Ansible Network Automation

Live Training Session
December 2021

$ whoami

Kirk Byers
Network Engineer
CCIE #6243 (emeritus)

Programmer
Netmiko
NAPALM
Nornir

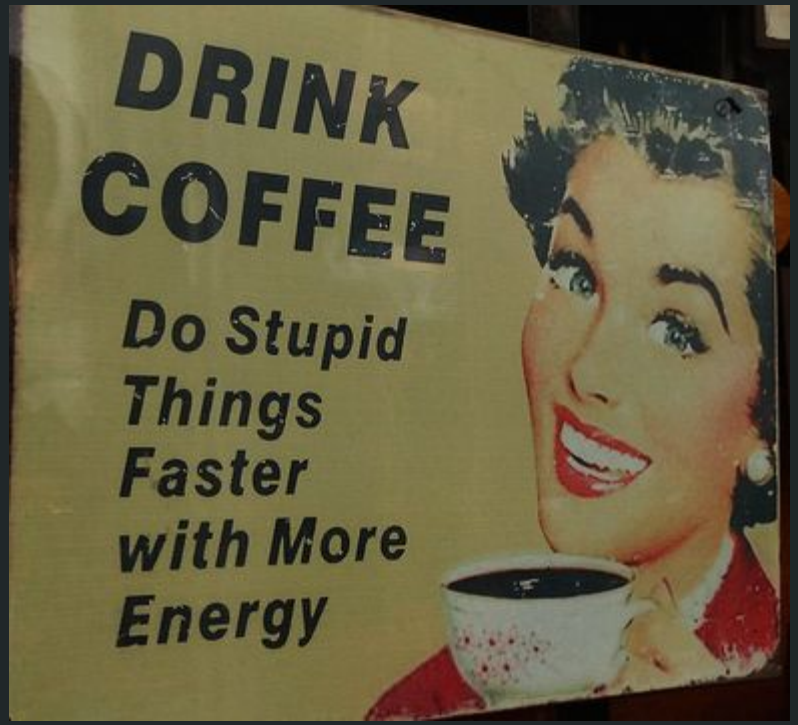
Teach Ansible, Python, Nornir in a Network Automation context

General:

Dec 20, Day1 (Mon) / 9AM-5PM Central
Dec 21, Day2 (Tue)
Dec 22, Day3 (Wed)
Dec 23, Day4 (Thu)

Focused/Minimize Distractions

The exercises are important.



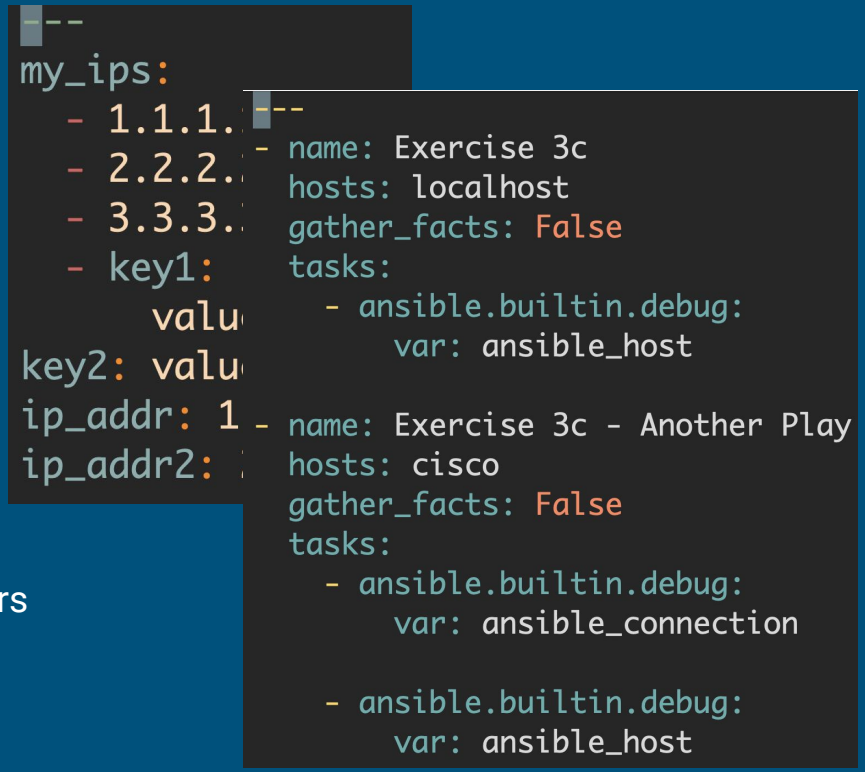
Flickr: Ben Sutherland

https://gems.vzw.com/checkin.php?id=REhOMQ%3D%3D

Verizon GEMS Check-In

Day1 Schedule - Ansible Fundamentals

YAML
Ansible Overview
Installing Ansible
ansible-cfg
Inventory
Playbook Structure
Executing Your First Playbook
Ansible Execution Behavior
Variables
Modules
Inventory and group_vars/host_vars
Tags/Limit/Check Mode
Conditionals
Loops



YAML

Reference Material in:
{{ github_repo }}/yaml_examples

Lists:

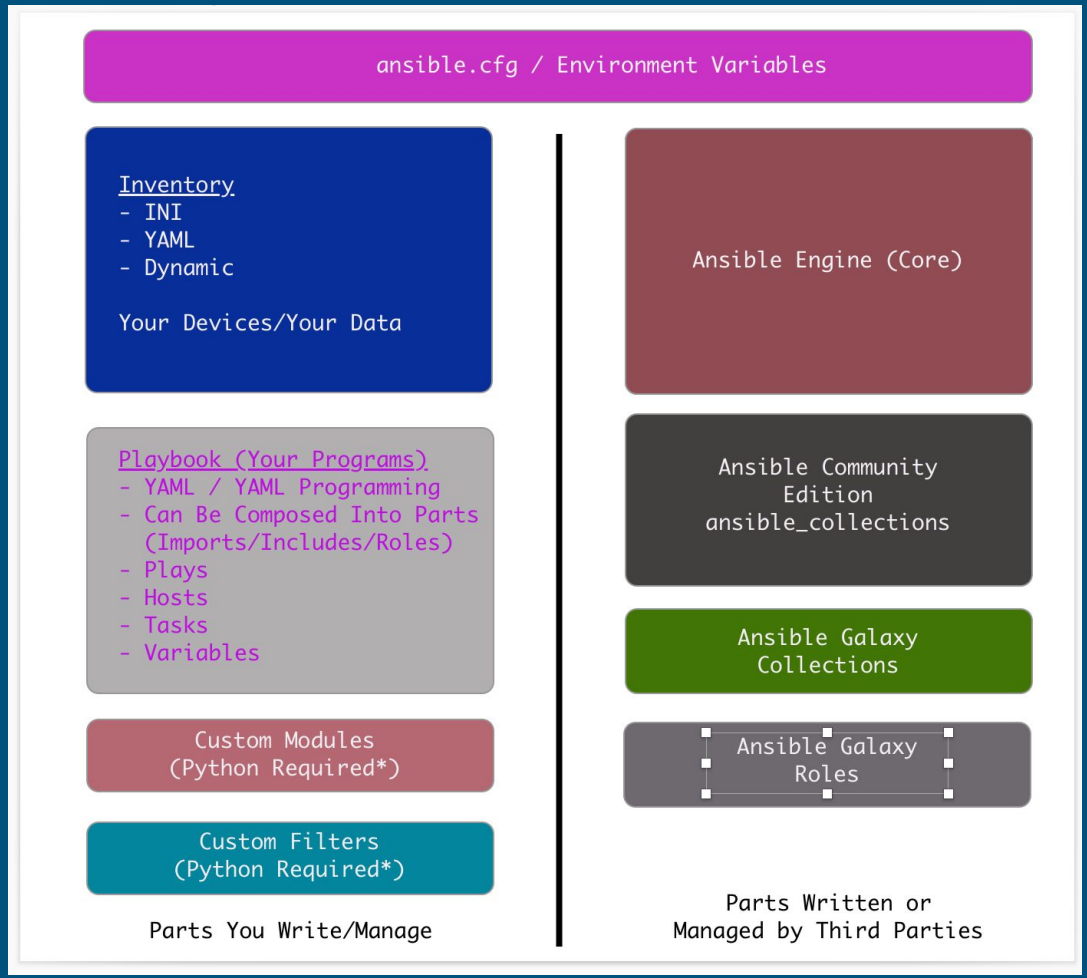
```
---
san_francisco:
  - 10.71.1.0/24
  - 10.71.2.0/24
  - 10.71.3.0/24
los_angeles:
  - '10.99.100.0/24'
  - '10.99.101.0/24'
  - '10.99.102.0/24'
denver:
  - "10.201.100.0/24"
  - "10.201.101.0/24"
  - "10.201.102.0/24"
```

Dictionaries:

```
---
san_francisco:
  rtr1:
    ip_addr: 10.71.1.1
    device_type: cisco_nxos
    username: admin
    password: cisco123
los_angeles:
  rtr1:
    ip_addr: 10.99.100.1
    device_type: cisco_nxos
    username: admin
    password: cisco123
```

Exercises:
./day1/yaml/ex1.txt

Ansible High-Level Overview



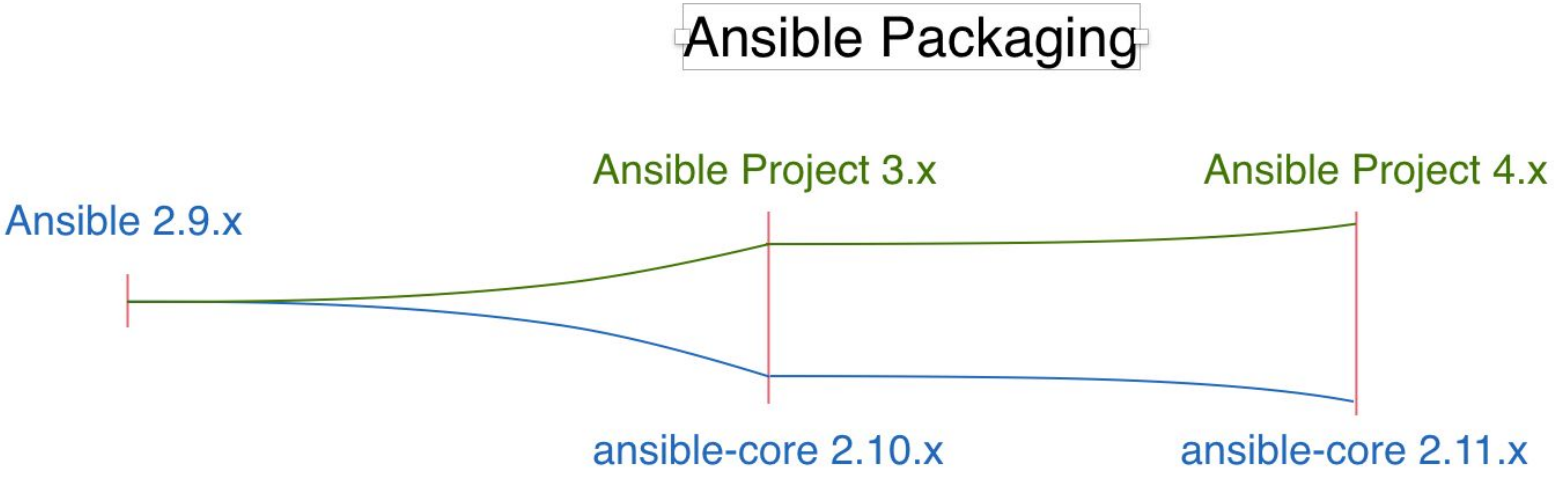
Installing Ansible

1. Create a Virtual Environment
2. PIP install ansible and ansible-core
3. Setup .ansible.cfg
4. Setup your inventory

```
$ pip list | grep ansible
ansible                                  4.9.0
ansible-core                             2.11.7
```

```
[defaults]
inventory = ~/ansible-hosts.ini
library = ~/ansible-extras/ntc-ansible/library
filter_plugins = ~/ansible-extras/ntc-ansible/filter_plugins
host_key_checking = False
retry_files_enabled = False
action_warnings = False
deprecation_warnings = False

[persistent_connection]
command_timeout = 40
```

Ansible Packaging



Ansible Config File

https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file

Changes can be made and used in a configuration file which will be searched for in the following order:

- `ANSIBLE_CONFIG` (environment variable if set)
- `ansible.cfg` (in the current directory)
- `~/.ansible.cfg` (in the home directory)
- `/etc/ansible/ansible.cfg`

Ansible will process the above list and use the first file found, all others are ignored.


```
[defaults]
inventory = ~/ansible-hosts.ini
library = ~/ansible-extras/ntc-ansible/library
filter_plugins = ~/ansible-extras/ntc-ansible/filter_plugins
host_key_checking = False
retry_files_enabled = False
action_warnings = False
deprecation_warnings = False

[persistent_connection]
command_timeout = 40
```

Ansible Inventory

Reference Material in:
{{ github_repo }}/inventory_examples

```
[all:vars]
ansible_connection=network_cli
ansible_python_interpreter=/home/ktbyers/VENV/ansible/bin/python
ansible_user=pyclass
ansible_ssh_pass=invalid

[local]
localhost ansible_connection=local

[arista]
arista5 ansible_host=arista5.lasthop.io
arista6 ansible_host=arista6.lasthop.io
arista7 ansible_host=arista7.lasthop.io
arista8 ansible_host=arista8.lasthop.io
```

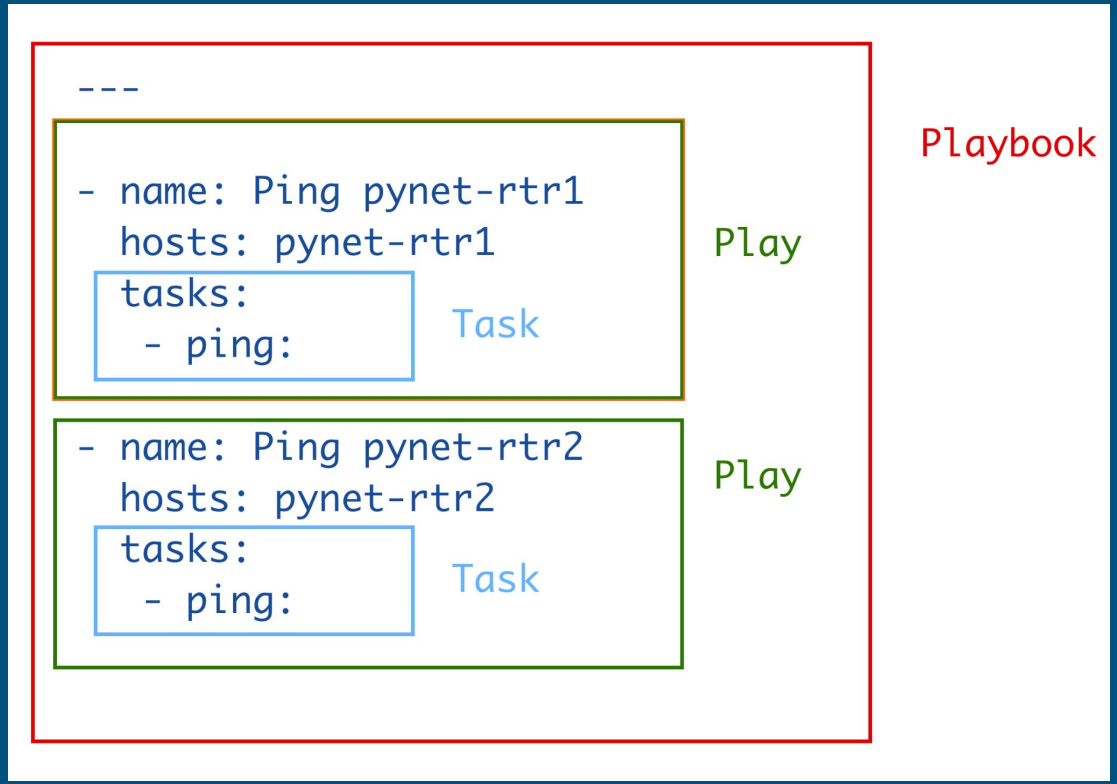
Ansible Inventory

```
[nxos:vars]
ansible_network_os=nxos
#ansible_connection=httpapi
ansible_httpapi_use_ssl=True
ansible_httpapi_validate_certs=False
ansible_httpapi_port=8443

[san_francisco:children]
arista
nxos

[denver:children]
cisco
nxos

[cisco_all:children]
cisco
nxos
```

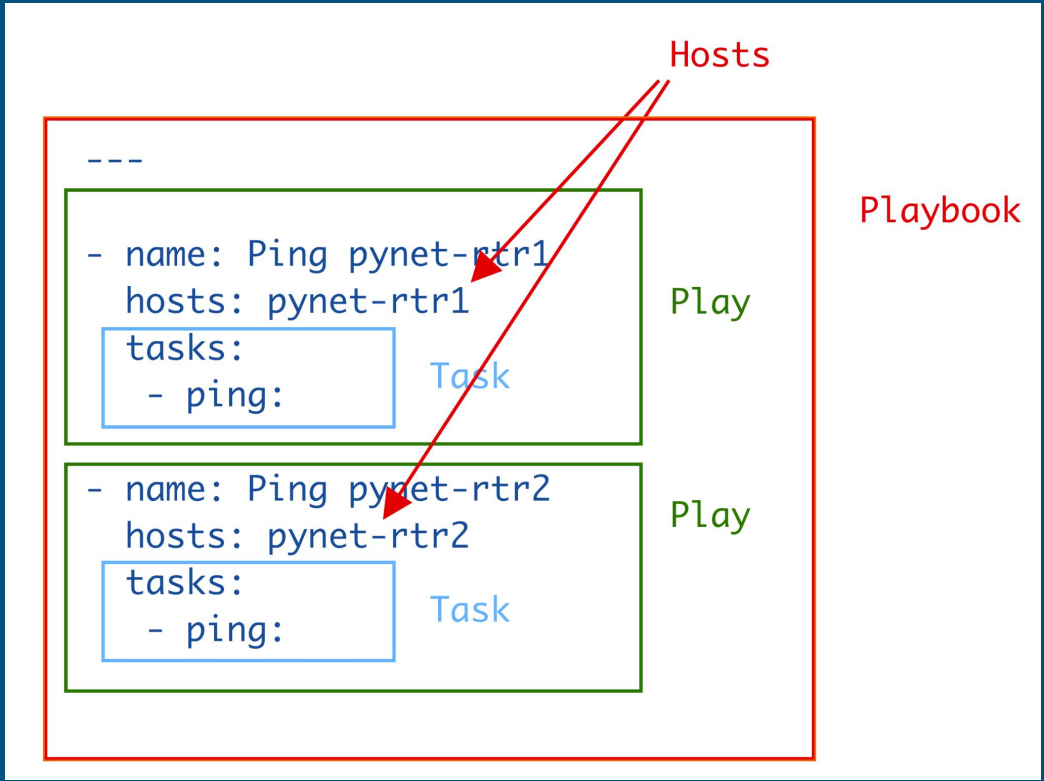
```
$ ansible-inventory -i ./ansible-hosts-alt.ini --list | jq "."
{
  "_meta": {
    "hostvars": {
      "arista5": {
        "ansible_connection": "network_cli",
        "ansible_host": "arista5.lasthop.io",
        "ansible_httpapi_use_ssl": true,
        "ansible_httpapi_validate_certs": false,
        "ansible_network_os": "eos",
        "ansible_python_interpreter": "/home/ktbyers/VENV/ansible/bin/python",
        "ansible_ssh_pass": "invalid",
        "ansible_user": "pyclass"
      },
```

Exercises:
./day1/inventory/ex1.txt
./day1/inventory/ex2.txt
./day1/inventory/ex3.txt

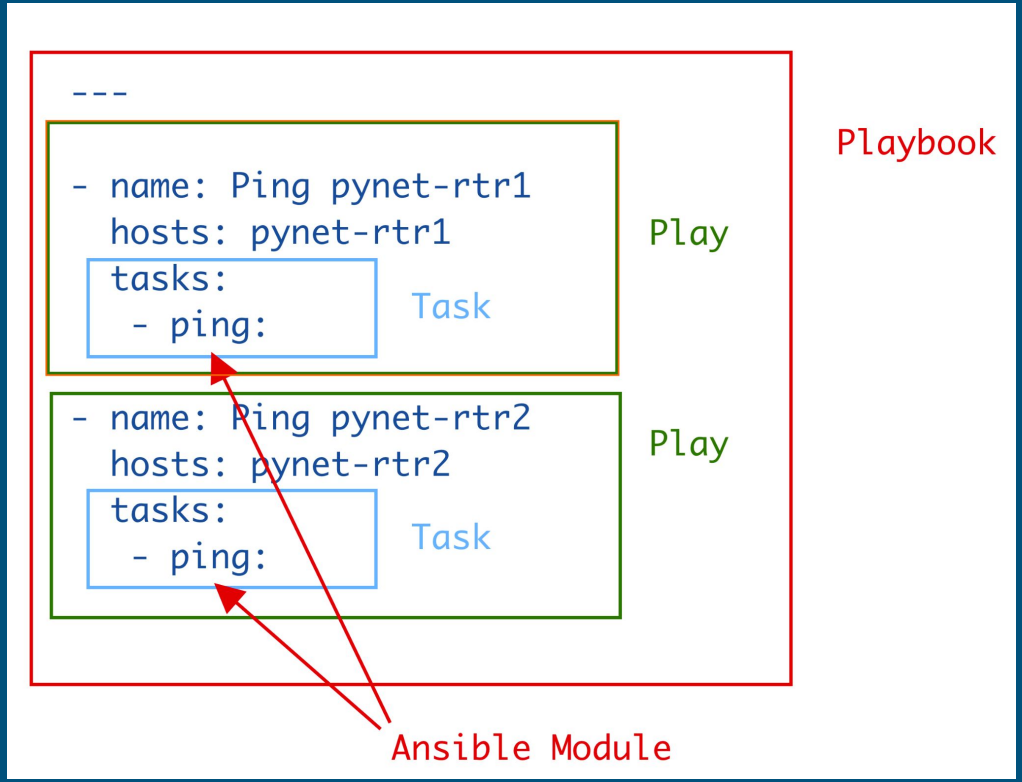
Ansible Playbook Structure



Ansible Playbook Structure



Ansible Playbook Structure



Ansible Playbook Structure

Play

Play Keywords:
name,
hosts,
tasks

Tasks

Task Keywords:
name,
{{ module_name }}

```yaml
---
- name: Print out variables
  hosts: cisco5
  gather_facts: False
  vars:
    dns1: 8.8.8.8
    dns2: 8.8.4.4
  vars_files:
    - my_vars.yml

  tasks:
    - name: Testing $var locally defined
      ansible.builtin.debug:
        msg: "{{ dns1 }}"

    - name: Testing $var from vars_files
      ansible.builtin.debug:
        msg: "{{ ip_addr1 }}"
```

First Playbook

Play

Play Keywords:
name,
hosts,
gather_facts,
tasks

Tasks

Task Keywords:
name,
{{ module_name }}

```yaml
---
- name: Simple playbook
  hosts: cisco5
  gather_facts: False

  tasks:
    - name: First task
      ansible.builtin.debug:
        msg: "Hello world"
```

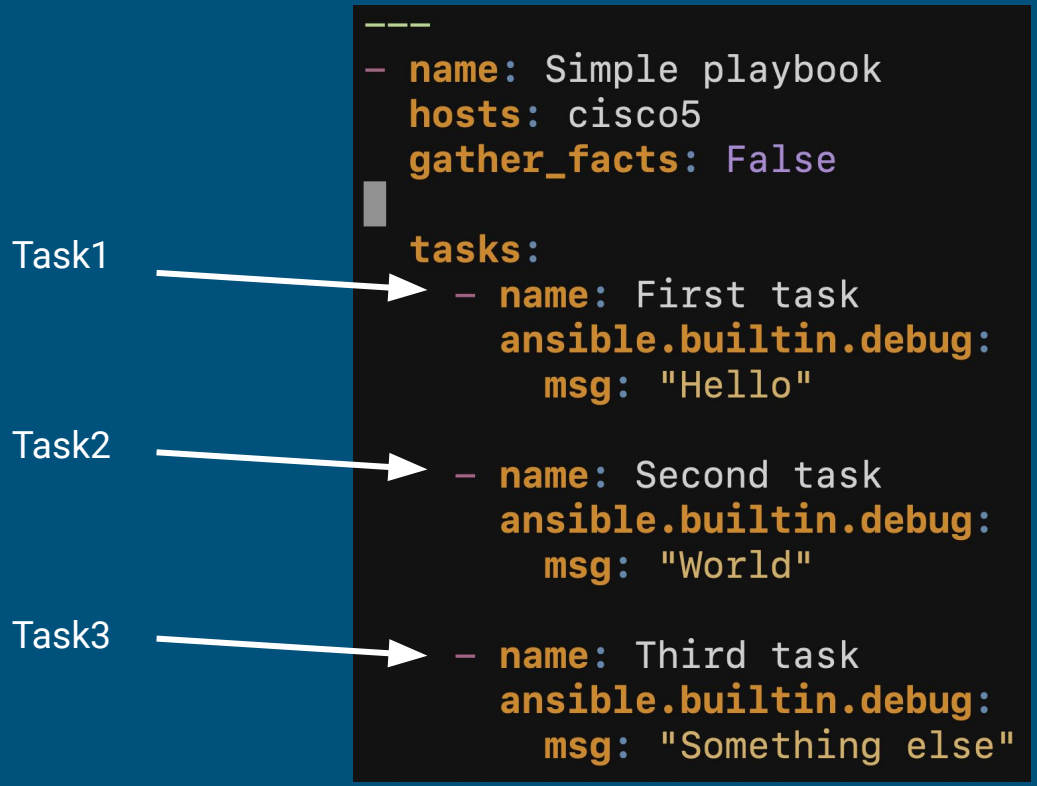
First Playbook: Execution


```
$ ansible-playbook first_pb.yml

PLAY [Simple playbook] *********************************************************

TASK [First task] **************************************************************
ok: [cisco5] => {
    "msg": "Hello world"
}

PLAY RECAP *********************************************************************
cisco5                     : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued
=0    ignored=0
```

Reference Material in:

{{ github_repo }}/playbook_examples

First Playbook (expanded)



First Playbook: Execution (expanded)


```
$ ansible-playbook pb_multi_tasks.yml

PLAY [Simple playbook] *********************************************************************

TASK [First task] **************************************************************************
ok: [cisco5] => {
    "msg": "Hello"
}

TASK [Second task] *************************************************************************
ok: [cisco5] => {
    "msg": "World"
}

TASK [Third task] **************************************************************************
ok: [cisco5] => {
    "msg": "Something else"
}

PLAY RECAP *********************************************************************************
cisco5                     : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued
=0    ignored=0
```

First Playbook (multiple plays)

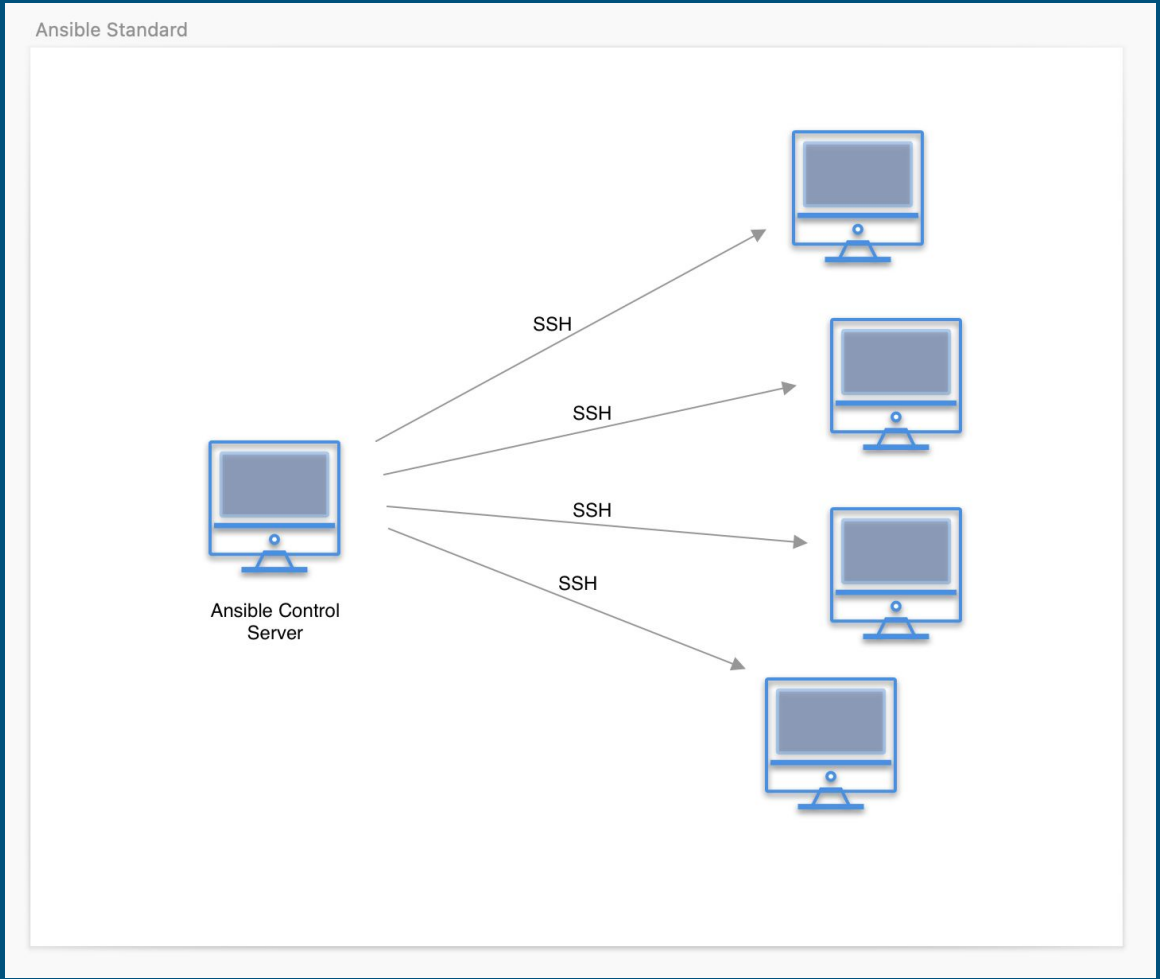
Play1 →

Play2 →

Note: the hosts changed

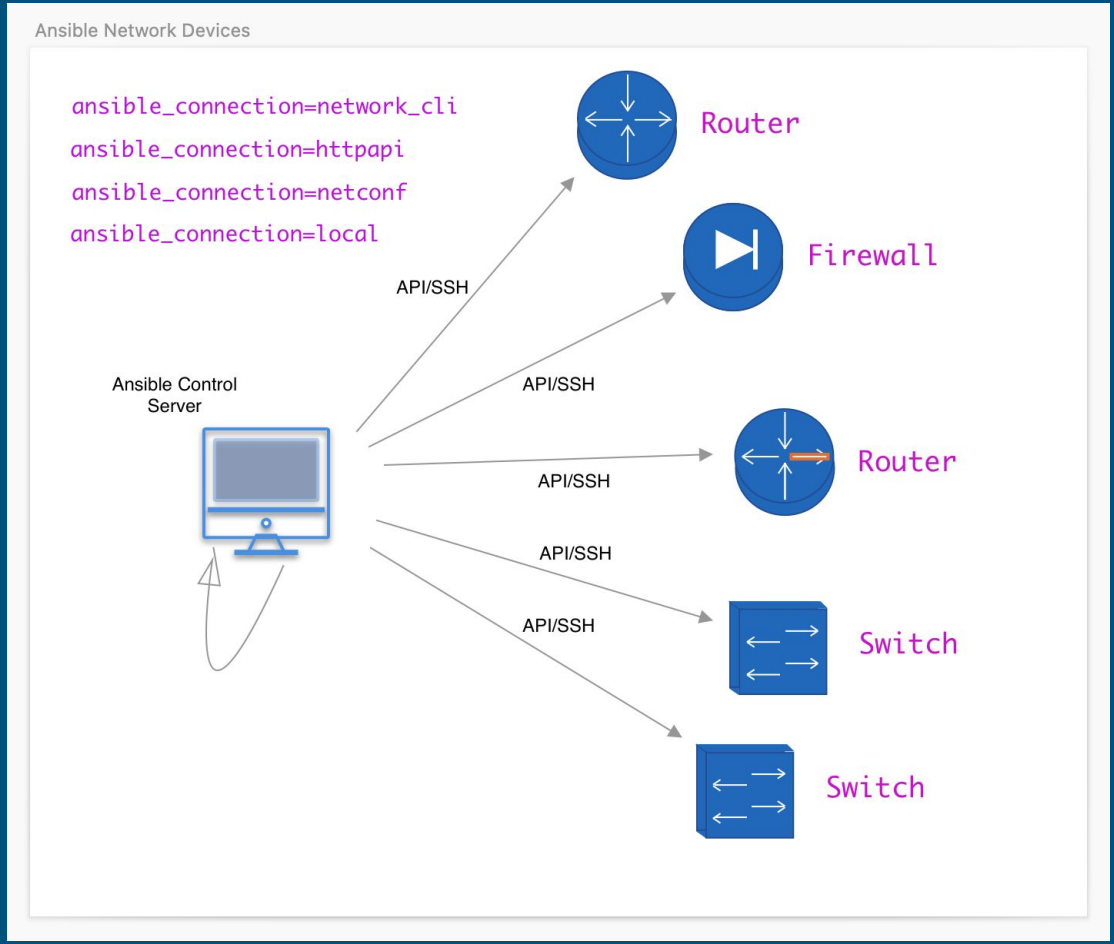
```yaml
---
- name: Play1
  hosts: cisco5
  gather_facts: False

  tasks:
    - name: Hello task
      ansible.builtin.debug:
        msg: "Hello"

- name: Play2
  hosts: cisco6
  gather_facts: False

  tasks:
    - name: Hello task
      ansible.builtin.debug:
        msg: "Hello"
```

Ansible Execution Behavior (Standard)

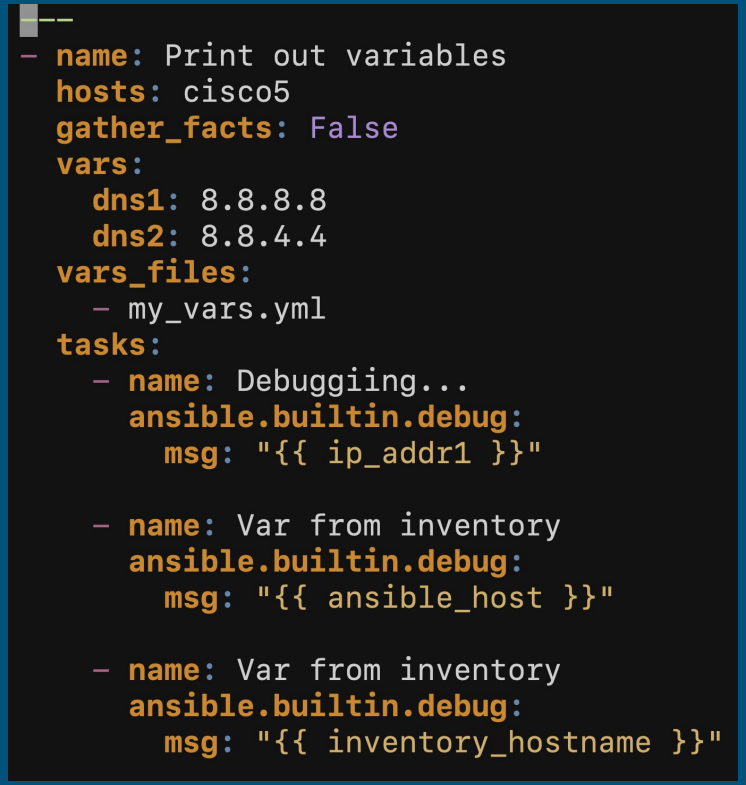


Ansible Execution Behavior: (connection: local)

Exercises:
./day1/first_playbook/ex1.txt



Ansible Variables



```yaml
---
- name: Print out variables
  hosts: cisco5
  gather_facts: False
  vars:
    dns1: 8.8.8.8
    dns2: 8.8.4.4
  vars_files:
    - my_vars.yml
  tasks:
    - name: Debuggiing...
      ansible.builtin.debug:
        msg: "{{ ip_addr1 }}"

    - name: Var from inventory
      ansible.builtin.debug:
        msg: "{{ ansible_host }}"

    - name: Var from inventory
      ansible.builtin.debug:
        msg: "{{ inventory_hostname }}"
```

Why do we need? “{{

Why sometimes do we NOT need? “{{

Reference Material in:
{{ github_repo }}/variable_examples

Variables from a lot of sources / Ansible Variable Precedence

1. command line values (for example, `-u my_user`, these are not variables)
2. role defaults (defined in role/defaults/main.yml) [1]
3. inventory file or script group vars [2]
4. inventory group_vars/all [3]
5. playbook group_vars/all [3]
6. inventory group_vars/* [3]
7. playbook group_vars/* [3]
8. inventory file or script host vars [2]
9. inventory host_vars/* [3]
10. playbook host_vars/* [3]
11. host facts / cached set_facts [4]
12. play vars
13. play vars_prompt
14. play vars_files
15. role vars (defined in role/vars/main.yml)
16. block vars (only for tasks in block)
17. task vars (only for the task)
18. include_vars
19. set_facts / registered vars
20. role (and include_role) params
21. include params
22. extra vars (for example, `-e "user=my_user"`)(always win precedence)

Lowest precedence to highest precedence

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#understanding-variable-precedence

Exercises:
./day1/variables/ex1.txt
./day1/variables/ex2.txt
./day1/variables/ex3.txt

Ansible Variables: Extracting Information


```yaml
- name: Extract Serial Number
  ansible.builtin.set_fact:
    serial_num: "{{ output['ansible_facts']['ansible_net_serialnum'] }}"

- name: Extract Serial Number (dot notation)
  ansible.builtin.set_fact:
    serial_num_alt: "{{ output.ansible_facts.ansible_net_serialnum }}"

- name: Print Serial Number
  ansible.builtin.debug:
    msg: |
      "{{ serial_num }}"
      "{{ serial_num_alt }}"
```

Ansible Module Documentation

https://docs.ansible.com/ansible/latest/collections/index.html  Collections Index

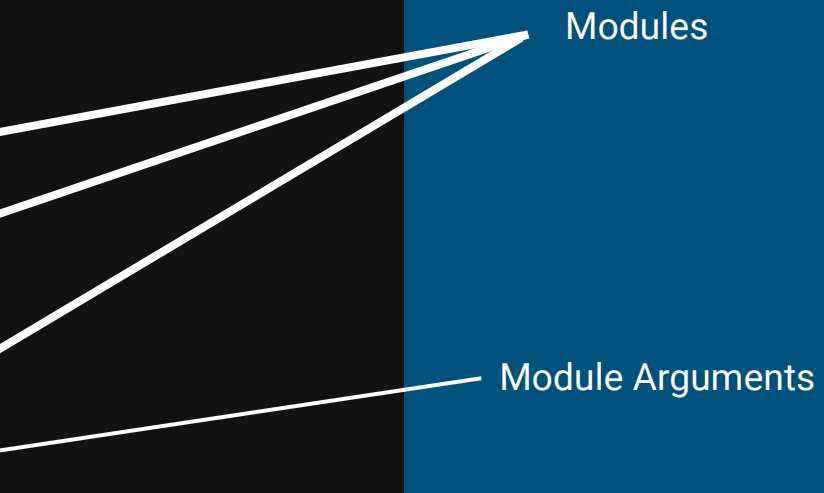
Collections in the Cisco Namespace

These are the collections with docs hosted on docs.ansible.com in the **cisco** namespace.

- cisco.aci
- cisco.asa
- cisco.intersight
- cisco.ios
- cisco.iosxr
- cisco.ise
- cisco.meraki
- cisco.mso
- cisco.nso
- cisco.nxos
- cisco.ucs

Ansible Modules

```yaml
---
- name: Simple Playbook
  hosts: servers
  tasks:
    - ansible.builtin.ping:

    - ansible.builtin.debug:
        msg: Hello World

    - name: Install Netmiko
      ansible.builtin.pip:
        name: netmiko
        virtualenv: /home/ktbyers/VENV/py3_venv
        version: 3.3.3
```



ANSIBLE

Expanding Inventory with host_vars and group_vars

```
$ tree -C host_vars/
host_vars/
├── nxos1
│   ├── bgp.yml
│   └── interfaces.yml
└── nxos2
    ├── bgp.yml
    └── interfaces.yml

2 directories, 4 files
```

```
$ tree -C group_vars/
group_vars/
└── nxos
    └── bgp.yml
```

$ cat group_vars/nxos/bgp.yml

```
---
bgp_asn: 22
```

$ cat host_vars/nxos1/interfaces.yml

```
---
eth1_4_ip_address: 172.31.254.1
eth1_4_netmask: 30

loopback101_ip_address: 172.31.101.101
loopback101_netmask: 32

loopback102_ip_address: 172.31.102.101
loopback102_netmask: 32
```

1. Place in inventory file directory
2. Place in playbook directory

Exercises:
./day1/host_group_vars/ex1.txt
./day1/host_group_vars/ex2.txt

Tags, Limit, and Check Mode

```yaml
tasks:
  - name: First task
    ansible.builtin.debug:
      msg: "Hello"
    tags: task1

  - name: Second task
    ansible.builtin.debug:
      msg: "World"
    tags: always

  - name: Third task
    ansible.builtin.debug:
      msg: "Something else"
    tags:
      - task3
      - cisco
```

```
$ ansible-playbook tags.yml --list-tags

playbook: tags.yml

  play #1 (cisco5): Simple playbook     TAGS: []
      TASK TAGS: [always, cisco, task1, task3]
```

```
$ ansible-playbook tags.yml --tags task1
```

```
$ ansible-playbook limit.yml --limit cisco6
```

```
$ ansible-playbook limit.yml --check
```

Conditionals (when)


```yaml
tasks:
  - name: Testing string equality
    ansible.builtin.debug:
      msg: whatever
    when: my_str == 'whatever'
```

Jinja2 Context

```yaml
tasks:
  - name: Substring in larger string
    ansible.builtin.debug:
      msg: This is Cisco IOS
    when: "'Cisco IOS' in version"
```

```yaml
tasks:
  - name: This variable is defined
    ansible.builtin.debug:
      msg: Variable defined
    when: version is defined
```

Exercises:
./day1/conditionals/ex1.txt
./day1/conditionals/ex2.txt
./day1/conditionals/ex3.txt

Reference Material in:
{{ github_repo }}/conditional_examples

Loopty-loops (loop and with_*)

```yaml
---
- name: With Testing
  hosts: local
  tasks:
    - name: Test with_items
      ansible.builtin.debug:
        msg: "{{ item }}"
      with_items:
        - 10.10.10.1
        - 10.10.20.1
        - 10.10.30.1
        - 10.10.40.1
        - 10.10.50.1

    - name: Test with_items (loop)
      ansible.builtin.debug:
        msg: "{{ item }}"
      loop:
        - 10.10.10.1
        - 10.10.20.1
        - 10.10.30.1
        - 10.10.40.1
        - 10.10.50.1
```

Loop variable: {{ item }}

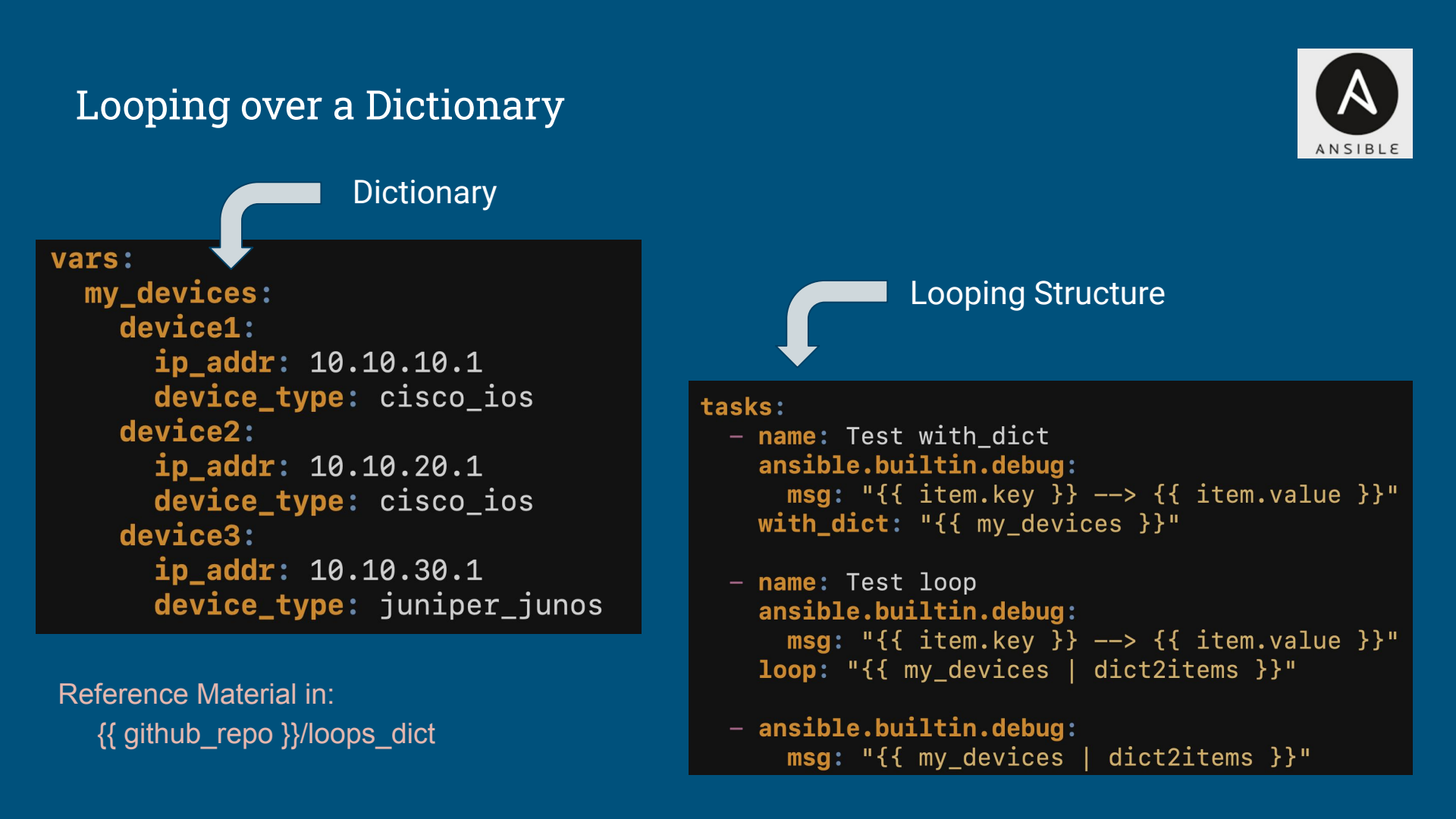
Reference Material in:

{{ github_repo }}/loop_examples

Looping over list-of-lists (flattening)


```yaml
vars:
  net_devices:
    - ["rtr1", "rtr2", "rtr3", "rtr4"]
    - ["sw1", "sw2", "sw3", "sw4"]

tasks:
  - name: Example of list flattening
    ansible.builtin.debug:
      msg: "{{ item }}"
    with_items: "{{ net_devices }}"

  - name: Do not flatten
    ansible.builtin.debug:
      msg: "{{ item }}"
    with_list: "{{ net_devices }}"

  - name: No flattening
    ansible.builtin.debug:
      msg: "{{ item }}"
    loop: "{{ net_devices }}"
```

Looping over a Dictionary




```yaml
vars:
  my_devices:
    device1:
      ip_addr: 10.10.10.1
      device_type: cisco_ios
    device2:
      ip_addr: 10.10.20.1
      device_type: cisco_ios
    device3:
      ip_addr: 10.10.30.1
      device_type: juniper_junos
```

Reference Material in:

{{ github_repo }}/loops_dict


```yaml
tasks:
  - name: Test with_dict
    ansible.builtin.debug:
      msg: "{{ item.key }} --> {{ item.value }}"
    with_dict: "{{ my_devices }}"

  - name: Test loop
    ansible.builtin.debug:
      msg: "{{ item.key }} --> {{ item.value }}"
    loop: "{{ my_devices | dict2items }}"

  - ansible.builtin.debug:
      msg: "{{ my_devices | dict2items }}"
```

Loops AND conditionals

```yaml
vars:
  my_devices:
      - 10.10.10.1
      - 10.10.20.1
      - 10.10.30.1
      - 10.10.40.1
      - 10.10.50.1
tasks:
  - name: Test with_items
    ansible.builtin.debug:
      msg: "{{ item }}"
    with_items: "{{ my_devices }}"
    when: "item == '10.10.10.1'"

  - name: Test using loop
    ansible.builtin.debug:
      msg: "{{ item }}"
    loop: "{{ my_devices }}"
    when: "item == '10.10.40.1'"
```

Reference Material in:

{{ github_repo }}/loops_and_when

Exercises:

./day1/loops/ex1.txt
./day1/loops/ex2.txt
./day1/loops/ex3.txt

https://gems.vzw.com/checkin.php?id=REhOMQ%3D%3D

Verizon GEMS Check-In