Managing 15,000 network devices with Ansible

Landon Holley & James Mighion
May 8, 2018
What is it
Combining the foundation of Ansible Engine with the enterprise abilities of Ansible Tower to automate physical networking devices.

INFRASTRUCTURE AS YAML
- Automate backup & restores
- Manage “golden” versions of configurations

CONFIGURATION MANAGEMENT
- Changes can be incremental or wholesale
- Make it part of the process: agile, waterfall, etc.

ENSURE AN ONGOING STEADY STATE
- Schedule tasks daily, weekly, or monthly
- Perform regular state checking and validation
Ansible for Network Engineers?

Networks will still exist, and the world will still need people who know physical networks!

Ansible makes network management easier but it’s a framework for building your automation.

Remember when we said Ansible was easy to learn? It’s as easy as you need it to be!

It needs to be built by the people who know it best.

YAML, Jinja2, and Python...oh my!
Yes!

Here’s a Playbook to login and do `show run`:

```yaml
---
- hosts: all
  connection: network_cli
  remote_user: admin

  tasks:
  - name: show run
    ios_command:
      commands:
        - show running-config
```

Yes (Again)!

Here’s a Playbook to perform a backup:

```yaml
---
- hosts: rtrl
  connection: network_cli
  remote_user: admin

  tasks:
  - name: Backup Configuration
    ios_config:
      backup: yes
```
And it’s getting even easier!

PROBLEM: Everyone is writing the same playbooks in a vacuum, per platform

SOLUTION: Ansible Roles

- Opinionated, task-focused solutions
- Developed, tested, distributed, and supported *
- Integration with DCI and Agile development models

create_vlan

*In plan for future release
How Does it All Work?

Network Connection Plug-ins
(NETCONF/SSH, CLI/SSH, API/SSH)

Ansible Network Roles*

Ansible Network Platform Modules

Network Visualization*

Job Templates
Workflows
Role-based Access
Job Scheduling
Enhanced Logging

API and GUI-based for large teams of network operators

CLI-based for individuals, developers, and small teams

*In plan for future release
Our Project
Our Goals

1) Automate manageability use cases for multiple vendors with a wide range of versions:
   - Cisco (Switching, Routing, Wireless)
     - IOS
     - IOS XR
     - IOS XE
     - NX-OS
     - AireOS
   - Arista EOS (Switching, Routing)
   - Aruba (Wireless)
   - F5 BIG-IP (Load Balancing)
   - Fortinet FortiManager (Firewall)

2) Configuration management that map to specific tasks for network operations:
   1. Device facts and configs
   2. SNMP polls/traps
   3. NTP
   4. Local passwords
   5. Syslog
   6. AAA
   7. ACLs
   8. Interfaces
   9. Address / Address Groups
Approach

Repo breakdown

Main repo
├── action_plugins
├── filter_plugins
├── group_vars
├── inventory
├── library
├── lookup_plugins
├── module_utils
├── parsers
├── roles
├── simple_tasks
├── terminal_plugins
├── top_level_playbooks.yml

Some of the roles
├── adhoc
├── config_aaa
├── config_acl
├── config_localpw
├── config_ntp
├── config_snmp
├── config_syslog
├── deploy_psk
├── get_wireless_baseline
├── network-cli
├── network-engine
├── network_facts
Approach
Role breakdown

roles/config_snmp/
| ├── defaults
| | └── main.yml
| ├── files
| | ├── f5_snmp_communities_parser.yml
| | └── f5_snmp_traps_parser.yml
| ├── handlers
| | └── main.yml
| ├── meta
| | └── main.yml

├── tasks
| ├── arista-os.yml
| └── aruba-mobility-controller.yml
| └── cisco-ios-xr.yml
| └── cisco-ios.yml
| └── cisco-nxos.yml
| └── ciscowlan.yml
| └── f5-os.yml
| └── linux.yml
| └── loglogic.yml
| └── main.yml
| └── main.yml

├── vars
| └── main.yml
Example

tasks/main.yml

- name: include device specific tasks
  include_tasks: "{{ device_os }}.yml"
Example

Continued

tasks/cisco-ios.yml

```yaml
# Add a line if the host is a 6500
- name: Add config line for 6500's
  set_fact:
    snmp_lines: "{{ snmp_lines }} + [ 'snmp-server ifindex persist' ]"
  when: model_number[0:2] | version_compare('65', 'eq')

- name: Apply snmp-server config lines
  ios_config:
    provider: "{{ cli }}"
    running_config: "{{ config }}"
    lines: "{{ snmp_lines }}"
    parents: "{{ snmp_parents | default }}"
    save: yes
  register: snmp_lines_applied
```

Continued
In scaling Ansible to manage any amount of network devices, these are the key factors that affect job performance:

1. Config size -- raw text output from `show run` for each device
2. Device performance -- how long it takes to login, send commands, and get output
3. Inventory sizes and devices families, e.g., IOS, NX, XR, EOS, etc...
4. Frequency and extent of scheduling device changes
5. Use or availability of Ansible network facts
Ansible at Scale, pt. 2

Sizing inventories and jobs

1. Linear gain when adding CPUs (everything runs locally)

2. Bigger isn’t always better:
   a. More small Tower hosts
   b. Create small inventories and use job limits
   c. Use lots of small jobs

3. Use facts and fact caching
## Results

**Single job: 500 hosts, 100 forks**

<table>
<thead>
<tr>
<th>Fact Collection (no changes):</th>
<th>Local Passwords:</th>
<th>SNMP Community Strings:</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOS 4:08</td>
<td>IOS 5:25</td>
<td>IOS 8:34</td>
</tr>
<tr>
<td>XR 4:25</td>
<td>XR 6:23</td>
<td>XR 10:12</td>
</tr>
<tr>
<td>NX 15:35</td>
<td>NX 19:44</td>
<td>NX 25:51</td>
</tr>
<tr>
<td>EOS 8:09</td>
<td>EOS 12:01</td>
<td>EOS 18:01</td>
</tr>
</tbody>
</table>

**All:** 2:03:15

**All:** 2:45:12

**All:** 3:34:32
New Development
The Open Source Way

All development has been contributed back to the community

- Aruba and AireOS
  - Command and config modules
  - Terminal and action plugins
- New save option
- CLI transport for F5’s bigip_command
- Minor fixes
  - Connection setup
  - Documentation
  - Multiple changes in ansible-network repos
Challenges and Lessons Learned

Challenges
- Limited hardware
- Variability of device versions
- Training and focus
- Scaling Ansible/Tower
- Snowflake devices
- Defining source of truth

Lessons Learned
- Effectively scaling Ansible/Tower
- Writing efficient roles and playbooks
- Implementing creative device logic
- Use facts and caching
Learning/Training
Where to get started with Ansible Networking

Overview
ansible.com/overview/networking

Ansible Docs - Networking
docs.ansible.com/ansible/latest/network/index.html

Ansible Linklight
github.com/network-automation/linklight

IRC freenode #ansible-network
Don’t miss these network automation and management sessions coming up this week

**MAY 8**
10:30-11:15AM
Managing 15,000 Network Devices with Ansible (Room 2001)

11:45-12:30PM
Hybrid Cloud Network Interconnect with Ansible (Room 2014)

4:30-5:15PM
How Walmart Uses Systems Management Tools to Manage Its Massive IT Operation at Scale (Room 2004)

**MAY 9**
10:30AM-11:45AM
How are customers automating F5 BIG-IP with Ansible Tower? (Partner Theater, Expo Hall)

11:45AM-12:30PM
Red Hat Management Roadmap and Strategy (Room 2015)

4:30PM-4:50PM
Top 3 F5 BIG-IP and Ansible Use Cases (Room 2010)

**MAY 10**
2:00-2:45PM
Network Automation with Ansible (Room 2102)
AUTOMATION & MANAGEMENT

Come see us in the Red Hat booth in the Ecosystem Expo.

‘Management - What’s New’
New products under development

#redhat #rhsummit
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHat