Networking Academy

NOMBRE: **Isidro Lara Lopez**
CARRERA: **INFRAESTRUCTURA DE REDES DIGITALES**
DOCENTE: **BARRON RODRIGUEZ GABRIEL**
NO. CONTROL: **1221100381**
FECHA: **14 DE DICIEMBRE DEL 2022**
GRUPO: **GIR0441**

Lab – NETCONF w/Python: Get Operational Data

Objectives

Part 1: Retrieve the IOS XE VM's Operational Data - Statistics

Background / Scenario

In this lab, you will learn how to use NETCONF to retrieve operational data from the network device.

Required Resources

- Access to a router with the IOS XE operating system version 16.6 or higher
- Python 3.x environment

Instructions

Part 1: Retrieve Interface Statistics

In this part, you will use the ncclient module to retrieve the device's operational data. The data are returned back in XML form. In the following steps this data will be transformed into a tabular output.

Step 1: Use ncclient to retrieve the device's running configuration.

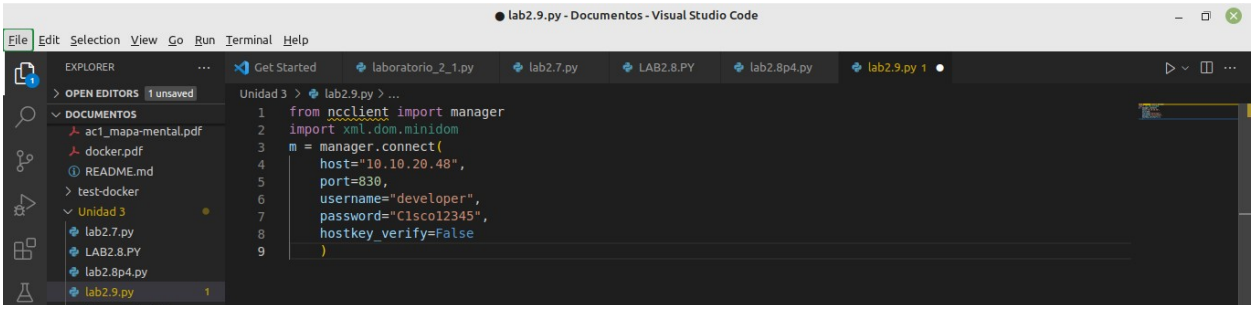
a. In Python IDLE, create a new Python script file:

b. In the new Python script file editor, import the "`manager`" class from the ncclient module and the xml.dom.minidom module:

```
from ncclient import manager
import xml.dom.minidom
```

c. Set up an **`m`** connection object using the `manager.connect()` function to the IOS XE device.

```
m = manager.connect(
        host="192.168.56.101",
        port=830,
        username="cisco",
        password="cisco123!",
        hostkey_verify=False
        )
```



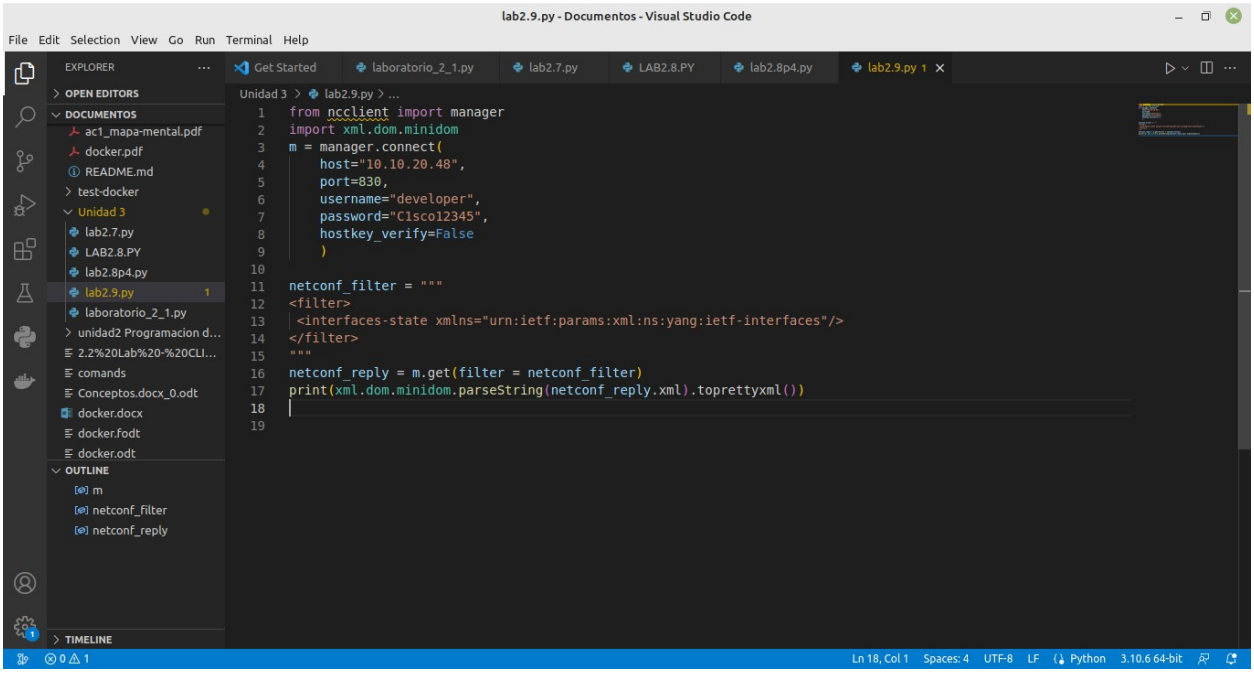
cambiamos el escrip con los datos correctos del dispositivo al que estamos conectados y utilizamos el modulo `ncclient` para recuperar la informacion del dispositivo.

d. After a successful NETCONF connection, using the "`get()`" function of the "**m**" NETCONF session object to retrieve and print the device's operational data. The `get()` function expects a "filter" string parameter that defines the NETCONF filter.

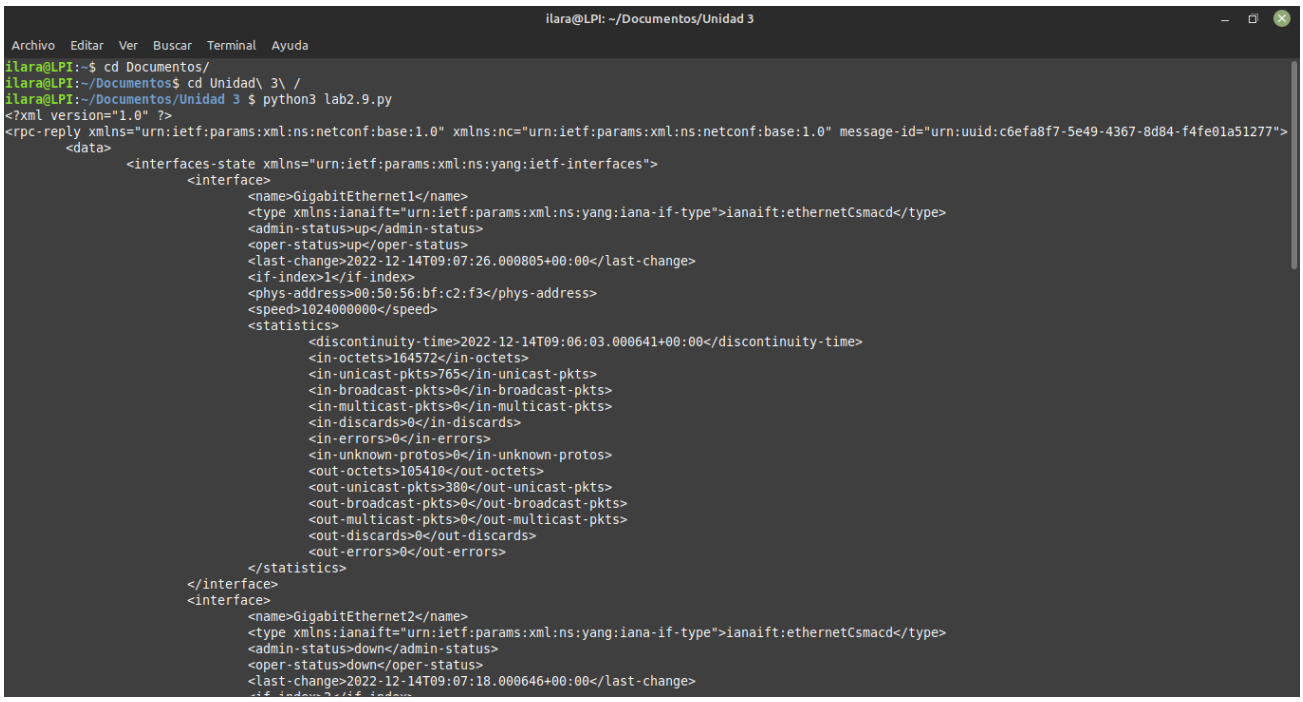
   The following filter retrieves the interfaces-state operational data (statistics), as defined in the ietf-interfaces YANG model:

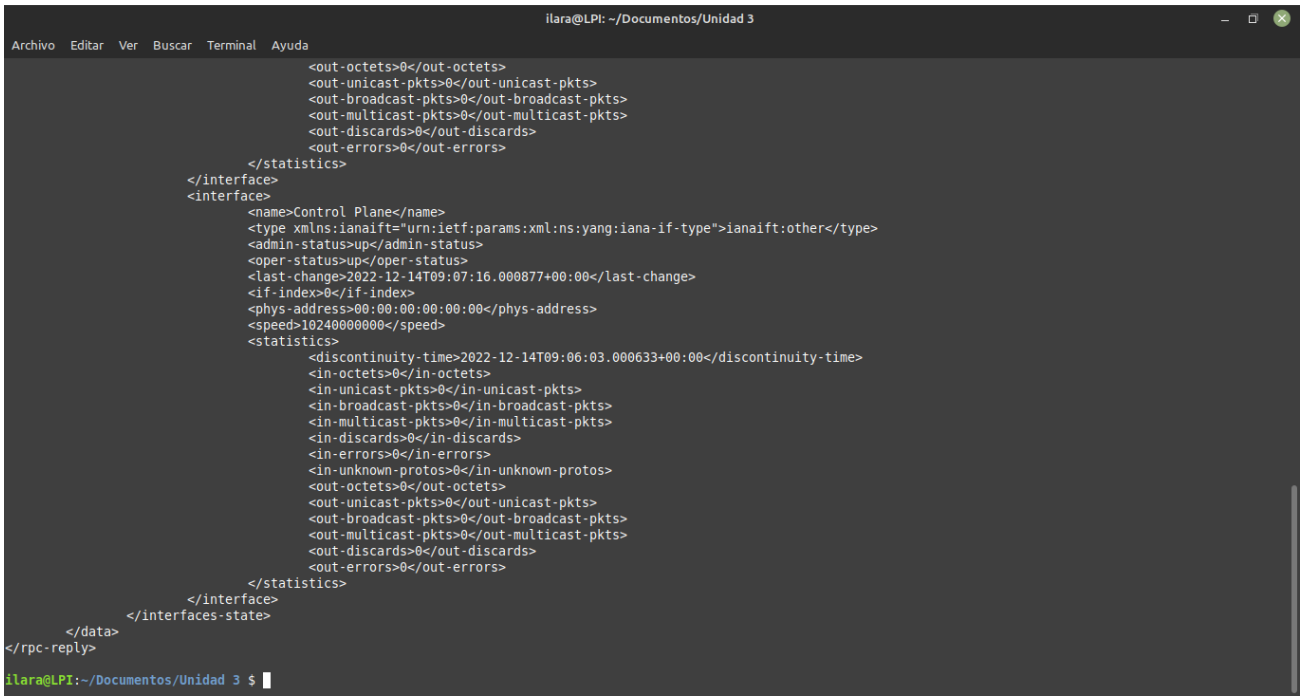
   Note: Executing the get() function without a filter is similar to execute "debug all".

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

netconf_reply = m.get(filter = netconf_filter)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```



e. Execute the Python script and explore the output.

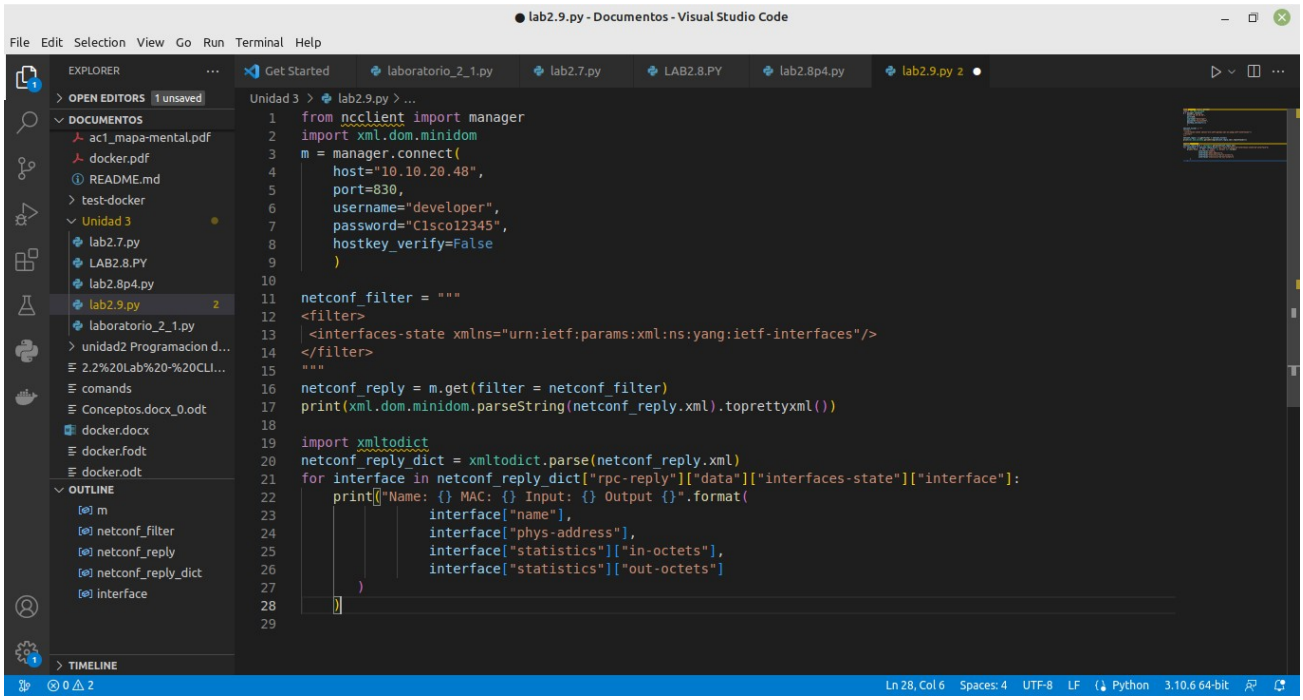




Esta es la salida que nos arroja el script

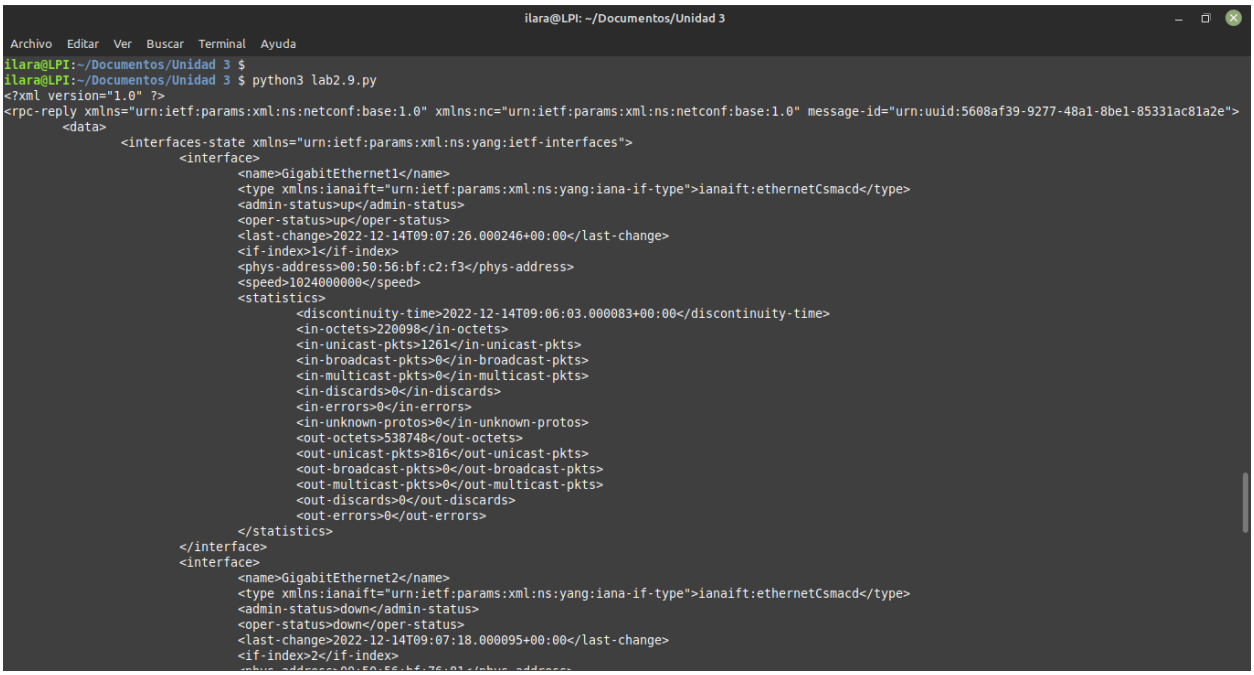
f. Convert the XML netconf_reply data to a Python dictionary using the “xmltodict” module. You can use a simple **for** loop to print a summary view of the statistical data:

```
import xmltodict

netconf_reply_dict = xmltodict.parse(netconf_reply.xml)

for interface in netconf_reply_dict["rpc-reply"]["data"]["interfaces-state"]["interface"]:
    print("Name: {} MAC: {} Input: {} Output {}".format(
                interface["name"],
                interface["phys-address"],
                interface["statistics"]["in-octets"],
                interface["statistics"]["out-octets"]
        )
    )
```




```
agregamos lo siguente para que la salida nos la muestre en forma de diccionario la informacion del
    dispositivo
```

g. Execute the script and explore the output.



En este laboratorio se utilizó ncclient para conectarse y revisar la configuración que se estaba ejecutando en el router al que erstamos conectados remotamente , mostrando una respuesta en XML con una fácil lectura para nosotros los humano, enviando comandos para revisar la configuración actual el router . Utilizando xmltodict que nos permite tratar archivos XML de una manera muy fácil, ya que convierte los archivos XML en una estructura de datos de tipo diccionario. Pero es tos pasos que realizamos es aquellos que venias observando durante los anteriores laboratorios. Estos modulos que importamos vienen co la libreria de Yang que es una modulo para administrar dispositivos conectados correctamente y ncclient sirve alo que entendi para recuperar la configuración en ejecución del dispositivo al que estamos conectados.