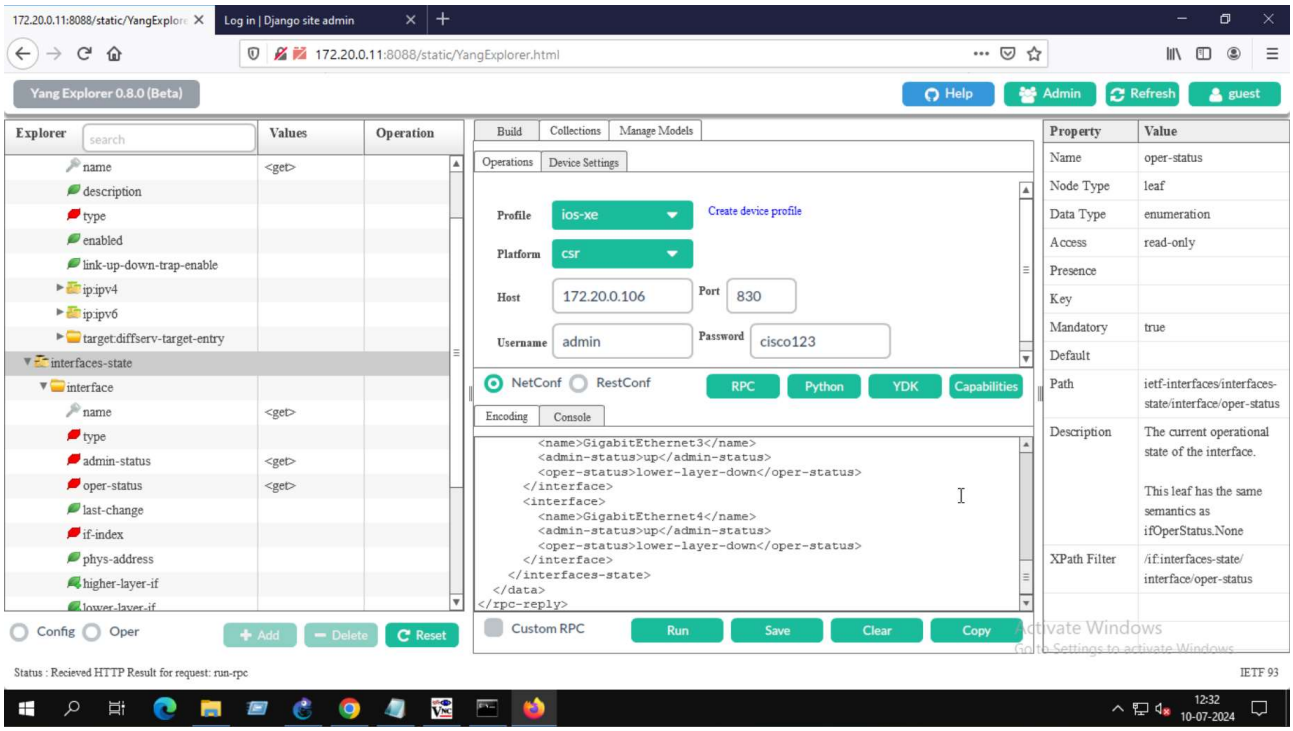
Click on Manage model and subscribe the ietf-interface yang model



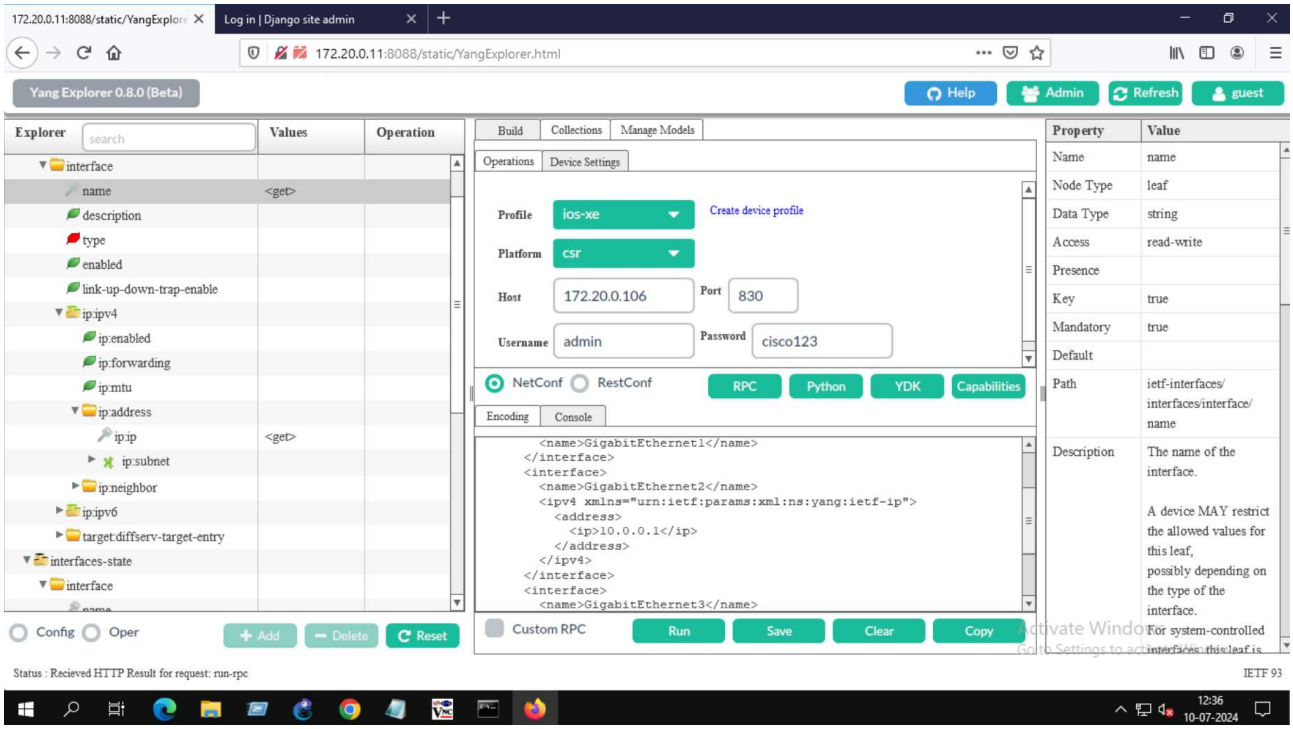
Example 1; Get the Router Interface Name



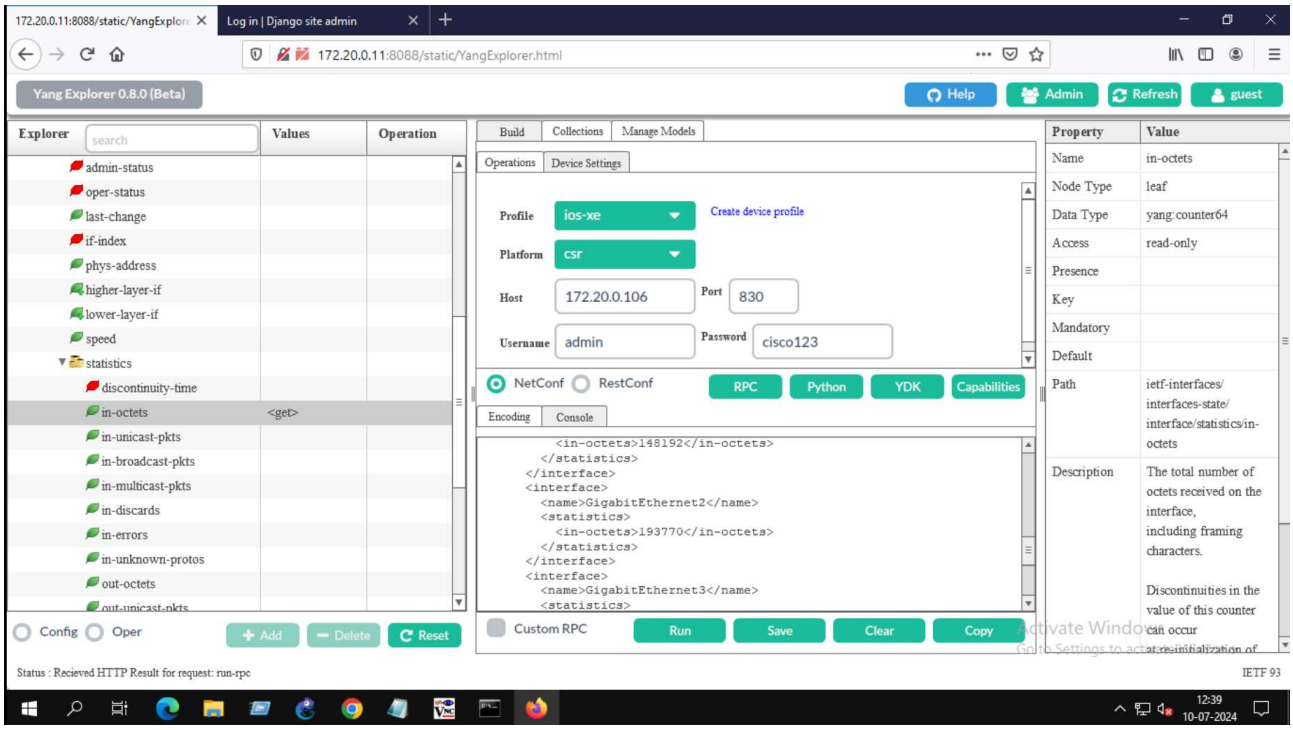
Example 2; Get the Router Interface admin and Operation state



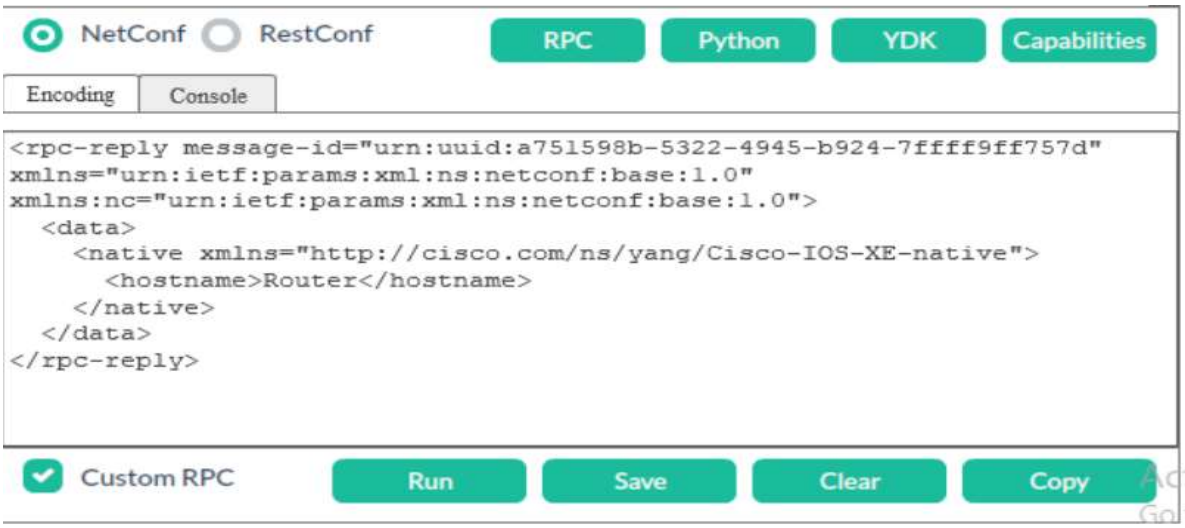
Example 3: Run the YANG query and get the Router interface and IP details



Example 4; Run the YANG query and get the Router interface stats



1: Get the router hostname:



2: Change the Router hostname:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc-reply message-id="urn:uuid:4344d02d-3c2f-469e-b3cd-73fb52bcae86"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc-reply message-id="urn:uuid:61b02590-a4cd-47a7-ba88-37fc803bacdf"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <hostname>ROUTER1</hostname>
    </native>
  </data>
</rpc-reply>
```

Custom RPC Run Save Clear Copy

3: Change the interface operational status:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet2</name>
          <enabled>false</enabled>
        </interface>
      </interfaces>
```

Custom RPC Run Save Clear Copy

4: Check the IP address:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc-reply message-id="urn:uuid:2354a8f8-ffa7-4e73-9f7b-950585a98e96"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet2</name>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>10.0.0.1</ip>
            <netmask>255.0.0.0</netmask>
          </address>
```

Custom RPC Run Save Clear Copy

5: Assign the IP address on interface:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet2</name>
          <enabled>true</enabled>
          <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
            <address>
              <ip>22.22.22.2</ip>
              <netmask>255.255.255.0</netmask>
            </address>
```

Custom RPC Run Save Clear Copy

6: configure the loopback IP address

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
        <Loopback>
         <name>4</name>
         <ip>
          <address>
           <primary>
            <address>14.1.1.1</address>
            <mask>255.255.255.0</mask>
           </primary>
          </address>
         </ip>
        </Loopback>
       </interface>
```

Custom RPC Run Save Clear Copy

7: Attempt to create new loopback interface with same IP address:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<nc:rpc-error xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nc:error-type>application</nc:error-type>
  <nc:error-tag>invalid-value</nc:error-tag>
  <nc:error-severity>error</nc:error-severity>
  <nc:error-message lang="en" xmlns="http://www.w3.org/XML/1998/
namespace">inconsistent value: Device refused one or more commands</
nc:error-message>
  <nc:error-info>
    <severity xmlns="http://cisco.com/yang/cisco-ia">error_cli</
severity>
    <detail xmlns="http://cisco.com/yang/cisco-ia">
      <bad-cli>
```

Custom RPC Run Save Clear Copy

8: Delete the Loopback IP address:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface operation='delete'>
   <Loopback>
    <name>4</name>
    <ip>
     <address operation='delete'>
      <primary>
       <address>14.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
```

Custom RPC Run Save Clear Copy

9: Delete the loopback interface:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <interface>
       <Loopback operation='delete'>
        <name>1</name>
       </Loopback>
      </interface>
    </native>
```

Custom RPC Run Save Clear Copy

10: Get Complete Config:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <get-config>
      <source>
          <running/>
      </source>
      <filter type="subtree">
             <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-
native">
            </native>
         </filter>
  </get-config>
</rpc>
```

Custom RPC Run Save Clear Copy

11: Get the filtered configuration:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <version>16.12</version>
      <boot-start-marker/>
      <boot-end-marker/>
      <memory>
        <free>
          <low-watermark>
            <processor>72291</processor>
          </low-watermark>
        </free>
      </memory>
```

Custom RPC Run Save Clear Copy

Enable Candidate Data Store

Conf t
user admin privilege 15 secret cisco123
aaa new-model
aaa authentication login default local
aaa authorization exec default local
Netconf-yang feature candidate-datastore
Netconf-yang

12: Enable Candidate data store:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <source>
          <candidate/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet2</name>
          <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
            <address/>
          </ipv4>
```

Custom RPC Run Save Clear Copy

13: make the changes on candidate data store:
Change the interface status to down (Gi2) in candidate data store:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>GigabitEthernet2</name>
          <enabled>false</enabled>
        </interface>
      </interfaces>
```

Custom RPC Run Save Clear Copy

Commit the changes:

Encoding Console

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <commit></commit>
</rpc>
```

Custom RPC Run Save Clear Copy

Create the Loopback interface on candidate data store:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
   <Loopback>
    <name>1</name>
    <ip>
     <address>
      <primary>
       <address>10.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
```

Custom RPC Run Save Clear Copy

Commit the changes:

Encoding Console

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <commit></commit>
</rpc>
```

Custom RPC Run Save Clear Copy

Change the router Hostname on candidate data store and commit the change:

NetConf RestConf RPC Python YDK Capabilities

Encoding Console

```
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-
native">
      <hostname>R1</hostname>
    </native>
  </config>
</edit-config>
</rpc>
```

Custom RPC Run Save Clear Copy

Commit the changes:

Encoding | Console

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <commit></commit>
</rpc>
```

Custom RPC | Run | Save | Clear | Copy

15. Copy the configuration from running data store to candidate data store:

NetConf | RestConf | RPC | Python | YDK | Capabilities

Encoding | Console

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <copy-config>
      <target>
         <candidate/>
      </target>
      <source>
          <running/>
      </source>
  </copy-config>
</rpc>
```

Custom RPC | Run | Save | Clear | Copy

16: Close the session:

NetConf | RestConf | RPC | Python | YDK | Capabilities

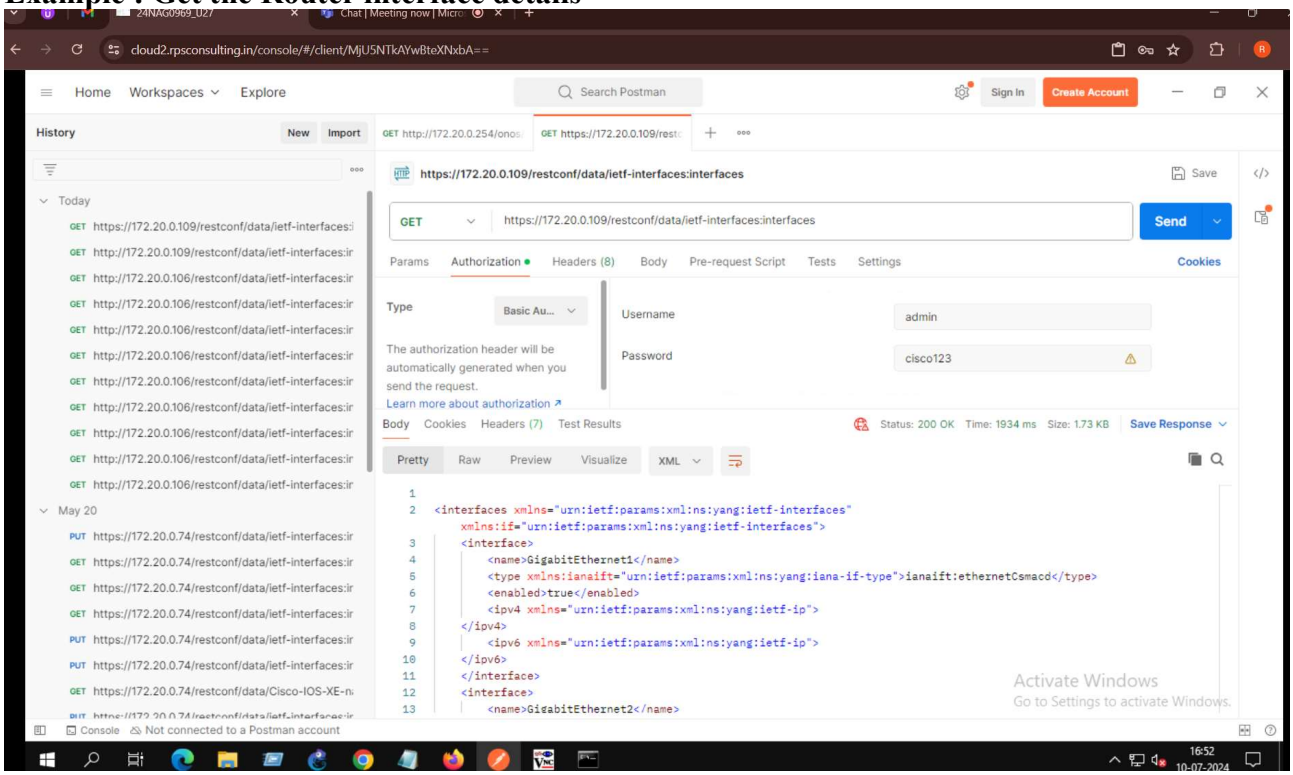
Encoding | Console

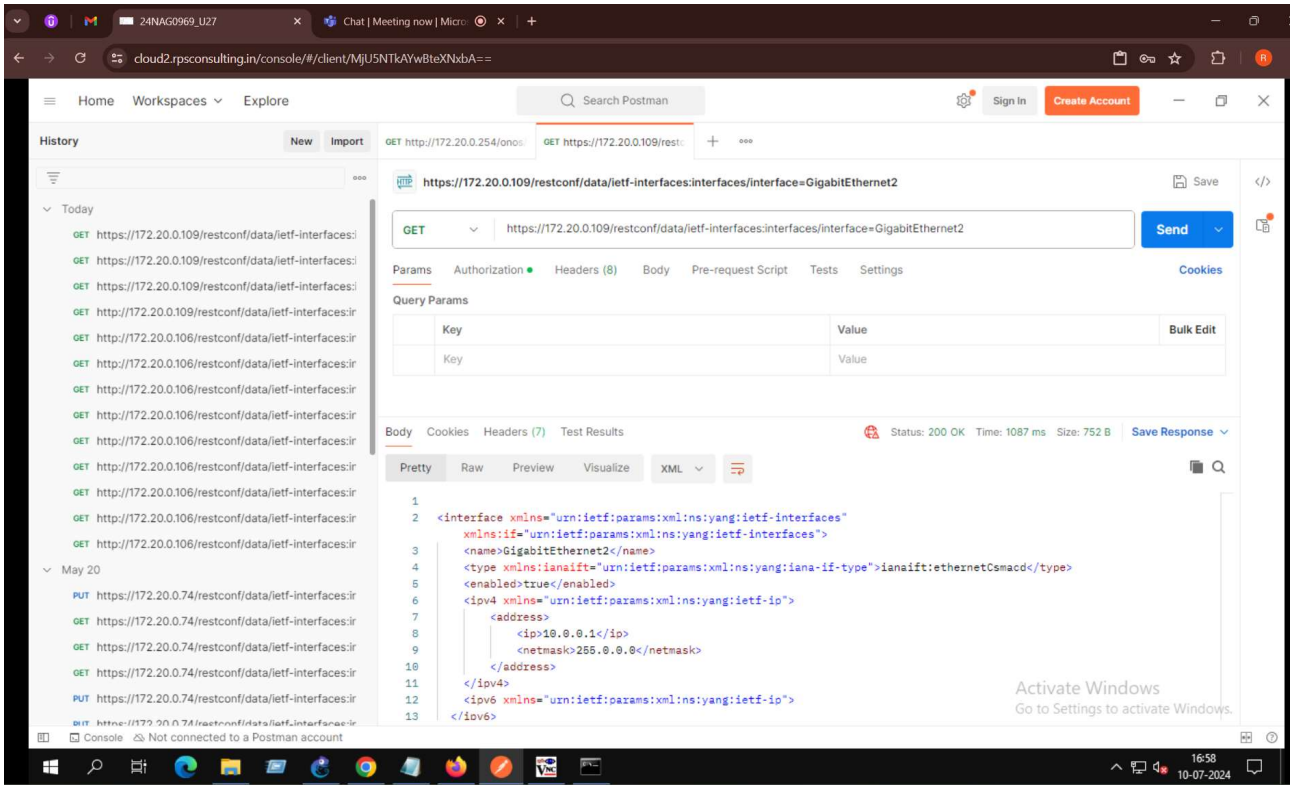
```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <close-session/>
</rpc>
```

Custom RPC | Run | Save | Clear | Copy

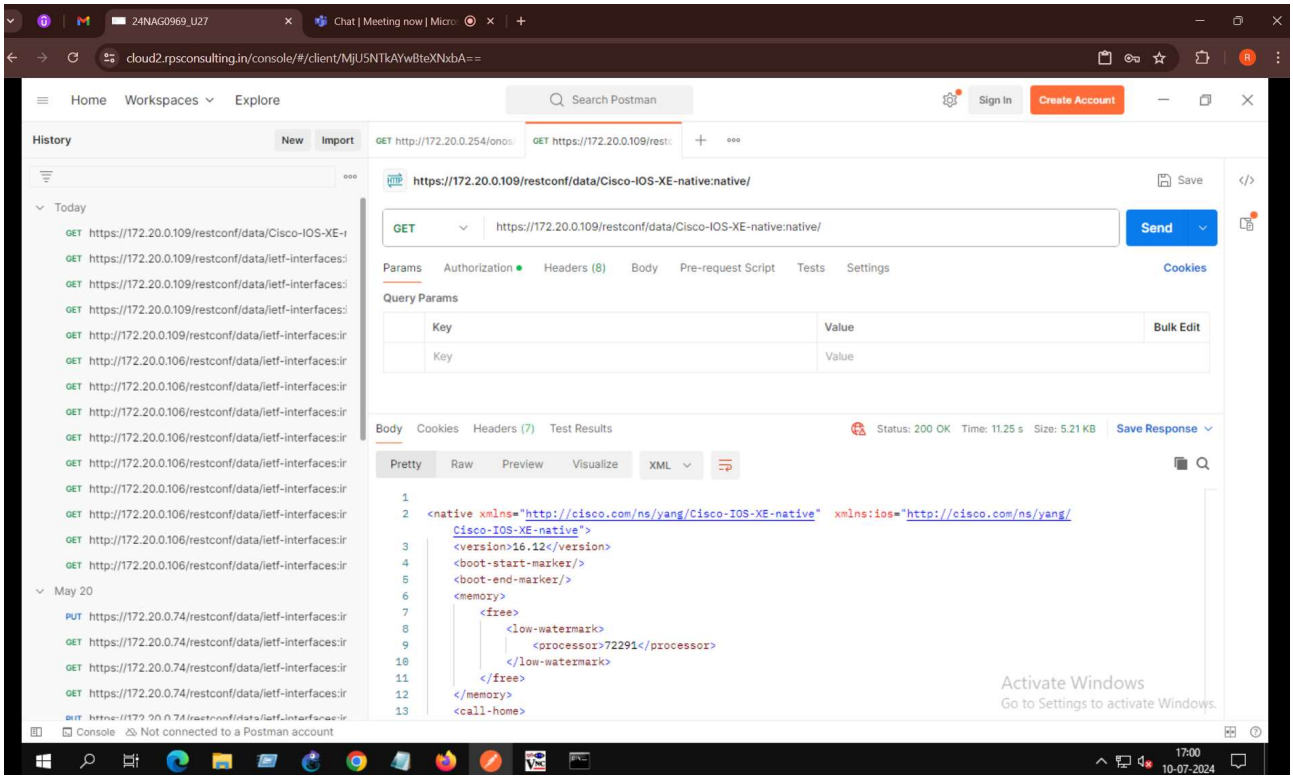
RestConf:
Enable the router with Restconf capability
Conf t
Restconf
Ip http secure-server

Login to Postman:
Example : Get the Router interface details





Get Config



Example :

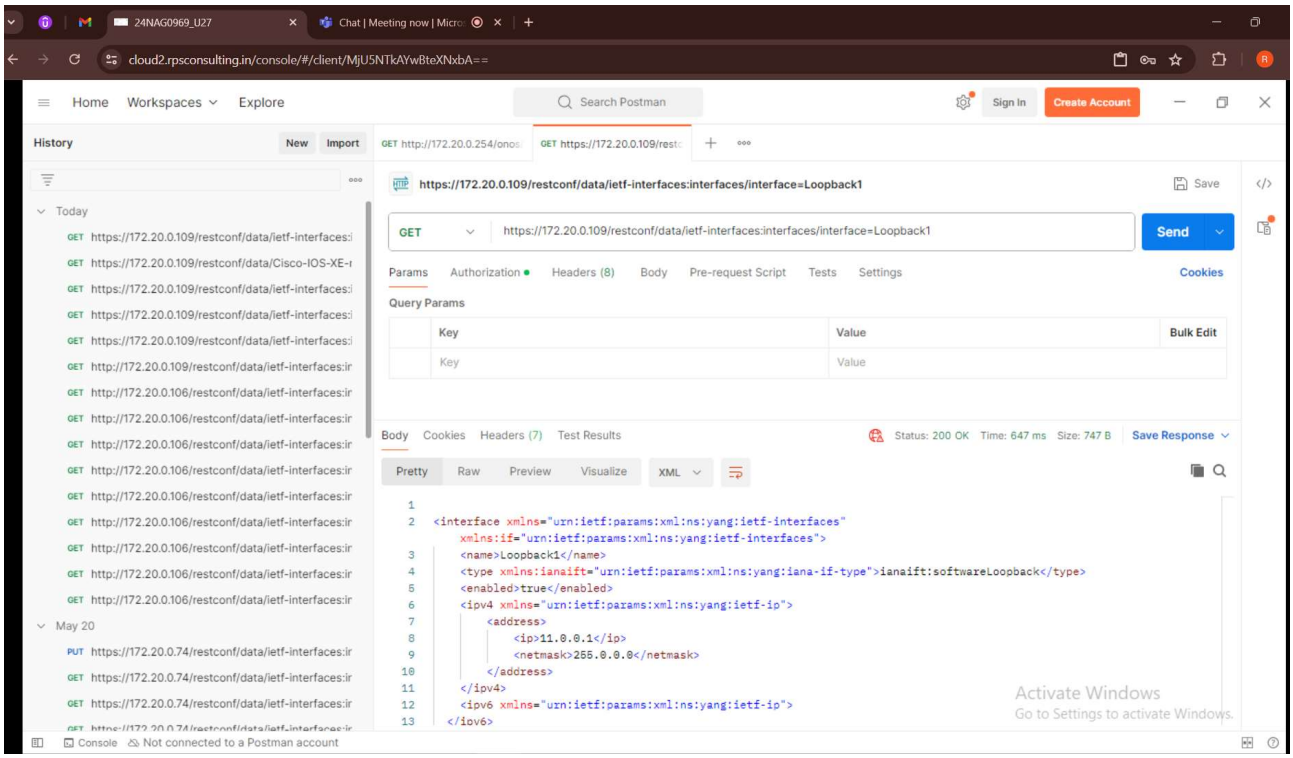
Create the new Loopback interface on router via CLI

Conf t

Int loopback1

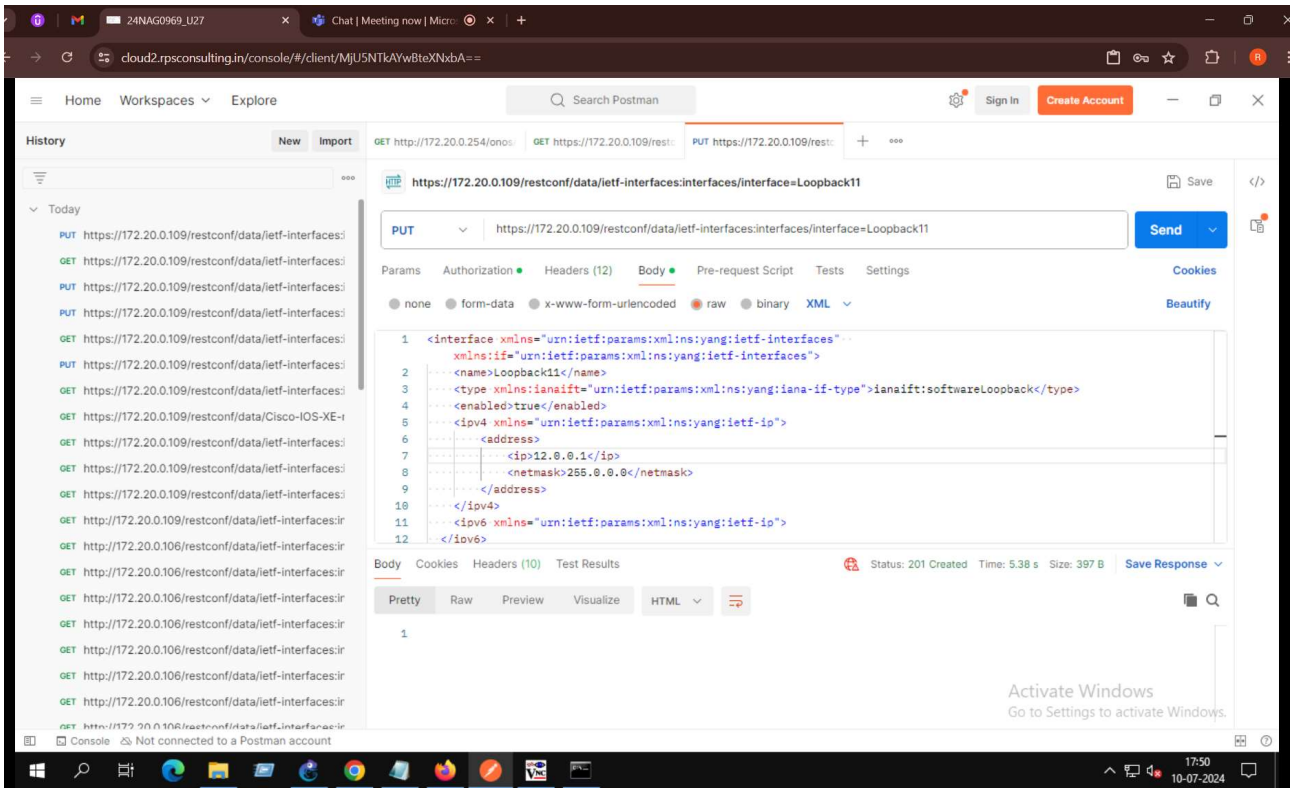
Ip address 11.0.0.1 255.0.0.0

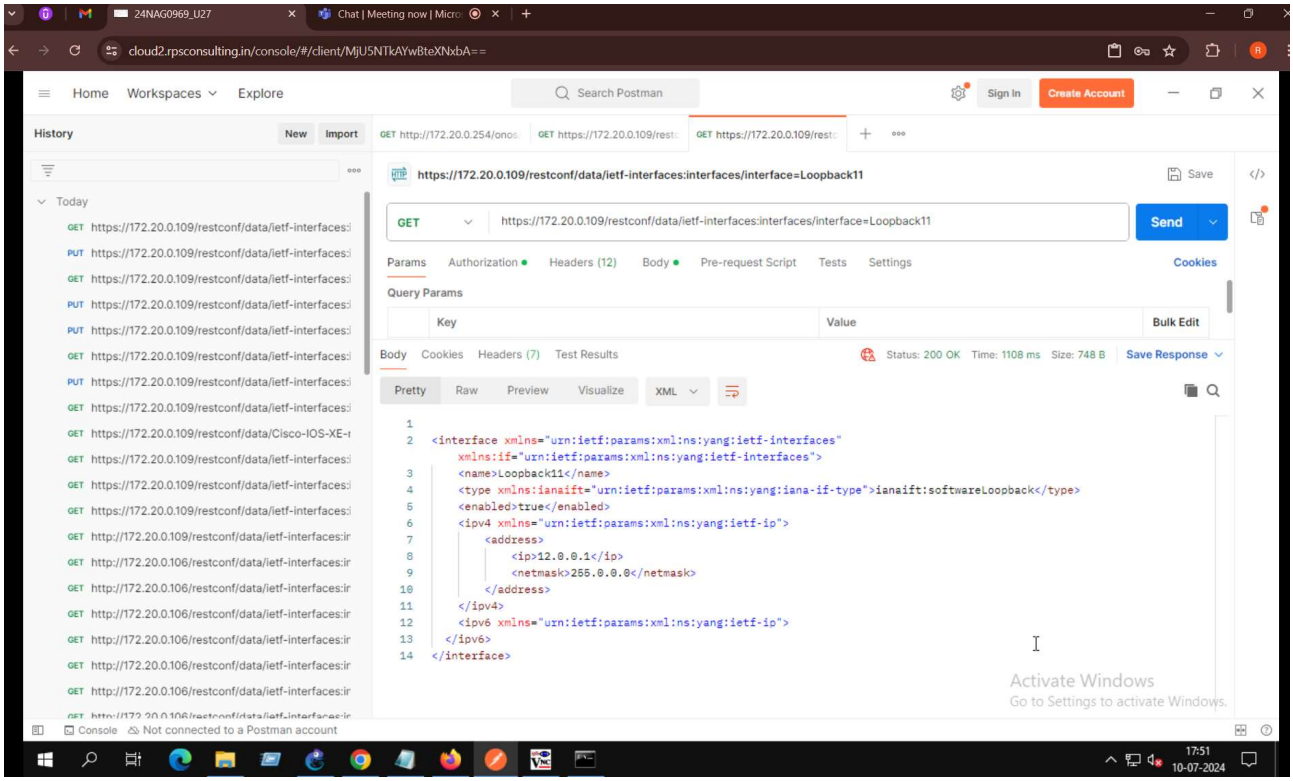
GET : https://IP address of server/restconf/data/ietf-interfaces:interfaces/interface=Loopback1



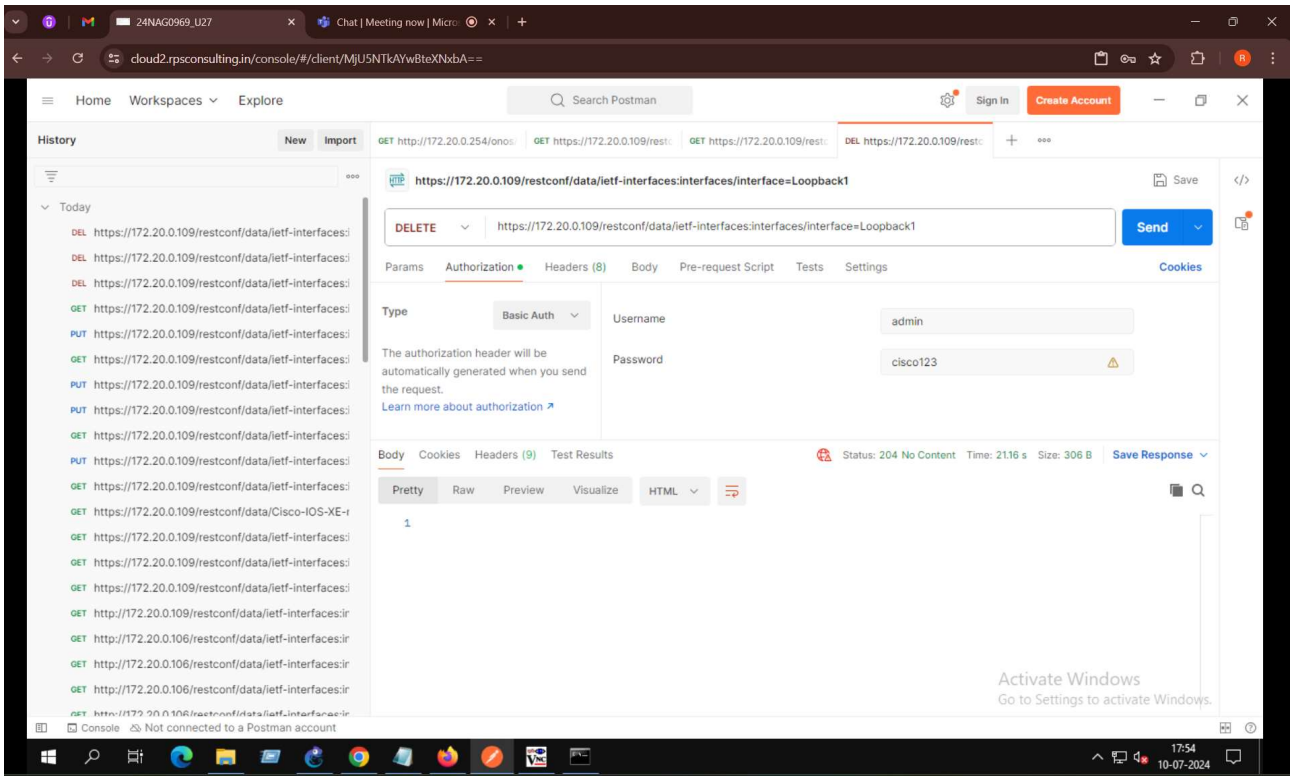
Example : Create the Loopback interface via Postman

PUT : https://IP address of server/restconf/data/ietf-interfaces:interfaces/interface=Loopback11

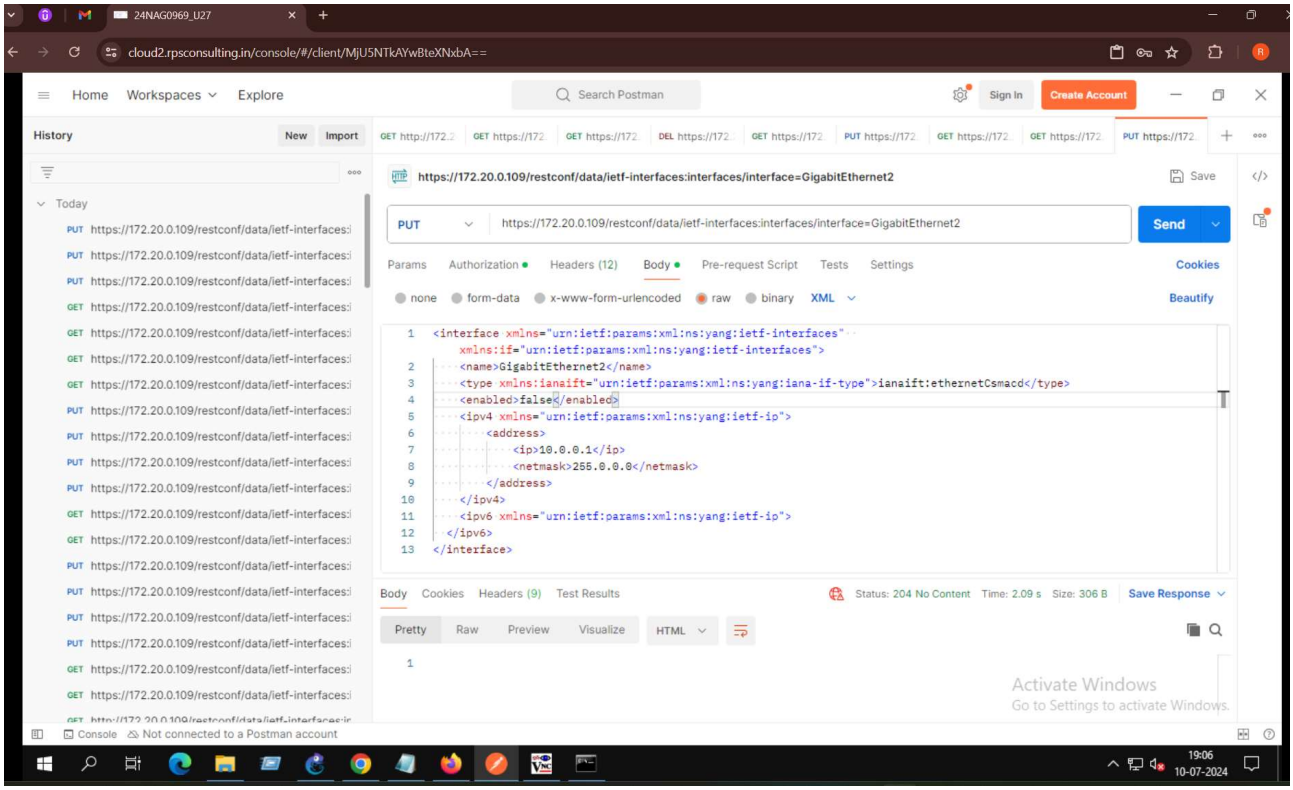




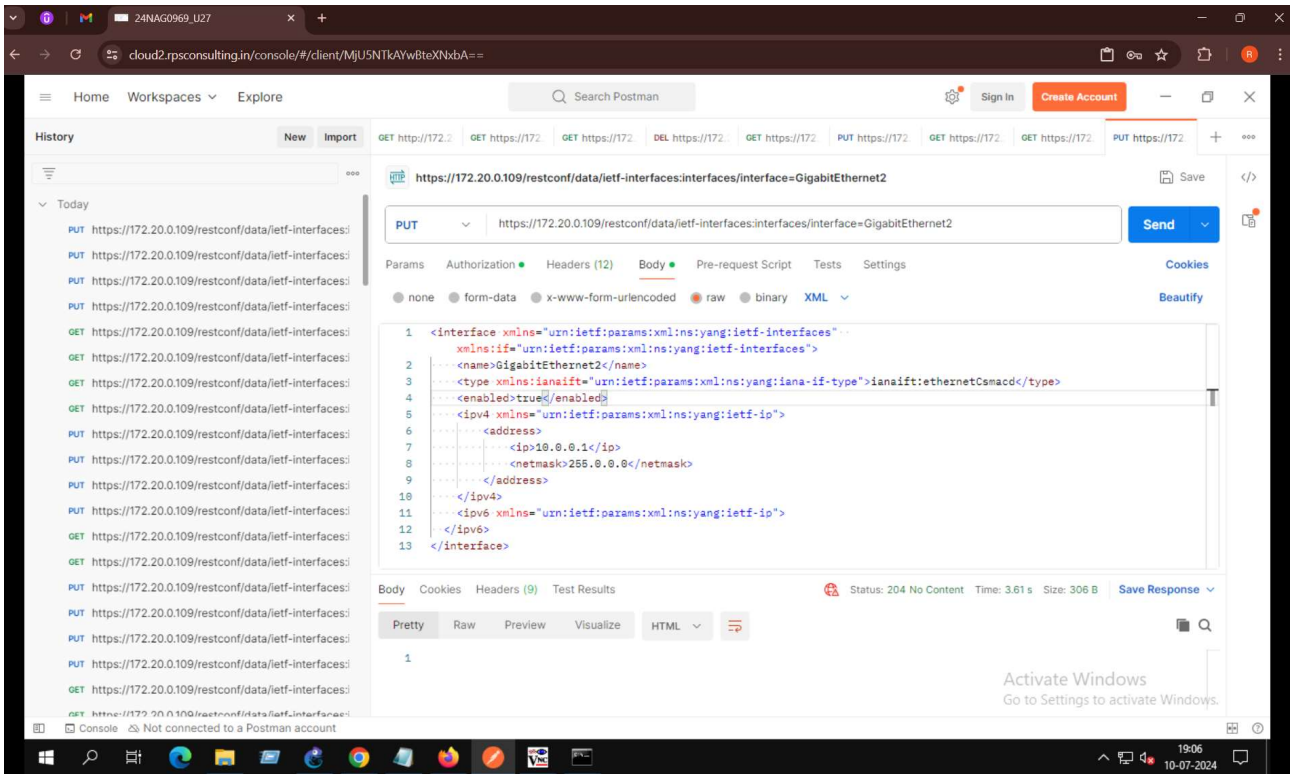
Delete the Loopback interface.



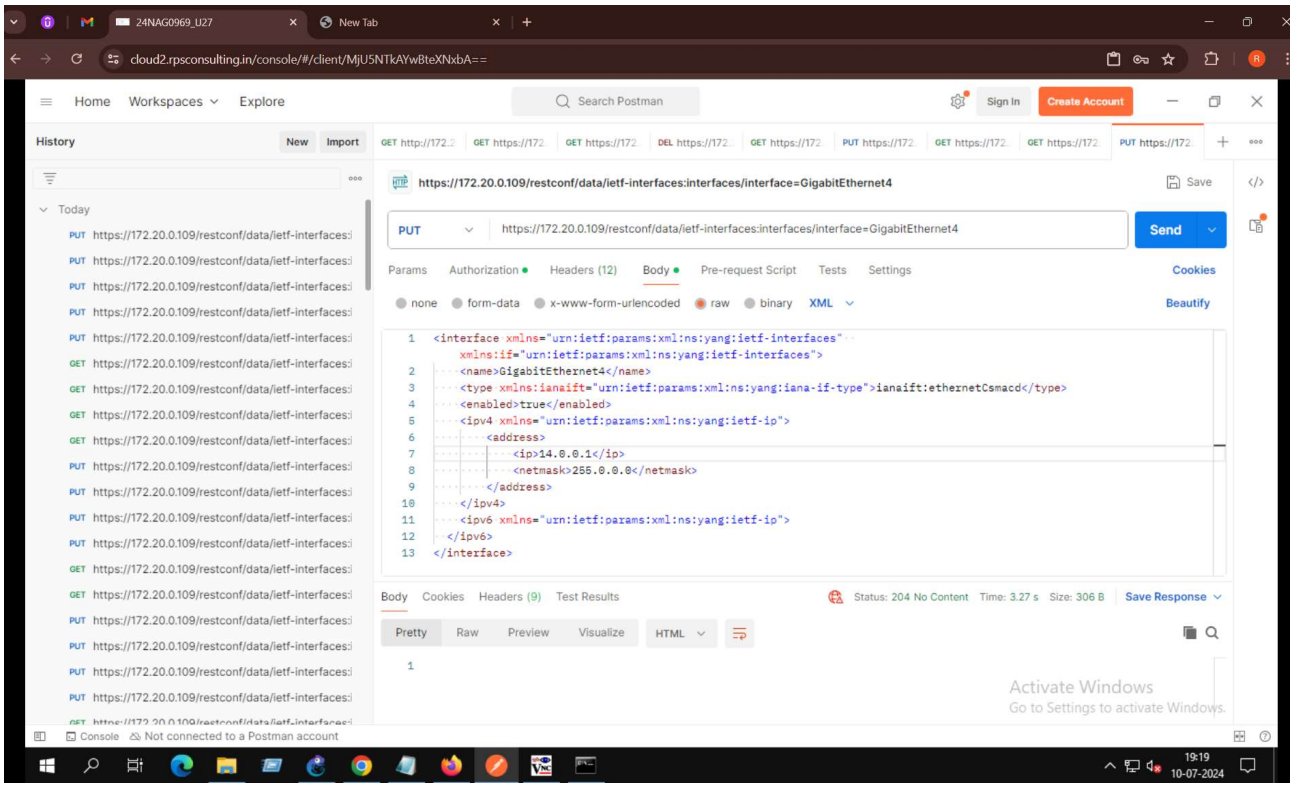
Change the Interface status to down for GigabitEthernet2 interface



Change the Interface status to UP for GigabitEthernet2 interface



Assign the interface IP address



Remove the Interface IP address