ANSIBLE TOWER OVERVIEW AND ROADMAP

Bill Nottingham
Senior Principal Product Manager
2017-05-03
WHY AUTOMATE?
WHY DO WE WANT AUTOMATION?

People make mistakes
People don't always have the best information
People have to drive with... other people
WHY DO WE WANT TO AUTOMATE?

Get your life back

Do more important things

Get where you're going faster
In the world of DevOps we know why we automate.

Automate the drudgery…

... so we get our lives back to do more strategic things

Share our automation and procedures…

... so everyone remains informed

Avoid manual mistakes…

... so our business can get where we're going faster
WHY ANSIBLE?
WHAT IS ANSIBLE?

It’s a simple automation language that can perfectly describe IT application environments and processes in Ansible Playbooks.

---

- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root
  tasks:
    - name: install httpd
      yum:
        pkg: httpd
        state: latest
    - name: write the apache config file
      template:
        src: /srv/httpd.j2
        dest: /etc/httpd.conf
    - name: start httpd
      service:
        name: httpd
        state: started
WHAT IS ANSIBLE?

It’s the open source **automation engine** that runs Ansible Playbooks.
**WHY ANSIBLE?**

**SIMPLE**
- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Usable by every team
- Get productive quickly

**POWERFUL**
- App deployment
- Configuration management
- Workflow orchestration
- Network automation
- Orchestrate the app lifecycle

**AGENTLESS**
- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- Get started immediately
- More efficient & more secure
WHAT IS ANSIBLE TOWER?

Ansible Tower is an enterprise framework for controlling, securing and managing your Ansible automation with a UI and RESTful API.
TOWER EMPOWERS TEAMS TO AUTOMATE

CONTROL
Scheduled and centralized jobs

KNOWLEDGE
Visibility and compliance

DELEGATION
Role-based access and self-service

SIMPLE
Everyone speaks the same language

POWERFUL
Designed for multi-tier deployments

AGENTLESS
Predictable, reliable, and secure

AT ANSIBLE’S CORE IS AN OPEN-SOURCE AUTOMATION ENGINE
ANSIBLE PLAYBOOKS

ANSIBLE CLI & CI SYSTEMS

ANSIBLE TOWER

ROLE-BASED ACCESS CONTROL

KNOWLEDGE & VISIBILITY

SCHEDULED & CENTRALIZED JOBS

SIMPLE USER INTERFACE

TOWER API

ANSIBLE

OPEN SOURCE MODULE LIBRARY

PLUGINS

PYTHON CODEBASE

TRANSPORT

SSH, WINRM, ETC.

AUTOMATE YOUR ENTERPRISE

INFRASTRUCTURE
LINUX, WINDOWS, UNIX ...

NETWORKS
ARISTA, CISCO, JUNIPER ...

CONTAINERS
DOCKER, LXC ...

CLOUD
AWS, GOOGLE CLOUD, AZURE ...

SERVICES
DATABASES, LOGGING, SOURCE CONTROL MANAGEMENT...

USE CASES

PROVISIONING

CONFIGURATION MANAGEMENT

APP DEPLOYMENT

CONTINUOUS DELIVERY

SECURITY & COMPLIANCE

ORCHESTRATION
ANSIBLE TOWER OVERVIEW
# WHY ANSIBLE TOWER?

<table>
<thead>
<tr>
<th>CONTROL</th>
<th>KNOWLEDGE</th>
<th>SCALE</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Avoid the wild west</em></td>
<td><em>Ensure environment consistency</em></td>
<td><em>Empower Teams to Automate</em></td>
</tr>
<tr>
<td>At-a-glance dashboard</td>
<td>Log all automation securely</td>
<td>Users from LDAP/SAML and more</td>
</tr>
<tr>
<td>Pull playbooks from SCM</td>
<td>Audit all Tower activity</td>
<td>Role-based access control</td>
</tr>
<tr>
<td>Schedule jobs at any time</td>
<td>Notifications inform users</td>
<td>Simple self-service surveys</td>
</tr>
<tr>
<td>Easily connect to inventory sources</td>
<td>Use new Enterprise Logging for analysis</td>
<td>Full REST API for all features</td>
</tr>
<tr>
<td>Model complex processes with new Workflows</td>
<td></td>
<td>Expand capacity with new Tower clusters</td>
</tr>
</tbody>
</table>
CONTROL WITH TOWER WORKFLOWS

Mix and re-use automation as needed without creating new playbooks

Provision ◦ Configure ◦ Deploy ◦ Scale
Build ◦ Test ◦ Promote ◦ Verify ◦ Deploy
KNOWLEDGE WITH TOWER LOGGING INTEGRATIONS

Integrate your automation with your systems analysis

![Logos of Ansible Tower, Elastic, Splunk, and Sumo Logic]
**SCALE** WITH TOWER CLUSTERS

Add capacity and redundancy with ease
ANSIBLE TOWER ROADMAP
BRING AUTOMATION TO YOUR WHOLE ENTERPRISE

DISCOVER
- Easily inventory whatever you have - systems, VMs, networking, and more

AUTOMATE
- Integrate into the tools and processes you already have
- Easily discover and remediate problems

SCALE
- Build out Tower environments for multiple deployment scenarios
- Build enterprise deployments using multiple clusters
Easily inventory whatever you have - systems, VMs, networking, and more
Integrate into the tools and processes you already have

AUTOMATE

RED HAT®
CLOUDFORMS

vmware

service now

splunk>
Easily discover and remediate problems

| ISSUE: Cluster integrity cannot be guaranteed because no fence devices available in HA cluster | AVAILABILITY |
| For an HA cluster without a fence device configured, data integrity cannot be guaranteed and the cluster configuration is functionally equivalent to not having any fence devices at all. |
| PLAN: Tronic Lab glibc plan (14065) |

| ISSUE: Data loss when target SCSI device responds illegal request | AVAILABILITY |
| SAN devices with connectivity issues can cause a target to return an illegal request error, making the SAN devices in Logical unit not supported in the /var/log/messages file. |
| PLAN: Tronic Lab glibc plan (14065) |

| ISSUE: Unexpected bonding behavior with incorrect syntax in bond configuration files | STABILITY |
| Bonding configuration has specific syntax that must be followed to prevent unexpected behavior such as bond failover. |
| PLAN: Payload Injection Fix (14405) |
Build Tower environments for multiple deployment scenarios - Introducing Tower Ramparts
Build enterprise deployments using multiple clusters

Use consistent API URLs to access the same job on multiple Tower clusters

Build automation definitions with Ansible playbooks and Workflow schemas

Containerized Tower for easier deployments

# Workflow schema
- job_template: Provision cloud instances
  success:
  - job_template: Apply standard configuration
    failure:
    - job_template: Destroy cloud instances
  - job_template: Deploy application
    failure:
    - job_template: Destroy cloud instances
THANK YOU

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

#redhat #rhsummit
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.