



Introduction to Ansible

for Network Engineers

Jason Belk

Senior Technical Advocate

What is Ansible?

- Free and Open source
 - Owned by Red Hat / IBM
 - Been around for 10+ years
 - Initially focused on compute
- Runbook style automation framework / engine
- Simple to use, easy to set up and broad industry adoption



Why Should You Care?

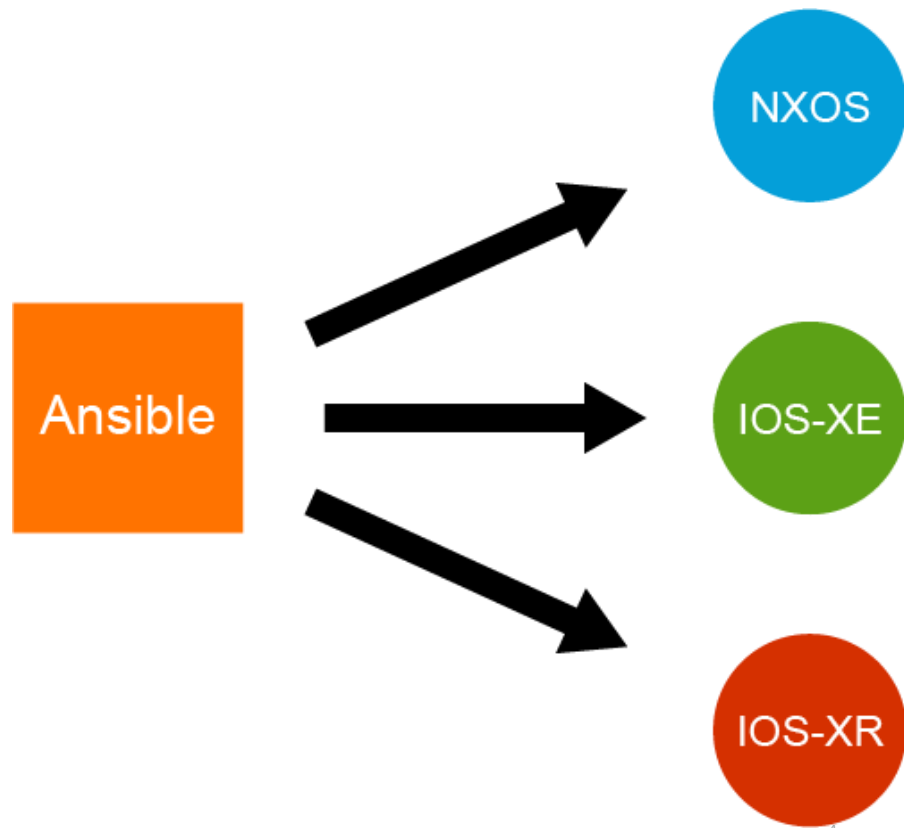


- Great for getting started in automation
- Built-in scaling for lots of devices
- Helpful for reusing within teams
- Not only talks to network devices, but compute, storage, ITSM, Cloud, etc.
- You can take existing cut sheets or processes and adapt to an Ansible workflow
- Powerful templating abilities

Ansible Overview (Cont.)

How Ansible Works

- Control Host
- Connects to device
 - SSH or APIs
- Execute Python code (modules)
 - On the control host
- Returns JSON object per task

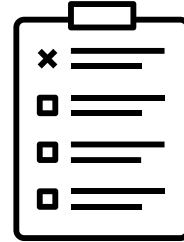
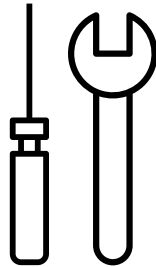


Basic Building Blocks of Ansible



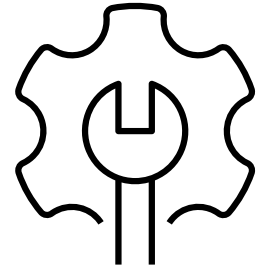
Inventory

Modules



Playbook

Application
Config



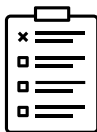


Ansible Inventory

- The device list with details like:
 - Device type (IOS, Nexus, etc.)
 - IP address / FQDN
 - **What you want to refer to the device in the Ansible Playbook**
 - Connection type
 - Username / Password (for learning in plaintext, use Vault for prod)
 - Device specific variables
- Device groups and group variables
- Can be either INI or YAML format

```
ansible > ❏ inventory.ini
1  [iosxe]
2  csr1 ansible_host=10.10.20.48 ansible_network_os=ios
3
4  [all:vars]
5  ansible_user=developer
6  ansible_ssh_pass=Cisco12345
7  ansible_connection=network_cli
```

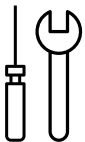
```
inv.yml > ...
1  ---
2  all:
3    children:
4      routers:
5        hosts:
6          sandbox-iosxe-recomm-1.cisco.com:
```



Ansible Playbook

```
ansible > ! pb-configure-snmp.yaml > ...
1  ---
2  - name: PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS
3    hosts: "csr1"
4    connection: network_cli
5    gather_facts: no
6    tasks:
7      - name: "TASK 1 in PLAY 1 - CONFIGURE SNMP LINES"
8        cisco.ios.ios_config:
9          lines:
10            - snmp-server community belk-demo R0
11              snmp-server location VEGAS
12              snmp-server contact JASON_BELK
13      - name: "TASK 2 in PLAY 1 - VERIFY SNMP LINES PRESENT"
14        cisco.ios.ios_command:
15          commands:
16            - "show run | include snmp-server"
```

- YAML file
 - Using indentation to indicate logical nesting / hierarchy
- A list of tasks to execute upon a device or group of devices from the inventory
- Sequentially executes each task in threads for each device
- Execution is defined by the inputs into the modules, one module per task

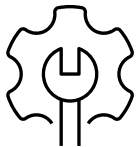


Ansible Modules

- Over 750 modules available
- Purpose built for IT uses like:
 - Configuring network devices of all major vendors
 - Gathering network state through show commands
 - Updating case notes for ServiceNow
 - Sending messages to a chat client like Slack or WebEx Teams
 - Building configuration templates with Jinja2
- Modules have structured required and optional inputs, with predefined output structures documented on Ansible Docs

```
cisco.ios.ios_config:
  lines:
    - snmp-server community belk-demo RO
    - snmp-server location VEGAS
    - snmp-server contact JASON_BELK
```

```
cisco.ios.ios_command:
  commands:
    - "show run | include snmp-server"
```

Ansible Application Config

```
etc > ansible > ⚙️ ansible.cfg
1  [defaults]
2  deprecation_warnings = False
3  gather_facts = False
4  host_key_checking = False
5  gathering = explicit
```

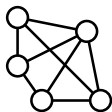
- Ansible looks for “ansible.cfg” either in current working directory or in /etc/ansible
- The application settings and turning Ansible features on or off
 - Things like checking for SSH host keys or not, any custom paths for modules, timeout values, etc.
- Does not change very often, usually set up once and forget about it

</> Ansible Templates & Variables

- Ansible uses Jinja2 to define config templates and YAML to define input variables for them
- Jinja2 includes many useful features like looping, conditionals and utility functions to transform or validate data
- Ansible loads in the variables per device and feeds them into the templates for you to then send the config to the device
- If the configuration is already present, Ansible checks the running-config first, so it doesn't send it again

```
templates > 📄 interface-template.j2
1  {% for iname, idata in interfaces.items() %}
2  interface {{ iname }}
3      description {{ idata.description }}
4      ip address {{ idata.ipv4addr }} {{idata.subnet}}
5  {% endfor %}
```

```
host_vars > ! sandbox-iosxe-recomm-1.cisco.com.yml > {} interfaces
Ansible Vars File (ansible-vars.json)
1  interfaces:
2      Loopback100:
3          description: "This is a loopback configured by ansible"
4          ipv4addr: "10.123.123.100"
5          subnet: "255.255.255.255"
6
7      Loopback101:
8          description: "This is a loopback configured by ansible"
9          ipv4addr: "10.123.123.101"
10         subnet: "255.255.255.255"
11
12     Loopback102:
13         description: "This is a loopback configured by ansible"
14         ipv4addr: "10.123.123.102"
15         subnet: "255.255.255.255"
16
17     Loopback103:
18         description: "This is a loopback configured by ansible"
19         ipv4addr: "10.123.123.103"
20         subnet: "255.255.255.255"
21
22     Loopback104:
23         description: "This is a loopback configured by ansible"
24         ipv4addr: "10.123.123.104"
25         subnet: "255.255.255.255"
```



Variables: Ansible Facts

```
ansible_net_iostype: IOS-XE
ansible_net_memfree_mb: 2060024.53125
ansible_net_memtotal_mb: 2392443.9765625
ansible_net_model: CSR1000V
ansible_net_neighbors: {}
ansible_net_python_version: 3.6.8
ansible_net_serialnum: 926V75BDNRJ
ansible_net_system: ios
ansible_net_version: 16.09.03
ansible_network_resources: {}
```

```
ansible_net_filesystems_info:
  'bootflash:':
    spacefree_kb: 5869820.0
    spacetotal_kb: 7712692.0
ansible_net_gather_network_resources: []
ansible_net_gather_subset:
  - interfaces
  - default
  - hardware
ansible_net_hostname: CSR-1000V
ansible_net_image: bootflash:packages.conf
ansible_net_interfaces:
  GigabitEthernet1:
    bandwidth: 1000000
    description: MANAGEMENT INTERFACE - DON'T TOUCH ME
    duplex: Full
    ipv4:
      - address: 10.10.20.48
        subnet: '24'
    lineprotocol: up
```

- When Ansible connects to a device, it gathers and parses a variety of useful facts such as:
 - OS Version
 - Hostname
 - Interface status, IP address, description, etc.
- Ansible facts can be stored to disk for later reference or used during a playbook as runtime variables to feed into conditional logic
 - For example: only configure devices with “SJC” in the hostname

