DevOpsDays Raleigh Ignites

@DevOpsDaysRDU | #DevOpsDaysRDU

What is Ignite?

- 20 slides
- 5 minutes
- Slides advance every 15 seconds

Ignite

Network Stability and Security with Cisco Equipment - Jessica Repka

Ansible and Docker

Providing security and stability for Cisco Equipment

Each Network has its problems

Every problem is an opportunity in disguise

- Network team devoting ever-increasing time to “hand-crafted” machine configs…
- …meaning, larger projects (network architecture overhaul) go nowhere.
- Technical debt “bankrupting” us; expected resolution >1 year.
- Security “best practices” painful to implement, when staff changed.

Why Ansible and Docker?

- Our networking team are CLI-devotees.
- Simple and powerful syntaxes lower barriers to entry.
- Inventory scripts offer faster, easier-to-manage deployments.
- Scripting is a part of the philosophy, and familiar to the team.

Docker for Switch Software Distribution

- Cisco code upgrades distributable via scp, tftp, or HTTP; HTTP “lowest friction.”
- Docker allows rapid HTTP server spin-up, that we use to serve static code images
- Jenkins CI keeps the standardized Docker image up-to-date and tested…
- …with emergency redeploy/rollback in case of outage, accident, etc.

Ansible Playbooks: Upgrade.yaml

- Mass deployment made easy
- Customizable, site by site and by hardware type
- Upgrades sourced via HTTPd in Docker
- “Look, ma! No hands!”

Ansible Playbooks: Pass_Change.yaml

- Bulk password change
- Split-able by device type; split per-site if necessary
- OBVIOUS use cases: breaches and “good hygiene”

Inventory Scripts

Ansible 2.5 adds a key new feature: inventory scripts.

These permit generating inventory lists from known *sources of truth.*

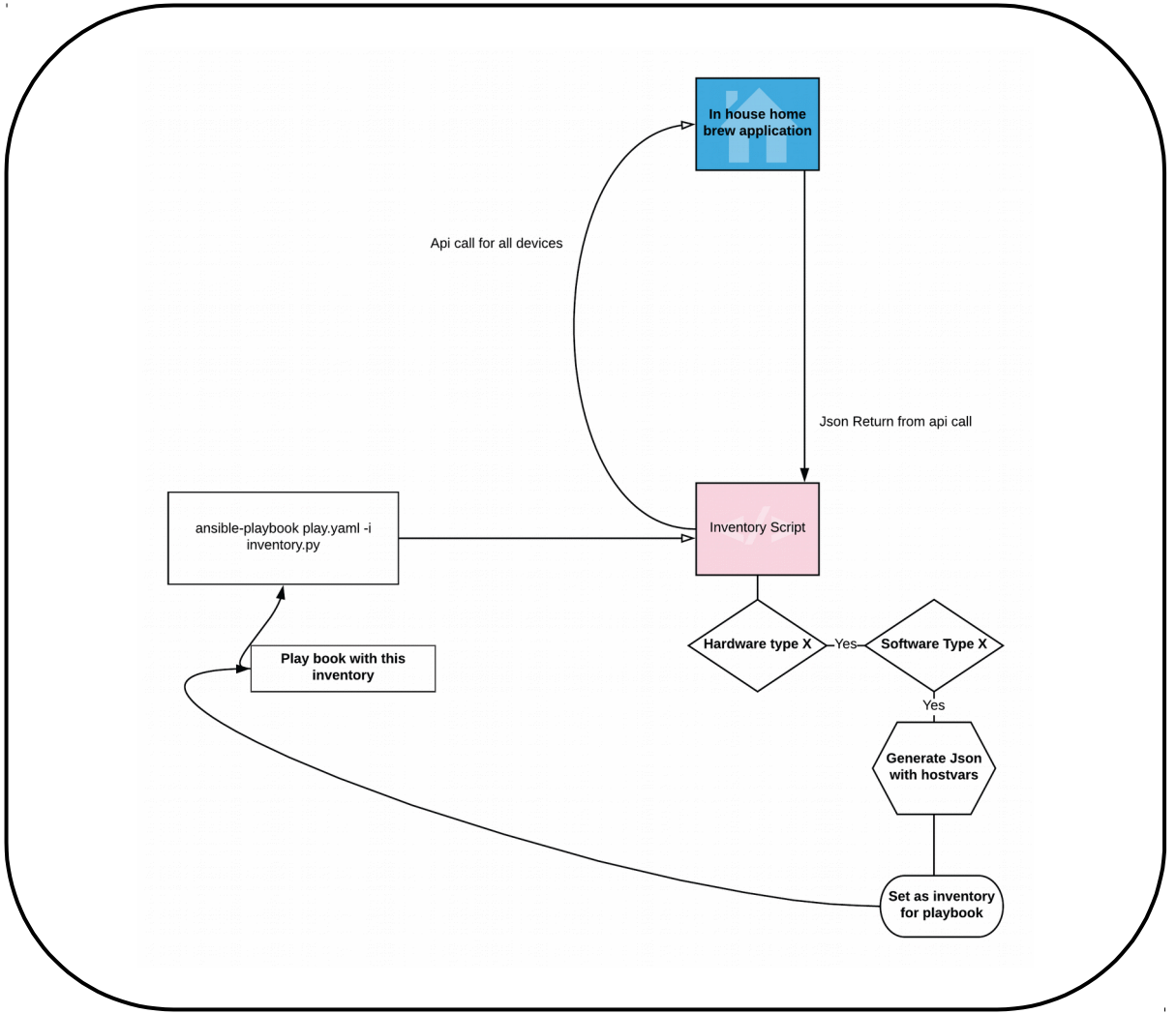
Our ”single source of truth”?

“Cartographer” – a Ruby on Rails application that warehouses over 90%** of the information relating to our network equipment; I query it via REST API.

**The fraction of our equipment not represented in Cartographer is steadily decreasing – but there’s one large category that I have to manually append in my inventory script.

Inventory Script Example Work Flow

- API call from main inventory application
- Parse fields to determine correct device type
- Generate hostvars and inventory
- Feed into ansible for playbook execution



Ansible Inventory File

WARNING: Examples ahead!

```
jr358@Artemis: ~/kirk/upgrade_cisco/s
- name: Upgrade a Cisco IOS switches
  gather_facts: false
  hosts: all
  connection: network_cli

  tasks:

     - name: Check if bin or full install
       ios_command:
         commands:
           - show boot | i variable
       tags: show
       register: pack
       when: inventory_hostname in groups['IOS']

     - name: Clean up, Copy install, Reboot only to fully install 3850's
       ios_command:
         commands:
           - software clean force
           - software install file http://ipaddr/images/image.bin
       when: inventory_hostname in groups['IOS'] and "flash:packages.conf" in  pack.stdout[0]

     - name: Clean up, Expand install, Reboot only to fully install 3850's. Expand Required
       ios_command:
         commands:
           - 'software clean file flash: force'
           - 'software expand file http://ipaddr/images/image.bin.bin to flash: '
       when: inventory_hostname in groups['IOS'] and "flash:packages.conf" not in pack.stdout[0]

     - name: Set boot configs for expantion
       ios_config:
         lines:
           - no boot system
           - boot system flash:packages.conf
           - do write memory
           - do show boot
       when: inventory_hostname in groups['IOS'] and "flash:packages.conf" not in pack.stdout[0]
~
```

Upgrade.yaml

- Upgrades are pulled from the HTTP server for installation
- First, we check the boot variable to determine what type of install we're performing
- Every upgrade requires a clean, before installation proceeds
- If the package file includes a .conf file, we do a full install
- Otherwise, a bin-expand install is required, which requires extra commands for the next reboot

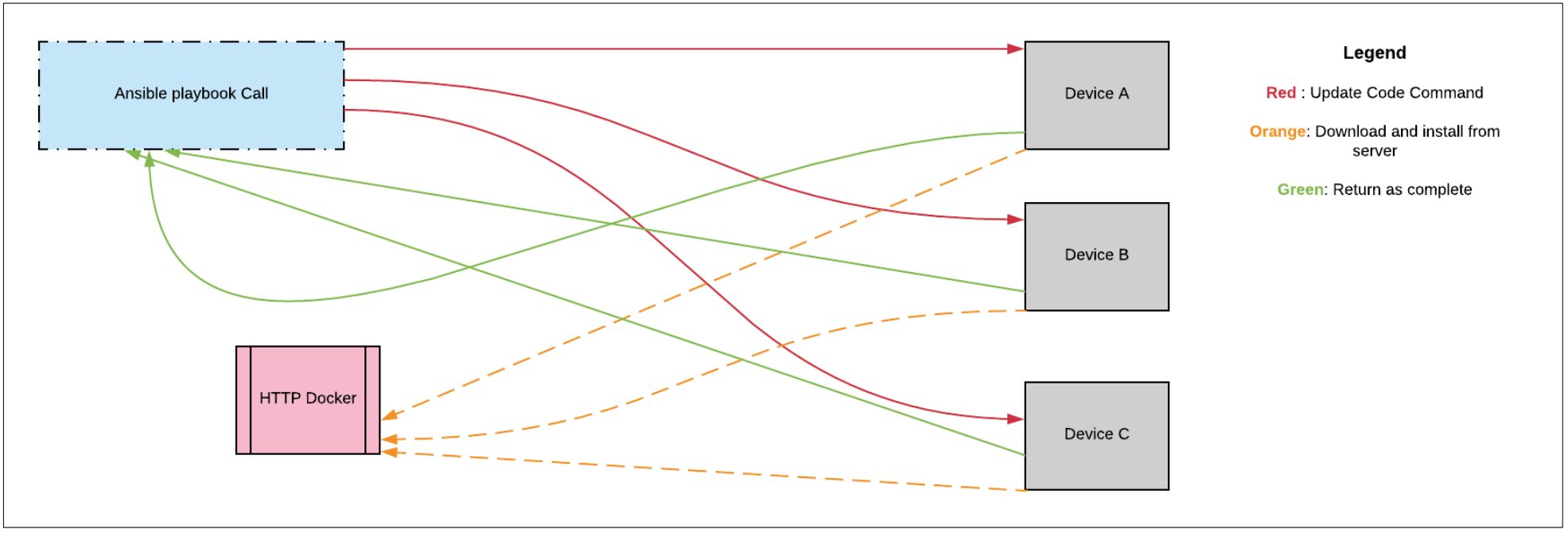
Upgrade.yaml:

> Example inventory configuration

```
[all:vars]
connection=network_cli
ansible_user=current_user
ansible_ssh_pass=password
ansible_become=yes
ansible_become_method=enable
ansible_network_os=ios
ansible_persistent_command_timeout=5000


[IOS]
10.0.0.0
```

Example inventory file for Upgrade.yaml

“network_cli” was the better choice for connection type, for us. This new addition in Ansible 2.5 makes it easier to set the Cisco software type and automatically connect.



Upgrade.yaml Work Flow

Pass_change.yaml

- Password changes are made per-device
- Each device group can have child groups
- Each device type is broken into inventory group with group vars
- All vars include the new password for change

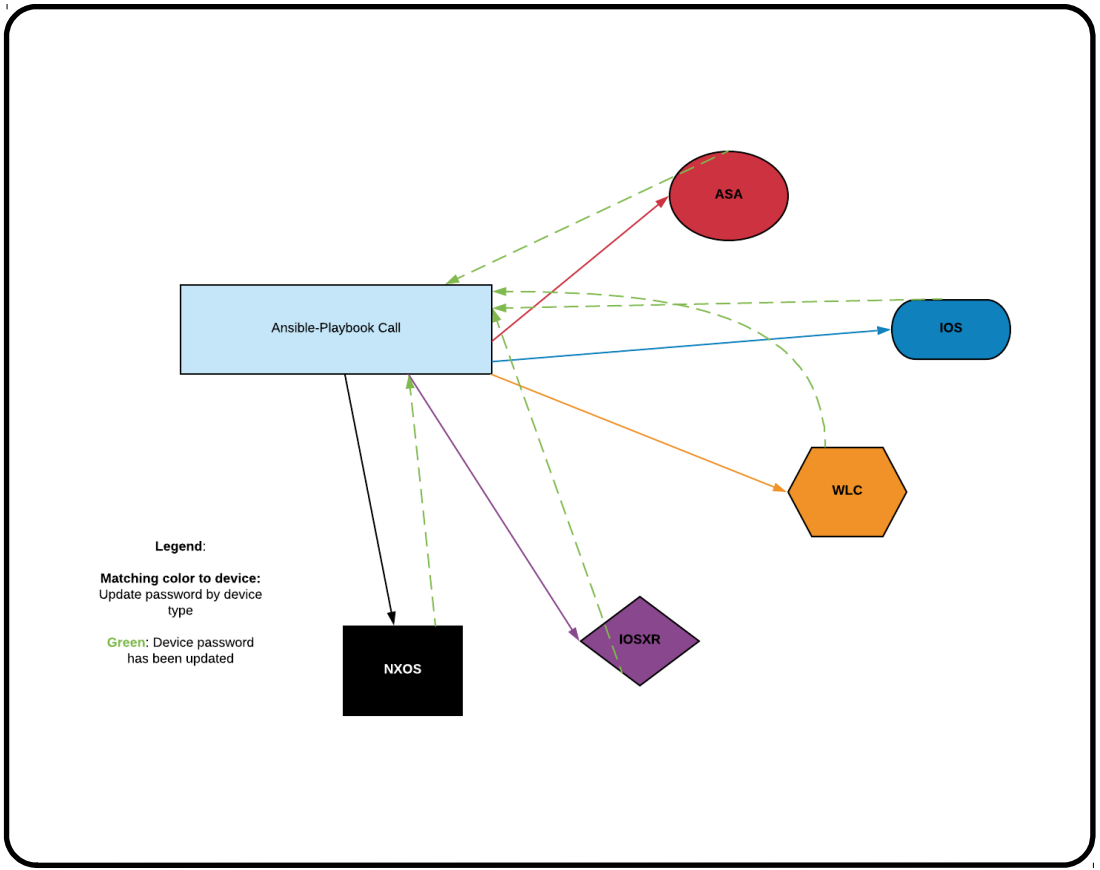
```
- name: Network Password Change
  hosts: devices
  gather_facts: no

  tasks:

    - name: IOS User Update
      ios_user:
        name: "{{ ansible_user }}"
        configure_password: "{{ new_pass }}"
        update_password: always
        state: present
      when: inventory_hostname in groups['ios']

    - name: IOS-XR User Update
      iosxr_user:
        name: "{{ ansible_user }}"
        configure_password: "{{ new_pass }}"
        update_password: always
        state: present
      when: inventory_hostname in groups['iosxr']

    - name: ASA User Update
      asa_config:
        lines:
          - enable password "{{ new_pass }}"
          - username admin password "{{ new_pass }}" privilege 15
        provider:
          username: "{{ user }}"
          password: "{{ pass }}"
          authorize: yes
          auth_pass: "{{ pass }}"
      when: inventory_hostname in groups['asa']

    - name: Nexus User Update
      nxos_user:
         name: "{{ ansible_user }}"
         configured_password: "{{ new_pass }}"
         update_password: always
         state: present
      when: inventory_hostname in groups['nxos']


    - name: Controller User Update
      aireos_config:
        lines:
          - netuser password "{{ user }}" "{{ new_pass }}"
          - ap mgmuser add "{{ user }}" "{{ new_pass }}"
      provider:
        username: "{{ user }}"
        password: "{{ pass }}"
      when: inventory_hostname in groups['wlc']
```

Example Inventory File for Pass_change.yaml

- We label child groups to process in a list named “devices” ; each represents a Cisco code/device type.
- Each group has its own connection vars
- All vars are set to the new password all devices will use

```
[all:vars]
new_pass = newpasswordhere

[devices:children]
ios
iosxr
nxos
asa
wlc

[iosxr]
x.x.x.x
x.x.x.x

[iosxr:vars]
connection=network_cli
ansible_user=current_user
ansible_ssh_pass=password
ansible_become=yes
ansible_become_method=enable
ansible_network_os=iosxr
ansible_persistent_command_timeout=5000

[ios]
x.x.x.x
x.x.x.x

[ios:vars]
connection=network_cli
ansible_user=current_user
ansible_ssh_pass=password
ansible_become=yes
ansible_become_method=enable
ansible_network_os=ios
ansible_persistent_command_timeout=5000

[nxos]
x.x.x.x
x.x.x.x

[nxos:vars]
connection=network_cli
ansible_user=current_user
ansible_ssh_pass=password
ansible_become=yes
ansible_become_method=enable
ansible_network_os=nxos
ansible_persistent_command_timeout=5000

[asa]
x.x.x.x
x.x.x.x

[asa:vars]
connection=local
user=username
pass=password


[wlc]
x.x.x.x
x.x.x.x

[wlc:vars]
ansible_connection=local
user=username
pass=password
```

Example Pass_change.yaml Work Flow



Points of note, regarding Connection types

The “network_cli” module misses major classes of devices.

We have to hack around that, using the old-style config.

Points of note, regarding the ASA modules

ASA module is key. It is also community owned.

This is good ... and bad.

Optional: Wrapper Script

We made a wrapper script with guided menu options, for ease of use. Each option gathers environment variables for passing to the inventory script.

Through these options, our team can pinpoint inventory in the play book, either by site or by hardware type. This provides a mechanism for limiting deployments to only specifically chosen equipment,

Jessica Repka

Duke University

Contact

Email:
Jessica.Repka@duke.edu

Twitter

Twitter:
@alynderthered1