Automating nationwide deployment of a 400Gbps network

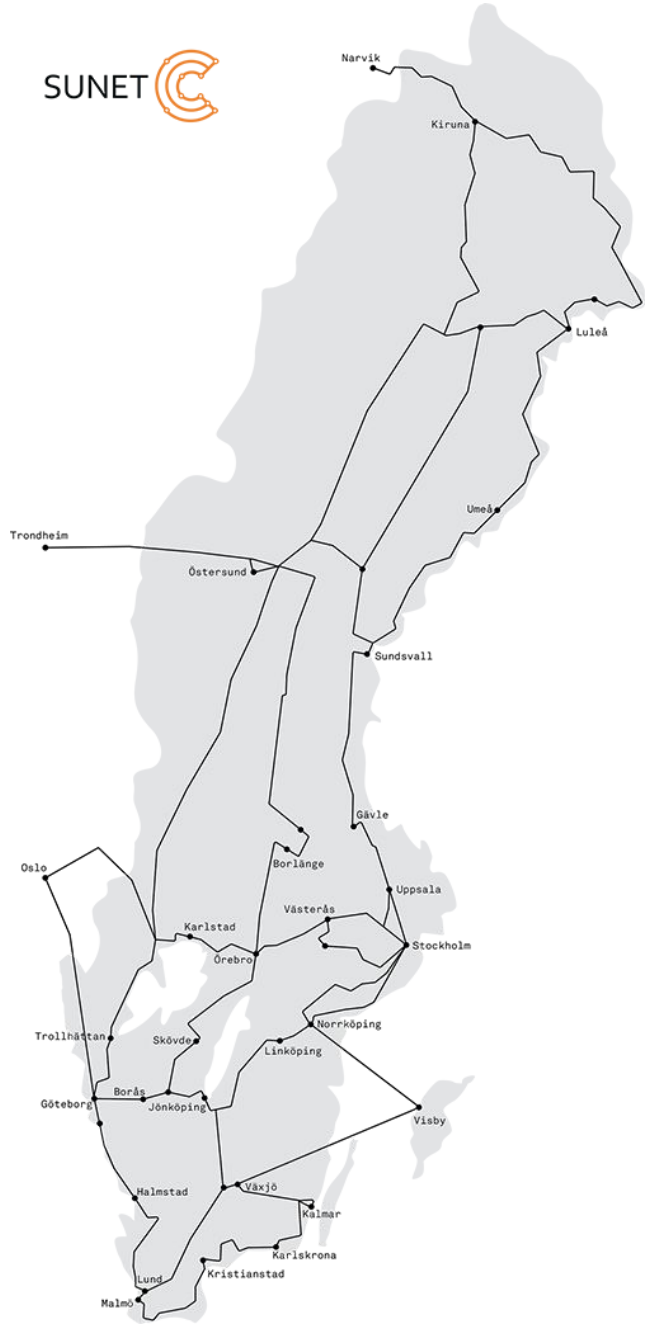
Magnus Bergroth, Dennis Wallberg, Olof Hagsand, Kristofer Hallin

(cd@sunet.se)

SUNET

Sunet Core Network

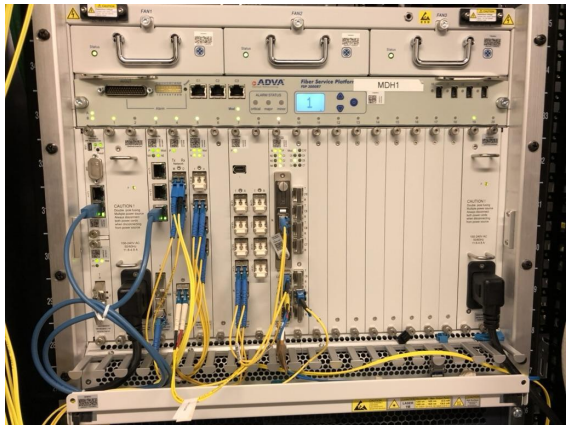
- ~115 connected organisations
- ~750 000 end users
- 50 sites
- 100+ routers
- Services:
  - Internet
  - IP VPN
  - L2 VPN
  - Peering
  - ...



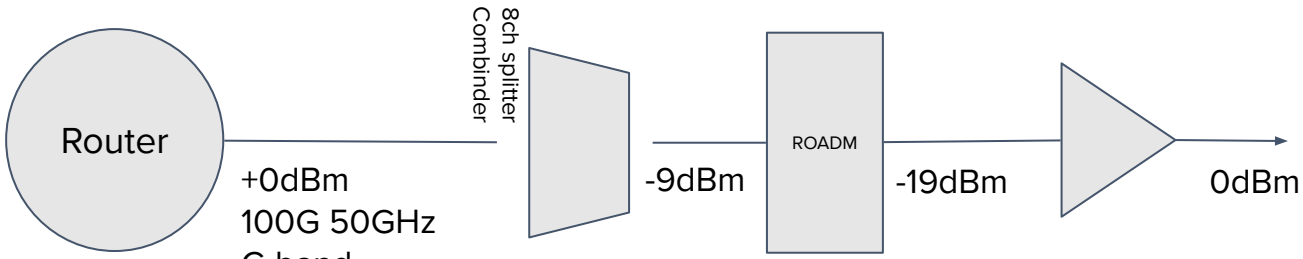
SUNET

Old equipment - Sunet

- 100 Gbps IP over DWDM



- 47 Universities, MX480
- 25 PoP, MX960
- 4 PoP, MX2010
- 1 PoP, MX2020
- 5 x MX10003
- 10 x MX204
- 15 x MX80



SUNET

Mission

- Upgrade the network from 100Gbps to 400Gbps
  - No more software support for DWDM optics
  - Old equipment, high maintenance cost
  - No way to increase bandwidth

- Upgrade existing or develop a new network controller
  - Expensive licensing model
  - Going from vendor lock-in and engineering lock-in to only engineering lock-in (yay)

SUNET

Mission

How hard can it be to NOT buy a network controller?

SUNET

Plan

- Keep the optical network
- Procurement for new routers and optics to be finished 2023
- Develop new network automation (started december 2022)
- Use data from old network as “source of truth” to create templates and generate configuration

SUNET

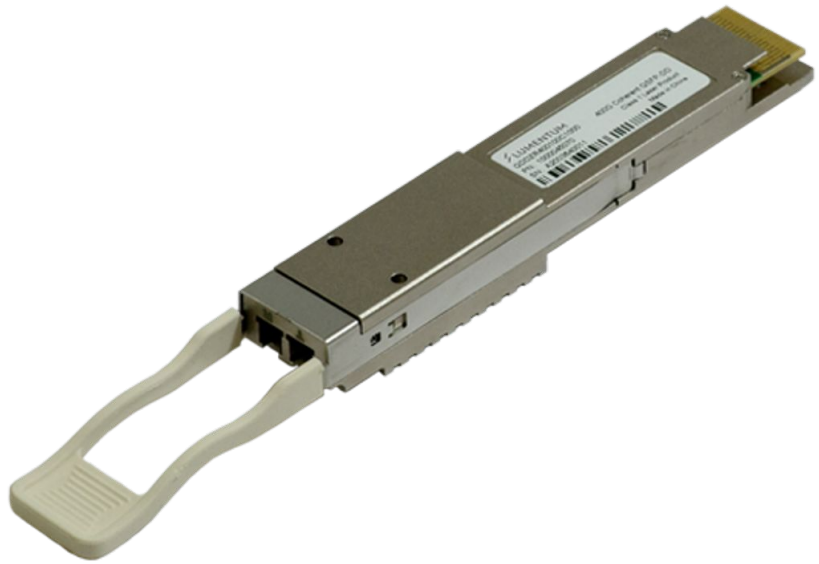
Plan - continued

- Deploy new routers and customers in parallel with old ones
- Bootstrap new routers with configuration templates
- Apply network services on new routers
- Decommission old routers once new routers are in production

SUNET

New equipment - Sunet CD

PTX10001-36MR36 network ports, 400GZR+ 0dBm



SUNET

Network Automation - SNC

- Development started December 2022

  •Based on existing open source “Clixon” project

- Based on NETCONF and YANG

  •Any device implementing RFC6241 + RFC 7950 “should” work (haha)

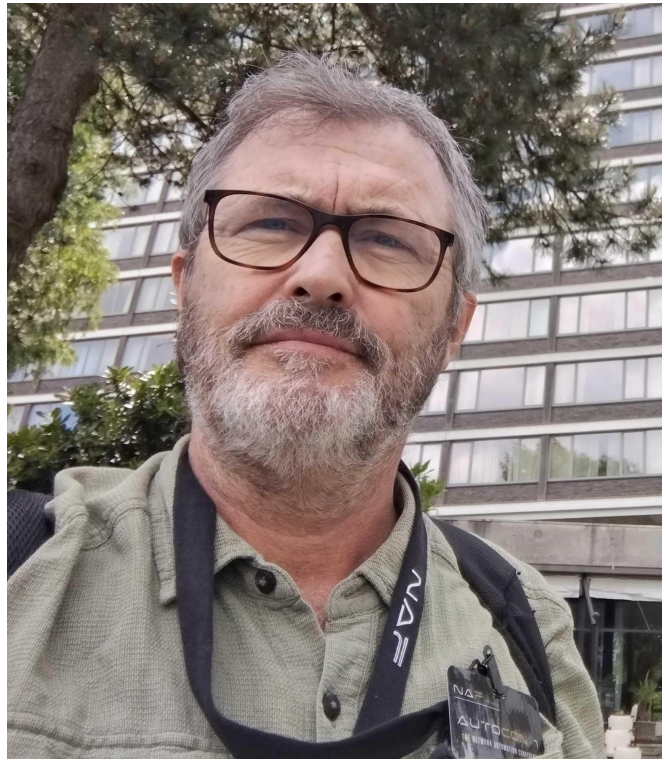
- Programmable services via a Python API

  •We developed services to handle BGP, user administration etc

- Support for multiple devices with different YANG schemas

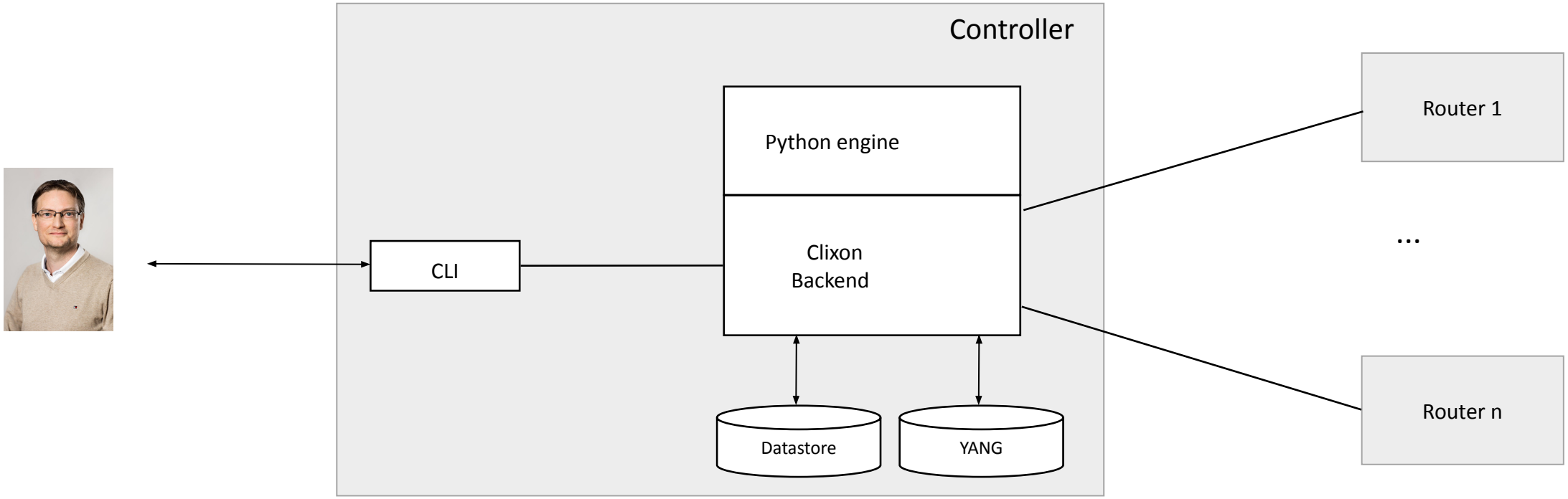
  •We have a mix of Juniper PTX, MX, QFX, Arista, Adtran etc

  •Possibility to add other vendors too



SUNET

SNC architecture

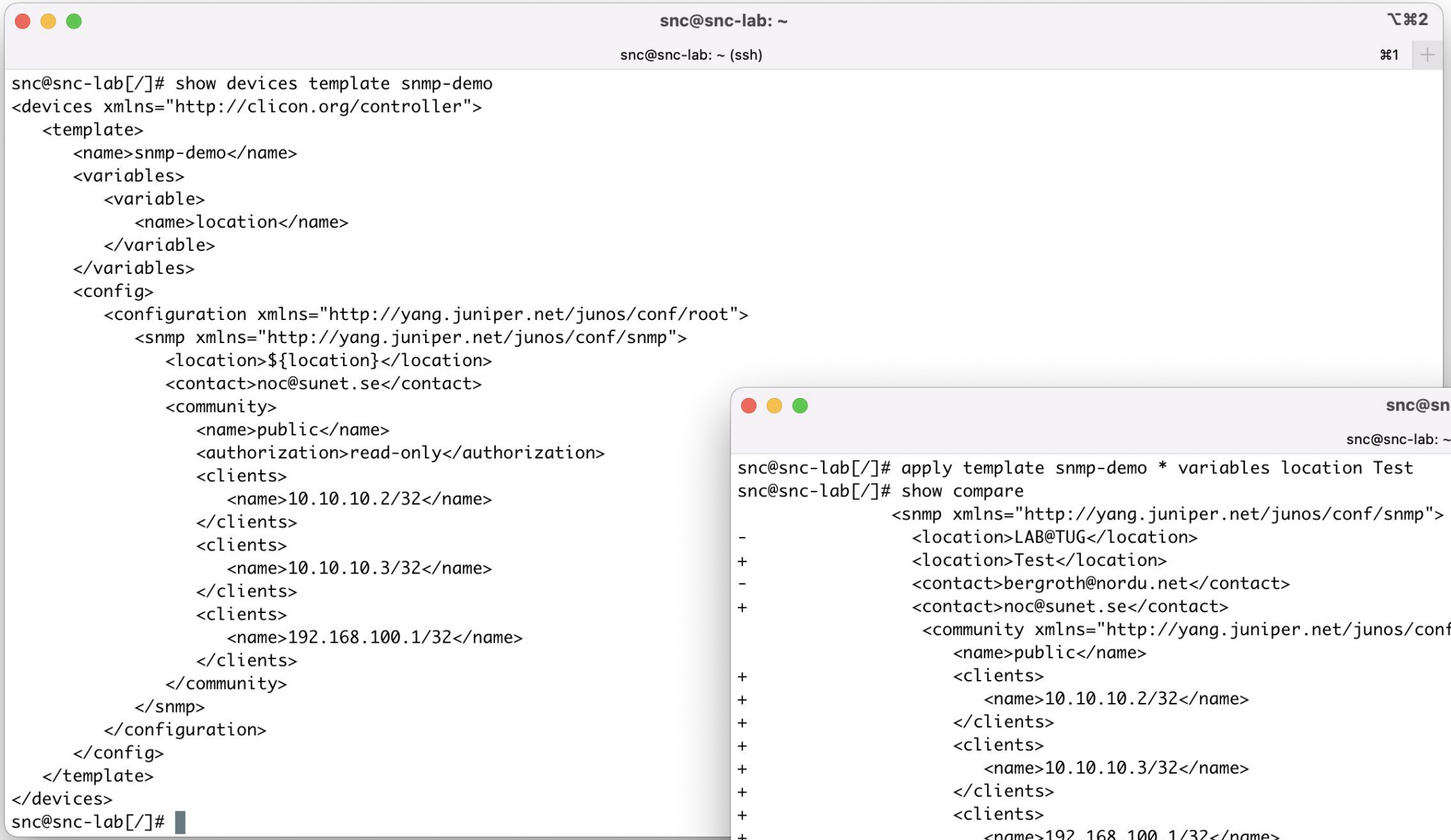


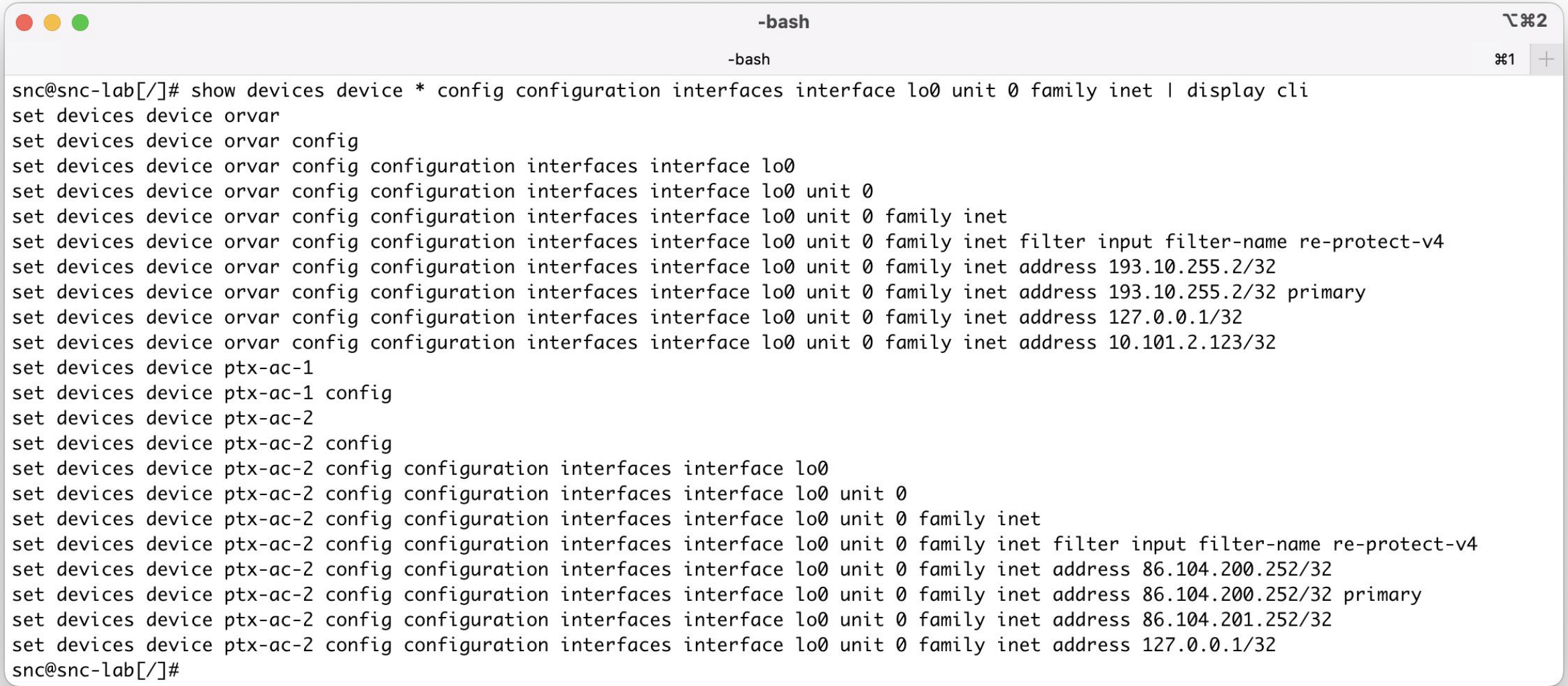
SUNET

SNC - Features

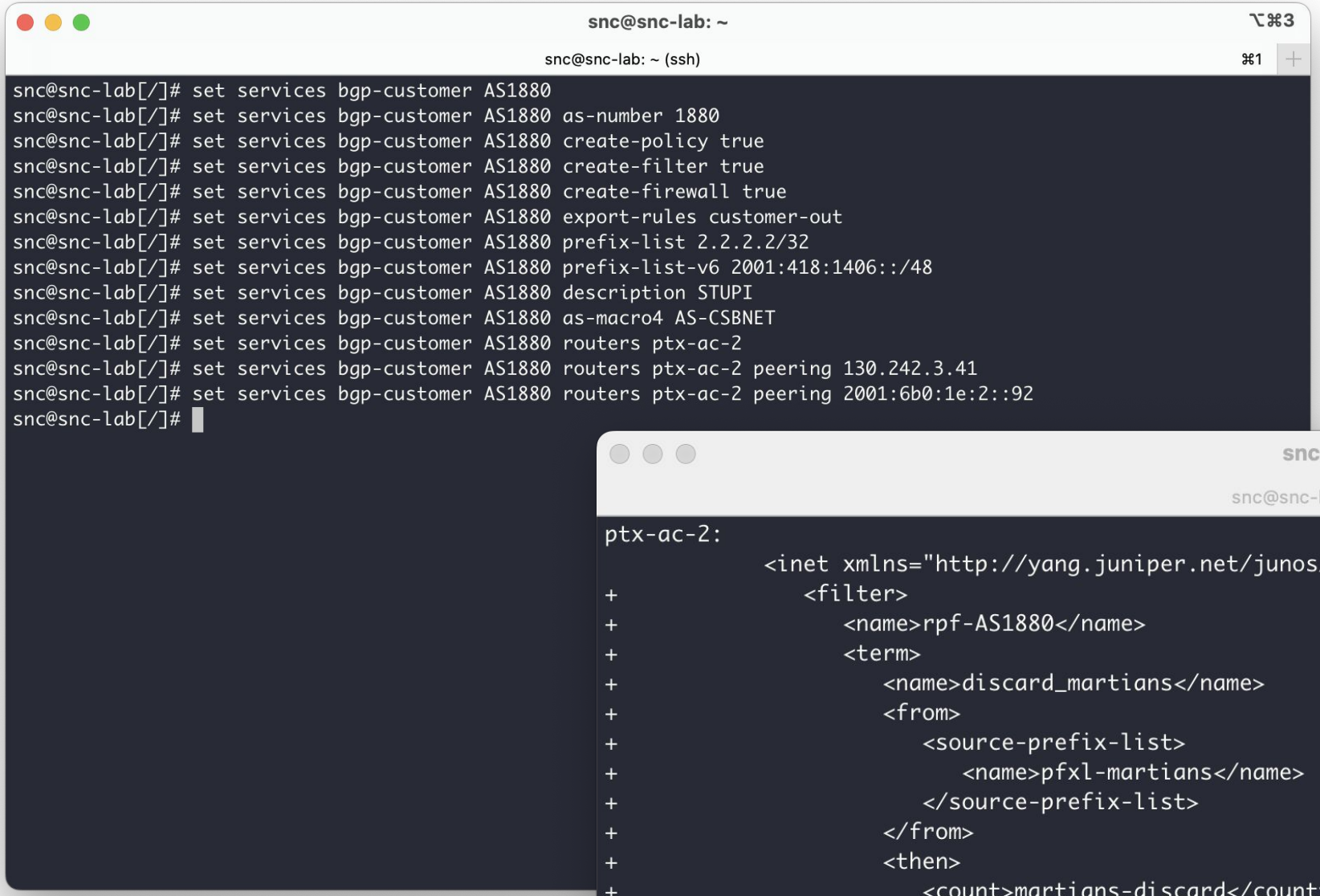
- Interactive CLI rendered from YANG models which looks like the devices CLI
- YANG models
- APIs: NETCONF, RESTCONF, SNMP, Python
- Datastore of device and controller configuration (XML)
- Python engine for network services
- Multi-vendor, multi-model devices
- Device push/pull
- Commit/transaction semantics across device-groups
- Edit, validate, commit
- Templates

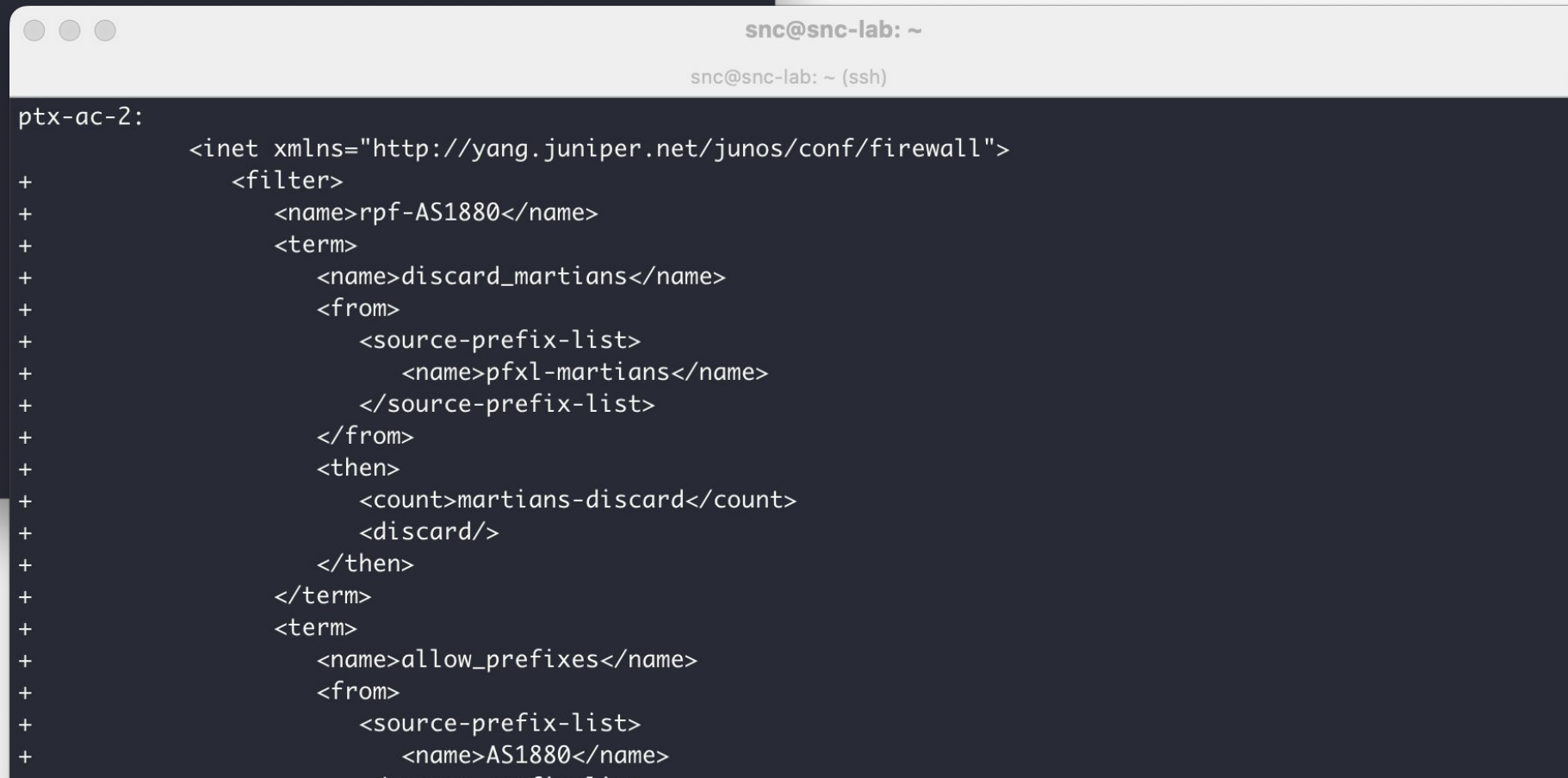
SUNET


snc@snc-lab: ~
⌥⌘2
snc@snc-lab: ~ (ssh)
⌘1
snc@snc-lab[/]# show devices template snmp-demo
<devices xmlns="http://clicon.org/controller">
<template>
<name>snmp-demo</name>
<variables>
<variable>
<name>location</name>
</variable>
</variables>
<config>
<configuration xmlns="http://yang.juniper.net/junos/conf/root">
<snmp xmlns="http://yang.juniper.net/junos/conf/snmp">
<location>${location}</location>
<contact>noc@sunet.se</contact>
<community>
<name>public</name>
<authorization>read-only</authorization>
<clients>
<name>10.10.10.2/32</name>
</clients>
<clients>
<name>10.10.10.3/32</name>
</clients>
<clients>
<name>192.168.100.1/32</name>
</clients>
</community>
</snmp>
</configuration>
</config>
</template>
</devices>
snc@snc-lab[/]#
snc@snc-lab: ~
snc@snc-lab: ~ (ssh)
snc@snc-lab[/]# apply template snmp-demo * variables location Test
snc@snc-lab[/]# show compare
<snmp xmlns="http://yang.juniper.net/junos/conf/snmp">
- <location>LAB@TUG</location>
+ <location>Test</location>
- <contact>bergroth@nordu.net</contact>
+ <contact>noc@sunet.se</contact>
<community xmlns="http://yang.juniper.net/junos/conf/snmp">
<name>public</name>
+ <clients>
+ <name>10.10.10.2/32</name>
+ </clients>
+ <clients>
+ <name>10.10.10.3/32</name>
+ </clients>
+ <clients>
+ <name>192.168.100.1/32</name>
+ </clients>
</community>
</snmp>


-bash
⌥⌘2
-bash
⌘1
snc@snc-lab[/]# show devices device * config configuration interfaces interface lo0 unit 0 family inet | display cli
set devices device orvar
set devices device orvar config
set devices device orvar config configuration interfaces interface lo0
set devices device orvar config configuration interfaces interface lo0 unit 0
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet filter input filter-name re-protect-v4
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet address 193.10.255.2/32
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet address 193.10.255.2/32 primary
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet address 127.0.0.1/32
set devices device orvar config configuration interfaces interface lo0 unit 0 family inet address 10.101.2.123/32
set devices device ptx-ac-1
set devices device ptx-ac-1 config
set devices device ptx-ac-2
set devices device ptx-ac-2 config
set devices device ptx-ac-2 config configuration interfaces interface lo0
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet filter input filter-name re-protect-v4
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet address 86.104.200.252/32
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet address 86.104.200.252/32 primary
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet address 86.104.201.252/32
set devices device ptx-ac-2 config configuration interfaces interface lo0 unit 0 family inet address 127.0.0.1/32
snc@snc-lab[/]#

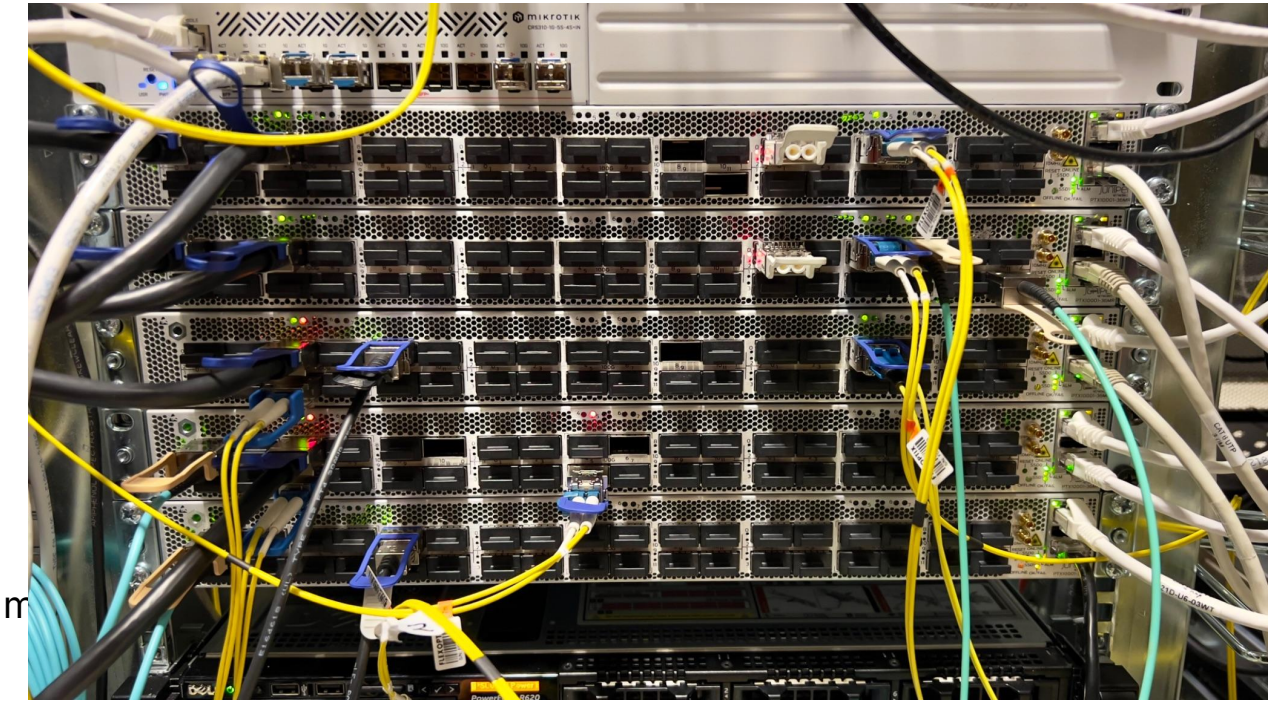
SUNET


snc@snc-lab: ~
⌥⌘3
snc@snc-lab: ~ (ssh)
⌘1
snc@snc-lab[/]# set services bgp-customer AS1880
snc@snc-lab[/]# set services bgp-customer AS1880 as-number 1880
snc@snc-lab[/]# set services bgp-customer AS1880 create-policy true
snc@snc-lab[/]# set services bgp-customer AS1880 create-filter true
snc@snc-lab[/]# set services bgp-customer AS1880 create-firewall true
snc@snc-lab[/]# set services bgp-customer AS1880 export-rules customer-out
snc@snc-lab[/]# set services bgp-customer AS1880 prefix-list 2.2.2.2/32
snc@snc-lab[/]# set services bgp-customer AS1880 prefix-list-v6 2001:418:1406::/48
snc@snc-lab[/]# set services bgp-customer AS1880 description STUPI
snc@snc-lab[/]# set services bgp-customer AS1880 as-macro4 AS-CSBNET
snc@snc-lab[/]# set services bgp-customer AS1880 routers ptx-ac-2
snc@snc-lab[/]# set services bgp-customer AS1880 routers ptx-ac-2 peering 130.242.3.41
snc@snc-lab[/]# set services bgp-customer AS1880 routers ptx-ac-2 peering 2001:6b0:1e:2::92
snc@snc-lab[/]#


snc@snc-lab: ~
snc@snc-lab: ~ (ssh)
ptx-ac-2:
<inet xmlns="http://yang.juniper.net/junos/conf/firewall">
+ <filter>
+ <name>rpf-AS1880</name>
+ <term>
+ <name>discard_martians</name>
+ <from>
+ <source-prefix-list>
+ <name>pfxl-martians</name>
+ </source-prefix-list>
+ </from>
+ <then>
+ <count>martians-discard</count>
+ <discard/>
+ </then>
+ </term>
+ <term>
+ <name>allow_prefixes</name>
+ <from>
+ <source-prefix-list>
+ <name>AS1880</name>

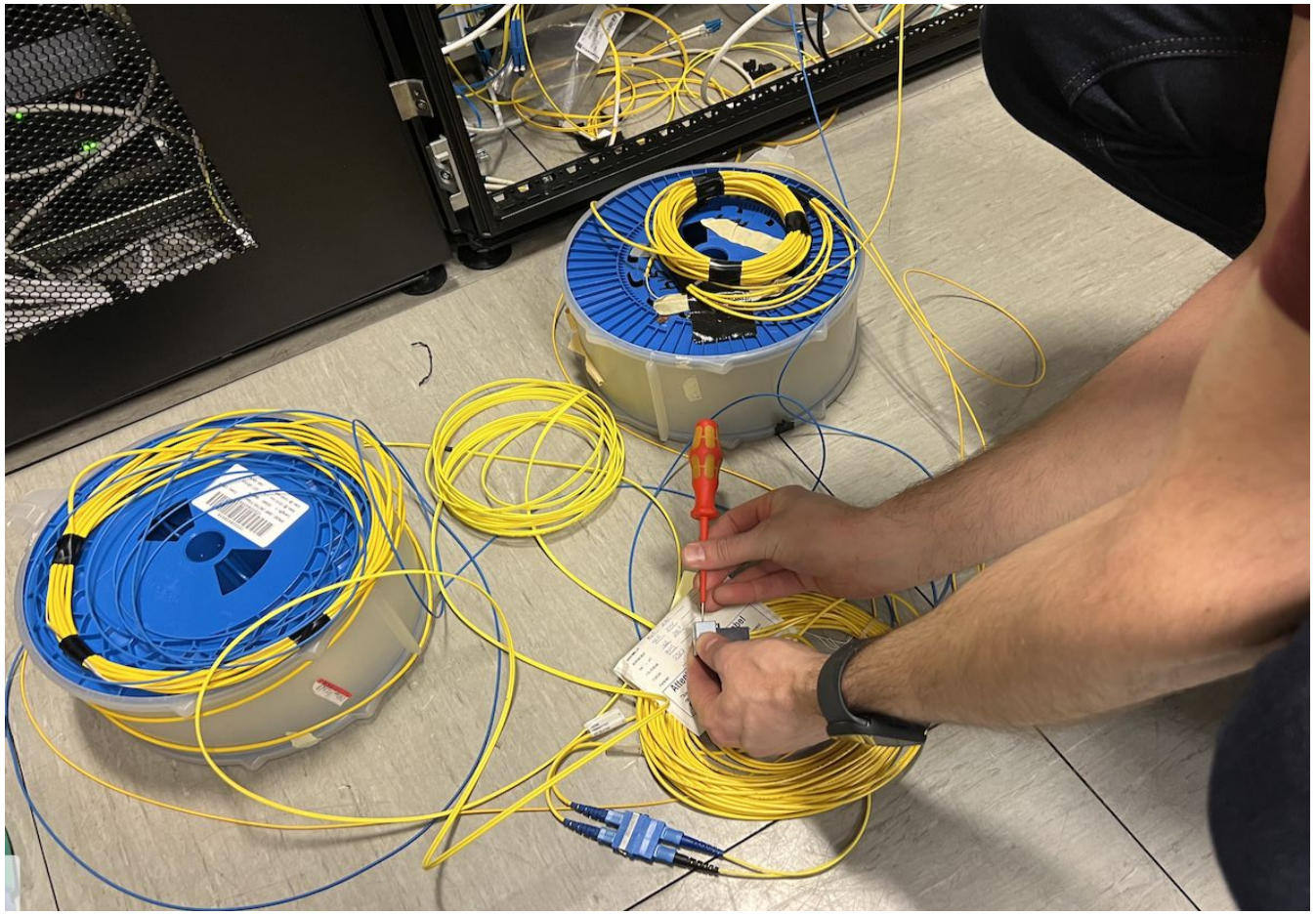
Tests

- Initial tests of hardware
  - •QSFP testing (Optics? 400G ZR+ HP QSFP-DD)
  - •Ixia
  - •POC at Juniper
- Experimenting with YANG and NETCONF
  - •Using all equipment we could get hold of.
  - •Juniper MX80, Arista switches
  - •Software (cRPD, vEVO, cEOS, OpenConfig etc)
- Problems with software based switches/routers
  - •Mismatching YANG models
  - •Broken NETCONF implementations
  - •Incomplete OpenConfig support
- Ended up using real hardware, 5*PTX, QFX, MX80 and m



SUNET

Tests



SUNET

Network roll out – three phases

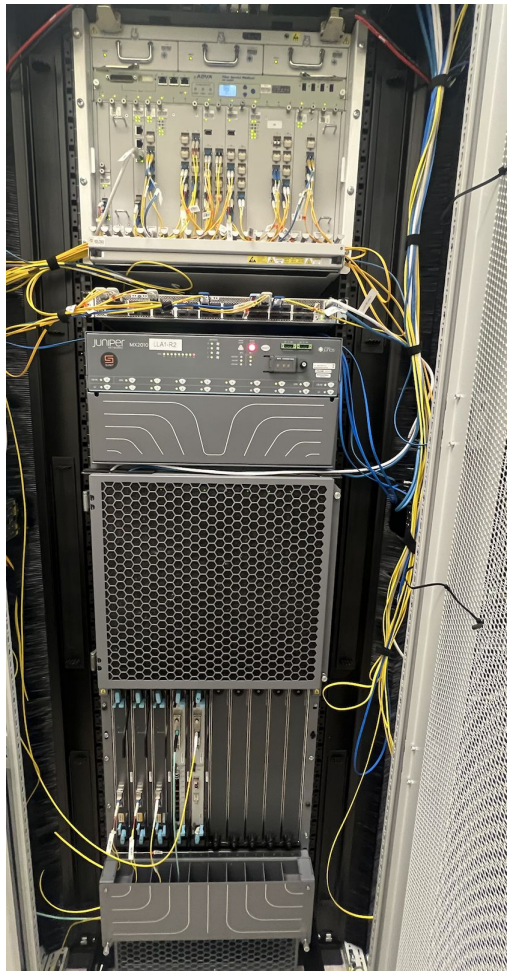
- **Installation:** routers are mounted and bo
- **Deployment:** services are applied for BG peering etc
- **Migration:** customer traffic moved from routers to new

Installation

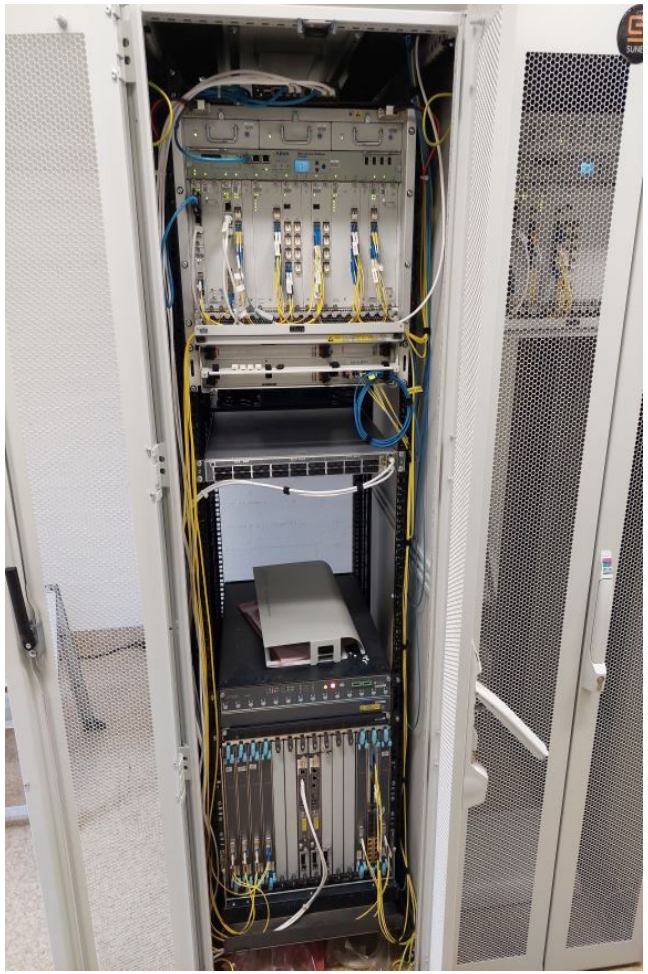
- Delivery on December 23
- Early Christmas gift
- 1620 kg
- Stairs

Installation

Sunet C



Sunet CD (lots of empty space)



SUNET

Deployment - initial configuration

- **JSON file (source of truth)**
  - ~6000 lines with node-names, addresses, metrics etc
  - Created with data extracted from Sunet C
- **Python script**
  - To deploy static configuration using templates
  - Initially over 4G and later changed to the real management address
- **Templates**
  - ~50 templates
  - Static configuration, configured once and forgotten
  - System, policies, interfaces etc
- **5000+ lines of configuration**
  - Deployed on each device with SNC

SUNET


```
{
  "hostname": "tug-r11",
  "type": "PTX10001-36MR",
  "IP": "86.104.999.99",
  "IP-v6": "2001:6b0:XXXX::YY",
  "NH_IP": "86.104.999.99",
  "NH_IP-v6": "2001:6b0:9999::99",
  "ISO": "47.0000.0000.0000.00",
  "SR_node_sid": 1023,
  "SR_node_sid-v6": 2023,
  "location": "Stockholm",
  "pop": "TUG",
  "interfaces": {
    "et-0/0/0": {
      "inet": "86.104.999.999/31",
      "inet6": "2001:6b0:2000:0000::0/127",
      "service_ID": "SU-S004068",
      "destination": "fsn2-r11",
      "destination_port": "et-0/2/0",
      "metric": 1200,
      "wavelength": "1554.84",
      "speed": "400g",
      "plug": "400GZR+",
      "dwdm_port": "EOM-3-FCU-I2-C5",
      "dwdm_node": "TUG",
      "dwdm_target": "FSN2"
    }
},
```

```
deploy-chassis.xml
deploy-class-of-service.xml
deploy-fan-speed.xml
deploy-firewall.xml
deploy-forwarding-options.xml
deploy-groups-eth.xml
deploy-groups-isis.xml
deploy-groups-sunet-aggregates.xml
deploy-interfaces-core.xml
deploy-interfaces-loopback.xml
deploy-policy-options-shared.xml
deploy-protocols-isis.xml
deploy-protocols-ldp.xml
deploy-protocols-mpls.xml
deploy-protocols-pim.xml
deploy-routing-options.xml
deploy-security.xml
deploy-services.xml
deploy-snmp.xml
deploy-system.xml
…
```

```
dennis@snc[/]# show devices template deploy-chassis
<devices xmlns="http://clicon.org/controller">
   <template>
      <name>deploy-chassis</name>
      <config>
         <configuration
xmlns="http://yang.juniper.net/junos/conf/root">
            <chassis
xmlns="http://yang.juniper.net/junos/conf/chassis">
               <aggregated-devices>
                  <ethernet>
                     <device-count>10</device-count>
                  </ethernet>
               </aggregated-devices>
               <fpc>
                  <name>0</name>
                  <sampling-instance>
                     <name>IPFIX</name>
                  </sampling-instance>
               </fpc>
               <alarm>
                  <management-ethernet>
                     <link-down>ignore</link-down>
                  </management-ethernet>
               </alarm>
…
```

SUNET

Deployment example

```
usage: deploy_cd.py [-h] [-f FILENAME] [-l] [-a] [-d] [-v] [-n] [-i]
                    [-c {all,firewall,interface,templates}]
                    [router]

Read routers from json and them to SNC network controller

positional arguments:
  router                router name

options:
  -h, --help            show this help message and exit
  -f FILENAME, --filename FILENAME
                        json file with router data, defaults to
                        SunetCD_routers.json
  -l, --list            list available routers
  -a, --add             add device to SNC
  -d, --deploy          deploy templates
  -v, --verbosity       increase output verbosity
  -n, --nothing         do nothing just print
  -i, --interface       only apply interfaces
  -c {all,firewall,interface,templates}, --clear {all,firewall,interface,templates}
                        delete existing config to be applied with templates
```

```
dennis@snc:~$ deploy_cd.py -d uu-r21
##########################################

 Using file /usr/local/bin/SunetCD_routers.json

##########################################

output: <!-- uu-r21: -->
<devices xmlns="http://clicon.org/controller">
   <device>
      <name>uu-r21</name>
      <addr>uu-r21.sunet.se</addr>
   </device>
</devices>

pull config / sync device
OK
output:

-- Applying all generic templates for uu-r21 --

applying deploy-security                          31/31  [##################################]  100%

-- Done applying generic templates for uu-r21 --


-- Applying device specific templates for uu-r21 --
```

SUNET

Migration

- Apply configuration with services
  - Add users, iBGP, peering, filters etc
- Move connectors from old router to new
- Move traffic from old router to new

SUNET

Lessons learned

- Memory leak in router software, resulted in crashes
- Incomplete YANG models: MACSec and temperature sensors lacking models etc
- Unstable connections over 4G: Avoid using jumphosts
- YANG models from the same vendor with the same name and revision might be different
- Large and complex YANG models results in huge memory footprint
- Transactions with device rollback etc was never a problem
- Optimising code is hard, some things still takes too long time
- Implementing services is complex
- Explaining the services model for others takes time and effort

SUNET

Remaining work

- Migrate NORDUnet to SNC
- More services
- Integration with internal systems
- Decommission old routers

More information

- https://sunet.se/snc
- https://sunet.se/om-sunet/sunets-nat
- https://github.com/clicon/clixon-controller

SUNET