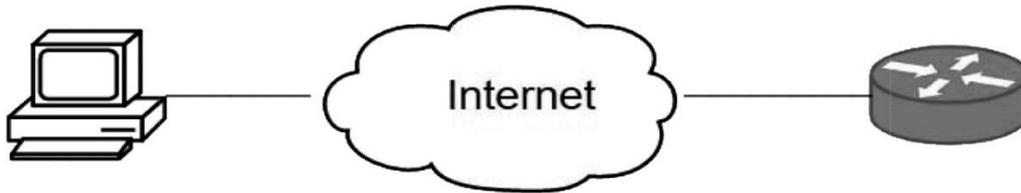


# LAB - VIẾT CHƯƠNG TRÌNH DÙNG NGÔN NGỮ PYTHON SỬ DỤNG NETCONF ĐỂ THÊM CỔNG LOOPBACK TRÊN THIẾT BỊ CISCO (SANDBOX)

## 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 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, xmltodict trên máy tính.

- Viết code bằng Python thực hiện yêu cầu:

+ Tạo template XML cho cổng.

+ Viết yêu cầu nhập thông tin cổng Loopback.

+ Tạo Netconf payload chứa dữ liệu cho cổng Loopback.

+ Tạo kết nối đến thiết bị bằng thư viện ncclient.

+ Tạo câu truy vấn Netconf sử dụng bộ lọc (payload trên).

+ In ra màn hình kết quả trả về.

## IV. Các bước thực hiện

### 4.1. Cài đặt thư viện

Thực hiện cài đặt thư viện như các công đoạn đã trình bày ở **Lab 13**.

### 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 `add_loopback.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
```

+ Khai báo kiểu cổng:

```
IETF_INTERFACE_TYPES = {  
    "loopback": "ianaift:softwareLoopback",  
    "ethernet": "ianaift:ethernetCsmacd"  
}
```

### + Tạo template XML cho công:

```
netconf_interface_template = ""
<config>
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
<interface>
    <name>{name}</name>
    <description>{desc}</description>
    <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
        {type}
    </type>
    <enabled>{status}</enabled>
    <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
        <address>
            <ip>{ip_address}</ip>
            <netmask>{mask}</netmask>
        </address>
    </ipv4>
</interface>
</interfaces>
</config>""
```

xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" là namespace (không gian tên), giúp phân biệt các thực thể cùng tên nằm trong các namespace 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.

### + Yêu cầu nhập thông tin công Loopback:

```
new_loopback = {}
new_loopback["name"] = "Loopback" + input("What loopback
number to add?")
new_loopback["desc"] = input("What description to use?")
new_loopback["type"] = IETF_INTERFACE_TYPES["loopback"]
new_loopback["status"] = "true"
new_loopback["ip_address"] = input("What IP address?")
new_loopback["mask"] = input("What network mask?")
```

+ Tạo Netconf payload chứa dữ liệu cho công:

```
netconf_data = netconf_interface_template.format(  
    name = new_loopback["name"],  
    desc = new_loopback["desc"],  
    type = new_loopback["type"],  
    status = new_loopback["status"],  
    ip_address = new_loopback["ip_address"],  
    mask = new_loopback["mask"]  
)
```

+ In ra màn hình thông tin sẽ gửi qua Netconf:

```
print("The configuration payload to be sent over NETCONF.\n")  
print(netconf_data)
```

+ 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 <edit-config> operation to the device.\n")  
netconf_reply = m.edit_config(netconf_data, target = 'running')
```

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.

Kết quả:

- Trước khi thêm cổng Loopback:

```
The interface status of the device is:
Interface GigabitEthernet1 enabled status is true
Interface GigabitEthernet2 enabled status is false
Interface GigabitEthernet3 enabled status is false
Interface Loopback3 enabled status is true
Interface Loopback16 enabled status is true
Interface Loopback319 enabled status is true
```

- Sau khi thêm cổng Loopback:

```
What loopback number to add? 15
What description to use? no
What IP address? 192.168.100.15
What network mask? 255.255.255.0
The configuration payload to be sent over NETCONF.

<config>
  <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
      <name>Loopback15</name>
      <description>no</description>
      <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
        ianaift:softwareLoopback
      </type>
      <enabled>true</enabled>
      <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
        <address>
          <ip>192.168.100.15</ip>
          <netmask>255.255.255.0</netmask>
        </address>
      </ipv4>
    </interface>
  </interfaces>
</config>

Opening NETCONF Connection to ios-xe-mgmt.cisco.com
Sending a <edit-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:b8e795a7-0a96-4255-b645-bb5723946f47" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```