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

- The journey to configuration management and automation across hybrid cloud environments
- Why use configuration management in hybrid cloud environments
- How are they integrated
  - CloudForms + Satellite 6
  - CloudForms + Ansible Tower

WHERE WE ALL STARTED

The Path To Configuration Management / DevOps

Scripts

```
#!/bin/sh
```

Documentation

Chaos

TO BE MORE AGILE

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 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%

Transform Your Organization

OPENNESS

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

Configuration Management Learning Curve

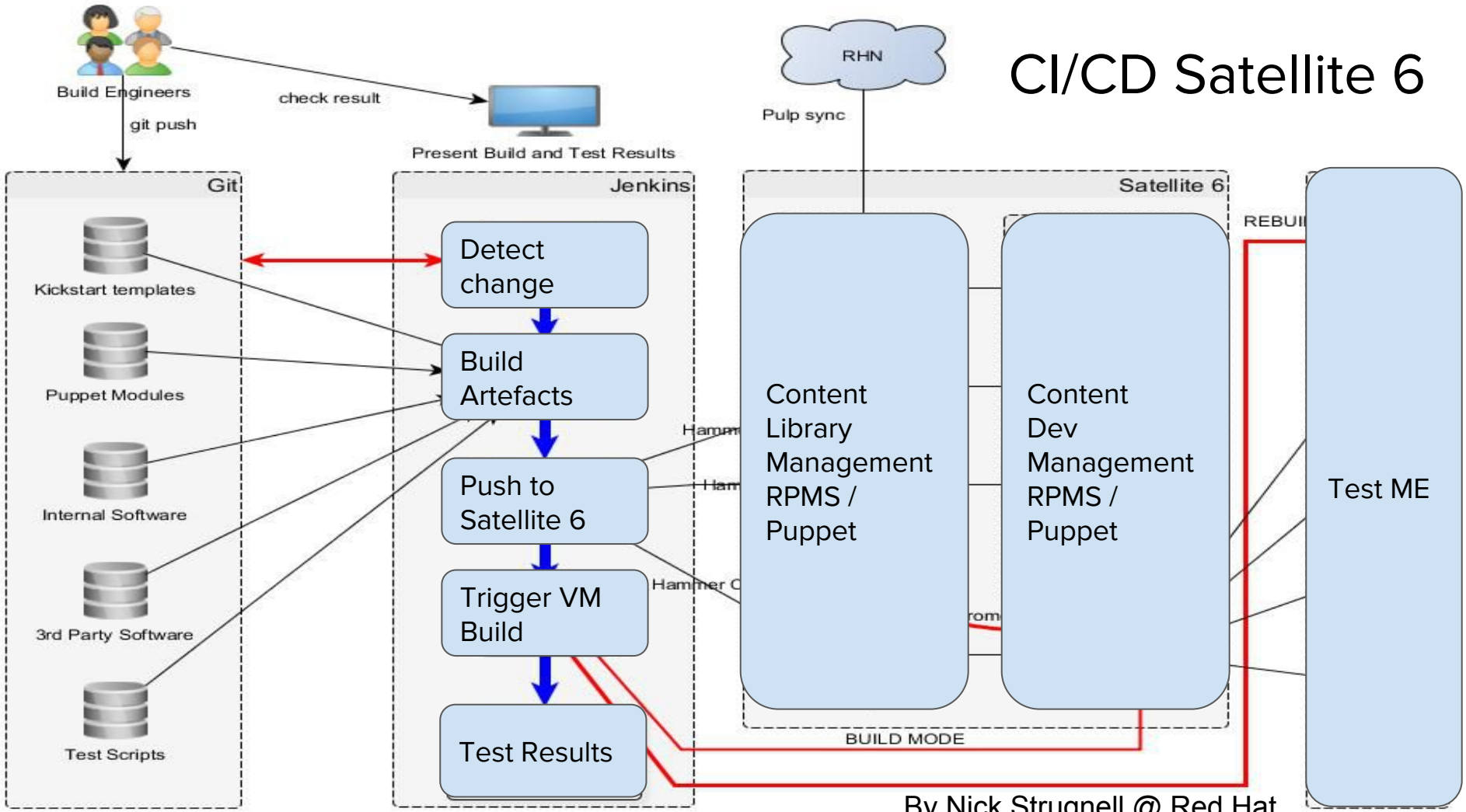
- Not understanding scale
- Write modules without keeping standards
- No linting or validating of code

- Reuse Modules
- Understand DSL/ERB/EPP/Jinja2
- Impact of CF-Mgt
- Parse, Validate Code

- Git
- 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



By Nick Strugnell @ Red Hat

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 Deployment Manager

Configuration Management for software/configuration management:

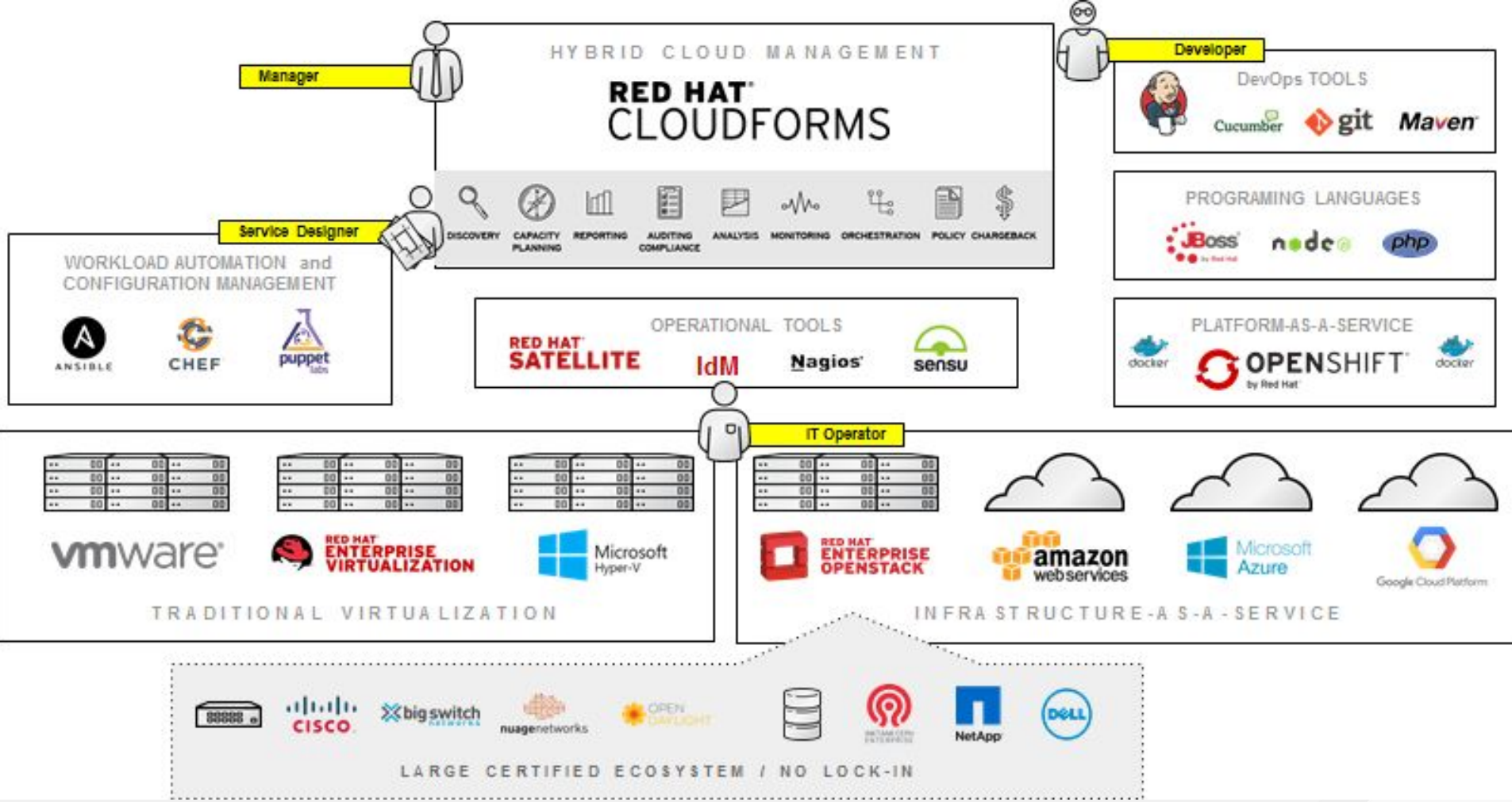
- Puppet / Ansible Tower

Red Hat MANAGEMENT TOOLS

Red Hat Management Tools

Red Hat Satellite 6	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 & configuration, and subscription management.
Red Hat CloudForms	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 & discovery and full operational control.
Ansible Tower by Red Hat	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.

Automation / Configuration / Orchestration / Governance

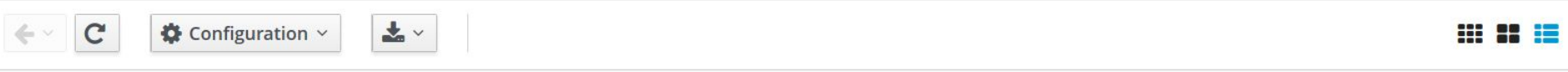

Manager
HYBRID CLOUD MANAGEMENT
RED HAT
CLOUDFORMS
Developer
DevOps TOOLS
Cucumber
git
Maven
Service Designer
DISCOVERY
CAPACITY PLANNING
REPORTING
AUDITING COMPLIANCE
ANALYSIS
MONITORING
ORCHESTRATION
POLICY
CHARGEBACK
PROGRAMING LANGUAGES
JBoss
node
php
WORKLOAD AUTOMATION and CONFIGURATION MANAGEMENT
ANSIBLE
CHEF
puppet
OPERATIONAL TOOLS
RED HAT SATELLITE
IdM
Nagios
sensu
PLATFORM-AS-A-SERVICE
docker
OPENSHIFT
by Red Hat
docker
IT Operator
vmware
RED HAT ENTERPRISE VIRTUALIZATION
Microsoft Hyper-V
TRADITIONAL VIRTUALIZATION
RED HAT ENTERPRISE OPENSTACK
amazon web services
Microsoft Azure
Google Cloud Platform
INFRASTRUCTURE-AS-A-SERVICE
CISCO
bigswitch
nuagenetworks
OPEN DAYLIGHT
NetApp
DELL
LARGE CERTIFIED ECOSYSTEM / NO LOCK-IN

RED HAT® CLOUDFORMS

+

RED HAT® SATELLITE

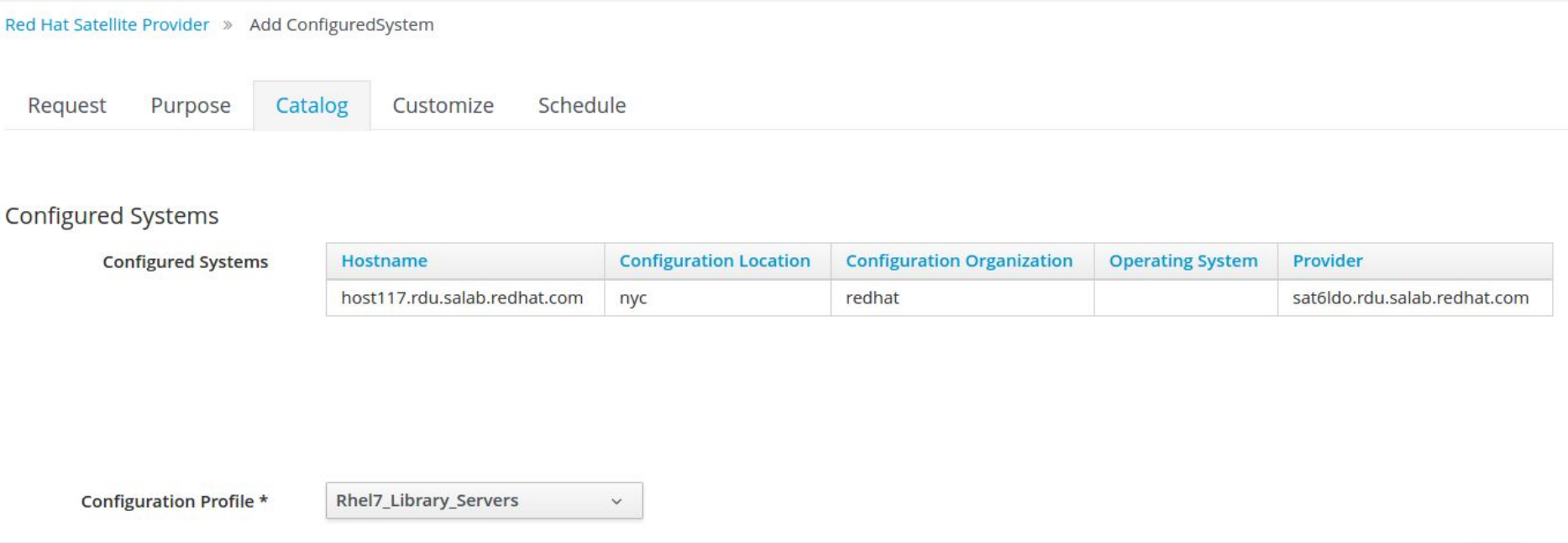
CloudForms Satellite 6 Integration



All Configuration Management Providers

		Provider Name	URL	Type	Zone	Last Refresh Date	Region Description	Status	Total Configured Systems
☐		sat6ldo.rdu.salab.redhat.com Configuration Manager	https://sat6ldo.rdu.salab.redhat.com	Configuration Manager (Red Hat Satellite)	default	06/16/16 15:08:59 UTC	Region 346	Valid	2
☐		towerldo.rdu.salab.redhat.com Configuration Manager	https://towerldo.rdu.salab.redhat.com/api/v1	Configuration Manager (Ansible Tower)	default	06/16/16 15:09:03 UTC	Region 346	Valid	16

CloudForms Satellite 6 Integration



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:

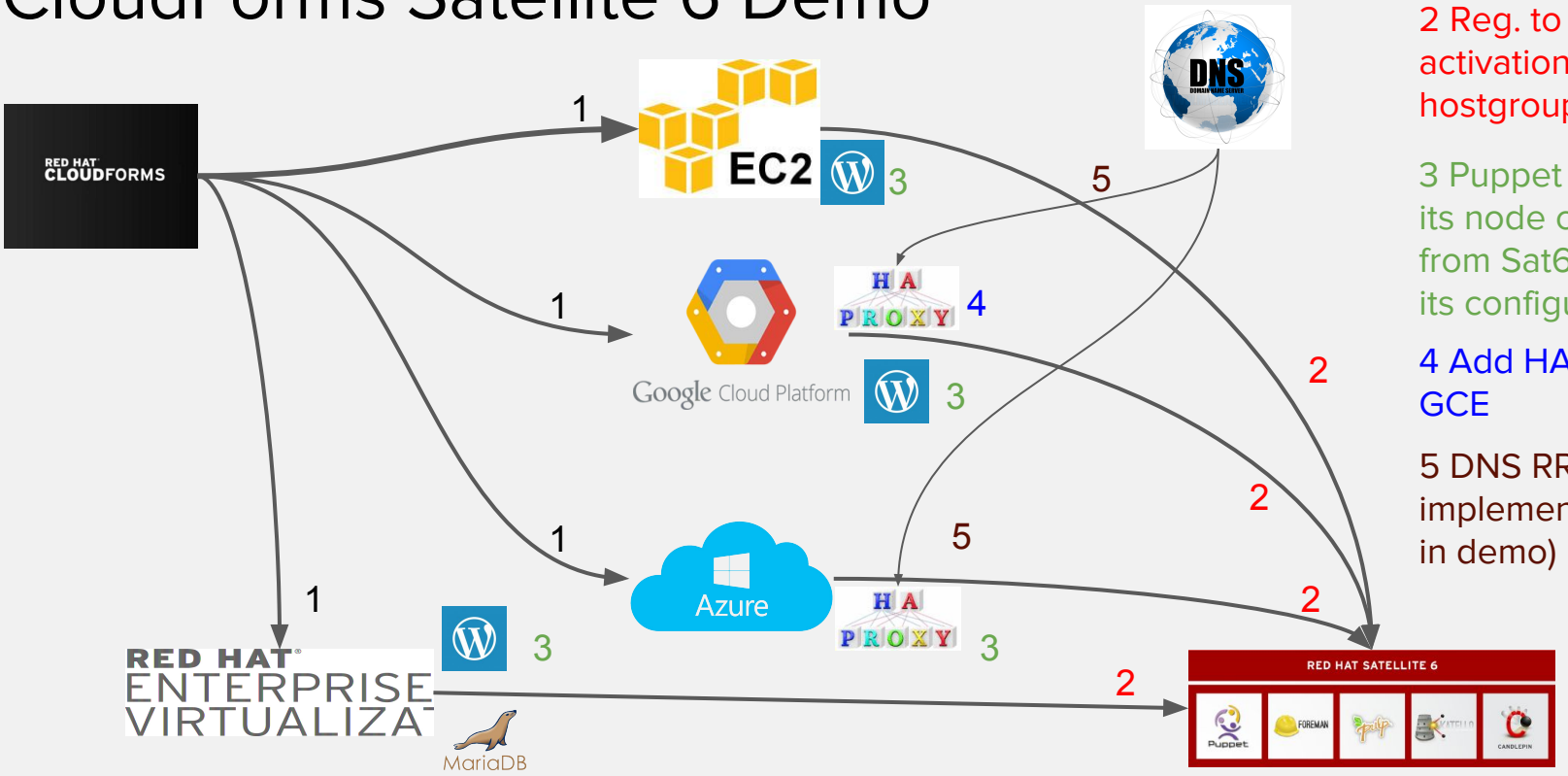
```
# /usr/local/sbin/bootstrap.py -l admin -p PASSWORD -s
sat6summit.osop.rhcloud.com -o 'redhat' -L 'nyc' -g
RHEL7_Library_Servers -a ak-Reg_To_Library --unmanaged
```

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.

CloudForms Satellite 6 Demo



1 Provision Instance

2 Reg. to Sat6 with activation key and hostgroup

3 Puppet is asking for its node classification from Sat6 and applies its configuration

4 Add HAProxy to GCE

5 DNS RR could be implemented (not in demo)

HAProxy Module (custom facts)

gce_public_hostname=$(getent hosts $(curl -s http://169.254.169.254 /computeMetadata/v1beta1/instance/network-interfaces/0/access-configs/0/external-ip) | awk {'print $2'})
gce_public_ipv4=$(curl http://169.254.169.254 /computeMetadata/v1beta1/instance/network-interfaces/0/access-configs/0/external-ip)

ec2_public_ipv4=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
ec2_public_hostname=$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)

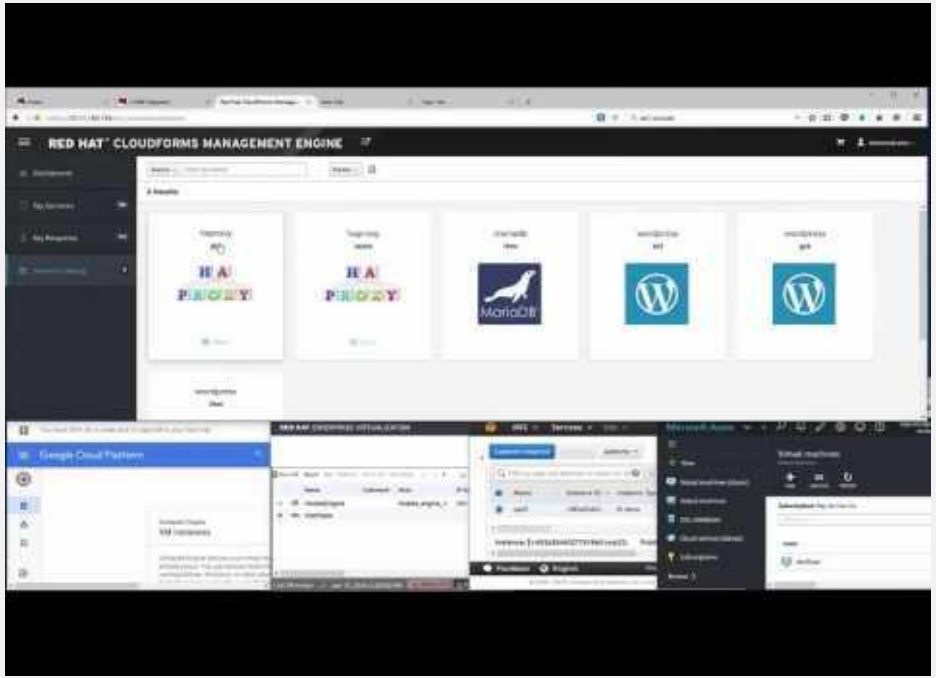
HAProxy Module (init.pp)

```
class haproxywp (
    $foreman_url   = "",
    $foreman_user  = "",
    $foreman_pass  = "",

) {

  $gce = { item  => 'fact_values',
    search       => "(name = gce_public_ipv4 or name = gce_public_hostname) and host !~ ${hostname}",
    per_page     => '20',
    foreman_url  => $foreman_url,
    foreman_user => $foreman_user,
    foreman_pass => $foreman_pass }

  $rhev = { item => 'fact_values',
    search       => '(name = rhev_public_ipv4 or name = rhev_public_hostname)',
    per_page     => '20',
    foreman_url  => $foreman_url,
    foreman_user => $foreman_user,
    foreman_pass => $foreman_pass }

  $ec2 = { item  => 'fact_values',
    search       => '(name = ec2_public_ipv4 or name = ec2_public_hostname) and host ~ %\.ec2\.internal',
    per_page     => '20',
    foreman_url  => $foreman_url,
    foreman_user => $foreman_user,
    foreman_pass => $foreman_pass }

  $gcehosts  = foreman($gce)
  $ec2hosts  = foreman($ec2)
  $rhevhosts = foreman($rhev)

  file {'/etc/haproxy/haproxy.cfg':
        content => template('haproxywp/haproxy.cfg.erb'),
        owner   => 'root'
```

HAProxy Module (haproxy.cfg.erb)

```
backend wordpress-backend
    balance     roundrobin
    mode        http

<% if @gcehosts -%>
  <% @gcehosts.each do |host,data| -%>
server <%= data['gce_public_hostname'] %> <%= data['gce_public_ipv4'] %>:80 check
  <% end -%>
<% end -%>
<% if @ec2hosts -%>
  <% @ec2hosts.each do |host,data| -%>
server <%= data['ec2_public_hostname'] %> <%= data['ec2_public_ipv4'] %>:80 check
  <% end -%>
<% end -%>
<% if @rhevhosts -%>
  <% @rhevhosts.each do |host,data| -%>
server <%= data['rhev_public_hostname'] %> <%= data['rhev_public_ipv4'] %>:80 check
  <% end -%>
<% end -%>
```

CloudForms Satellite 6 Hybrid Cloud Integration Demo



https://www.youtube.com/v/nu9wMOIkRqA

+

RED HAT®
CLOUDFORMS

Simple things should be simple and hard things possible

- Alan Kay

SIMPLE

Human readable

No special coding skills

Get productive quickly

POWERFUL

App deployment

Configuration management

Orchestrate the app lifecycle

AGENTLESS

Uses OpenSSH & WinRM

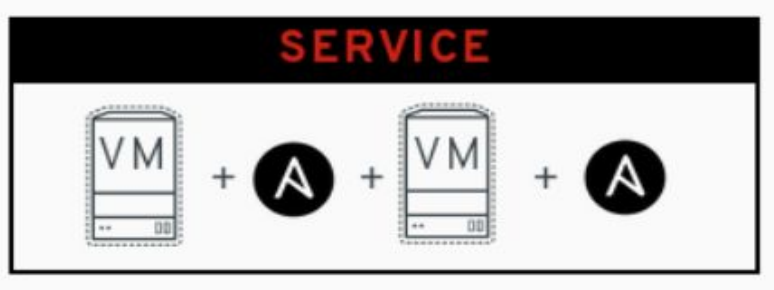
No agents to exploit or update

More efficient & more secure

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

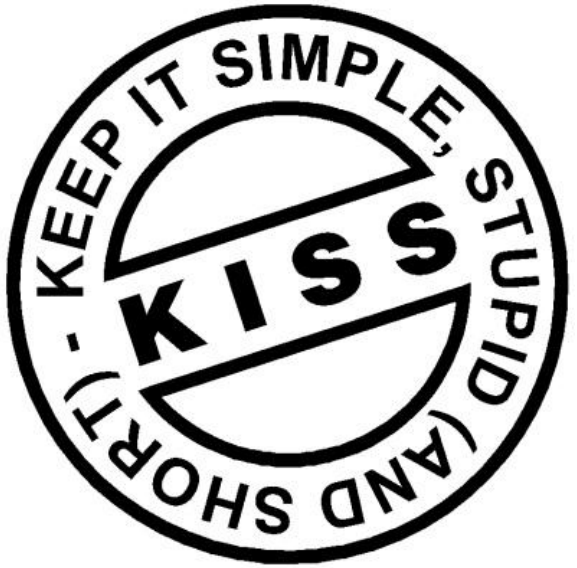
1. Setup playbooks in Tower

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
```



Status Successful

Timing Started 05/31/16 23:45:19 Finished 05/31/16 23:45:38 Elapsed 00:00:18

more

Plays Play Name All Failed

Started	Elapsed	Status	Name
23:45:21	00:00:16		Update Linux Systems

Tasks Task Name All Failed

Started	Elapsed	Status	Name	Host Status
23:45:21	00:00:03		Gathering Facts	1
23:45:25	00:00:13		upgrade all packes	1

Host Events Host Name All Failed

Status	Host	Item	Message
	192.168.124.95		

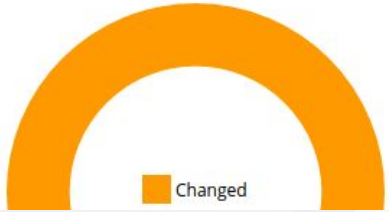
Events Summary Host Name All Failed

OK Changed Unreachable Failed

Host	Completed Tasks
192.168.124.95	2 1

1 host (100%) Changed.

Host Summary

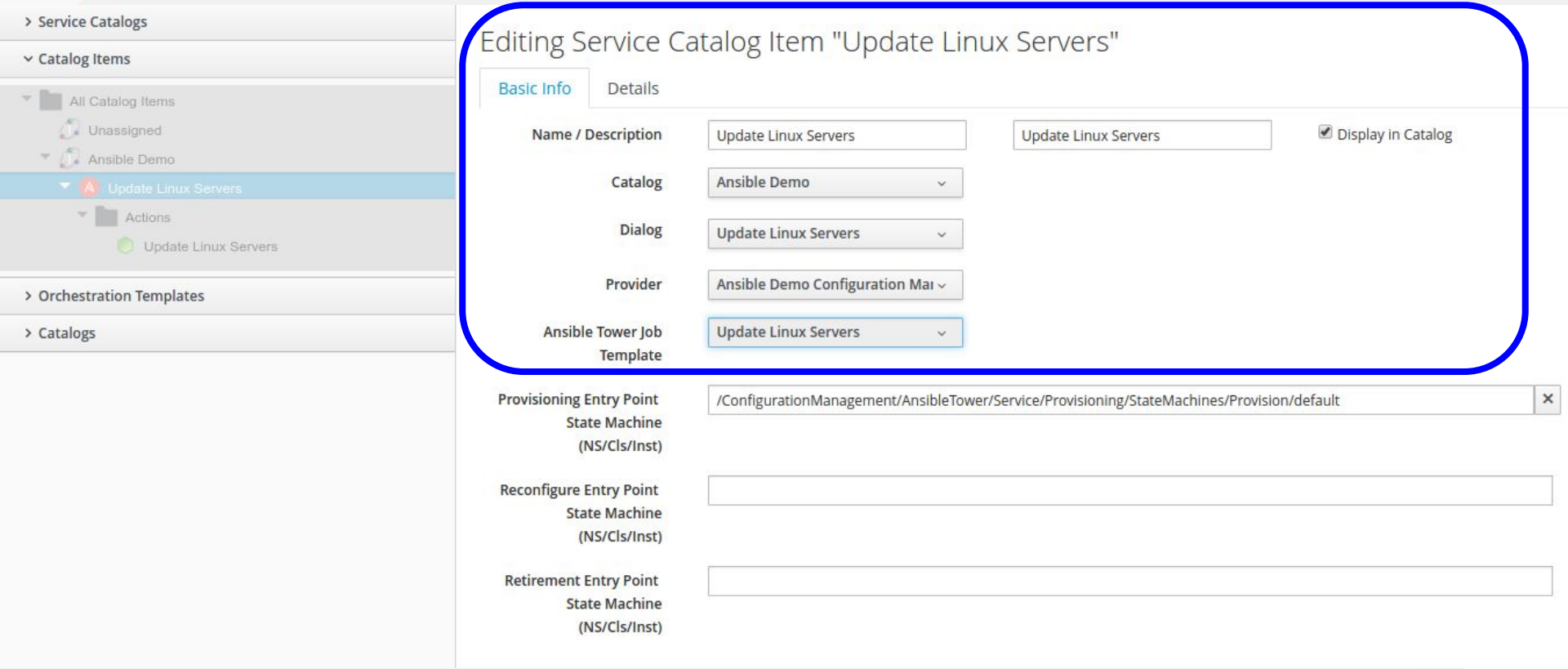


Job Status successful

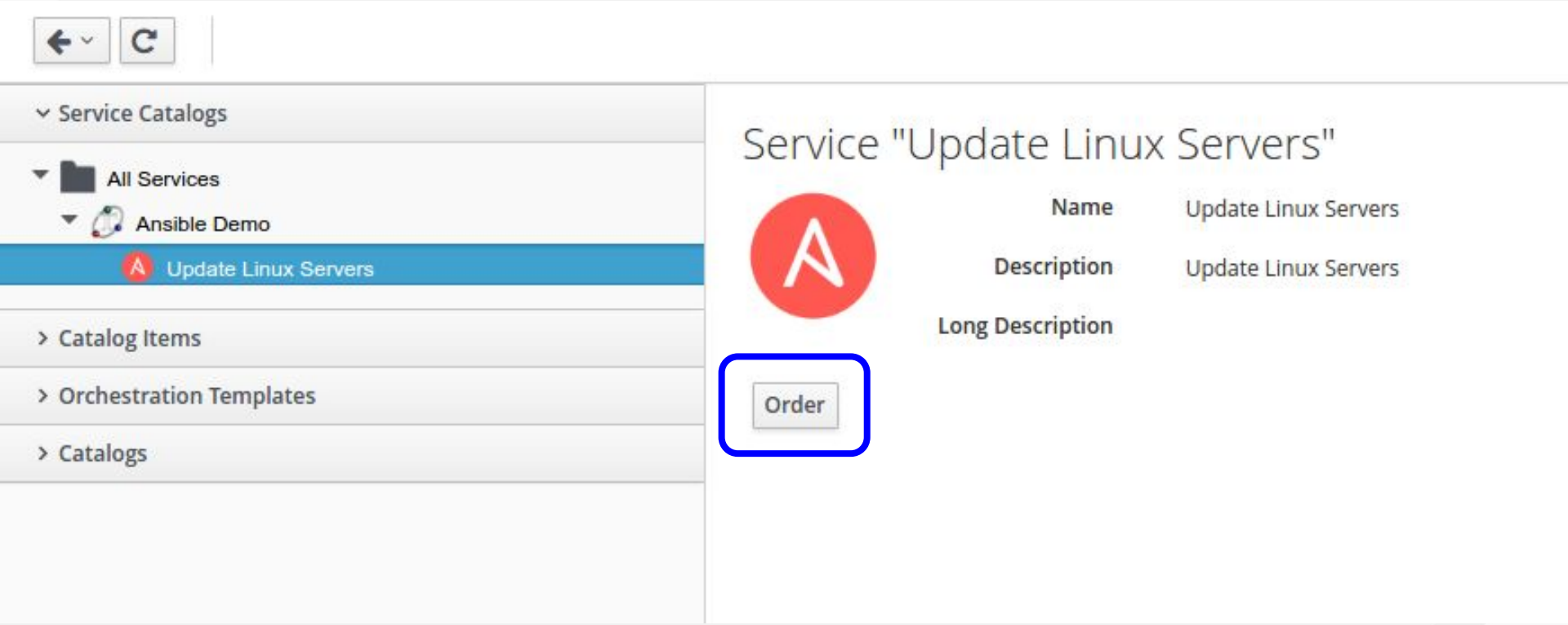
Standard Output

```
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
```

CloudForms Admin UI



CloudForms Admin UI



RED HAT® CLOUDFORMS MANAGEMENT ENGINE

Administrator

- Dashboard
- My Services 2
- My Requests 5
- Service Catalog 1

« Back to Service Catalog » **Service:** Update Linux Servers

Update Linux Servers

Update Linux Servers

Add to Shopping Cart

Update Linux Servers

Basic Information

Options

Enter list of hosts separated by : *

Cloud Intel
Red Hat Insights
Services
Compute
Configuration
Networks
Control
Automate
Optimize

Reload

Requests

Order Request was Submitted

Filter By

Requester: Administrator

Approval State: Approved Denied Pending Approval

Type: All

Request Date: Last 7 Days

Reason:

Apply Reset Default

	Status	Request State	Request ID	Requester	Request Type	Completed	Description	Approval State
	Ok	Active	1,000,000,000,004	Administrator	Service Provision		Provisioning Service [Update Linux Servers] from [Update Linux Servers]	Approved

Configuration | Policy

- Add a new Button Group
- Add a new Button
- Edit this Item
- Remove Item from the VMDB

Add a new Button

OPTIONAL

Service Catalog Item "Update Linux Ser

Basic Information

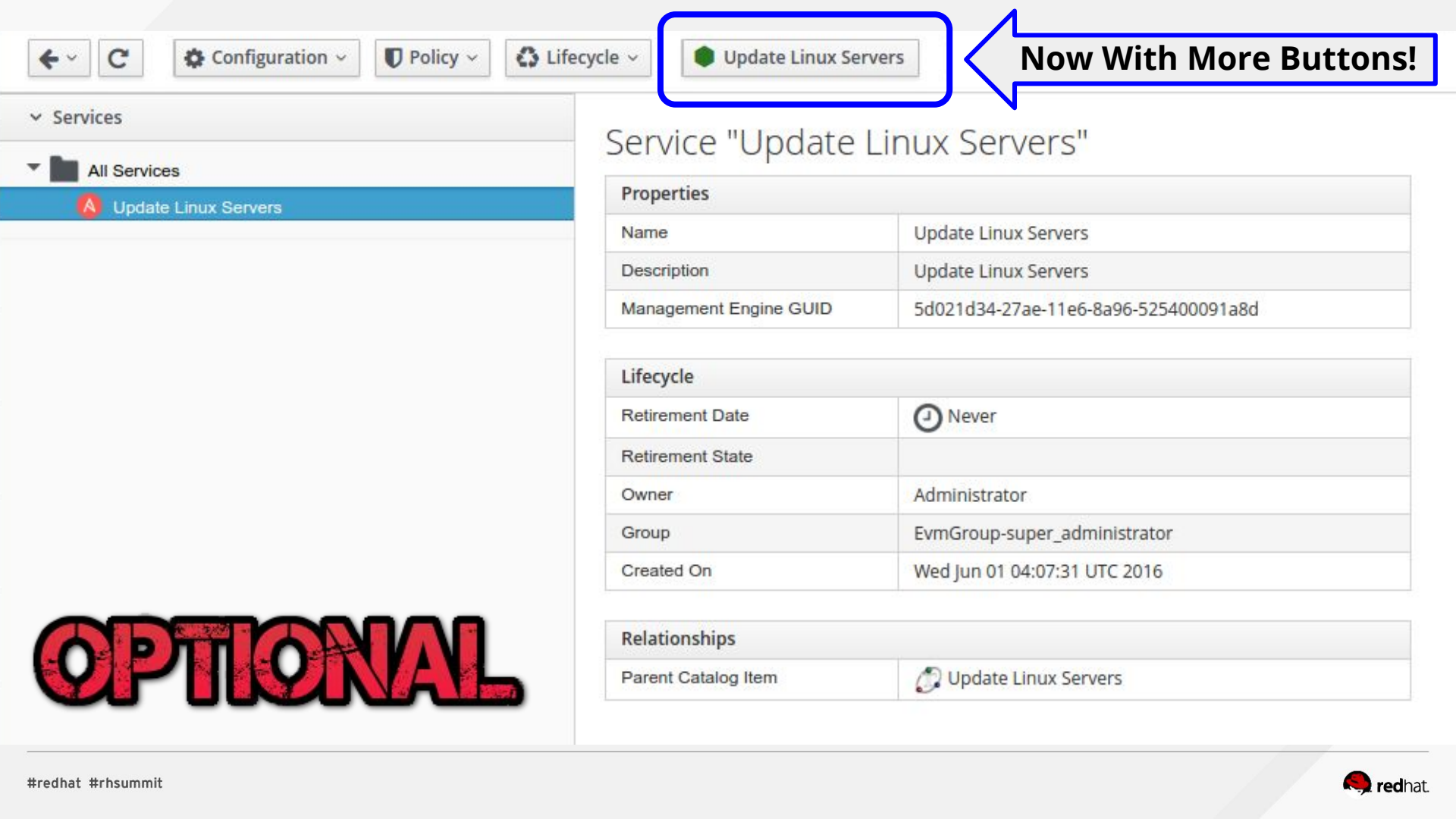
Name / Description	Update Linux Servers / Update Linux Servers Display in Catalog
Catalog	Ansible Demo
Dialog	Update Linux Servers
Ansible Tower Job Template	Update Linux Servers
Provisioning Entry Point State Machine (NS/Cls/Inst)	/ConfigurationManagement/AnsibleTower/Service/Provisioning/StateMachines/Provision/default

Custom Image

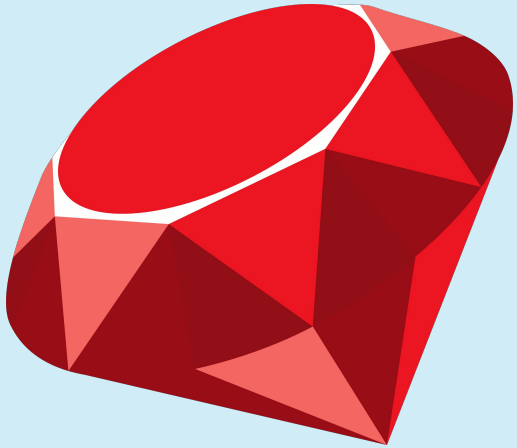
Add Sweet Custom Graphics

No file chosen | Choose file | Upload | * Requirements: File-type - PNG; Dimensions - 350x70.

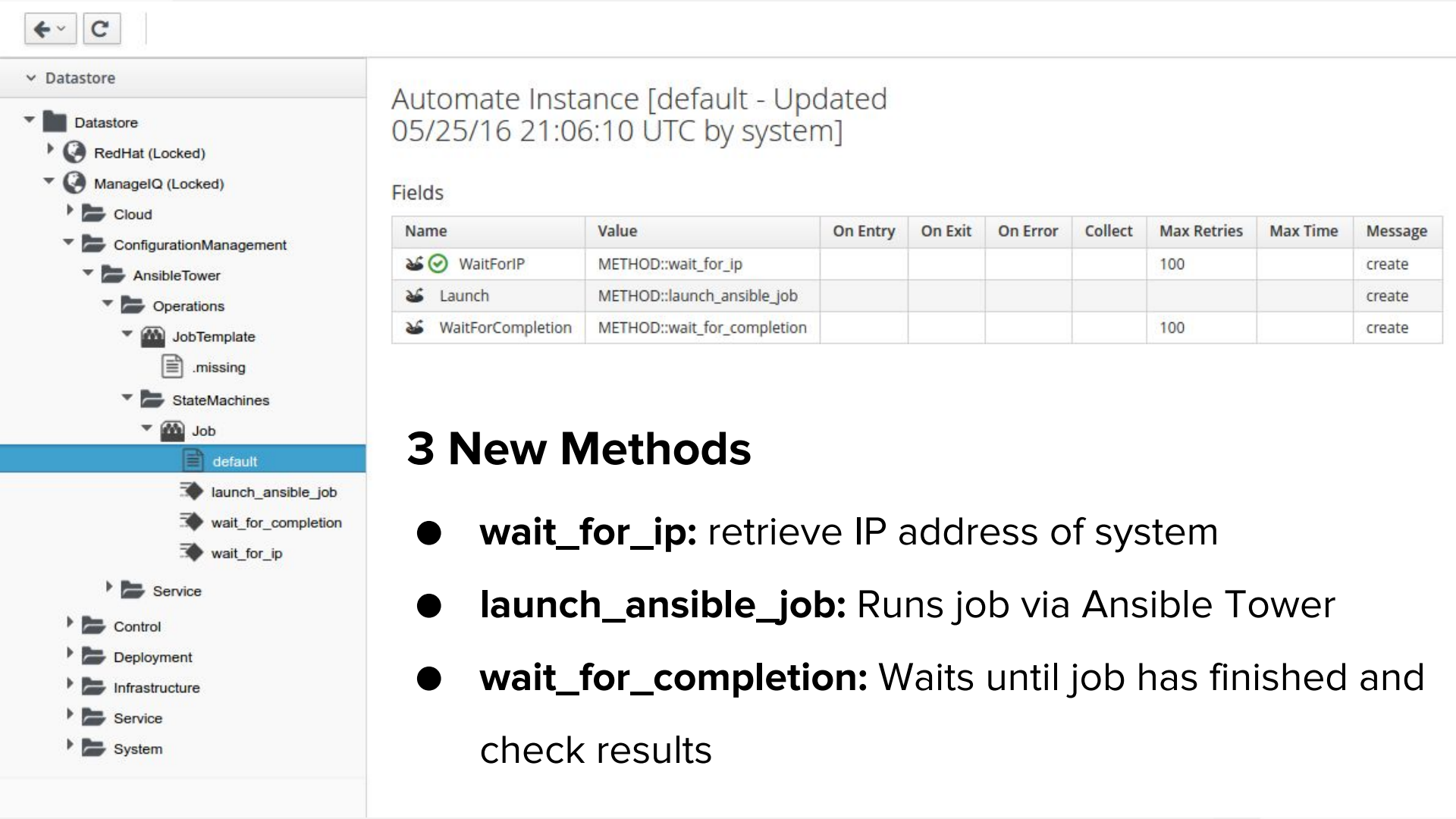
- Service Catalog
- Catalog Items
- All Catalog
- Unassigned
- Ansible Demo
  - Update Linux Servers
    - Actions
      - Update Linux Servers
- Orchestration Templates
- Catalogs


Configuration
Policy
Lifecycle
Update Linux Servers
Now With More Buttons!
Services
All Services
Update Linux Servers
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
Owner
Administrator
Group
EvmGroup-super_administrator
Created On
Wed Jun 01 04:07:31 UTC 2016
Relationships
Parent Catalog Item
Update Linux Servers
OPTIONAL

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

RED HAT® CLOUDFOF

ANSIBLE

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

Hard problems 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 Requests/sec and Bytes/sec of web servers

$ ansible webservers -m shell -a "apachectl status | grep Status"

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

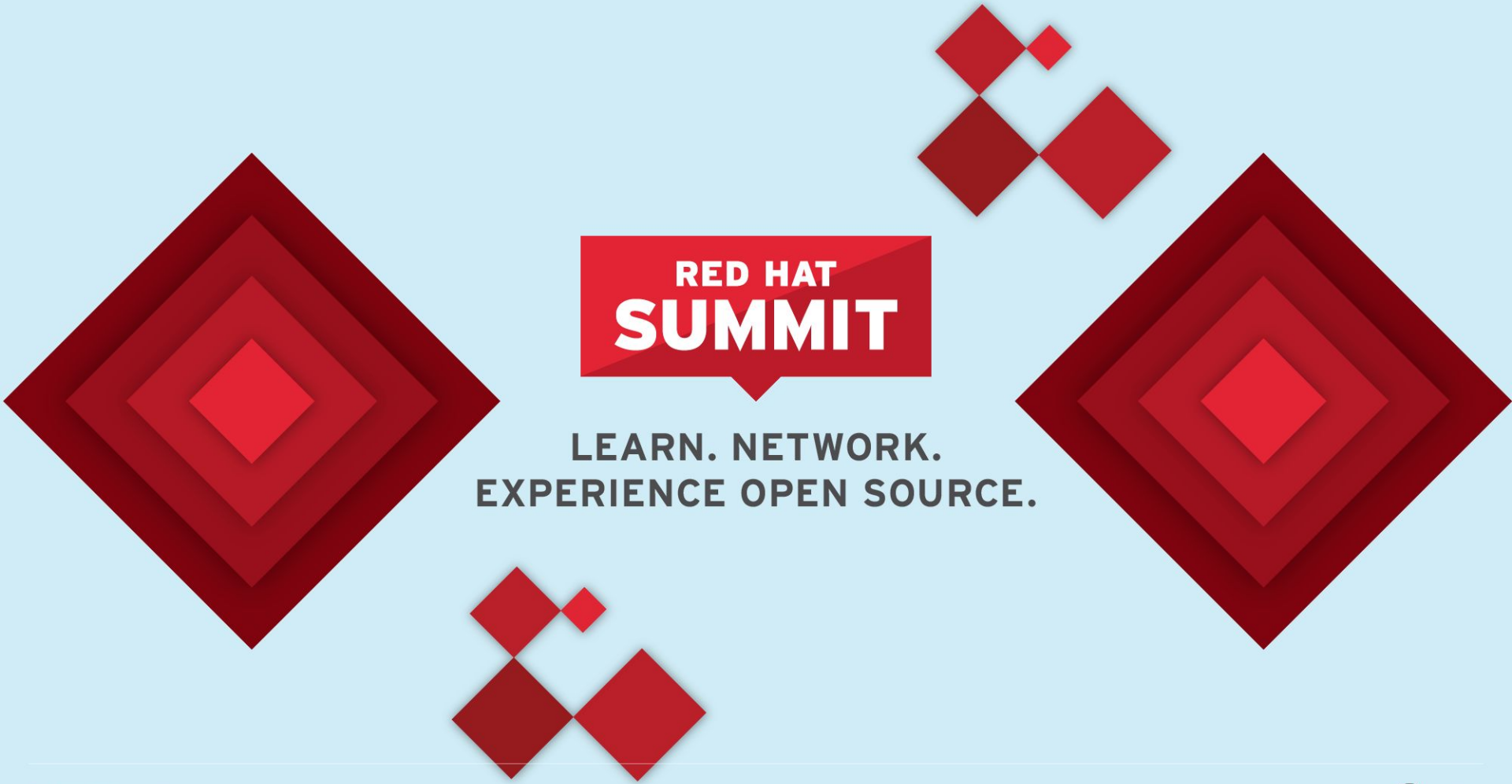
Hybrid Cloud Management Sessions

Compliance, security automation, and remediation with Red Hat CloudForms, Red Hat Satellite, and Ansible Tower by Red Hat	**Thurs, Jun 30, 3:30 PM - 4:30 PM – 2005**
Mastering CloudForms Automation - Book Signing with Peter McGowan	**Thurs, Jun 30, 11:15 AM - North Upper Lobby**
Red Hat CloudForms: Cutting VM creation time by 75% at General Mills	**Thurs, Jun 30, 10:15 AM - 11:15 AM – 2004**
Automation and configuration management across hybrid clouds with Red Hat CloudForms, Red Hat Satellite 6, and Ansible Tower	Wed, Jun 29, 4:45 PM - 5:45 PM – 2007
Automating Azure public and private clouds with Red Hat CloudForms 4	Wed, Jun 29, 4:45 PM - 5:45 PM – 2004
Red Hat CloudForms 2016 roadmap	Wed, Jun 29, 11:30 AM - 12:30 PM – 2004
Hands-on introduction to Red Hat CloudForms	Wed, Jun 29, 10:15 AM - 12:15 PM – 3016 - Lab II

QUESTIONS?

THANK YOU

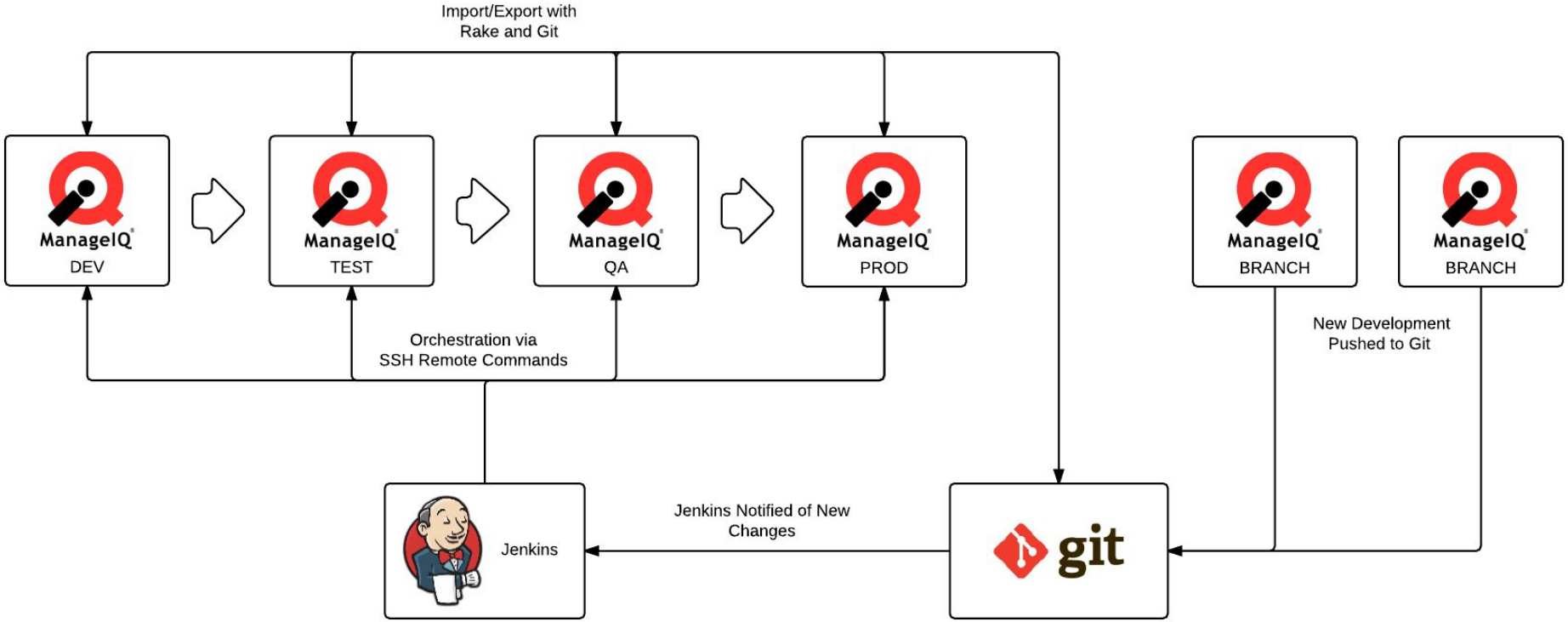
Contact info:
laurent@redhat.com
miked@redhat.com



References

- https://github.com/RedHatEMEA/soe-ci (Satellite 6 CI/CD)
- https://github.com/rhtconsulting/miq-ci (CloudForms CI/CD)
- https://github.com/ldomb (puppet modules + cloud init)
- http://blog.domb.net (demo videos)

CloudForms CI/CD pipeline



By Cameron Wyatt @ Red Hat