NETWORK AUTOMATION TUTORIAL

David Barroso <dbarrosop@dravetech.com>

HOW TO NAVIGATE THIS TUTORIAL?

Press `left` and `right` to change sections. Press `up` and `down` to move within sections. Press `?` anytime for help.

Twitter | Linkedin | Github
@dbarrosop

- **Network Systems Engineer** at Fastly
- Previously:
  - **Network Engineer** at Spotify
  - **Network Engineer** at NTT
  - **Network & Systems Engineer** at Atlas IT
- Creator of:
  - N.A.P.A.L.M.
  - SDN Internet Router

AGENDA

- Lab preparation
- Hello world
- Abstract Vendor Interfaces
- Abstract Vendor Configurations
- Data-Driven Configuration
- Data-Driven Configuration with a Backend
- Abstractions

LAB PREPARATION

Link to the code on github

OBJECTIVES

- Install all of the python requirements
- Start the devices on your lab

INSTALLING PYTHON REQUIREMENTS

```
➜  workspace git clone git@github.com:dravetech/network-tutorials.git
Cloning into 'network-tutorials'...
remote: Counting objects: 277, done.
remote: Compressing objects: 100% (120/120), done.
remote: Total 277 (delta 33), reused 0 (delta 0), pack-reused 151
Receiving objects: 100% (277/277), 1.77 MiB | 932.00 KiB/s, done.
Resolving deltas: 100% (73/73), done.
Checking connectivity... done.
➜  workspace cd network-tutorials
➜  network-tutorials git:(master) virtualenv venv
New python executable in /Users/dbarroso/workspace/network-tutorials/venv/bin/python2.7
Also creating executable in /Users/dbarroso/workspace/network-tutorials/venv/bin/python
Installing setuptools, pip, wheel...done.
➜  network-tutorials git:(master) . venv/bin/activate
(venv) ➜  network-tutorials git:(master) pip install -U -r requirements.txt
Collecting PyYAML (from -r requirements.txt (line 1))
Collecting ansible==2.1.1.0 (from -r requirements.txt (line 2))
Collecting junos-eznc (from -r requirements.txt (line 3))
Collecting napalm (from -r requirements.txt (line 4))
Collecting pyeapi (from -r requirements.txt (line 5))
Collecting pylama (from -r requirements.txt (line 6))
  Using cached pylama-7.1.0-py2.py3-none-any.whl
Collecting pynsot (from -r requirements.txt (line 7))
Collecting invoke (from -r requirements.txt (line 8))
  Using cached invoke-0.13.0-py2-none-any.whl
Collecting yamllint (from -r requirements.txt (line 9))
  Downloading yamllint-1.5.0.tar.gz (76kB)
    100% |████████████████████████████████| 81kB 1.4MB/s
```

STARTING THE LAB

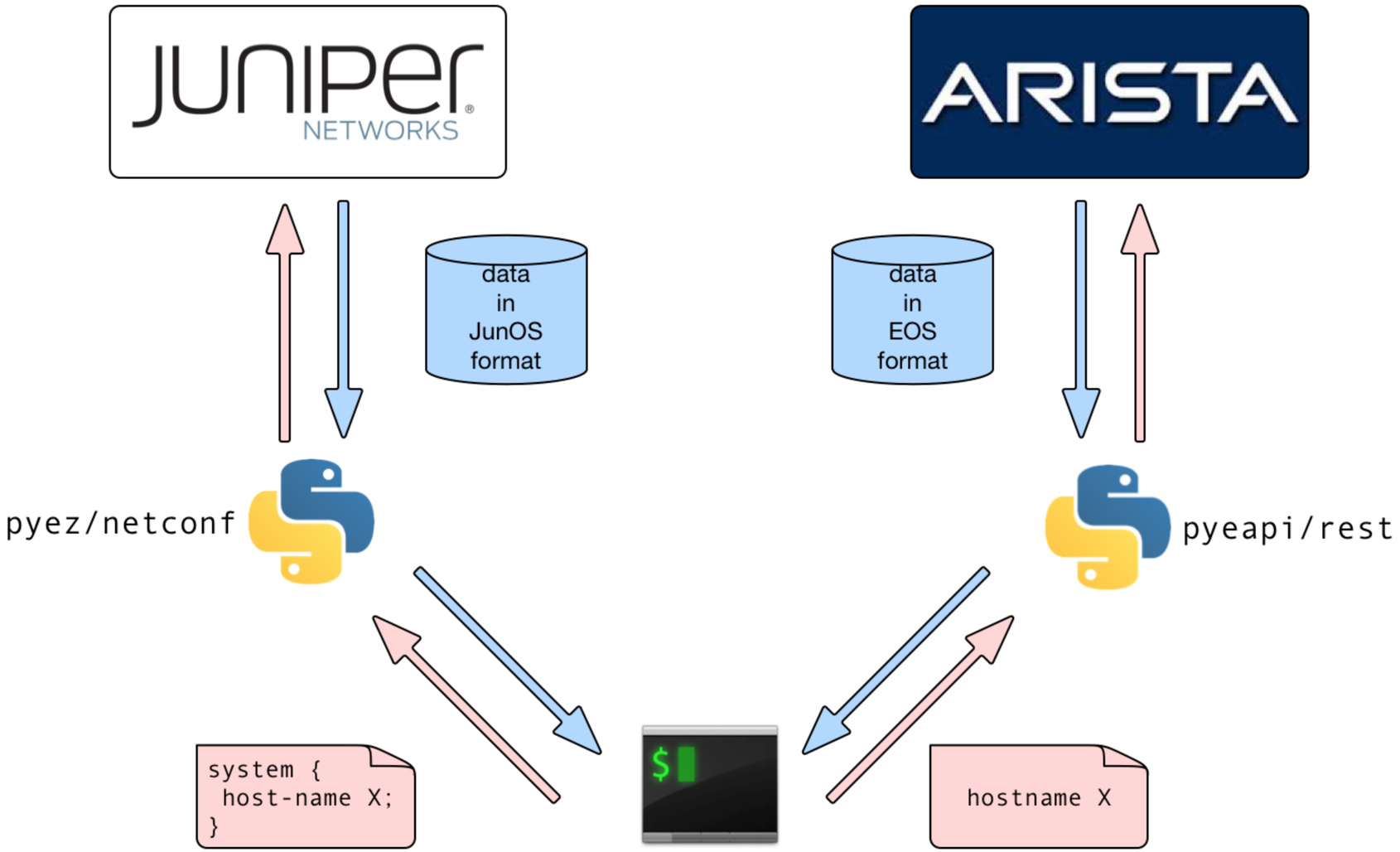
```
(venv) ➜  network-tutorials git:(master) cd labs/lab1
(venv) ➜  lab1 git:(master) vagrant up
Adding ssh key to the ssh agent
ssh-add /opt/vagrant/embedded/gems/gems/vagrant-1.8.1/keys/vagrant
Identity added: /opt/vagrant/embedded/gems/gems/vagrant-1.8.1/keys/vagrant (/opt/vagrant/embedded/gems/gems/vagrant-1.8.1/key
/Users/dbarroso/.vagrant.d/boxes/vEOS-lab-4.16.6M/0/virtualbox/Vagrantfile:15: warning: already initialized constant VAGRANTI
/Users/dbarroso/workspace/network-tutorials/labs/lab1/Vagrantfile:6: warning: previous definition of VAGRANTFILE_API_VERSION
Bringing machine 'veos' up with 'virtualbox' provider...
Bringing machine 'junos' up with 'virtualbox' provider...
==> veos: Importing base box 'vEOS-lab-4.16.6M'...
==> veos: Matching MAC address for NAT networking...
==> veos: Setting the name of the VM: lab1_veos_1476088690154_98596
==> veos: Clearing any previously set network interfaces...
==> veos: Preparing network interfaces based on configuration...
    veos: Adapter 1: nat
    veos: Adapter 2: intnet
    veos: Adapter 3: intnet
==> veos: Forwarding ports...
    veos: 443 (guest) => 12443 (host) (adapter 1)
    veos: 22 (guest) => 2222 (host) (adapter 1)
==> veos: Booting VM...
==> veos: Waiting for machine to boot. This may take a few minutes...
    veos: SSH address: 127.0.0.1:2222
    veos: SSH username: root
    veos: SSH auth method: private key
```

1 - HELLO WORLD

Link to the code on github

OBJECTIVES

Learn how to connect to different devices, gather data such as the serial number or the device model and how to manipulate the device configuration using their respective APIs.


JUNIPER
NETWORKS
ARISTA
data
in
JunOS
format
data
in
EOS
format
pyez/netconf
pyeapi/rest
system {
host-name X;
}
$
hostname X

JUNOS - RETRIEVING FACTS

```
>>> from jnpr.junos import Device
>>> import pprint
>>> pp = pprint.PrettyPrinter(indent=4)
>>>
>>> junos = Device(host='127.0.0.1', user='vagrant', port=12203)
>>> junos.open()
Device(127.0.0.1)
>>> junos_facts = junos.facts
>>> pp.pprint(junos_facts)
{   '2RE': False,
    'HOME': '/cf/var/home/vagrant',
    'RE0': {   'last_reboot_reason': 'Router rebooted after a normal shutdown.',
               'model': 'FIREFLY-PERIMETER RE',
               'status': 'Testing',
               'up_time': '3 minutes, 36 seconds'},
    'domain': None,
    'fqdn': 'vsrx',
    'hostname': 'vsrx',
    'model': 'FIREFLY-PERIMETER',
    'personality': 'SRX_BRANCH',
    'serialnumber': '5b2b599a283b',
    'srx_cluster': False,
    'version': '12.1X47-D20.7',
    'version_info': junos.version_info(major=(12, 1), type=X, minor=(47, 'D', 20), build=7),
    'virtual': True}
>>> junos.close()
```

EOS - RETRIEVING FACTS

```
>>> import pyeapi
>>> import pprint
>>> pp = pprint.PrettyPrinter(indent=4)
>>> connection = pyeapi.client.connect(transport='https', host='127.0.0.1', username='vagrant',
...                                    password='vagrant', port=12443,)
>>> eos = pyeapi.client.Node(connection)
>>> eos_facts = eos.run_commands(['show version'])
>>> pp.pprint(eos_facts)
[   {   u'architecture': u'i386',
        u'bootupTimestamp': 1475859369.15,
        u'hardwareRevision': u'',
        u'internalBuildId': u'e796e94c-ba3b-4355-afcf-ef0abfbfaee3',
        u'internalVersion': u'4.16.6M-3205780.4166M',
        u'isIntlVersion': False,
        u'memFree': 56204,
        u'memTotal': 1897596,
        u'modelName': u'vEOS',
        u'serialNumber': u'',
        u'systemMacAddress': u'08:00:27:52:27:ce',
        u'version': u'4.16.6M'}]
```

JUNOS - CHANGING THE HOSTNAME

```
>>> from jnpr.junos import Device
>>> from jnpr.junos.utils.config import Config
>>>
>>> junos = Device(host='127.0.0.1', user='vagrant', port=12203)
>>> junos.open()
Device(127.0.0.1)
>>>
>>> print(junos.facts['hostname'])
vsrx
>>> junos.bind(cu=Config)
>>> junos.cu.lock()
True
>>> junos.cu.load("system {host-name new-hostname;}", format="text", merge=True)
>>> junos.cu.commit()
True
>>> junos.cu.unlock()
True
>>> junos.facts_refresh()
>>> print(junos.facts['hostname'])
new-hostname
>>> junos.close()
```

EOS - CHANGING THE HOSTNAME

```
>>> import pyeapi
>>> connection = pyeapi.client.connect(
...     transport='https',
...     host='127.0.0.1',
...     username='vagrant',
...     password='vagrant',
...     port=12443,
... )
>>> eos = pyeapi.client.Node(connection)
>>> print(eos.run_commands(['show hostname'])[0]['hostname'])
localhost
>>> eos.run_commands(['configure', 'hostname a-new-hostname'])
[{}, {}]
>>> print(eos.run_commands(['show hostname'])[0]['hostname'])
a-new-hostname
```

SUMMARY

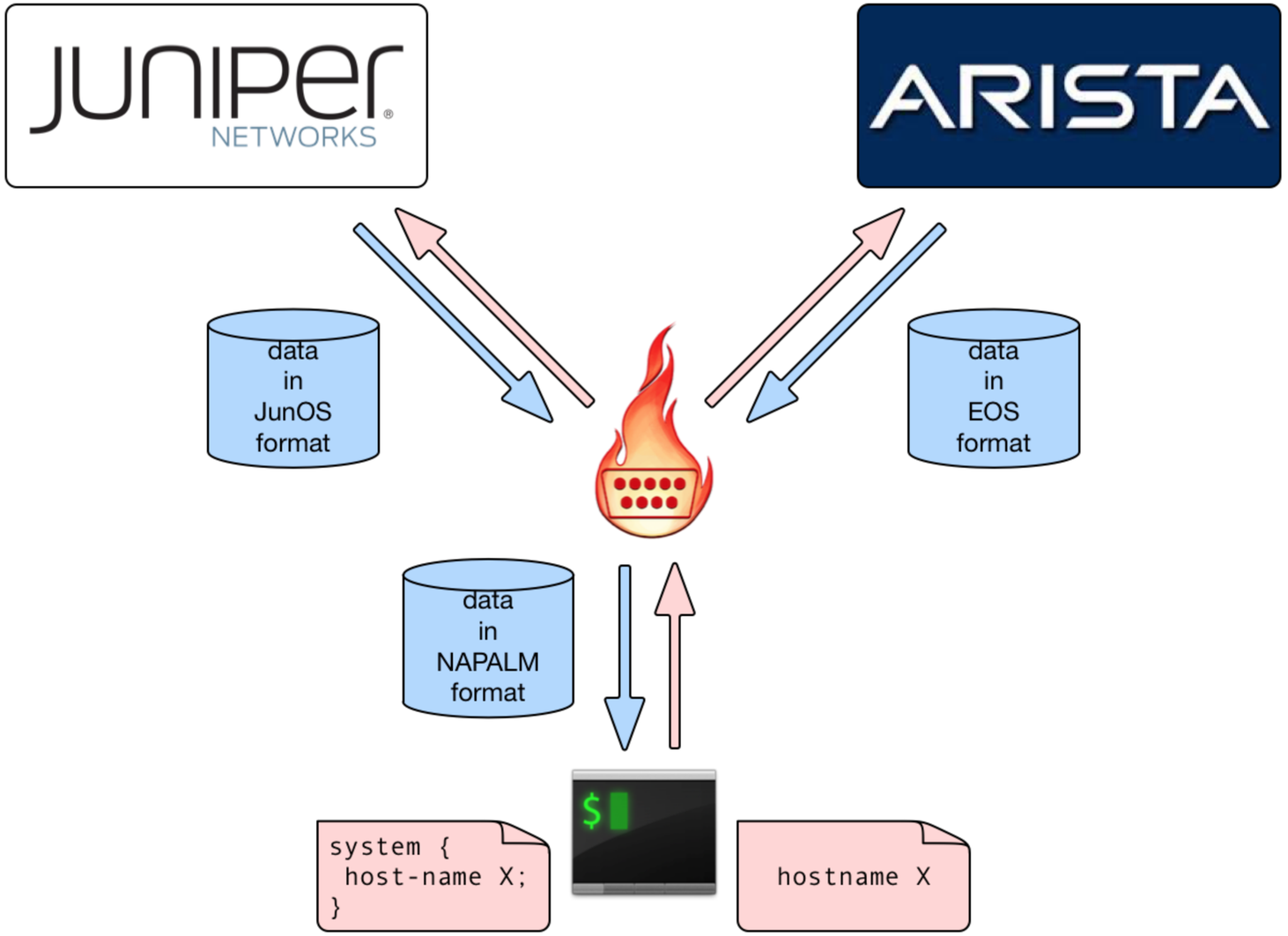
- Different devices have different APIs
- Different APIs behave differently so code is different
- Similar methods may return similar but different data structures and with different formats

2 - ABSTRACT VENDOR INTERFACES

Link to the code on github

OBJECTIVES

Learn how to use napalm to **abstract vendor's APIs**. The code we will produce in this section will have the same functionality as in the previous section, however, it will work across different vendors.


JUNIPER
NETWORKS
ARISTA
data
in
JunOS
format
data
in
EOS
format
data
in
NAPALM
format
system {
host-name X;
}
$
hostname X

NAPALM - RETRIEVING FACTS

```
>>> from napalm_base import get_network_driver
>>> import pprint
>>> pp = pprint.PrettyPrinter(indent=4)
>>>
>>> junos_driver =  get_network_driver('junos')
>>> eos_driver =  get_network_driver('eos')
>>>
>>> junos_configuration = {
...     'hostname': '127.0.0.1',
...     'username': 'vagrant',
...     'password': '',
...     'optional_args': {'port': 12203}
... }
>>>
>>> eos_configuration = {
...     'hostname': '127.0.0.1',
...     'username': 'vagrant',
...     'password': 'vagrant',
...     'optional_args': {'port': 12443}
... }
>>>
```

```
>>> with junos_driver(**junos_configuration) as junos:
...     pp.pprint(junos.get_facts())
...
{   'fqdn': u'new-hostname',
    'hostname': u'new-hostname',
    'interface_list': [   'ge-0/0/0', 'gr-0/0/0',
                          'lt-0/0/0', 'mt-0/0/0', 'vlan'],
    'model': u'FIREFLY-PERIMETER',
    'os_version': u'12.1X47-D20.7',
    'serial_number': u'5b2b599a283b',
    'uptime': 1080,
    'vendor': u'Juniper'}
>>>
>>> with eos_driver(**eos_configuration) as eos:
...     pp.pprint(eos.get_facts())
...
{   'fqdn': u'a-new-hostname',
    'hostname': u'a-new-hostname',
    'interface_list': [u'Ethernet1', u'Ethernet2',
                       u'Management1'],
    'model': u'vEOS',
    'os_version': u'4.16.6M-3205780.4166M',
    'serial_number': u'',
    'uptime': 1217,
    'vendor': u'Arista'}
```

NAPALM - CHANGING THE HOSTNAME

```
>>> from napalm_base import get_network_driver
>>> junos_driver =  get_network_driver('junos')
>>> eos_driver =  get_network_driver('eos')
>>>
>>> junos_configuration = {
...     'hostname': '127.0.0.1',
...     'username': 'vagrant',
...     'password': '',
...     'optional_args': {'port': 12203}
... }
>>>
>>> eos_configuration = {
...     'hostname': '127.0.0.1',
...     'username': 'vagrant',
...     'password': 'vagrant',
...     'optional_args': {'port': 12443}
... }
>>>
>>> def change_configuration(device, configuration):
...     device.load_merge_candidate(config=configuration)
...     print(device.compare_config())
...     device.commit_config()
...
>>>
```

NAPALM - CHANGING THE HOSTNAME (CONT'D)

```
>>> with junos_driver(**junos_configuration) as junos:
...     change_configuration(
...             junos,
...             "system {host-name yet-another-hostname;}
...     )
...
[edit system]
-  host-name new-hostname;
+  host-name yet-another-hostname;
>>>
```

```
>>> with junos_driver(**junos_configuration) as junos:
...     change_configuration(
...             junos,
...             "system {host-name yet-another-hostname;}
...     )
...

>>>
```

```
>>> with eos_driver(**eos_configuration) as eos:
...     change_configuration(
...             eos,
...             'hostname yet-another-hostname'
...     )
...
@@ -8,7 +8,7 @@
 !
 transceiver qsfp default-mode 4x10G
 !
-hostname a-new-hostname
+hostname yet-another-hostname
 !
 spanning-tree mode mstp
 !
>>>
```

```
>>> with eos_driver(**eos_configuration) as eos:
...     change_configuration(
...             eos,
...             'hostname yet-another-hostname'
...     )
...

>>>
```

Changes are idempotent

SUMMARY

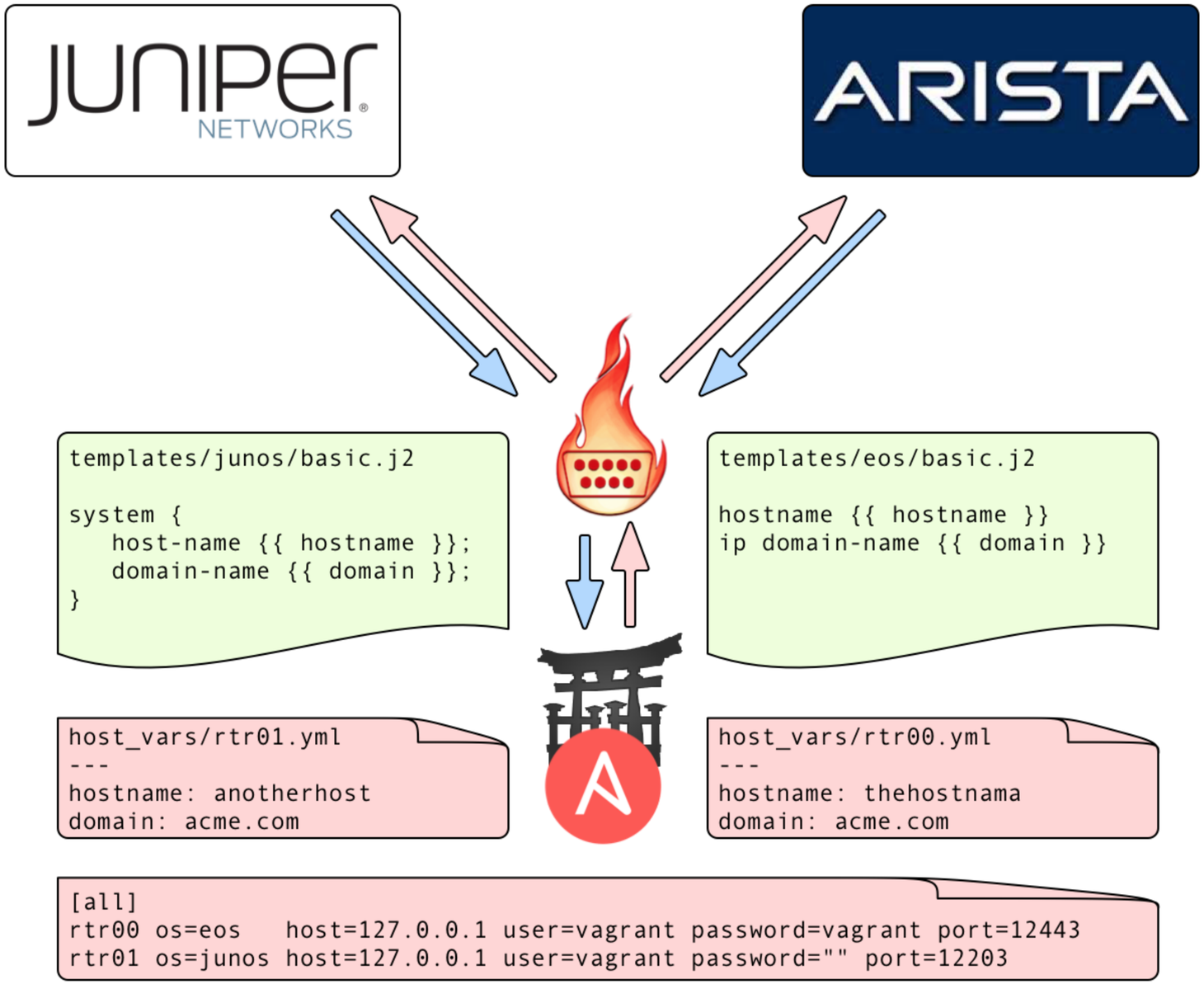
- NAPALM allows you to write code that works across vendors; workflows and data retrieval are unified across platforms
- Configuration syntax is still vendor dependent (YANG support is under progress)
- Configuration changes are idempotent

3 - ABSTRACT VENDOR CONFIGURATIONS

Link to the code on github

OBJECTIVES

Learn how to use `ansible` and `jinja2` templates to **abstract configuration syntax** and focus on the parameters to change


JUNIPER
NETWORKS
ARISTA
templates/junos/basic.j2
system {
host-name {{ hostname }};
domain-name {{ domain }};
}
templates/eos/basic.j2
hostname {{ hostname }}
ip domain-name {{ domain }}
host_vars/rtr01.yml
---
hostname: anotherhost
domain: acme.com
host_vars/rtr00.yml
---
hostname: thehostnama
domain: acme.com
[all]
rtr00 os=eos host=127.0.0.1 user=vagrant password=vagrant port=12443
rtr01 os=junos host=127.0.0.1 user=vagrant password="" port=12203

ANSIBLE - GETTING INFORMATION - PLAYBOOK

```yaml
---
- name: "Get facts"
  hosts: all
  connection: local
  gather_facts: no
  vars:

  tasks:
    - name: "get facts from device"
      napalm_get_facts:
        hostname: "{{ host }}"
        username: "{{ user }}"
        dev_os: "{{ os }}"
        password: "{{ password }}"
        optional_args:
          port: "{{ port }}"
        filter: ['facts']
      register: napalm_facts
    - name: Facts
      debug:
        msg: "{{ napalm_facts.ansible_facts.facts|to_nice_json }}"
      tags: [print_action]
```

ANSIBLE - GETTING INFORMATION - RUN (1)

```
(network-tutorials) ➜  tutorial-3-abstract-vendor-configuration git:(master) ✗ ansible-playbook playbook_facts.yml
..
# Facts ***************************************************************************************************************
  * rtr00                       - changed=False ---------------------------------------------
    {
        "fqdn": "yet-another-hostname",
        "hostname": "yet-another-hostname",
        "interface_list": [
            "Ethernet1",
            "Ethernet2",
            "Management1"
        ],
        "model": "vEOS",
        "os_version": "4.16.6M-3205780.4166M",
        "serial_number": "",
        "uptime": 267325,
        "vendor": "Arista"
    }
  * rtr01                       - changed=False ---------------------------------------------
    {
        "fqdn": "yet-another-hostname",
        "hostname": "yet-another-hostname",
```

ANSIBLE - GETTING INFORMATION - RUN (2)

```
        "vendor": "Arista"
    }
* rtr01                      - changed=False ------------------------------------------------
    {
        "fqdn": "yet-another-hostname",
        "hostname": "yet-another-hostname",
        "interface_list": [
            "ge-0/0/0",
            "gr-0/0/0",
            "ge-0/0/1",
            "ge-0/0/2",
            ".local.",
        ],
        "model": "FIREFLY-PERIMETER",
        "os_version": "12.1X47-D20.7",
        "serial_number": "de219312ae14",
        "uptime": 265200,
        "vendor": "Juniper"
    }

# STATS ************************************************************************************************
rtr00     : ok=2 changed=0       failed=0        unreachable=0
rtr01     : ok=2 changed=0       failed=0        unreachable=0
```

ANSIBLE - CHANGING CONFIG - DATA

```
➜ cat hosts
[all]
rtr00 os=eos   host=127.0.0.1 user=vagrant password=vagrant port=12443
rtr01 os=junos host=127.0.0.1 user=vagrant password="" port=12203
```

```
➜ cat group_vars/all.yml
---
ansible_python_interpreter: "/usr/bin/env python"

domain: acme.com
```

```
➜ cat host_vars/rtr00.yml
---
hostname: thehostnama
```

```
➜ cat host_vars/rtr01.yml
---
hostname: another-host
```

ANSIBLE - CHANGING CONFIG - TEMPLATES

```
➜ cat templates/eos/simple.j2
hostname {{ hostname }}
ip domain-name {{ domain }}
```

```
➜ cat templates/junos/simple.j2
system {
  host-name {{ hostname}};
  domain-name {{ domain }};
}
```

ANSIBLE - CHANGING CONFIG - PLAYBOOK

```yaml
---
  tasks:
    - name: A simple template with some configuratic
      template:
        src: "{{ os }}/simple.j2"
        dest: "{{ host_dir }}/simple.conf"
      changed_when: False
      when: commit_changes == 0
    - name: Load configuration into the device
      napalm_install_config:
        hostname: "{{ host }}"
        username: "{{ user }}"
        dev_os: "{{ os }}"
        password: "{{ password }}"
        optional_args:
          port: "{{ port }}"
        config_file: "{{ host_dir }}/simple.conf"
        commit_changes: "{{ commit_changes }}"
        replace_config: false
        get_diffs: True
        diff_file: "{{ host_dir }}/diff"
      tags: [print_action]
```

ANSIBLE - CHANGING CONFIG - DRY-RUN

```
(network-tutorials) ➜  tutorial-3-abstract-vendor-configuration git:(master) ✗ ansible-playbook playbook_configure.yml -C
......
# Load configuration into the device ***************************************************************************
  * rtr00                       - changed=True ---------------------------------------------------
    @@ -8,7 +8,8 @@
     !
     transceiver qsfp default-mode 4x10G
     !
    -hostname yet-another-hostname
    +hostname thehostnama
    +ip domain-name acme.com
     !
     spanning-tree mode mstp
     !
  * rtr01                       - changed=True ---------------------------------------------------
    [edit system]
    -  host-name yet-another-hostname;
    +  host-name another-host;
    +  domain-name acme.com;

# STATS ********************************************************************************************************
rtr00    : ok=4 changed=1       failed=0        unreachable=0
rtr01    : ok=4 changed=1       failed=0        unreachable=0
```

ANSIBLE - CHANGING CONFIG - COMMIT

```
(network-tutorials) ➜  tutorial-3-abstract-vendor-configuration git:(master) ✗ ansible-playbook playbook_configure.yml
......
# Load configuration into the device ********************************************************************************
  * rtr00                       - changed=True ----------------------------------------------------
    @@ -8,7 +8,8 @@
     !
     transceiver qsfp default-mode 4x10G
     !
    -hostname yet-another-hostname
    +hostname thehostnama
    +ip domain-name acme.com
     !
     spanning-tree mode mstp
     !
  * rtr01                       - changed=True ----------------------------------------------------
    [edit system]
    -  host-name yet-another-hostname;
    +  host-name another-host;
    +  domain-name acme.com;

# STATS ************************************************************************************************************
rtr00    : ok=4 changed=1       failed=0        unreachable=0
rtr01    : ok=4 changed=1       failed=0        unreachable=0
```

ANSIBLE - CHANGING CONFIG - COMMIT (2)

```
(network-tutorials) ➜  tutorial-3-abstract-vendor-configuration git:(master) ✗ ansible-playbook playbook_configure.yml
......
# Load configuration into the device ********************************************************************************
  * rtr00                       - changed=False --  ------------------------------------------
  * rtr01                       - changed=False --  ------------------------------------------

# STATS ************************************************************************************************************
rtr00    : ok=4 changed=0       failed=0       unreachable=0
rtr01    : ok=4 changed=0       failed=0       unreachable=0
```

Changes are idempotent

SUMMARY

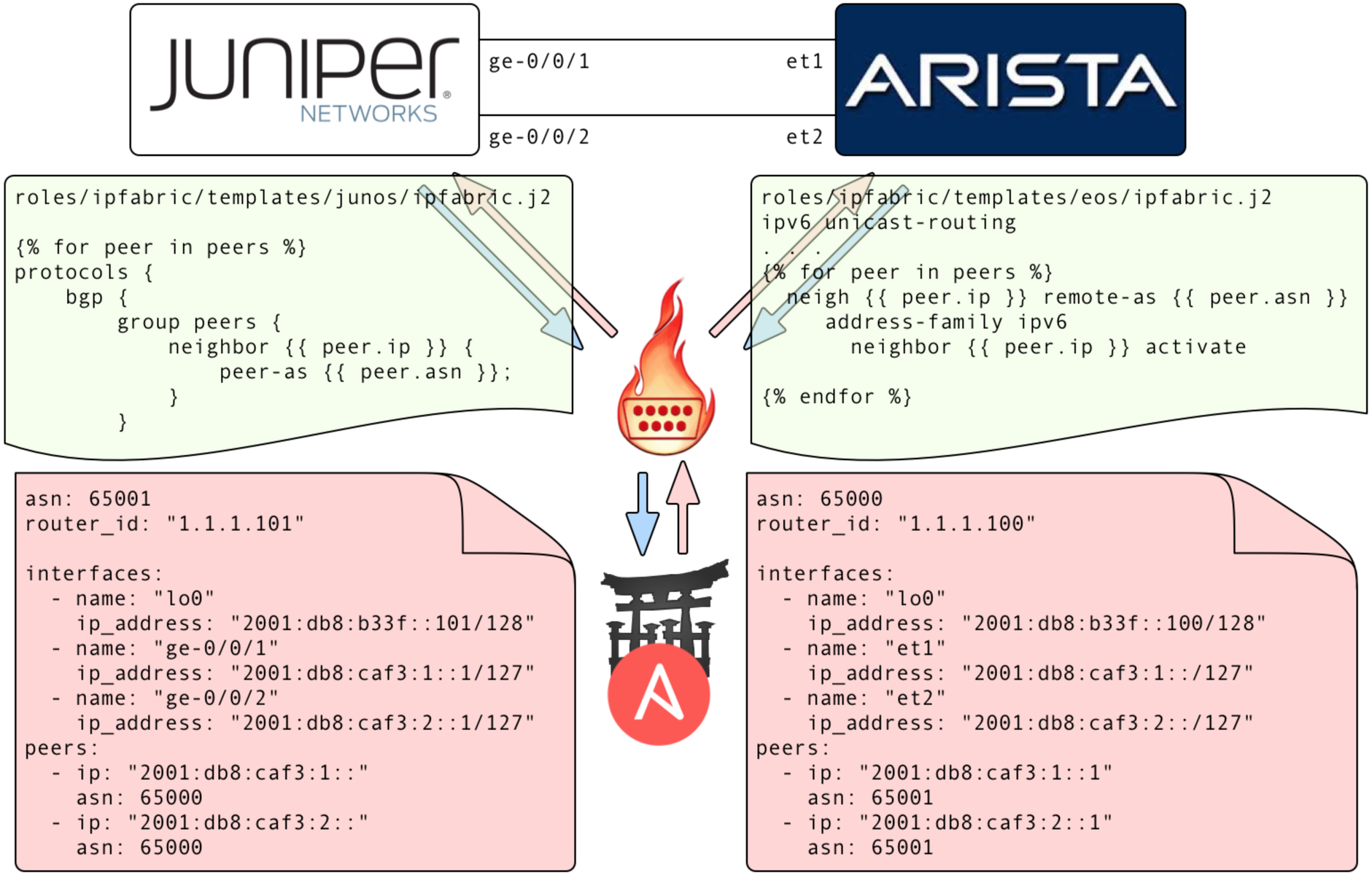
1. We used ansible as our framework to build our configuration management system
2. We stored variables in `YAML`
3. We used `jinja2` templates to *translate* variables into device configuration
4. We leveraged on `NAPALM` to seamlessly deploy the resulting configuration and to gather information

4 - DATA-DRIVEN CONFIGURATION

Link to the code on github

OBJECTIVES

We are going to leverage on what we have learnt so far to build an IP fabric using **vendor-agnostic data** to drive the configuration.


JUNIPER NETWORKS
ge-0/0/1
et1
ge-0/0/2
et2
ARISTA
roles/ipfabric/templates/junos/ipfabric.j2
{% for peer in peers %}
protocols {
bgp {
group peers {
neighbor {{ peer.ip }} {
peer-as {{ peer.asn }};
}
}
roles/ipfabric/templates/eos/ipfabric.j2
ipv6 unicast-routing
. . .
{% for peer in peers %}
neigh {{ peer.ip }} remote-as {{ peer.asn }}
address-family ipv6
neighbor {{ peer.ip }} activate
{% endfor %}
asn: 65001
router_id: "1.1.1.101"
interfaces:
- name: "lo0"
ip_address: "2001:db8:b33f::101/128"
- name: "ge-0/0/1"
ip_address: "2001:db8:caf3:1::1/127"
- name: "ge-0/0/2"
ip_address: "2001:db8:caf3:2::1/127"
peers:
- ip: "2001:db8:caf3:1::"
asn: 65000
- ip: "2001:db8:caf3:2::"
asn: 65000
asn: 65000
router_id: "1.1.1.100"
interfaces:
- name: "lo0"
ip_address: "2001:db8:b33f::100/128"
- name: "et1"
ip_address: "2001:db8:caf3:1::/127"
- name: "et2"
ip_address: "2001:db8:caf3:2::/127"
peers:
- ip: "2001:db8:caf3:1::1"
asn: 65001
- ip: "2001:db8:caf3:2::1"
asn: 65001

D-D - CHANGING CONFIG - DATA

```
➜ cat host_vars/rtr00.yml
---
hostname: rtr00

asn: 65000
router_id: "1.1.1.100"

interfaces:
  - name: "lo0"
    ip_address: "2001:db8:b33f::100/128"
  - name: "et1"
    ip_address: "2001:db8:caf3:1::/127"
  - name: "et2"
    ip_address: "2001:db8:caf3:2::/127"
peers:
  - ip: "2001:db8:caf3:1::1"
    asn: 65001
  - ip: "2001:db8:caf3:2::1"
    asn: 65001
```

```
➜ cat host_vars/rtr01.yml
---
hostname: rtr01

asn: 65001
router_id: "1.1.1.101"

interfaces:
  - name: "lo0"
    ip_address: "2001:db8:b33f::101/128"
  - name: "ge-0/0/1"
    ip_address: "2001:db8:caf3:1::1/127"
  - name: "ge-0/0/2"
    ip_address: "2001:db8:caf3:2::1/127"
peers:
  - ip: "2001:db8:caf3:1::"
    asn: 65000
  - ip: "2001:db8:caf3:2::"
    asn: 65000
```

```
➜ cat hosts
[all]
rtr00 os=eos   host=127.0.0.1 user=vagrant password=vagrant port=12443
rtr01 os=junos host=127.0.0.1 user=vagrant password="" port=12203
```

D-D CHANGING CONFIG - TEMPLATES

```
➜ cat (...)/templates/eos/ipfabric.j2
ipv6 unicast-routing

{% for interface in interfaces %}

default interface {{ interface.name }}
interface {{ interface.name }}
  {{ 'no switchport' if not interface.name.startsw
  ipv6 address {{ interface.ip_address }}

{% endfor %}

route-map EXPORT-LO0 permit 10
   match interface Loopback0

no router bgp
router bgp {{ asn }}
   router-id {{ router_id }}
   redistribute connected route-map EXPORT-LO0

{% for peer in peers %}
     neighbor {{ peer.ip }} remote-as {{ peer.asn
     address-family ipv6
       neighbor {{ peer.ip }} activate

{% endfor %}
```

```
➜ cat (...)/templates/junos/ipfabric.j2
{% for interface in interfaces %}

interfaces {
    replace:
    {{ interface.name }} {
        unit 0 {
            family inet6 {
                address {{ interface.ip_address }}
            }
        }
    }
}

{% endfor %}

routing-options {
    router-id {{ router_id }};
    autonomous-system {{ asn }};
}

policy-options {
    policy-statement EXPORT_LO0 {
        from interface lo0.0;
        then accept;
    }
    policy-statement PERMIT_ALL {
        from protocol bgp;
        then accept;
    }
```

D-D CHANGING CONFIG - PLAYBOOK

```yaml
...
- name: Basic Configuration
  hosts: all
  connection: local
  roles:
    - base

- name: Fabric Configuration
  hosts: all
  connection: local
  roles:
    - ipfabric
...
```

```
➜ tree roles
├── base
│   ├── tasks
│   │   └── main.yml
│   └── templates
│       ├── eos
│       │   └── simple.j2
│       └── junos
│           └── simple.j2
└── ipfabric
    ├── tasks
    │   └── main.yml
    └── templates
        ├── eos
        │   └── ipfabric.j2
        │
        └── junos
            └── ipfabric.j2
```

D-D CHANGING CONFIG - RUN (1)

```
(network-tutorials) ➜  tutorial-4-data-driven-configuration git:(master) ✗ ansible-playbook playbook_configure.yml -C
..................
# Load configuration into the device ***********************************************************************************
  * rtr00                       - changed=True ---------------------------------------------------
    @@ -8,7 +8,7 @@
     !
     transceiver qsfp default-mode 4x10G
     !
    -hostname thehostnama
    +hostname rtr00
     ip domain-name acme.com
     !
     spanning-tree mode mstp
    @@ -21,14 +21,37 @@
     username ******** privilege 15 role network-admin secret 5 $1$gPwz4Oc5$.FRwGrhtFFEIKHAeGN5Dq/
     !
     interface Ethernet1
    +   no switchport
    +   ipv6 address 2001:db8:caf3:1::/127
     !
     interface Ethernet2
    +   no switchport
    +   ipv6 address 2001:db8:caf3:2::/127
    +!
    +interface Loopback0
    +   ipv6 address 2001:db8:b33f::100/128
     !
```

D-D CHANGING CONFIG - RUN (2)

```
   !
  +ipv6 unicast-routing
  +!
  +route-map EXPORT-LO0 permit 10
  +   match interface Loopback0
  +!
  +router bgp 65000
  +   router-id 1.1.1.100
  +   neighbor 2001:db8:caf3:1::1 remote-as 65001
  +   neighbor 2001:db8:caf3:1::1 maximum-routes 12000
  +   neighbor 2001:db8:caf3:2::1 remote-as 65001
  +   neighbor 2001:db8:caf3:2::1 maximum-routes 12000
  +   redistribute connected route-map EXPORT-LO0
  +   address-family ipv6
  +      neighbor 2001:db8:caf3:1::1 activate
  +      neighbor 2001:db8:caf3:2::1 activate
  +!
    management api http-commands
       no shutdown
    !
* rtr01                       - changed=True ------------------------------------------------
  [edit system]
  -  host-name another-host;
  +  host-name rtr01;
  [edit interfaces]
  +   ge-0/0/1 {
  +       unit 0 {
```

D-D CHANGING CONFIG - RUN (3)

```
+            family inet6 {
+                address 2001:db8:caf3:1::1/127;
+            }
+        }
+    }
+    ge-0/0/2 {
+        unit 0 {
+            family inet6 {
+                address 2001:db8:caf3:2::1/127;
+            }
+        }
+    }
+    lo0 {
+        unit 0 {
+            family inet6 {
+                address 2001:db8:b33f::101/128;
+            }
+        }
+    }
[edit]
+  routing-options {
+      router-id 1.1.1.101;
+      autonomous-system 65001;
+  }
```

D-D CHANGING CONFIG - RUN (4)

```
+  protocols {
+      bgp {
+          import PERMIT_ALL;
+          export [ EXPORT_LO0 PERMIT_ALL ];
+          group peers {
+              neighbor 2001:db8:caf3:1:: {
+                  peer-as 65000;
+              }
+              neighbor 2001:db8:caf3:2:: {
+                  peer-as 65000;
+              }
+          }
+      }
+  }
+  policy-options {
+      policy-statement EXPORT_LO0 {
+          from interface lo0.0;
+          then accept;
+      }
+      policy-statement PERMIT_ALL {
+          from protocol bgp;
+          then accept;
+      }
+  }
```

```
# STATS ******************************************************************************************
rtr00    : ok=9 changed=1       failed=0        unreachable=0
rtr01    : ok=9 changed=1       failed=0        unreachable=0
```

D-D VERIFYING BGP STATE - PLAYBOOK

```
- name: "get facts from device"
  napalm_get_facts:
    hostname: "{{ host }}"
    username: "{{ user }}"
    dev_os: "{{ os }}"
    password: "{{ password }}"
    optional_args:
      port: "{{ port }}"
    filter: ['bgp_neighbors']
  register: napalm_facts
- name: "Check BGP sessions are healthy"
  assert:
    that:
      - item.value.is_up
    msg: "{{ item.key }} is down"
  with_dict: "{{ napalm_facts.ansible_facts.bgp_neighbors.global.peers }}"
- name: "Check BGP sessions are receiving prefixes"
  assert:
    that:
      - item.value.address_family.ipv6.received_prefixes > 0
    msg: "{{ item.key }} is not receiving any prefixes"
  with_dict: "{{ napalm_facts.ansible_facts.bgp_neighbors.global.peers }}"
```

D-D VERIFYING BGP STATE - RUN (1)

```
(network-tutorials) ➜  tutorial-4-data-driven-configuration git:(master) ✗ ansible-playbook playbook_verify.yml
..
# Check BGP sessions are healthy ************************************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:1::1          - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:1::           - changed=False -- all assertions passed --------------------------
# Check BGP sessions are receiving prefixes *************************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:1::1          - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:1::           - changed=False -- all assertions passed --------------------------

# STATS *************************************************************************************************************
rtr00    : ok=3 changed=0       failed=0        unreachable=0
rtr01    : ok=3 changed=0       failed=0        unreachable=0
```

D-D VERIFYING BGP STATE - RUN (2)

```
(network-tutorials) ➜  tutorial-4-data-driven-configuration git:(master) ✗ ansible-playbook playbook_verify.yml
..
# Check BGP sessions are healthy ************************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:1::1          - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:1::           - changed=False -- all assertions passed --------------------------
# Check BGP sessions are receiving prefixes *************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:1::1          - changed=False -- all assertions passed --------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - FAILED!!! -- One or more items failed ---------------------------
      * 2001:db8:caf3:2::           - FAILED!!! -------------------------------------------------------
        2001:db8:caf3:2:: is not receiving any prefixes
      * 2001:db8:caf3:1::           - changed=False -- all assertions passed --------------------------

# STATS *************************************************************************************************
rtr00    : ok=3 changed=0       failed=0       unreachable=0
rtr01    : ok=2 changed=0       failed=1       unreachable=0
```

Maybe a faulty policy or missing communities?

D-D VERIFYING BGP STATE - RUN (3)

```
(network-tutorials) ➜  tutorial-4-data-driven-configuration git:(master) ✗ ansible-playbook playbook_verify.yml
..
# Check BGP sessions are healthy ***********************************************************************************
  * rtr00                      - FAILED!!! -- One or more items failed --------------------------
      * 2001:db8:caf3:1::1          - FAILED!!! -- 2001:db8:caf3:1::1 is down -------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed -------------------------
  * rtr01                      - FAILED!!! -- One or more items failed --------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed -------------------------
      * 2001:db8:caf3:1::           - FAILED!!! -- 2001:db8:caf3:1:: is down -------------------------

# STATS ************************************************************************************************************
rtr00    : ok=1 changed=0       failed=1        unreachable=0
rtr01    : ok=1 changed=0       failed=1        unreachable=0
```

Most probably a faulty link

D-D VERIFYING BGP STATE - RUN (4)

```
(network-tutorials) ➜  tutorial-4-data-driven-configuration git:(master) ✗ ansible-playbook playbook_verify.yml
..
# Check BGP sessions are healthy ************************************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed --------------------------
# Check BGP sessions are receiving prefixes *************************************************************************
  * rtr00                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::1          - changed=False -- all assertions passed --------------------------
  * rtr01                      - changed=False -- All items completed ----------------------------
      * 2001:db8:caf3:2::           - changed=False -- all assertions passed --------------------------

# STATS *************************************************************************************************************
rtr00    : ok=3 changed=0       failed=0        unreachable=0
rtr01    : ok=3 changed=0       failed=0        unreachable=0
```

Looks fine but it is not, there is a link with no BGP session configured

SUMMARY

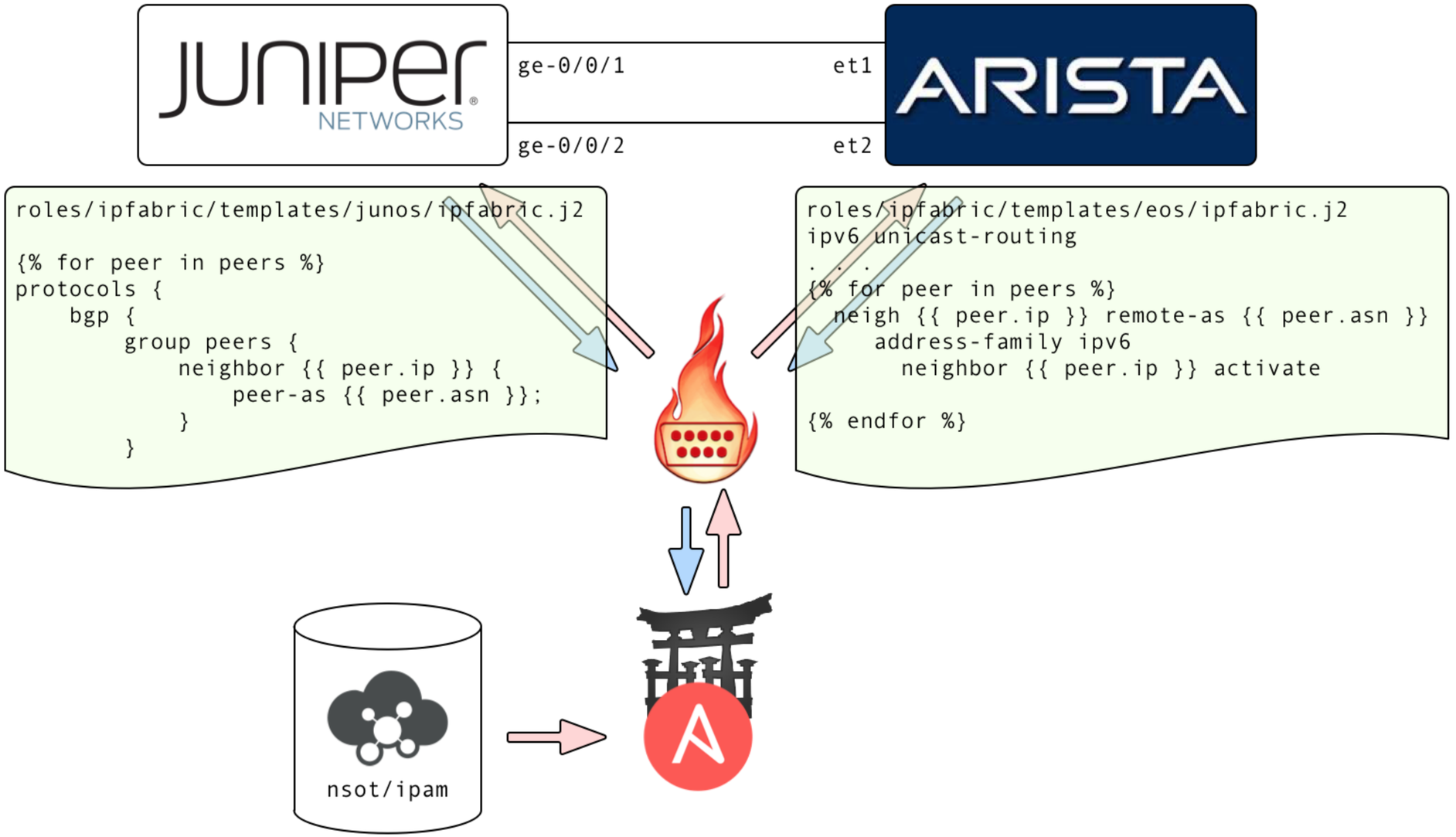
- We leveraged on what we learnt so far to build something more complex
- Storing data in static files makes them hard to manipulate but it's great for mocking new services or for data that doesn't change often or doesn't need to be queried
- We verified state of the network using the network as source of truth rather than using our knowledge about the network

5 - DATA DRIVEN CONFIGURATION WITH A BACKEND

Link to the code on github

OBJECTIVES

Similar to the previous scenario, however, this time we will use a backend to store and manipulate the data.


JUNIPER
NETWORKS
ge-0/0/1
et1
ge-0/0/2
et2
ARISTA
roles/ipfabric/templates/junos/ipfabric.j2
{% for peer in peers %}
protocols {
bgp {
group peers {
neighbor {{ peer.ip }} {
peer-as {{ peer.asn }};
}
}
roles/ipfabric/templates/eos/ipfabric.j2
ipv6 unicast-routing
. . .
{% for peer in peers %}
neigh {{ peer.ip }} remote-as {{ peer.asn }}
address-family ipv6
neighbor {{ peer.ip }} activate
{% endfor %}
nsot/ipam

DDB - BEFORE WE BEGIN - START NSOT VM

```
➜  ~ cd workspace/network-tutorials/labs/nsot
➜  nsot git:(master) ✗ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: A newer version of the box 'ubuntu/trusty64' is available! You currently
==> default: have version '20160926.0.1'. The latest is version '20161004.0.0'. Run
==> default: `vagrant box update` to update.
==> default: Setting the name of the VM: nsot_default_1475926003501_89657
==> default: Clearing any previously set forwarded ports...
==> default: Fixed port collision for 22 => 2222. Now on port 2200.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 80 (guest) => 8080 (host) (adapter 1)
    default: 8990 (guest) => 8990 (host) (adapter 1)
    default: 22 (guest) => 2200 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2200
```

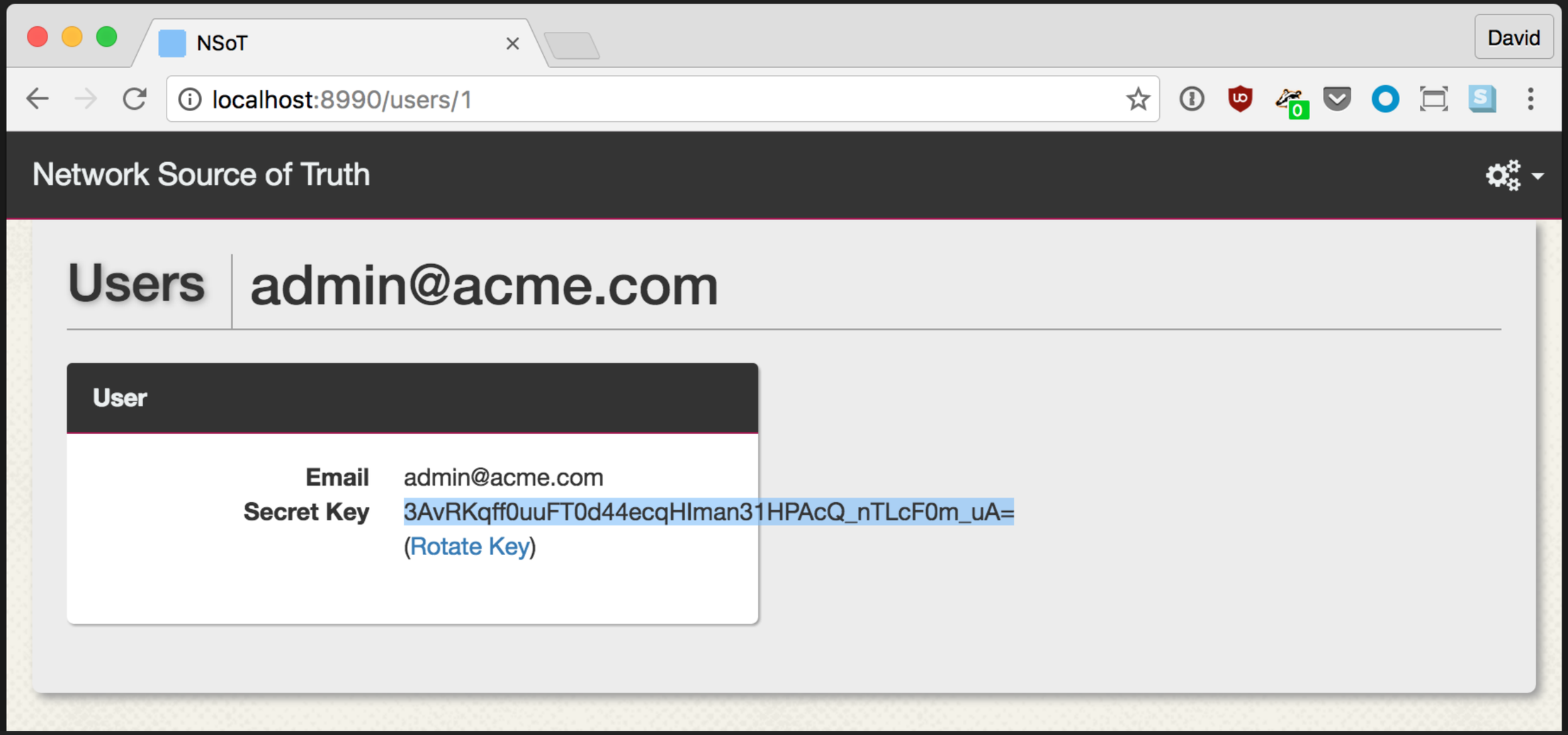
DDB - BEFORE WE BEGIN - START NSOT

```
==> default: 145 static files copied to '/usr/local/lib/python2.7/dist-packages/nsot/staticfiles'.
➜  nsot git:(master) ✗ vagrant ssh
vagrant@vagrant-ubuntu-trusty-64:~$ nsot-server start
Performing upgrade before service startup...

You have installed Django's auth system, and don't have any superusers defined.
Would you like to create one now? (yes/no): yes
Email address: admin@acme.com
Password:
Password (again):
Superuser created successfully.
Performing collectstatic before service startup...

0 static files copied to '/usr/local/lib/python2.7/dist-packages/nsot/staticfiles', 145 unmodified.
Running service: 'http', num workers: 4, worker timeout: 30
[2016-10-08 06:32:02 +0000] [26759] [INFO] Starting gunicorn 19.3.0
[2016-10-08 06:32:02 +0000] [26759] [INFO] Listening at: http://0.0.0.0:8990 (26759)
[2016-10-08 06:32:02 +0000] [26759] [INFO] Using worker: gevent
[2016-10-08 06:32:02 +0000] [26766] [INFO] Booting worker with pid: 26766
[2016-10-08 06:32:02 +0000] [26767] [INFO] Booting worker with pid: 26767
[2016-10-08 06:32:02 +0000] [26768] [INFO] Booting worker with pid: 26768
[2016-10-08 06:32:02 +0000] [26769] [INFO] Booting worker with pid: 26769
```

DDB - BEFORE WE BEGIN - GATHER NSOT TOKEN



DDB - BEFORE WE BEGIN - NSOT CLIENT CONF

```
~/.pynsotrc
[pynsot]
url = http://localhost:8990/api
secret_key = 3AvRKqff0uuFT0d44ecqHIman31HPAcQ_nTLcF0m_uA=  # Token
auth_method = auth_token
email = admin@acme.com
```

Make sure you replace the `secret_key` with the token you gathered in the previous step.

DDB - BEFORE WE BEGIN - CREATE THE DATA

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ ./create_data.sh
+ nsot sites add --name my_demo_site
[SUCCESS] Added site!
+ nsot attributes add --site-id 1 --resource-name device --name os
[SUCCESS] Added attribute!
+ nsot attributes add --site-id 1 --resource-name device --name host
[SUCCESS] Added attribute!
+ nsot attributes add --site-id 1 --resource-name device --name user
[SUCCESS] Added attribute!
+ nsot attributes add --site-id 1 --resource-name device --name password --allow-empty
[SUCCESS] Added attribute!
+ nsot attributes add --site-id 1 --resource-name device --name port
[SUCCESS] Added attribute!
+ nsot devices add --site-id 1 --hostname rtr00
[SUCCESS] Added device!
+ nsot devices update --site-id 1 --id 1 -a os=eos -a host=127.0.0.1 -a user=vagrant -a password=vagrant -a port=12443
[SUCCESS] Updated device!
+ nsot devices add --site-id 1 --hostname rtr01
[SUCCESS] Added device!
+ nsot devices update --site-id 1 --id 2 -a os=junos -a host=127.0.0.1 -a user=vagrant -a password= -a port=12203
[SUCCESS] Updated device!
+ nsot attributes add --site-id 1 --resource-name device --name asn
[SUCCESS] Added attribute!
+ nsot attributes add --site-id 1 --resource-name device --name router_id
[SUCCESS] Added attribute!
+ nsot devices update --site-id 1 --id 1 -a asn=65001 -a router_id=10.1.1.1
```

DDB - BEFORE WE BEGIN - VERIFY THE DATA (1)

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ nsot devices list -s 1
+-----------------------------------+
| ID   Hostname   Attributes        |
+-----------------------------------+
| 1    rtr00      router_id=10.1.1.1 |
|                 asn=65001         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=vagrant  |
|                 os=eos            |
|                 port=12443        |
| 2    rtr01      router_id=10.1.1.2 |
|                 asn=65002         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=         |
|                 os=junos          |
|                 port=12203        |
+-----------------------------------+
```

DDB - BEFORE WE BEGIN - VERIFY THE DATA (2)

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ nsot interfaces list -s 1
+-----------------------------------------------------------------------------------+
| ID   Device   Name       MAC    Addresses                Attributes                |
+-----------------------------------------------------------------------------------+
| 1    1        lo0        None   2001:db8:b33f::100/128   connects_to_iface=lo0       |
|                                                          connects_to_device=loopback |
|                                                          link_type=loopback          |
| 2    2        lo0        None   2001:db8:b33f::101/128   connects_to_iface=lo0       |
|                                                          connects_to_device=loopback |
|                                                          link_type=loopback          |
| 3    1        et1        None   2001:db8:caf3::/128      connects_to_iface=ge-0/0/1  |
|                                                          link_type=fabric            |
|                                                          connects_to_device=rtr01    |
| 4    1        et2        None   2001:db8:caf3::2/128     connects_to_iface=ge-0/0/2  |
|                                                          link_type=fabric            |
|                                                          connects_to_device=rtr01    |
| 5    2        ge-0/0/1   None   2001:db8:caf3::1/128     connects_to_iface=et1       |
|                                                          link_type=fabric            |
|                                                          connects_to_device=rtr00    |
| 6    2        ge-0/0/2   None   2001:db8:caf3::3/128     connects_to_iface=et2       |
|                                                          link_type=fabric            |
|                                                          connects_to_device=rtr00    |
+-----------------------------------------------------------------------------------+
```

DDB - BACKEND INTEGRATION - INVENTORY

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ ./hosts.py | jq
{
  "my_demo_site": {
    "hosts": [
      "rtr00",
      "rtr01"
    ],
    "children": [],
    "vars": {}
  },
  "_meta": {
    "hostvars": {
      "rtr01": {
        "router_id": "10.1.1.2",
        "interfaces": {
          "lo0": {
            "description": "",
            "type": 6,
            "id": 2,
            "ipv6": [
              "2001:db8:b33f::101/64"
            ],
            "name": "lo0",
```

DDB - BACKEND INTEGRATION - SMARTER DATA

```
(network-tutorials) → tutorial-5-data-driven-configuration_with_backend git:(master) ✗ nsot interfaces list -s 1 -a link_type=fabric
+-----------------------------------------------------------------------------+
| ID   Device   Name       MAC    Addresses              Attributes             |
+-----------------------------------------------------------------------------+
| 3    1        et1        None   2001:db8:caf3::/128    connects_to_iface=ge-0/0/1 |
|                                                        link_type=fabric       |
|                                                        connects_to_device=rtr01   |
| 4    1        et2        None   2001:db8:caf3::2/128   connects_to_iface=ge-0/0/2 |
|                                                        link_type=fabric       |
|                                                        connects_to_device=rtr01   |
| 5    2        ge-0/0/1   None   2001:db8:caf3::1/128   connects_to_iface=et1      |
|                                                        link_type=fabric       |
|                                                        connects_to_device=rtr00   |
| 6    2        ge-0/0/2   None   2001:db8:caf3::3/128   connects_to_iface=et2      |
|                                                        link_type=fabric       |
|                                                        connects_to_device=rtr00   |
+-----------------------------------------------------------------------------+
```

```
{% for iface, iface_data in ifaces.items() if iface_data.attributes.link_type == "fabric" %}
    {% set peer =iface_data.attributes.connects_to_device %}
    {% set peer_iface =iface_data.attributes.connects_to_iface %}
    {% set peer_ip = hostvars[peer]['interfaces'][peer_iface]['ipv6'][0].split('/')[0] %}
    {% set peer_asn = hostvars[peer]['asn'] %}
     neighbor {{ peer_ip }} remote-as {{ peer_asn }}
     neighbor {{ peer_ip }} description {{ peer }}:{{ peer_iface }}
     address-family ipv6
       neighbor {{ peer_ip }} activate
{% endfor %}
```

DDB - CONFIGURATION - RUN (1)

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ ansible-playbook playbook_configure.yml
.................
# Load configuration into the device ***************************************************************************
  * rtr01                       - changed=True ------------------------------------------------
    [edit interfaces ge-0/0/1]
    +   description rtr00:et1;
    [edit interfaces ge-0/0/1 unit 0 family inet6]
    +       address 2001:db8:caf3::1/127;
    -       address 2001:db8:caf3:1::1/127;
    [edit interfaces ge-0/0/2]
    +   description rtr00:et2;
    [edit interfaces ge-0/0/2 unit 0 family inet6]
    +       address 2001:db8:caf3::3/127;
    -       address 2001:db8:caf3:2::1/127;
    [edit interfaces lo0]
    +   description loopback:lo0;
    [edit routing-options]
    -  router-id 1.1.1.101;
    +  router-id 10.1.1.2;
    -  autonomous-system 65001;
    +  autonomous-system 65002;
    [edit protocols bgp group peers]
    +     neighbor 2001:db8:caf3::2 {
    +         description rtr00:et2;
    +         peer-as 65001;
    +     }
    +     neighbor 2001:db8:caf3:: {
    +         description rtr00:et1;
    +         peer-as 65001;
    +     }
```

DDB - CONFIGURATION - RUN (2)

```
-     neighbor 2001:db8:caf3:1:: {
-         peer-as 65000;
-     }
-     neighbor 2001:db8:caf3:2:: {
-         peer-as 65000;
-     }
* rtr00                        - changed=True ------------------------------------------------
  @@ -21,14 +21,17 @@
   username ******** privilege 15 role network-admin secret 5 $1$gPwz4Oc5$.FRwGrhtFFEIKHAeGN5Dq/
   !
   interface Ethernet1
  +   description rtr01:ge-0/0/1
      no switchport
  -   ipv6 address 2001:db8:caf3:1::/127
  +   ipv6 address 2001:db8:caf3::/127
   !
   interface Ethernet2
  +   description rtr01:ge-0/0/2
      no switchport
  -   ipv6 address 2001:db8:caf3:2::/127
  +   ipv6 address 2001:db8:caf3::2/127
   !
   interface Loopback0
  +   description loopback:lo0
      ipv6 address 2001:db8:b33f::100/128
   !
   interface Management1
  @@ -41,16 +44,18 @@
```

DDB - CONFIGURATION - RUN (3)

```
-router bgp 65000
-   router-id 1.1.1.100
-   neighbor 2001:db8:caf3:1::1 remote-as 65001
-   neighbor 2001:db8:caf3:1::1 maximum-routes 12000
-   neighbor 2001:db8:caf3:2::1 remote-as 65001
-   neighbor 2001:db8:caf3:2::1 maximum-routes 12000
+router bgp 65001
+   router-id 10.1.1.1
+   neighbor 2001:db8:caf3::1 remote-as 65002
+   neighbor 2001:db8:caf3::1 description rtr01:ge-0/0/1
+   neighbor 2001:db8:caf3::1 maximum-routes 12000
+   neighbor 2001:db8:caf3::3 remote-as 65002
+   neighbor 2001:db8:caf3::3 description rtr01:ge-0/0/2
+   neighbor 2001:db8:caf3::3 maximum-routes 12000
    redistribute connected route-map EXPORT-LO0
    address-family ipv6
-      neighbor 2001:db8:caf3:1::1 activate
-      neighbor 2001:db8:caf3:2::1 activate
+      neighbor 2001:db8:caf3::1 activate
+      neighbor 2001:db8:caf3::3 activate
 !
management api http-commands
   no shutdown

# STATS ************************************************************************************************************
rtr00    : ok=9 changed=1       failed=0        unreachable=0
rtr01    : ok=9 changed=1       failed=0        unreachable=0
```

DDB - VERIFYING - PLAYBOOK (1)

```
- name: "Get facts from device"
  napalm_get_facts:
    hostname: "{{ host }}"
    username: "{{ user }}"
    dev_os: "{{ os }}"
    password: "{{ password }}"
    optional_args:
      port: "{{ port }}"
    filter: ['bgp_neighbors']
  register: napalm_facts
```

DDB - VERIFYING - PLAYBOOK (2)

```
- name: "Check all BGP sessions are configured"
  assert:
    that:
      - hostvars[item.value.attributes.connects_to_device]['interfaces'][item.value.attributes.conne
    msg: "{{ '{}:{}'.format(item.value.attributes.connects_to_iface, item.value.attributes.connects_
  with_dict: "{{ interfaces }}"
  when: "{{ item.value.attributes.link_type == 'fabric' }}"
  tags: [print_action]
```

Let's take a look:

```
hostvars[item.value.attributes.connects_to_device]['interfaces']\
        [item.value.attributes.connects_to_iface]['ipv6'][0].split('/')[0]\
        in napalm_facts.ansible_facts.bgp_neighbors.global.peers
item.value                                                  # interface
interface.attributes.connects_to_device                     # peer
interface.attributes.connects_to_iface                      # peer_iface
napalm_facts.ansible_facts.bgp_neighbors.global.peers       # my_configured_peers

hostvars[peer]['interfaces'][peer_iface]['ipv6'] in my_configured_peers
```

DDB - VERIFYING - PLAYBOOK (3)

```
- name: "Check BGP sessions are healthy"
  assert:
    that:
      - napalm_facts.ansible_facts.bgp_neighbors.global.peers[hostvars[item.value.attributes.connect
    msg: "{{ '{}:{}'.format(item.value.attributes.connects_to_iface, item.value.attributes.connects_
  with_dict: "{{ interfaces }}"
  when: "{{ item.value.attributes.link_type == 'fabric' }}"
  tags: [print_action]
- name: "Chack BGP sessions are receiving prefixes"
  assert:
    that:
      - napalm_facts.ansible_facts.bgp_neighbors.global.peers[hostvars[item.value.attributes.connect
    msg: "{{ '{}:{}'.format(item.value.attributes.connects_to_iface, item.value.attributes.connects_
  with_dict: "{{ interfaces }}"
  when: "{{ item.value.attributes.link_type == 'fabric' }}"
  tags: [print_action]
```

DDB - VERIFYING - RUN (1)

```
# Check all BGP sessions are configured **********************************************************************
  * rtr00                        - changed=False -- All items completed ----------------------------
      * et2                          - changed=False -- all assertions passed --------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * et1                          - changed=False -- all assertions passed --------------------------
  * rtr01                        - changed=False -- All items completed ----------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * ge-0/0/2                     - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                     - changed=False -- all assertions passed --------------------------
# Check BGP sessions are healthy *****************************************************************************
  * rtr00                        - changed=False -- All items completed ----------------------------
      * et2                          - changed=False -- all assertions passed --------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * et1                          - changed=False -- all assertions passed --------------------------
  * rtr01                        - changed=False -- All items completed ----------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * ge-0/0/2                     - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                     - changed=False -- all assertions passed --------------------------
# Chack BGP sessions are receiving prefixes ******************************************************************
  * rtr00                        - changed=False -- All items completed ----------------------------
      * et2                          - changed=False -- all assertions passed --------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * et1                          - changed=False -- all assertions passed --------------------------
  * rtr01                        - changed=False -- All items completed ----------------------------
      * lo0                          - changed=False --  ----------------------------------------------
      * ge-0/0/2                     - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                     - changed=False -- all assertions passed --------------------------
```

We verify links, not BGP sessions

DDB - VERIFYING - RUN (2)

```
(network-tutorials) ➜  tutorial-5-data-driven-configuration_with_backend git:(master) ✗ ansible-playbook playbook_verify.yml
..
# Check all BGP sessions are configured ********************************************************************************
  * rtr00                       - FAILED!!! -- One or more items failed --------------------------
      * et2                         - changed=False -- all assertions passed -------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * et1                         - FAILED!!! ------------------------------------------------------
        ge-0/0/1:rtr01 (2001:db8:caf3::1) is missing
  * rtr01                       - changed=False -- All items completed ---------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * ge-0/0/2                    - changed=False -- all assertions passed -------------------------
      * ge-0/0/1                    - changed=False -- all assertions passed -------------------------
# Check BGP sessions are healthy ***************************************************************************************
  * rtr01                       - FAILED!!! -- One or more items failed --------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * ge-0/0/2                    - changed=False -- all assertions passed -------------------------
      * ge-0/0/1                    - FAILED!!! ------------------------------------------------------
        et1:rtr00 (2001:db8:caf3::) is configured but down

# STATS ****************************************************************************************************************
rtr00    : ok=1 changed=0       failed=1        unreachable=0
rtr01    : ok=2 changed=0       failed=1        unreachable=0
```

We can correlate BGP sessions with interfaces easily. That might allows us to infer the reason behind the errors. For example, in the picture above it's easy to identify that the reason why `2001:db8:caf3::` is down in `rtr01` it's because `2001:db8:caf3::1` is not configured in `rtr00`.

SUMMARY

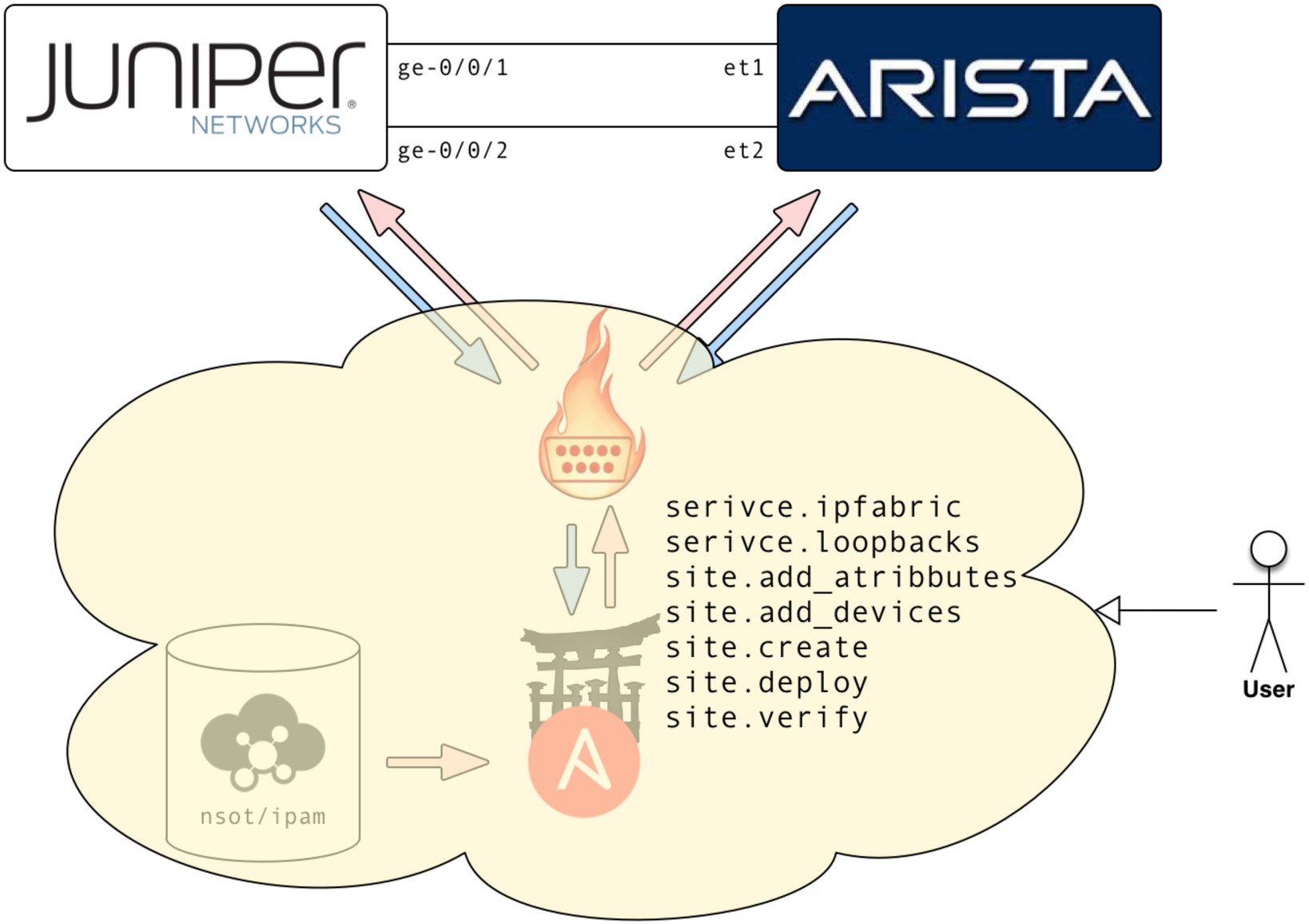
- We replaced all those `YAML` files with a backend (`nsot`)
- We added richer data to `nsot` that we previously had
- We leveraged on the richer data to build the configurations with less information (BGP peers were inferred)
- We leveraged on the richer data to help diagnose problems

6 - ABSTRACTIONS

Link to the code on github

OBJECTIVES

We are going to simplify operations. Sites and services will be abstracted and defined in simple definition files and added to the backend using an easy to use tool. Finally, ansible will be hidden behind the same tool.


JUNIPER
NETWORKS
ge-0/0/1
et1
ge-0/0/2
et2
ARISTA
serivce.ipfabric
serivce.loopbacks
site.add_atribbutes
site.add_devices
site.create
site.deploy
site.verify
nsot/ipam
User

ABSTRATIONS - SITE DEFINITION - ACME

data/acme/devices.yml

```yaml
---
devices:
  rtr00:
    os: eos
    host: 127.0.0.1
    domain: acme.com
    user: vagrant
    password: vagrant
    port: '12443'
    asn: '65001'
    router_id: 10.1.1.1
  rtr01:
    os: junos
    host: 127.0.0.1
    domain: acme.com
    user: vagrant
    password: ''
    port: '12203'
    asn: '65002'
    router_id: 10.1.1.2
```

data/acme/attributes.yml

```yaml
---
attributes:
  Device:
    os: {}
    host: {}
    user: {}
    password:
      constraints:
        allow_empty: true
    port: {}
    asn: {}
    router_id: {}
    domain: {}
  Network:
    type: {}
    service: {}
  Interface:
    link_type: {}
    connects_to_device: {
    connects_to_iface: {}
```

data/acme/services.yml

```yaml
---
loopbacks:
  network_ranges:
    loopbacks: 2001:db8:feed::/48
  definition: {}

ipfabric:
  network_ranges:
    fabric_links: 2001:db8:cafe::/48
  definition:
    links:
      - left_device: rtr00
        left_iface: et1
        right_device: rtr01
        right_iface: ge-0/0/1
      - left_device: rtr00
        left_iface: et2
        right_device: rtr01
        right_iface: ge-0/0/2
```

ABSTRATIONS - SITE DEFINITION - EVIL

data/acme/devices.yml

```yaml
---
devices:
  evil00:
    os: eos
    host: 127.0.0.1
    domain: evilcorp.com
    user: vagrant
    password: vagrant
    port: '12443'
    asn: '65666'
    router_id: 10.6.66.1
  evil01:
    os: junos
    host: 127.0.0.1
    domain: evil.com
    user: vagrant
    password: ''
    port: '12203'
    asn: '65666'
    router_id: 10.6.66.2
```

data/acme/attributes.yml

```yaml
---
attributes:
  Device:
    os: {}
    host: {}
    user: {}
    password:
      constraints:
        allow_empty: true
    port: {}
    asn: {}
    router_id: {}
    domain: {}
  Network:
    type: {}
    service: {}
  Interface:
    link_type: {}
    connects_to_device: {
    connects_to_iface: {}
```

data/acme/services.yml

```yaml
---
loopbacks:
  network_ranges:
    loopbacks: 2001:db8:dead::/48
  definition: {}

ipfabric:
  network_ranges:
    fabric_links: 2001:db8:c0ff::/48
  definition:
    links:
      - left_device: rtr00
        left_iface: et1
        right_device: rtr01
        right_iface: ge-0/0/1
      - left_device: rtr00
        left_iface: et2
        right_device: rtr01
        right_iface: ge-0/0/2
```

ABSTRACTIONS - COMMAND-LINE

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv --list
Available tasks:

  db.reset               Resets the database. All sites and their related data are wiped out.
  serivce.ipfabric       Read service definition for ipfabric service and add data to the backend.
  serivce.loopbacks      Read service definition for loopbacks service and add data to the backend.
  site.add_atribbutes    Add attributes to a site.
  site.add_devices       Add devices to a site.
  site.create            Create a site.
  site.deploy            Deploy a site.
  site.verify            Verify site is healthy.
```

ABSTRACTIONS - CREATING THE SITES - ACME(1)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.create -n acme -d "Acme Corp."
2016-10-08 20:07:53,712 - tutorial-6-abstractions - INFO - {u'description': u'Acme Corp.', u'id': 2, u'name': u'acme'}
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.add_atribbutes -s acme -f data/acme/attributes.yml
2016-10-08 20:07:57,852 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'router_id'}
2016-10-08 20:07:57,877 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'domain'}
2016-10-08 20:07:57,904 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'asn'}
2016-10-08 20:07:57,927 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'host'}
2016-10-08 20:07:57,947 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'user'}
2016-10-08 20:07:57,970 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'password', 'constraints'
: {'allow_empty': True}}
2016-10-08 20:07:57,991 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'os'}
2016-10-08 20:07:58,013 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Device', 'name': 'port'}
2016-10-08 20:07:58,036 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Interface', 'name': 'connects_to_iface'}
2016-10-08 20:07:58,057 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Interface', 'name': 'link_type'}
2016-10-08 20:07:58,077 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Interface', 'name': 'connects_to_device'}
2016-10-08 20:07:58,098 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Network', 'name': 'type'}
2016-10-08 20:07:58,120 - tutorial-6-abstractions - INFO - Creating attribute: {'resource_name': 'Network', 'name': 'service'}
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.add_devices -s acme -f data/acme/devices.yml
2016-10-08 20:08:10,347 - tutorial-6-abstractions - INFO - Creating device: {'attributes': {'router_id': '10.1.1.2', 'domain': 'acme.com', '
port': '12203', 'host': '127.0.0.1', 'user': 'vagrant', 'password': '', 'os': 'junos', 'asn': '65002'}, 'hostname': 'rtr01'}
2016-10-08 20:08:10,400 - tutorial-6-abstractions - INFO - Creating device: {'attributes': {'router_id': '10.1.1.1', 'domain': 'acme.com', '
port': '12443', 'host': '127.0.0.1', 'user': 'vagrant', 'password': 'vagrant', 'os': 'eos', 'asn': '65001'}, 'hostname': 'rtr00'}
```

ABSTRACTIONS - CREATING THE SITES - ACME(2)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv serivce.loopbacks -s acme -f data/acme/services.yml
2016-10-08 20:08:14,980 - tutorial-6-abstractions - INFO - Creating network: {'attributes': {'type': 'loopbacks', 'service': 'loopback'}, 'c
idr': '2001:db8:feed::/48'}
2016-10-08 20:08:15,033 - tutorial-6-abstractions - INFO - Creating iface: {'device': 4, 'attributes': {'connects_to_iface': 'loopback', 'co
nnects_to_device': 'loopback', 'link_type': 'loopbacks'}, 'name': 'lo0', 'addresses': [u'2001:db8:feed::1/128']}
2016-10-08 20:08:15,086 - tutorial-6-abstractions - INFO - {u'description': u'', u'type': 6, u'id': 7, u'name': u'lo0', u'parent_id': None,
u'mac_address': u'00:00:00:00:00:00', u'device': 4, u'attributes': {u'connects_to_iface': u'loopback', u'link_type': u'loopbacks', u'connect
s_to_device': u'loopback'}, u'speed': 1000, u'networks': [u'2001:db8:feed::/48'], u'addresses': [u'2001:db8:feed::1/128']}
2016-10-08 20:08:15,105 - tutorial-6-abstractions - INFO - Creating iface: {'device': 3, 'attributes': {'connects_to_iface': 'loopback', 'co
nnects_to_device': 'loopback', 'link_type': 'loopbacks'}, 'name': 'lo0', 'addresses': [u'2001:db8:feed::2/128']}
2016-10-08 20:08:15,153 - tutorial-6-abstractions - INFO - {u'description': u'', u'type': 6, u'id': 8, u'name': u'lo0', u'parent_id': None,
u'mac_address': u'00:00:00:00:00:00', u'device': 3, u'attributes': {u'connects_to_iface': u'loopback', u'link_type': u'loopbacks', u'connect
s_to_device': u'loopback'}, u'speed': 1000, u'networks': [u'2001:db8:feed::/48'], u'addresses': [u'2001:db8:feed::2/128']}
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv serivce.ipfabric -s acme -f data/acme/services.yml
2016-10-08 20:08:18,782 - tutorial-6-abstractions - INFO - Creating network: {'attributes': {'type': 'fabric_links', 'service': 'ipfabric'},
 'cidr': '2001:db8:cafe::/48'}
2016-10-08 20:08:18,837 - tutorial-6-abstractions - INFO - Creating iface: {'device': 4, 'attributes': {'connects_to_iface': 'ge-0/0/1', 'co
nnects_to_device': 'rtr01', 'link_type': 'fabric_links'}, 'name': 'et1', 'addresses': []}
2016-10-08 20:08:18,869 - tutorial-6-abstractions - INFO - {u'description': u'', u'type': 6, u'id': 9, u'name': u'et1', u'parent_id': None,
u'mac_address': u'00:00:00:00:00:00', u'device': 4, u'attributes': {u'connects_to_iface': u'ge-0/0/1', u'link_type': u'fabric_links', u'conn
ects_to_device': u'rtr01'}, u'speed': 1000, u'networks': [], u'addresses': []}
2016-10-08 20:08:18,877 - tutorial-6-abstractions - INFO - Creating iface: {'device': 3, 'attributes': {'connects_to_iface': 'et1', 'connect
s_to_device': 'rtr00', 'link_type': 'fabric_links'}, 'name': 'ge-0/0/1', 'addresses': []}
2016-10-08 20:08:18,913 - tutorial-6-abstractions - INFO - {u'description': u'', u'type': 6, u'id': 10, u'name': u'ge-0/0/1', u'parent_id':
None, u'mac_address': u'00:00:00:00:00:00', u'device': 3, u'attributes': {u'connects_to_iface': u'et1', u'link_type': u'fabric_links', u'con
```

ABSTRACTIONS - CREATING THE SITES - EVIL

```
inv site.create --name evil
inv site.add_atribbutes -s evil -f data/evil/attributes.yml
inv site.add_devices -s evil -f data/evil/devices.yml
inv serivce.loopbacks -s evil -f data/evil/services.yml
inv serivce.ipfabric -s evil -f data/evil/services.yml
```

ABSTRACTIONS - VERIFY DATA - ACME

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot sites list -n acme
+-------------------------+
| ID   Name   Description |
+-------------------------+
| 2    acme   Acme Corp.  |
+-------------------------+
```

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot devices list -s 2
+-----------------------------------+
| ID   Hostname   Attributes        |
+-----------------------------------+
| 4    rtr00      router_id=10.1.1.1 |
|                 domain=acme.com   |
|                 asn=65001         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=vagrant  |
|                 os=eos            |
|                 port=12443        |
| 3    rtr01      router_id=10.1.1.2 |
|                 domain=acme.com   |
|                 asn=65002         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=         |
|                 os=junos          |
|                 port=12203        |
+-----------------------------------+
```

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot interfaces list -s 2
+-----------------------------------------------------------------------------------------------------------------+
| ID   Device   Name       MAC                Addresses             Attributes                                   |
+-----------------------------------------------------------------------------------------------------------------+
| 7    4        lo0        00:00:00:00:00:00  2001:db8:feed::1/128   connects_to_iface=loopback                  |
|                                                                    link_type=loopbacks                          |
|                                                                    connects_to_device=loopback                  |
| 8    3        lo0        00:00:00:00:00:00  2001:db8:feed::2/128   connects_to_iface=loopback                  |
|                                                                    link_type=loopbacks                          |
|                                                                    connects_to_device=loopback                  |
| 9    4        et1        00:00:00:00:00:00  2001:db8:cafe::/128    connects_to_iface=ge-0/0/1                   |
|                                                                    link_type=fabric_links                       |
|                                                                    connects_to_device=rtr01                     |
| 10   3        ge-0/0/1   00:00:00:00:00:00  2001:db8:cafe::1/128   connects_to_iface=et1                        |
|                                                                    link_type=fabric_links                       |
|                                                                    connects_to_device=rtr00                     |
| 11   4        et2        00:00:00:00:00:00  2001:db8:cafe::2/128   connects_to_iface=ge-0/0/2                   |
|                                                                    link_type=fabric_links                       |
|                                                                    connects_to_device=rtr01                     |
| 12   3        ge-0/0/2   00:00:00:00:00:00  2001:db8:cafe::3/128   connects_to_iface=et2                        |
|                                                                    link_type=fabric_links                       |
|                                                                    connects_to_device=rtr00                     |
+-----------------------------------------------------------------------------------------------------------------+
```

For demonstrational purposes ;)

ABSTRACTIONS - VERIFY DATA - EVIL

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot sites list -n evil
+-------------------------+
| ID   Name   Description |
+-------------------------+
| 3    evil               |
+-------------------------+
```

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot devices list -s 3
+-----------------------------------+
| ID   Hostname   Attributes        |
+-----------------------------------+
| 6    evil00     router_id=10.6.66.1 |
|                 domain=evilcorp.com |
|                 asn=65666         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=vagrant  |
|                 os=eos            |
|                 port=12443        |
| 5    evil01     router_id=10.6.66.2 |
|                 domain=evil.com   |
|                 asn=65666         |
|                 host=127.0.0.1    |
|                 user=vagrant      |
|                 password=         |
|                 os=junos          |
|                 port=12203        |
+-----------------------------------+
```

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ nsot interfaces list -s 3
+-----------------------------------------------------------------------------------------------------------------+
| ID   Device   Name      MAC                Addresses             Attributes                  |
+-----------------------------------------------------------------------------------------------------------------+
| 13   6        et1       00:00:00:00:00:00  2001:db8:c0ff::/128    connects_to_iface=ge-0/0/1  |
|                                                                    link_type=fabric_links      |
|                                                                    connects_to_device=evil01   |
| 14   5        ge-0/0/1  00:00:00:00:00:00  2001:db8:c0ff::1/128  connects_to_iface=et1       |
|                                                                    link_type=fabric_links      |
|                                                                    connects_to_device=evil00   |
| 15   6        et2       00:00:00:00:00:00  2001:db8:c0ff::2/128  connects_to_iface=ge-0/0/2  |
|                                                                    link_type=fabric_links      |
|                                                                    connects_to_device=evil01   |
| 16   5        ge-0/0/2  00:00:00:00:00:00  2001:db8:c0ff::3/128  connects_to_iface=et2       |
|                                                                    link_type=fabric_links      |
|                                                                    connects_to_device=evil00   |
| 17   6        lo0       00:00:00:00:00:00  2001:db8:dead::1/128  connects_to_iface=loopback  |
|                                                                    link_type=loopbacks         |
|                                                                    connects_to_device=loopback |
| 18   5        lo0       00:00:00:00:00:00  2001:db8:dead::2/128  connects_to_iface=loopback  |
|                                                                    link_type=loopbacks         |
|                                                                    connects_to_device=loopback |
+-----------------------------------------------------------------------------------------------------------------+
```

For demonstrational purposes ;)

ABSTRACTIONS - DEPLOY SITE - ACME (1)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.deploy --site acme
.................
# Load configuration into the device ********************************************************************************
  * rtr00                        - changed=True ----------------------------------------------
    @@ -23,16 +23,16 @@
     interface Ethernet1
        description rtr01:ge-0/0/1
        no switchport
    -   ipv6 address 2001:db8:caf3::/127
    +   ipv6 address 2001:db8:cafe::/127
     !
     interface Ethernet2
        description rtr01:ge-0/0/2
        no switchport
    -   ipv6 address 2001:db8:caf3::2/127
    +   ipv6 address 2001:db8:cafe::2/127
     !
     interface Loopback0
    -   description loopback:lo0
    -   ipv6 address 2001:db8:b33f::100/128
    +   description loopback:loopback
    +   ipv6 address 2001:db8:feed::1/48
     !
     interface Management1
        ip address 10.0.2.15/24
    @@ -46,12 +46,16 @@
     !
```

ABSTRACTIONS - DEPLOY SITE - ACME (2)

```
   router bgp 65001
      router-id 10.1.1.1
   -  neighbor 2001:db8:caf3::3 remote-as 65002
   -  neighbor 2001:db8:caf3::3 description rtr01:ge-0/0/2
   -  neighbor 2001:db8:caf3::3 maximum-routes 12000
   +  neighbor 2001:db8:cafe::1 remote-as 65002
   +  neighbor 2001:db8:cafe::1 description rtr01:ge-0/0/1
   +  neighbor 2001:db8:cafe::1 maximum-routes 12000
   +  neighbor 2001:db8:cafe::3 remote-as 65002
   +  neighbor 2001:db8:cafe::3 description rtr01:ge-0/0/2
   +  neighbor 2001:db8:cafe::3 maximum-routes 12000
      redistribute connected route-map EXPORT-LO0
      address-family ipv6
   -     neighbor 2001:db8:caf3::3 activate
   +     neighbor 2001:db8:cafe::1 activate
   +     neighbor 2001:db8:cafe::3 activate
    !
   management api http-commands
      no shutdown
* rtr01                      - changed=True ------------------------------------------------
  [edit interfaces ge-0/0/1 unit 0 family inet6]
  +       address 2001:db8:cafe::1/127;
  -       address 2001:db8:caf3::1/127;
  [edit interfaces ge-0/0/2 unit 0 family inet6]
  +       address 2001:db8:cafe::3/127;
  -       address 2001:db8:caf3::3/127;
```

ABSTRACTIONS - DEPLOY SITE - ACME (3)

```
[edit interfaces lo0]
-   description loopback:lo0;
+   description loopback:loopback;
[edit interfaces lo0 unit 0 family inet6]
+       address 2001:db8:feed::2/48;
-       address 2001:db8:b33f::101/128;
[edit protocols bgp group peers]
+     neighbor 2001:db8:cafe::2 {
+         description rtr00:et2;
+         peer-as 65001;
+     }
+     neighbor 2001:db8:cafe:: {
+         description rtr00:et1;
+         peer-as 65001;
+     }
-     neighbor 2001:db8:caf3::2 {
-         description rtr00:et2;
-         peer-as 65001;
-     }
-     neighbor 2001:db8:caf3:: {
-         description rtr00:et1;
-         peer-as 65001;
-     }

# STATS ************************************************************************************************
rtr00    : ok=9 changed=1       failed=0        unreachable=0
rtr01    : ok=9 changed=1       failed=0        unreachable=0
```

ABSTRACTIONS - DEPLOY SITE - ACME (4)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.deploy --site acme --commit
.................
# Load configuration into the device ********************************************************************************
  * rtr00                        - changed=True ----------------------------------------------
    @@ -23,16 +23,16 @@
     interface Ethernet1
        description rtr01:ge-0/0/1
        no switchport
    -   ipv6 address 2001:db8:caf3::/127
    +   ipv6 address 2001:db8:cafe::/127
     !
     interface Ethernet2
        description rtr01:ge-0/0/2
        no switchport
    -   ipv6 address 2001:db8:caf3::2/127
    +   ipv6 address 2001:db8:cafe::2/127
     !
     interface Loopback0
    -   description loopback:lo0
    -   ipv6 address 2001:db8:b33f::100/128
    +   description loopback:loopback
    +   ipv6 address 2001:db8:feed::1/48
     !
     interface Management1
        ip address 10.0.2.15/24
    @@ -46,12 +46,16 @@
     !
```

ABSTRACTIONS - DEPLOY SITE - ACME (5)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.deploy --site acme
.................
# Load configuration into the device ***************************************************************************
  * rtr00                       - changed=False --  ------------------------------------------
  * rtr01                       - changed=False --  ------------------------------------------

# STATS ********************************************************************************************************
rtr00    : ok=9 changed=0       failed=0        unreachable=0
rtr01    : ok=9 changed=0       failed=0        unreachable=0
```

ABSTRACTIONS - VERIFY SITE - ACME

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.verify --site acme
..
# Check all BGP sessions are configured ********************************************************************************
  * rtr00                       - changed=False -- All items completed ----------------------------
      * et2                         - changed=False -- all assertions passed --------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * et1                         - changed=False -- all assertions passed --------------------------
  * rtr01                       - changed=False -- All items completed ----------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * ge-0/0/2                    - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                    - changed=False -- all assertions passed --------------------------
# Check BGP sessions are healthy ***************************************************************************************
  * rtr00                       - changed=False -- All items completed ----------------------------
      * et2                         - changed=False -- all assertions passed --------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * et1                         - changed=False -- all assertions passed --------------------------
  * rtr01                       - changed=False -- All items completed ----------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * ge-0/0/2                    - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                    - changed=False -- all assertions passed --------------------------
# Chack BGP sessions are receiving prefixes ****************************************************************************
  * rtr00                       - changed=False -- All items completed ----------------------------
      * et2                         - changed=False -- all assertions passed --------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * et1                         - changed=False -- all assertions passed --------------------------
  * rtr01                       - changed=False -- All items completed ----------------------------
      * lo0                         - changed=False --  ----------------------------------------------
      * ge-0/0/2                    - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                    - changed=False -- all assertions passed --------------------------

# STATS ****************************************************************************************************************
rtr00    : ok=4 changed=0       failed=0       unreachable=0
rtr01    : ok=4 changed=0       failed=0       unreachable=0
```

ABSTRACTIONS - DEPLOY SITE - EVIL (1)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.deploy --site evil
.................
# Load configuration into the device ********************************************************************************
  * evil00                      - changed=True ------------------------------------------------
    @@ -8,8 +8,8 @@
     !
     transceiver qsfp default-mode 4x10G
     !
    -hostname rtr00
    -ip domain-name acme.com
    +hostname evil00
    +ip domain-name evilcorp.com
     !
     spanning-tree mode mstp
     !
    @@ -21,18 +21,18 @@
     username ******** privilege 15 role network-admin secret 5 $1$gPwz4Oc5$.FRwGrhtFFEIKHAeGN5Dq/
     !
     interface Ethernet1
    -   description rtr01:ge-0/0/1
    +   description evil01:ge-0/0/1
        no switchport
    -   ipv6 address 2001:db8:cafe::/127
    +   ipv6 address 2001:db8:c0ff::/127
     !
     interface Ethernet2
    -   description rtr01:ge-0/0/2
    +   description evil01:ge-0/0/2
```

ABSTRACTIONS - DEPLOY SITE - EVIL (2)

```
  -router bgp 65001
  -   router-id 10.1.1.1
  -   neighbor 2001:db8:cafe::1 remote-as 65002
  -   neighbor 2001:db8:cafe::1 description rtr01:ge-0/0/1
  -   neighbor 2001:db8:cafe::1 maximum-routes 12000
  -   neighbor 2001:db8:cafe::3 remote-as 65002
  -   neighbor 2001:db8:cafe::3 description rtr01:ge-0/0/2
  -   neighbor 2001:db8:cafe::3 maximum-routes 12000
  +router bgp 65666
  +   router-id 10.6.66.1
  +   neighbor 2001:db8:c0ff::1 remote-as 65666
  +   neighbor 2001:db8:c0ff::1 description evil01:ge-0/0/1
  +   neighbor 2001:db8:c0ff::1 maximum-routes 12000
  +   neighbor 2001:db8:c0ff::3 remote-as 65666
  +   neighbor 2001:db8:c0ff::3 description evil01:ge-0/0/2
  +   neighbor 2001:db8:c0ff::3 maximum-routes 12000
      redistribute connected route-map EXPORT-LO0
      address-family ipv6
  -      neighbor 2001:db8:cafe::1 activate
  -      neighbor 2001:db8:cafe::3 activate
  +      neighbor 2001:db8:c0ff::1 activate
  +      neighbor 2001:db8:c0ff::3 activate
   !
   management api http-commands
      no shutdown
* evil01                    - changed=True ------------------------------------------------
  [edit system]
  -  host-name rtr01;
  +  host-name evil01;
```

ABSTRACTIONS - DEPLOY SITE - EVIL (3)

```
* evil01                    - changed=True ------------------------------------------------
  [edit system]
  -  host-name rtr01;
  +  host-name evil01;
  -  domain-name acme.com;
  +  domain-name evil.com;
  [edit interfaces ge-0/0/1]
  -   description rtr00:et1;
  +   description evil00:et1;
  [edit interfaces ge-0/0/1 unit 0 family inet6]
  +       address 2001:db8:c0ff::1/127;
  -       address 2001:db8:cafe::1/127;
  [edit interfaces ge-0/0/2]
  -   description rtr00:et2;
  +   description evil00:et2;
  [edit interfaces ge-0/0/2 unit 0 family inet6]
  +       address 2001:db8:c0ff::3/127;
  -       address 2001:db8:cafe::3/127;
  [edit interfaces lo0 unit 0 family inet6]
  +       address 2001:db8:dead::2/48;
  -       address 2001:db8:feed::2/48;
  [edit routing-options]
  -  router-id 10.1.1.2;
  +  router-id 10.6.66.2;
  -  autonomous-system 65002;
  +  autonomous-system 65666;
```

ABSTRACTIONS - DEPLOY SITE - EVIL (4)

```
[edit protocols bgp group peers]
+     neighbor 2001:db8:c0ff::2 {
+         description evil00:et2;
+         peer-as 65666;
+     }
+     neighbor 2001:db8:c0ff:: {
+         description evil00:et1;
+         peer-as 65666;
+     }
-     neighbor 2001:db8:cafe::2 {
-         description rtr00:et2;
-         peer-as 65001;
-     }
-     neighbor 2001:db8:cafe:: {
-         description rtr00:et1;
-         peer-as 65001;
-     }

# STATS ******************************************************************************************
evil00     : ok=9        changed=1       failed=0        unreachable=0
evil01     : ok=9        changed=1       failed=0        unreachable=0
```

ABSTRACTIONS - DEPLOY SITE - EVIL (5)

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.deploy --site evil --commit
.................
# Load configuration into the device ********************************************************************************
  * evil00                      - changed=True ------------------------------------------------
    @@ -8,8 +8,8 @@
     !
     transceiver qsfp default-mode 4x10G
     !
    -hostname rtr00
    -ip domain-name acme.com
    +hostname evil00
    +ip domain-name evilcorp.com
     !
     spanning-tree mode mstp
     !
    @@ -21,18 +21,18 @@
     username ******** privilege 15 role network-admin secret 5 $1$gPwz4Oc5$.FRwGrhtFFEIKHAeGN5Dq/
     !
     interface Ethernet1
    -   description rtr01:ge-0/0/1
    +   description evil01:ge-0/0/1
        no switchport
    -   ipv6 address 2001:db8:cafe::/127
    +   ipv6 address 2001:db8:c0ff::/127
     !
     interface Ethernet2
    -   description rtr01:ge-0/0/2
    +   description evil01:ge-0/0/2
```

ABSTRACTIONS - VERIFY SITE - EVIL

```
(network-tutorials) ➜  tutorial-6-abstractions git:(master) ✗ inv site.verify --site evil
..
# Check all BGP sessions are configured ********************************************************************************
  * evil00                     - changed=False -- All items completed ----------------------------
      * et2                        - changed=False -- all assertions passed --------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * et1                        - changed=False -- all assertions passed --------------------------
  * evil01                     - changed=False -- All items completed ----------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * ge-0/0/2                   - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                   - changed=False -- all assertions passed --------------------------
# Check BGP sessions are healthy ***************************************************************************************
  * evil00                     - changed=False -- All items completed ----------------------------
      * et2                        - changed=False -- all assertions passed --------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * et1                        - changed=False -- all assertions passed --------------------------
  * evil01                     - changed=False -- All items completed ----------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * ge-0/0/2                   - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                   - changed=False -- all assertions passed --------------------------
# Chack BGP sessions are receiving prefixes ****************************************************************************
  * evil00                     - changed=False -- All items completed ----------------------------
      * et2                        - changed=False -- all assertions passed --------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * et1                        - changed=False -- all assertions passed --------------------------
  * evil01                     - changed=False -- All items completed ----------------------------
      * lo0                        - changed=False --  ---------------------------------------------
      * ge-0/0/2                   - changed=False -- all assertions passed --------------------------
      * ge-0/0/1                   - changed=False -- all assertions passed --------------------------

# STATS ****************************************************************************************************************
evil00     : ok=4        changed=0       failed=0        unreachable=0
evil01     : ok=4        changed=0       failed=0        unreachable=0
```

SUMMARY

- We built a tool that focused on services
- The tool would read simple defintion files that users can easily create/manipulate and do the necessary changes in the backend.
- As the tool is very simple and easy to use, it can be used by any type of user, not only by network engineers
- Tool could be a web form rather than a cli tool

QUESTIONS?

dbarrosop@dravetech.com

Slides - https://www.dravetech.com/presos/network_automation_tutorial.html

Code - https://github.com/dravetech/network-tutorials