

## 2.9 Lab - NETCONF wPython Get Operational Data

En este laboratorio, aprenderá a usar NETCONF para recuperar datos operativos del dispositivo de red.

Utilice ncclient para recuperar la configuración en ejecución del dispositivo.

```
*LAB 2.9.py - D:\Laboratorios\LAB 2.9.py (3.10.9)*
File Edit Format Run Options Window Help

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 xmltodict
netconf_reply_dict = xmltodict.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"]
    )
)
```

```
IDLE Shell 3.10.9
File Edit Shell Debug Options Window Help

Python 3.10.9 (tags/v3.10.9:1dd9bee, 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:\Laboratorios\LAB 2.9.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:283cc878-eb4d-4f9d-b349-9cbfa91e668f">
  <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">ianaif:ethernetCsmacd</type>
        <admin-status>up</admin-status>
        <oper-status>up</oper-status>
        <last-change>2022-12-11T21:29:33.000478+00:00</last-change>
        <if-index>1</if-index>
        <phys-address>00:50:56:bfe3:4d</phys-address>
        <speed>1024000000</speed>
        <statistics>
          <discontinuity-time>2022-12-11T21:27:57.000454+00:00</discontinuity-time>
          <in-octets>497315</in-octets>
          <in-unicast-pkts>7446</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>1104520</out-octets>
          <out-unicast-pkts>7621</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>
      <interface>
        <name>GigabitEthernet2</name>
        <type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:ethernetCsmacd</type>
        <admin-status>down</admin-status>
        <oper-status>down</oper-status>
        <last-change>2022-12-11T21:29:23.000629+00:00</last-change>
      </interface>
    </interfaces-state>
  </data>
</rpc-reply>
>>>
```

Con este laboratorio vimos nuevamente el funcionamiento de NETCONF así mostrándonos una librería muy extensa en la cual podemos encontrar información que necesitamos.

De esta manera también podemos hacer que nos muestre información en específico para que no nos muestre todo, así logrando reducir la información y obteniendo la que necesitamos