
# Ansible Network Automation

Live Training Session
December 2021

# Day3 Schedule - Network Configuration Changes

Feature Specific Modules
Using platform_config Modules
Using cli_config
Write Mem and Handlers
Config Hierarchy and platform_config
Deploying Jinja2 Generated Configurations
SSH Key-based Authentication (optional)
Hostvars
Module Path
Collections
Include vs Import - Dynamic vs Static

# Feature Specific Modules

```yaml
tasks:
  - name: Configure 'switchport'
    cisco.nxos.nxos_interface:
      name: "{{ item }}"
      mode: layer2
    loop:
      - Ethernet1/3
      - Ethernet1/4

  - name: Ensure mode is trunk
    cisco.nxos.nxos_l2_interface:
      name: Ethernet1/4
      mode: trunk
      native_vlan: "4"
    tags: trunk
```

```yaml
- name: Configure the login banner
  cisco.ios.ios_banner:
    banner: login
    text: |
      ************************************
      Cisco IOS banner configured via Ansible

      Skynet take-over commencing...
      ************************************
    state: present
  tags: banner_add

- name: Configure the login banner
  cisco.ios.ios_banner:
    banner: login
    text: "{{ lookup('file', './ios_banner.txt') }}"
    state: present
  tags: banner_add_file
```

Reference Material in:

{{ github_repo }}/feature_modules

# (More) General Configuration - *_config

```
---
- name: Example1
  hosts: arista5
  gather_facts: False
  tasks:
    - name: Global config
      arista.eos.eos_config:
        lines:
          - ip domain-name bogus.com
          - ntp server 130.126.24.24
          - ntp server 152.2.21.1
```

Reference Material in:

{{ github_repo }}/platform_config

Remember: platform_config is a simple string parser
Remember: the module does not know whether your configuration is valid.

Reference Material in:
{{ github_repo }}/cli_config

# cli_config

```
- name: EOS Example (cli_config)
  hosts: arista
  gather_facts: False
  tasks:
    - ansible.netcommon.cli_config:
        config: |
          ip domain-name {{ domain_name }}
          ntp server {{ ntp_server1 }}
          ntp server {{ ntp_server2 }}
          ip name-server vrf default {{ dns_server1 }}
          ip name-server vrf default {{ dns_server2 }}
```

# cli_config

```
- name: Multiple Platforms Example (cli_config)
  hosts: arista:nxos:juniper
  gather_facts: False
  vars:
    ansible_connection: network_cli
  tasks:
    - ansible.netcommon.cli_config:
        config: "{{ lookup('template', 'templates/{{ ansible_network_os }}/base_config.j2') }}"
```

Exercises:
./day3/platform_config/ex1.txt
./day3/platform_config/ex2.txt

Reference Material in:
{{ github_repo }}/handlers_wrmem

# Write Mem and Handlers

```
- name: NXOS Example
  hosts: nxos
  tasks:
    - name: Configure NEXUS VLANs
      cisco.nxos.nxos_vlan:
        vlan_id: "{{ item.vlan_id }}"
        admin_state: "{{ item.admin_state }}"
        name: "{{ item.name }}"
      loop: "{{ vlans }}"
      notify: write mem

  handlers:
    - name: write mem
      cisco.nxos.nxos_command:
        commands: copy run start
      changed_when: True
```

# Config Hierarchy and platform_config

Reference Material in:
{{ github_repo }}/config_hierarchy

```
---
- name: EOS Example (hierarchy)
  hosts: arista6
  gather_facts: False
  tasks:
    - name: Config with hierarchy
      arista.eos.eos_config:
        before: no interface Loopback99
        parents: interface Loopback99
        lines:
          - description * Test via Ansible *
          - ip address 172.16.31.1/32
          - ipv6 address 2001:db8:0:1::2/128
        match: line
        replace: block
```

# Config Hierarchy and platform_config

```
- name: IOS Example (hierarchy)
  hosts: cisco1
  gather_facts: False
  tasks:
    - name: Config with hierarchy
      cisco.ios.ios_config:
        before: no ip access-list extended TEST99
        parents: ip access-list extended TEST99
        lines:
          - permit ip host 1.1.1.1 any
          - permit ip host 2.2.2.2 any
          - permit ip host 3.3.3.3 any
          - permit ip host 4.4.4.4 any
          - permit ip host 5.5.5.5 any
        match: exact
        replace: block
```

Match:
line
strict
exact

# Config Hierarchy and platform_config

Match “line” - Order does NOT matter. Subset/superset does NOT matter.

Match “strict” - Order matters. Subset/superset does NOT matter.

Match “exact” - Order matters. Cannot be subset/superset (must be all of the elements).

If your “before” drops the parent object, then ALWAYS do “replace: block”.

Exercises:
./day3/config_hierarchy/ex1.txt
./day3/config_hierarchy/ex2.txt

# Deploying Jinja2 Generated Configurations

```yaml
vars:
  ike_policy:
    - ["10", "aes"]
    - ["20", "aes 192"]
    - ["30", "aes 256"]

tasks:
  - name: Generate IKE configuration
    ansible.builtin.template:
      src: ike_template.j2
      dest: cisco_cfg_ike.txt
    tags: two_stage

  - name: Push templated config
    cisco.ios.ios_config:
      src: cisco_cfg_ike.txt
      save_when: changed
    tags: two_stage

  - name: Generate and deploy
    cisco.ios.ios_config:
      src: ike_template.j2
      save_when: changed
    tags: one_stage
```

```
{% for policy_id, encr in ike_policy %}
crypto isakmp policy {{ policy_id }}
 encr {{ encr }}
 authentication pre-share
 group 5
!
{% endfor %}
```

Reference Material in:

{{ github_repo }}/jinja2_config

Exercises:
./day3/jinja2_deploy/ex1.txt

Reference Material in:
{{ github_repo }}/ssh_keys

# SSH Key-based Authentication (optional)

```
$ ansible-playbook ios_config1.yml -i ./ansible-hosts.ini --private-key ~/.ssh/student_key
```

```
[all:vars]
ansible_connection=network_cli
ansible_python_interpreter="~/VENV/ansible/bin/python"
ansible_user=student1

[cisco]
cisco1 ansible_host=cisco1.lasthop.io
cisco2 ansible_host=cisco2.lasthop.io

[cisco:vars]
ansible_network_os=ios
ansible_ssh_private_key_file="~/.ssh/student_key"
```

# hostvars (and NOT host_vars)


Ansible is very host-oriented. Variables are bound to hosts.

By default, Ansible will make all variables for all hosts available.

```
- name: hostvars example
  hosts: arista8
  gather_facts: False
  tasks:
    - ansible.builtin.debug:
        var: hostvars
        # var: groups
        # var: inventory_hostname
```

Reference Material in:
{{ github_repo }}/hostvars

# Module Path - Where does Ansible look for modules?


- ./library folder relative to the location of your playbook
- ./library folder in a role
- Specify --module-path <directory>
- Update “library” argument in .ansible.cfg file

# Collections - Ansible's new way of distributing modules.


A large set of collections are installed with "pip install ansible".
These collections will be automatically available to your playbook.

```
$ ls site-packages/ansible_collections/
amazon              check_point    cyberark    google     junipernetworks  ngine_io     sensu          wti
ansible             chocolatey     dellemc     hetzner    kubernetes       openstack    servicenow
ansible_release.py  cisco          f5networks  hpe        mellanox         openvswitch  splunk
arista              cloudscale_ch  fortinet    ibm        netapp           ovirt        theforeman
awx                 community      frr         infinidat  netapp_eseries   purestorage  t_systems_mms
azure               containers     gluster     inspur     netbox           __pycache__  vyos
```

You can also install collections from Ansible Galaxy (https://galaxy.ansible.com/)

# Collections - Installing collections


```
$ ansible-galaxy collection install napalm.napalm
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/download/napalm-napalm-0.9.13.tar.gz to /home/ktbyers/.ansible/tmp/ans
ible-local-24311vunr51uv/tmpopsd_9fw/napalm-napalm-0.9.13-7h__u1yh
Installing 'napalm.napalm:0.9.13' to '/home/ktbyers/.ansible/collections/ansible_collections/napalm/napalm'
napalm.napalm:0.9.13 was installed successfully
$ ls
clay584  napalm
$ pwd
/home/ktbyers/.ansible/collections/ansible_collections
```

# Installing collections


```
$ tree -C ./napalm/
./napalm/
└── napalm
    ├── build
    ├── docs
    ├── FILES.json
    ├── LICENSE
    ├── MANIFEST.json
    ├── meta
    │   └── runtime.yml
    ├── plugins
    │   ├── action
    │   │   ├── __init__.py
    │   │   ├── napalm_get_facts.py -> napalm.py
    │   │   ├── napalm_install_config.py -> napalm.py
    │   │   ├── napalm_parse_yang.py -> napalm.py
    │   │   ├── napalm_ping.py -> napalm.py
    │   │   ├── napalm.py
    │   │   └── napalm_validate.py -> napalm.py
    │   └── modules
    │       ├── __init__.py
    │       ├── napalm_cli.py
    │       ├── napalm_diff_yang.py
    │       ├── napalm_get_facts.py
    │       ├── napalm_install_config.py
    │       ├── napalm_parse_yang.py
    │       ├── napalm_ping.py
    │       ├── napalm_translate_yang.py
    │       └── napalm_validate.py
    ├── README.md
    └── tests

8 directories, 21 files
```

Reference Material in:
{{ github_repo }}/include_import_tasks

# Dynamic vs Static (and other mysteries of the universe)

- import_* == Static

  Static is generally the simpler solution.

  Tags = Propagate downwards

  Conditionals (when) = Propagate downwards.

  BUT does NOT support LOOPS

  BUT does NOT support variables from inventory sources nor from dynamic facts.

# Dynamic vs Static (and other mysteries of the universe)

- include_* == Dynamic

  Tags = Apply to the include_task task itself. Do NOT propagate downward (would need to add the tags into the included tasks)

  Conditionals (when) = Apply to the include_task task itself. Do NOT propagate downward.

  BUT loops are supported!

  BUT inventory and dynamic variables are supported!

# Include vs Import (and other mysteries of the universe)

```
- hosts: local
  tasks:
    - name: "Import (static): tags DO propogate downward"
      ansible.builtin.import_tasks: common_tasks2.yml
      tags: test_tag

    - name: "Include (dynamic): tags do NOT propogate downward"
      ansible.builtin.include_tasks: common_tasks2.yml
      tags: test_tag
```

# Include vs Import (and other mysteries of the universe)

```yaml
---
- hosts: local
  tasks:
    - name: "Import (static) - loops fail"
      ansible.builtin.import_tasks: common_tasks.yml
      # ansible.builtin.import_tasks: common_tasks_w_loop.yml
      loop:
        - 10.1.1.1
        - 10.1.1.2
        - 10.1.1.3
        - 10.1.1.4

    - name: "Include (dynamic) - loops work"
      ansible.builtin.include_tasks: common_tasks.yml
      loop:
        - 10.1.1.1
        - 10.1.1.2
        - 10.1.1.3
        - 10.1.1.4
```

Exercises:
./day3/import_vs_include/ex1.txt
./day3/import_vs_include/ex2.txt
./day3/import_vs_include/ex3.txt
./day3/import_vs_include/ex4.txt

# Include vs Import (and other mysteries of the universe)

```
---
- name: Global configuration using includes
  hosts: cisco:arista:nxos
  gather_facts: True
  tasks:
    - name: Include IOS and IOS-XE tasks
      ansible.builtin.include_tasks: "example_tasks/{{ ansible_facts.net_iostype }}/tasks.yml"
      when: ansible_facts.net_iostype is defined

    - name: Include EOS and NX-OS tasks
      ansible.builtin.include_tasks: "example_tasks/{{ ansible_network_os }}/tasks.yml"
      when: 'ansible_network_os in ["eos", "nxos"]'

- name: Global configuration using includes
  hosts: juniper
  gather_facts: False
  tasks:
    - name: Include Junos tasks
      ansible.builtin.include_tasks: "example_tasks/{{ ansible_network_os }}/tasks.yml"
```