Building Towers with Ansible

Alex Bron
Conclusion Xforce
Who am I?

- Alex Bron
- Consultant at Conclusion Xforce
- Working for Alphabet / BMW
  - Openstack Solution Building team
    - Mainly situated at Alphabet Breda
    - Team members in Regensburg and München
How does BMW use Openstack?

- Region 1+2: Multiproject in DE
- Region 3+4: Openshift on Openstack DE
- Region 5+6: Openshift on Openstack US
- Region 7+8: Openshift on Openstack CN
- Region 9: Multiproject in DE
- Region 10 (planned) in DE
- Region 11 (planned)
- Region 12 (planned)
- Region 13 (planned)

- Region 0 / 90 (Integration, “QA”)
- Region 97 / 98 (Lab)
Ansible for Openstack?

Ansible *Cook*-book

- Management playbook for all openstack environments
- Each region has its own set of playbooks (Openstack and Ceph plays)

Monitoring playbook

- Configuration of Prometheus + Alert manager
- Picks up new nodes when added to the inventory

Grand Unified Inventory

- Central Ansible inventory, inventory files per Openstack region
- Contains variables that can be globally used
  - Proxy settings
  - LDAP authentication settings
  - No vault-files, but vaulted strings
Why use Ansible towers?

- One tower for Integration
  - Scheduled to run the master branch of the cookbook
  - Current schedule: run each hour
  - Used as QA for new releases of the cookbook
- Towers in DE, US and CN
  - Scheduled to run tagged branches of the cookbook
  - Current schedule: run each hour

- All towers extensively use GIT as source for Inventories and Job templates
- All towers use LDAP authentication against AD

Currently we use AWX 3.0.1
Why building towers with Ansible?

It all started with a crash…

• Need to rebuild tower and tower-dev
• Expansion to US and CN:
  • Need to build tower-us
  • Need to build tower-cn
  • Need to add new regions to tower

• Need to build all towers in a consistent, uniform way
• Don’t want to click through a web user interface
How do we build the towers?

1. Play build_tower.yml
   • Setup persistent storage for the containers
   • Installs custom ansible.cfg
   • Adds additional ansible versions
   • Launches the containers for AWX

2. Play load_initial_data.yml
   • Configures the tower
   • Configures authentication
   • Load all data into tower
How do you define a Tower?

awx_towers:
- awx_name: tower
  awx_container: awx_tower
  awx_docker: <docker-host>
  awx_kind: production
  awx_port: 8989
  awx_ansible: "{{ awx_ansible_versions }}"
  awx_tower: "{{ awx_tower_version }}"
- awx_name: tower_test
  awx_container: awx_tower_test
  awx_docker: <docker-host>
  awx_kind: production
  awx_testmode: yes
  awx_port: 8989
  awx_ansible:
    - "2.4.6"
    - "2.7.11"
  awx_tower: "7.0.0"

Every tower is defined:
- Name of the awx container (when using upstream AWX)
- Name of the container host
- Whether the tower targets Production or Integration
- Which Ansible versions should be added
- Which AWX version is to be used
- Testmode flags a tower for testing purposes
Initial Data Loading - part 1

Configure Tower
- Apply all AWX config settings

Remove Demo
- Remove Demo Project
- Remove Demo Job Template
- Remove Demo Inventory
- Remove Demo Credentials

Configure LDAP Authentication
- Apply all settings for LDAP authentication

Load Organization Structure
- Define the Organizations
- Define the Teams

Load Credentials
- Define Machine accounts
- Define GIT accounts
- Define Vault accounts
Initial Data Loading - part 2

Load Projects
- Define all projects
- Set project permissions
- Trigger project update
- Define project update schedule

Load Inventories
- Define inventories
- Define inventory sources
- Trigger inventory update
- Set inventory permissions

Load Notifications
- Define Slack notifications
- Define Email notifications

Load Job Templates
- Define all job templates
- Set the Ansible `virtualenv`; set `extra_vars` where needed
- Associate notifications; set job template permissions
- Define job template schedule
How do you interact with the Tower?

- Ansible modules for Tower
- Call tower-cli
  - Use Ansible shell module
- Call Rest-API
  - Not used in the Ansible Tower Builder

• tower_credential – create, update, or destroy Ansible Tower credential
• tower_credential_type – Create, update, or destroy custom Ansible Tower credential type
• tower_group – create, update, or destroy Ansible Tower group
• tower_host – create, update, or destroy Ansible Tower host
• tower_inventory – create, update, or destroy Ansible Tower inventory
• tower_inventory_source – create, update, or destroy Ansible Tower inventory source
• tower_job_cancel – Cancel an Ansible Tower Job
• tower_job_launch – Launch an Ansible Job
• tower_job_list – List Ansible Tower jobs
• tower_job_template – create, update, or destroy Ansible Tower job template
• tower_job_wait – Wait for Ansible Tower job to finish
• tower_label – create, update, or destroy Ansible Tower label
• tower_notification – create, update, or destroy Ansible Tower notification
• tower_organization – create, update, or destroy Ansible Tower organizations
• tower_project – create, update, or destroy Ansible Tower projects
• tower_receive – Receive assets from Ansible Tower
• tower_role – create, update, or destroy Ansible Tower role
• tower_send – Send assets to Ansible Tower
• tower_settings – Modify Ansible Tower settings
• tower_team – create, update, or destroy Ansible Tower team
• tower_user – create, update, or destroy Ansible Tower user
• tower_workflow_launch – Run a workflow in Ansible Tower
• tower_workflow_template – create, update, or destroy Ansible Tower workflow template
Configuration of the Towers

Configuration variables:

- Several paths
- Persistent directories for the AWX containers
- Ansible versions to support
  - Default version for Ansible
  - Default version for AWX
- Global Tower configuration items
- Proxy settings (not shown here)

awx_docker_dir: "/path/to/docker"
awx_virtualenv_dir: "/path/to/virtualenv"
awx_virtualenv: "awx_docker"

awx_persistent_dirs:
  - "config/awx_task"
  - "config/awx_web"
  - "data/postgresql"
  - "data/web"

awx_ansible_versions:
  - "2.4.6"
  - "2.7.11"

awx_ansible_default_version: "2.7.11"
awx_tower_version: "3.0.1"

tower_config_settings:
  - name: TOWER_URL_BASE
    value: "https://{{ tower_url }}/"
  - name: REMOTE_HOST_HEADERS
    value:
      - REMOTE_ADDR
      - REMOTE_HOST
      - HTTP_X_FORWARDED_FOR
Example of Job Template definitions

awx_templates:
- name: "monitoring_server_all"
  description: "Ansible Monitoring playbook"
  inventory: "{{ awx_default_inventory }}"
  project: "Prometheus"
  playbook: "server_all.yml"
  extra_vars: "prometheus_region={{ geo_region }}"
  schedule: false
  ansible_version: "2.7.11"
  job_machine_cred: "machine_ssh"
  job_vault_cred: "my_vault"
- name: "monitoring_prometheus"
  description: "Ansible Monitoring setup"
  inventory: "{{ awx_default_inventory }}"
  project: "Prometheus"
  playbook: "prometheus.yml"
  extra_vars: "prometheus_region={{ geo_region }}"
  schedule: true
  schedule_time: "000000"
  schedule_freq: "MINUTELY"
  schedule_interval: "30"
  job_machine_cred: "machine_ssh"
  job_vault_cred: "my_vault"

awx_template_acl:
- name: "monitoring_server_all"
  team: "Monitoring Administrators"
  permission: "execute"
- name: "monitoring_prometheus"
  team: "Monitoring Administrators"
  permission: "execute"
Freshly installed AWX instance
AWX 7.0.0 with Ansible 2.8.4
Loading projects ...
Project details
Loading inventories ...
Inventory details
Loading templates ...
Template details
What did we achieve

• New Ansible plays can easily and consistently be pushed to all towers
• When a tower disappears (for whatever reason), it can be rebuild within 30 minutes
  • Initial launch of a tower takes 15 minutes
  • Ansible plays to build and load data take less time
• Upgrading towers can be done very easily
  • Remove existing tower
  • Build new tower with new version information
• New tags for the Ansible cookbook are easily pushed to all “Production” towers

Complete project is a Conclusion Xforce gitlab project:
https://gitlab.com/conclusionxforce/ansible-tower-builder.git