



Universidad Tecnológica
del Norte de Guanajuato
Organismo Público Descentralizado del Gobierno del Estado

Facultad de: *Infraestructura de Redes Digitales*

Nombre del Alumno(a): *Ángel Armando Ramírez Vázquez*

Matrícula: *1221100627*

Materia:

Programación de Redes

Nombre de la Actividad:

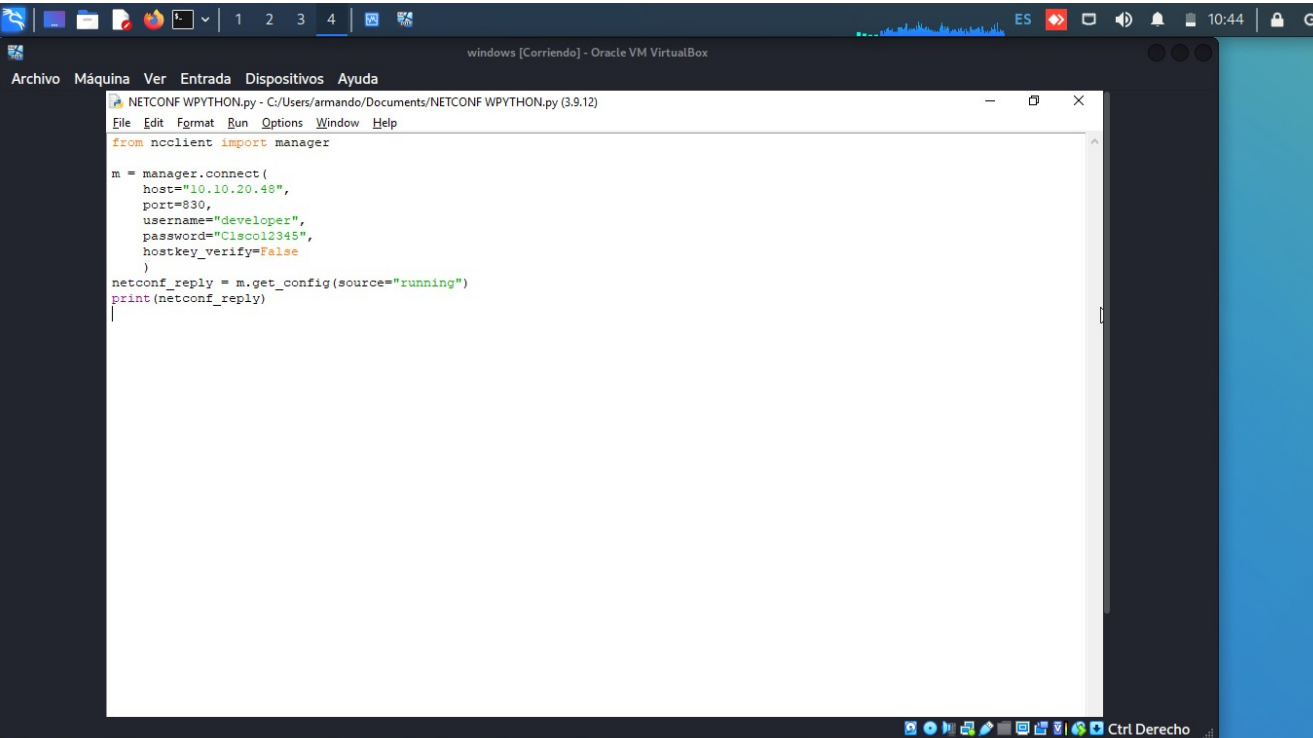
2.8 Lab - NETCONF wPython Device Configuration

Profesor:

Barron Rodríguez Gabriel

Lugar y Fecha de Presentación: *Dolores Hidalgo C.I.N.*; hoy *6* de *Diciembre* del
2022

Paso 1. Usa ncclient para obtener la configuración de los dispositivos.



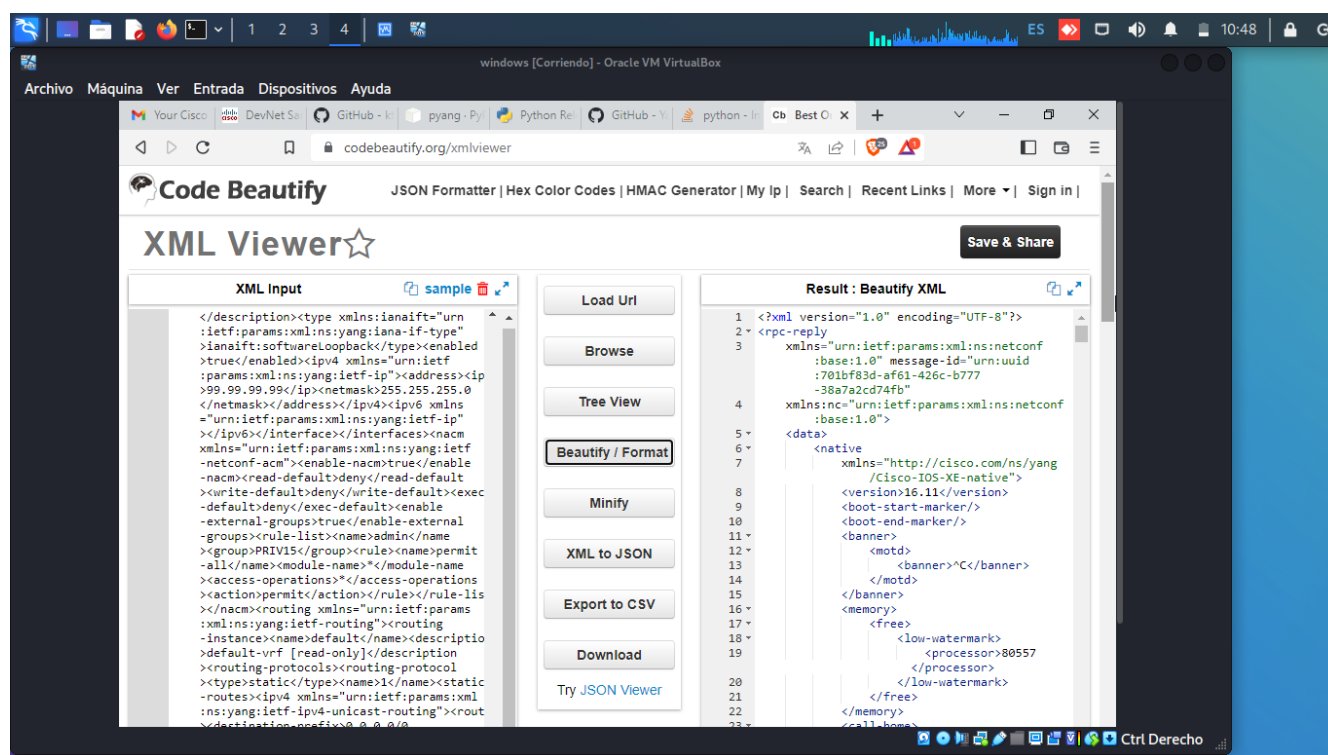
The screenshot shows a Windows Virtual Machine (VM) running on Oracle VM VirtualBox. The VM is named "windows [Corriendo]". The desktop environment is visible, showing a taskbar with various icons and a system tray. A Notepad++ window is open, displaying a Python script named "NETCONF WPYTHON.py" (3.9.12). The script imports the "manager" module from "ncclient" and connects to a host at "10.10.20.48" on port "830" with the username "developer" and password "Cisc0012345". The script then calls "m.get_config(source='running')" and prints the result.

```
NETCONF WPYTHON.py - C:/Users/armando/Documents/NETCONF WPYTHON.py (3.9.12)
File Edit Format Run Options Window Help
from ncclient import manager

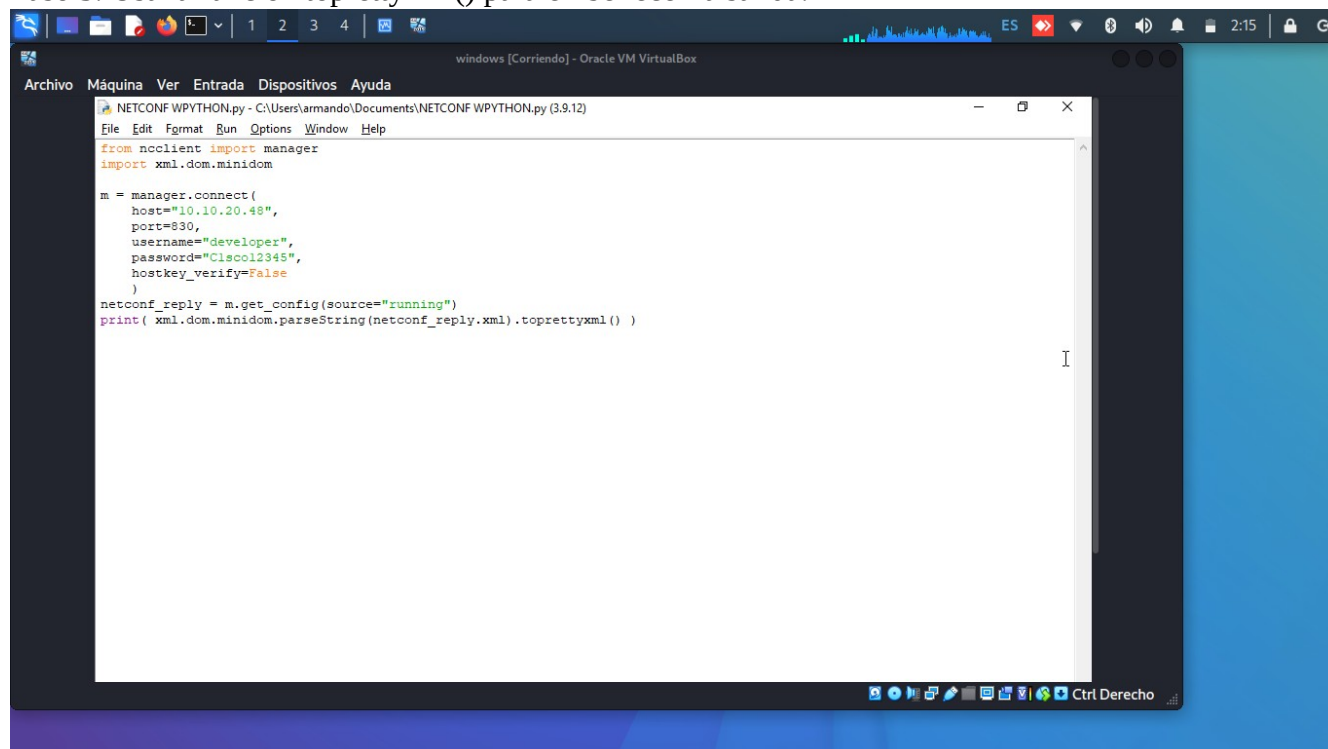
m = manager.connect(
    host="10.10.20.48",
    port=830,
    username="developer",
    password="Cisc0012345",
    hostkey_verify=False
)

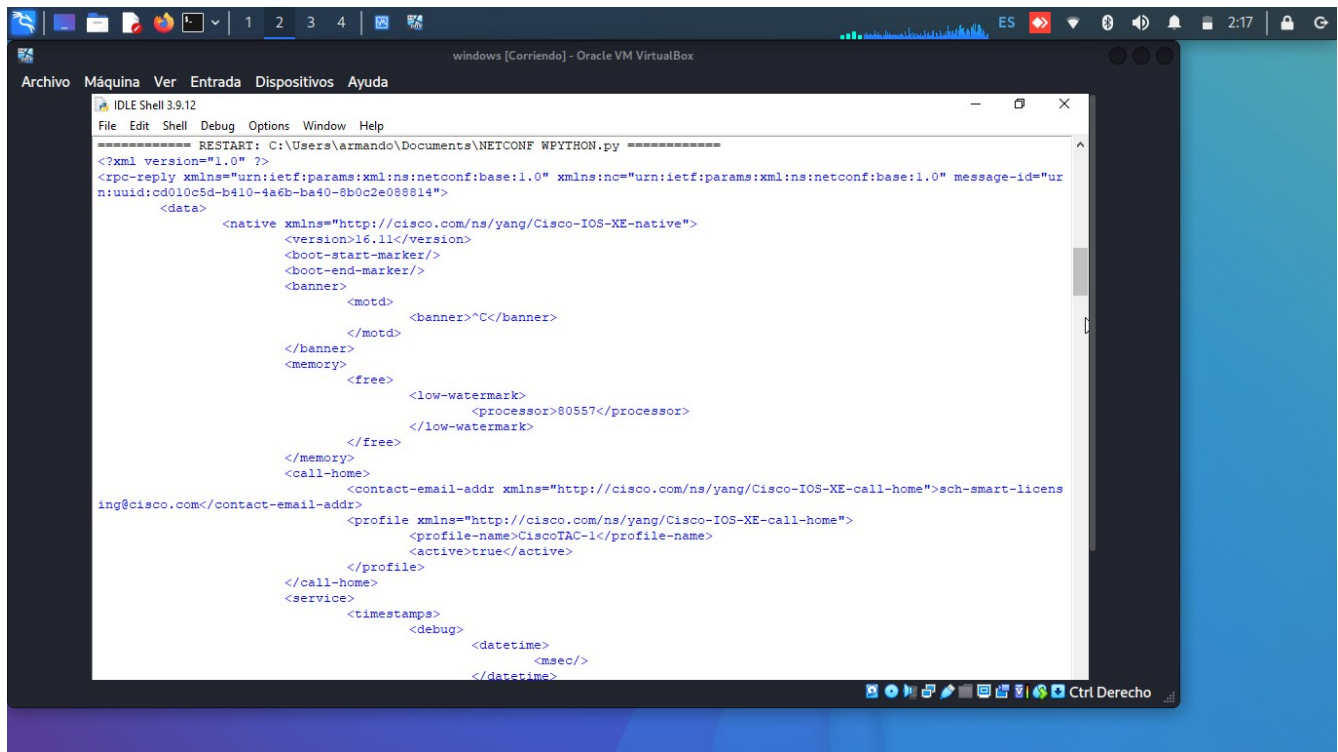
netconf_reply = m.get_config(source="running")
print(netconf_reply)
```

Paso 2. Usa CodeBeauty.com para evaluar la respuesta.



Paso 3. Usa la función `toprettyxml()` para embellecer la salida.

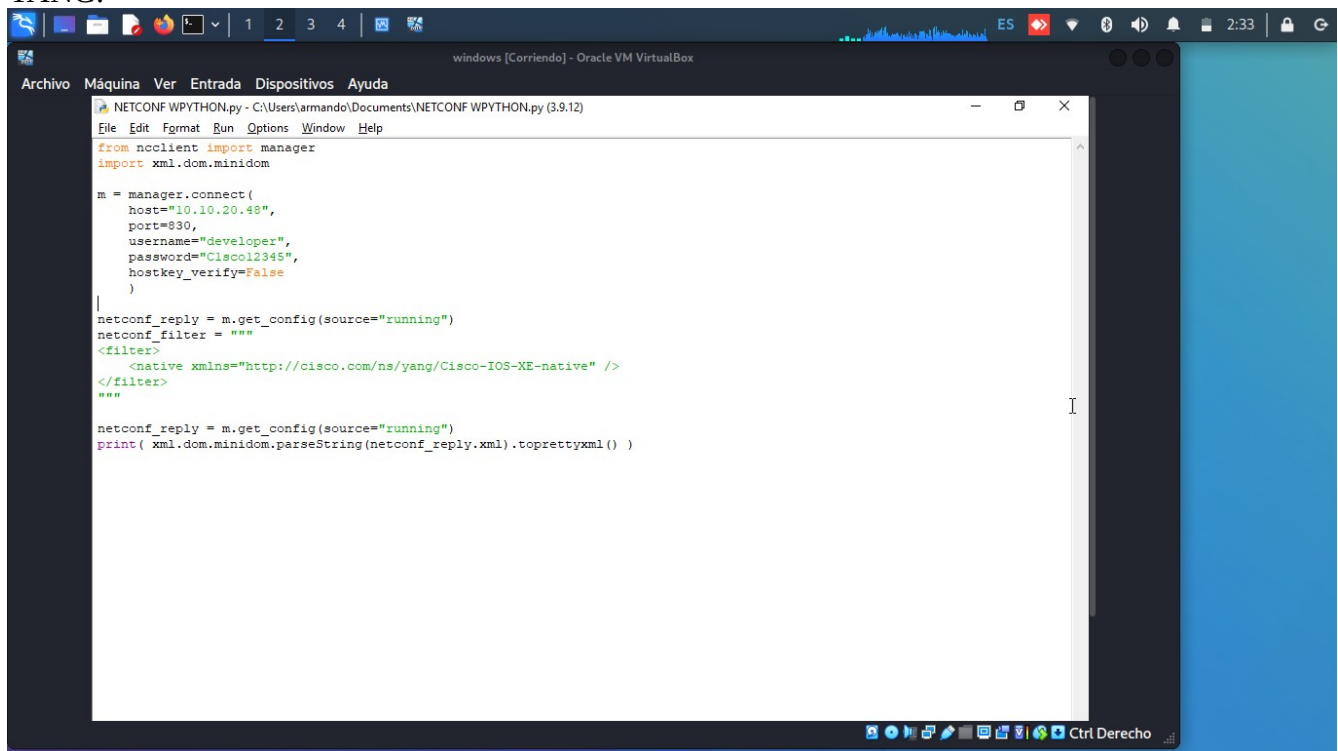




The screenshot shows a Windows Virtual Machine window titled "windows [Corriendo] - Oracle VM VirtualBox". Inside the VM, an IDLE Shell window is open, displaying the output of a RESTCONF request. The output is an XML document representing a Cisco IOS-XE configuration. The XML includes a version, boot markers, a banner, memory configuration, a call-home profile, and service settings. The IDLE Shell window has a menu bar with File, Edit, Shell, Debug, Options, Window, and Help. The status bar at the bottom of the VM window shows system icons and the text "Ctrl Derecho".

```
===== RESTART: C:\Users\armando\Documents\NETCONF WPYTHON.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:cd010c5d-b410-4a6b-ba40-8b0c2e088814">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <version>16.11</version>
      <boot-start-marker/>
      <boot-end-marker/>
      <banner>
        <motd>
          <banner>^C</banner>
        </motd>
      </banner>
      <memory>
        <free>
          <low-watermark>
            <processor>80557</processor>
          </low-watermark>
        </free>
      </memory>
      <call-home>
        <contact-email-addr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">sch-smart-licens
ing@cisco.com</contact-email-addr>
        <profile xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">
          <profile-name>CiscoTAC-1</profile-name>
          <active>true</active>
        </profile>
      </call-home>
      <service>
        <timestamps>
          <debug>
            <datetime>
              <msec/>
            </datetime>
          </debug>
        </timestamps>
      </service>
    </native>
  </data>
</rpc-reply>
```

Paso 4. Usa filtros para devolver la configuración definida como un modelo específico de modelo YANG.



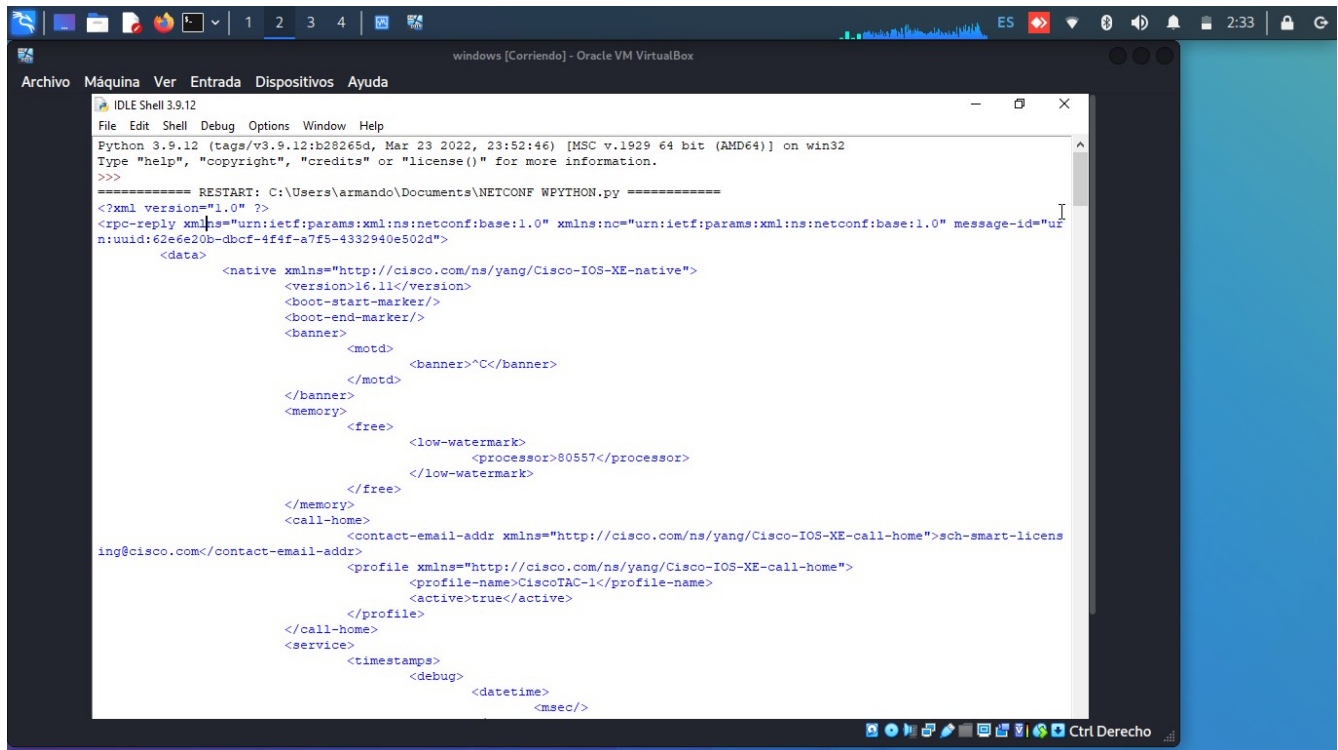
The screenshot shows the same Windows Virtual Machine window. The IDLE Shell window now displays a Python script that uses the netconfclient library to connect to a RESTCONF server and retrieve configuration data. The script defines a filter to select only the native configuration model and then prints the resulting XML. The Python code is as follows:

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

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

netconf_reply = m.get_config(source="running")
netconf_filter = """
<filter>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native" />
</filter>
"""

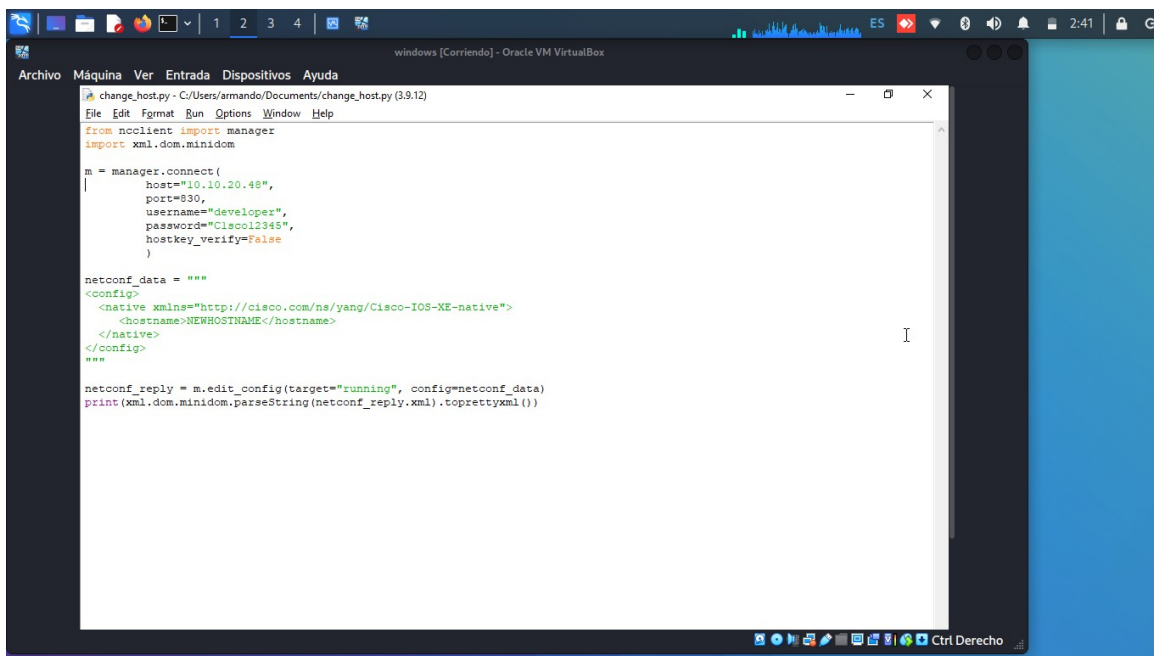
netconf_reply = m.get_config(source="running")
print( xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml() )
```



```
File Edit Shell Debug Options Window Help
Python 3.9.12 (tags/v3.9.12:b28265d, Mar 23 2022, 23:52:46) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\armando\Documents\NETCONF WPYTHON.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:62e6e20b-dbcf-4f4f-a7f5-4332940e502d">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <version>16.11</version>
      <boot-start-marker/>
      <boot-end-marker/>
      <banner>
        <motd>
          <banner>^C</banner>
        </motd>
      </banner>
      <memory>
        <free>
          <low-watermark>
            <processor>80557</processor>
          </low-watermark>
        </free>
      </memory>
      <call-home>
        <contact-email-addr xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">sch-smart-licens
ing@cisco.com</contact-email-addr>
        <profile xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-call-home">
          <profile-name>CiscoTAC-1</profile-name>
          <active>true</active>
        </profile>
      </call-home>
      <service>
        <timestamps>
          <debug>
            <datetime>
              <msec/>
            </datetime>
          </debug>
        </timestamps>
      </service>
    </native>
  </data>
</rpc-reply>
```

Parte 2. Actualiza la configuración del dispositivos.

Paso 1. Crea un nuevo script de python y configuralo para cambiar el hostname.

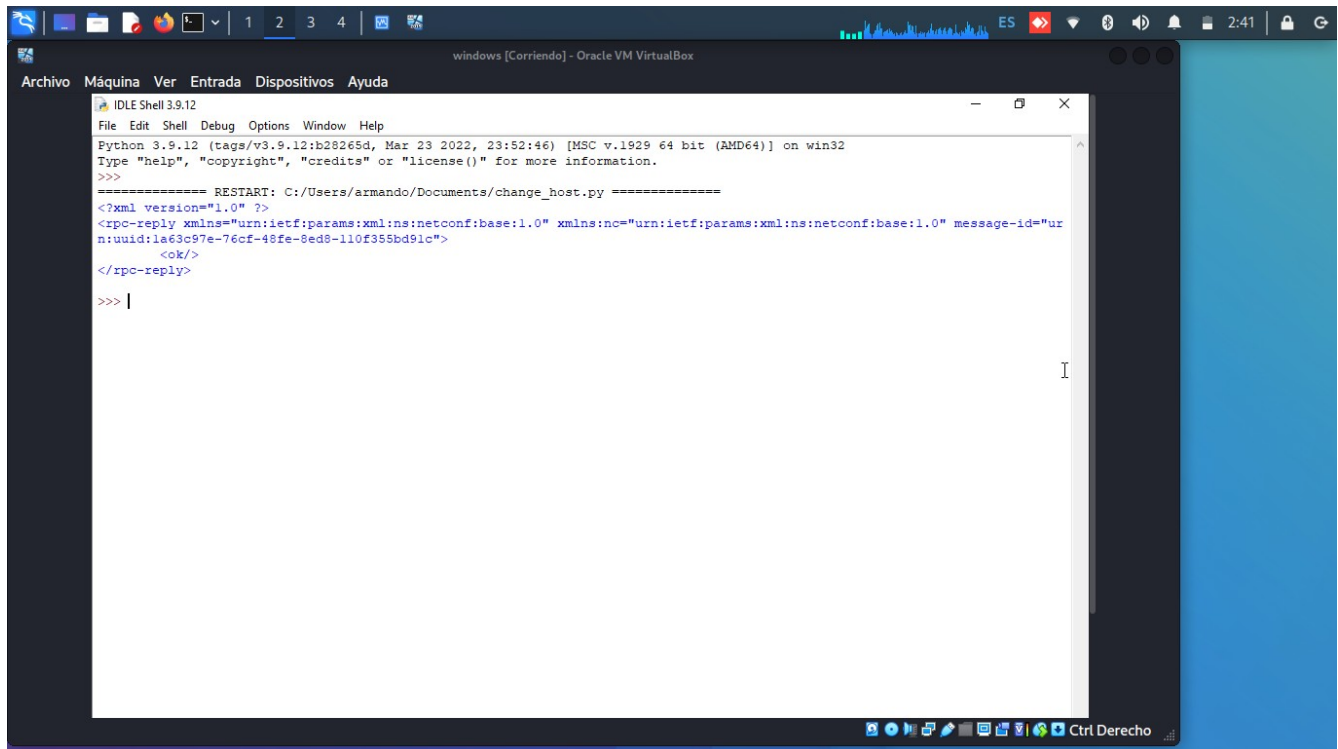


```
File Edit Format Run Options Window Help
change_host.py - C:\Users\armando\Documents\change_host.py (3.9.12)
from ncclient import manager
import xml.dom.minidom

m = manager.connect(
    host="10.10.20.48",
    port=830,
    username="developer",
    password="Cisco12345",
    hostkey_verify=False
)

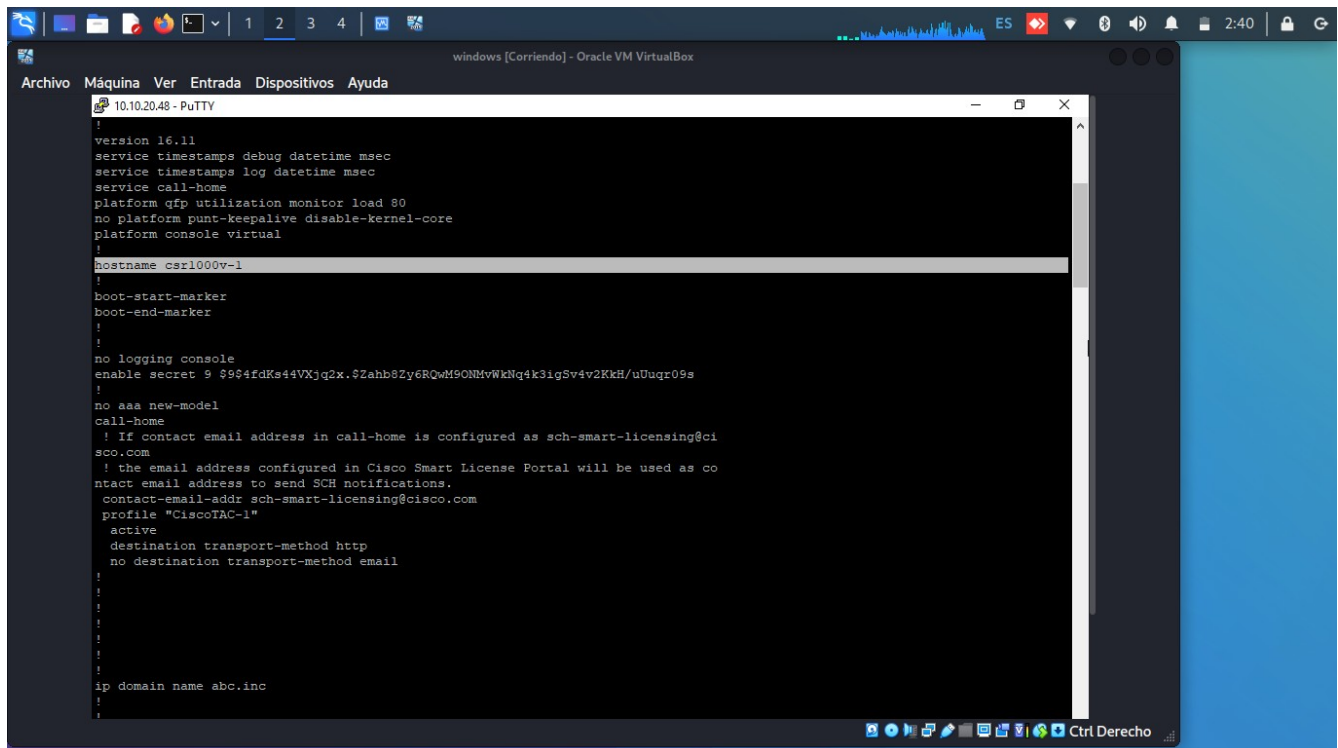
netconf_data = """
<config>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <hostname>NEWHOSTNAME</hostname>
  </native>
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_data)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

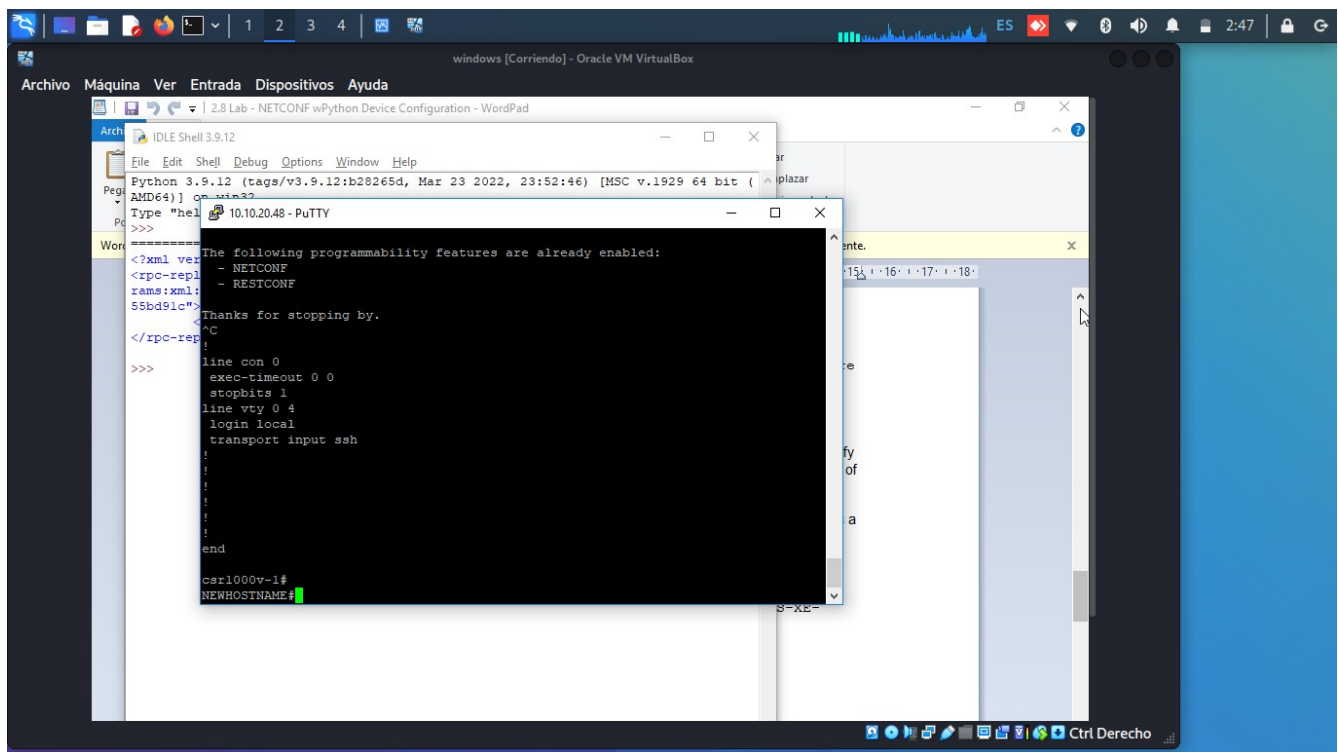


```
File Edit Shell Debug Options Window Help
Python 3.9.12 (tags/v3.9.12:b28265d, Mar 23 2022, 23:52:46) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/armando/Documents/change_host.py =====
<?xml version="1.0" ?>
<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:1a63c97e-76cf-48fe-8ed8-110f355bd91c">
  <ok/>
</rpc-reply>
>>> |
```

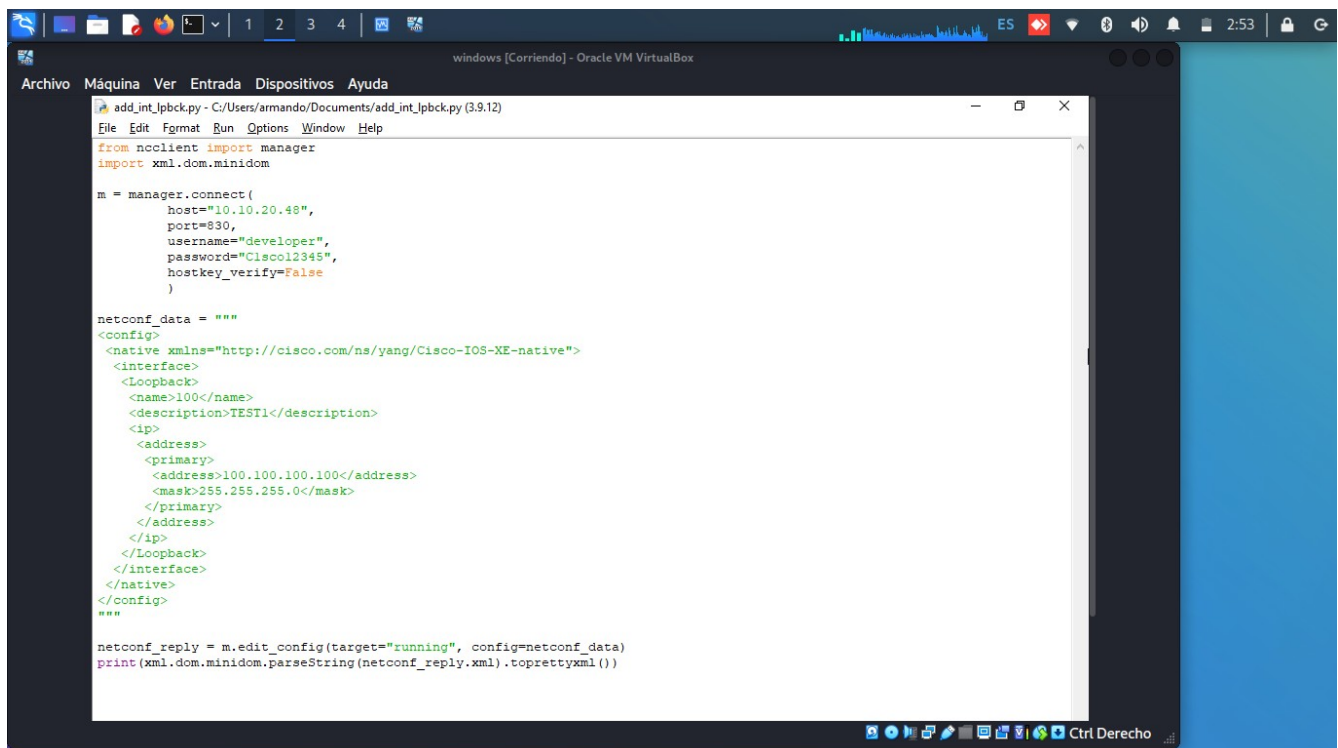
Verificamos:

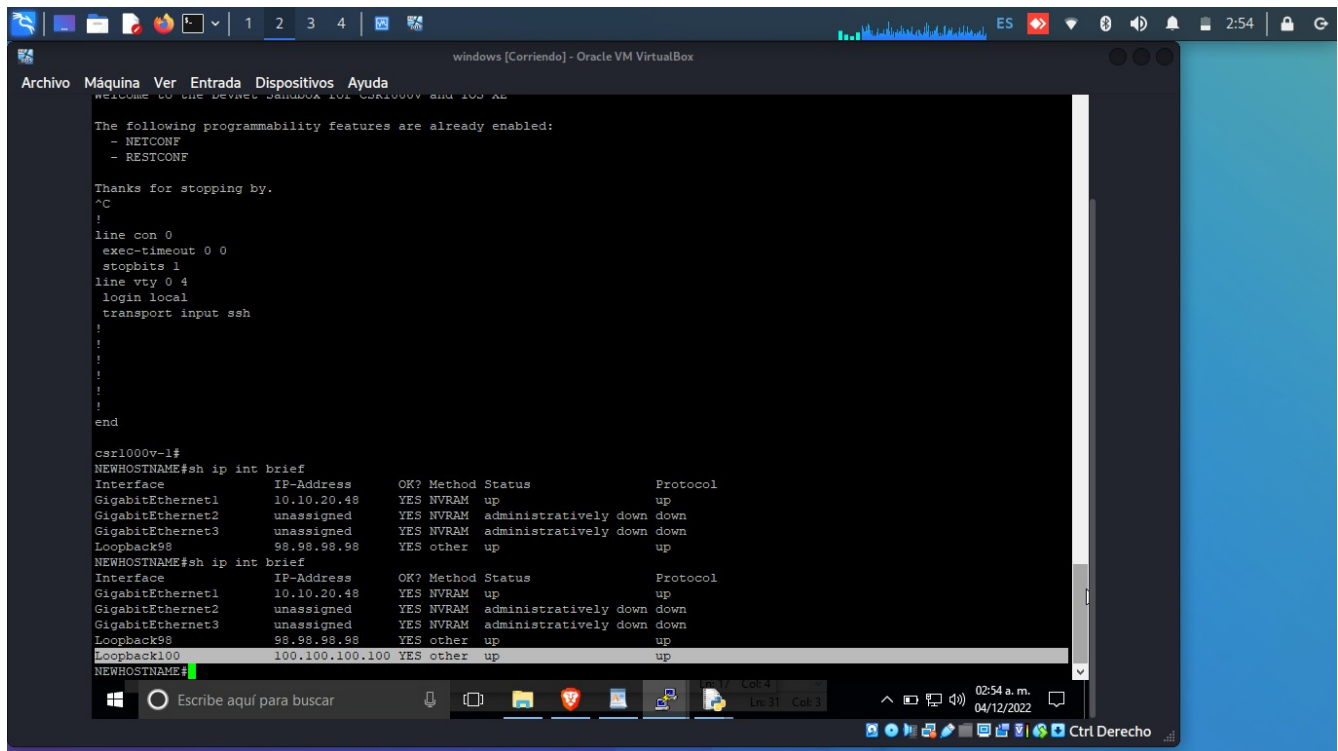


```
10.10.20.48 - PuTTY
!
version 16.11
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname csr1000v-1
!
boot-start-marker
boot-end-marker
!
!
no logging console
enable secret 9 $9$4fdKs44VXjq2x.$2ahb82y6RQwM9ONMvWkNq4k3igSw4v2KkH/uUuqr09s
!
no aaa new-model
call-home
! If contact email address in call-home is configured as sch-smart-licensing@ci
sco.com
! the email address configured in Cisco Smart License Portal will be used as co
ntract email address to send SCH notifications.
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
active
destination transport-method http
no destination transport-method email
!
!
!
!
!
ip domain name abc.inc
!
```

Paso 2. Crearemos una interfaz loopback.





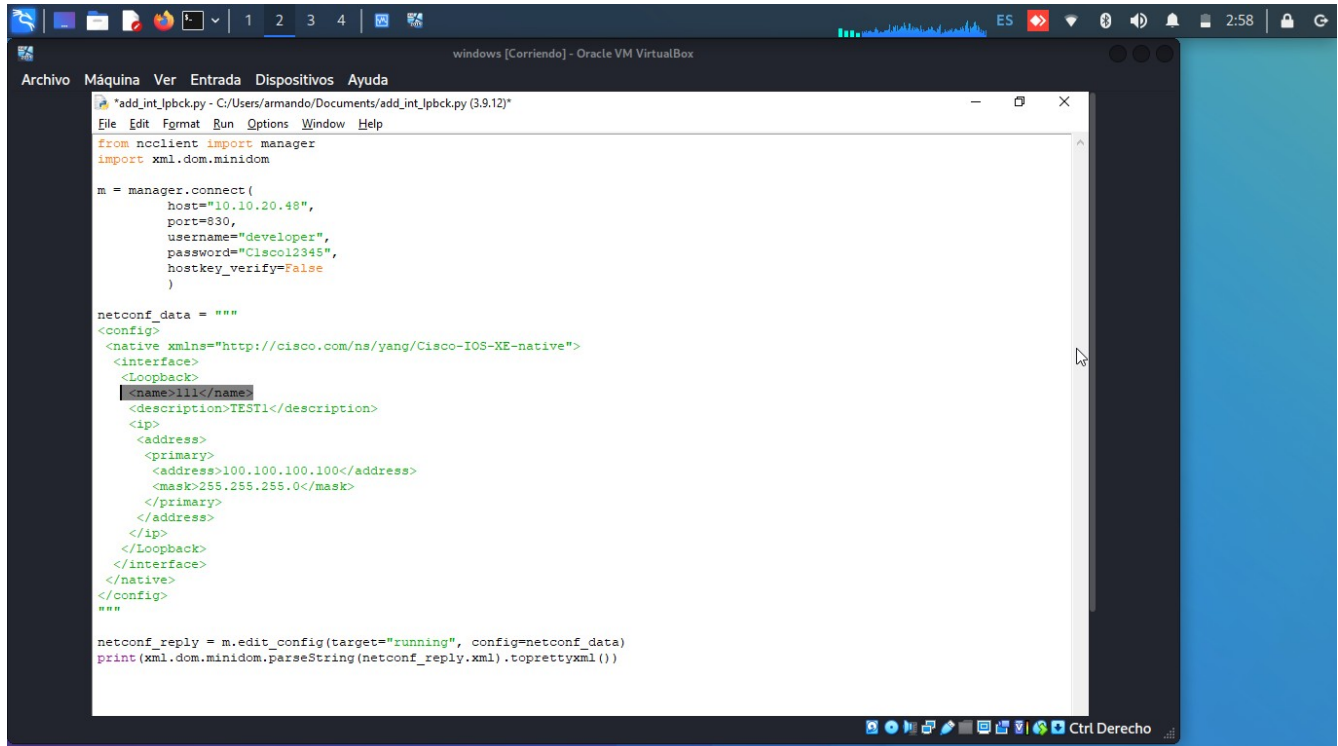
```
welcome to the Devnet Sandbox 101 CSR1000V and 100 AE

The following programmability features are already enabled:
- NETCONF
- RESTCONF

Thanks for stopping by.
^C
!
line con 0
exec-timeout 0 0
stopbits 1
line vty 0 4
login local
transport input ssh
!
!
!
!
!
!
end

csr1000v-1#
NEWHOSTNAME#sh ip int brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet1 10.10.20.48 YES NVRAM up up
GigabitEthernet2 unassigned YES NVRAM administratively down down
GigabitEthernet3 unassigned YES NVRAM administratively down down
Loopback98 98.98.98.98 YES other up up
NEWHOSTNAME#sh ip int brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet1 10.10.20.48 YES NVRAM up up
GigabitEthernet2 unassigned YES NVRAM administratively down down
GigabitEthernet3 unassigned YES NVRAM administratively down down
Loopback98 98.98.98.98 YES other up up
Loopback100 100.100.100.100 YES other up up
NEWHOSTNAME#
```

Paso 4: Crea una nueva interfaz loopback con conflicto de dirección IP.



```
"add_int_ipbck.py - C:/Users/armando/Documents/add_int_ipbck.py (3.9.12)"
File Edit Format Run Options Window Help

from netclient import manager
import xml.dom.minidom

m = manager.connect(
    host="10.10.20.48",
    port=830,
    username="developer",
    password="Cisc0ol2345",
    hostkey_verify=False
)

netconf_data = """
<config>
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
<interface>
<loopback>
<name>l11</name>
<description>TEST1</description>
<ip>
<address>
<primary>
<address>100.100.100.100</address>
<mask>255.255.255.0</mask>
</primary>
</address>
</ip>
</loopback>
</interface>
</native>
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_data)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

Windows [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

IDLE Shell 3.9.12

File Edit Shell Debug Options Window Help

Python 3.9.12 (tags/v3.9.12:b28265d, Mar 23 2022, 23:52:46) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

```
>>>
===== RESTART: C:/Users/armando/Documents/change_host.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:1a63c97e-76cf-48fe-8ed8-110f355bd91c">
  <ok/>
</rpc-reply>

>>>
===== RESTART: C:/Users/armando/Documents/add_int_lpbck.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:28bfalda-leel-4f5b-bb93-2af2087ce02c">
  <ok/>
</rpc-reply>

>>>
===== RESTART: C:/Users/armando/Documents/add_int_lpbck.py =====
Traceback (most recent call last):
  File "C:/Users/armando/Documents/add_int_lpbck.py", line 33, in <module>
    netconf_reply = m.edit_config(target="Running", config=netconf_data)
  File "C:/Users/armando/AppData/Local/Programs/Python/Python39/lib/site-packages/ncclient\manager.py", line 246, in execute
    return cls(self._session,
  File "C:/Users/armando/AppData/Local/Programs/Python/Python39/lib/site-packages/ncclient\operations\edit.py", line 76, in request
    return self._request(node)
ncclient.operations.rpc.RPCError: inconsistent value: Device refused one or more commands

>>>
```

Ctrl Derecho