#ANSIBLEFEST2019

Deep Dive into Ansible Network Resource Module

Trishna Guha
Senior Software Engineer
tguha@redhat.com

Red Hat | ANSIBLE

WHAT WE ARE COVERING TODAY

1. ADDRESSING PROBLEMS
2. GOALS
3. FACTS ENHANCEMENTS
4. RESOURCE MODULE
5. RESOURCE MODULE BUILDER
6. DEMONSTRATION

ADDRESSING PROBLEMS

Enterprises are ***NOT*** automating devices from a single vendor, but from many.

Facts gathering not native for network

Lack of normalized behavior / state

Lack of normalized configuration data

Roles are too heavyweight

GOALS

NORMALIZE *ARGUMENTS*

PAST VS. CURRENT

```
# past

<Argspec>
  interface:
  min_links:
  max_links:
  rate:
  port_priority:
  graceful_convergence:
  vpc_conergence:
```

```
# current

<Argspec>
  - name:
    rate:
    port_priority:
    links:
      min:
      max:
    convergence:
      graceful:
      vpc:
```

NORMALIZE *RETURN PAYLOADS*

SAMPLE RETURN PAYLOAD

```
ok: [nxos101] =>
  result:
    after:
      contact: IT Support
      location: Room E, Building 6, Seattle, WA 98134
      users:
      - algorithm: md5
        group: network-admin
        localized_key: true
        password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        username: admin
    before:
      contact: IT Support
      location: Room E, Building 5, Seattle HQ
      users:
      - algorithm: md5
        group: network-admin
        localized_key: true
        password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        username: admin
    changed: true
    commands:
    - snmp-server location Room E, Building 6, Seattle, WA 98134
    failed: false
```

SUPPORT OPERATING ON *COLLECTIONS*

Subresources into a SINGLE Resource Module

PAST VS. CURRENT

```
# past

nxos_bgp:
  asn:
  vrf:
  rate:
  cluster_id:
  isolate:
  .
  .
  .

nxos_bgp_neighbor:
  asn:
  vrf:
  description:
  bfd:
  neighbor:
  .
  .
```

```
# current

<nxos_bgp>
  - bgp_as:
    cluster_id:
    islote:
    neighbors:
      - neighbor:
        description:
        remote_as:
```

CONSISTENT set of operations

FACTS ENHANCEMENTS

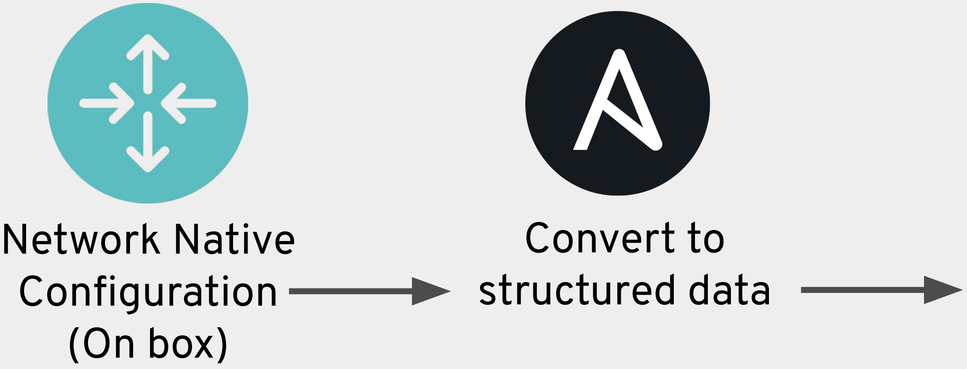
```
gather_facts: True
```

Share Common Data Structure

FACTS GATHERING

```
- hosts: arista
  module_defaults:
    eos_facts:
      gather_subset: min
      gather_network_resources:
      - interfaces
  gather_facts: True
```

Configuration Facts



Network Native Configuration (On box)

Convert to structured data

Example facts parameter
gather_network_resources:
interfaces

```
ansible_facts:
  ansible_net_api: cliconf
  ansible_net_fqdn: rtr2
  ansible_net_gather_network_resources:
  - interfaces
  ansible_net_gather_subset:
  - default
  ansible_net_hostname: rtr2
  ansible_net_image: flash:EOS.swi
  ansible_net_model: vEOS
  ansible_net_python_version: 2.7.5
  ansible_net_serialnum: D00E130991A37B49F970714D8CCF7FCB
  ansible_net_system: eos
  ansible_net_version: 4.22.0F
  ansible_network_resources:
    interfaces:
    - enabled: true
      name: Ethernet1
      mtu: 1476
    - enabled: true
      name: Loopback0
<<rest of output removed for slide brevity>>
```

Facts should be usable by modules *without* transformation

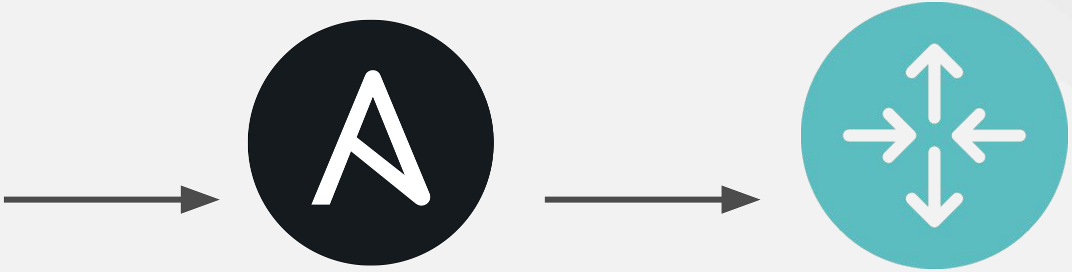
(RE-)USABLE FACTS

```
- name: example of facts being pushed right back to device
  hosts: arista
  gather_facts: yes
  module_defaults:
    eos_facts:
      gather_subset: min
      gather_network_resources: l3_interfaces


# "{{ ansible_facts[‘network_resources’]['l3_interfaces'] }}"
```

RESOURCE MODULES

RESOURCE MODULES

```
interfaces:
- enabled: true
  name: Ethernet1
  mtu: 1476
- enabled: true
  name: Loopback0
- enabled: true
  name: Loopback1
- enabled: true
  mtu: 1476
  name: Tunnel0
- enabled: true
  name: Ethernet1
- enabled: true
  name: Tunnel1
```



Structure Data

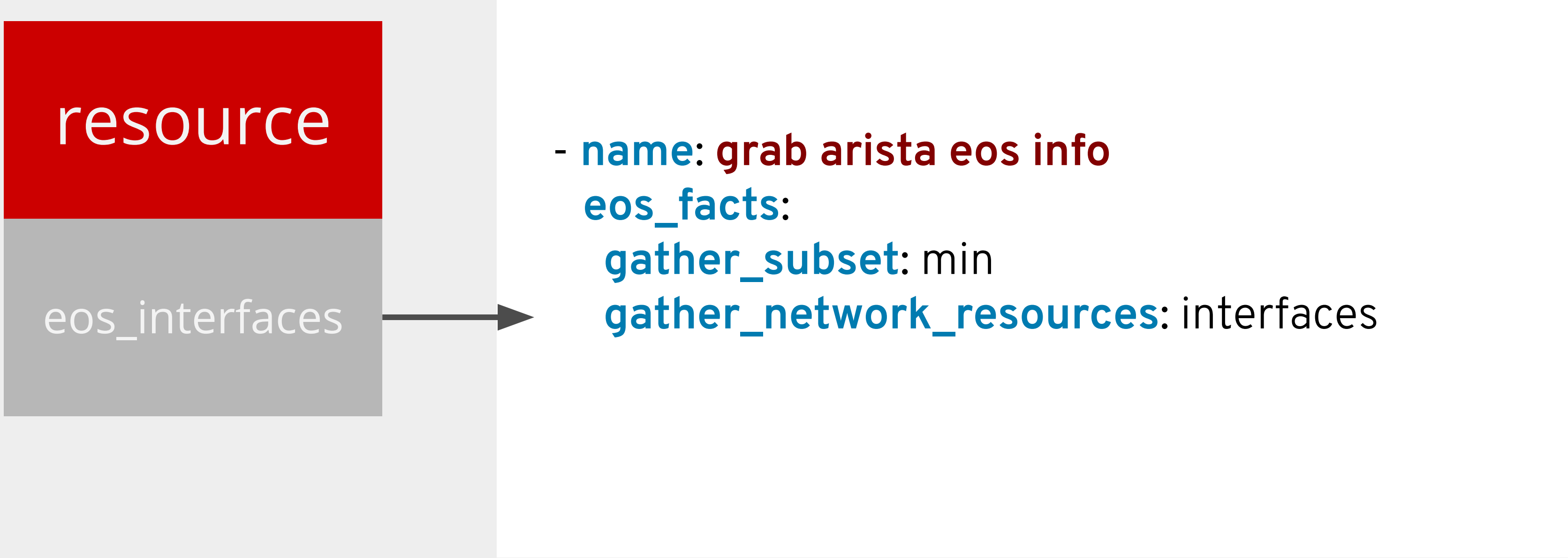
Resource Module

Network Native Configuration (On box)

Resource modules have corresponding facts

resource module	gather_network_resources
*os_interfaces (e.g. ios_interfaces for Cisco IOS-XE)	interfaces
*os_l2_interfaces	l2_interfaces
*os_l3_interfaces	l3_interfaces
*os_lacp	lacp
*os_vlans	vlans

Resource modules have corresponding facts



```
- name: grab arista eos info
  eos_facts:
    gather_subset: min
    gather_network_resources: interfaces
```

CHANGES

SINGULAR → PLURAL

Module has a parent “config” and “state” key

PAST VS. CURRENT

```
# past
nxos_ospf_interface:
  area:
  cost:
  dead_interval:
  hello_interval:
  interface:
  message_digest:
  message_digest_algorithm_type:
  message_digest_encryption_type:
  message_digest_key_id:
  message_digest_password:
  ospf:
  passive_interface:
  state:
```

```
# current
nxos_ospf_interfaces:
  config:
    - name:
      ospf:
      - area:
        cost:
        intervals:
          hello:
          dead:
        message_digest:
          enabled:
          algorithm:
          encryption:
          key:
          password:
        passive:
        process:
  state: replaced
```

RESOURCE MODULE: state

- **merged:** The configuration will be merged with the provided configuration
- **replaced:** The configuration of the provided resources will be replaced with the provided configuration
- **deleted:** The configuration of the provided resources will be deleted/defaulted
- **overridden:** The configuration of the provided resources will be replaced with the provided configuration, extraneous resource instances will be removed

RESOURCE MODULE: return

- **before:** Resource modules leverage the facts subsystem to collect the native network configuration and convert it to structured data. The configuration prior to module execution is always returned.

- **after:** If commands have been issued on the device, use the facts subsystem to collect the configuration post module execution

- **commands:** Resource modules compare the user provided data to the `before` and generate a delta command set for the device

SAMPLE RETURN PAYLOAD

```
ok: [nxos101] =>
  result:
    after:
      contact: IT Support
      location: Room E, Building 6, Seattle, WA 98134
      users:
      - algorithm: md5
        group: network-admin
        localized_key: true
        password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        privacy_password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        username: admin
    before:
      contact: IT Support
      location: Room E, Building 5, Seattle HQ
      users:
      - algorithm: md5
        group: network-admin
        localized_key: true
        password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        privacy_password: '0x73fd9a2cc8c53ed3dd4ed8f4ff157e69'
        username: admin
    changed: true
    commands:
    - snmp-server location Room E, Building 6, Seattle, WA 98134
    failed: false
```

WHAT RESOURCE MODULE CAN DO

```
- name: example of facts being pushed right back to device
  hosts: arista
  gather_facts: yes
  module_defaults:
    eos_facts:
      gather_subset: min
      gather_network_resources: l3_interfaces

  tasks:
  - name: ensure that the IP address information is accurate
    eos_l3_interfaces:
      config: "{{ ansible_network_resources['l3_interfaces'] }}"
      register: result

  - name: ensure config did not change
    assert:
      that: not result.changed
```

RESOURCE MODULE: documentation

DOCUMENTATION

- The documentation for each resource module will include the following
    - An example of each state (merged, replaced, deleted, overridden)
    - Each example will include:
        - The effective text configuration prior to module invocation
        - The effective text configuration after module completion
        - The full return payload, (before, command, after)
    - Changes to both single and sets of resources
- Documentation should answer the question “If I want to ______, how do I use this module?”

NETWORK OPERATING SYSTEM SUPPORT

EOS

IOS

IOS-XR

NXOS

VYOS

JUNOS

ANSIBLE 2.9

Arista EOS

- eos_interfaces
- eos_l2_interfaces
- eos_l3_interfaces
- eos_lacp
- eos_lldp_global
- eos_lldp_interfaces
- eos_lacp_interfaces
- eos_lag_interfaces
- eos_vlans

Cisco IOS XR

- iosxr_interfaces
- iosxr_l2_interfaces
- iosxr_l3_interfaces
- iosxr_lacp
- iosxr_lldp_global
- iosxr_lldp_interfaces
- iosxr_lacp_interfaces
- iosxr_lag_interfaces
- iosxr_vlans

Juniper JunOS

- junos_interfaces
- junos_l2_interfaces
- junos_l3_interfaces
- junos_lacp
- junos_lldp_global
- juos_lldp_interfaces
- junos_lacp_interfaces
- junos_lag_interfaces
- junos_vlans

Cisco IOS

- ios_interfaces
- ios_l2_interfaces
- ios_l3_interfaces
- ios_lacp
- ios_lldp_global
- ios_lldp_interfaces
- ios_lacp_interfaces
- ios_lag_interfaces
- ios_vlans

Cisco NX-OS

- nxos_interfaces
- nxos_l2_interfaces
- nxos_l3_interfaces
- nxos_lacp
- nxos_lldp_global
- nxos_lacp_interfaces
- nxos_lag_interfaces
- nxos_vlans
- nxos_telemetery
- nxos_bfd_interfaces

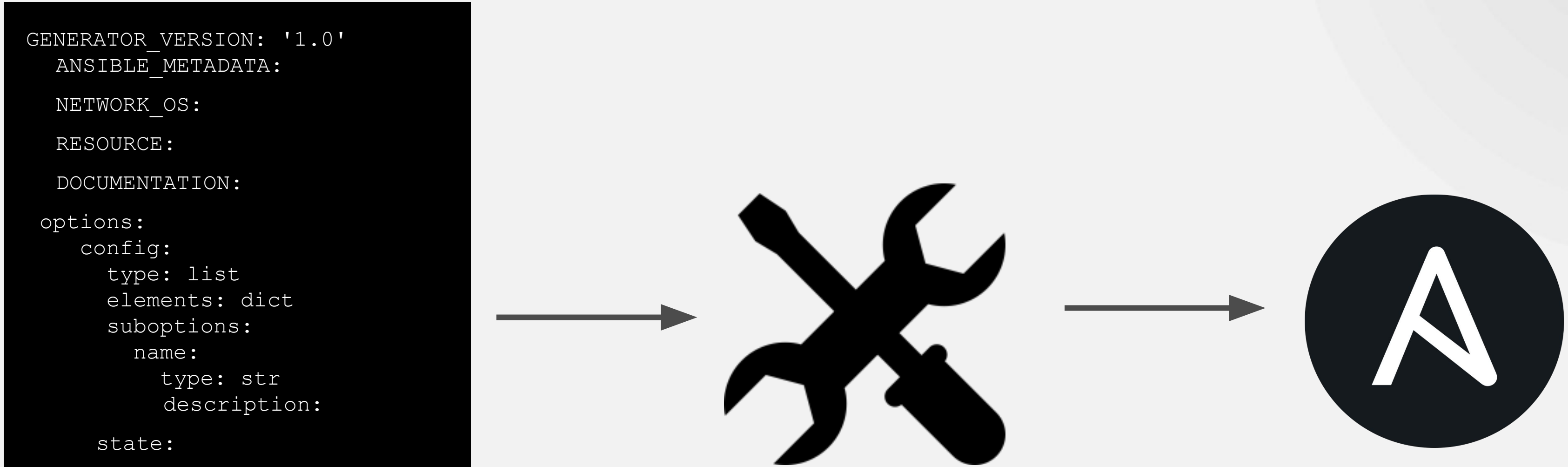
VyOS

- vyos_interfaces
- vyos_l3_interfaces
- vyos_lldp_global
- vyos_lldp_interfaces
- vyos_lag_interfaces

EXOS

- exos_lldp_global

RESOURCE MODULE BUILDER



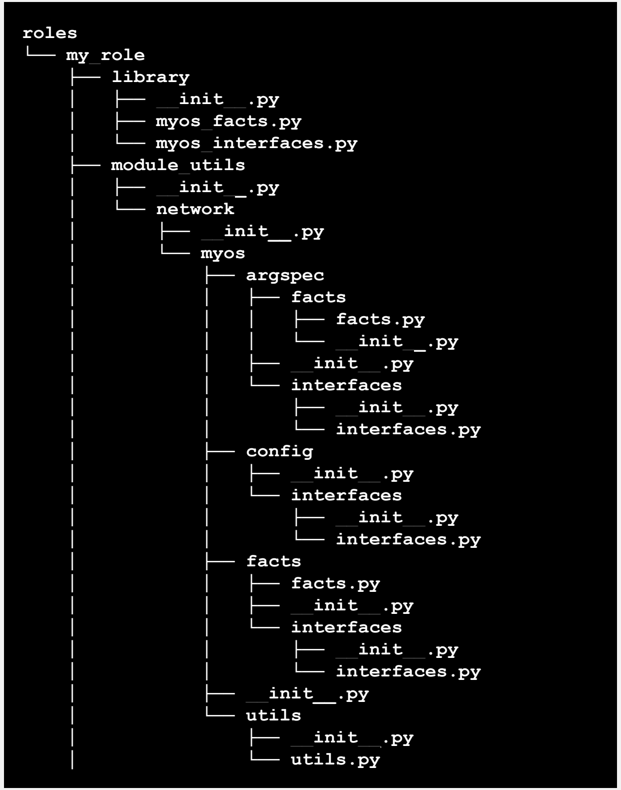
```
GENERATOR_VERSION: '1.0'
  ANSIBLE_METADATA:

  NETWORK_OS:

  RESOURCE:

  DOCUMENTATION:

 options:
    config:
      type: list
      elements: dict
      suboptions:
        name:
          type: str
          description:

     state:

        description:

        type: str

        choices:

        - merged
```

MODEL

Resource
Module Builder

Resource Module

SCAFFOLDING



```
roles
└── my_role
    ├── library
    │   ├── __init__.py
    │   ├── myos_facts.py
    │   └── myos_interfaces.py
    ├── module_utils
    │   ├── __init__.py
    │   └── network
    │       ├── __init__.py
    │       └── myos
    │           ├── argspec
    │           │   ├── facts
    │           │   │   ├── facts.py
    │           │   │   └── __init__.py
    │           │   ├── __init__.py
    │           │   └── interfaces
    │           │       ├── __init__.py
    │           │       └── interfaces.py
    │           ├── config
    │           │   ├── __init__.py
    │           │   └── interfaces
    │           │       ├── __init__.py
    │           │       └── interfaces.py
    │           ├── facts
    │           │   ├── facts.py
    │           │   ├── __init__.py
    │           │   └── interfaces
    │           │       ├── __init__.py
    │           │       └── interfaces.py
    │           ├── __init__.py
    │           └── utils
    │               ├── __init__.py
    │               └── utils.py
    │
```

RMB Prerequisites

Prerequisites include:

- Ansible
- A resource module builder model

RESOURCE MODULE MODEL

The resource module builder model is a yaml file containing the information necessary to:

- Build stub resource and facts module
- Template the argspec and docstring
- Appropriately name files, name directories, and format includes

HOW DO I DEVELOP RESOURCE MODULE

https://docs.ansible.com/ansible/devel/network/dev_guide/developing_resource_modules_network.html

DEMO

Questions?

#ANSIBLEFEST2019

THANK YOU

youtube.com/AnsibleAutomation

facebook.com/ansibleautomation

linkedin.com/company/Red-Hat

twitter.com/ansible