

Lab – raw NETCONF

Objectives

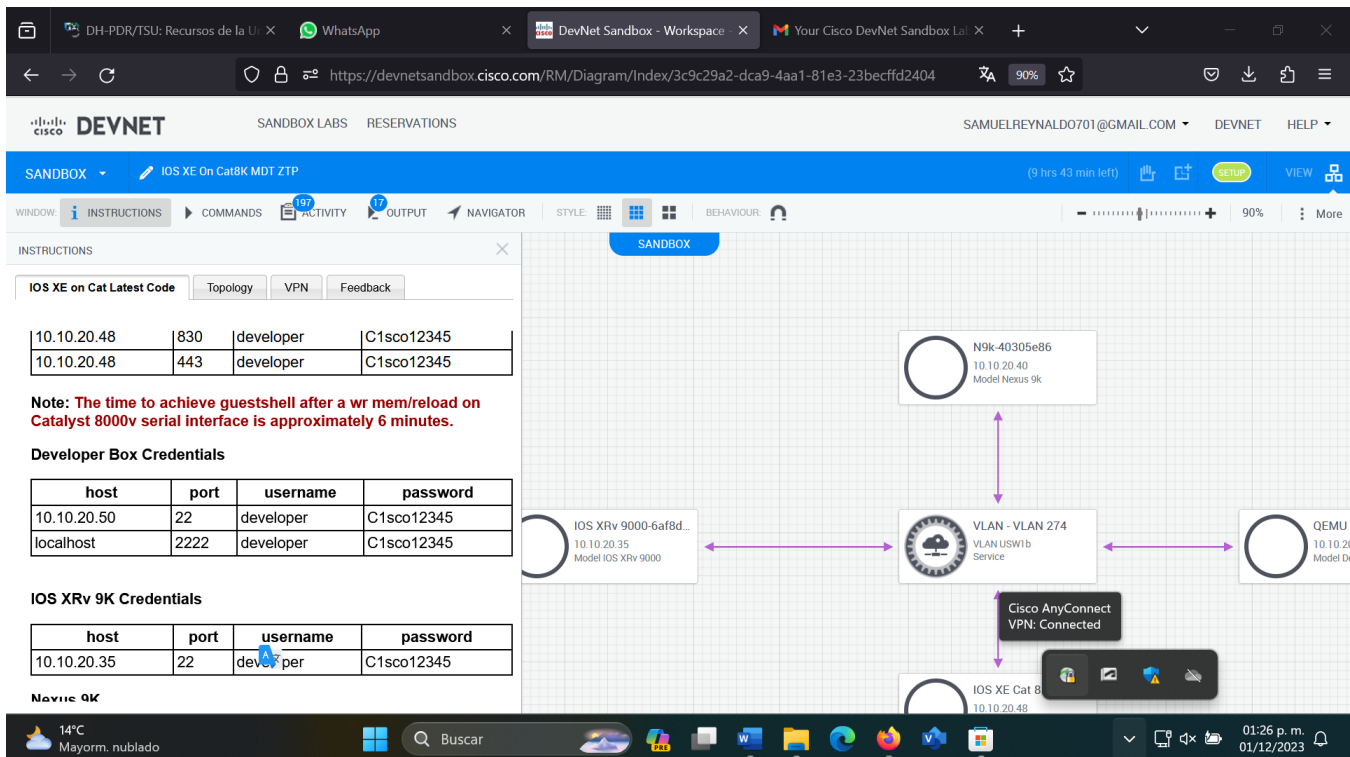
Part 1: Verify that NETCONF is Running on the IOS XE

Background / Scenario

In this lab, you will learn how to verify that the NETCONF service is running on the device by directly connecting to its port using an SSH client. You will be sending raw NETCONF Remote Procedure Calls encoded in XML structures.

Required Resources

- Access to a router with the IOS XE operating system version 16.6 or higher
- Putty



The screenshot shows the Cisco DevNet Sandbox interface. On the left, there are instructions and tables for configuring the lab. The main area displays a network diagram with various devices connected.

Instructions:

IOS XE on Cat Latest Code

Host	Port	Username	Password
10.10.20.48	830	developer	C1sco12345
10.10.20.48	443	developer	C1sco12345

Note: The time to achieve guestshell after a wr mem/reload on Catalyst 8000v serial interface is approximately 6 minutes.

Developer Box Credentials

host	port	username	password
10.10.20.50	22	developer	C1sco12345
localhost	2222	developer	C1sco12345

IOS XRv 9K Credentials

host	port	username	password
10.10.20.35	22	developer	C1sco12345

Network Diagram:

- IOS XRv 9000-6af8d... (10.10.20.35, Model IOS XRv 9000)
- N9k-40305e86 (10.10.20.40, Model Nexus 9k)
- VLAN - VLAN 274 (VLAN USW1b Service)
- QEMU (10.10.20.21, Model D)
- IOS XE Cat 8... (10.10.20.48)

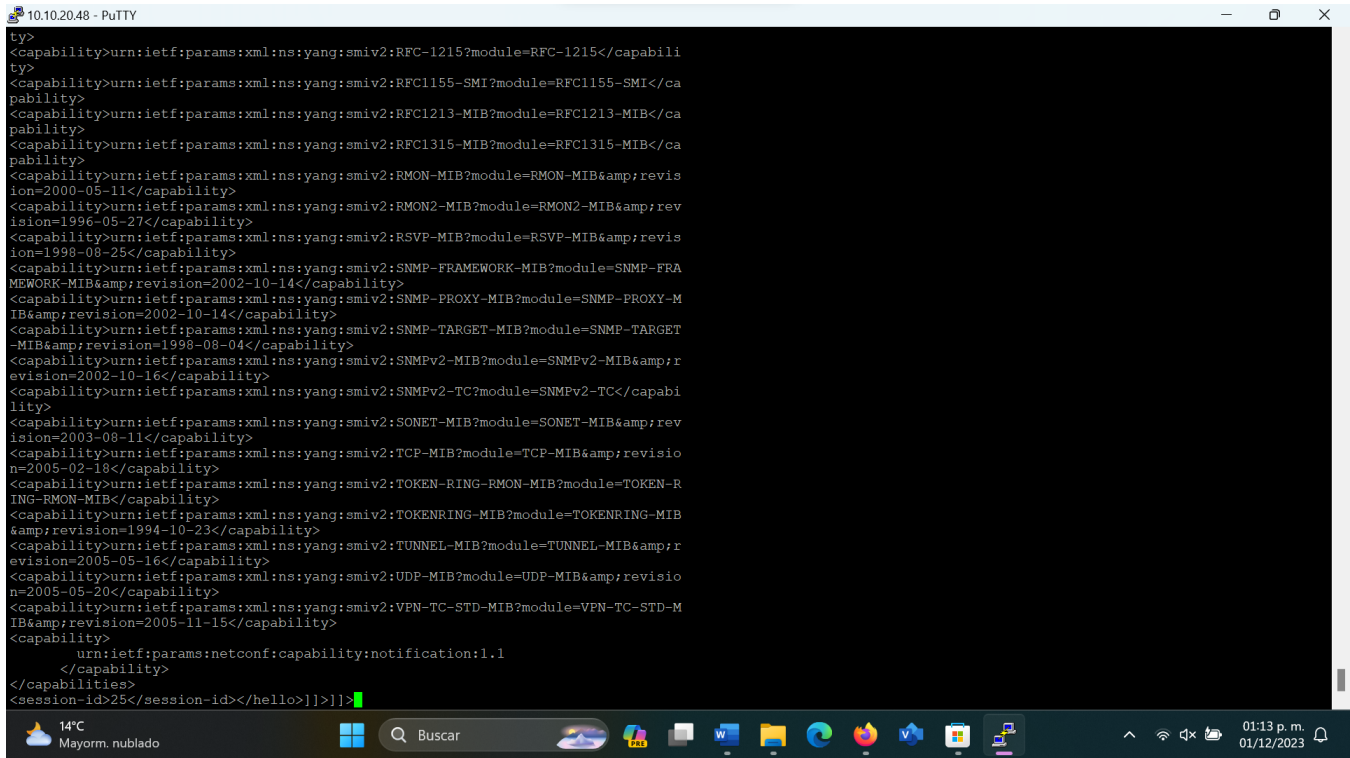
Connections are shown between these devices. A status bar at the bottom indicates "Cisco AnyConnect VPN: Connected".

Part 1: Verify that NETCONF is Running on the IOS XE

Step 1: Use Putty as an SSH client to connect to the NETCONF service.

- Start Putty.
- Using Putty, connect to host "192.168.56.101" (Adjust the IP address to match the router's current address.) and port "830".
- Login as "cisco" with the password "cisco123!" that was configured in IOS XE VM.

- d. After a successful login to the NETCONF server, you should see a server “hello” message with an XML formatted list of supported YANG models (capabilities).
- e. The end of the message is identified with “]]>]]>”.



```

10.10.20.48 - PuTTY
ty>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RFC-1215?module=RFC-1215</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RFC1155-SMI?module=RFC1155-SMI</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RFC1213-MIB?module=RFC1213-MIB</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RFC1315-MIB?module=RFC1315-MIB</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RMON-MIB?module=RMON-MIB&revision=2000-05-11</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RMON2-MIB?module=RMON2-MIB&revision=1996-05-27</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:RSVP-MIB?module=RSVP-MIB&revision=1998-08-25</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SNMP-FRAMEWORK-MIB?module=SNMP-FRAMEWORK-MIB&revision=2002-10-14</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SNMP-PROXY-MIB?module=SNMP-PROXY-MIB&revision=2002-10-14</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SNMP-TARGET-MIB?module=SNMP-TARGET-MIB&revision=1998-08-04</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-MIB?module=SNMPv2-MIB&revision=2002-10-16</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-TC?module=SNMPv2-TC</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:SONET-MIB?module=SONET-MIB&revision=2003-08-11</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:TCP-MIB?module=TCP-MIB&revision=2005-02-18</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:TOKEN-RING-RMON-MIB?module=TOKEN-RING-RMON-MIB</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:TOKENRING-MIB?module=TOKENRING-MIB&revision=1994-10-23</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:TUNNEL-MIB?module=TUNNEL-MIB&revision=2005-05-16</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:UDP-MIB?module=UDP-MIB&revision=2005-05-20</capabilities>
<capabilities>urn:ietf:params:xml:ns:yang:smiv2:VPN-TC-STD-MIB?module=VPN-TC-STD-MIB&revision=2005-11-15</capabilities>
<capabilities>urn:ietf:params:netconf:capability:notification:1.1</capabilities>
<session-id>25</session-id></hello>]]>]]>

```

- f. To start a NETCONF session, the client needs to send its own hello message in a response:

```

<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>
]]>]]>

```

```

10.10.20.48 - PuTTY
<session=2003-08-11</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:TCP-MIB?module=TCP-MIB&revision=2005-02-18</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:TOKEN-RING-FMON-MIB?module=TOKEN-RING-FMON-MIB</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:TOKENRING-MIB?module=TOKENRING-MIB&revision=1994-10-23</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:TUNNEL-MIB?module=TUNNEL-MIB&revision=2005-05-16</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:UDP-MIB?module=UDP-MIB&revision=2005-05-20</capability>
<capability>urn:ietf:params:xml:ns:yang:smiv2:VPN-TC-STD-MIB?module=VPN-TC-STD-MIB&revision=2005-11-15</capability>
<capability>
  urn:ietf:params:netconf:capability:notification:1.1
</capability>
</capabilities>
<session-id>29</session-id></hello>]]>]]><hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>
]]>]]>
<rpc message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
    </filter>
  </get>
</rpc>
]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="103"><data>
  <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"><interface><name>GigabitEthernet1</name><description>MANAGEMENT INTERFACE - DON'T TOUCH ME</description><type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:ethernetCsmacd</type><enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"><address><ip>10.10.20.48</ip><netmask>255.255.255.0</netmask></address></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface><interface><name>GigabitEthernet2</name><description>Network Interface</description><type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:ethernetCsmacd</type><enabled>false</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></in

```

- g. After the client hello message has been sent, the NETCONF session is ready to process RPC messages. For example, the following XML formatted RPC message will return the ietf-interfaces model data. Please note that the returned XML data are designed to be consumed by an application. By default, this data might be difficult to for humans to read.

```

<rpc message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
    </filter>
  </get>
</rpc>
]]>]]>

```

```

10.10.20.48 - PuTTY
</hello>
]]>]]>
<rpc message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
    </filter>
  </get>
</rpc>
]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="103"><data>
<interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"><interface><name>
GigabitEthernet1</name><description>MANAGEMENT INTERFACE - DON'T TOUCH ME</description>
<type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:ethernetCsmacd</type>
<enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"><address><ip>10.10.20.48</ip>
<netmask>255.255.255.0</netmask></address></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface>
<interface><name>GigabitEthernet2</name><description>Network Interface</description>
<type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:ethernetCsmacd</type>
<enabled>false</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface>
<interface><name>Loopback0</name><type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:softwareLoopback</type>
<enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"><address><ip>10.0.0.1</ip>
<netmask>255.255.255.0</netmask></address></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface>
<interface><name>Loopback10</name><type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:softwareLoopback</type>
<enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface>
<interface><name>Loopback109</name><description>Configured by RESTCONF ga jadi</description>
<type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:softwareLoopback</type>
<enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"><address><ip>10.255.255.9</ip>
<netmask>255.255.255.0</netmask></address></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface>
<interface><name>VirtualPortGroup0</name><type xmlns:ianaif="urn:ietf:params:xml:ns:yang:iana-if-type">ianaif:propVirtual</type>
<enabled>true</enabled><ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"><address><ip>192.168.1.1</ip>
<netmask>255.255.255.0</netmask></address></ipv4><ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv6></interface></interfaces></data></rpc-reply>]]>]]>

```

h. To close the NETCONF session, the client needs to send the following message:

```

<rpc message-id="9999999" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session />
</rpc>
]]>]]>

```

Aquí en este comando Cierra la pantalla del puty