



zebra



DEVNET

Partner University Deutschland

Network engineer 2.0 skills
deep dive into open source
libraries and neops.io



zebra

we **make** things **happen**



DEVNET

Partner University Deutschland

Network engineer 2.0 skills
deep dive into open source
libraries and neops.io



Simon Obi

simon.obi@zebbra.ch

Network Digitalization Officer
@ zebbra AG

10 years experience as lead network engineer @ ISP

NetOps software projects (scripts, monitoring solutions, ansible playbooks, config generator and more)

lead dev @zebbra for [neops.io](#), full-stack network management solution based on Python, Django, Celery, nornir and Vue.JS

zebra is a collective of agile



we ideate & build surprisingly

zebra is a collective of agile
Thinkers



we ideate & build surprisingly

zebra is a collective of agile
Thinkers **Doers**



we ideate & build surprisingly

zebra is a collective of agile
Thinkers **Doers**



we ideate & build surprisingly

simple

pragmatic



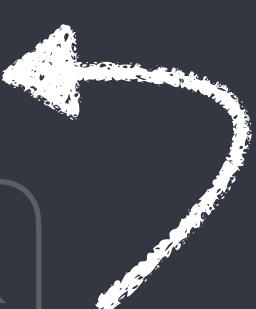
zebra is a collective of agile
Thinkers **Doers**



Solutions
we ideate & build surprisingly

simple

pragmatic



small team,
big impact 🤘



Marco Stieger
Co-CEO



Markus Roth
Co-CEO / Dev



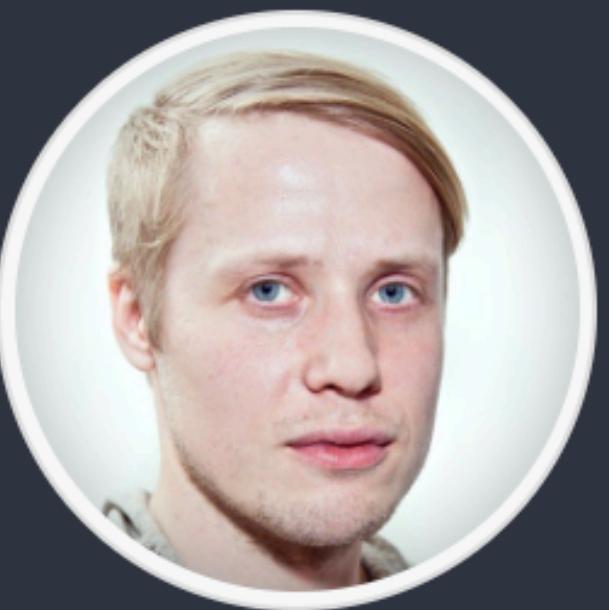
Roland Mamie
Network Guru



Mike Steinmann
DevOps



Franziska Obbens
HR & Admin



Marek Paulson
Dev / UC



Michael Bärtschi
Dev



Marcel Wälti
Digital Innovation / UX



Hannes Wüthrich
Dev / DevOps



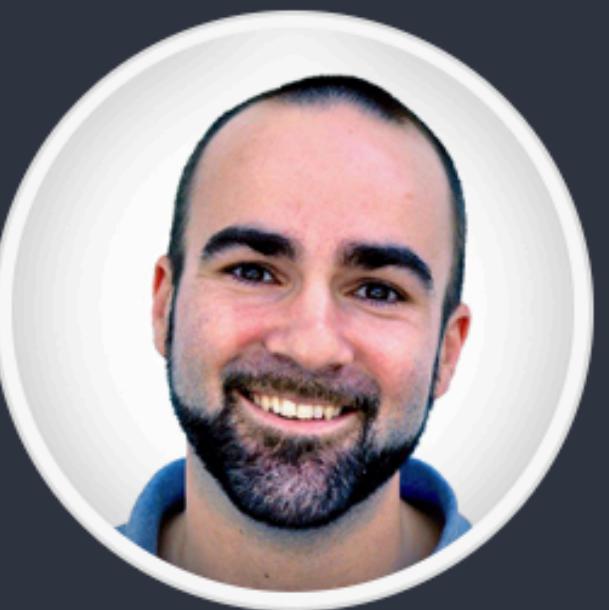
Claudio Siegenthaler
Dev / DevOps



Michael Aschwanden
Network



Alex Tull
Network / UC / Dev



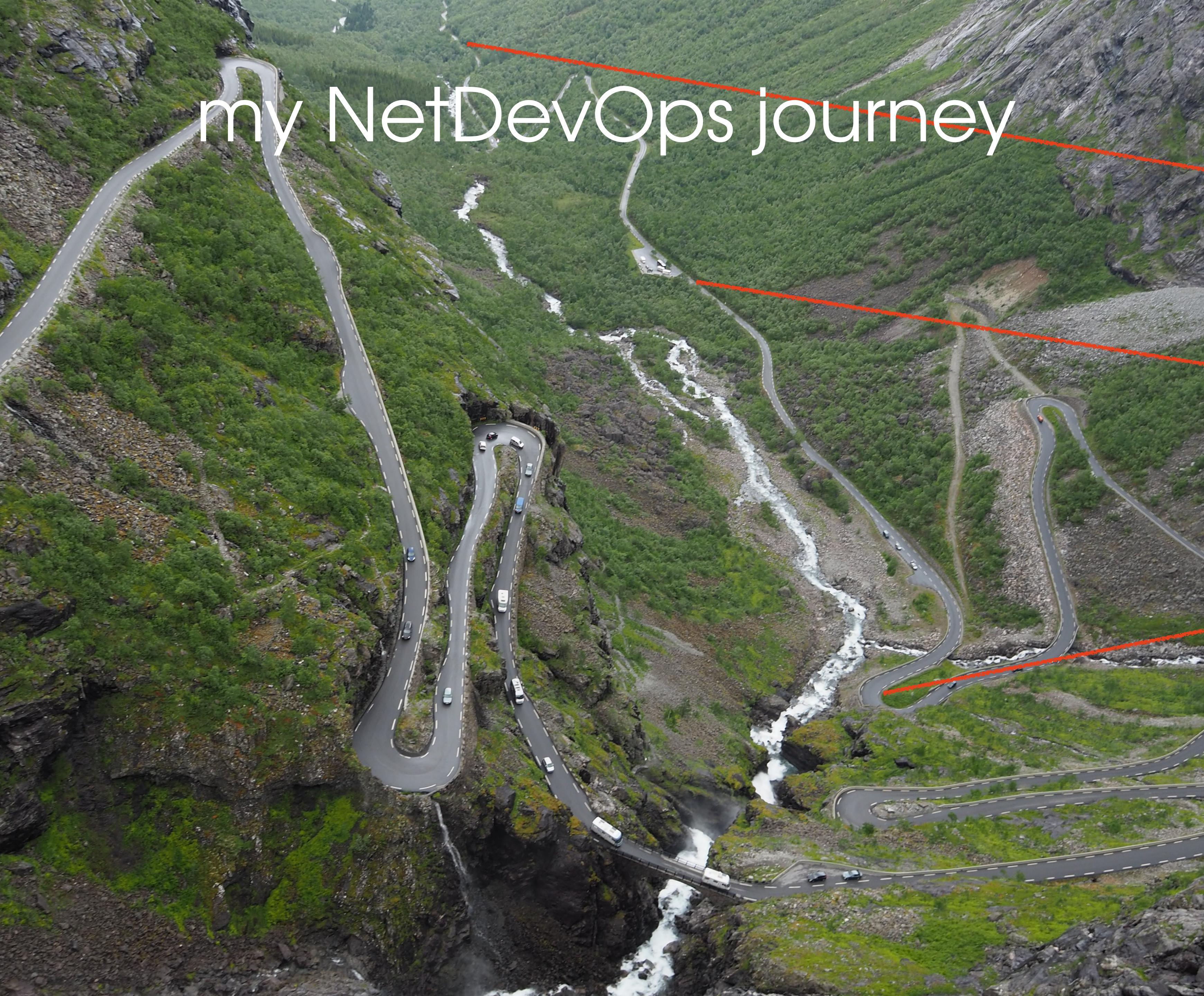
Remo Ryter
PO / PM / DevOps



Simon Obi
Network / Dev



Matthias Schärer
Network



my NetDevOps journey

- 🚧 2003 !/usr/bin/expect
- 🚧 2009 monitoring software based on C# and SNMP
- 🚧 2014 config generator based on apache velocity and angular



Agenda

Agenda

■ Toolset

■ Backend libraries

- netmiko

- nornir

- testFSM

- noteworthy libraries

- CLI and NaC/IaC

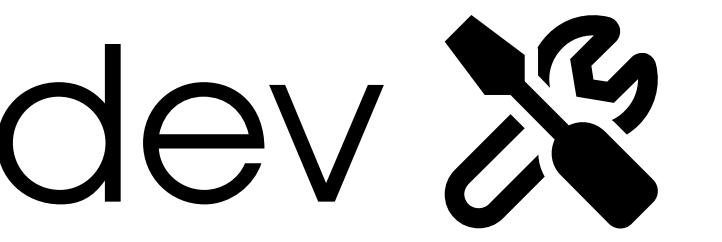
■ Why UX matters and frontend libraries

- json schema form

■ Putting it together - [neops.io](#)



Tools



■ Use proper IDE

- Syntax highlighting

- Linting

■ Testing during dev process

- Dedicated LAB

- Cisco VIRL

- GNS3



■ Code version tracking



git

■ Pipelines CI/CD



■ Automated testing

■ Application virtualisation/Virtual Environment



docker

Takeaways

- ⌘ You got to learn a bit but then...
- ⌘ good tooling saves you **lots** of time

... continue

■ 2015/2016 python/
netmiko based
change scripts



netmiko

Some years ago

- Started to automate simple tasks by scripts
- netmiko is based on paramiko (ssh connection library)
- It's the de facto standard for cli based connections
- <https://github.com/ktbyers/netmiko>

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.123.21',
    'username': 'neops',
    'password': 'cisco',
}
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {  
    'device_type': 'cisco_ios',  
    'ip': '192.168.123.21',  
    'username': 'neops',  
    'password': 'cisco',  
}  
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.123.21',
    'username': 'neops',
    'password': 'cisco',
}
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {  
    'device_type': 'cisco_ios',  
    'ip': '192.168.123.21',  
    'username': 'neops',  
    'password': 'cisco',  
}  
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.123.21',
    'username': 'neops',
    'password': 'cisco',
}
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.123.21',
    'username': 'neops',
    'password': 'cisco',
}
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Initiate connection

```
from netmiko import ConnectHandler
```

```
my_device = {
    'device_type': 'cisco_ios',
    'ip': '192.168.123.21',
    'username': 'neops',
    'password': 'cisco',
}
my_device.update({'global_delay_factor': 2}) #on slow devices
```

```
my_device_session = ConnectHandler(**my_device)
```

netmiko - Execute command

```
print(my_device_session.send_command('show int desc'))
```

Interface	Status	Protocol	Description
Gi0/0	up	up	dist-sw01@gi1/0
Gi0/1	down	down	
Gi0/2	down	down	
Gi0/3	down	down	
Gi1/0	down	down	
Gi1/1	down	down	
Gi1/2	down	down	
Gi1/3	down	down	
Gi2/0	up	up	CLIENT-A
Gi2/1	down	down	
Gi2/2	down	down	
Gi2/3	down	down	
Gi3/0	down	down	s=u,sla=s,r=,d=weu,u=,m
=s,o=,t=			
Gi3/1	down	down	s=u,sla=s,r=,d=dij,u=,m
=s,o=,t=			
Gi3/2	down	down	
Gi3/3	down	down	TEST
Vl10	up	up	MGMT

netmiko - Execute command

```
print(my_device_session.send_command('show int desc'))
```

Interface	Status	Protocol	Description
Gi0/0	up	up	dist-sw01@gi1/0
Gi0/1	down	down	
Gi0/2	down	down	
Gi0/3	down	down	
Gi1/0	down	down	
Gi1/1	down	down	
Gi1/2	down	down	
Gi1/3	down	down	
Gi2/0	up	up	CLIENT-A
Gi2/1	down	down	
Gi2/2	down	down	
Gi2/3	down	down	
Gi3/0	down	down	s=u,sla=s,r=,d=weu,u=,m
=s,o=,t=			
Gi3/1	down	down	s=u,sla=s,r=,d=dij,u=,m
=s,o=,t=			
Gi3/2	down	down	
Gi3/3	down	down	TEST
Vl10	up	up	MGMT

netmiko - Execute command

```
print(my_device_session.send_command('show int desc'))
```

Interface	Status	Protocol	Description
Gi0/0	up	up	dist-sw01@gi1/0
Gi0/1	down	down	
Gi0/2	down	down	
Gi0/3	down	down	
Gi1/0	down	down	
Gi1/1	down	down	
Gi1/2	down	down	
Gi1/3	down	down	
Gi2/0	up	up	CLIENT-A
Gi2/1	down	down	
Gi2/2	down	down	
Gi2/3	down	down	
Gi3/0	down	down	s=u,sla=s,r=,d=weu,u=,m
=s,o=,t=			
Gi3/1	down	down	s=u,sla=s,r=,d=dij,u=,m
=s,o=,t=			
Gi3/2	down	down	
Gi3/3	down	down	TEST
Vl10	up	up	MGMT

netmiko - Execute command

```
print(my_device_session.send_command('reload'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-5-306c2278f1e3> in <module>
----> 1 print(my_device_session.send_command('reload'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/base_connection.py in send_command(self, command_string, expect_string, del
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz
e, use_textfsm, textfsm_template, use_genie, cmd_verify)
    1427         new_data = self.read_channel()
    1428     else: # nobreak
-> 1429         raise IOError(
    1430             "Search pattern never detected in send_command_expect:
{}".format(
    1431                 search_pattern

OSError: Search pattern never detected in send_command_expect: sw01\#
```

netmiko - Execute command

```
print(my_device_session.send_command('reload'))
```

```
-----  
OSError                                     Traceback (most recent call last)  
<ipython-input-5-306c2278f1e3> in <module>  
----> 1 print(my_device_session.send_command('reload'))  
  
~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi  
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)  
    345         if self.global_cmd_verify is not None:  
    346             kwargs["cmd_verify"] = self.global_cmd_verify  
--> 347         return func(self, *args, **kwargs)  
    348  
    349     return wrapper_decorator  
  
~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi  
ko/base_connection.py in send_command(self, command_string, expect_string, del  
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz  
e, use_textfsm, textfsm_template, use_genie, cmd_verify)  
    1427         new_data = self.read_channel()  
    1428     else: # nobreak  
-> 1429         raise IOError(  
    1430             "Search pattern never detected in send_command_expect:  
{}".format(  
    1431                 search_pattern  
  
OSError: Search pattern never detected in send_command_expect: sw01\#
```

netmiko - Execute command

```
print(my_device_session.send_command('reload'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-5-306c2278f1e3> in <module>
----> 1 print(my_device_session.send_command('reload'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/base_connection.py in send_command(self, command_string, expect_string, del
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz
e, use_textfsm, textfsm_template, use_genie, cmd_verify)
    1427         new_data = self.read_channel()
    1428     else: # nobreak
-> 1429         raise IOError(
    1430             "Search pattern never detected in send_command_expect:
{}".format(
    1431                 search_pattern

OSError: Search pattern never detected in send_command_expect: sw01\#
```

netmiko - Execute command

```
print(my_device_session.send_command('reload'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-5-306c2278f1e3> in <module>
----> 1 print(my_device_session.send_command('reload'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/base_connection.py in send_command(self, command_string, expect_string, del
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz
e, use_textfsm, textfsm_template, use_genie, cmd_verify)
    1427         new_data = self.read_channel()
    1428     else: # nobreak
-> 1429         raise IOError(
    1430             "Search pattern never detected in send_command_expect:
{}".format(
    1431                 search_pattern
```

OSError: Search pattern never detected in send_command_expect: sw01\#

netmiko - Execute command

```
print(my_device_session.send_command('reload'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-5-306c2278f1e3> in <module>
----> 1 print(my_device_session.send_command('reload'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/base_connection.py in send_command(self, command_string, expect_string, del
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz
e, use_textfsm, textfsm_template, use_genie, cmd_verify)
    1427         new_data = self.read_channel()
    1428     else: # nobreak
-> 1429         raise IOError(
    1430             "Search pattern never detected in send_command_expect:
{}".format(
    1431                 search_pattern

OSError: Search pattern never detected in send_command_expect: sw01\#
```

netmiko - Execute command

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

netmiko - Execute command

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

netmiko - Execute command

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

netmiko - Enable logging

```
import io  
my_log = io.BytesIO()  
  
my_device.update({'session_log': my_log})  
  
my_device_session = ConnectHandler(**my_device)
```

netmiko - Enable logging

```
import io  
my_log = io.BytesIO()  
  
my_device.update({'session_log': my_log})  
  
my_device_session = ConnectHandler(**my_device)
```

netmiko - Enable logging

```
import io  
my_log = io.BytesIO()  
  
my_device.update({'session_log': my_log})  
  
my_device_session = ConnectHandler(**my_device)
```

netmiko - Enable logging

```
import io  
my_log = io.BytesIO()  
  
my_device.update({'session_log': my_log})  
  
my_device_session = ConnectHandler(**my_device)
```

netmiko - Enable logging

```
import io  
my_log = io.BytesIO()  
  
my_device.update({'session_log': my_log})  
  
my_device_session = ConnectHandler(**my_device)
```

netmiko - Enable logging

```
print(my_device_session.send_command('reload'))
```

```
OSError                                     Traceback (most recent call last)
<ipython-input-11-306c2278f1e3> in <module>
----> 1 print(my_device_session.send_command('reload'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/base_connection.py in send_command(self, command_string, expect_string, del
ay_factor, max_loops, auto_find_prompt, strip_prompt, strip_command, normaliz
e, use_textfsm, textfsm_template, use_genie, cmd_verify)
    1427             new_data = self.read_channel()
    1428         else: # nobreak
-> 1429             raise IOError(
    1430                 "Search pattern never detected in send_command_expect:
{}".format(
    1431                     search_pattern

OSError: Search pattern never detected in send_command_expect: sw01\#
```

netmiko - Enable logging

```
print(my_log.getvalue().decode())
```

```
C*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part,   *
* of the IOSv Software or Documentation to any third party for any        *
* purposes is expressly prohibited except as otherwise authorized by      *
* Cisco in writing.                                                       *
*****
```

```
sw01#
sw01#terminal length 0
sw01#terminal width 511
sw01#
sw01#reload
Proceed with reload? [confirm]
```

netmiko - Enable logging

```
print(my_log.getvalue().decode())
```

```
C*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part,   *
* of the IOSv Software or Documentation to any third party for any        *
* purposes is expressly prohibited except as otherwise authorized by      *
* Cisco in writing.                                                       *
*****  
  
sw01#
sw01#terminal length 0
sw01#terminal width 511
sw01#
sw01#reload
Proceed with reload? [confirm]
```

netmiko - Enable logging

```
print(my_log.getvalue().decode())
```

```
C*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part,   *
* of the IOSv Software or Documentation to any third party for any        *
* purposes is expressly prohibited except as otherwise authorized by      *
* Cisco in writing.                                                       *
*****
```

```
sw01#
sw01#terminal length 0
sw01#terminal width 511
sw01#
sw01#reload
Proceed with reload? [confirm]
```

netmiko - Find prompt

```
print(my_device_session.send_command('show version'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-13-f7a84e3b82de> in <module>
----> 1 print(my_device_session.send_command('show version'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/param
  1196             # this doesn't seem useful, but it is the documented b
  1197             # of Socket
-> 1198             raise socket.error("Socket is closed")
  1199             size = self._wait_for_send_window(size)
  1200             if size == 0:

OSError: Socket is closed
```

netmiko - Find prompt

```
print(my_device_session.send_command('show version'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-13-f7a84e3b82de> in <module>
----> 1 print(my_device_session.send_command('show version'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/param
  1196             # this doesn't seem useful, but it is the documented b
  1197             # of Socket
-> 1198             raise socket.error("Socket is closed")
  1199             size = self._wait_for_send_window(size)
  1200             if size == 0:

OSError: Socket is closed
```

netmiko - Find prompt

```
print(my_device_session.send_command('show version'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-13-f7a84e3b82de> in <module>
----> 1 print(my_device_session.send_command('show version'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/param
  1196             # this doesn't seem useful, but it is the documented b
  1197             # of Socket
-> 1198             raise socket.error("Socket is closed")
  1199             size = self._wait_for_send_window(size)
  1200             if size == 0:

OSError: Socket is closed
```

netmiko - Find prompt

```
print(my_device_session.send_command('show version'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-13-f7a84e3b82de> in <module>
----> 1 print(my_device_session.send_command('show version'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/param
  1196             # this doesn't seem useful, but it is the documented b
  1197             # of Socket
-> 1198             raise socket.error("Socket is closed")
  1199             size = self._wait_for_send_window(size)
  1200             if size == 0:
```

OSError: Socket is closed

netmiko - Find prompt

```
print(my_device_session.send_command('show version'))
```

```
-----
OSError                                     Traceback (most recent call last)
<ipython-input-13-f7a84e3b82de> in <module>
----> 1 print(my_device_session.send_command('show version'))

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/netmi
ko/utilities.py in wrapper_decorator(self, *args, **kwargs)
    345         if self.global_cmd_verify is not None:
    346             kwargs["cmd_verify"] = self.global_cmd_verify
--> 347         return func(self, *args, **kwargs)
    348
    349     return wrapper_decorator

~/local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/param
  1196             # this doesn't seem useful, but it is the documented b
  1197             # of Socket
-> 1198             raise socket.error("Socket is closed")
  1199             size = self._wait_for_send_window(size)
  1200             if size == 0:

OSError: Socket is closed
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359     # Find the current router prompt
1360     if expect_string is None:
1361         if auto_find_prompt:
1362             try:
1363                 prompt = self.find_prompt(delay_factor=delay_factor)
1364             except ValueError:
1365                 prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368             search_pattern = re.escape(prompt.strip())
1369         else:
1370             search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         ...
1301         auto_find_prompt=True,
1302         strip_prompt=True,
1303         strip_command=True,
1304         normalize=True,
1305         use_textfsm=False,
1306         textfsm_template=None,
1307         use_genie=False,
1308         cmd_verify=True,
1309     ):
1310         """Execute command_string on the SSH channel using a pattern
1311         used for show commands. By default this method will keep waiting
1312         until the network device prompt is detected. The current network device
1313         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     ...
1364                     prompt = self.find_prompt(delay_factor=delay_factor)
1365                 except ValueError:
1366                     prompt = self.base_prompt
1367             else:
1368                 prompt = self.base_prompt
1369                 search_pattern = re.escape(prompt.strip())
1370             else:
1371                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         ...
1090         self.write_channel(self.RETURN)
1091         sleep_time = delay_factor * 0.1
1092         time.sleep(sleep_time)
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Find prompt

```
1293     @select_cmd_verify
1294     def send_command(
1295         self,
1296         command_string,
1297         expect_string=None,
1298         delay_factor=1,
1299         max_loops=500,
1300         auto_find_prompt=True,
1301         strip_prompt=True,
1302         strip_command=True,
1303         normalize=True,
1304         use_textfsm=False,
1305         textfsm_template=None,
1306         use_genie=False,
1307         cmd_verify=True,
1308     ):
1309         """Execute command_string on the SSH channel using a pattern
1310         used for show commands. By default this method will keep waiting
1311         until the network device prompt is detected. The current network device
1312         automatically.
```

```
1359         # Find the current router prompt
1360         if expect_string is None:
1361             if auto_find_prompt:
1362                 try:
1363                     prompt = self.find_prompt(delay_factor=delay_factor)
1364                 except ValueError:
1365                     prompt = self.base_prompt
1366             else:
1367                 prompt = self.base_prompt
1368                 search_pattern = re.escape(prompt.strip())
1369             else:
1370                 search_pattern = expect_string
1081     def find_prompt(self, delay_factor=1):
1082         """Finds the current network device prompt, last line only.
1083
1084         :param delay_factor: See __init__: global_delay_factor
1085         :type delay_factor: int
1086         """
1087         delay_factor = self.select_delay_factor(delay_factor)
1088         self.clear_buffer()
1089         self.write_channel(self.RETURN)
1090         sleep_time = delay_factor * 0.1
1091         time.sleep(sleep_time)
1092
```

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

netmiko - Cancel / send_command_timing

```
my_device_session = ConnectHandler(**my_device)
```

```
print(my_device_session.send_command('reload', expect_string='confirm'))
```

Proceed with reload? [confirm]

```
from netmiko import CNTL_SHIFT_6
print(my_device_session.send_command_timing('\x03'))
print(my_device_session.send_command_timing(CNTL_SHIFT_6))
```

^C
sw01#

Takeaways

- ⌘ netmiko supports nearly 50 device types/vendors
- ⌘ Attention on expect_strings and auto_find_prompt
- ⌘ Use session_log param for logging
- ⌘ global_delay_factor on lame devices



... and it goes on

The image shows a winding mountain road, likely the Trollstigen in Norway, viewed from an aerial perspective. The road is a two-lane asphalt path with white dashed lines, curving through a steep, rocky hillside covered in green vegetation. A small stream or waterfall is visible on the right side of the road. In the bottom right corner, there is a red graphic element consisting of three horizontal bars of varying lengths. Two red lines extend from this graphic to the text labels on the right.

2018 napalm/
stackstorm

2019 nornir



nornir



Parallelism and abstract connection handling required

- Task executions tooks too long
- nornir is a pure python automation framework
 - Why not ansible - DSL is nice but have you ever implemented some logic?
- nornir handles multiple types of connections (connection plugins)
- <https://github.com/nornir-automation/nornir>

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io


class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io


class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Inventory

```
%cat nornir_inventory.py
```

```
from nornir.core.deserializer.inventory import Inventory
import io
```

```
class MyInventory(Inventory):
    def __init__(self, **kwargs):
        hosts = {
            "sw1": {
                "data": {
                    "foo": "bar",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.21",
                "username": "neops",
                "password": "cisco",
                "platform": "ios",
                "groups": [],
                "connection_options": {
                    "napalm": {
                        "extras": {
                            "optional_args": {}
                        }
                    },
                    "netmiko": {
                        "extras": {}
                    }
                }
            },
            "sw2": {
                "data": {
                    "foo": "foo",
                    "log": io.BytesIO()
                },
                "hostname": "192.168.123.22",
                "username": "neops",
                "password": "cisco",
            }
        }
    }
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()
```

```
dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()
```

```
dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Init & filters

```
from nornir import InitNornir
from nornir.core.deserializer.inventory import Inventory
nr = InitNornir(
    core={"num_workers": 20},
    dry_run=True,
    logging={
        "enabled": False
    },
    inventory={
        "plugin": "nornir_inventory.MyInventory",
    },
)
```

```
nr.filter(name="sw1").inventory.hosts.items()

dict_items([('sw1', Host: sw1)])
```

```
from nornir.core.filter import F
nr.filter(F(data__foo="bar") | F(data__foo="foo")).inventory.hosts.keys()

dict_keys(['sw1', 'sw2'])
```

nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

    class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

    class MockTask(Processor):
```



nornir - Processors

```
from nornir.core.processor import Processor
from nornir.core.inventory import Host
from nornir.core.task import AggregatedResult, MultiResult, Task
import io

class PrintTaskState(Processor):
    def task_started(self, task: Task) -> None:
        print(f'task {task.name} started\n')

    def task_completed(self, task: Task, result: AggregatedResult) -> None:
        print(f'task {task.name} completed\n')

    def task_instance_started(self, task: Task, host: Host) -> None:
        print(f'task instance {task.name} on {host.name} started\n')

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'task instance {task.name} on {host.name} completed\n')

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        print(f'subtask instance {task.name} on {host.name} started\n')

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        print(f'subtask instance {task.name} on {host.name} completed\n')

class MockTask(Processor):
```



nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass
    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```

nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        pass

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
            self, task: Task, host: Host, result: MultiResult
        ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass

    def subtask_instance_completed(
            self, task: Task, host: Host, result: MultiResult
        ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```

nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass
    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```



nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host):
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host):
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass

    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```



nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass
    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```

nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass
    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```

nornir - Processors - Log

```
class TaskLog(Processor):
    # set logging
    def task_started(self, host: Host) -> None:
        for key, item in hosts.items():
            item["connection_options"]["napalm"]["extras"]["optional_args"]["session_log"] = item["data"]["log"]

    def task_completed(self, host: Host) -> None:
        item["connection_options"]["netmiko"]["extras"]["session_log"] = item["data"]["log"] # noqa 501
        pass

    def task_instance_started(self, task: Task, host: Host) -> None:
        pass

    def task_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)

    def subtask_instance_started(self, task: Task, host: Host) -> None:
        pass
    def subtask_instance_completed(
        self, task: Task, host: Host, result: MultiResult
    ) -> None:
        logger = host.data.get('log')
        if logger and isinstance(logger, io.BytesIO):
            last_log = logger.getvalue().decode()
            print(f"log of {host.name}\n{last_log}\n")
            logger.truncate(0)
            logger.seek(0)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task: Task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
nr_pr = nr.with_processors([PrintTaskState(), TaskLog()])
```

```
from nornir.plugins.tasks import networking

def my_task(task, **kwargs):
    print(f"run task on {task.host}\n")
    r1 = task.run(
        task=networking.napalm_cli,
        commands=["show version"])
    r2 = task.run(
        task=networking.napalm_cli,
        commands=["show int desc"])
    return Result(host=task.host, result=True)

nr_pr.run(
    task=my_task
)
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started
task instance my_task on sw2 started
```

```
run task on sw1
run task on sw2
```

```
subtask instance napalm_cli on sw1 started
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****
* IOSv is strictly limited to use for evaluation, demonstration and
* education. IOSv is provided as-is and is not supported by Cisco's
* Technical Advisory Center. Any use or disclosure, in whole or in part
* of the IOSv Software or Documentation to any third party for any
* purposes is expressly prohibited except as otherwise authorized by
* Cisco in writing.
```

```
sw02#
```

```
sw02#terminal length 0
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```



nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started
task instance my_task on sw2 started
```

```
run task on sw1
run task on sw2
```

```
subtask instance napalm_cli on sw1 started
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****
* IOSv is strictly limited to use for evaluation, demonstration and
* education. IOSv is provided as-is and is not supported by Cisco's
* Technical Advisory Center. Any use or disclosure, in whole or in part
* of the IOSv Software or Documentation to any third party for any
* purposes is expressly prohibited except as otherwise authorized by
* Cisco in writing.
```

```
sw02#
```

```
sw02#terminal length 0
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started  
task instance my_task on sw2 started
```

```
run task on sw1  
run task on sw2
```

```
subtask instance napalm_cli on sw1 started  
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed  
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****  
* IOSv is strictly limited to use for evaluation, demonstration and IC  
* education. IOSv is provided as-is and is not supported by Cisco's  
* Technical Advisory Center. Any use or disclosure, in whole or in par  
* of the IOSv Software or Documentation to any third party for any  
* purposes is expressly prohibited except as otherwise authorized by  
* Cisco in writing.  
*****
```

```
sw02#
```

```
sw02#terminal length 0  
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers  
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E  
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started  
task instance my_task on sw2 started
```

```
run task on sw1  
run task on sw2
```

```
subtask instance napalm_cli on sw1 started  
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed  
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****  
* IOSv is strictly limited to use for evaluation, demonstration and IC  
* education. IOSv is provided as-is and is not supported by Cisco's  
* Technical Advisory Center. Any use or disclosure, in whole or in par  
* of the IOSv Software or Documentation to any third party for any  
* purposes is expressly prohibited except as otherwise authorized by  
* Cisco in writing.  
*****
```

```
sw02#
```

```
sw02#terminal length 0  
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers  
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E  
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started
task instance my_task on sw2 started
```

```
run task on sw1
run task on sw2
```

```
subtask instance napalm_cli on sw1 started
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****
* IOSv is strictly limited to use for evaluation, demonstration and
* education. IOSv is provided as-is and is not supported by Cisco's
* Technical Advisory Center. Any use or disclosure, in whole or in part
* of the IOSv Software or Documentation to any third party for any
* purposes is expressly prohibited except as otherwise authorized by
* Cisco in writing.
```

```
sw02#
```

```
sw02#terminal length 0
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```



nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started  
task instance my_task on sw2 started
```

```
run task on sw1  
run task on sw2
```

```
subtask instance napalm_cli on sw1 started  
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed  
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****  
* IOSv is strictly limited to use for evaluation, demonstration and IC  
* education. IOSv is provided as-is and is not supported by Cisco's  
* Technical Advisory Center. Any use or disclosure, in whole or in par  
* of the IOSv Software or Documentation to any third party for any  
* purposes is expressly prohibited except as otherwise authorized by  
* Cisco in writing.  
*****
```

```
sw02#
```

```
sw02#terminal length 0  
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers  
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E  
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```



nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started  
task instance my_task on sw2 started
```

```
run task on sw1  
run task on sw2
```

```
subtask instance napalm_cli on sw1 started  
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed  
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****  
* IOSv is strictly limited to use for evaluation, demonstration and IC  
* education. IOSv is provided as-is and is not supported by Cisco's  
* Technical Advisory Center. Any use or disclosure, in whole or in par  
* of the IOSv Software or Documentation to any third party for any  
* purposes is expressly prohibited except as otherwise authorized by  
* Cisco in writing.  
*****
```

```
sw02#
```

```
sw02#terminal length 0  
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers  
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E  
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started
task instance my_task on sw2 started
```

```
run task on sw1
run task on sw2
```

```
subtask instance napalm_cli on sw1 started
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****
* IOSv is strictly limited to use for evaluation, demonstration and IC
* education. IOSv is provided as-is and is not supported by Cisco's
* Technical Advisory Center. Any use or disclosure, in whole or in par
* of the IOSv Software or Documentation to any third party for any
* purposes is expressly prohibited except as otherwise authorized by
* Cisco in writing.
```

```
sw02#
sw02#terminal length 0
sw02#terminal width 511
sw02#
sw02#
sw02#show version
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
Technical Support: http://www.*****.com/techsupport
```

nornir - Processors - Log

```
task my_task started
```

```
task instance my_task on sw1 started
task instance my_task on sw2 started
```

```
run task on sw1
run task on sw2
```

```
subtask instance napalm_cli on sw1 started
subtask instance napalm_cli on sw2 started
```

```
subtask instance napalm_cli on sw2 completed
subtask instance napalm_cli on sw1 completed
```

```
log of sw2
```

```
*****
* IOSv is strictly limited to use for evaluation, demonstration and
* education. IOSv is provided as-is and is not supported by Cisco's
* Technical Advisory Center. Any use or disclosure, in whole or in part
* of the IOSv Software or Documentation to any third party for any
* purposes is expressly prohibited except as otherwise authorized by
* Cisco in writing.
```

```
sw02#
```

```
sw02#terminal length 0
sw02#terminal width 511
```

```
sw02#
```

```
sw02#
```

```
sw02#show version
```

```
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Vers
DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
Technical Support: http://www.\*\*\*\*\*.com/techsupport
```

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

sw01#

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

log of sw2

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

```
Status
```

```
Protocol
```

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

v1.1.0
sw01#

up

up

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

```
Status
```

```
Protocol
```

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

sw01#

```
subtask instance napalm_cli on sw2 started
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed
subtask instance napalm_cli on sw2 completed
```

log of sw2

```
sw02#show int desc
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed



nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

	Status	Protocol	
			task instance my_task on sw2 completed task instance my_task on sw1 completed

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

```
log of sw2
```

```
log of sw1
```

```
task my_task completed
```

nornir - Processors - Log

```
sw01#
```

```
subtask instance napalm_cli on sw2 started  
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed  
subtask instance napalm_cli on sw2 completed
```

```
log of sw2
```

```
sw02#show int desc  
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed  
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed

nornir - Processors - Log

sw01#

```
subtask instance napalm_cli on sw2 started
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed
subtask instance napalm_cli on sw2 completed
```

log of sw2

```
sw02#show int desc
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed

nornir - Processors - Log

sw01#

```
subtask instance napalm_cli on sw2 started
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed
subtask instance napalm_cli on sw2 completed
```

log of sw2

```
sw02#show int desc
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed

nornir - Processors - Log

sw01#

```
subtask instance napalm_cli on sw2 started
subtask instance napalm_cli on sw1 started
```

```
subtask instance napalm_cli on sw1 completed
subtask instance napalm_cli on sw2 completed
```

log of sw2

```
sw02#show int desc
Interface
```

Status

Protocol

```
task instance my_task on sw2 completed
task instance my_task on sw1 completed
```

log of sw2

log of sw1

task my_task completed

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    r = task.run(task=my_subtask, name="subtask1")
    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

```
subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started
```

```
subtask instance subtask1 on sw2 completed
```

```
log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed

subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed

subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2
```

```
Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed

subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started
```

```
log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw1 started
subtask instance subtask1 on sw2 started

subtask instance subtask1 on sw2 completed

log of sw2

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-12-70c096714b66>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed

subtask instance subtask1 on sw1 completed
subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed

log of sw2
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    try:
        r = task.run(task=my_subtask, name="subtask1")
    except Exception:
        task.results.pop()

    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    try:
        r = task.run(task=my_subtask, name="subtask1")
    except Exception:
        task.results.pop()

    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

nornir - Exceptions

```
nr_pr.data.reset_failed_hosts()

def my_subtask(task: Task, **kwargs):
    if task.host.name == "sw1":
        raise Exception("failed")
    return Result(host=task.host, result=True)

def my_task(task: Task, **kwargs):
    try:
        r = task.run(task=my_subtask, name="subtask1")
    except Exception:
        task.results.pop()

    r = task.run(task=my_subtask, name="subtask2")
    return Result(host=task.host, result=r.result)

nr_pr.run(
    task=my_task
)
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw2 started
subtask instance subtask1 on sw1 started

subtask instance subtask1 on sw2 completed

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-13-c2e5aaac46b9>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
log of sw2

subtask instance subtask1 on sw1 completed

subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed
subtask instance subtask2 on sw1 started
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw2 started
subtask instance subtask1 on sw1 started

subtask instance subtask1 on sw2 completed

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-13-c2e5aaac46b9>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
log of sw2

subtask instance subtask1 on sw1 completed

subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed
subtask instance subtask2 on sw1 started
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw2 started
subtask instance subtask1 on sw1 started

subtask instance subtask1 on sw2 completed

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-13-c2e5aaac46b9>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
log of sw2

subtask instance subtask1 on sw1 completed

subtask instance subtask2 on sw2 started

log of sw1

subtask instance subtask2 on sw2 completed
subtask instance subtask2 on sw1 started
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw2 started
subtask instance subtask1 on sw1 started

subtask instance subtask1 on sw2 completed

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-13-c2e5aaac46b9>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
log of sw2

subtask instance subtask1 on sw1 completed

subtask instance subtask2 on sw2 started

log of sw1
```

```
subtask instance subtask2 on sw2 completed
subtask instance subtask2 on sw1 started
```

nornir - Exceptions

```
task my_task started

task instance my_task on sw1 started
task instance my_task on sw2 started

subtask instance subtask1 on sw2 started
subtask instance subtask1 on sw1 started

subtask instance subtask1 on sw2 completed

Host 'sw1': task 'subtask1' failed with traceback:
Traceback (most recent call last):
  File "/Users/obi/.local/share/virtualenvs/pud-2020-A32_uhK5/lib/python3.8/site-packages/nornir/core/task.py", line
85, in start
    r = self.task(self, **self.params)
  File "<ipython-input-13-c2e5aaac46b9>", line 5, in my_subtask
    raise Exception("failed")
Exception: failed
```

```
log of sw2

subtask instance subtask1 on sw1 completed

subtask instance subtask2 on sw2 started

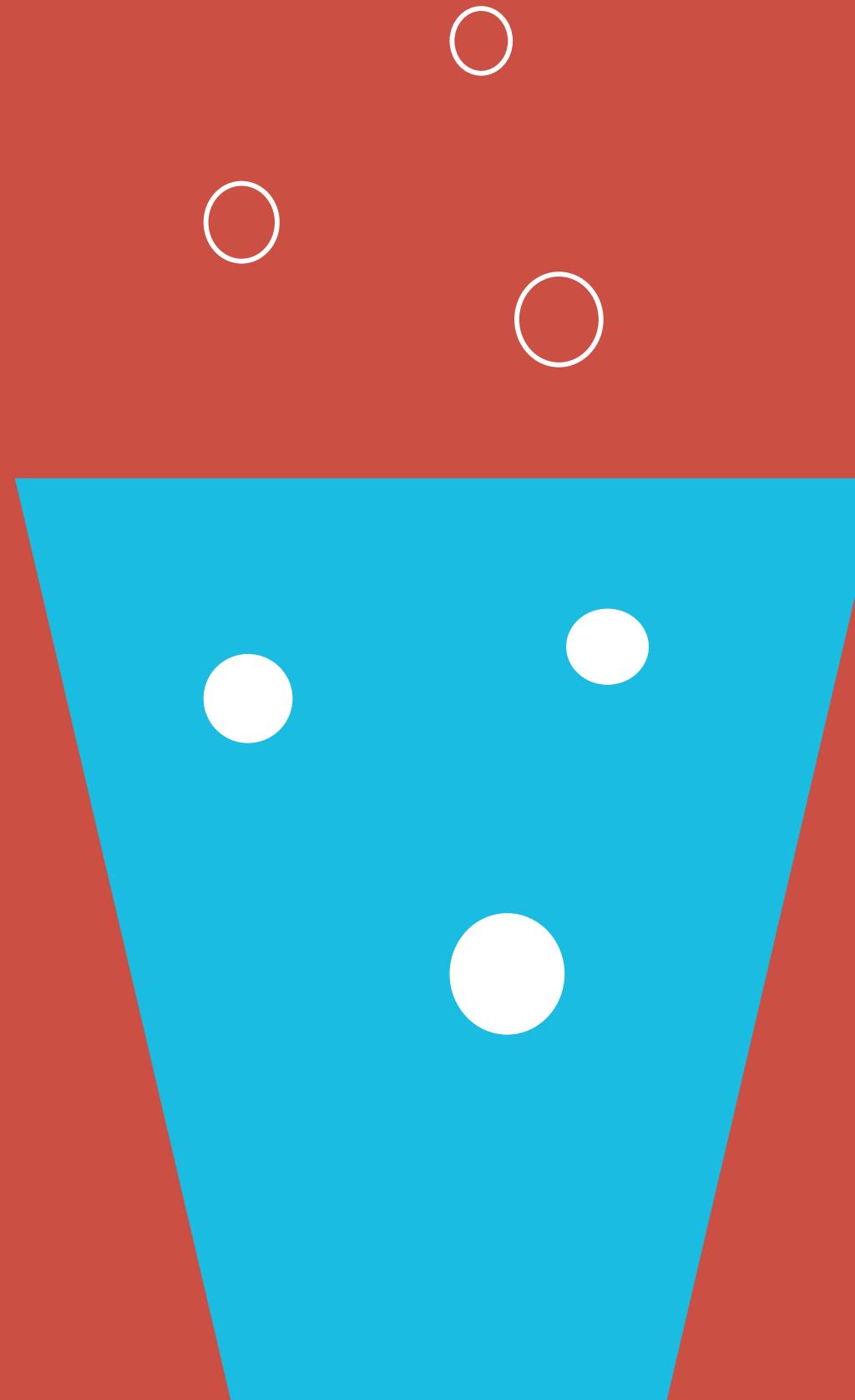
log of sw1

subtask instance subtask2 on sw2 completed
subtask instance subtask2 on sw1 started
```

Takeaways

- # Pure python
- # Connection and task handling (abstraction)
- # Advanced filter functions
- # Supports processors
- # On exception no more tasks per device are executed

Time for a break!





2019 textFSM



textFSM

Handle semi structured data

- How to handle that bunch of data, regular expressions?
 - yes, but line per line?
- TextFSM is a parsing engine by google for parsing semi structured data
- And it uses regular expressions ☺
- <https://github.com/google/textfsm>

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+${UPTIME}
^${SYSTEM}\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+${RELOAD_REASON}\s+$$
^[Pp]rocessor\s+board\s+ID\s+${SERIAL}
^[Cc]isco\s+${HARDWARE}.+
^[Cc]onfiguration\s+register\s+is\s+${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+${MAC}
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^${SYSTEM} [Ss]erial [Nn]umber\s+: \s+${SERIAL}
^${MODEL}\s+[Nn]umber\s+: \s+${HARDWARE}\s*
^[Cc]onfiguration\s+register\s+is\s+${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+${MAC}'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})'''
```

Start

```
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is
```

Stack

```
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\), \sVersion\s${VERSION}, *\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)": ${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason: |System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE} .
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is
```

```
Stack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE}\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s+\${VERSION},*\s+RELEASE.*
^ROM: \${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+\${UPTIME}
^${SYSTEM}\s+image\s+file\s+is\s+"(.*)":\${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+\${RELOAD_REASON}\s+$$
^Processor\s+board\s+ID\s+\${SERIAL}
^Cisco\s+\${HARDWARE}.+
^Configuration\s+register\s+is\s+\${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+\${MAC}
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^${SYSTEM} [Ss]erial [Nn]umber\s+: \s+\${SERIAL}
^${MODEL}\s+[Nn]umber\s+: \s+\${HARDWARE}\s*
^Configuration\s+register\s+is\s+\${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+\${MAC}'''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE}.\+
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE}\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+${UPTIME}
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+${RELOAD_REASON}\s+$$
^[Pp]rocessor\s+board\s+ID\s+${SERIAL}
^[Cc]isco\s+${HARDWARE}.+
^[Cc]onfiguration\s+register\s+is\s+${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+${MAC}
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is
```

```
Stack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+${SERIAL}
^[Mm]odel\s+[Nn]umber\s+: \s+${HARDWARE}\s*
^[Cc]onfiguration\s+register\s+is\s+${CONFIG_REGISTER}
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+${MAC}''
```

```
tfsm = TextFSM(StringIO(tpl))
```

textfsm - Template

```
from textfsm import TextFSM
from io import StringIO
```

```
tpl = '''Value VERSION (.+?)
Value ROMMON (\S+)
Value HOSTNAME (\S+)
Value UPTIME (.+)
Value RELOAD_REASON (.+?)
Value RUNNING_IMAGE (\S+)
Value List HARDWARE (\S+\d\S+)
Value List SERIAL (\S+)
Value CONFIG_REGISTER (\S+)
Value List MAC ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})\n\nStart
^.*Software\s+\),\sVersion\s${VERSION},*\s+RELEASE.*
^ROM: ${ROMMON}
^${HOSTNAME}\s+uptime\s+is\s+$UPTIME
^[sS]ystem\s+image\s+file\s+is\s+"(.*)":${RUNNING_IMAGE}"
^(?:[lL]ast\s+reload\s+reason:|System\s+returned\s+to\s+ROM\s+by)\s+$RELOAD_REASON\s+$$
^[Pp]rocessor\s+board\s+ID\s+$SERIAL
^[Cc]isco\s+$HARDWARE>.
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC
^Switch Port -> Stack
# Capture time-stamp if vty line has command time-stamping turned on
^Load\s+for\s+
^Time\s+source\s+is\n\nStack
^[Ss]ystem [Ss]erial [Nn]umber\s+: \s+$SERIAL
^[Mm]odel\s+[Nn]umber\s+: \s+$HARDWARE\s*
^[Cc]onfiguration\s+register\s+is\s+$CONFIG_REGISTER
^Base [Ee]thernet MAC [Aa]ddress\s+: \s+$MAC'''
```

```
tfsm = TextFSM(StringIO(tpl))
```



textfsm - Input

```
output = '''Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSC
Technical Support: http://www.*****.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Tue 19-Jun-18 06:06 by mmem
```

ROM: Bootstrap program is IOSv

```
sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
System returned to ROM by reload
System image file is "flash0:/vios_12-adventerprisek9-m"
Last reload reason: Unknown reason
```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.*****.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@*****.com.

```
Cisco IOSv () processor (revision 1.0) with 574713K/209920K bytes of memory.
Processor board ID 9EI00FHC05X
1 Virtual Ethernet interface
16 Gigabit Ethernet interfaces
DRAM configuration is 72 bits wide with parity disabled.
256K bytes of non-volatile configuration memory.
2097152K bytes of ATA System CompactFlash 0 (Read/Write)
0K bytes of ATA CompactFlash 1 (Read/Write)
0K bytes of ATA CompactFlash 2 (Read/Write)
```

textfsm - Result

```
tfsm.Reset()
tfsm.ParseTextToDicts(output)

[{'VERSION': '',
 'ROMMON': 'Bootstrap',
 'HOSTNAME': 'sw02',
 'UPTIME': '1 week, 1 day, 9 hours, 22 minutes',
 'RELOAD_REASON': 'Unknown reason',
 'RUNNING_IMAGE': '/vios_12-adventureprisek9-m',
 'HARDWARE': [],
 'SERIAL': ['9EI00FHC05X'],
 'CONFIG_REGISTER': '0x101',
 'MAC': []}]
```

textfsm - Result

```
tfsm.Reset()  
tfsm.ParseTextToDicts(output)
```

```
[{'VERSION': '',  
 'ROMMON': 'Bootstrap',  
 'HOSTNAME': 'sw02',  
 'UPTIME': '1 week, 1 day, 9 hours, 22 minutes',  
 'RELOAD_REASON': 'Unknown reason',  
 'RUNNING_IMAGE': '/vios_12-adventureprisek9-m',  
 'HARDWARE': [],  
 'SERIAL': ['9EI00FHC05X'],  
 'CONFIG_REGISTER': '0x101',  
 'MAC': []}]
```

textfsm - Result

```
tfsm.Reset()
tfsm.ParseTextToDicts(output)

[{'VERSION': '',
 'ROMMON': 'Bootstrap',
 'HOSTNAME': 'sw02',
 'UPTIME': '1 week, 1 day, 9 hours, 22 minutes',
 'RELOAD_REASON': 'Unknown reason',
 'RUNNING_IMAGE': '/vios_12-adventureprisek9-m',
 'HARDWARE': [],
 'SERIAL': ['9EI00FHC05X'],
 'CONFIG_REGISTER': '0x101',
 'MAC': []}]
```

textfsm - Result

```
tfsm.Reset()  
tfsm.ParseTextToDicts(output)
```

```
[{'VERSION': '',  
 'ROMMON': 'Bootstrap',  
 'HOSTNAME': 'sw02',  
 'UPTIME': '1 week, 1 day, 9 hours, 22 minutes',  
 'RELOAD_REASON': 'Unknown reason',  
 'RUNNING_IMAGE': '/vios_12-adventureprisek9-m',  
 'HARDWARE': [],  
 'SERIAL': ['9EI00FHC05X'],  
 'CONFIG_REGISTER': '0x101',  
 'MAC': []}]
```

textfsm - Result

```
tfsm.Reset()
tfsm.ParseTextToDicts(output)

[{'VERSION': '',
 'ROMMON': 'Bootstrap',
 'HOSTNAME': 'sw02',
 'UPTIME': '1 week, 1 day, 9 hours, 22 minutes',
 'RELOAD_REASON': 'Unknown reason',
 'RUNNING_IMAGE': '/vios_12-adventureprisek9-m',
 'HARDWARE': [],
 'SERIAL': ['9EI00FHC05X'],
 'CONFIG_REGISTER': '0x101',
 'MAC': []}]
```

Takeaways

- ⌘ Many templates - ntc templates (network to code)
- ⌘ Easy to adopt
- ⌘ Get structured data for further processing
(abstraction)



some more?

napalm

- Mostly based on netmiko
- Unify device access and get facts
- restrictive
- Extendable (napalm_custom)
- <https://github.com/napalm-automation/napalm>

pyATS/Genie

■ Cisco based

■ Parsing semi structured data

■ Testing

■ <https://github.com/CiscoTestAutomation>

batfish

■ Testing/mock

■ Analysis

■ <https://github.com/batfish/batfish>



CLI

CLI is dead

- CLI is dead, long lives the CLI
- RestConf, NetConf
- Platform support, lot of 2960er out there
- No unique configuration method
- Habits...

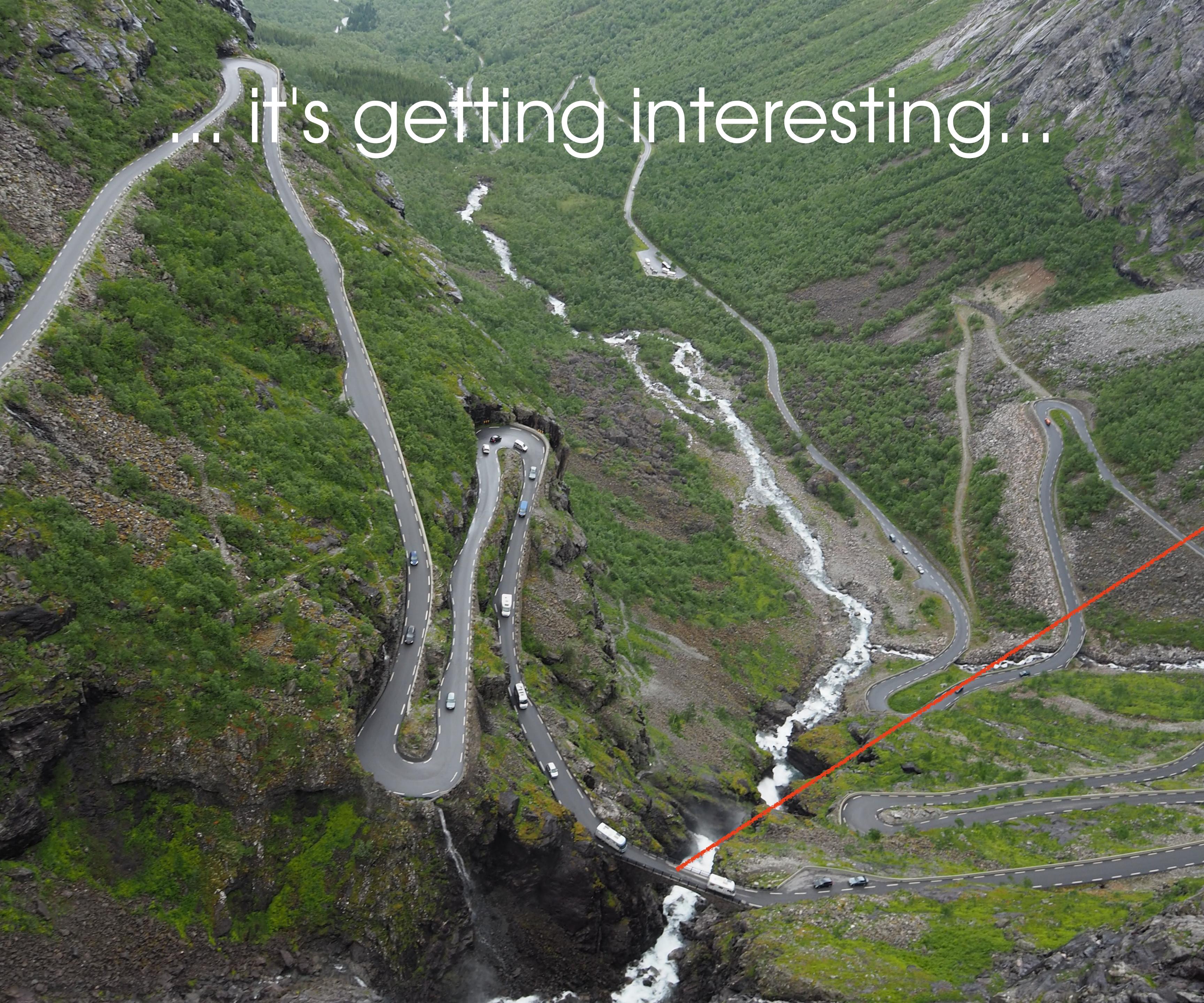
Network/Infrastructure as code

■ Nice!!

■ But a long `/i\` to go

■ Automated testing, yes but..

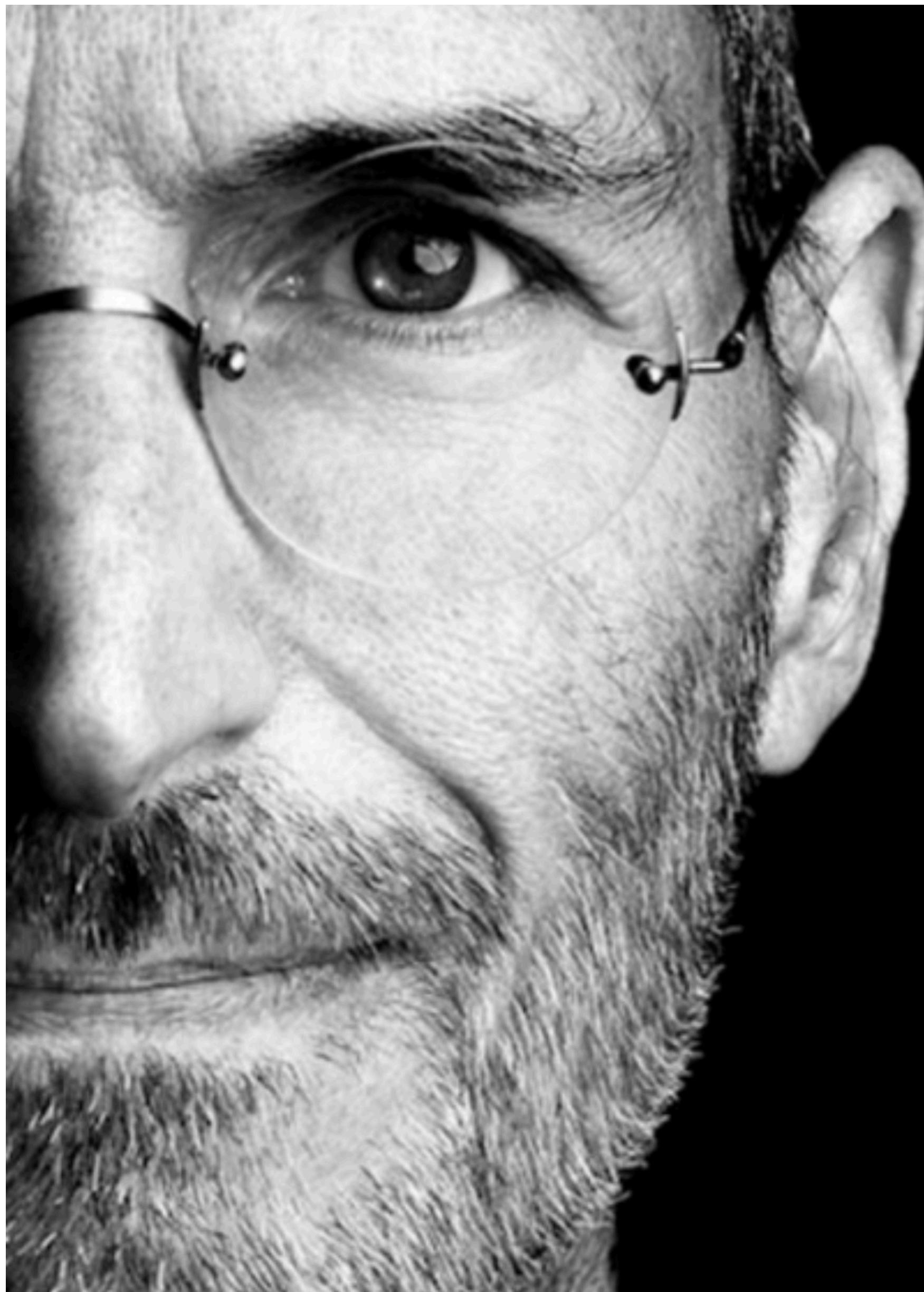
... it's getting interesting...



2020 vue.js / json
schema forms



UX



“

You've got to start with the
CUSTOMER EXPERIENCE and
work backwards to the technology.

”

- Steve Jobs

UI/UX matters

- Huge amount of information - better overview
- Even more important @ network automation
- Reduces errors, shorter troubleshooting times
- Collaboration
- Delegation / abstraction



vuetify jsonschema form

Backend to frontend

■ Data structure definition

■ Rendering in the Frontend

■ Define user inputs within the task

■ <https://github.com/koumoul-dev/vuetify-jsonschema-form>

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
    "$id": "https://neops.io/schema/example.json",  
    "$schema": "http://json-schema.org/draft-07/schema#",  
    "title": "Demo Form JSON Schma",  
    "type": "object",  
    "required": [  
        "foo",  
        "bar"  
    ],  
    "properties": {  
        "foo": {  
            "title": "Foo",  
            "type": "string",  
            "description": "Foo rendered as normal input field"  
        },  
        "bar": {  
            "title": "Bar",  
            "type": "string",  
            "description": "as select",  
            "enum": [  
                "foo",  
                "bar"  
            ]  
        },  
        "bool": {  
            "title": "bool",  
            "type": "boolean",  
            "description": "boolean is rendered as checkbox"  
        }  
    }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
    "$id": "https://neops.io/schema/example.json",  
    "$schema": "http://json-schema.org/draft-07/schema#",  
    "title": "Demo Form JSON Schma",  
    "type": "object",  
    "required": [  
        "foo",  
        "bar"  
    ],  
    "properties": {  
        "foo": {  
            "title": "Foo",  
            "type": "string",  
            "description": "Foo rendered as normal input field"  
        },  
        "bar": {  
            "title": "Bar",  
            "type": "string",  
            "description": "as select",  
            "enum": [  
                "foo",  
                "bar"  
            ]  
        },  
        "bool": {  
            "title": "bool",  
            "type": "boolean",  
            "description": "boolean is rendered as checkbox"  
        }  
    }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
>],  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
>},  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
>      ]  
>    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
>    }  
>  }  
>}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
>],  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    <"bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]>  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  ],  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

```
{  
  "$id": "https://neops.io/schema/example.json",  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "title": "Demo Form JSON Schma",  
  "type": "object",  
  "required": [  
    "foo",  
    "bar"  
,  
  "properties": {  
    "foo": {  
      "title": "Foo",  
      "type": "string",  
      "description": "Foo rendered as normal input field"  
    },  
    "bar": {  
      "title": "Bar",  
      "type": "string",  
      "description": "as select",  
      "enum": [  
        "foo",  
        "bar"  
      ]  
    },  
    "bool": {  
      "title": "bool",  
      "type": "boolean",  
      "description": "boolean is rendered as checkbox"  
    }  
  }  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo
foofoo

i

Bar
bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
  "foo": "foofoo",  
  "bar": "bar",  
  "bool": true  
}
```

JSON schema

Foo

foofoo

i

Bar

bar

as select

i

bool

```
{  
    "foo": "foofoo",  
    "bar": "bar",  
    "bool": true  
}
```

VueJS

```
<script>
  Vue.component('VJsf', VJsf.default)
  new Vue({
    el: "#app",
    vuetify: new Vuetify(),
    data: {
      schema: {...},
      dataModel: {},
    },
    computed: {
      prettyModel: function () {
        return JSON.stringify(this.dataModel, null, 2);
      }
    }
  });
</script>
```

```
<div id="app">
  <v-app>
    <v-main>
      <v-container>
        <v-form>
          <v-jsf v-model="dataModel" :schema="schema"/>
        </v-form>
        <br />
        <pre>
{{ prettyModel }}
        </pre>
      </v-container>
    </v-main>
  </v-app>
</div>
```

VueJS

```
<script>
  Vue.component('VJsf', VJsf.default)
  new Vue({
    el: "#app",
    vuetify: new Vuetify(),

    data: {
      schema: {...},
      dataModel: {},
    },
    computed: {
      prettyModel: function () {
        return JSON.stringify(this.dataModel, null, 2);
      }
    });
  </script>
```

```
<div id="app">
  <v-app>
    <v-main>
      <v-container>
        <v-form>
          <v-jsf v-model="dataModel" :schema="schema"/>
        </v-form>
        <br />
        <pre>
{{ prettyModel }}
        </pre>
      </v-container>
    </v-main>
  </v-app>
</div>
```

VueJS

```
<script>
  Vue.component('VJsf', VJsf.default)
  new Vue({
    el: "#app",
    vuetify: new Vuetify(),
    data: {
      schema: {...},
      dataModel: {},
    },
    computed: {
      prettyModel: function () {
        return JSON.stringify(this.dataModel, null, 2);
      }
    }
  });
</script>
```

```
<div id="app">
  <v-app>
    <v-main>
      <v-container>
        <v-form>
          <v-jsf v-model="dataModel" :schema="schema"/>
        </v-form>
        <br />
        <pre>
{{ prettyModel }}
        </pre>
      </v-container>
    </v-main>
  </v-app>
</div>
```

VueJS

```
<script>
  Vue.component('VJsf', VJsf.default)
  new Vue({
    el: "#app",
    vuetify: new Vuetify(),

    data: {
      schema: {...},
      dataModel: {},
    },
    computed: {
      prettyModel: function () {
        return JSON.stringify(this.dataModel, null, 2);
      }
    }
  });
</script>
```

```
<div id="app">
  <v-app>
    <v-main>
      <v-container>
        <v-form>
          <v-jsf v-model="dataModel" :schema="schema"/>
        </v-form>
        <br />
        <pre>
{{ prettyModel }}
        </pre>
      </v-container>
    </v-main>
  </v-app>
</div>
```

VueJS

```
<script>
  Vue.component('VJsf', VJsf.default)
  new Vue({
    el: "#app",
    vuetify: new Vuetify(),
    data: {
      schema: {...},
      dataModel: {},
    },
    computed: {
      prettyModel: function () {
        return JSON.stringify(this.dataModel, null, 2);
      }
    }
  });
</script>
```

```
<div id="app">
  <v-app>
    <v-main>
      <v-container>
        <v-form>
          <v-jsf v-model="dataModel" :schema="schema"/>
        </v-form>
        <br />
        <pre>
{{ prettyModel }}
        </pre>
      </v-container>
    </v-main>
  </v-app>
</div>
```



just a bit more...

today?



Today

Putting it together - neops.io

- nornir as task engine

- task registry/resolving (model/vendor support)

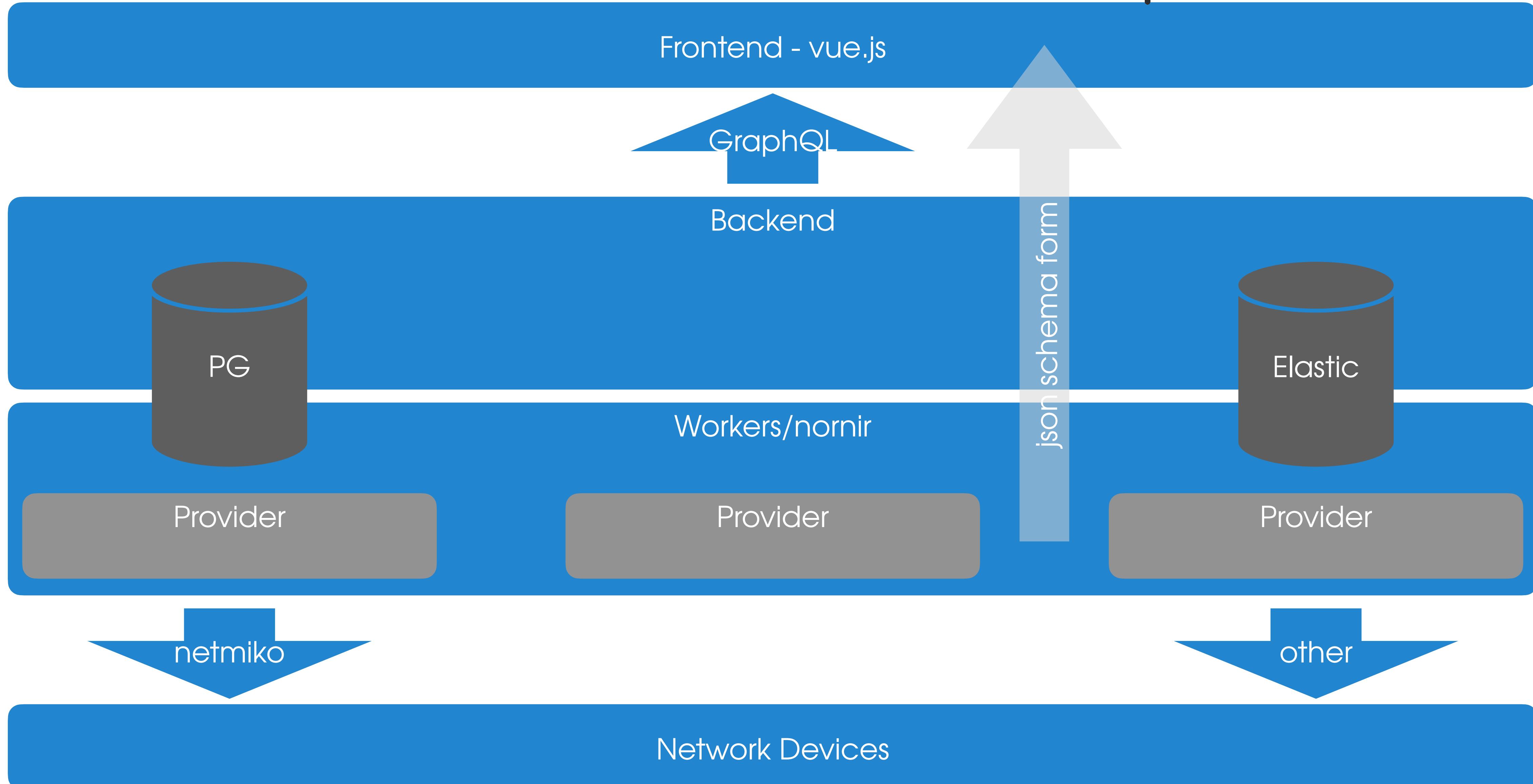
- flexible data structures

- searchable

- flexible ui - task delegation

- json schema forms

Full Stack Solution - neops.io





Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search devices.facts.vlans.name: CLIENT-C

Hostname	IP	Platform	Location	Interfaces	VLANs		ID		
dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2			
sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1			

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vrf2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search: devices.facts.vlans.name: CLIENT-C

	Hostname	IP	Platform	Location	Interfaces	VLANs	ID		
0/2									
	dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2		
	sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1		

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vrf2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search devices.facts.vlans.name: CLIENT-C

Hostname	IP	Platform	Location	Interfaces	VLANs		ID		
dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2			
sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1			

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vrf2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by

Search devices.facts.vlans.name: CLIENT-C

Hostname	IP	Platform	Location	Interfaces	VLANs		ID		
dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2			
sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1			

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vlan2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30 ▾

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search devices.facts.vlans.name: CLIENT-C

Hostname	IP	Platform	Location	Interfaces	VLANs		ID		
dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2			
sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1			

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vrf2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search devices.facts.vlans.name: CLIENT-C

	Hostname	IP	Platform	Location	Interfaces	VLANs	ID	⋮
0/2								
	dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2	
	sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1	

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vlan2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27>
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



choose a task

Rename VLAN on

PREVIEW





Network - Devices

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by



Search devices.facts.vlans.name: CLIENT-C

	Hostname	IP	Platform	Location	Interfaces	VLANs	ID		
0/2									
	dist-sw01.neops.io	192.168.123.11	ios	Zebra Office	17	default MGMT CLIENT-ABC CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	2		
	sw02.neops.io	192.168.123.22	ios	Zebra Office	17	default MGMT CLIENT-B CLIENT-C fddi-default token-ring-default fddinet-default trnet-default	1		

FACTS

```
1> {
2>   "mac": [ ],
3>   "vtp": [ ],
4>   "info": { },
5>   "snmp": "tetst",
6>   "vrfs": [],
7>   "vlan2": { },
8>   "vlans": [
9>     { },
10>     { },
11>     { },
12>     {
13>       "name": "CLIENT-C",
14>       "status": "active",
15>       "vlan_id": "102",
16>       "interfaces": [
17>         "Gi2/0"
18>       ]
19>     },
20>     { },
21>     { },
22>     { },
23>     { }
24>   ],
25>   "uptime": "3 weeks, 1 day, 12 hours, 24 minutes",
26>   "version": "f5a1"
27> }
```

CHECKS

CONFIG

Rows per page: 30

1-2 of 2



GROUPS

choose a task

Rename VLAN on

PREVIEW



Execution Details

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by
 zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

	State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
	▲	sw02.neops.io		Yes				ios
		config to apply		applied changes			log	
		<pre>! ! Rename VLAN 'CLIENT-CC' -> 'CLIENT-C' ! ! ! ! VLAN Config, Device is in VTP Transparent or Primary mode ! ! [{"name': 'default', 'status': 'active', 'vlan_id': '1', 'interfaces': ['Gi0/1', 'Gi0/2', 'Gi0/3', 'Gi1/0', 'Gi1/1', 'Gi1/2', 'Gi1/3', 'Gi2/1', 'Gi2/2', 'Gi2/3', 'Gi3/0', 'Gi3/1', 'Gi3/2', 'Gi3/3']}, {"name': 'MGMT', 'status': 'active', 'vlan_id': '10', 'interfaces': []}, {"name': 'CLIENT-B', 'status': 'active', 'vlan_id': '101', 'interfaces': []}, {"name': 'CLIENT-CC', 'status': 'active', 'vlan_id': '102', 'interfaces': ['Gi2/0']}, {'name': 'fddi-default', 'status': 'act/unsup', 'vlan_id': '1002', 'interfaces': []}, {"name': 'token-ring-default', 'status': 'act/unsup', 'vlan_id': '1003', 'interfaces': []}, {"name': 'fddinet-default', 'status': 'act/unsup', 'vlan_id': '1004', 'interfaces': []}, {"name': 'trnet-default', 'status': 'act/unsup', 'vlan_id': '1005', 'interfaces': []}] ! IGNORED VLAN ! vlan 1 ! name default</pre>	<pre>! ! Contextual Config Diffs: vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C</pre>	<pre>Task execute_neops_nornir_task started Subtask commands_textfsm started Subtask napalm_cli started</pre>	<pre>***** * IOSv is strictly limited to use for evaluation, demonstration and IOS * * education. IOSv is provided as-is and is not supported by Cisco's * * Technical Advisory Center. Any use or disclosure, in whole or in part, * of the IOSv Software or Documentation to any third party for any * * purposes is expressly prohibited except as otherwise authorized by * * Cisco in writing. *</pre>	<pre>***** sw02# sw02#terminal length 0 sw02#terminal width 511 sw02#</pre>		
	▼	sw01.neops.io		No				ios
	▼	router1-demo.neops.io		No				ios
	▼	dist-sw01.neops.io		Yes				ios

Rows per page:

30

1-4 of 4





Execution Details



Dashboard



Network



Tasks



Providers



Now Running



History



Cron Jobs



Task Tools



powered by
 zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
▲	sw02.neops.io		Yes				ios
	config to apply	applied changes		log			
	<pre>! ! Rename VLAN 'CLIENT-CC' -> 'CLIENT-C' ! ! VLAN Config, Device is in VTP Transparent or Primary mode ! ! [{'name': 'default', 'status': 'active', 'vlan_id': '1', 'interfaces': ['Gi0/1', 'Gi0/2', 'Gi0/3', 'Gi1/0', 'Gi1/1', 'Gi1/2', 'Gi1/3', 'Gi2/1', 'Gi2/2', 'Gi2/3', 'Gi3/0', 'Gi3/1', 'Gi3/2', 'Gi3/3'], { 'name': 'MGMT', 'status': 'active', 'vlan_id': '10', 'interfaces': [] }, { 'name': 'CLIENT-B', 'status': 'active', 'vlan_id': '101', 'interfaces': [] }, { 'name': 'CLIENT-CC', 'status': 'active', 'vlan_id': '102', 'interfaces': ['Gi2/0'], { 'name': 'fddi-default', 'status': 'act/unsup', 'vlan_id': '1002', 'interfaces': [] }, { 'name': 'token-ring-default', 'status': 'act/unsup', 'vlan_id': '1003', 'interfaces': [] }, { 'name': 'fddinet-default', 'status': 'act/unsup', 'vlan_id': '1004', 'interfaces': [] }, { 'name': 'trnet-default', 'status': 'act/unsup', 'vlan_id': '1005', 'interfaces': [] }] ! ! IGNORED VLAN ! ! vlan 1 ! ! name default</pre>	<pre>! ! Contextual Config Diffs: vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C</pre>	<pre>Task execute_neops_nornir_task started Subtask commands_textfsm started Subtask napalm_cli started ***** * IOSv is strictly limited to use for evaluation, demonstration and IOS * * education. IOSv is provided as-is and is not supported by Cisco's * * Technical Advisory Center. Any use or disclosure, in whole or in part, * of the IOSv Software or Documentation to any third party for any * * purposes is expressly prohibited except as otherwise authorized by * * Cisco in writing.</pre>				
▼	sw01.neops.io		No				ios
▼	router1-demo.neops.io		No				ios
▼	dist-sw01.neops.io		Yes				ios

Rows per page:

30

1-4 of 4

1



Execution Details

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by
zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

	State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
	▲	sw02.neops.io		Yes				ios
	config to apply			applied changes			log	
					!Contextual Config Diffs:			
					vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C			
							Task execute_neops_nornir_task started	
							Subtask commands_textfsm started	
							Subtask napalm_cli started	

							* IOSv is strictly limited to use for evaluation, demonstration and	
							IOS *	
							* education. IOSv is provided as-is and is not supported by Cisco's	
							*	
							* Technical Advisory Center. Any use or disclosure, in whole or in	
							part, *	
							* of the IOSv Software or Documentation to any third party for any	
							*	
							* purposes is expressly prohibited except as otherwise authorized by	
							*	
							* Cisco in writing.	
							*	

							sw02#	
							sw02#terminal length 0	
							sw02#terminal width 511	
							sw02#	
	▼	sw01.neops.io		No				ios
	▼	router1-demo.neops.io		No				ios
	▼	dist-sw01.neops.io		Yes				ios

Rows per page:

30

1-4 of 4





Execution Details



Dashboard



Network



Tasks



Providers



Now Running



History



Cron Jobs



Task Tools



powered by
 zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
▲	sw02.neops.io		Yes				ios
	config to apply	applied changes		log			
	<pre>! ! Rename VLAN 'CLIENT-CC' -> 'CLIENT-C' ! ! Contextual Config Diffs: vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C ! ! ! VLAN Config, Device is in VTP Transparent or Primary mode ! ! [{'name': 'default', 'status': 'active', 'vlan_id': '1', 'interfaces': ['Gi0/1', 'Gi0/2', 'Gi0/3', 'Gi1/0', 'Gi1/1', 'Gi1/2', 'Gi1/3', 'Gi2/1', 'Gi2/2', 'Gi2/3', 'Gi3/0', 'Gi3/1', 'Gi3/2', 'Gi3/3'], { 'name': 'MGMT', 'status': 'active', 'vlan_id': '10', 'interfaces': [] }, { 'name': 'CLIENT-B', 'status': 'active', 'vlan_id': '101', 'interfaces': [] }, { 'name': 'CLIENT-CC', 'status': 'active', 'vlan_id': '102', 'interfaces': ['Gi2/0'], { 'name': 'fddi-default', 'status': 'act/unsup', 'vlan_id': '1002', 'interfaces': [] }, { 'name': 'token-ring-default', 'status': 'act/unsup', 'vlan_id': '1003', 'interfaces': [] }, { 'name': 'fddinet-default', 'status': 'act/unsup', 'vlan_id': '1004', 'interfaces': [] }, { 'name': 'trnet-default', 'status': 'act/unsup', 'vlan_id': '1005', 'interfaces': [] }] ! IGNORED VLAN ! vlan 1 ! name default</pre>	<pre>Task execute_neops_nornir_task started Subtask commands_textfsm started Subtask napalm_cli started ***** * IOSv is strictly limited to use for evaluation, demonstration and IOS * * education. IOSv is provided as-is and is not supported by Cisco's * * Technical Advisory Center. Any use or disclosure, in whole or in part, * of the IOSv Software or Documentation to any third party for any * * purposes is expressly prohibited except as otherwise authorized by * * Cisco in writing. * *****</pre>					
▼	sw02.neops.io		No				ios
▼	router1-demo.neops.io		No				ios
▼	dist-sw01.neops.io		Yes				ios

Rows per page:

30 ▼

1-4 of 4

1



Execution Details

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

Settings

powered by
 zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

	State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
		sw02.neops.io		Yes				ios
		config to apply	applied changes				log	
		<pre>! ! Rename VLAN 'CLIENT-CC' -> 'CLIENT-C' ! ! ! ! VLAN Config, Device is in VTP Transparent or Primary mode ! ! [{"name': 'default', 'status': 'active', 'vlan_id': '1', 'interfaces': ['Gi0/1', 'Gi0/2', 'Gi0/3', 'Gi1/0', 'Gi1/1', 'Gi1/2', 'Gi1/3', 'Gi2/1', 'Gi2/2', 'Gi2/3', 'Gi3/0', 'Gi3/1', 'Gi3/2', 'Gi3/3']}, {"name': 'MGMT', 'status': 'active', 'vlan_id': '10', 'interfaces': []}, {"name': 'CLIENT-B', 'status': 'active', 'vlan_id': '101', 'interfaces': []}, {"name': 'CLIENT-CC', 'status': 'active', 'vlan_id': '102', 'interfaces': ['Gi2/0']}, {'name': 'fddi-default', 'status': 'act/unsup', 'vlan_id': '1002', 'interfaces': []}, {"name': 'token-ring-default', 'status': 'act/unsup', 'vlan_id': '1003', 'interfaces': []}, {"name': 'fddinet-default', 'status': 'act/unsup', 'vlan_id': '1004', 'interfaces': []}, {"name': 'trnet-default', 'status': 'act/unsup', 'vlan_id': '1005', 'interfaces': []}] ! IGNORED VLAN ! vlan 1 ! name default</pre>	<pre>!Contextual Config Diffs: vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C</pre>					
		sw01.neops.io		No				ios
		router1-demo.neops.io		No				ios
		dist-sw01.neops.io		Yes				ios

Rows per page:

30

1-4 of 4





Execution Details



Dashboard



Network



Tasks



Providers



Now Running



History



Cron Jobs



Task Tools



powered by
 zebra

TASK EXECUTION OVERVIEW

REMOTE DEVICES

PROCESSING NEOPS ITEMS

State	Hostname	Last Update	Changed	Vendor	Model	OS Release	Platform
Up	sw02.neops.io		Yes				ios
	config to apply	applied changes		log			
	<pre>! ! Rename VLAN 'CLIENT-CC' -> 'CLIENT-C' ! ! VLAN Config, Device is in VTP Transparent or Primary mode ! ! [{'name': 'default', 'status': 'active', 'vlan_id': '1', 'interfaces': ['Gi0/1', 'Gi0/2', 'Gi0/3', 'Gi1/0', 'Gi1/1', 'Gi1/2', 'Gi1/3', 'Gi2/1', 'Gi2/2', 'Gi2/3', 'Gi3/0', 'Gi3/1', 'Gi3/2', 'Gi3/3']}, {'name': 'MGMT', 'status': 'active', 'vlan_id': '10', 'interfaces': []}, {'name': 'CLIENT-B', 'status': 'active', 'vlan_id': '101', 'interfaces': []}, {'name': 'CLIENT-CC', 'status': 'active', 'vlan_id': '102', 'interfaces': ['Gi2/0']}, {'name': 'fddi-default', 'status': 'act/unsup', 'vlan_id': '1002', 'interfaces': []}, {'name': 'token-ring-default', 'status': 'act/unsup', 'vlan_id': '1003', 'interfaces': []}, {'name': 'fddinet-default', 'status': 'act/unsup', 'vlan_id': '1004', 'interfaces': []}, {'name': 'trnet-default', 'status': 'act/unsup', 'vlan_id': '1005', 'interfaces': []}] ! IGNORED VLAN ! vlan 1 ! name default</pre>	<pre>! ! Contextual Config Diffs: vlan 102 +name CLIENT-CC vlan 102 -name CLIENT-C !</pre>	<pre>Task execute_neops_nornir_task started Subtask commands_textfsm started Subtask napalm_cli started ***** * IOSv is strictly limited to use for evaluation, demonstration and IOS * * education. IOSv is provided as-is and is not supported by Cisco's * * Technical Advisory Center. Any use or disclosure, in whole or in part, * of the IOSv Software or Documentation to any third party for any * * purposes is expressly prohibited except as otherwise authorized by * * Cisco in writing. *</pre>				
	sw01.neops.io		No				ios
	router1-demo.neops.io		No				ios
	dist-sw01.neops.io		Yes				ios

Rows per page:

30

1-4 of 4

1

Abstraction - Level appropriate

Abstraction - Level appropriate



Network Engineer

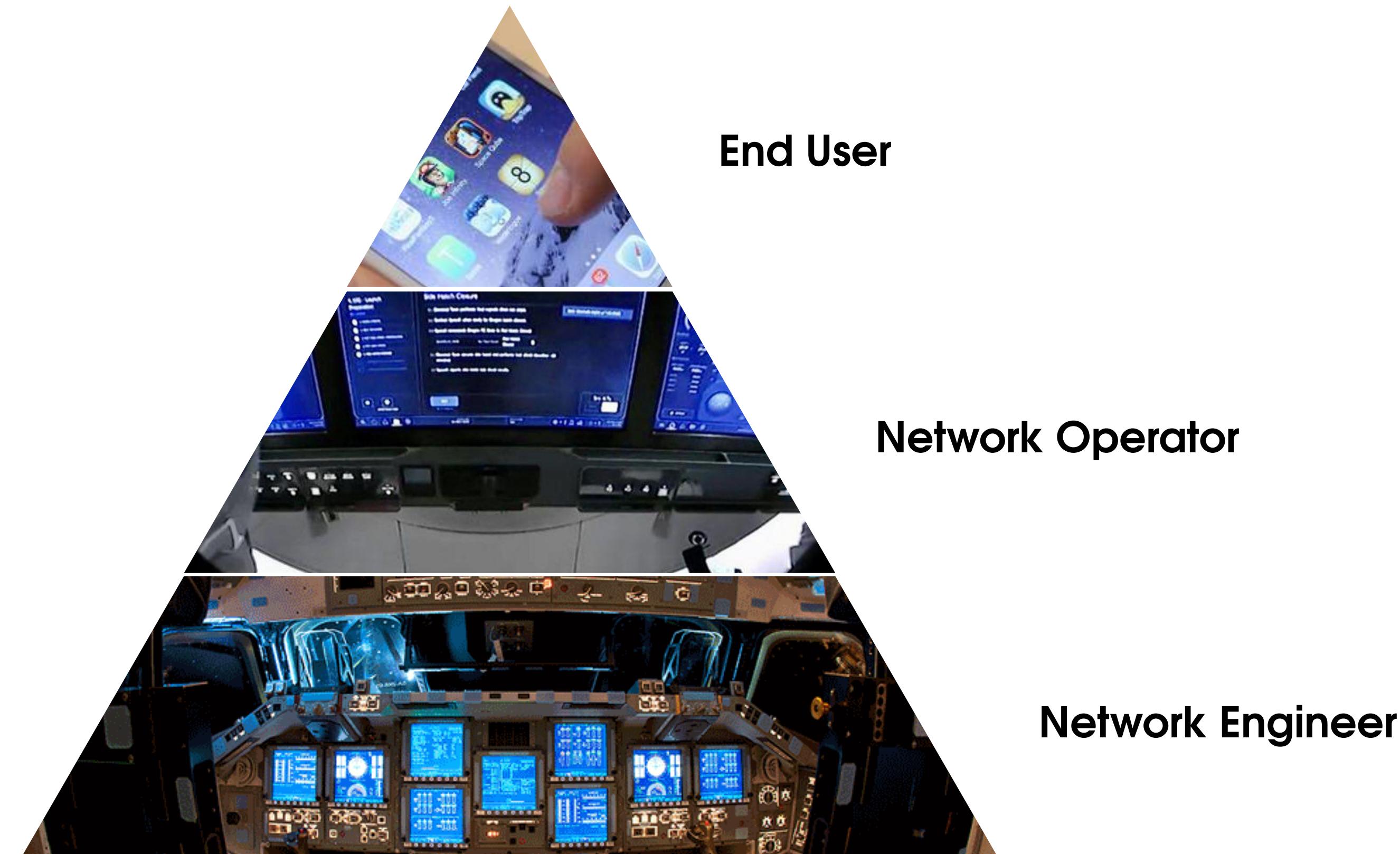
Abstraction - Level appropriate



Network Operator

Network Engineer

Abstraction - Level appropriate





Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```



Test TextFSM Templates

0 ➔ demo

Dashboard

Network

Tasks

Providers

Now Running

History

Cron Jobs

Task Tools

Help

powered by
 zebra
commit-eba056a

Command Output

```
1 Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(CML_NIGHTLY_20180619)FLO_DSGS7, EARLY DEPLOYMENT DEVELOPMENT BUILD, synced to V152_6_0_81_E
2 Technical Support: http://www.*****.com/techsupport
3 Copyright (c) 1986-2018 by Cisco Systems, Inc.
4 Compiled Tue 19-Jun-18 06:06 by mmn
5
6
7 ROM: Bootstrap program is IOSv
8
9 sw02 uptime is 1 week, 1 day, 9 hours, 22 minutes
10 System returned to ROM by reload
11 System image file is "flash0:/vios_12-adventureprisek9-m"
12 Last reload reason: Unknown reason
```

PLAIN TEXTFSM TEMPLATE

USE NTC TEMPLATE

TextFSM Template

```
1 Value VERSION (.+?)
2 Value UPTIME (.+)
3
4 Start
5 ^.*Software\s+\),\sVersion\s${VERSION},
6 ^\s*\S+\s+uptime\s+is\s+$UPTIME
```

RENDERS

Result

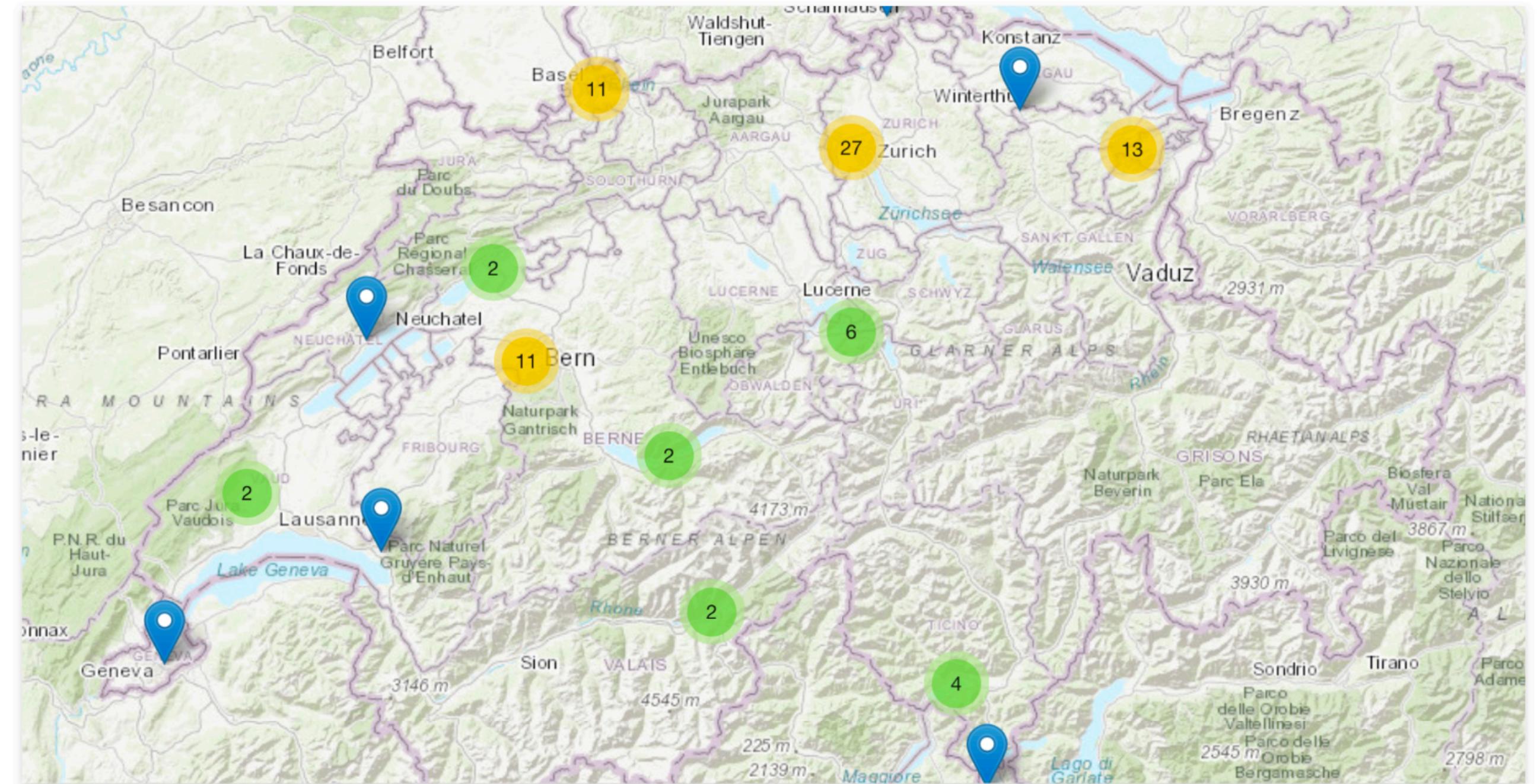
```
1 [
2 {
3   "version": "15.2(CML_NIGHTLY_20180619)FLO_DSGS7",
4   "uptime": "1 week, 1 day, 9 hours, 22 minutes"
5 }
6 ]
```

[Search a location](#)[My Network](#)

Welcome back to neops.io! Check your network's status below and select a location for more details.

Select a location or hostname

Search task



Task xyz

Task description for xyz

 Search entity I
 Search entity II[Run](#)

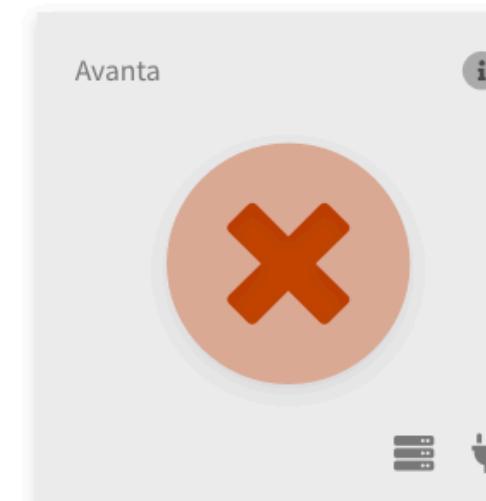
Reboot AP

Power Cycle Interfaces with APs

 Select Device
 Select Interface[Run](#)

Configure Port

Interface and configuration to apply

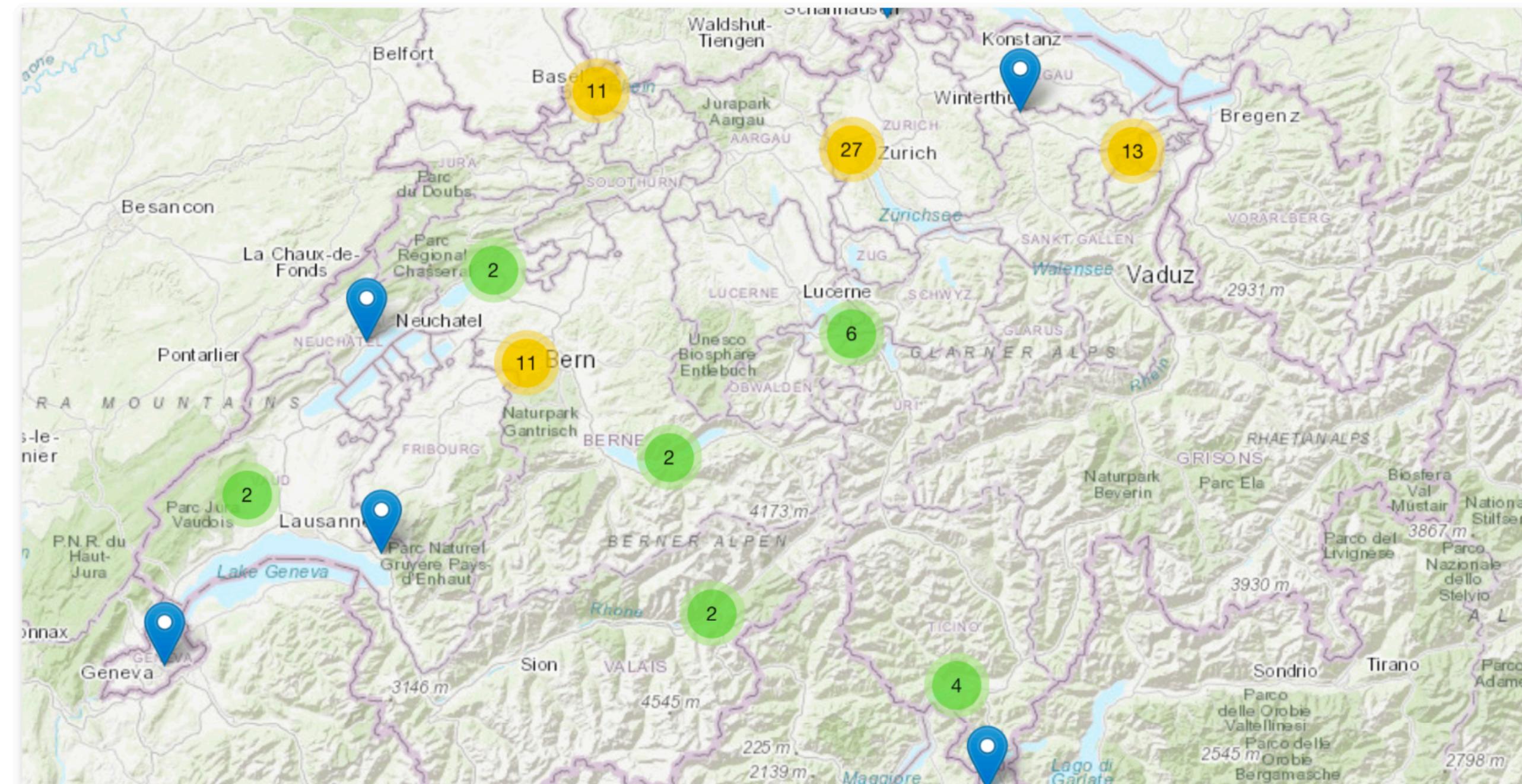
 Select Device Select Port
 Select Configuration[Apply Configuration](#)[Help Center](#)[Settings](#)

[Search a location](#)[My Network](#)

Welcome back to neops.io! Check your network's status below and select a location for more details.

Select a location or hostname

Search task



Task xyz

Task description for xyz

 Search entity I
 Search entity II[Run](#)

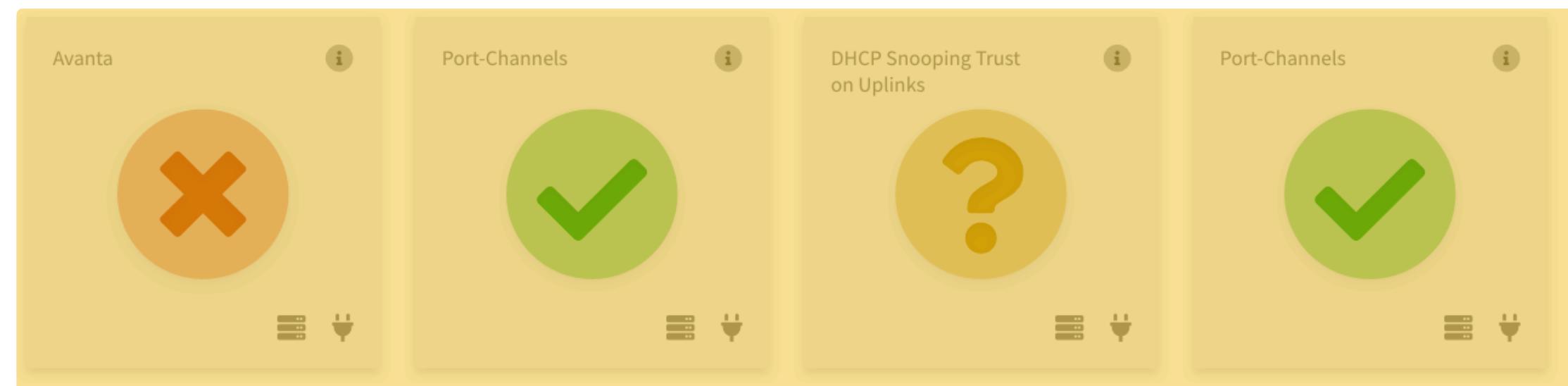
Reboot AP

Power Cycle Interfaces with APs

 Select Device
 Select Interface[Run](#)

Configure Port

Interface and configuration to apply

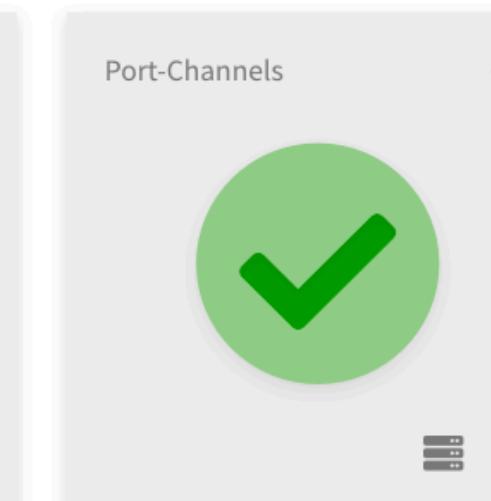
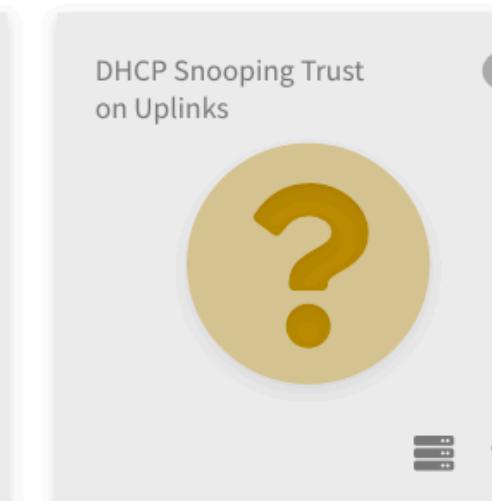
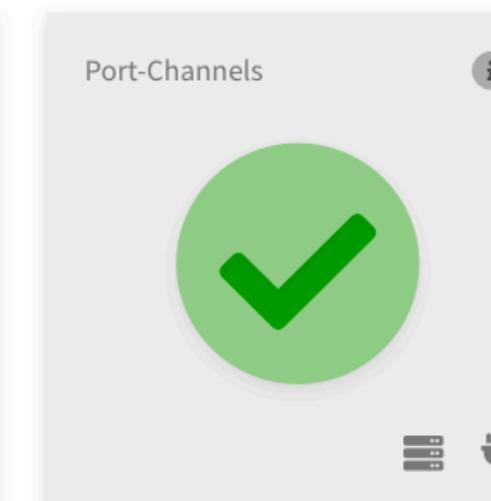
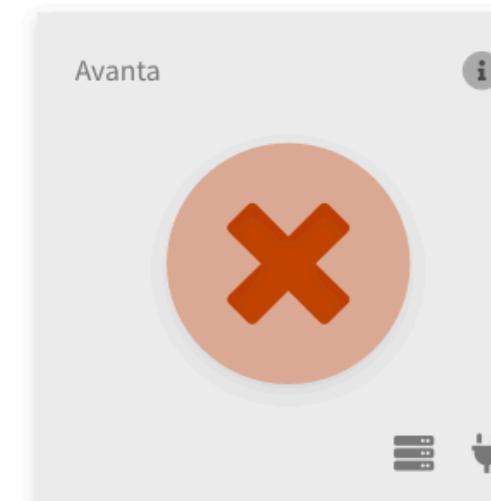
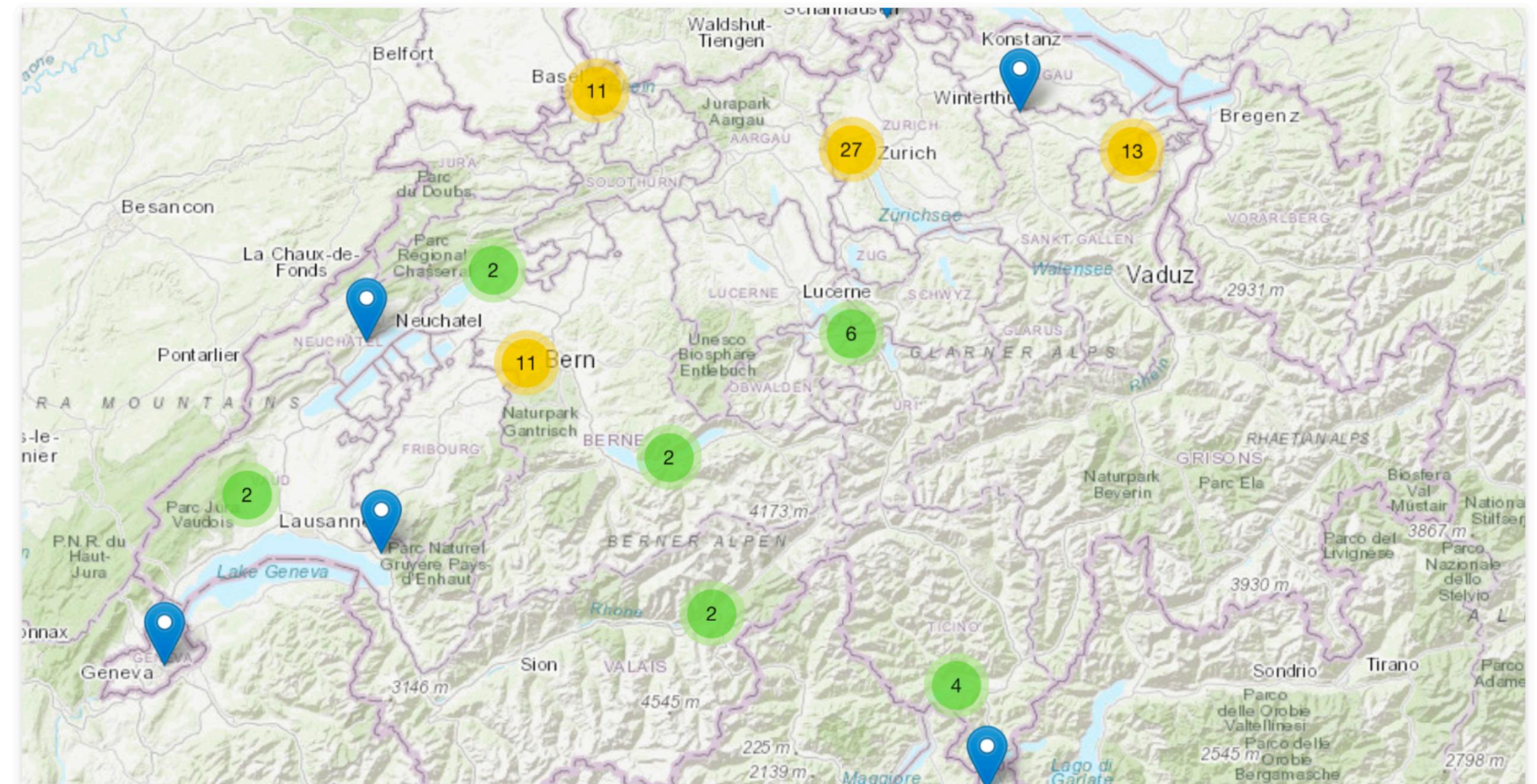
 Select Device Select Port
 Select Configuration[Apply Configuration](#)[Help Center](#)[Settings](#)

[Search a location](#)[My Network](#)

Welcome back to neops.io! Check your network's status below and select a location for more details.

Select a location or hostname

Search task



Task xyz

Task description for xyz

Select Device I

Select Entity II

[Run](#)

Reboot AP

Power Cycle Interfaces with APs

Select Device

Select Interface

[Run](#)

Configure Port

Interface and configuration to apply

Select Device

Select Port

Select Configuration

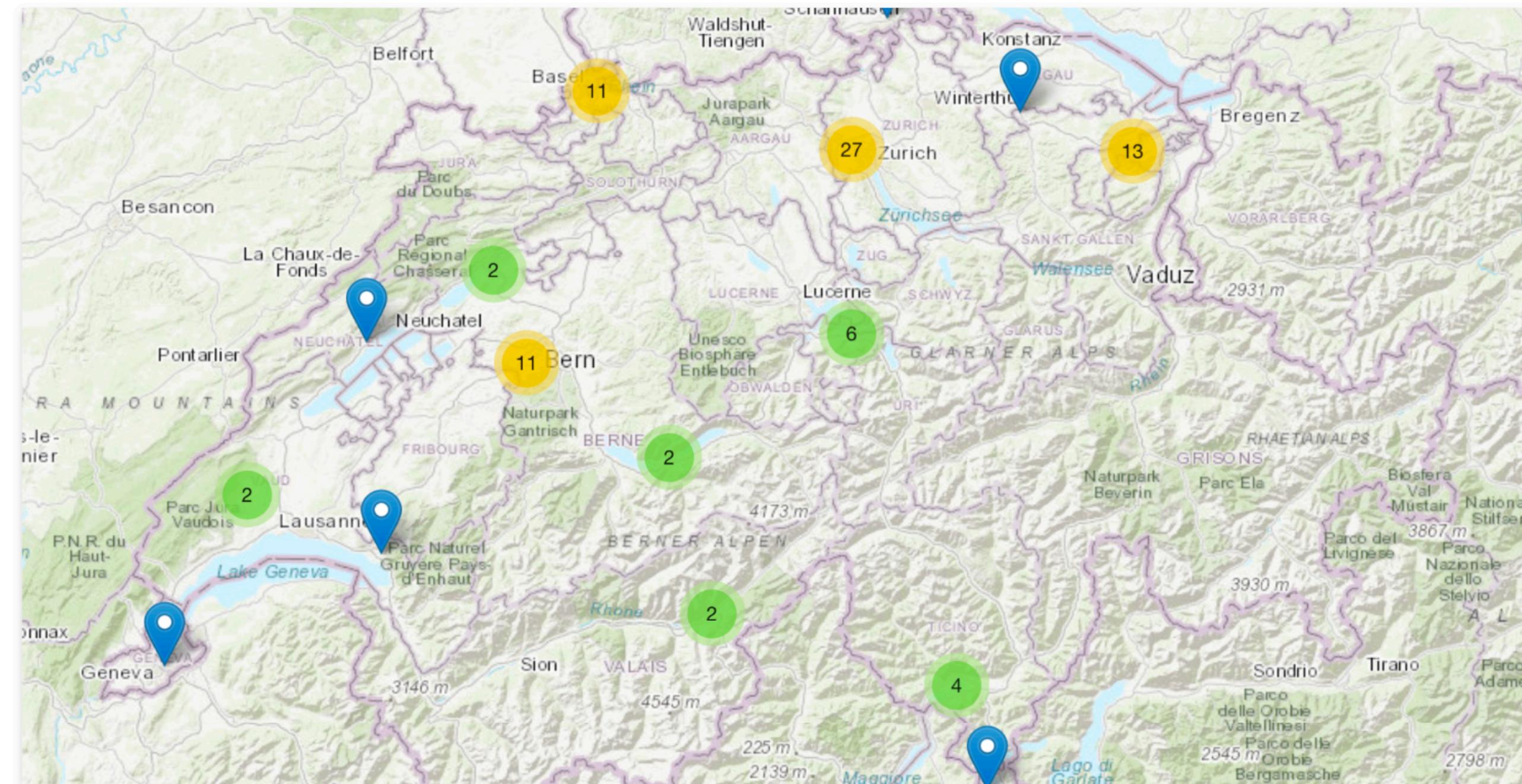
[Apply Configuration](#)

[Help Center](#)[Settings](#)

[Search a location](#)[My Network](#)

Welcome back to neops.io! Check your network's status below and select a location for more details.

Select a location or hostname



Search task

Task xyz

Task description for xyz

Search entity I

Search entity II

[Run](#)**Reboot AP**

Power Cycle Interfaces with APs

Select Device

Select Interface

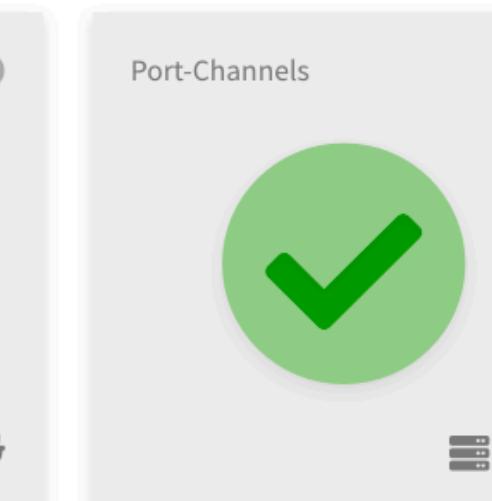
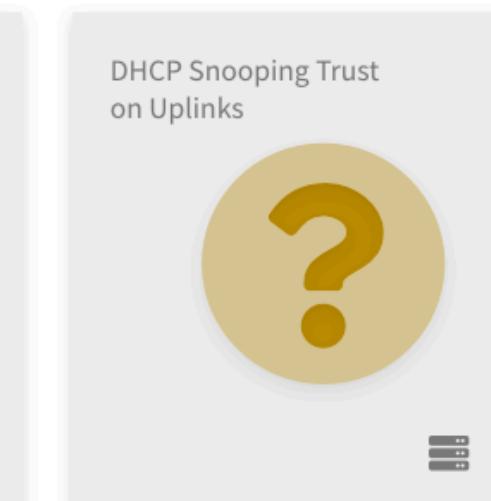
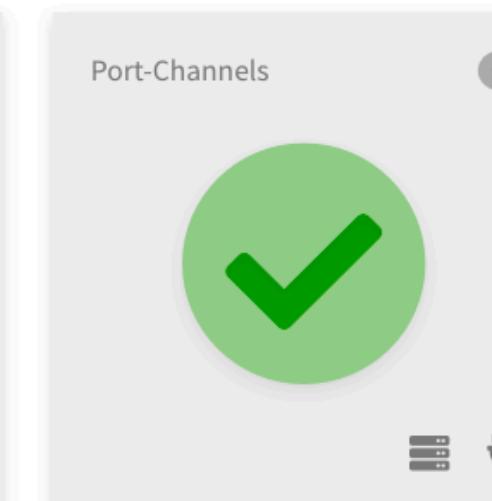
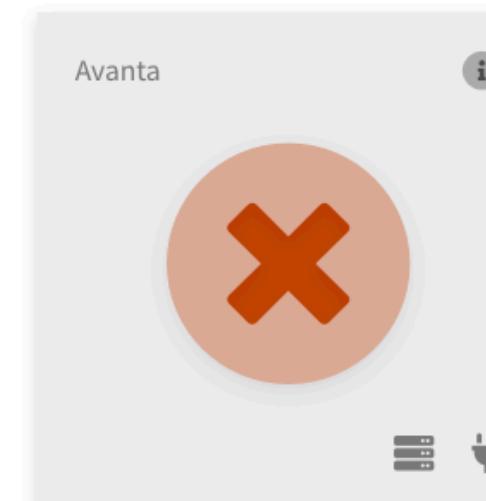
[Run](#)**Configure Port**

Interface and configuration to apply

Select Device

Select Port

Select Configuration

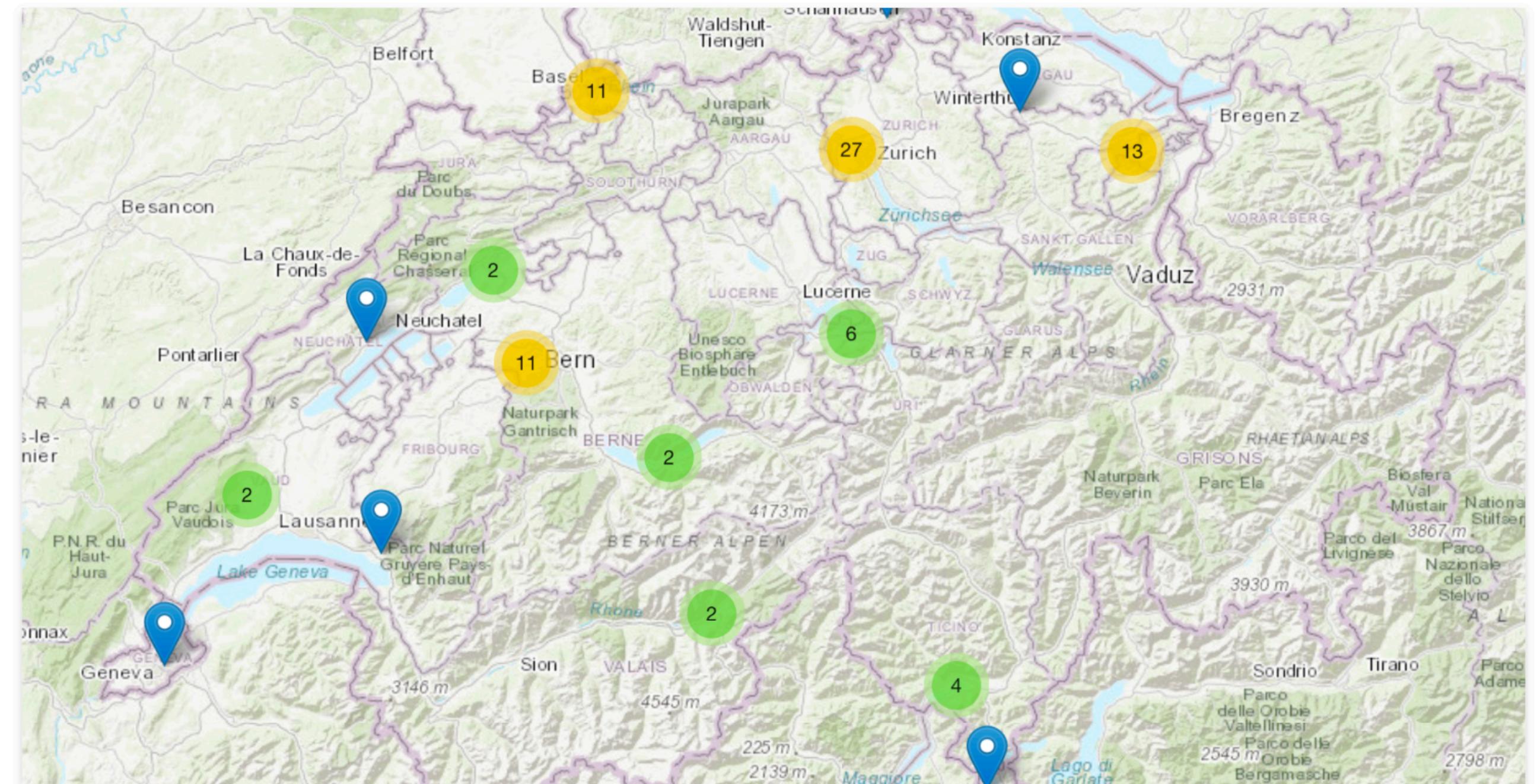
[Apply Configuration](#)[Help Center](#)[Settings](#)

[Search a location](#)[My Network](#)

Welcome back to neops.io! Check your network's status below and select a location for more details.

Select a location or hostname

Search task



Task xyz

Task description for xyz

Search entity I

Search entity II

[Run](#)

Reboot AP

Power Cycle Interfaces with APs

Select Device

Select Interface

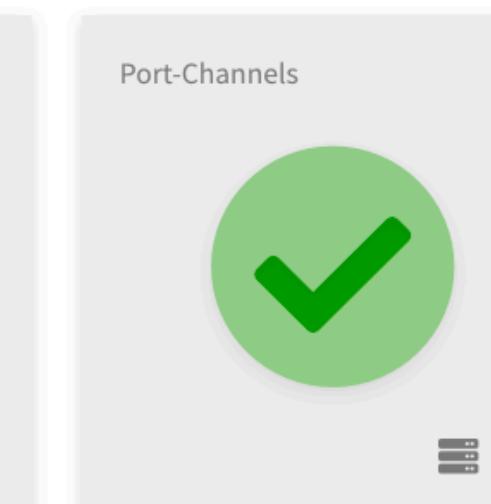
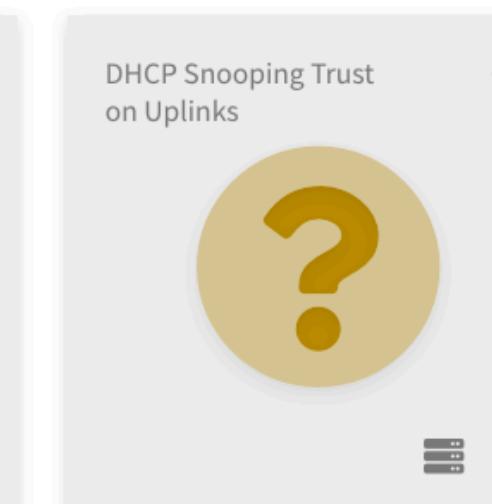
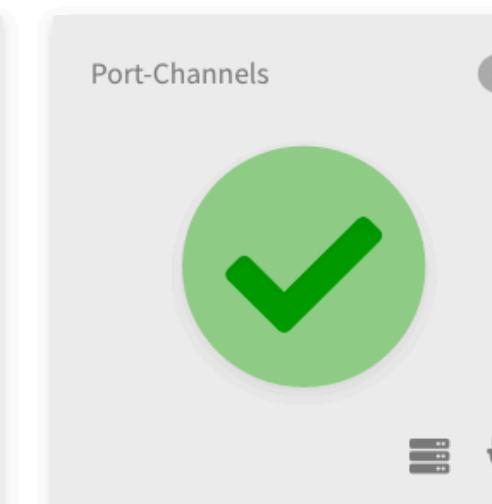
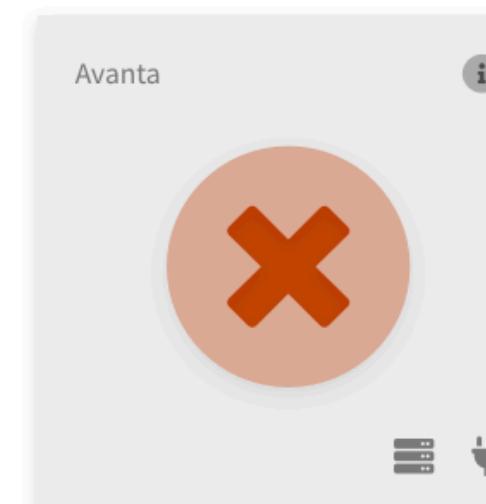
[Run](#)

Configure Port

Interface and configuration to apply

Select Device Select Port

Select Configuration

[Apply Configuration](#)[Help Center](#)[Settings](#)

Takeaways

- ⌘ UX matters!
- ⌘ Take small steps and start/continue automation

my NetDevOps journey, to be continued :)



next... ?



Thank you!

& see you soon

Simon Obi

simon.obi@zebbra.ch

Network Digitalization Officer
@ zebbra AG