Integrating Red Hat Satellite and Ansible

Calvin Smith, Red Hat
Senior Solutions Architect
Core Infrastructure
What we’ll discuss today

- Introduction to Satellite
  - Satellite Brief Overview
  - What’s New in 6.7
- Ansible Integration in Satellite
- Satellite and Ansible Tower
  - Integrations
  - Use Case: Orchestrate Provisioning
  - Use Case: Automated Patching
Introduction to Satellite
Why Red Hat Satellite?

- Manage Red Hat® infrastructure
- Streamlined content management
- Developed to scale
- Simplified system integration
- Enhanced drift and configuration management
Content Management

- **Content Repository**: any type of content made available to any host.
- **Curation**: of content prior to distribution.
- **Distribution**: of content as close as possible to the end point.
Patch Management

**Report** on hosts that need updates, fixes, or enhancements

**Group** homogeneous systems so that you can easily work with them

**Respond** quickly to patching requirements using scalable automation
Provisioning Management

- **Provision** to bare metal, virtual, private, and public clouds
- **Import** non-provisioned hosts
- **Automate** using Ansible roles to perform post-provisioning steps
Subscription Management

- Centrally manage subscription usage
- Maintain accurate inventory and utilization information
- Report on subscription consumption
Additional Satellite Capabilities

- **Configuration Management** using Ansible
- **Automation** through integration with Ansible Tower
- **Compliance** using OpenSCAP policies
Standard Operating Environment hosts are the same across your environment.

Reliable and Resilient Using Red Hat Insights.

Secure your systems are patched, up to date, and compliant with security policies.

Confidence in your subscription utilization.
Red Hat Satellite Components

**Red Hat Satellite Server**
- Facilitates multi-tenant services
- Offers on-premise repository management
- Gives user and group role-based access control (RBAC)
- Delivers powerful user interfaces (GUI, API, and CLI)*
- Exports content to other Satellite servers

**Red Hat Capsule Server**
- Allows scaling of your Satellite environment
- Provides local content, provisioning, and integration services
- Discovers new physical and virtual machines

*Graphical user interface (GUI), application program interface (API), and command line interface (CLI)
Smart Management for Red Hat Enterprise Linux

Combine the powerful infrastructure capabilities of Red Hat Satellite with the simplicity of cloud management

Improve operational efficiency by 28%*
Overcome scale, skill, and security gaps

Patch  Provision  Report  Control

Identify & Remediate

ANSIBLE AUTOMATES  *Source: Satellite IDC Business Value Whitepaper
Red Hat Smart Management

Buy

Red Hat Smart Management

Get

Red Hat Satellite

Cloud Connector
Introduction to Satellite

Satellite 6.7 New Features
Satellite 6.7 Released!

In addition to performance and security enhancements

Provisioning enhancements

Integrations

Automation enhancements

Release Announcement
https://access.redhat.com/announcements/4982191
Provisioning Enhancements

**Azure Provisioning Support**
Support for provisioning hosts in Microsoft Azure

**Google Compute Engine enhancements**
Added CLI and API endpoints for Google Compute Engine
Red Hat Enterprise Linux Integrations

Web Console Integration
Ability to open the Web Console for a specific host inside Satellite

System Purpose Enhancements
Ability to use Activation Keys with system purpose

Module Stream Enhancements
Ability to filter on modules in Content Views
Ability to incrementally update a Content View
Introducing Cloud Connector

Smart Management subscription enables push-button remediation of issues identified by Insights

Cloud Connector
Connects your federated Satellite infrastructure to Insights to execute remediation playbooks at scale.

*Multiple Satellites are supported, but optional
Automation Enhancements
Continuing improvements with Red Hat Ansible® Automation Platform

Improved Performance for Dynamic Inventory
Better performance of inventory update in Ansible Tower

Ansible Enhancements
Inclusion of Ansible Runner to execute Ansible
Ansible Integration in Satellite
Ansible Integration in Satellite

Basic Ansible capabilities are part of Satellite

Satellite has integration with Ansible for the purposes of remote execution and desired state management.

**Remote Execution**
Run Ansible Playbooks inside of Satellite

**Use Roles for Configuration Management**
Deploy roles to hosts managed by Satellite
Build and run playbooks in Red Hat Satellite

1. User creates playbook through Satellite
2. Create remote execution job template for ansible playbook
3. Playbook is run by Satellite
4. Can also execute insights playbook
Import roles

Import and deploy ansible roles

- Import Ansibles roles into satellite
- Deploy roles to client servers for desired state configuration management
- RHEL system roles ship with RHEL to configure systems and can be imported
- Sample RHEL systems roles: network, kdump, storage, selinux
Satellite and Ansible
Tower together
Red Hat Management Portfolio

**Red Hat Ansible Automation Platform**
- Automation for Everyone
  - Simple, Powerful, Agentless
  - Cross Platform
  - Human Readable

**Red Hat Insights**
- Prevent Critical Issues Before They Occur
  - Continuous Insights
  - Verified Knowledge
  - Proactive Resolution

**Red Hat Smart Management**
- Flexibility to Manage Red Hat Enterprise Linux
  - On-Premise with Satellite
  - In the cloud with cloud management services for Red Hat Enterprise Linux
Satellite and Ansible Tower
Working together to manage your Red Hat environment

Satellite can ….
- Manage content repositories
- Manage content lifecycles
- Patch RHEL servers
- Provision RHEL servers physical, virtual or cloud

Tower can ….
- Orchestration across platforms
- Automate all the things
- Integrate multiple tools and workflows

Together Satellite and Tower can …
- Orchestrate provisioning
- Automate patching
- Full cross platform management
Satellite and Tower together

Integrations
By integrating Red Hat Satellite with Red Hat Ansible® Tower, administrators can now perform the following functions:

**Dynamic inventory**
Allows Ansible Tower to use Satellite as a dynamic inventory source

**Post-Provisioning callbacks**
Allows systems provisioned via Satellite to “callback” to Ansible Tower so that playbook runs can happen post-provisioning.
Dynamic Inventory
Using Satellite as a dynamic inventory source

Red Hat Satellite ➔ Inventory ➔ Red Hat Ansible Tower

Host A  Host B  Host C  Host D
Dynamic Inventory
Post-Provisioning Callbacks

Allows systems provisioned via satellite to “call back” to Tower

1. Satellite
2. Server provisioned by Satellite
3. New server
4. New server calls back to Tower
5. Tower executes playbook on new server
### Provisioning Templates

<table>
<thead>
<tr>
<th>Name</th>
<th>Host Group / Environment</th>
<th>Kind</th>
<th>Snippet</th>
<th>Locked</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kickstart default</td>
<td></td>
<td>Provisioning template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kickstart default IPXE</td>
<td></td>
<td>Provisioning template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kickstart default PXEGrub</td>
<td></td>
<td>Finish template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kickstart default PXEGrub2</td>
<td></td>
<td>Provisioning template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kickstart default PXELinux</td>
<td></td>
<td>PXELinux template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kickstart default user data</td>
<td></td>
<td>User data template</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kickstart_ifcfg_bonded_interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kickstart_ifcfg_bond_interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kickstart_ifcfg_generic_interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kickstart_ifcfg_get_identifier_names</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kickstart_networking_setup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PXELinux global default</td>
<td></td>
<td>PXELinux template</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**POST-PROVISIONING CALLBACK**
POST-PROVISIONING CALLBACK

ExecStart=/usr/bin/curl -k -s --data "host_config_key=<%=
host_param('ansible_host_config_key') -%>" https://<%=
host_param('ansible_tower_fqdn') -%>/api/v2/job_templates/<%=
host_param('ansible_job_template_id') -%>/callback/
Automating Satellite
Satellite can be automated through it’s API
foreman-ansible-modules
Ansible modules to automate your satellite*

- name: "Create a host"
  foreman_host:
    username: "admin"
    password: "changeme"
    server_url: "https://satellite.example.com"
    name: "new host"
    hostgroup: my_hostgroup
    state: present

Galaxy:
theforeman.foreman

Github:
https://github.com/theforeman/foreman-ansible-modules

* Not yet supported
Satellite and Tower together

Use case: Orchestrate Complex Provisioning
Orchestrated Provisioning

An example scenario

The administrator would like to provision a server sequentially. Once the server is provisioned:

- The current configuration is applied
- The server is run through a validation workflow
- The server is added to external tools, like monitoring or load balancers
Orchestrated Provisioning Workflow
Using Tower to integrate provisioning process in workflow

1 - Admin initiates provisioning job through satellite
2 - Satellite provisions server
3 - New Host uses post-provisioning callback to notify tower
4 - Tower completes configuration on new host
5 - Tower initiates any additional workflow in order to add host to external tools

ANSIBLE AUTOMATES
Satellite and Tower together

Use case: Automated Patching
Automated Patching
An example scenario

The administrator would like to automate patching a series of servers, in a specific order. Before moving to the next server:

- All applicable patches are applied
- The patched host is rebooted
- The application is verified working

Once the job is completed, notification is sent to the administrator
Automated Patching
Using Tower to automate patches through your environment

1. At scheduled time, the patch process is initiated by Tower.
2. Tower job starts that patches hosts sequentially.
3. Satellite provides content specific to each host.
4. Tower emails a report to the admin once completed.
Satellite resources

- Red Hat Satellite blog
- Red Hat Satellite product page
- Red Hat Satellite customer portal
- Red Hat Satellite documentation
- Red Hat Consulting offering: Transition to Red Hat Satellite 6

Satellite training and videos

NEW COURSE
- RH053: Satellite Technical Overview (also available on Udemy)
- RH403: Red Hat Satellite 6 Administration
ANSIBLE AUTOMATES

THANK YOU