Automation and configuration management across hybrid clouds with CloudForms, Satellite 6, Ansible Tower

Laurent Domb
Sr. Cloud Specialist Solutions Architect

Michael Dahlgren
Cloud Specialist Solutions Architect

June 2016
About Us

Laurent Domb
Sr. Cloud Specialist Solutions Architect
RHCA IV, PCP, ITILv2, eMBA
Red Hat

Michael Dahlgren
Cloud Specialist Solutions Architect
RHCE, MBA
Red Hat
OVERVIEW

● Why use configuration management in hybrid cloud environments
● Different configuration management solutions
  ○ Puppet
  ○ Ansible
● How are they integrated
  ○ CloudForms + Satellite 6 + Ansible Tower
THE JOURNEY TO CONFIGURATION MANAGEMENT AND AUTOMATION ACROSS HYBRID CLOUDS
The Path To Configuration Management / DevOps

Scripts

Documentation

Chaos

TO BE MORE AGILE

#!/bin/sh
Traditional Responsibilities Between Dev and Ops

- Operations responsibilities:
  - Work on multiple projects at the same time:
    - Business Projects
    - Internal Projects
    - Planned Changes
    - Unplanned Changes

- While working on multiple projects uptime needs to be 99.9%
Traditional Responsibilities Between Dev and Ops

- **Developer responsibilities:**
  - Work on one project:
    - Write code for new products
    - New Features
    - Security updates
    - Bugfixes

- Once the code is ready it gets passed on to operations which now needs to deploy and run the code
Traditional Infrastructure / Development Pains

- Communication between teams is non existent as they are in different silos
- Bureaucracy / Approvals
- Slow development lifecycle
- Code is not in version control, collaboration therefore difficult and rollbacks almost impossible
- Applications / services are monolithic and are not api first
Transform Your Organization

OPENNESS + MERGE + Processes
Architect The Enterprise For The Future

- Change is the new normal
- Understand the business strategy and define an IT road map which supports that strategy
- Create a short term 1-2 years strategy (tech is changing fast)
- Keep the agility to change your plan based on technology changes and observe how you and your team operate with it
CONFIGURATION MANAGEMENT
The Configuration Management Tool Of Choice

- DSL based on Ruby
- ERB, EPP
- Extensions (ruby)
- Puppet Forge
- Red Hat Satellite 6

- Configurations in pure YAML
- Jinja2
- Extensions (Python)
- Ansible Galaxy
- Ansible Tower by Red Hat
Configuration Management Learning Curve

- Not understanding scale
- Write modules without keeping standards
- No linting or validating of code
- Reuse Modules
- Understand DSL/ERB/EPP/Jinja2
- Impact of CF-Mgt
- Parse, Validate Code
- Git / SCM
- Automated testing - Jenkins
- Using tools like Satellite 6 or Ansible Tower
- Cares about clean code

Time

- 1-3 Month
- 3-6 Month
- 6-x Month
CI/CD Satellite 6

- Build Engineers
  - git push

- Git
  - Kickstart templates
  - Puppet Modules
  - Internal Software
  - 3rd Party Software
  - Test Scripts

- Jenkins
  - Detect change
  - Build Artefacts
  - Push to Satellite 6
  - Trigger VM Build
  - Trigger Tap / Test Results

- Satellite 6
  - Content Library Management
    - RPMS / Puppet
  - Content Dev Management
    - RPMS / Puppet

- Test ME

- Pulp sync

- RHN

- https://github.com/RedHatEMEA/soe-ci
WHY CONFIGURATION MANAGEMENT IN THE CLOUD
Why Configuration Management In Cloud Environments

Provider specific templates build for resource management:

- AWS CloudFormations
- Azure ARM Templates JSON Orchestration Templates
- OpenStack Heat
- GCE
- Cloud-init

Configuration Management for software/configuration management:

- Puppet / Ansible
Red Hat MANAGEMENT TOOLS
Red Hat Management Tools

<table>
<thead>
<tr>
<th><strong>Red Hat Satellite 6</strong></th>
<th>Red Hat Satellite 6 delivers your Red Hat software efficiently and securely. Satellite 6 optimizes your Red Hat infrastructure and investment with full software lifecycle control, provisioning &amp; configuration, and subscription management.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Red Hat CloudForms</strong></td>
<td>Red Hat CloudForms controls your hybrid-cloud infrastructure. CloudForms is a single-pane-of-glass for controlling your hybrid-cloud environment that unifies management across clouds, with comprehensive insight &amp; discovery and full operational control.</td>
</tr>
<tr>
<td><strong>Ansible Tower by Red Hat</strong></td>
<td>Ansible automates your IT processes and applications deploys. Ansible Tower is an enterprise framework for controlling, securing and managing your Ansible automation. Tower provides automation job control, security and auditing, and delegation of automation jobs.</td>
</tr>
</tbody>
</table>
Automation / Configuration / Orchestration / Governance
RED HAT®
CLOUDFORMS
+
RED HAT®
SATELLITE
## CloudForms Satellite 6 Integration

### All Configuration Management Providers

<table>
<thead>
<tr>
<th>Provider Name</th>
<th>URL</th>
<th>Type</th>
<th>Zone</th>
<th>Last Refresh Date</th>
<th>Region Description</th>
<th>Status</th>
<th>Total Configured Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>sat6ldo.rdu.salab.redhat.com</td>
<td><a href="https://sat6ldo.rdu.salab.redhat.com">https://sat6ldo.rdu.salab.redhat.com</a></td>
<td>Configuration Manager (Red Hat Satellite)</td>
<td>default</td>
<td>06/16/16 15:08:59 UTC</td>
<td>Region 346</td>
<td>Valid</td>
<td>2</td>
</tr>
<tr>
<td>towerldo.rdu.salab.redhat.com</td>
<td><a href="https://towerldo.rdu.salab.redhat.com/api/v1">https://towerldo.rdu.salab.redhat.com/api/v1</a></td>
<td>Configuration Manager (Ansible Tower)</td>
<td>default</td>
<td>06/16/16 15:09:03 UTC</td>
<td>Region 346</td>
<td>Valid</td>
<td>16</td>
</tr>
</tbody>
</table>
CloudForms Satellite 6 Integration

Red Hat Satellite Provider ➤ Add ConfiguredSystem

Request  Purpose  Catalog  Customize  Schedule

Configured Systems

<table>
<thead>
<tr>
<th>Configured Systems</th>
<th>Hostname</th>
<th>Configuration Location</th>
<th>Configuration Organization</th>
<th>Operating System</th>
<th>Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>host117.rdu.salab.redhat.com</td>
<td>nyc</td>
<td>redhat</td>
<td></td>
<td>sat6kdo.rdu.salab.redhat.com</td>
</tr>
</tbody>
</table>

Configuration Profile *

Rhel7_Library_Servers
CloudForms Satellite 6 Integration

● Integration via configuration bootstrap.py script or api
● Bootstrap.py enables integration of new hosts with satellite 6 no matter where they are
  ○ Useful for Cloud Deployments where CloudForms manages the host:

```
# /usr/local/sbin/bootstrap.py -l admin -p PASSWORD -s sat6summit.osop.rhcloud.com -o 'redhat' -L 'nyc' -g RHEL7_Library_Servers -a ak-Reg_To_Library --unmanaged
```
CloudForms Satellite 6 Integration

This demo is inspired by the outage of a cloud provider in June 2016 in Australia.

It gives you an idea on how to truly do hybrid compute and application provisioning across on premise and all major cloud providers (Azure, AWS, GCE) and triage which clouds you would like to provision to.
1 Provision Instance
2 Reg. to Sat6 with activation key
3 Puppet is asking for its node classification from Sat6 and applies its configuration
4 DNS RR could be implemented (not in demo)
CloudForms Satellite 6 Integration Demo

https://www.youtube.com/v/nu9wMOIkRqA
Simple things should be simple and hard things possible
- Alan Kay
SIMPLE

- Human readable
- No special coding skills

POWERFUL

- Application deployment
- Configuration Management

AGENTLESS

- Uses OpenSSH & WinRM
- No agents to exploit or update
ANSIBLE MAKES CLOUDFORMS EASIER TO EXTEND

- Ansible is (much) easier to write than Ruby
- Leverage existing Playbooks!

CLOUDFORMS + TOWER SIMPLIFIES SERVICES

- CloudForms calls Tower
- Basis for cross-cloud portable applications
Ansible as a Service

1. Setup playbooks in Tower
2. Attach to CloudForms as a Service
3. Add a button to CloudForms (optional)
Example Playbook

---
- name: Update Linux Systems
  hosts: all
  remote_user: root
  tasks:
    - name: upgrade all packages
      yum: name=* state=latest
Standard Output

SSH password:

PLAY [Update Linux Systems] ********************************************

GATHERING FACTS ********************************************
ok: [192.168.124.95]

TASK: [upgrade all packes] ********************************************
changed: [192.168.124.95]

PLAY RECAP ********************************************
192.168.124.95 : ok=2  changed=1  unreachable=0  failed=0
<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Finished</th>
<th>Type</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Successful</td>
<td>5/31/2016 11:45:38 PM</td>
<td>Playbook Run</td>
<td>Update Linux Servers</td>
</tr>
<tr>
<td>25</td>
<td>Successful</td>
<td>5/31/2016 11:08:05 PM</td>
<td>Playbook Run</td>
<td>Update Linux Servers</td>
</tr>
<tr>
<td>21</td>
<td>Successful</td>
<td>5/31/2016 10:50:09 PM</td>
<td>Playbook Run</td>
<td>Update Linux Servers</td>
</tr>
<tr>
<td>20</td>
<td>Canceled</td>
<td>5/31/2016 10:40:25 PM</td>
<td>Playbook Run</td>
<td>Update Linux Servers</td>
</tr>
<tr>
<td>19</td>
<td>Successful</td>
<td>5/31/2016 10:34:19 PM</td>
<td>Playbook Run</td>
<td>Ansible Tower Test</td>
</tr>
<tr>
<td>18</td>
<td>Failed</td>
<td>5/31/2016 10:32:51 PM</td>
<td>Playbook Run</td>
<td>Ansible Tower Test</td>
</tr>
<tr>
<td>17</td>
<td>Failed</td>
<td>5/31/2016 10:31:50 PM</td>
<td>Playbook Run</td>
<td>Ansible Tower Test</td>
</tr>
<tr>
<td>16</td>
<td>Failed</td>
<td>5/31/2016 10:27:50 PM</td>
<td>Playbook Run</td>
<td>Ansible Tower Test</td>
</tr>
</tbody>
</table>
Now With More Buttons!

Service "Update Linux Servers"

Properties
- Name: Update Linux Servers
- Description: Update Linux Servers
- Management Engine GUID: 5d021d34-27ae-11e6-8a96-525400091a8d

Lifecycle
- Retirement Date: Never
- Retirement State: Administrator
- Group: EvmGroup-super_administrator

Relationships
- Parent Catalog Item: Update Linux Servers

#redhat #rhsummit
CloudForms Admin UI
## Requests

- **Order Request was Submitted**

### Filter By

- **Requester:** Administrator
- **Approval State:**
  - Approved
  - Denied
  - Pending Approval
- **Type:** All
- **Request Date:** Last 7 Days
- **Reason:**

### Table

<table>
<thead>
<tr>
<th>Status</th>
<th>Request State</th>
<th>Request ID</th>
<th>Requester</th>
<th>Request Type</th>
<th>Completed</th>
<th>Description</th>
<th>Approval State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ok</td>
<td>Active</td>
<td>1,000,000,000,004</td>
<td>Administrator</td>
<td>Service Provision</td>
<td></td>
<td>Provisioning Service [Update Linux Servers] from [Update Linux Servers]</td>
<td>Approved</td>
</tr>
</tbody>
</table>
Hard Problems Solved With One Line Of Ansible

Removing files from servers (Without rm -fR)
$ ansible webservers -m file -a "dest=/path/to/file state=absent"

Run backup script in background (20 Hr timeout)
$ ansible webserver -B 72000 -P 0 -a "'/bin/backup_cmd --do-stuff"

Show status of all web servers (10 in parallel)
$ ansible webservers -a "service nginx status " -f 10
Ansible as Automation

Direct integration into the CloudForms State machine through new methods
3 New Methods

- **wait_for_ip**: retrieve IP address of system
- **launch_ansible_job**: Runs job via Ansible Tower
- **wait_for_completion**: Waits until job has finished and check results
Self Service

CloudForms
CloudForms SSP
Ticketing Systems
Web Services
REST API

Requirements

RBAC Rules?
Quota enforcement? (Size, storage, or cost)
Approval Required? (If over a certain size?)
Workload placement (Cost, Capacity, etc.)
End of Life policies?

Build

Register IPAM / DNS
Create VM
Add networking
Add Storage
Add to Domain / LDAP
Register system

Configure

Update NTP/DNS
Perform OS Updates
Create user accounts
Install backup agents
Configure applications
Check against policies

Self Service

CloudForms
CloudForms SSP
Ticketing Systems
Web Services
REST API

Requirements

RBAC Rules?
Quota enforcement? (Size, storage, or cost)
Approval Required? (If over a certain size?)
Workload placement (Cost, Capacity, etc.)
End of Life policies?
SUMMARY

- Configuration tools can provide significant time savings
- There are multiple tools in the Red Hat toolbox
- Anything is possible when combining CloudForms, Satellite 6 and Ansible Tower
## Hybrid Cloud Management Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat CloudForms: Cutting VM creation time by 75% at General Mills</td>
<td>Thurs, Jun 30, 10:15 AM - 11:15 AM</td>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>Compliance, security automation, and remediation with Red Hat CloudForms, Red Hat Satellite, and Ansible Tower by Red Hat</td>
<td>Thurs, Jun 30, 3:30 PM - 4:30 PM</td>
<td>2005</td>
<td></td>
</tr>
<tr>
<td>Automation and configuration management across hybrid clouds with Red Hat CloudForms, Red Hat Satellite 6, and Ansible Tower</td>
<td>Wed, Jun 29, 4:45 PM - 5:45 PM</td>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>Automating Azure public and private clouds with Red Hat CloudForms 4</td>
<td>Wed, Jun 29, 4:45 PM - 5:45 PM</td>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>Red Hat CloudForms 2016 roadmap</td>
<td>Wed, Jun 29, 11:30 AM - 12:30 PM</td>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>Hands-on introduction to Red Hat CloudForms</td>
<td>Wed, Jun 29, 10:15 AM - 12:15 PM</td>
<td>3016 - Lab II</td>
<td></td>
</tr>
<tr>
<td>Enabling digital transformation via the Red Hat management portfolio</td>
<td>Tues, Jun 28, 10:15 AM - 11:15 AM</td>
<td>2004</td>
<td></td>
</tr>
</tbody>
</table>
QUESTIONS?
THANK YOU

Contact info:
laurent@redhat.com
miked@redhat.com

Material
http://blog.domb.net
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.