Button Push Deployments With Integrated Red Hat Open Management

The power of automation

Laurent Domb
Principal Cloud Solutions Architect

Michael Dahlgren
Senior Cloud Solutions Architect

Maxim Burgerhout
Senior Solutions Architect

May, 2017
About US

Michael Dahlgren
Sr. Cloud Specialist Solutions Architect
miked@redhat.com
RHCA
Red Hat

Laurent Domb
P. Cloud Specialist Solutions Architect
laurent@redhat.com
RHCA VI
Red Hat

Maxim Burgerhout
Sr. Solutions Architect
maxim@redhat.com
RHCA V
Red Hat

#redhat #rhsummit
Intro
Ansible Tower, CloudForms, Insights, Satellite 6

- Why do we care?
- What problems are we solving?
- How do the Red Hat tools address this?
- What does it look like in action?
75% of IT spend: $ Trillions per year
“The worst day in a company's life is the day they buy a large piece of software.”

75% of IT SPEND

$ TRILLIONS PER YEAR
"The worst day in a company's life is the day they buy a large piece of software."

**THE WHAT**

75% of IT SPEND

**THE HOW**

AUTOMATION IS THE HOW WITHOUT THE OVERHEAD

**THE WHERE**

$ TRILLIONS PER YEAR
One Button Push Away From Red Hat Management
Red Hat Management Automated
From start to finish in less than 3 hours with these ingredients

Satellite 6
CloudForms
Ansible Tower
Insights
Red Hat Management Automated

Prerequisites

- https://github.com/ldomb/rhsummit2017
- Minimum requirement ansible 2.2.1
- Ansible vault file with your passwords, private keys ...
- Ansible Tower License can be requested here:
  - https://www.ansible.com/license
- Satellite 6 Manifest
- An AWS account (AWS Cli )
- Private key for AWS instances
- CloudForms image in AWS ( uploadcfme.yaml )
Red Hat Management Automated
From start to finish in less than 3 hours with these ingredients

summit2017$ ansible-playbook buildrhmgmt.yaml --private-key=ldomb.pem --vault-password-file=./vaultpass -vv
RH-MANAGEMENT CORE
One click to rule them all - Foundation Installation Flow

Ansible Core

AWS
RH-MANAGEMENT ANSIBLE TOWER

One click to rule them all - Foundation Installation Flow
RH-MANAGEMENT TOOLS

One click to rule them all - Foundation Installation Flow

```bash
[Task] [manage-e2-instances : add_host] ******************************************************
changed: [localhost] => [item=(kernel): None, root_device_type: u'eb3', private_dns_name: u'ip-172-31-146-153.ec2.internal', public_ip: u'54.144.64.252', private_ip: u'172.31.146.153', id: u'i-abbe685dfbea72888', u'eb3 optimized': False, u'hostname': u'running', u'virtualization_type': u'hvm', root_device_name: u'/dev/sda1', u'ramdisk': None, block_device_mapping: {}], u'devices': [u'status': u'attached', u'delete_on_termination': True, u'volume_id': u'vol-0a88b8c3145e4626a01'], u'key_name': u'udlombo', u'image_id': u'ami-bf3769a1', u'tenancy': u'default', group_ids: [u'sg-56b1b3e2'], u'public_dns_name': u'e2-54.144.64.252.compute-1.amazonaws.com', u'state_code': 16, u'tags': [u'Environment': u'production', u'Type': u'tower', u'region': u'us-east-1'], u'launch_index': u'0', u'dns_name': u'e2-54.144.64.252.compute-1.amazonaws.com', u'region': u'us-east-1', u'launch_time': u'2017-04-28T15:43:22.002Z', u'instance_type': u'm3.large', u'architecture': u'x86_64', hypervisor: u'xen']
```

```bash
[Task] [manage-e2-instances : Wait for SSH banners] ******************************************************
ok: [localhost] => [item=(kernel): None, root_device_type: u'eb3', private_dns_name: u'ip-172-31-146-153.ec2.internal', public_ip: u'54.144.64.252', private_ip: u'172.31.146.153', id: u'i-abbe685dfbea72888', u'eb3 optimized': False, u'hostname': u'running', u'virtualization_type': u'hvm', root_device_name: u'/dev/sda1', u'ramdisk': None, block_device_mapping: {}], u'devices': [u'status': u'attached', u'delete_on_termination': True, u'volume_id': u'vol-0a88b8c3145e4626a01'], u'key_name': u'udlombo', u'image_id': u'ami-bf3769a1', u'tenancy': u'default', group_ids: [u'sg-56b1b3e2'], u'public_dns_name': u'e2-54.144.64.252.compute-1.amazonaws.com', u'state_code': 16, u'tags': [u'Environment': u'production', u'Type': u'tower', u'region': u'us-east-1'], u'launch_index': u'0', u'dns_name': u'e2-54.144.64.252.compute-1.amazonaws.com', u'region': u'us-east-1', u'launch_time': u'2017-04-28T15:43:22.002Z', u'instance_type': u'm3.large', u'architecture': u'x86_64', hypervisor: u'xen']
```

PLAY [create tower] ***********************************************

```bash
[Task] [setup] ***********************************************
ok: [54.144.64.252]
```

```bash
[Task] [buildansibletower : get tar for ansibletower] ***********************************************
changed: [54.144.64.252]
```

```bash
[Task] [buildansibletower : untar /tmp/ansible-tower-setup-bundle.tar.gz] *********
changed: [54.144.64.252]
```

```bash
[Task] [buildansibletower : replace /tmp/ansible-tower-setup-bundle-3.1.2-1.el7/roles/nginx/tasks/tasks.yml] ***
changed: [54.144.64.252]
```

```bash
[Task] [buildansibletower : add /etc/tower path to setting.py] *********
changed: [54.144.64.252]
```

```bash
[Task] [buildansibletower : copy inventory to setup folder] ***********
changed: [54.144.64.252]
```

```bash
[Task] [buildansibletower : execute the tower installation] ************
```
Red Hat Management Automated
Building Ansible Tower - Gains - Workflow Editor

Ansible Tower orchestrates your infrastructure in a secure fashion
RH-MANAGEMENT SATELLITE 6 / CLOUDFORMS

One click to rule them all - Foundation Installation Flow
RH-MANAGEMENT TOOLS

One click to rule them all - Foundation Installation Flow

Job 1

PLAY [create cfme] **************************** 13:47:12

  Ok: [52.23.172.218]
  META: ran handlers

  changed: [52.23.172.218]

TASK [buildcfme : ensure chrony service is started and enabled] **************************** 13:47:19
  Ok: [52.23.172.218]

TASK [buildcfme : ensure chrony is getting restarted if necessary] **************************** 13:47:20
  changed: [52.23.172.218]

TASK [buildcfme : perform appliance basic configuration] **************************** 13:47:21
  changed: [52.23.172.218]

RH-MANAGEMENT CLOUDFORMS

One click to rule them all - Foundation Installation Flow
Red Hat Management Automated

CloudForms - Ansible Tower integration gains

<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>Ansible Tower Configuration Manager</td>
<td><a href="https://ip-172-31-226-121.ec2.internal/api/v1">https://ip-172-31-226-121.ec2.internal/api/v1</a></td>
<td>Configuration Manager (Ansible Tower)</td>
<td>default</td>
<td>04/24/17 15:32:22 UTC</td>
<td>Region 99</td>
<td>Valid</td>
<td>51</td>
</tr>
</tbody>
</table>
Red Hat Management Automated

CloudForms - Ansible Tower integration gains

AWS01 (Summary)

Properties
- Region: US East (Northern Virginia)
- Type: Amazon EC2
- Management Engine Build: a9ebf7b6-1ecdf11c7-83c8-12119dd96408
- Region: us-east-1

Status
- Default Credentials: Valid
- Last Refresh: Success - 17 Minutes Ago

Configuration
- Arbitration Profiles: 0

Relationships
- Network Manager: AWS01 Network Manager
- Availability zones: 5
- Host aggregates: 0
- Cloud tenants: 0
- Flavors: 76
- Security Groups: 25
- Instances: 12

INVENTORIES

SEARCH

NAME ▲ ORGANIZATION ▼

- aws (Default)
- cloudforms (Default)
Red Hat Management Automated

CloudForms - Ansible Tower integration gains

Inventory Groups under Ansible Tower Provider "Ansible Tower Configuration Manager"

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Configured Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws</td>
<td>34</td>
</tr>
<tr>
<td>cloudforms</td>
<td>7</td>
</tr>
<tr>
<td>Demo Inventory</td>
<td>1</td>
</tr>
<tr>
<td>satellite6</td>
<td>9</td>
</tr>
</tbody>
</table>
Red Hat Management Automated

CloudForms - Ansible Tower integration gains

Job Templates under "Ansible Tower Configuration Manager"

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Created On</th>
<th>Updated On</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildrhmanagement</td>
<td>Job Template (Ansible Tower)</td>
<td>Build RH Management</td>
<td>04/17/17 12:29:01 UTC</td>
<td>04/17/17 12:29:01 UTC</td>
</tr>
<tr>
<td>cis-compliance-test</td>
<td>Job Template (Ansible Tower)</td>
<td>04/17/17 07:43:07 UTC</td>
<td>04/17/17 07:43:07 UTC</td>
<td></td>
</tr>
<tr>
<td>createcfme</td>
<td>Job Template (Ansible Tower)</td>
<td>04/17/17 15:44:26 UTC</td>
<td>04/17/17 15:44:26 UTC</td>
<td></td>
</tr>
<tr>
<td>createcfmshell</td>
<td>Job Template (Ansible Tower)</td>
<td>04/17/17 15:44:27 UTC</td>
<td>04/17/17 15:44:27 UTC</td>
<td></td>
</tr>
<tr>
<td>createsat6</td>
<td>Job Template (Ansible Tower)</td>
<td>04/17/17 15:44:27 UTC</td>
<td>04/17/17 15:44:27 UTC</td>
<td></td>
</tr>
<tr>
<td>createsat6shell</td>
<td>Job Template (Ansible Tower)</td>
<td>04/17/17 15:44:27 UTC</td>
<td>04/17/17 15:44:27 UTC</td>
<td></td>
</tr>
</tbody>
</table>

#redhat #rhsummit
Red Hat Management Automated

CloudForms - Ansible Tower integration gains

CloudForms + Ansible Tower = Build anything anywhere, any time
RH-MANAGEMENT SATELLITE 6 / CLOUDFORMS

One click to rule them all - Foundation Installation Flow

Ansible Core

Job 1

Job 2

AWS
RH-MANAGEMENT TOOLS

One click to rule them all - Foundation Installation Flow
RH-MANAGEMENT SATELLITE 6

One click to rule them all - Foundation Installation Flow
# Red Hat Management Automated

**Satellite 6 - CloudForms - Ansible Tower - Integration Gains**

## 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>Ansible Tower Configuration Manager</td>
<td><a href="https://ip-172-31-226-121.ec2.internal/api/v1">https://ip-172-31-226-121.ec2.internal/api/v1</a></td>
<td>Configuration Manager (Ansible Tower)</td>
<td>default</td>
<td>04/24/17 15:32:22 UTC</td>
<td>Region 99</td>
<td>Valid</td>
<td>51</td>
</tr>
</tbody>
</table>
## Red Hat Management Automated

**Satellite 6 - CloudForms - Ansible Tower - Integration Gains**

---

### Red Hat Satellite Provider

**Add ConfiguredSystem**

<table>
<thead>
<tr>
<th>Request</th>
<th>Purpose</th>
<th>Catalog</th>
<th>Customize</th>
<th>Schedule</th>
</tr>
</thead>
</table>

**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>host79.rdu.salab.redhat.com</td>
<td>nyc</td>
<td>redhat</td>
<td></td>
<td>sat8clo</td>
</tr>
</tbody>
</table>

**Configuration Profile**

- RHEL7_Crash_Base

---

Note: Fields marked with * are required.
## Red Hat Management Automated

**Satellite 6 - CloudForms - Ansible Tower - Integration Gains**

<table>
<thead>
<tr>
<th>INVENTORIES</th>
<th>ORGANIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws</td>
<td>Default</td>
</tr>
<tr>
<td>cloudforms</td>
<td>Default</td>
</tr>
<tr>
<td>satellite6</td>
<td>Default</td>
</tr>
</tbody>
</table>
Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains

Hosts

Filter ...

<table>
<thead>
<tr>
<th>Name</th>
<th>Operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip-172-31-159-178.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-165-67.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-177-77.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-238-93.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-45-59.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-47-45.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-54-120.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
<tr>
<td>ip-172-31-96-218.ec2.internal</td>
<td>RedHat 7.3</td>
</tr>
</tbody>
</table>

Displaying all 8 entries - 0 selected

CloudForms

Compliance: OpenSSL Security

- **VM and Instance Compliance**: DROWN OpenSSL Vulnerability
  - Vulnerable DROWN openssl packages (RHEL5/6/7)
  - VM Compliance Check
    - Generate log message
    - Mark as Non-Compliant

- **Policy**: DROWN OpenSSL Vulnerability
  - **Condition**: Vulnerable DROWN openssl packages (RHEL5/6/7)
    - Compliance Check on: 03/01/16 13:35:07 AEST
    - Compliance Check on: 03/01/16 12:58:47 AEST
    - Compliance Check on: 03/01/16 12:50:58 AEST
CloudForms + Ansible Tower + Satellite = Build anything anywhere anytime and make it secure!
RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow
RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow
RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow
Red Hat Management Automated

Insights - CloudForms - Ansible Tower - Satellite 6 - integration gains

Use this chart to drill down and discover problems within your organization.

There are 29 actions detected from systems in your organization.

Overview

<table>
<thead>
<tr>
<th>Section</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>24</td>
</tr>
<tr>
<td>Stability</td>
<td>4</td>
</tr>
<tr>
<td>Performance</td>
<td>1</td>
</tr>
</tbody>
</table>

∀ 3 systems are not checking in

VIEW SYSTEMS AND RESOLVE

#redhat #rhsummit
Kdump crashkernel reservation failed due to improper configuration of crashkernel parameter.

Kdump is unable to reserve memory for the kdump kernel. The kdump service has not started and a vmcore will not be captured if the host crashes, which will make it difficult for our support technicians to determine why the machine crashed.

## Impacted Systems

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Reported</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo-insights-rhel65</td>
<td>about 9 hours ago</td>
<td>View</td>
</tr>
<tr>
<td>demo-insights-rhel70.demo.mbu.redhat.com</td>
<td>about a month ago</td>
<td>View</td>
</tr>
<tr>
<td>localhost.localdomain.localdomain</td>
<td>2 months ago</td>
<td>View</td>
</tr>
</tbody>
</table>
Red Hat Management Automated

Insights - CloudForms - Ansible Tower - Satellite 6 - integration gains

Performance > NUMA performance regression on specific kernels

Detected issue

This host is a NUMA system running kernel version 2.6.32-431.el6.x86_64.

A change was introduced in Red Hat Enterprise Linux 6.5 to make machines with weird topologies bootable. However, for normal systems this change can lead to a NUMA mapping with incorrect cpu_power settings for all domains other than the first. As a result, under some workloads, performance issues can be observed.

Steps to resolve

To fix this issue, Red Hat recommends that you update the deployed kernel to version 2.6.32-431.20.3.el6 or later.

```
# yum update kernel
```

If you are unable to update your kernel at this time, an effective workaround is to use the taskset command to force a process to run on a specific CPU.
CloudForms + Ansible Tower + Satellite + Insights = Build anything, anywhere, anytime, make it secure and keep a piece of mind that you’ve done it right.
Provision and manage servers and networking anywhere, anytime and be sure it's secure and compliant. Keep in mind we are watching you!
One Button Push To RH Management Suite

http://bit.ly/2oQwxxF
Here’s a practical example, kids!
A lot of applications out there follow the n-tier paradigm
  ○ This means applications functions are split out into multiple servers

Traditionally, deploying applications like this has involved a lot of scripting

Enter the combination of CloudForms, Satellite 6 and Ansible
AN N-TIER APPLICATION

Internet → Load balancer → Web server → Database
HOW DOES THAT WORK?

- Automating the deploying an n-tier application requires
  - Something to create the initial systems
  - Something to configure the initial systems
  - Something to get the software from
  - Something to tie things together
  - Someplace my end users can go to, to press a button labeled ‘gimme’
HOW DOES THAT WORK?

- Automating the deploying an n-tier application requires
  - Something to create the initial systems ➢ CloudForms
  - Something to configure the initial systems ➢ Satellite 6
  - Something to get the software from ➢ Satellite 6
  - Something to tie things together ➢ Ansible Tower by Red Hat
  - Someplace my end users can go to, to press a button labeled ‘gimme’ ➢ CloudForms
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata

Automatically deploy Insights client as well!
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata

Automatically deploy Insights client as well!

Pass control to Ansible Tower for application deployment

Automatically deploy Insights client as well!
CLOUDFORMS SELF-SERVICE
How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata

Pass control to Ansible Tower for application deployment

Automatically deploy Insights client as well!

CloudForms monitors progress and informs user when done
CLOUDFORMS SELF-SERVICE

How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata

Pass control to Ansible Tower for application deployment

Automatically deploy Insights client as well!

CloudForms monitors progress and informs user when done
How hard is that?

For VMs or groups of VMs, setting up self-service in CloudForms is actually fairly straightforward.

- CloudForms can consume and store:
  - Heat templates
  - CloudFormation templates
  - Azure templates

- CloudForms can automatically create dialogs from the parameters in those templates
How hard is that?

Offering Ansible Job Templates to your users isn’t much different. (As already mentioned.)

- CloudForms connects to Ansible Tower
- Create service dialogs based on the surveys in Ansible Job Templates
- You can customize these after creating them
So now we have two self service items

This calls for a bundle!
BUNDLING CATALOG ITEMS

Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata

Pass control to Ansible Tower for application deployment

Automatically deploy Insights client as well!

CloudForms monitors progress and informs user when done
# A Catalog Bundle!

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Action Order</th>
<th>Provision Order</th>
<th>Action</th>
<th>Delay (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudFormation: Loadbalanced Web Cluster</td>
<td>Four node, load-balanced Web Apache / MariaDB Cluster</td>
<td>1</td>
<td>1</td>
<td>Power On</td>
<td></td>
</tr>
<tr>
<td>Ansible Tower: Deploy Wordpress Cluster</td>
<td>Deploy Wordpress Cluster based on Satellite 6 hostgroups</td>
<td>2</td>
<td>2</td>
<td>Do Nothing</td>
<td>5</td>
</tr>
</tbody>
</table>

Service Catalog Item "Bundle: Load-balanced Wordpress Cluster"
A CatalogBundle!
How does this work? With a state machine!

- A state machine is like a production line, with robots at stations along the line to perform actions

- Each of my catalog items has a state machine that defines the steps to deliver the item
  - a set of predefined steps
  - a set of empty placeholders

- Use the placeholders to execute additional, custom steps for deployment
Customizing state machines: example 1

- For the example, we customized the state machine for CloudFormation deployments
- Deployment should only then be finished when the Satellite part is done
- **Solution**: use one of the placeholders to query Satellite API for existence and configuration status of the new machines
- I’ve put this script up as a Gist on Github, so you can copy and improve upon it

https://gist.github.com/wzzrd/7cc7bab19b049eb4aa8842d2bf77026e
Customizing state machines: example 2

- We needed to pass the VMs created during the first catalog item (CloudFormation) to the Ansible Tower Job Template

- **Solution**: store the names of the newly created VMs in a variable, read the variable during the initialization of the Ansible Job catalog item

- Saving of the hostnames Happens in same script as previous customization example
- Customized method to start the Ansible Tower Job Template:

  https://gist.github.com/wzzrd/8a0c9e38f91668589049e32d20943eb0
How hard is customizing state machines?

- A state machine is stored in a CloudForms Automation domain
- A table with rows for each “robot” along the assembly line
- Stored in Git as YAML
- Copy the ones that ship with CFME to your own domain, edit as required
- Each “robot” is a Ruby method, and we ship many examples :)

```bash
# redhat # rhsummit
```
Want to learn more?

- There is an excellent book on CloudForms automation
- It’s freely available on our website

http://red.ht/2oYQttJ
I have a demo video, but it didn’t fit this presentation :(  
Good news is, it’s up on YouTube as of RIGHT NOW!

YouTube: http://bit.ly/2qqkc0f

Let us know what you think!
Our email addresses are on the intro slide, ask us any question by mail, or drop by the CloudForms booth: we’ll all be manning it this week!
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos

facebook.com/redhatinc
twitter.com/RedHatNews
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.
Resources

Links to resources used in this presentation

Resources used for this presentation

- https://access.redhat.com/articles/2258471 (hammer cheat sheet)
- https://github.com/rhtconsulting/cfme-rhconsulting-scripts
- https://galaxy.ansible.com/juliovp01/satellite6-install/ (original playbook for sat6)