Lab: Cisco IOS XR NETCONF API

Exercise: IOS XR NETCONF

In this Exercise you will configure an IOS XR router with NETCONF

Task 1: Enable crypto, SSH and NETCONF on the IOS XRV device.

In this task you will enable crypto, SSH and NETCONF. NETCONF will not work by default without SSH enabled, and SSH will not work without crypto enabled.

How-to Steps

1. Starting from your login page on Remotelabs.com, click on **Connect via Topology**.

2. The lab topology diagram will open. Notice how you can hover your mouse over the individual computers to see which you can connect into directly form the web portal.

NOTE_CALLOUT
In this Global Knowledge lab environment, you can open any of the virtual machines with a "hot link," but you can have only one VM open at any one time. Opening a second VM will close the first connection you were in, without losing any of your settings or internal network connections.

3. Open **Win7**.
4. Open PuTTY and connect into **IOSXRv** with the credentials:
   User: **root**
   Password: **root**
5. Study the limited pre-configurations of this IOS XRv router with **show run**. Note there are no **crypto** commands yet configured.

6. From the privilege exec mode, generate a dsa crypto key with the default **1024** bits.

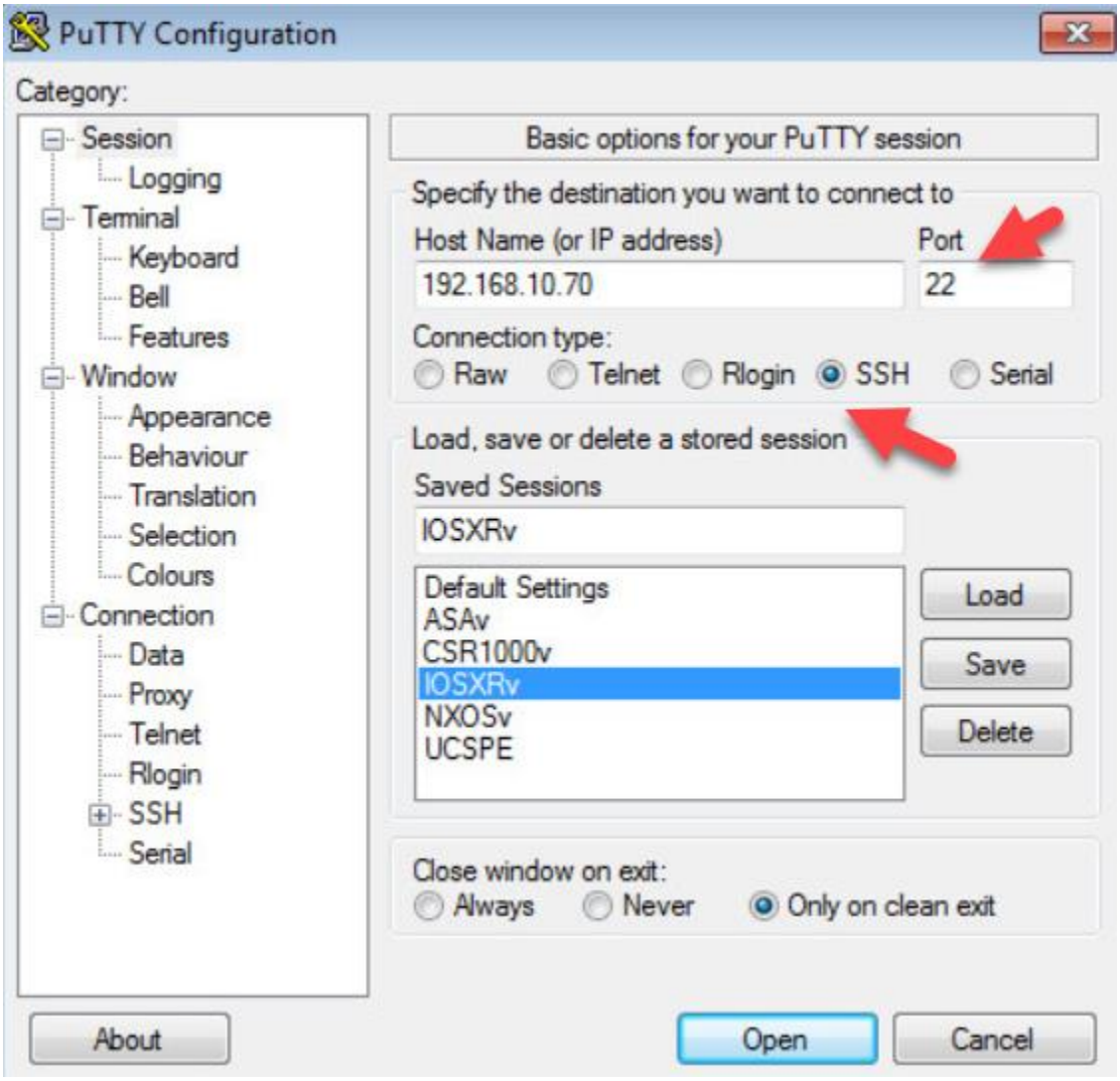
```
crypto key generate dsa
```

```
RP/0/RP0/CPU0:IOSXRv#crypto key generate dsa
Wed Oct  3 19:18:19.833 UTC
The name for the keys will be: the_default
  Choose the size of your DSA key modulus. Modulus size can be 512, 768, or 1024 bits. Choosing
odulus
How many bits in the modulus [1024]:
Generating DSA keys ...
Done w/ crypto generate keypair
[OK]

RP/0/RP0/CPU0:IOSXRv#
```

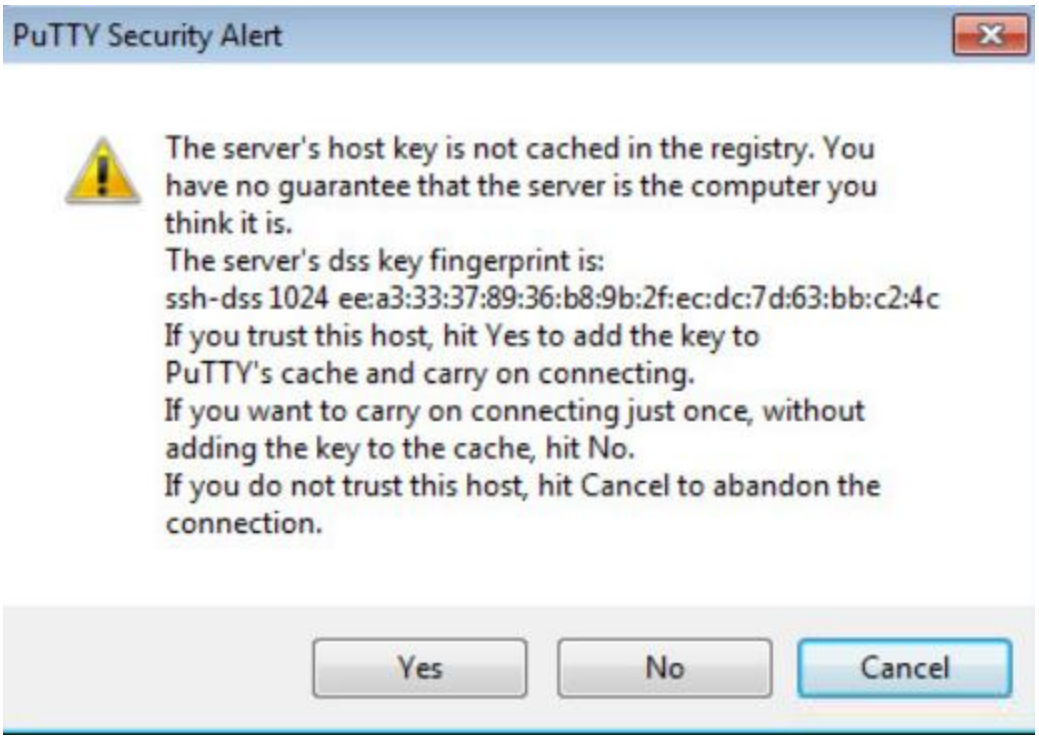
7. Enter config mode and enable **SSH** and **commit** your configuration.

```
RP/0/RP0/CPU0:IOSXRv#conf t
Wed Oct  3 19:21:31.012 UTC
RP/0/RP0/CPU0:IOSXRv(config)#ssh server v2
RP/0/RP0/CPU0:IOSXRv(config)#ssh server vrf default
RP/0/RP0/CPU0:IOSXRv(config)#commit
Wed Oct  3 19:22:05.565 UTC
RP/0/RP0/CPU0:IOSXRv(config)#end
```

8. Open a new Putty session this time with **SSH** to the **IOSXRv** device.



9. Accept the security prompt with **Yes**.



10. Login with the same credentials of user: **root** and password: **root**.
11. Further validate your SSH session.

```
RP/0/RP0/CPU0:IOSXRv#show ssh session details
Wed Oct  3 19:28:12.516 UTC
SSH version : Cisco-2.0

id  key-exchange                 pubkey     incipher    outcipher    inmac       outmac
-----------------------------------------------------------------------------------------
Incoming Session
0   diffie-hellman-group14     ssh-dss    aes256-ctr    aes256-ctr   hmac-sha2-256  hmac-sha2-256

Outgoing connection
RP/0/RP0/CPU0:IOSXRv#
```

12. Verify your TCP port 22 SSH session.

```
RP/0/RP0/CPU0:IOSXRv#show tcp brief
Wed Oct  3 19:29:21.893 UTC
   PCB              VRF-ID     Recv-Q Send-Q Local Address          Foreign Address        State
0x00007faaf4026bc8 0x60000000      0      0  :::22                  :::0                   LISTEN
0x00007faaf4022248 0x00000000      0      0  :::22                  :::0                   LISTEN
0x00007fab2c007f38 0x60000000      0      0  192.168.10.70:23       192.168.10.40:54212    ESTAB
0x00007fab2c00c5a8 0x60000000      0    144  192.168.10.70:22       192.168.10.40:54217    ESTAB
0x00007faaf4026948 0x60000000      0      0  0.0.0.0:22             0.0.0.0:0              LISTEN
0x00007faac4000fc8 0x00000000      0      0  0.0.0.0:22             0.0.0.0:0              LISTEN
0x00007faac400a4c8 0x60000000      0      0  0.0.0.0:23             0.0.0.0:0              LISTEN
0x00007faac4006038 0x00000000      0      0  0.0.0.0:23             0.0.0.0:0              LISTEN
RP/0/RP0/CPU0:IOSXRv#
```

13. From config mode, enable **NETCONF** with **SSH** and commit your settings. This may take several seconds to commit.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#conf t
Wed Oct  3 19:30:41.282 UTC
RP/0/RP0/CPU0:IOSXRv(config)#netconf-yang agent
RP/0/RP0/CPU0:IOSXRv(config-ncy-agent)#ssh
RP/0/RP0/CPU0:IOSXRv(config-ncy-agent)#commit
Wed Oct  3 19:30:55.153 UTC


RP/0/RP0/CPU0:IOSXRv(config-ncy-agent)#
```

14. Exit out of the **netconf-yang** mode and further configure **SSH** and **NETCONF**.

```
RP/0/RP0/CPU0:IOSXRv(config-ncy-agent)#exit
RP/0/RP0/CPU0:IOSXRv(config)#
RP/0/RP0/CPU0:IOSXRv(config)#ssh server netconf vrf default
RP/0/RP0/CPU0:IOSXRv(config)#netconf agent tty
```

15. Exit out of the **netconf** mode and configure the **netconf** port of 830 (this is the default setting). **Commit** your settings and **exit** from the config mode.

```
RP/0/RP0/CPU0:IOSXRv(config-netconf-tty)#exit
RP/0/RP0/CPU0:IOSXRv(config)#
RP/0/RP0/CPU0:IOSXRv(config)#ssh server netconf port 830
RP/0/RP0/CPU0:IOSXRv(config)#commit
Wed Oct  3 19:33:56.000 UTC
RP/0/RP0/CPU0:IOSXRv(config)#end
```

16. Verify your router is listening on TCP port 830 for NETCONF connections.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#show tcp brief
Wed Oct  3 19:40:02.632 UTC
   PCB              VRF-ID     Recv-Q Send-Q Local Address          Foreign Address        State
0x00007faabc005798 0x60000000      0      0  :::830                 :::0                   LISTEN
0x00007faabc001278 0x00000000      0      0  :::830                 :::0                   LISTEN
0x00007faaf4026bc8 0x60000000      0      0  :::22                  :::0                   LISTEN
0x00007faaf4022248 0x00000000      0      0  :::22                  :::0                   LISTEN
0x00007fab2c00c5a8 0x60000000      0    144  192.168.10.70:22       192.168.10.40:54217    ESTAB
0x00007faac400efe8 0x60000000      0      0  0.0.0.0:830            0.0.0.0:0              LISTEN
0x00007faac400ac48 0x00000000      0      0  0.0.0.0:830            0.0.0.0:0              LISTEN
0x00007faaf4026948 0x60000000      0      0  0.0.0.0:22             0.0.0.0:0              LISTEN
0x00007faac4000fc8 0x00000000      0      0  0.0.0.0:22             0.0.0.0:0              LISTEN
0x00007faac400a4c8 0x60000000      0      0  0.0.0.0:23             0.0.0.0:0              LISTEN
0x00007faac4006038 0x00000000      0      0  0.0.0.0:23             0.0.0.0:0              LISTEN
RP/0/RP0/CPU0:IOSXRv#
```

17. Go to your **Ubuntu Configured** VM. Verify you can login to your router from the terminal with **SSH** with the user **root** and the password: **root**

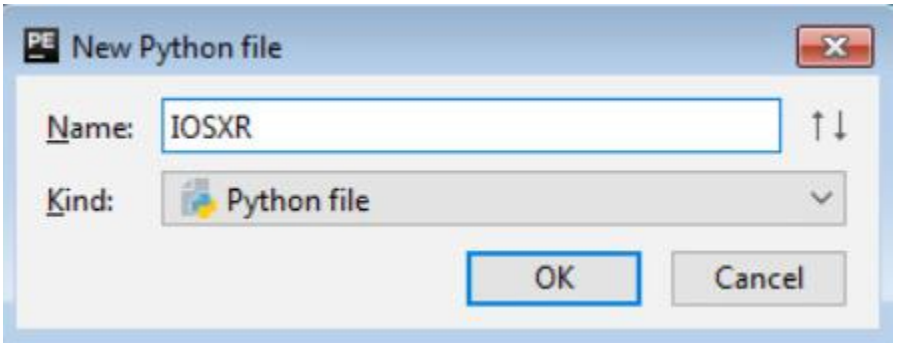
```
student@devops-full:~$ ssh root@192.168.10.70
Password:


RP/0/RP0/CPU0:IOSXRv#
```

Task 2: Perform a configuration change in the IOS XRv device with NETCONF and Python.

In this task you modify an existing Python script to configure a Loopback interface with NETCONF and XML.

How-to Steps

1. Open **PyCharm** on **Win7**.
2. Under your **class** project, create a new Python file called **IOSXR.**



3. Select **File > Open**. Select the file **Z:\Python\xr_nc_configure_ipinterface.py**
4. Copy the code into the **IOSXR.py** window.
5. Double click the title **IOSXR.py** to maximize the code window. Study the code.


```
#!/usr/bin/env python

import ...

if __name__ == "__main__":

    with manager.connect(host='xrv', port=830, username='cisco', password='cisco',
                         hostkey_verify=False, device_params={'name': 'iosxr'},
                         allow_agent=False, look_for_keys=False) as device:


        nc_filter = """
            <config>
              <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg">
               <interface-configuration>
                <active>act</active>
                <interface-name>Loopback100</interface-name>
                <interface-virtual/>
                <ipv4-network xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ipv4-io-cfg">
                 <addresses>
                  <primary>
                   <address>100.100.1.1</address>
                   <netmask>255.255.255.0</netmask>
                  </primary>
                 </addresses>
                </ipv4-network>
               </interface-configuration>
              </interface-configurations>
            </config>
        """

        nc_reply = device.edit_config(target='candidate', config=nc_filter)
        print nc_reply
        device.commit()
```

6. This code was not written for this lab environment. Make the following changes to this Python script:
   Host: **192.168.10.70**
   Port: **22**
   Username: **root**

Password: **root**

Note: Even though you previously configured TCP port 830 for NETCONF, this IOS XE image in the lab responds better to TCP port 22.

7. Remove the shebang as we will be running this in Windows.

xe_nc_configure_interface.py | IOSXR.py

```
from lxml import etree
from ncclient import manager

if __name__ == "__main__":

    with manager.connect(host='192.168.10.70', port=22, username='root', password='root',
                         hostkey_verify=False, device_params={'name': 'iosxr'},
                         allow_agent=False, look_for_keys=False) as device:


        nc_filter = """
            <config>
              <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg">
               <interface-configuration>
                <active>act</active>
                <interface-name>Loopback100</interface-name>
                <interface-virtual/>
                <ipv4-network xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ipv4-io-cfg">
                 <addresses>
                  <primary>
                   <address>100.100.1.1</address>
                   <netmask>255.255.255.0</netmask>
```

8. In your Putty session, verify that you do not have an existing interface **Loopback 100**.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#sh run int loopback 100
Wed Oct  3 19:48:42.527 UTC
% No such configuration item(s)

RP/0/RP0/CPU0:IOSXRv#
```

9. Run or step through your entire python script and ensure there are no errors.
10. Verify that you now have a configured interface **Loopback 100** interface.

```
RP/0/RP0/CPU0:IOSXRv#sh run int loo 100
Wed Oct  3 20:51:39.814 UTC
interface Loopback100
 ipv4 address 100.100.1.1 255.255.255.0
!

RP/0/RP0/CPU0:IOSXRv#
```

11. To further validate the NETCONF functionality, enable NETCONF debugging.

```
RP/0/RP0/CPU0:IOSXRv#terminal mon
Wed Oct  3 21:03:03.001 UTC
RP/0/RP0/CPU0:IOSXRv#debug netconf-yang all
```

12. Run your same python script again from PyCharm and validate that you didn't get any errors. Return to your Putty session and note all the debug output. Like most "all" options with debug, you will see more info then you need, but you will see full NETCONF functionality.
13. Return to **PyCharm** and edit your XML to make a new interface. Be careful not to alter any other XML characters.

```
Loopback 200
Ip address 10.100.1.1/24
```

```
nc_filter = """
    <config>
     <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg">
      <interface-configuration>
       <active>act</active>
       <interface-name>Loopback200</interface-name>
       <interface-virtual/>
       <ipv4-network xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ipv4-io-cfg">
        <addresses>
         <primary>
          <address>10.100.1.1</address>
          <netmask>255.255.255.0</netmask>
         </primary>
        </addresses>
       </ipv4-network>
      </interface-configuration>
     </interface-configurations>
    </config>
"""
```

14. Execute your script again and ensure no errors.
15. Return to your Putty session and note all the new debug output. You can turn off debug.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#unde all
```

16. Verify you have a new Interface Loopback 200 now configured.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#sh run int loo 200
Wed Oct  3 21:11:15.904 UTC
interface Loopback200
 ipv4 address 10.100.1.1 255.255.255.0
!
```

17. You can also disable **terminal monitor** if you don't want to see any other logging messages.

```
RP/0/RP0/CPU0:IOSXRv#
RP/0/RP0/CPU0:IOSXRv#terminal monitor disable
Wed Oct  3 21:12:51.597 UTC
RP/0/RP0/CPU0:IOSXRv#
```

Challenge results

This lab demonstrated editing the IOS XRv router with a Python script