#**
# **HOW TO use "ncclient" NETCONF python module to configure/provision in SROS** #
#**

#
# **Requirement**: First need to install the “ncclient” python library module using “pip”:
#

```
user@ubuntu:~/ pip install ncclient --user
```

#
# **Example-1**: Basic script to test connectivity to the SROS router using ncclient
#

```
#!/usr/bin/python

from ncclient import manager

device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={}, allow_agent=False,
look_for_keys=False)

print(device)

dir(device)
```

#
# **Example-2**: Using “**get**” operation to run “**show**” CLI command (‘show system information’)
#

```
#!/usr/bin/python
from ncclient import manager
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={}, allow_agent=False,
look_for_keys=False)

print(device)

get_filter = """
   <oper-data-format-cli-block>
        <cli-show>system information</cli-show>
   </oper-data-format-cli-block>
"""

nc_get_reply = device.get(('subtree', get_filter))

print(nc_get_reply)
```

```
ncclient.manager.Manager object at 0x7fea550b6fd0>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:9e289b38-7c69-4c76-845a-238e62a1e08d" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data xmlns="urn:alcatel-lucent.com:sros:ns:yang:cli-content-layer-r13">
        <oper-data-format-cli-block>
        <item>
```

```
                <cli-show>system information</cli-show>
                <response>
===============================================================================
System Information
===============================================================================
System Name            : XRS-20(140)
System Type            : 7950 XRS-20
Chassis Topology       : Standalone
System Version         : C-13.0.R10
System Contact         :
System Location        :
System Coordinates     :
System Active Slot     : A
System Up Time         : 7 days, 00:22:05.68 (hr:min:sec)

SNMP Port              : 161
SNMP Engine ID         : 0000197f00000ca402a5d801
SNMP Engine Boots      : 240
SNMP Max Message Size  : 9216
SNMP Admin State       : Disabled
SNMP Oper State        : Disabled
SNMP Index Boot Status : Persistent
SNMP Sync State        : Mismatch

Tel/Tel6/SSH/FTP Admin : Disabled/Disabled/Enabled/Disabled
Tel/Tel6/SSH/FTP Oper  : Down/Down/Up/Down

BOF Source             : cf3:
Image Source           : primary
Config Source          : primary
Last Booted Config File: cf3:\config.cfg
Last Boot Cfg Version  : THU JUL 26 15:51:21 2018 UTC
Last Boot Config Header: # TiMOS-C-13.0.R10 cpm/hops64 ALCATEL XRS 7950
                         Copyright (c) 2000-2016 Alcatel-Lucent. # All rights
                         reserved. All use subject to applicable license
                         agreements. # Built on Wed Jun 22 20:03:59 PDT 2016
                         by builder in /rel13.0/b1/R10/panos/main # Generated
                         THU JUL 26 15:51:21 2018 UTC
Last Boot Index Version: THU JUL 26 15:51:21 2018 UTC
Last Boot Index Header : # TiMOS-C-13.0.R10 cpm/hops64 ALCATEL XRS 7950
                         Copyright (c) 2000-2016 Alcatel-Lucent. # All rights
                         reserved. All use subject to applicable license
                         agreements. # Built on Wed Jun 22 20:03:59 PDT 2016
                         by builder in /rel13.0/b1/R10/panos/main # Generated
                         THU JUL 26 15:51:21 2018 UTC
Last Saved Config      : cf3:\config.cfg
Time Last Saved        : 2018/08/02 13:31:44
===============================================================================
                </response>
                </item>
                </oper-data-format-cli-block>
                </data>
</rpc-reply>
```

#
# **Example-3**: Using “**get**” operation to run miscellaneous “**show**” CLI commands
#

```
#!/usr/bin/python
from ncclient import manager
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={}, allow_agent=False,
look_for_keys=False)

print(device)

get_filter = """
   <oper-data-format-cli-block>
        <cli-show>router interface "system"</cli-show>
        <cli-show>system security ssh</cli-show>
        <cli-show>router route-table</cli-show>
   </oper-data-format-cli-block>
"""

nc_get_reply = device.get(('subtree', get_filter))

print(nc_get_reply)
```

```
<ncclient.manager.Manager object at 0x7f12f2314e50>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:7408cc0b-4b77-4b49-8c04-ebc286bf2700" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data xmlns="urn:alcatel-lucent.com:sros:ns:yang:cli-content-layer-r13">
        <oper-data-format-cli-block>
        <item>
        <cli-show>router interface "system"</cli-show>
        <response>
===============================================================================
Interface Table (Router: Base)
===============================================================================
Interface-Name                   Adm         Opr(v4/v6)  Mode    Port/SapId
   IP-Address                                PfxState
-------------------------------------------------------------------------------
system                           Up          Up/Down     Network system
   140.140.140.140/32                                    n/a
-------------------------------------------------------------------------------
Interfaces : 1
===============================================================================
        </response>
        </item>
        <item>
        <cli-show>system security ssh</cli-show>
        <response>
===============================================================================
SSH Server
===============================================================================
Administrative State    : Enabled
Operational State       : Up
Preserve Key            : Disabled
 SSH Protocol Version 1              : Disabled
SSH Protocol Version 2              : Enabled
DSA Host Key Fingerprint  : db:a2:ae:8b:be:29:29:3d:0a:1c:02:ae:a1:76:d9:eb
RSA Host Key Fingerprint  : 9a:57:4b:d2:61:f3:73:52:bb:74:f2:d7:98:d8:e5:e1
-------------------------------------------------------------------------------
Connection                          Username
              Version Cipher                       ServerName  Status
-------------------------------------------------------------------------------
aa.bb.cc.dd                         admin
              2           aes256-ctr           cli          connected
aa.bb.cc.dd                         netcfg_user
              2           aes128-ctr           netconf      connected
aa.bb.cc.dd                         netcfg_user
              2           aes128-ctr           netconf      connected
-------------------------------------------------------------------------------
Number of SSH sessions : 3
===============================================================================
              </response>
              </item>
              <item>
              <cli-show>router route-table</cli-show>
              <response>
===============================================================================
Route Table (Router: Base)
===============================================================================
Dest Prefix[Flags]                   Type       Proto      Age          Pref
              Next Hop[Interface Name]                     Metric
-------------------------------------------------------------------------------
10.10.10.10/32                  Local   Local           07d02h33m  0
              loopback                                  0
40.40.40.40/32                  Local   Local           00h34m50s  0
              test                                      0
140.140.140.140/32              Local   Local           07d02h33m  0
              system                                    0
192.168.1.0/31                  Local   Local           07d02h30m  0
              testing                                   0
-------------------------------------------------------------------------------
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
              B = BGP backup route available
              L = LFA nexthop available
              S = Sticky ECMP requested
===============================================================================
              </response>
              </item>
              </oper-data-format-cli-block>
              </data>
</rpc-reply>
```

#
# **Example-4**: Using "**get-config**" operation to retrieve the entire '**running-config**'
# (Note: Running the "get-config" method without a filter argument, will retrieve the entire config)
#

```
#!/usr/bin/python
from ncclient import manager
```

```
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={'name':'alu'}, allow_agent=False,
look_for_keys=False)

print(device)

nc_get_reply = device.get_config('running')

print(nc_get_reply)
```

#
# **Example-5**: Using “**get-config**” operation to retrieve just a portion of the configuration.
# Specifically, in this case to get the "system" interface configuration
#

```
#!/usr/bin/python
from ncclient import manager
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={'name':'alu'}, allow_agent=False,
look_for_keys=False)

print(device)

get_filter = """
<filter>
  <configure>
        <router>
        <interface>
        <interface-name>"system"</interface-name>
        </interface>
        </router>
  </configure>
</filter>
"""

nc_get_reply = device.get_config('running', get_filter)

print(nc_get_reply)
```

```
<ncclient.manager.Manager object at 0x7fb063fe0f10>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:8b070f74-fb9c-4b37-9775-7500ff9c9189" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data>
        <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
        <router>
        <router-name>Base</router-name>
        <interface>
        <interface-name>system</interface-name>
        <address>
                <ip-address-mask>140.140.140.140/32</ip-address-mask>
        </address>
        <shutdown>false</shutdown>
        </interface>
        </router>
        </configure>
        </data>
</rpc-reply>
```

#
# **Example-6**: Using "**get-config**" operation to retrieve a "loopback" interface configuration.
#

```
#!/usr/bin/python
from ncclient import manager
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={'name':'alu'}, allow_agent=False,
look_for_keys=False)

print(device)

get_filter = """
<filter>
  <configure>
        <router>
        <interface>
        <interface-name>"loopback"</interface-name>
        </interface>
        </router>
  </configure>
</filter>
"""

nc_get_reply = device.get_config('running',get_filter)

print(nc_get_reply)
```

```
<ncclient.manager.Manager object at 0x7f20fdb98f10>
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:5de03199-f980-48a7-a4bd-df1efbfa1516" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data>
        <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
        <router>
        <router-name>Base</router-name>
        <interface>
        <interface-name>loopback</interface-name>
        <address>
                <ip-address-mask>10.10.10.10/32</ip-address-mask>
        </address>
        <loopback>true</loopback>
        <shutdown>false</shutdown>
        </interface>
        </router>
        </configure>
        </data>
</rpc-reply>
```

#
# **Example-7**: Using “**edit-config**” operation to create a new loopback interface.
#

```
#!/usr/bin/python
from ncclient import manager
device = manager.connect(host='ww.xx.yy.zz', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={'name':'alu'}, allow_agent=False,
look_for_keys=False)

print(device)

cfg = """
 <config>
        <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
                <router>
                <router-name>Base</router-name>
                <interface>
                        <interface-name>test</interface-name>
                        <address>
                        <ip-address-mask>40.40.40.40/32</ip-address-mask>
                        </address>
                        <loopback>true</loopback>
                        <shutdown>false</shutdown>
                </interface>
                </router>
        </configure>
 </config>
"""

nc_set_reply = device.edit_config(target='running', config=cfg)

print(nc_set_reply)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:509dfa15-3cbb-452b-93ae-954b06df441c" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <ok/>
</rpc-reply>
```

#***

Interactive session in python shell

#***

```
user@ubuntu:~/netconf/ncclient$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
```

#

# Basic stuff to test connectivity to the SROS router using ncclient

#

```
>>>
>>> from ncclient import manager
>>>
>>> device = manager.connect(host='aa.bb.cc.dd', port=830, username='netcfg_user',
password='netconf', hostkey_verify=False, device_params={'name':'alu'}, allow_agent=False,
look_for_keys=False)
>>>
>>> print(device)
<ncclient.manager.Manager object at 0x7f47d3f28350>
>>>
```

#

# Which methods are available?

#

```
>>>
>>> dir(device)
['_Manager__set_async_mode', '_Manager__set_raise_mode', '_Manager__set_timeout', '__class__',
'__delattr__', '__dict__', '__doc__', '__enter__', '__exit__', '__format__', '__getattr__',
'__getattribute__', '__hash__', '__init__', '__module__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
'__weakref__', '_async_mode', '_device_handler', '_raise_mode', '_session', '_timeout',
'async_mode', 'cancel_commit', 'channel_id', 'channel_name', 'client_capabilities',
'close_session', 'commit', 'connected', 'copy_config', 'create_subscription', 'delete_config',
'discard_changes', 'dispatch', 'edit_config', 'execute', 'get', 'get_config', 'get_schema',
'kill_session', 'lock', 'locked', 'poweroff_machine', 'raise_mode', 'reboot_machine', 'scp',
'server_capabilities', 'session', 'session_id', 'take_notification', 'timeout', 'unlock',
'validate']
>>>
```

#

# Using “**get**” operation to run “**show**” CLI command (‘show system information’)

#

```
>>>
>>> get_filter = """
...     <oper-data-format-cli-block>
...     <cli-show>system information</cli-show>
...     </oper-data-format-cli-block>
... """
>>>
```

```
>>> nc_get_reply = device.get(('subtree', get_filter))
>>>
>>> print(nc_get_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:6d23caf1-9f36-4679-8d16-7fd317a28dc7"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data xmlns="urn:alcatel-lucent.com:sros:ns:yang:cli-content-layer-r13">
        <oper-data-format-cli-block>
                <item>
                <cli-show>system information</cli-show>
                <response>
===============================================================================
System Information
===============================================================================
System Name            : XRS-20(140)
System Type            : 7950 XRS-20
Chassis Topology       : Standalone
System Version         : C-13.0.R10
System Contact         :
System Location        :
System Coordinates     :
System Active Slot     : A
System Up Time         : 7 days, 02:22:55.42 (hr:min:sec)

SNMP Port              : 161
SNMP Engine ID         : 0000197f00000ca402a5d801
SNMP Engine Boots      : 240
SNMP Max Message Size  : 9216
SNMP Admin State       : Disabled
SNMP Oper State        : Disabled
SNMP Index Boot Status : Persistent
SNMP Sync State        : Mismatch

Tel/Tel6/SSH/FTP Admin : Disabled/Disabled/Enabled/Disabled
Tel/Tel6/SSH/FTP Oper  : Down/Down/Up/Down

BOF Source             : cf3:
Image Source           : primary
Config Source          : primary
Last Booted Config File: cf3:\config.cfg
Last Boot Cfg Version  : THU JUL 26 15:51:21 2018 UTC
Last Boot Config Header: # TiMOS-C-13.0.R10 cpm/hops64 ALCATEL XRS 7950
                         Copyright (c) 2000-2016 Alcatel-Lucent. # All rights
                         reserved. All use subject to applicable license
                         agreements. # Built on Wed Jun 22 20:03:59 PDT 2016
                         by builder in /rel13.0/b1/R10/panos/main # Generated
                         THU JUL 26 15:51:21 2018 UTC
Last Boot Index Version: THU JUL 26 15:51:21 2018 UTC
Last Boot Index Header : # TiMOS-C-13.0.R10 cpm/hops64 ALCATEL XRS 7950
                         Copyright (c) 2000-2016 Alcatel-Lucent. # All rights
                         reserved. All use subject to applicable license
                         agreements. # Built on Wed Jun 22 20:03:59 PDT 2016
                         by builder in /rel13.0/b1/R10/panos/main # Generated
                         THU JUL 26 15:51:21 2018 UTC
Last Saved Config      : cf3:\config.cfg
```

```
Time Last Saved        : 2018/08/02 13:31:44
Changes Since Last Save: Yes
User Last Modified     : netcfg_user
Time Last Modified     : 2018/08/02 18:00:04
Max Cfg/BOF Backup Rev : 5
Cfg-OK Script          : N/A
Cfg-OK Script Status    : not used
Cfg-Fail Script        : N/A
Cfg-Fail Script Status : not used

Management IP Addr     : aa.bb.cc.dd/24
Primary DNS Server     : N/A
Secondary DNS Server    : N/A
Tertiary DNS Server    : N/A
DNS Domain             : (Not Specified)
DNS Resolve Preference : ipv4-only
DNSSEC AD Validation    : False
DNSSEC Response Control: drop
BOF Static Routes      :
  To                   Next Hop
  0.0.0.0/1            ww.xx.yy.zz

  128.0.0.0/1          zz.yy.xx.ww

ATM Location ID        : 01:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
ATM OAM Retry Up       : 2
ATM OAM Retry Down     : 4
ATM OAM Loopback Period: 10

ICMP Vendor Enhancement: Disabled
EFM OAM Grace Tx Enable: False
===============================================================================
               </response>
               </item>
       </oper-data-format-cli-block>
       </data>
</rpc-reply>
>>>
```

#

# Using “**get**” operation to run miscellaneous “**show**” CLI commands

#

```
>>>
>>> get_filter = """
...     <oper-data-format-cli-block>
...     <cli-show>router interface "system"</cli-show>
...     <cli-show>system security ssh</cli-show>
...     <cli-show>router route-table</cli-show>
...     </oper-data-format-cli-block>
... """
>>> nc_get_reply = device.get(('subtree', get_filter))
>>> print(nc_get_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:e69061ba-4fe5-4b4d-b0ba-b2de71b8f2ee"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
        <data xmlns="urn:alcatel-lucent.com:sros:ns:yang:cli-content-layer-r13">
        <oper-data-format-cli-block>
                <item>
                <cli-show>router interface "system"</cli-show>
                <response>
===============================================================================
Interface Table (Router: Base)
===============================================================================
Interface-Name                 Adm            Opr(v4/v6)  Mode       Port/SapId
   IP-Address                                                        PfxState
-------------------------------------------------------------------------------
system                         Up             Up/Down        Network system
   140.140.140.140/32                                                n/a
-------------------------------------------------------------------------------
Interfaces : 1
===============================================================================
                </response>
                </item>
                <item>
                <cli-show>system security ssh</cli-show>
                <response>
===============================================================================
SSH Server
===============================================================================
Administrative State   : Enabled
Operational State      : Up
Preserve Key           : Disabled

SSH Protocol Version 1 : Disabled

SSH Protocol Version 2 : Enabled
DSA Host Key Fingerprint  : db:a2:ae:8b:be:29:29:3d:0a:1c:02:ae:a1:76:d9:eb
RSA Host Key Fingerprint  : 9a:57:4b:d2:61:f3:73:52:bb:74:f2:d7:98:d8:e5:e1
-------------------------------------------------------------------------------
Connection                                    Username
        Version Cipher                        ServerName  Status
-------------------------------------------------------------------------------
ll.mm.nn.pp                           admin
        2       aes256-ctr                    cli            connected
ll.mm.nn.pp                           netcfg_user
        2       aes128-ctr                    netconf        connected
-------------------------------------------------------------------------------
Number of SSH sessions : 2
===============================================================================
                </response>
                </item>
                <item>
                <cli-show>router route-table</cli-show>
                <response>
===============================================================================
Route Table (Router: Base)
===============================================================================
Dest Prefix[Flags]                            Type    Proto   Age     Pref
        Next Hop[Interface Name]                                      Metric
-------------------------------------------------------------------------------
10.10.10.10/32                                Local   Local  07d02h34m  0
        loopback                                                      0
```

```
40.40.40.40/32                               Local    Local   00h36m02s  0
       test                                                          0
140.140.140.140/32                           Local    Local   07d02h34m  0
       system                                                        0
192.168.1.0/31                               Local    Local   07d02h32m  0
       testing                                                       0
-------------------------------------------------------------------------------
No. of Routes: 4
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
===============================================================================
              </response>
              </item>
       </oper-data-format-cli-block>
       </data>
</rpc->eply>
```

#
# Using “**get-config**” operation to retrieve the complete ‘**running-config**’
# (Note: Running the "**get-config**" method without a filter arg, will retrieve the entire config)
#

```
nc_get_reply = device.get_config('running')
print(nc_get_reply)
<...output omitted...>
```

#
# Using “**get-config**” operation to retrieve just a portion of the configuration.
# Specifically, in this case to get the "**system**" interface configuration
#

```
>>> get_filter = """
...<filter>
...  <configure>
...    <router>
...    <interface>
...            <interface-name>"system"</interface-name>
...    </interface>
...    </router>
...   </configure>
... </filter>
... """
>>>
>>> nc_get_reply = device.get_config('running',get_filter)
>>> print(nc_get_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:72414b0b-bf1a-4cfa-8912-f08bcabb04b3"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
       <data>
       <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
              <router>
              <router-name>Base</router-name>
```

```
                <interface>
                        <interface-name>system</interface-name>
                        <address>
                        <ip-address-mask>140.140.140.140/32</ip-address-mask>
                        </address>
                        <shutdown>false</shutdown>
                </interface>
                </router>
</configu>e>
nfigure>
...     <router>
...     <interface>
...             <interface-name>"system"</interface-name>
...     </interface>
...     </router>
...   </configure>
... </filter>
... """
>>>
>>> nc_get_reply = device.get_config('running',get_filter)
>>> print(nc_get_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:a4d865b1-3610-47a4-ada4-de1677251ba3"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data>
        <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
                <router>
                <router-name>Base</router-name>
                <interface>
                        <interface-name>system</interface-name>
                        <address>
                        <ip-address-mask>140.140.140.140/32</ip-address-mask>
                        </address>
                        <shutdown>false</shutdown>
                </interface>
                </router>
        </configure>
        </data>
</rpc-reply>
>>>
```

#
# Using “**get-config**” operation to retrieve a “**loopback**” interface configuration
#

```
>>>
>>> get_filter = """
... <filter>
...   <configure>
...     <router>
...     <interface>
...             <interface-name>"loopback"</interface-name>
...     </interface>
...     </router>
...   </configure>
... </filter>
... """
>>>
>>> nc_get_reply = device.get_config('running',get_filter)
>>> print(nc_get_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:382a52bf-dc80-4570-bb58-47ed33c51879"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <data>
        <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
                <router>
                <router-name>Base</router-name>
                <interface>
                        <interface-name>loopback</interface-name>
                        <address>
                        <ip-address-mask>10.10.10.10/32</ip-address-mask>
                        </address>
                        <loopback>true</loopback>
                        <shutdown>false</shutdown>
                </interface>
                </router>
        </configure>
        </data>
nfig>
```

#
# Using “**edit-config**” operation to create a new loopback interface called “**test**”
#

```
>>> cfg = """
... <config>
...             <configure xmlns="urn:alcatel-lucent.com:sros:ns:yang:conf-r13">
...             <router>
...                     <router-name>Base</router-name>
...                     <interface>
...                     <interface-name>test</interface-name>
...                     <address>
...                             <ip-address-mask>40.40.40.40/32</ip-address-mask>
...                     </address>
...                     <loopback>true</loopback>
...                     <shutdown>false</shutdown>
...                     </interface>
...             </router>
...             </configure>
...
...  </config>
... """
>>>
>>> nc_set_reply = device.edit_config(target='running', config=cfg)
>>> print(nc_set_reply)
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="urn:uuid:509dfa15-3cbb-452b-93ae-954b06df441c"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <ok/>
</rpc-reply>
>>>
```