>>> network .toCode()

Introduction to Ansible for Network Automation

Hector Isaza
Network to Code

Who am I?

- I'm a Network Automation Engineer and Trainer for Network to Code
- My job is to teach and empower the network engineers of the future.
- I started as a network engineer and then started getting into software development a few years ago.

Agenda

- What is Ansible?
- Why Ansible?
- Ansible High Level Architecture
- Ansible Network Automation Use Cases
- Quick Demo

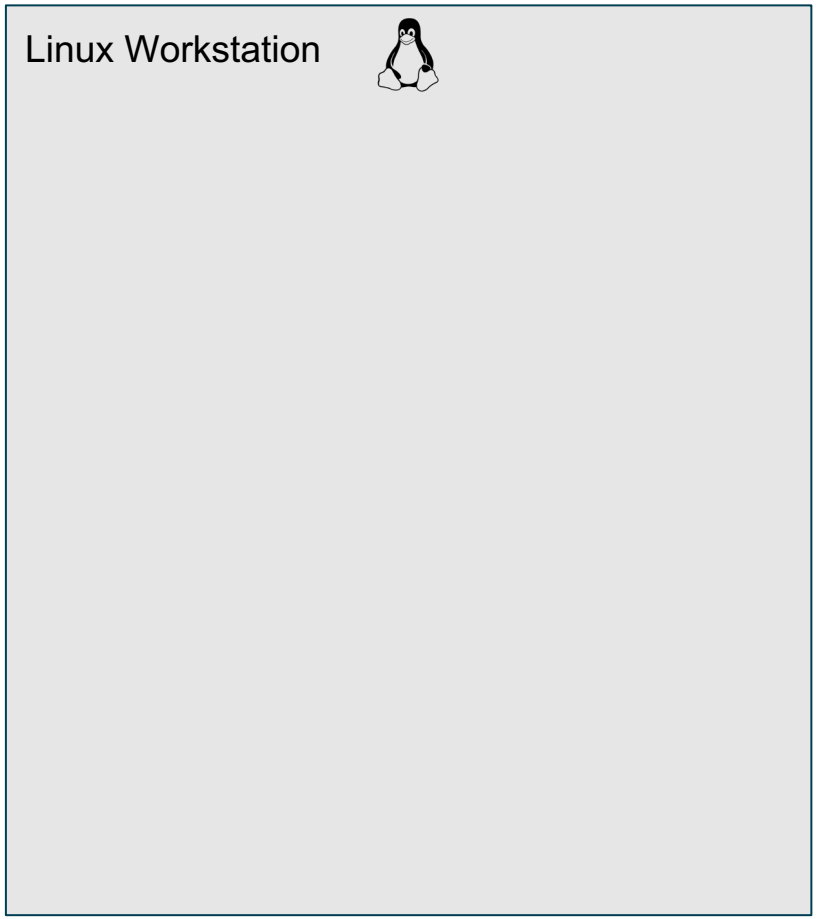
What is Ansible?

- Open source DevOps Configuration management, automation, and orchestration platform
- Started in systems and cloud environments
- Thousands of modules for cloud, monitoring, windows, storage, etc. and now networking
- Exploded in the network automation field

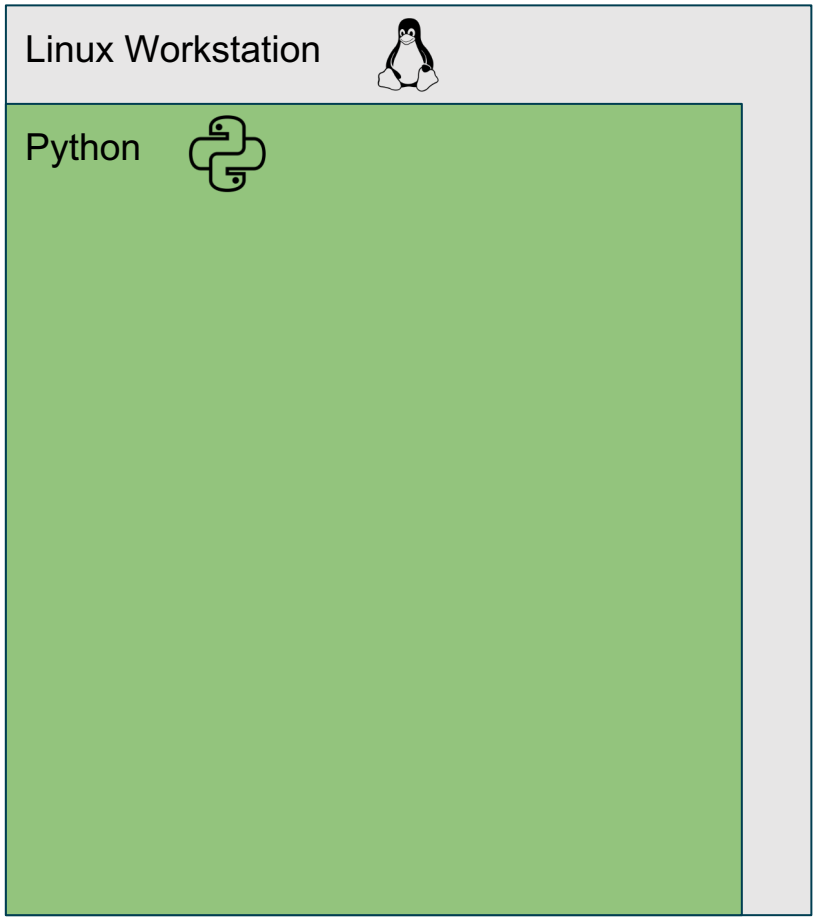
Why Ansible?

- Written in Python
- Low[er] barrier to entry (Get up and running quickly)
- Can be extended in any language (not common for open source modules)
- Automation instructions are defined in YAML
- Native integration with Jinja2 templates
- Agentless

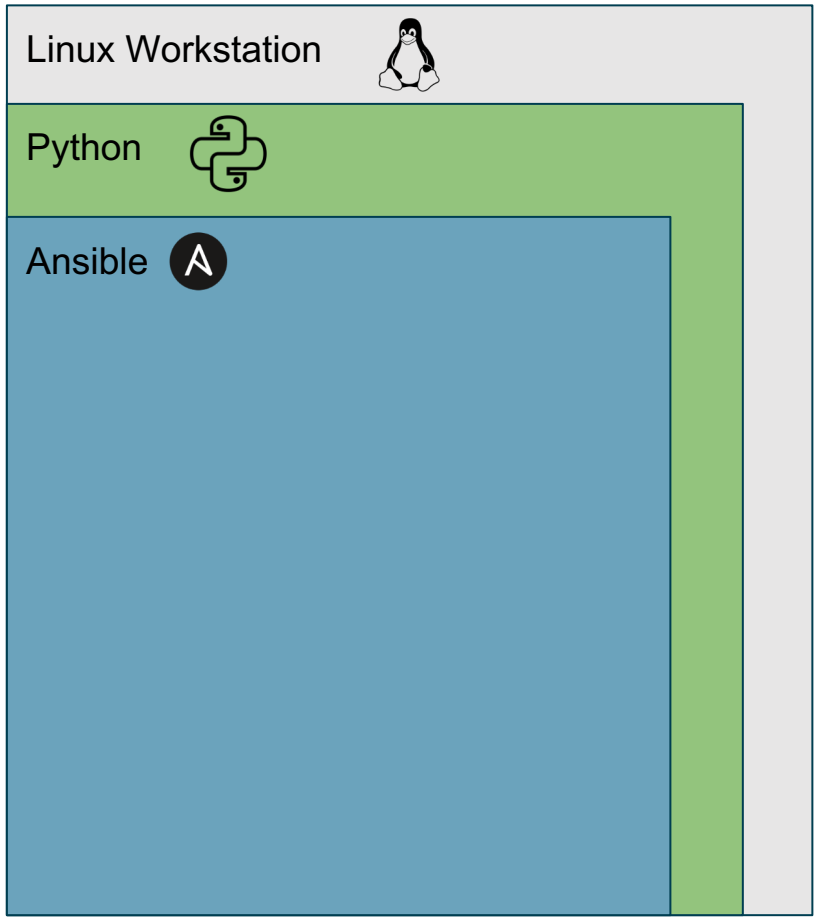
Ansible High Level Architecture



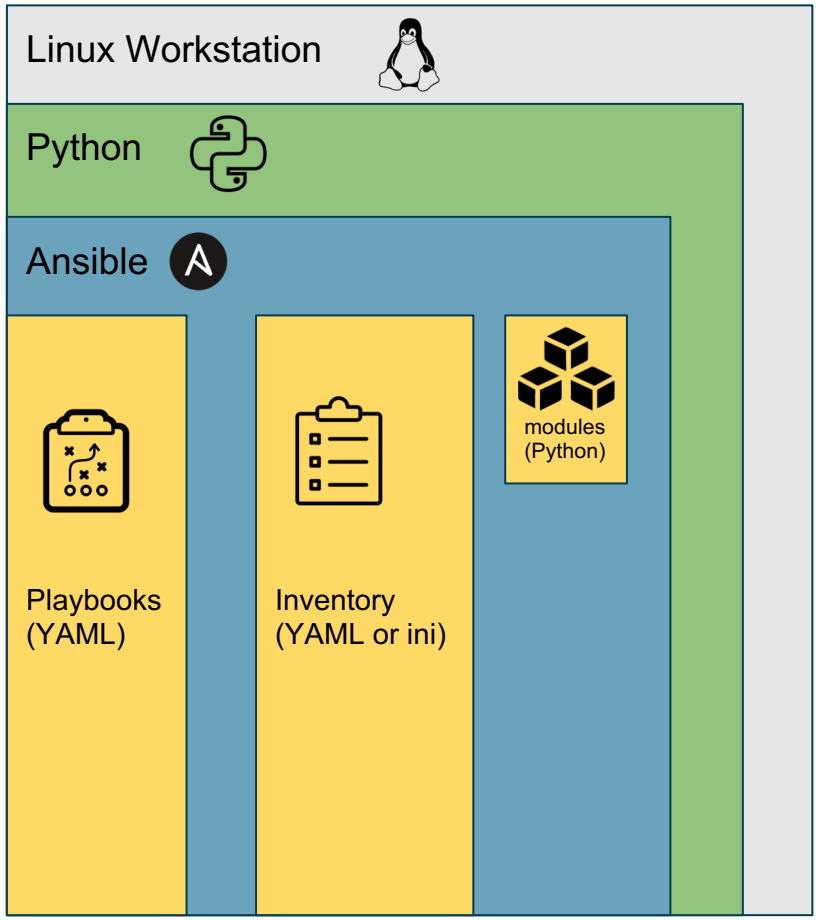
Ansible High Level Architecture



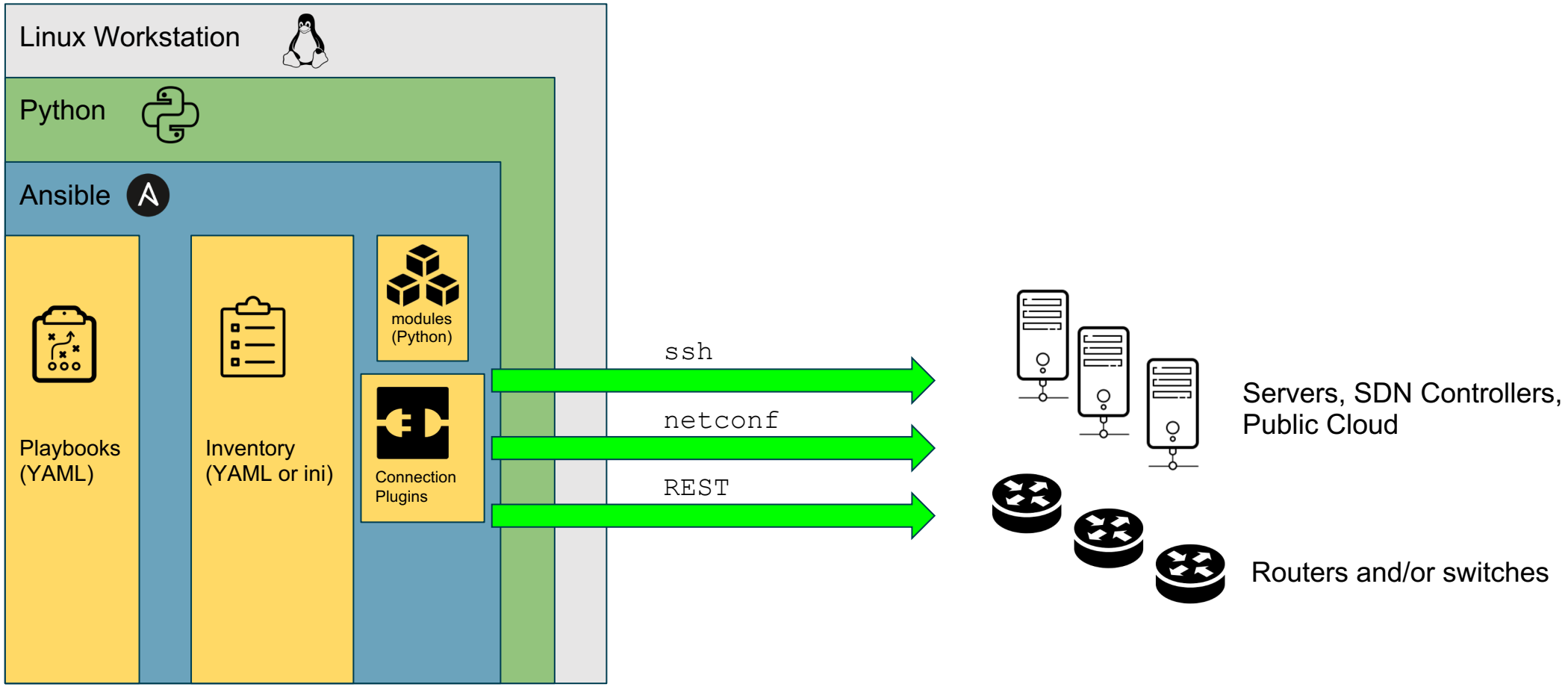
Ansible High Level Architecture



Ansible High Level Architecture



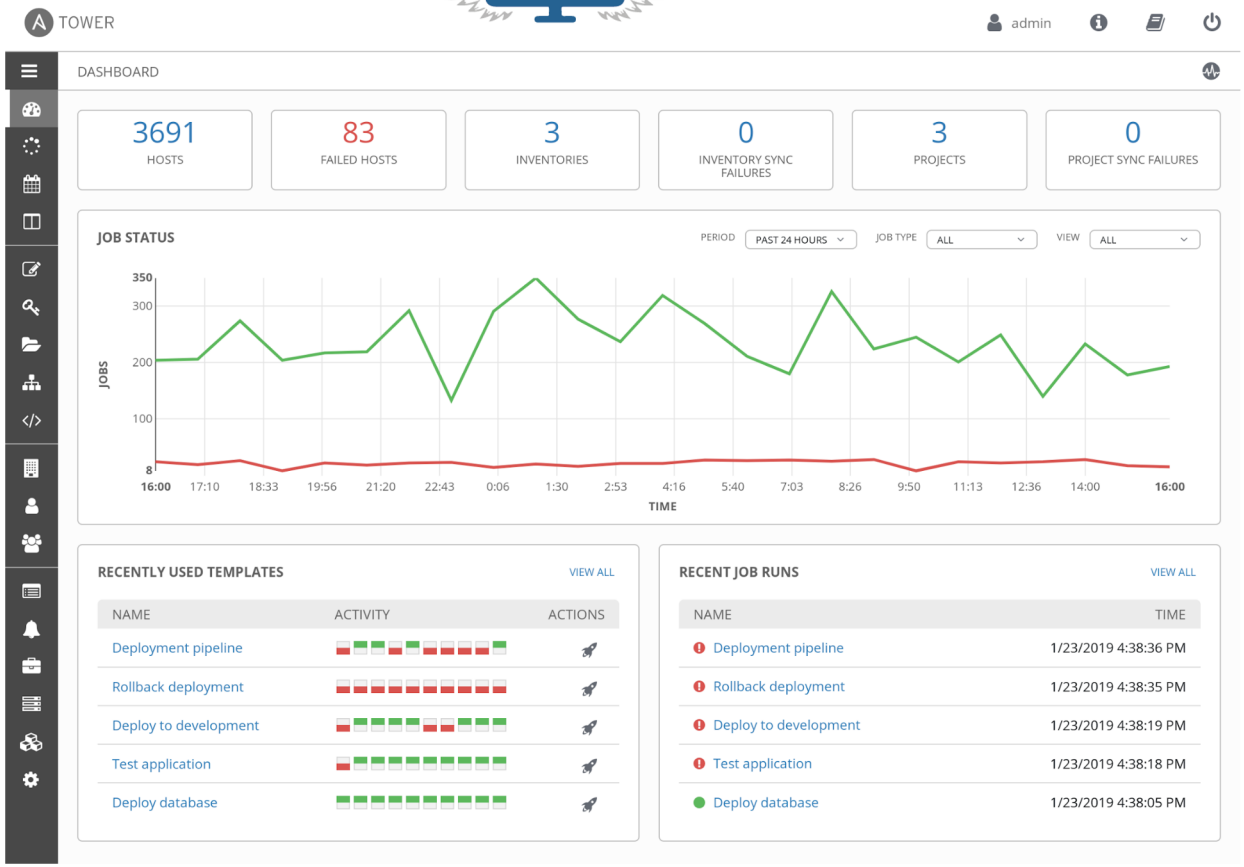
Ansible High Level Architecture



Ansible AWX & Tower

RED HAT® ANSIBLE® Tower

- Enterprise Features layered on top of Ansible Engine including API & UI
- Offers greater control and tooling:
  - Role Based access control
  - Job Scheduling
  - Graphical inventory management
  - Real Time job status updates
- AWX: Free and Open Source
- Tower: Commerical deritative of AWX that includes Red Hat support



Ansible Network Automation Use Cases

Use Case #1 - Updating SNMP Community Strings

Inventory File

```
[iosxe]
csr1
csr2
csr3

[iosxe:vars]
ansible_user=admin
ansible_ssh_pass=cisco123
ansible_network_os=ios
snmp_com=PUBLIC
snmp_contact=Network_to_Code
snmp_location=New_York
```

Note: Use YAML files for production

Use Case #1 - Updating SNMP Community Strings

Inventory File

```
[iosxe]
csr1
csr2
csr3

[iosxe:vars]
ansible_user=admin
ansible_ssh_pass=cisco123
ansible_network_os=ios
snmp_com=PUBLIC
snmp_contact=Network_to_Code
snmp_location=New_York
```

Note: Use YAML files for production

Ansible Playbook

```
---

- name: DEPLOY SNMP COMMUNITY STRINGS ON IOS DEVICES
  hosts: iosxe
  connection: network_cli
  gather_facts: no

  tasks:
```

Use Case #1 - Updating SNMP Community Strings

Inventory File

```
[iosxe]
csr1
csr2
csr3

[iosxe:vars]
ansible_user=admin
ansible_ssh_pass=cisco123
ansible_network_os=ios
snmp_com=PUBLIC
snmp_contact=Network_to_Code
snmp_location=New_York
```

Note: Use YAML files for production

Ansible Playbook

```
---

- name: DEPLOY SNMP COMMUNITY STRINGS ON IOS DEVICES
  hosts: iosxe
  connection: network_cli
  gather_facts: no

  tasks:

    - name: USE COMMANDS IN THE PLAYBOOK
      ios_config:
        commands:
          - "snmp-server community {{ snmp_com }} RO"
          - "snmp-server contact {{ snmp_contact }}"
          - "snmp-server location {{ snmp_location }}"

    - name: DEPLOY FROM CONFIG FILE
      ios_config:
        src: "configs/snmp.cfg"

    - name: DEPLOY USING JINJA2 TEMPLATE
      ios_config:
        src: "snmp.j2"
```

Use Case #1 - Updating SNMP Community Strings

Inventory File

```
[iosxe]
csr1
csr2
csr3

[iosxe:vars]
ansible_user=admin
ansible_ssh_pass=cisco123
ansible_network_os=ios
snmp_com=PUBLIC
snmp_contact=Network_to_Code
snmp_location=New_York
```

Note: Use YAML files for production

Ansible Playbook

```
---

- name: DEPLOY SNMP COMMUNITY STRINGS ON IOS DEVICES
  hosts: iosxe
  connection: network_cli
  gather_facts: no

  tasks:

    - name: USE COMMANDS IN THE PLAYBOOK
      ios_config:
        commands:
          - "snmp-server community {{ snmp_com }} RO"
          - "snmp-server contact {{ snmp_contact }}"
          - "snmp-server location {{ snmp_location }}"

    - name: DEPLOY FROM CONFIG FILE
      ios_config:
        src: "configs/snmp.cfg"

    - name: DEPLOY USING JINJA2 TEMPLATE
      ios_config:
        src: "snmp.j2"
```

Use Case #2 - Generate LLDP Neighbor Reports

Playbook (ONE TASK!)

```
---

- name: DC P1
  hosts: nxos-spine1
  connection: network_cli
  gather_facts: yes

  tasks:

    - name: BUILD MARKDOWN TABLE
      template:
        src: "neighbors.j2"
        dest: "files/neighbors-table.md"
```

Use Case #2 - Generate LLDP Neighbor Reports

Playbook (ONE TASK!)

```
---

 - name: DC P1
   hosts: nxos-spine1
   connection: network_cli
   gather_facts: yes

   tasks:

     - name: BUILD MARKDOWN TABLE
       template:
         src: "neighbors.j2"
         dest: "files/neighbors-table.md"
```

Jinja2 Template

```
# neighbors.j2
| Source     | Interface    |         Neighbor          | Interface   |
| ---------- |--------------|---------------------------|-------------|
{% for local_int, details in ansible_net_neighbors.items() %}
{% for neigh_data in details %}
| {{ inventory_hostname }}| {{ local_int }}  |  {{ neigh_data.sysname }} | {{ neigh_data.port  }} |
{% endfor %}
{% endfor %}
```

Use Case #2 - Generate LLDP Neighbor Reports

Playbook (ONE TASK!)

```yaml
---

- name: DC P1
  hosts: nxos-spine1
  connection: network_cli
  gather_facts: yes

  tasks:

    - name: BUILD MARKDOWN TABLE
      template:
        src: "neighbors.j2"
        dest: "files/neighbors-table.md"
```

Markdown Report

Source	Interface	Neighbor	Interface
nxos-spine1	Ethernet2/4	nxos-spine2(TB604B14E3B)	Ethernet2/4
nxos-spine1	Ethernet2/3	nxos-spine2(TB604B14E3B)	Ethernet2/3
nxos-spine1	Ethernet2/2	nxos-spine2(TB604B14E3B)	Ethernet2/2
nxos-spine1	Ethernet2/1	nxos-spine2(TB604B14E3B)	Ethernet2/1

Jinja2 Template

```
# neighbors.j2
| Source     | Interface    |         Neighbor           | Interface   |
| ---------- |--------------|----------------------------|-------------|
{% for local_int, details in ansible_net_neighbors.items() %}
{% for neigh_data in details %}
| {{ inventory_hostname }}| {{ local_int }}  |  {{ neigh_data.sysname }} | {{ neigh_data.port  }} |
{% endfor %}
{% endfor %}
```

Use Case #3 - Auto-configure Interface Descriptions based on LLDP data

Playbook (`auto-configure.yml`)

```
---

- name: AUTO-CONFIGURE PORT DESCRIPTIONS
  hosts: nxos
  connection: network_cli
  gather_facts: yes

  tasks:

    - name: AUTO-CONFIGURE PORT DESCRIPTIONS BASED ON LLDP
DATA
      nxos_config:
        src: "lldp_neighbors.j2"
```

Use Case #3 - Auto-configure Interface Descriptions based on LLDP data

Playbook (`auto-configure.yml`)

```
---

- name: AUTO-CONFIGURE PORT DESCRIPTIONS
  hosts: nxos
  connection: network_cli
  gather_facts: yes

  tasks:

    - name: AUTO-CONFIGURE PORT DESCRIPTIONS BASED ON LLDP
DATA
      nxos_config:
        src: "lldp_neighbors.j2"
```

Jinja2 Template (`lldp_neighbors.j2`)

```
{% for interface, neigh_data in ansible_net_neighbors.items() %}
{% if interface != 'mgmt0' %}
interface {{ interface }}
{% endif %}
{% for data in neigh_data %}
  description Connects to {{ data['port'] }} on {{ data['host'] }}
{% endfor %}
{% endfor %}
```

Use Case #3 - Auto-configure Interface Descriptions based on LLDP data

Playbook (`auto-configure.yml`)

```
---

- name: AUTO-CONFIGURE PORT DESCRIPTIONS
  hosts: nxos
  connection: network_cli
  gather_facts: yes

  tasks:

    - name: AUTO-CONFIGURE PORT DESCRIPTIONS BASED ON LLDP
DATA
      nxos_config:
        src: "lldp_neighbors.j2"
```

Jinja2 Template (`lldp_neighbors.j2`)

```
{% for interface, neigh_data in ansible_net_neighbors.items() %}
{% if interface != 'mgmt0' %}
interface {{ interface }}
{% endif %}
{% for data in neigh_data %}
  description Connects to {{ data['port'] }} on {{ data['host'] }}
{% endfor %}
{% endfor %}
```

Dry Run

```
ntc@ntc-training:auto-configure$ ansible-playbook -i inventory auto-configure.yml --check -v

PLAY [AUTO-CONFIGURE PORT DESCRIPTIONS]
****************************************************************************************

TASK [Gathering Facts] ***************************************************************
ok: [nxos-spine1]
ok: [nxos-spine2]

TASK [AUTO-CONFIGURE PORT DESCRIPTIONS BASED ON LLDP DATA]
****************************************************************************************
changed: [nxos-spine2] => {
    "changed": true,
    "commands": [
        "interface Ethernet1/1",
        "description Connects to Ethernet1/1 on nxos-spine1.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/5",
        "description Connects to Gi5 on csr2.ntc.com",
        "interface Ethernet1/2",
        "description Connects to Ethernet1/2 on nxos-spine1.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/3",
        "description Connects to Ethernet1/3 on nxos-spine1.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/4",
        "description Connects to Ethernet1/4 on nxos-spine1.ntc.com(9X487AEJUEE)"
    ],
   ..omitted
changed: [nxos-spine1] => {
    "changed": true,
    "commands": [
        "interface Ethernet1/1",
        "description Connects to Ethernet1/1 on nxos-spine2.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/5",
        "description Connects to Gi5 on csr1.ntc.com",
        "interface Ethernet1/2",
        "description Connects to Ethernet1/2 on nxos-spine2.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/3",
        "description Connects to Ethernet1/3 on nxos-spine2.ntc.com(9X487AEJUEE)",
        "interface Ethernet1/4",
        "description Connects to Ethernet1/4 on nxos-spine2.ntc.com(9X487AEJUEE)"
    ],
   ..omitted
```

Use Case #4 - Configuration Backups

- Multi-Vendor Inventory
  - Juniper
  - Cisco
  - Arista

```
[iosxe]
csr[1:3]

[vmx]
vmx[1:3]

[eos]
eos-spine[1:2]
eos-leaf[1:2]

[nxos]
nxos-spine[1:2]

[all:vars]
ansible_user=admin
ansible_ssh_pass=admin123

[iosxe:vars]
ansible_network_os=ios

[vmx:vars]
ansible_network_os=junos

[eos:vars]
ansible_network_os=eos

[nxos:vars]
ansible_network_os=nxos
```

Use Case #4 - Configuration Backups

Backup Device Configurations Playbook

```
---

- name: BUILD BACKUP
  hosts: all
  connection: network_cli
  gather_facts: no

  tasks:

    - name: BACKUP DEVICE CONFIGURATIONS
      cli_config:
        backup: yes
```

Result after executing the playbook

```
ntc@ntc-training:build_backups$ tree backup/
backup/
├── csr1_config.2020-09-10@18:40:37
├── csr2_config.2020-09-10@18:40:37
├── csr3_config.2020-09-10@18:40:37
├── eos-leaf1_config.2020-09-10@18:40:40
├── eos-leaf2_config.2020-09-10@18:40:40
├── eos-spine1_config.2020-09-10@18:40:40
├── eos-spine2_config.2020-09-10@18:40:40
├── nxos-spine1_config.2020-09-10@18:40:43
├── nxos-spine2_config.2020-09-10@18:40:44
├── vmx1_config.2020-09-10@18:40:37
├── vmx2_config.2020-09-10@18:40:37
└── vmx3_config.2020-09-10@18:40:40

0 directories, 12 files
```

QUICK DEMO