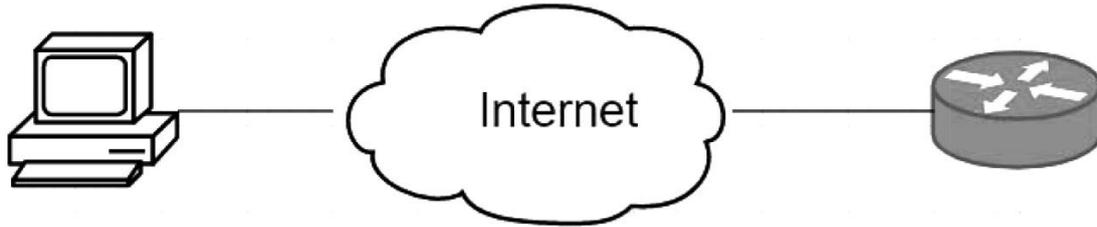


LAB - VIẾT CHƯƠNG TRÌNH DÙNG NGÔN NGỮ PYTHON SỬ DỤNG NETCONF ĐỂ THỰC HIỆN LẤY DANH SÁCH THÔNG TIN CÁC CỔNG TRÊN THIẾT BỊ CISCO (SANDBOX)

I. Sơ đồ



Hình 1: Sơ đồ bài Lab

II. Mô tả

- Sơ đồ bài thực hành gồm 1 PC và 1 thiết bị chạy hệ điều hành IOS XE (CSR1000V) được đấu nối với nhau như hình 1.

- Trên sơ đồ này, học viên thực hiện kết nối đến Router CSR, viết chương trình để lấy thông tin bằng ngôn ngữ Python.

- Máy PC phải đáp ứng yêu cầu đã cài đặt trạm làm việc cho developer.

III. Yêu cầu kỹ thuật

- Học viên thực hiện kết nối máy tính đến Agent bằng mạng Internet.

- Cài đặt thư viện ncclient, xmlltodict trên máy tính.

- Viết code bằng Python thực hiện yêu cầu:

+ Tạo bộ lọc XML cho truy vấn Netconf.

+ Mở kết nối đến thiết bị bằng ncclient và tạo câu truy vấn Netconf (get_config) sử dụng bộ lọc trên.

+ In kết quả trả về dạng XML.

+ Tạo danh sách interfaces và in báo cáo trạng thái các cổng up/down.

IV. Các bước thực hiện

4.1. Cài đặt thư viện

- Mở cmd bằng tổ hợp phím Win+R: gõ cmd.
- Gõ python -m pip install ncclient - -user.

```
C:\Users\TuanHoanh>python -m pip install ncclient --user
Collecting ncclient
  Using cached https://files.pythonhosted.org/packages/36/b1/1f909193588dfd35726e8ae35e01463386bbcf670b90ed61c4930ce447d
b/ncclient-0.6.6.tar.gz
Requirement already satisfied: setuptools>0.6 in c:\program files\python37\lib\site-packages (from ncclient) (40.8.0)
Requirement already satisfied: paramiko>=1.15.0 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (fro
m ncclient) (2.6.0)
Requirement already satisfied: lxml>=3.3.0 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from ncc
lient) (4.4.1)
Requirement already satisfied: six in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from ncclient) (
1.12.0)
Requirement already satisfied: pynacl>=1.0.1 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from p
aramiko>=1.15.0->ncclient) (1.3.0)
Requirement already satisfied: cryptography>=2.5 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (fr
om paramiko>=1.15.0->ncclient) (2.7)
Requirement already satisfied: bcrypt>=3.1.3 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from p
aramiko>=1.15.0->ncclient) (3.1.7)
Requirement already satisfied: cffi>=1.4.1 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from pyn
acl>=1.0.1->paramiko>=1.15.0->ncclient) (1.12.3)
Requirement already satisfied: asn1crypto>=0.21.0 in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (f
rom cryptography>=2.5->paramiko>=1.15.0->ncclient) (0.24.0)
Requirement already satisfied: pycparser in c:\users\tuanhoanh\appdata\roaming\python\python37\site-packages (from cffi>
=1.4.1->pynacl>=1.0.1->paramiko>=1.15.0->ncclient) (2.19)
Installing collected packages: ncclient
  Running setup.py install for ncclient ... done
Successfully installed ncclient-0.6.6
```

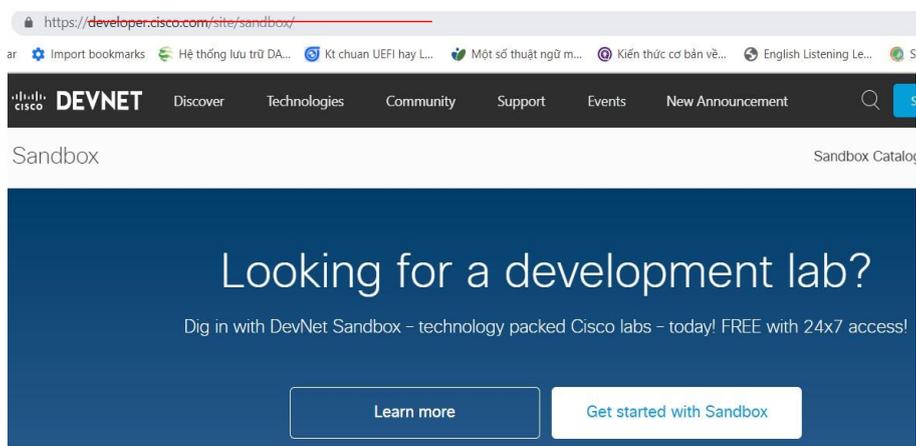
- Gõ python -m pip install xmldict - -user:

```
C:\Users\TuanHoanh>python -m pip install xmldict --user
Collecting xmldict
  Using cached https://files.pythonhosted.org/packages/28/fd/30d5c1d
5/xmldict-0.12.0-py2.py3-none-any.whl
Installing collected packages: xmldict
Successfully installed xmldict-0.12.0
```

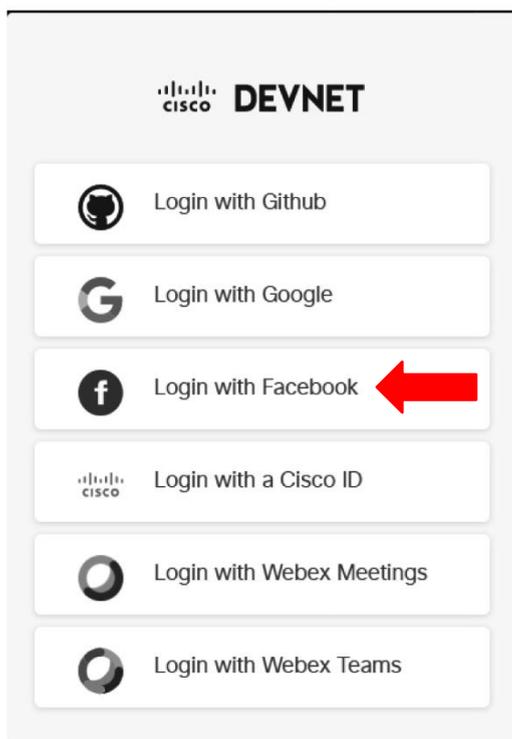
Kiểm tra đi được Internet: ping 8.8.8.8

Vào sandbox của cisco để lấy thông tin CSR1000V:

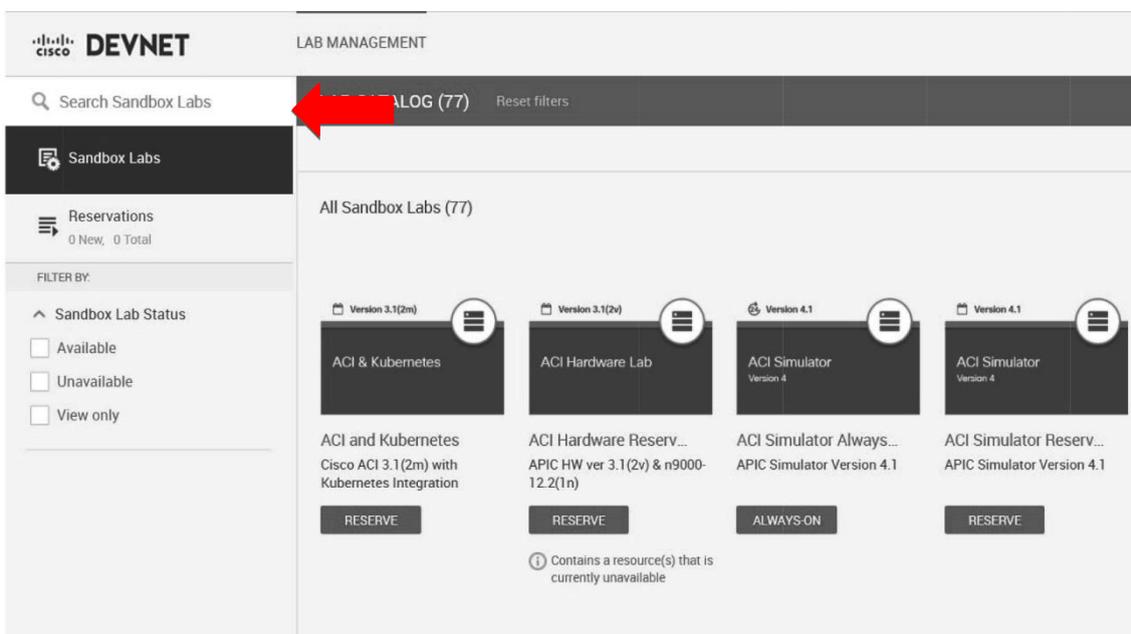
- Vào chrome search từ khóa: devnet sandbox tìm trang như hình sau:



Đăng nhập vào website devnet dùng cisco ID hoặc dùng tài khoản Facebook, Google:



Khi đăng nhập xong màn hình sẽ hiện lên giao diện web tất cả sandbox lab:



Vào ô Search Sandbox Labs gõ IOS XE để tìm kiếm các lab sử dụng IOS XE, chọn lab có tên Recommended Code AlwaysOn:

The screenshot shows the 'LAB CATALOG (6)' page for 'IOS XE'. On the left, there is a sidebar with 'Sandbox Labs' and 'Reservations' sections. The main area displays 'All Sandbox Labs (6)' with a grid of lab cards. Each card includes a version number, a title (e.g., 'IOS XE on Catalyst 9000 16.12 EFT Code'), a description, and a 'RESERVE' or 'ALWAYS ON' button. A large grey arrow points downwards from the top right of the grid.

Ở phía bên trái kéo xuống phần Access Details ta có được thông tin để kết nối đến CSR1000V:

The screenshot shows the 'DEVNET LAB MANAGEMENT' interface. The top navigation bar includes 'SANDBOX LAB' and the selected lab name 'IOS XE on CSR Recommended Code AlwaysOn'. Below the navigation bar, there are tabs for 'WINDOW: i INSTRUCTIONS', 'NAVIGATOR', 'STYLE', and 'BEHAVIOUR: u'. The 'INSTRUCTIONS' tab is active, displaying the following content:

NETCONF/YANG-RESTCONF

shared interface. This is a place to explore, learn or verify interoperability.

Access Details:

Developers and network engineers access the **IOS XE on CSR Recommended Code Always On** Sandbox directly using the following information:

- CSR1000V Host: ios-xe-mgmt.cisco.com
 - SSH Port: 8181
 - NETCONF Port: 10000
 - RESTCONF Ports: 9443 (HTTPS)
- Username: **developer**
- Password: **C1sco12345**

Hello World:

Check out this repository for various samples to use as examples to

4.2. Viết chương trình

Tạo file `device_info.py` để ghi thông tin thiết bị:

```
iosxe = {  
    "address": "ios-xe-mgmt.cisco.com",  
    "netconf_port": 10000,  
    "restconf_port": 9443,  
    "ssh_port": 8181,  
    "username": "developer",  
    "password": "C1sco12345"  
}
```

Các thông tin: `netconf_port`, `restconf_port`, `ssh_port` là các port do thiết bị trên Sandbox đặt. Username, password là tên và mật khẩu khi cấu hình SSH.

Tạo file `get_interface_list.py`:

+ Import các thư viện cần thiết:

```
from ncclient import manager  
import sys  
import xmltodict  
import xml.dom.minidom  
  
from device_info import iosxe as device
```

+ Viết bộ lọc XML cho truy vấn Netconf:

```
netconf_filter = """  
<filter>  
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">  
<interface></interface>  
</interfaces>  
</filter>"""
```

“`xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"`” là không gian tên, giúp phân biệt các thực thể cùng tên nằm trong các không gian khác nhau. Ví dụ như trong namespace N1 có object O, N2 cũng có object O. Để phân biệt object O của namespace nào thì có thể ghi N1:O để biết là object O này của N1.

Mở kết nối đến thiết bị bằng ncclient và tạo câu truy vấn Netconf (get_config) sử dụng bộ lọc trên:

```
print("Opening NETCONF Connection to {}".format(device["address"]))
with manager.connect(
    host=device["address"],
    port=device["netconf_port"],
    username=device["username"],
    password=device["password"],
    hostkey_verify=False
) as m:
    print("Sending a <get-config> operation to the device.\n")
    netconf_reply = m.get_config(source = 'running',
                                filter = netconf_filter)
```

Tạo kết nối đến thiết bị bằng thư viện ncclient (modun manager).

Device là tên gán từ dòng “from device_info import iosxe as device” khai báo ở trên.

Lưu ý: Phần khoảng cách phải đúng.

+ In kết quả trả về dạng XML:

```
print("Here is the raw XML data returned from the device.\n")
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
print("")
```

“xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml()” dùng để in kết quả trả về giúp ta dễ đọc hơn.

+ Tạo danh sách interfaces và in báo cáo trạng thái các cổng up/down:

```
netconf_data = xmldict.parse(netconf_reply.xml)
                ["rpc-reply"]["data"]

interfaces = netconf_data["interfaces"]["interface"]
print("The interface status of the device is: ")
for interface in interfaces:
    print("Interface {} enabled status is {}".format(
        interface["name"],
        interface["enabled"]
    )
        )
print("\n")
```

Xmltodict giúp ta chuyển từ dạng xml sang dictionary để có thể để vào vòng lặp for duyệt từng phần tử trong dictionary. Việc tạo danh sách này giúp ta biết được cổng nào đang up hay down nhanh chóng hơn.

Kết quả:

```
Sending a <get-config> operation to the device.

Here is the raw XML data returned from the device.

<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:b17cf8cb-8cfb-48aa-922b-06ef1f965fa3" 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>GigabitEthernet1</name>
        <description>MANAGEMENT INTERFACE - DON'T TOUCH ME</description>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:
ethernetCsmacd</type>
        <enabled>true</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>10.10.20.48</ip>
            <netmask>255.255.255.0</netmask>
          </address>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
      <interface>
        <name>GigabitEthernet2</name>
        <description>Configured by NETCONF</description>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:
ethernetCsmacd</type>
        <enabled>true</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>10.255.255.1</ip>
            <netmask>255.255.255.0</netmask>
          </address>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
      <interface>
        <name>GigabitEthernet3</name>
        <description>Configured by NETCONF</description>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:
ethernetCsmacd</type>
        <enabled>false</enabled>
        <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>10.255.255.1</ip>
            <netmask>255.255.255.0</netmask>
          </address>
        </ipv4>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"/>
      </interface>
    </interfaces>
  </data>
</rpc-reply>

The interface status of the device is:
Interface GigabitEthernet1 enabled status is true
Interface GigabitEthernet2 enabled status is true
Interface GigabitEthernet3 enabled status is false
Interface Loopback101 enabled status is true
Interface Loopback102 enabled status is true
Interface Loopback2101 enabled status is true
Interface Loopback2102 enabled status is true
Interface Loopback2103 enabled status is true
Interface Loopback2104 enabled status is true
Interface Loopback2105 enabled status is true
Interface Loopback2107 enabled status is true
Interface Loopback2108 enabled status is true
```