

## Lab – NETCONF w/Python: Get Operational Data

Use ncclient to retrieve the device's running configuration.

- Execute the script and explore the output.

### SCRIPT

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
lab2.9.py - D:/UTNG/UNIVERSIDAD/Cuarto cuatrimestre/Programación de redes/Unidad III/...
File Edit Format Run Options Window Help
#Canchola Ramírez Mariana
#LABORATORIO 2.9
from ncclient import manager
import xml.dom.minidom

m = manager.connect(
    host="10.10.20.48",
    port=830,
    username="developer",
    password="Cisc012345",
    hostkey_verify=False
)

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())

import xmldict
netconf_reply_dict = xmldict.parse(netconf_reply.xml)
for interface in netconf_reply_dict["rpc-reply"]["data"]["interfaces-state"]["in-
    print("Name: {} MAC: {} Input: {} Output {}".format(
        interface["name"],
        interface["phys-address"],
        interface["statistics"]["in-octets"],
        interface["statistics"]["out-octets"]
    )
)
|
ns:yang:iana-if-type">ianaift
t-pkts>
t-pkts>
-interfaces">
ns:yang:iana-if-type">ianaift
0:00</last-change>
ress>
:27:49.000887+00:00</disconti
t-pkts>
t-pkts>
t-pkts>
protos>
st-pkts>
ast-pkts>
ast-pkts>
ns:yang:iana-if-type">ianaift
```

### Resultado

```
IDLE Shell 3.10.9
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/UTNG/UNIVERSIDAD/Cuarto cuatrimestre/Programación de redes/Unidad III/Scripts/lab2.9.py
<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:528bb889-a5f5-4cd7-b0e8-87ee591808d1">
  <data>
    <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>GigabitEthernet1</name>
        <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:ethernetCsmacd</type>
        <admin-status>up</admin-status>
        <oper-status>up</oper-status>
        <last-change>2022-12-08T23:29:12.000453+00:00</last-change>
        <if-index>1</if-index>
        <phys-address>00:50:56:b3:bf:96</phys-address>
        <speed>1024000000</speed>
        <statistics>
          <discontinuity-time>2022-12-08T23:27:49.000887+00:00</discontinuity-time>
          <in-octets>127246</in-octets>
          <in-unicast-pkts>1070</in-unicast-pkts>
          <in-broadcast-pkts>0</in-broadcast-pkts>
          <in-multicast-pkts>0</in-multicast-pkts>
          <in-discards>0</in-discards>
          <in-errors>0</in-errors>
          <in-unknown-protos>0</in-unknown-protos>
          <out-octets>996182</out-octets>
          <out-unicast-pkts>1584</out-unicast-pkts>
          <out-broadcast-pkts>0</out-broadcast-pkts>
          <out-multicast-pkts>0</out-multicast-pkts>
          <out-discards>0</out-discards>
          <out-errors>0</out-errors>
        </statistics>
      </interface>
    </interfaces-state>
  </data>
</rpc-reply>
<?xml version="1.0" ?>
<type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:ethernetCsmacd</type>
<admin-status>down</admin-status>
<oper-status>down</oper-status>
```

## Conclusiones

Este laboratorio imprime, envía configuración y filtra datos, por lo que hace mucho más con este protocolo, por ello esta es una de las ventajas que veo en esta biblioteca,

Ya que se pueden imprimir algunos datos específicos, mientras que no lo hacemos con netmiko. o lo probamos y sentimos que este módulo es muy completo, en la parte de configuración y programación, además del uso de NETCONF hay más explicaciones al desarrollar este módulo.

### NETCONF Protocol Stack

