

- 1) Verbinden Sie sich mit dem Jump Host auf dem Guacamole Server mit den vom Trainer zur Verfügung gestellten Zugangsdaten. Dort sollte bereits ein Fenster mit einer RDP Sitzung zu einer Ubuntu VM in der Testnetz-Umgebung geöffnet sein, das Sie maximieren oder auf Vollbild einstellen sollten. Im Verzeichnis /NSO1 unterhalb des Home-Verzeichnisses finden Sie diverse für die Übungen nützliche Dateien.
- 2) Aktivieren sie auf der Ubuntu VM mit dem Icon MATE-Terminal auf deren Desktop eine Shell und wechseln Sie in das Directory nso-5.1/Lab:

```
et@ubuntu:~$ cd nso-5.1/Lab
```

- 3) Definieren Sie den Pfad zum Shell-Skript zur Ausführung der NSO-Kommandos:

```
et@ubuntu:~/nso-5.1/Lab$ source $HOME/nso-5.1/ncsrc
```

Alle NSO-Kommandos stehen nun im aktuellen Verzeichnis zur Verfügung.

- 4) Das Verzeichnis Lab ist als funktionstüchtige Runtime-Umgebung für den NSO Demon mit einem NETCONF NED für JUNOS präpariert worden. Starten Sie den NSO Demon mit

```
et@ubuntu:~/nso-5.1/Lab$ ncs
```

- 5) Starten Sie das CLI des NSO:

```
et@ubuntu:~/nso-5.1/Lab$ ncs_cli -u admin
```

```
admin connected from 127.0.0.1 using console on ubuntu
admin@ncs>
```

- 6) Verifizieren sie dass der NED für JUNOS operabel ist:

```
admin@ncs> show packages package oper-status
packages package juniper-junos-nc-3.0
oper-status up
```

- 7) Überprüfen Sie mit Ping die Erreichbarkeit der Juniper Router von der VM aus:

```
et@ubuntu:~$ ping 192.168.255.110
PING 192.168.255.110 (192.168.255.110) 56(84) bytes of data.
64 bytes from 192.168.255.110: icmp_seq=1 ttl=128 time=9.77 ms
64 bytes from 192.168.255.110: icmp_seq=2 ttl=128 time=6.89 ms
64 bytes from 192.168.255.110: icmp_seq=3 ttl=128 time=7.05 ms
^C
--- 192.168.255.110 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 6.897/7.908/9.778/1.325 ms
```

```
et@ubuntu:~$ ping 192.168.255.120
```

```
PING 192.168.255.120 (192.168.255.120) 56(84) bytes of data.  
64 bytes from 192.168.255.120: icmp_seq=1 ttl=128 time=9.31 ms  
64 bytes from 192.168.255.120: icmp_seq=2 ttl=128 time=6.90 ms  
64 bytes from 192.168.255.120: icmp_seq=3 ttl=128 time=7.12 ms
```

(Die angegebenen IP-Adressen sind nur exemplarisch. Diese müssen Sie durch die wirklichen IP Adressen der Juniper Router in ihrer Gruppe ersetzen!)

- 8) Bauen Sie von einer zusätzlichen Shell aus je eine CLI Session zu beiden Juniper Routern auf (wahlweise mit Telnet oder SSH, Login lab/lab123). Damit soll später die Wirkung der Service Creation durch den NSO verifiziert werden:

```
et@ubuntu:~$ telnet <IP-Adresse>    oder    et@ubuntu:~$ ssh -l lab <IP-Adresse>
```

- 9) Wechseln die mit configure in den Configuration Mode des NSO. Konfigurieren Sie eine Authentication Group myJuniper mit den Credentials lab/lab123 für die Juniper Router:

```
[edit]  
admin@ncs% set devices authgroups group myJuniper default-map  
admin@ncs% set devices authgroups group myJuniper default-map remote-name lab  
admin@ncs% set devices authgroups group myJuniper default-map remote-password ...  
... lab123
```

- 10) Konfigurieren Sie auf dem NSO je eine Device-Instanz für die beiden Juniper Router mit den Namen j1 und j2. Das sollte, wenn es fertig ist, ungefähr so aussehen (für den Router j1, die IP-Adresse ist hier nur exemplarisch und Sie müssen diese für ihre beiden Router entsprechend anpassen):

```
[edit]  
admin@ncs% set devices device j1 address 192.168.255.110  
admin@ncs% set devices device j1 port 830  
admin@ncs% set devices device j1 authgroup myJuniper  
admin@ncs% set devices device j1 device-type netconf ned-id juniper-junos-nc-3.0  
admin@ncs% set devices device j1 state admin-state unlocked  
  
[edit]  
admin@ncs% commit
```

- 11) Erzeugen Sie für die NETCONF Session mit den beiden Juniper Routern ein Schlüsselpaar:

```
admin@ncs> request devices device j1 ssh fetch-host-keys  
admin@ncs> request devices device j2 ssh fetch-host-keys
```

- 12) Nun können Sie den NSO mit den beiden Routern verbinden:

```
admin@ncs> request devices connect  
connect-result {
```

```

device j1
result true
info (admin) Connected to j1 - 192.168.255.110:830
}
connect-result {
device j2
result true
info (admin) Connected to j2 - 192.168.255.120:830
}

```

- 13) Synchronisieren Sie die lokale Konfiguration der Device-Instanzen auf dem NSO mit der tatsächlichen Konfiguration der beiden Juniper-Router:

```
admin@ncs> request devices sync-from
```

```

sync-result {
device j1
result true
}
sync-result {
device j2
result true
}

```

- 14) In dem config-Bereich der Device-Instanzen sollten Sie nun die aktuelle Konfiguration der beiden Juniper Router sehen können:

```

admin@ncs> show configuration devices device j1 config
admin@ncs> show configuration devices device j2 config

```

Vergleichen Sie dies mit der Konfiguration, die sie direkt auf den Routern über das CLI abrufen können :

```
lab@vSRX_xy> show configuration
```

Abgesehen von der System-und Management-Konfiguration sollten Sie ein physisches Interface mit einer IP-Adresse sehen. Darüber sind die beiden Router einer Gruppe untereinander verbunden und sollten sich per Ping erreichen können. Verifizieren sie das über die CLI Sessions zu den Routern.

- 15) Damit ist alles für die Einrichtung eines Services (Service Building) auf dem NSO präpariert. Dieser soll folgendes leisten:

- Zuweisung einer individuellen Loopback-Adresse und Router ID
- Zuweisung einer gemeinsamen AS-Nummer für BGP
- Automatische Aktivierung von OSPF auf allen aktiven Schnittstellen in der Area 0
- Automatischer Aufbau einer iBGP-Nachbarschaft zwischen den beiden Routern

- 16) Erzeugen Sie dazu ein Service Package vom Typ Template Only mit dem Namen IBGP. Öffne sie als Vorbereitung dafür eine weitere Shell, wechseln Sie in das Verzeichnis /nso-5.1/Lab/packages und definieren sie die Source:

```
et@ubuntu:~/nso-5.1/Lab/packages$ source $HOME/nso-5.1/ncsrc
```

Zur Erzeugung des Service Package dient das Kommando ncs-make-package mit geeigneten Optionen. Achten Sie dabei unbedingt darauf, dass der Package Name IBGP stimmt!! Die folgenden Aufgaben und einige der Vorlagen, die Sie nutzen können, beziehen sich auf genau diesen Namen.

```
et@ubuntu:~/nso-5.1/Lab/packages$ ncs-make-package --service-skeleton template IBGP
```

Im Verzeichnis /packages ist nun ein weiteres Unterverzeichnis mit dem Package Namen IBGP aufgetaucht:

```
et@ubuntu:~/nso-5.1/Lab/packages$ ls
IBGP juniper-junos-nc-3.0
```

- 17) Wechseln Sie in das Verzeichnis Lab/packages/IBGP/src/yang. Öffnen Sie das YANG Skeleton IBGP.yang mit dem Sublime-Editor

```
et@ubuntu:~/nso-5.1/Lab/packages/IBGP/src/yang$ subl IBGP.yang
```

und editieren Sie die Datei gemäß folgender Vorgaben:

- Import des YANG-Moduls tailf-common mit Prefix tailf
- Ein Leaf Node ASN für die gemeinsame AS Nummer mit Type uint16
- Zwei Container Peer1 und Peer2 für jeden BGP Nachbarn
- In jedem Container Peer1 bzw. Peer2 ein Leaf Node device und ein Leaf Node loopback-address
- Der Leaf Node device soll die im NSO tatsächlich konfigurierten Devices referenzieren

```
leaf device {
  type leafref {
    path "/ncs:devices/ncs:device/ncs:name";
  }
}
```

- Der Leaf Node loopback-address hat den Type inet:ipv4-address

```
leaf loopback-address {
  type inet:ipv4-address;
}
```

Sie können sich bei der Konfiguration des Service YANG-Modells an der Vorlage NSO_YANG_Vorlage.yang orientieren. Wie könnte man das YANG-Modell in der Vorlage

optimieren (verkürzen)?

Falls ihre eigene Konfiguration nicht gelingt, können sie den kompletten Inhalt der Vorlage NSO_YANG_Vorlage.yang in die Zwischenablage kopieren und nach IBGP.yang pasten. Nach getaner Arbeit in Sublime mit Strg-S abspeichern!

Untersuchen Sie den Aufbau des YANG Models und versuchen Sie seine Wirkung bei der Service Creation zu verstehen.

18) Compilieren Sie nun das YANG-Modell mit dem Makefile unter Lab/packages/IBGP/src:

```
et@ubuntu:~/nso-5.1/Lab/packages/IBGP/src$ make clean all
rm -rf ../load-dir
mkdir -p ../load-dir
/home/et/nso-5.1/bin/ncsc `ls IBGP-ann.yang` > /dev/null 2>&1 && echo "-a IBGP-ann.yang" \
-c -o ../load-dir/IBGP.fxs yang/IBGP.yang
```

Sollten hier Fehler gemeldet werden, versuchen Sie diese zu beheben. Rufen Sie notfalls den Trainer zur Hilfe.

19) Wechseln Sie in das Verzeichnis Lab/packages/IBGP/templates und öffnen sie das XML Skeleton IBGP-template.xml mit Sublime. Editieren Sie die Datei gemäß folgender Vorgaben:

- Je ein Block <device> ... </device> für das Device in Container Peer1 und Peer2
- Für jeden Block folgende Konfigurationen im XML Format:
 - + interface lo0 mit IP-Adresse aus Leaf Node loopback-address
 - + routing-options mit AS Nummer aus Leaf Node ASN und Router ID aus Leaf Node loopback-address
 - + protocols ospf mit allen Interface in der Area 0 mit Ausnahme des Management-Interfaces fxp0.0
 - + protocols bgp mit einer Peer Group namens internal-Peers und in der Peer Group type internal, local-address mit der IP-Adresse im Leaf Node loopback-address des lokalen Devices und neighbor mit der IP-Adresse aus dem Leaf Node loopback-address des Peers

Sie können die erforderlichen Konfigurations-Teile wahlweise direkt auf den Routern im CLI oder in der Device-Konfiguration des NSO erzeugen. Das sieht dann im Falle der Konfiguration des Loopback-Interfaces in etwa wie folgt aus:

Direkt mit dem CLI (hier müssen die JUNOS-spezifischen Attribute im Element <configuration> durch den im NSO verwendeten Namespace xmlns=http://xml.juniper.net/xnm/1.1/xnm ersetzt werden):

```
[edit]
lab@vSRX_xy# set interfaces lo0 unit 0 family inet address 1.2.3.4/32
[edit]
lab@vSRX_xy# show interfaces lo0 | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/15.1R6/junos">
```

```

<configuration junos:changed-seconds="1574328067" junos:changed-localtime="2019-11-21
09:21:07 UTC">
  <interfaces>
    <interface>
      <name>lo0</name>
      <unit>
        <name>0</name>
        <family>
          <inet>
            <address>
              <name>1.2.3.4/32</name>
            </address>
          </inet>
        </family>
      </unit>
    </interface>
  </interfaces>
</configuration>
<cli>
  <banner>[edit]</banner>
</cli>
</rpc-reply>

```

Auf dem NSO:

```
[edit devices device j1 config configuration]
```

```
admin@ncs% set interfaces interface lo0 unit 0 family inet address 1.2.3.4/32
```

```
[edit devices device j1 config configuration]
```

```
admin@ncs% commit dry-run outformat xml
```

```

result-xml {
  local-node {
    data <devices xmlns="http://tail-f.com/ns/ncs">
      <device>
        <name>j1</name>
        <config>
          <configuration xmlns="http://xml.juniper.net/xnm/1.1/xnm">
            <interfaces>
              <interface>
                <name>lo0</name>
                <unit>
                  <name>0</name>
                  <family>
                    <inet>
                      <address>
                        <name>1.2.3.4/32</name>
                      </address>
                    </inet>
                  </family>
                </unit>
              </interface>
            </interfaces>
          </configuration>
        </config>
      </device>
    </data>
  }
}

```

```

        </unit>
      </interface>
    </interfaces>
  </configuration>
</config>
</device>
</devices>
}
}

```

Die eingegebenen Statement müssen danach mit

```

[edit devices device j1 config configuration]
admin@ncs% revert

```

annulliert werden.

Sie müssen dann die Konfigurations-Parameter durch die aus den Leaf Nodes des YANG-Modells erzeugten Variablen ersetzen. Also im voranstehenden Beispiel die Adresse 1.2.3.4 durch {/Peer1/loopback-address} bzw. {/Peer2/loopback-address}. Der rot hervorgehobene Teil muss im XML Template für jedes Device in den Bereich zwischen <config> und </config> eingefügt werden.

Sie können sich bei der Arbeit gerne an der Vorlage NSO_XML_Vorlage.xml orientieren. Sollten sie sich beim manuellen Editieren verzetteln, können sie auch hier aus der Vorlage, die den vollständigen finalen Inhalt des Templates enthält, kopieren. Speichern Sie abschließend mit Strg-S ab.

- 20) Zum Aktivieren des Service Package muss im CLI des NSO ein Reload aller Packages erzwungen werden:

```

admin@ncs> request packages reload

```

```

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
  package IBGP
  result true
}
reload-result {
  package juniper-junos-nc-3.0
  result true
}

```

Das YANG-Modell IBGP.yang wird dadurch in den Configuration Mode des NSO integriert. Die Statements aus dem YANG-Modell stehen als Konfigurations-Kommandos zur Verfügung.

- 21) Konfigurieren Sie jetzt auf dem NSO eine Instanz des Service IBGP mit einem beliebigen Namen. Das sollte am Ende ungefähr folgendermaßen aussehen:

```

[edit IBGP TEST]
admin@ncs% show
ASN 65000;
Peer1 {
    device    j1;
    loopback-address 1.1.1.1;
}
Peer2 {
    device    j2;
    loopback-address 2.2.2.2;
}

```

Mit Hilfe von set-Kommandos gibt man das wie folgt ein:

```

[edit]
admin@ncs% set IBGP TEST ASN 65000
[edit]
admin@ncs% set IBGP TEST Peer1 device j1
[edit]
admin@ncs% set IBGP TEST Peer1 loopback-address 1.1.1.1
[edit]
admin@ncs% set IBGP TEST Peer2 device j2
[edit]
admin@ncs% set IBGP TEST Peer2 loopback-address 2.2.2.2

```

```

admin@ncs% commit dry-run
cli {
    local-node {
        data devices {
            device j1 {
                config {
                    junos:configuration {
                        interfaces {
+                interface lo0 {
+                unit 0 {
+                family {
+                inet {
+                address 1.1.1.1/32;

```

<snip>

```

[edit IBGP TEST]
admin@ncs% commit

```


Commit complete.

[ok][2019-11-20 14:39:50]

- 22) Verifizieren Sie anhand der CLI Session mit den Juniper Routern die Funktionstüchtigkeit der Service-konfiguration:

lab@vSRX_xy> [show ospf neighbor](#)

lab@vSRX_xy> [show bgp summary](#)

- 23) Spielen sie anhand der existierenden Service-Konfiguration die Operationen Update (z.B. Ändern der ASN oder Loopback-Adresse) und Delete durch und verifizieren Sie die Wirkung in der lokalen Device-Konfiguration auf dem NSO und mit Hilfe der CLI Session direkt auf den Routern.
- 24) Zusatzaufgabe: Was geschieht, wenn Sie eine zweite Service-Instanz konfigurieren? Wie kombiniert sich deren Wirkung mit der vorhandenen?