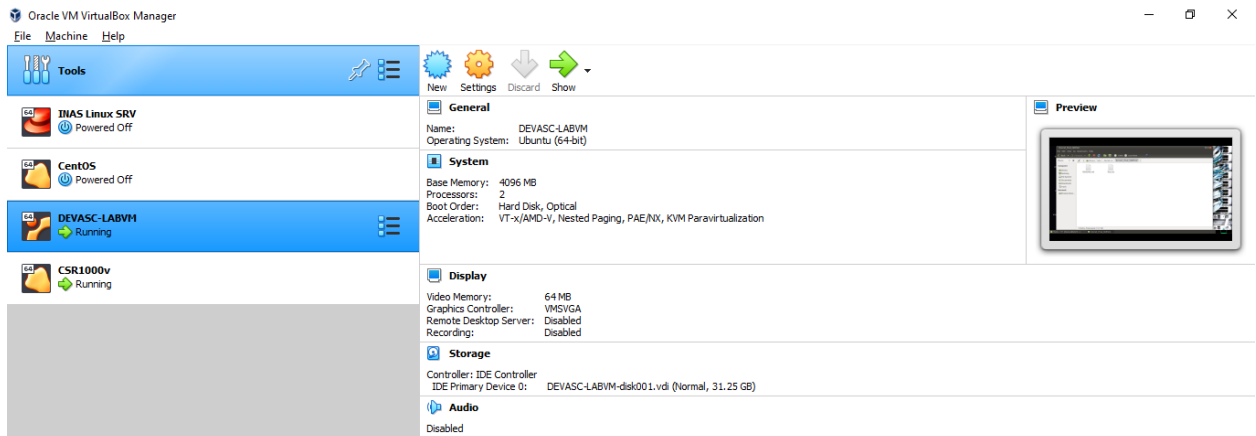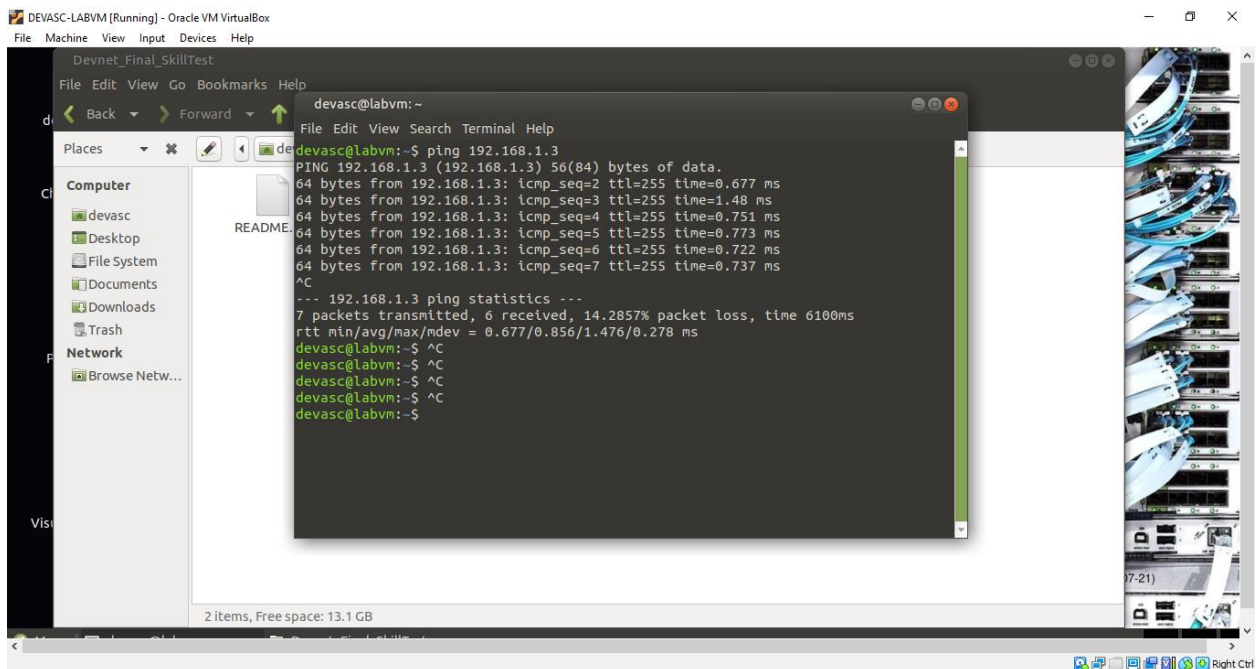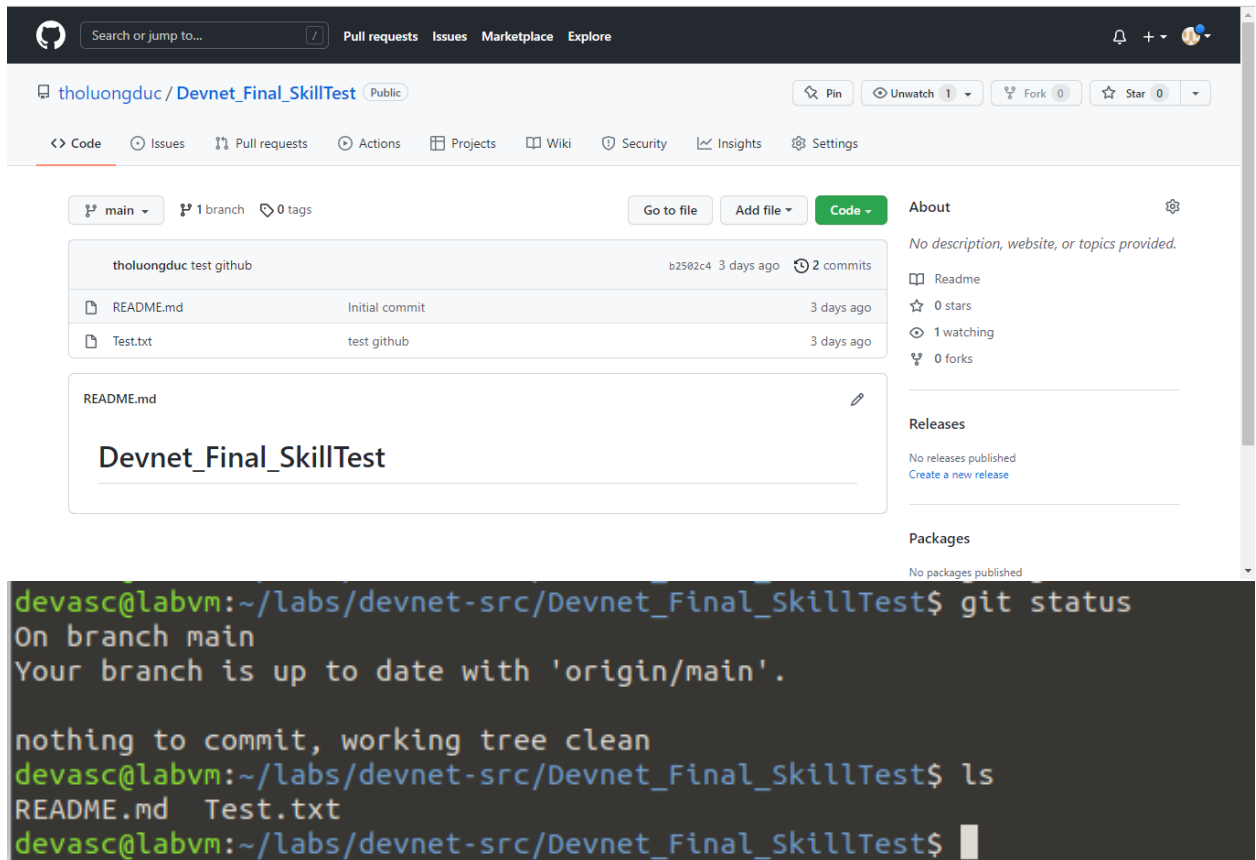# DEVNET SKILL EXAM - 2022

Tho Luong Duc

1. Open CSR1000v router on VirtualBox

Check connection to router: ping successful to router's IP address 192.168.1.3.

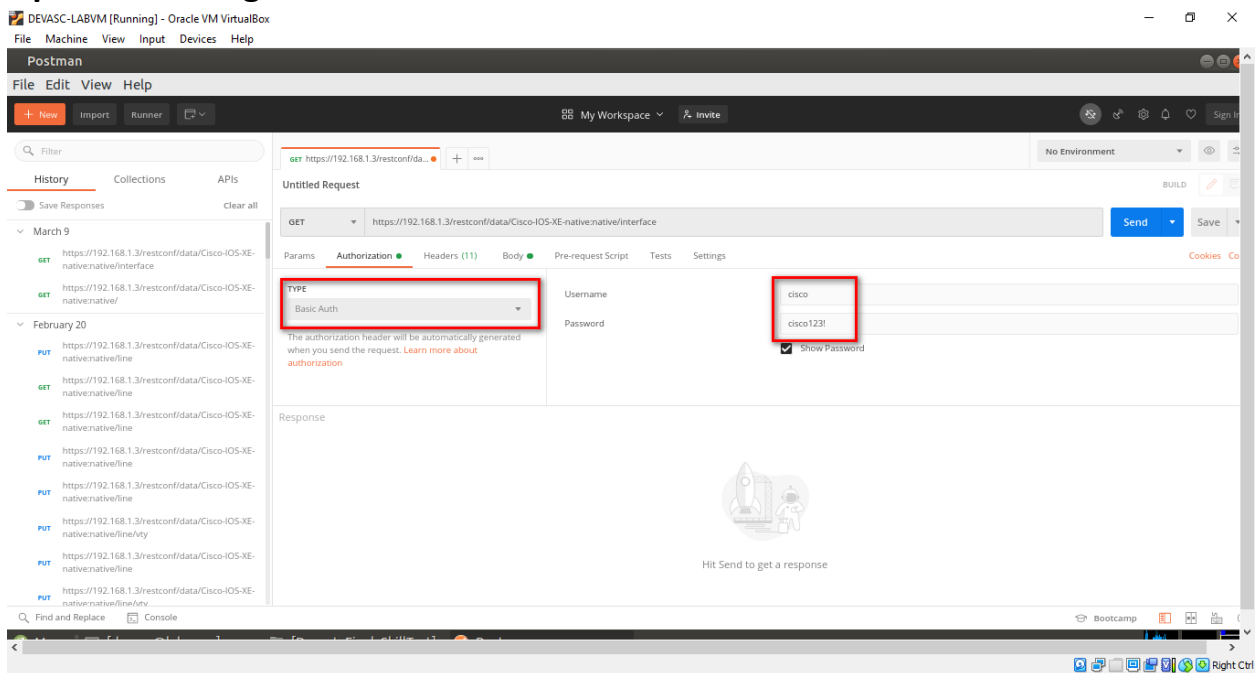## 2. Create a repository on Github and clone on local machine

3. Open and configure POSTMAN

4. Activities using POSTMAN
- Get device information
  - Version of IOS
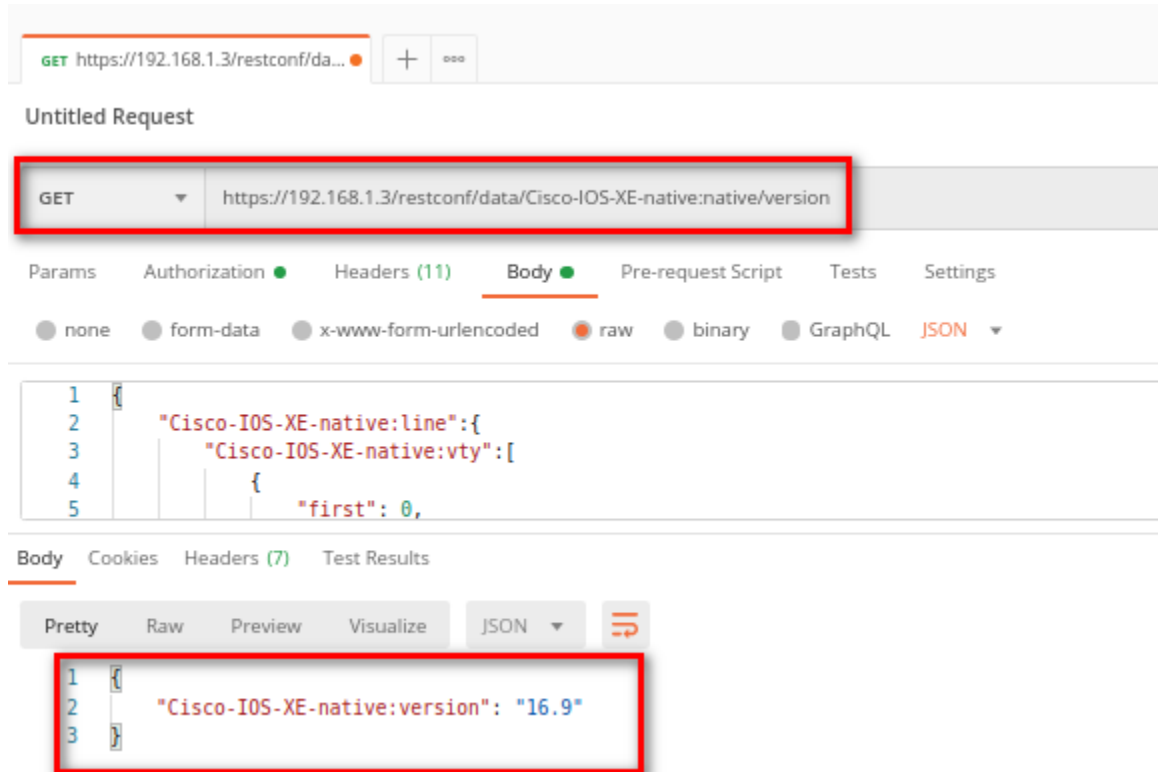
- **Routing Information**


```json
{
    "Cisco-IOS-XE-native:router": {
        "Cisco-IOS-XE-ospf:ospf": [
            {
                "id": 1,
                "network": [
                    {
                        "ip": "10.1.1.0",
                        "mask": "0.0.0.255",
                        "area": 10
                    },
                    {
                        "ip": "10.2.1.0",
                        "mask": "0.0.0.255",
                        "area": 20
                    },
                    {
```

- **Interface Information**


```json
{
    "Cisco-IOS-XE-native:interface": {
        "GigabitEthernet": [
            {
                "name": "1",
                "description": "VBox",
                "ip": {
                    "address": {
                        "primary": {
                            "address": "192.168.1.3",
                            "mask": "255.255.255.0"
                        }
                    }
                },
                "mop": {
                    "enabled": false,
                    "sysid": false
```

- Configure some parameters:
  - Create interface loopback 101
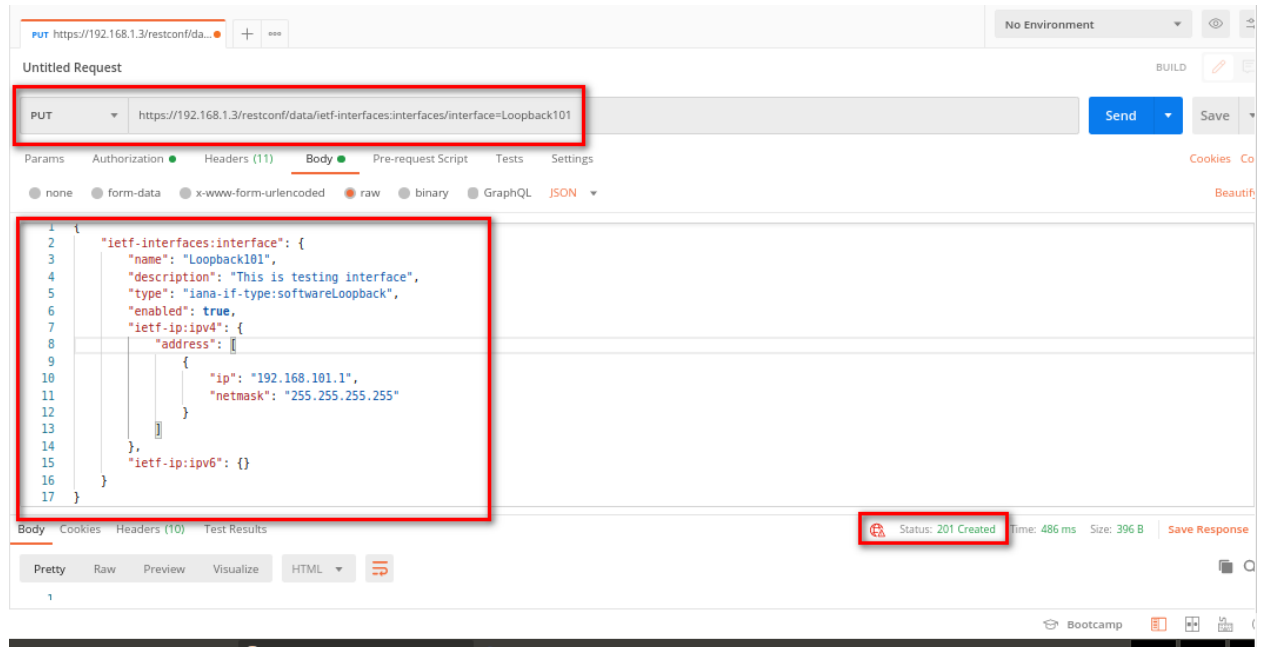
5. Configure device for RESTCONF Access

- Use python script to send GET request:

```python
import json
import requests
import sys

requests.packages.urllib3.disable_warnings()

headers = { "Accept": "application/yang-data+json",
            "Content-type":"application/yang-data+json"
            }
basicauth = ("cisco", "cisco123!")
sys.stdout = open("Python_Get_Request_Result.txt","w")
#Get IOS Version
api_url0 = "https://192.168.1.3/restconf/data/Cisco-IOS-XE-native:native/version"
resp0 = requests.get(api_url0, auth=basicauth, headers=headers, verify=False)
response0_json = resp0.json()
print(f'IOS Version: \n {response0_json}')
#Get Routing Information
api_url1 = "https://192.168.1.3/restconf/data/Cisco-IOS-XE-native:native/router"
resp1 = requests.get(api_url1, auth=basicauth, headers=headers, verify=False)
response1_json = resp1.json()
print(f'Routing Information: \n {response1_json}')
#Get Interfaces Information
api_url2 = "https://192.168.1.3/restconf/data/Cisco-IOS-XE-native:native/interface"
resp2 = requests.get(api_url2, auth=basicauth, headers=headers, verify=False)
response2_json = resp2.json()
print(f'Interface Information: \n {response2_json}')
sys.stdout.close()
```

- Use a python script to send PUT request:

```
Python_script_PUT_Request.py ×

home > devasc > labs > devnet-src > Devnet_Final_SkillTest > Python_script_PUT_Request.py
  1   import json
  2   import requests
  3   import sys
  4
  5   requests.packages.urllib3.disable_warnings()
  6
  7   headers = { "Accept": "application/yang-data+json",
  8               "Content-type":"application/yang-data+json"
  9             }
 10   basicauth = ("cisco", "cisco123!")
 11   sys.stdout = open("Python_PUT_Request_Result.txt","w")
 12
 13   api_url = "https://192.168.1.3/restconf/data/ietf-interfaces:interfaces/interface=Loopback102"
 14
 15   sys.stdout = open("Python_PUT_Request_Result.txt","w")
 16   #Create interface Loopback 102
 17   yangConfig ={
 18       "ietf-interfaces:interface": {
 19           "name": "Loopback102",
 20           "description": "This is created by Python script",
 21           "type": "iana-if-type:softwareLoopback",
 22           "enabled": True,
 23           "ietf-ip:ipv4": {
 24               "address": [
 25                   {
 26                       "ip": "192.168.102.1",
 27                       "netmask": "255.255.255.255"
 28                   }
 29               ]
 30           },
 31           "ietf-ip:ipv6": {}
 32       }
 33   }
 34   resp = requests.put(api_url, data=json.dumps(yangConfig), auth=basicauth, headers=headers, verify=False)
 35
 36   if(resp.status_code >= 200 and resp.status_code <= 299):
```

6. Upload all files to Github
   Github link: https://github.com/tholuongduc/Devnet_Final_SkillTest.git

```
devasc@labvm:~/labs/devnet-src/Devnet_Final_SkillTest$ git add *
devasc@labvm:~/labs/devnet-src/Devnet_Final_SkillTest$ git commit -m "Devnet Skill Exam 2022"
[main 4772ab9] Devnet Skill Exam 2022
 5 files changed, 166 insertions(+)
 create mode 100644 POSTMAN_Get Result.txt
 create mode 100644 Python_Get_Request_Result.txt
 create mode 100644 Python_PUT_Request_Result.txt
 create mode 100644 Python_script_GET_request.py
 create mode 100644 Python_script_PUT_Request.py
devasc@labvm:~/labs/devnet-src/Devnet_Final_SkillTest$ git push origin main
Username for 'https://github.com': tholuongduc
Password for 'https://tholuongduc@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.58 KiB | 2.58 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/tholuongduc/Devnet_Final_SkillTest.git
   b2502c4..4772ab9  main -> main
devasc@labvm:~/labs/devnet-src/Devnet_Final_SkillTest$
```