RED HAT SUMMIT

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

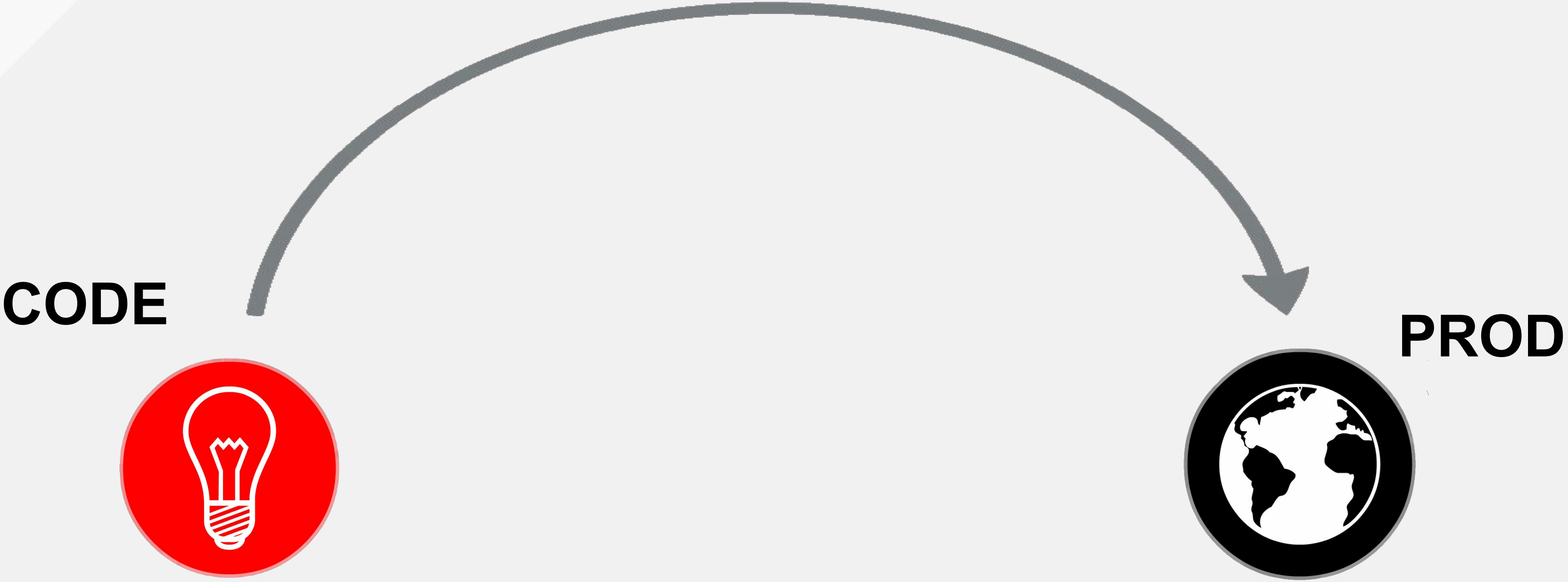
RHCA V

Red Hat

Intro

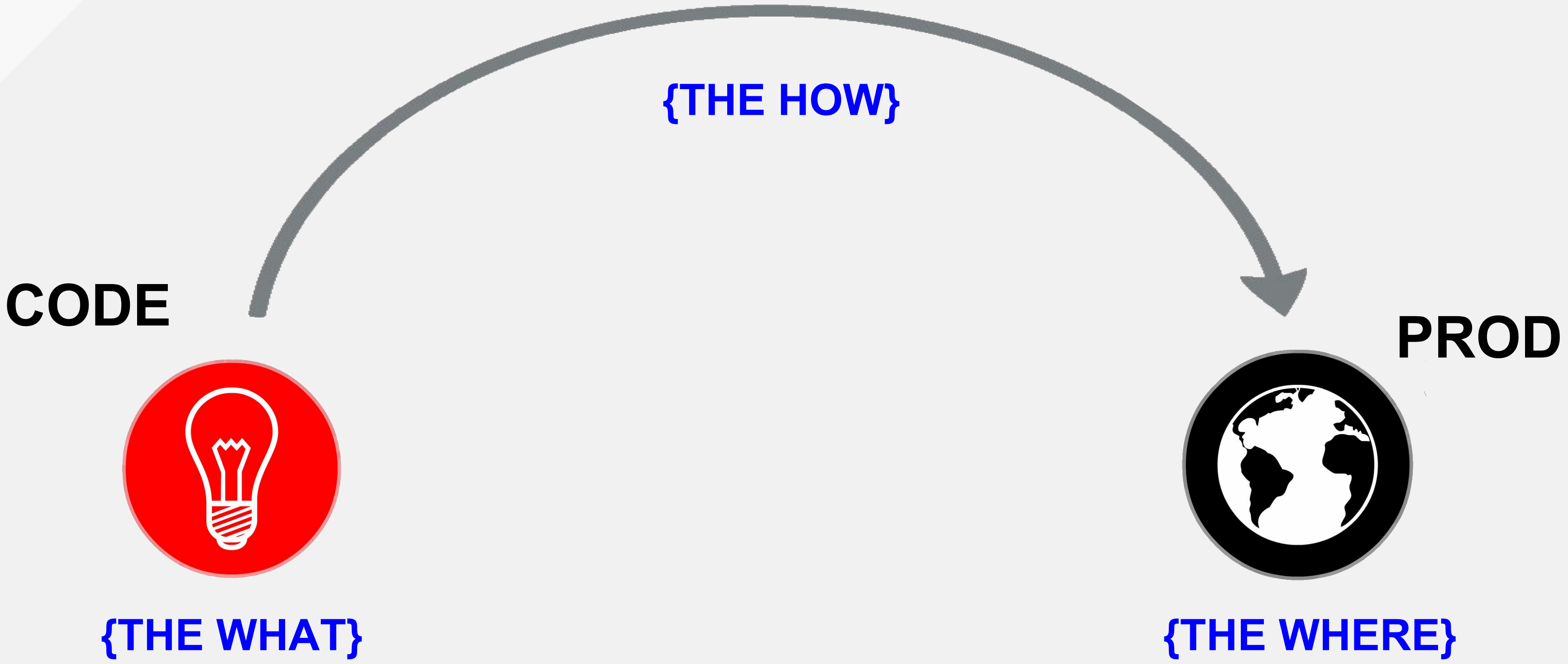
Ansible Tower, CloudForms, Insights, Satellite 6

- Why do we care?
- What problems are we solving?
- How do the Red Hat tools address this?
- What does it look like in action?



CODE

PROD



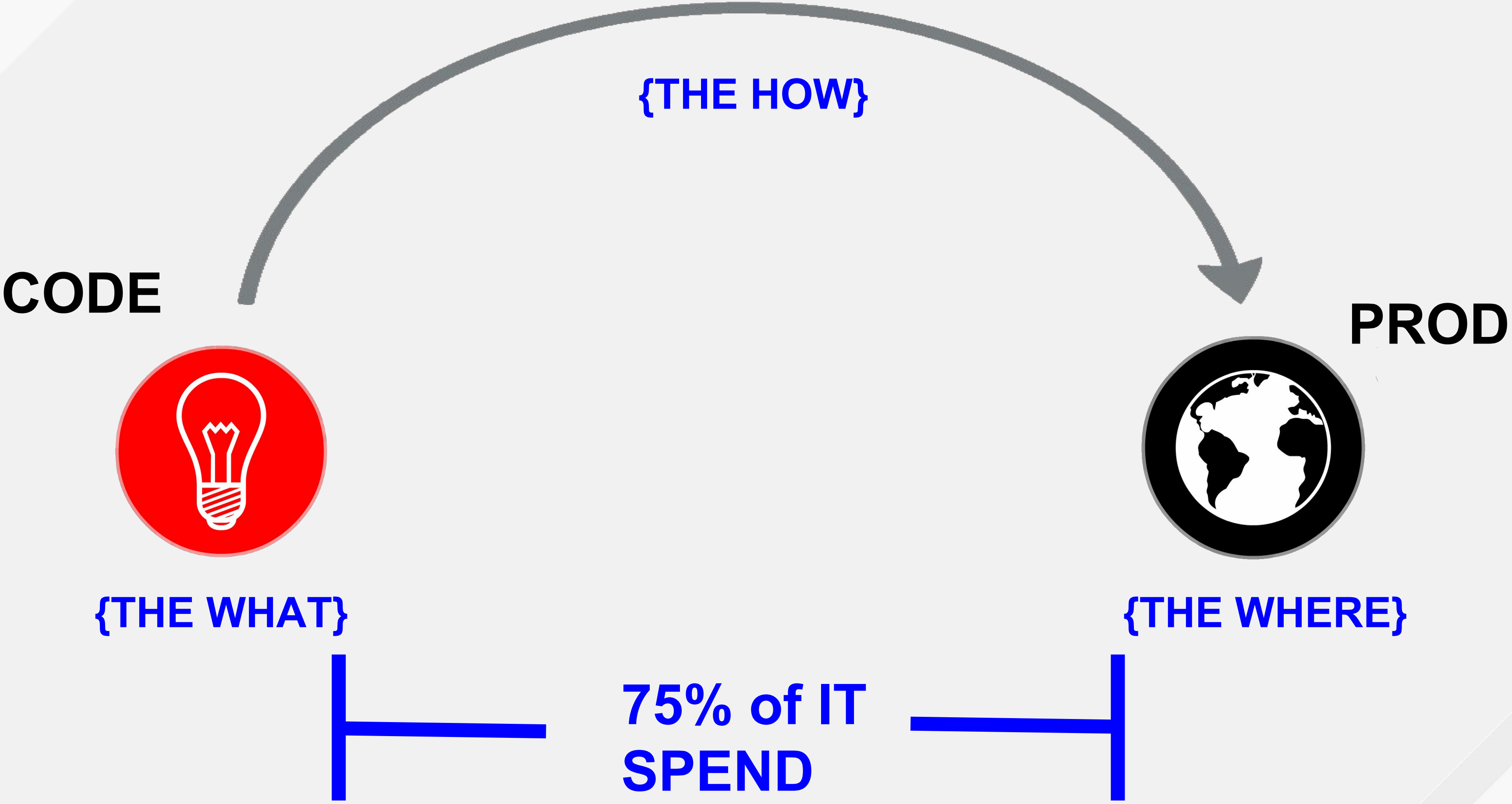
CODE

{THE HOW}

PROD

{THE WHAT}

{THE WHERE}



CODE

{THE HOW}

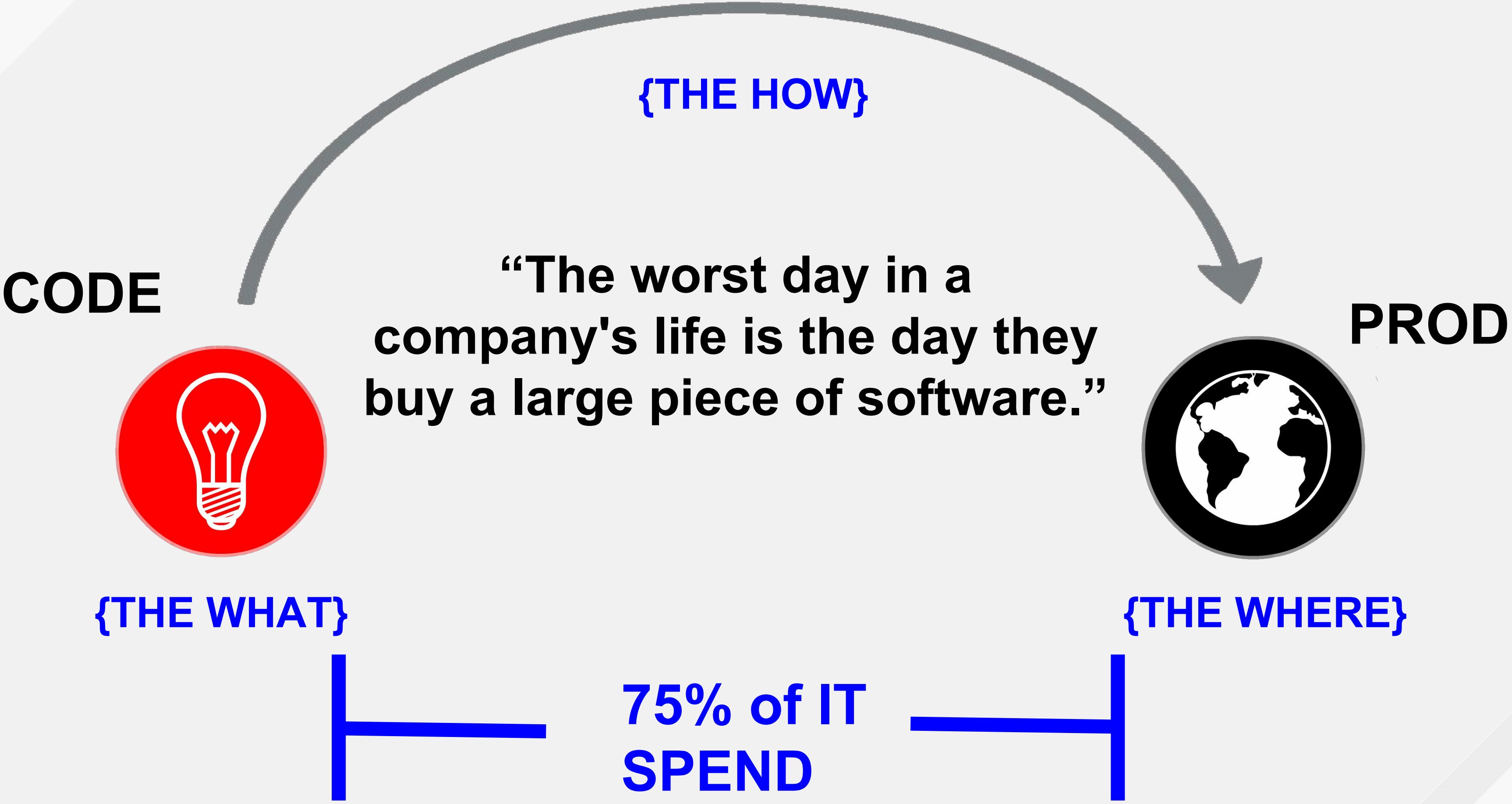
PROD

{THE WHAT}

{THE WHERE}

75% of IT SPEND

$ TRILLIONS PER YEAR



{THE HOW}

CODE

“The worst day in a company's life is the day they buy a large piece of software.”

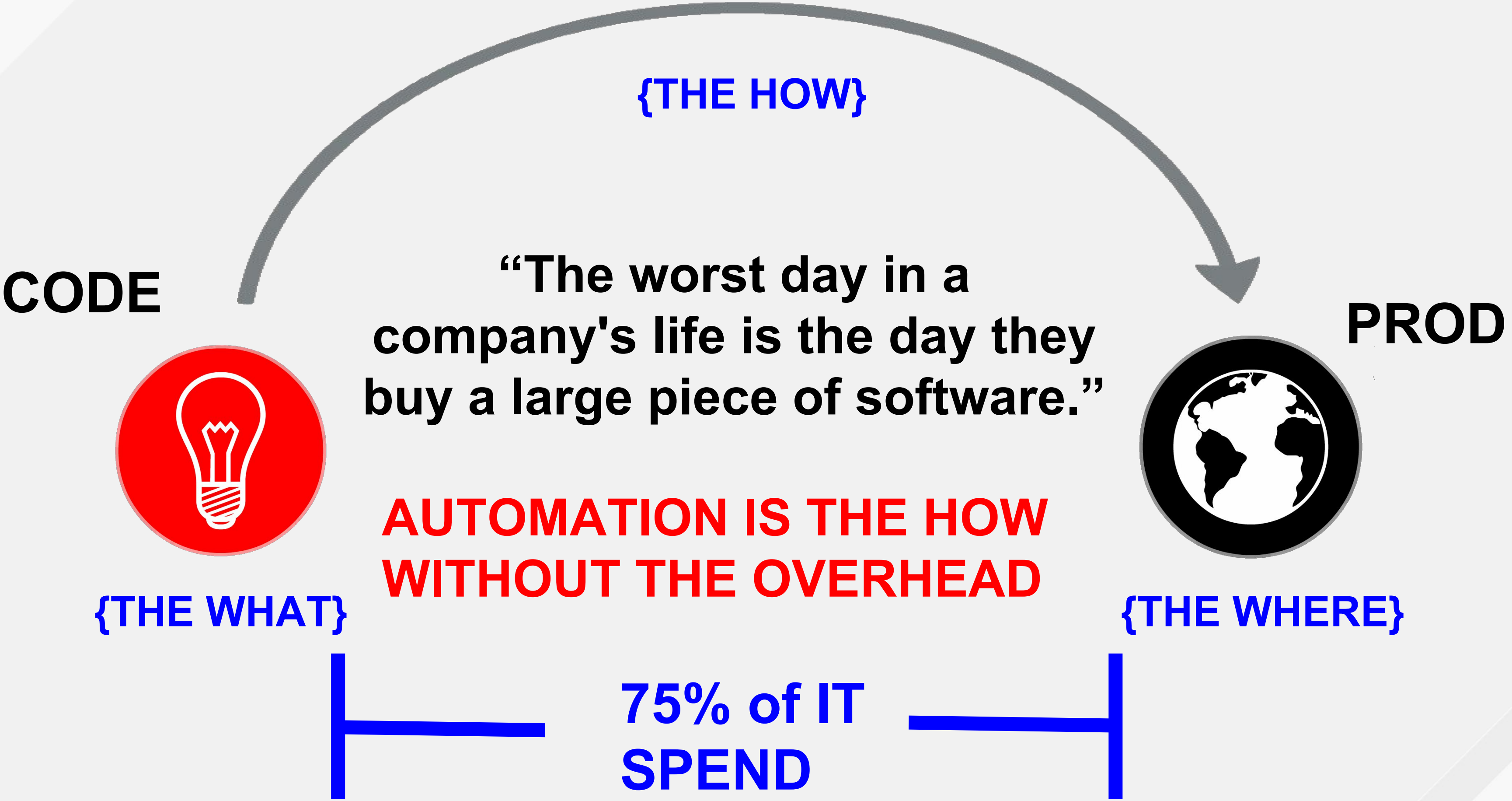
PROD

{THE WHAT}

{THE WHERE}

75% of IT SPEND

$ TRILLIONS PER YEAR



CODE

{THE HOW}

PROD

“The worst day in a company's life is the day they buy a large piece of software.”

AUTOMATION IS THE HOW WITHOUT THE OVERHEAD

{THE WHAT}

{THE WHERE}

75% of IT SPEND

$ 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

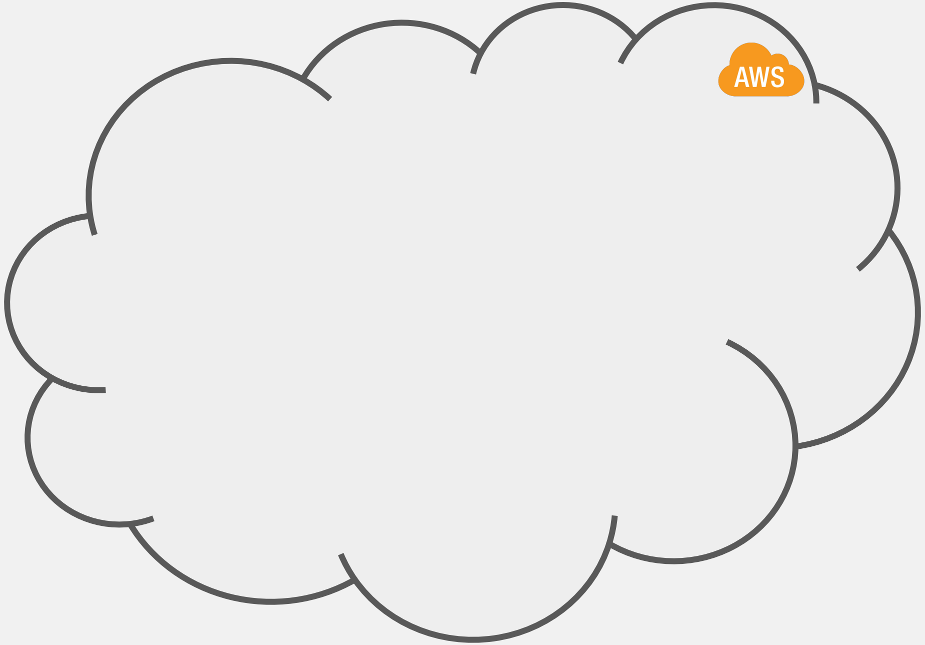
From start to finish in less than 3 hours with these ingredients

```
summit2017$ ansible-playbook buildrhmgmt.yaml --private-key=ldomb.pem
--vault-password-file=../vaultpass -vv
```

RH-MANAGEMENT CORE

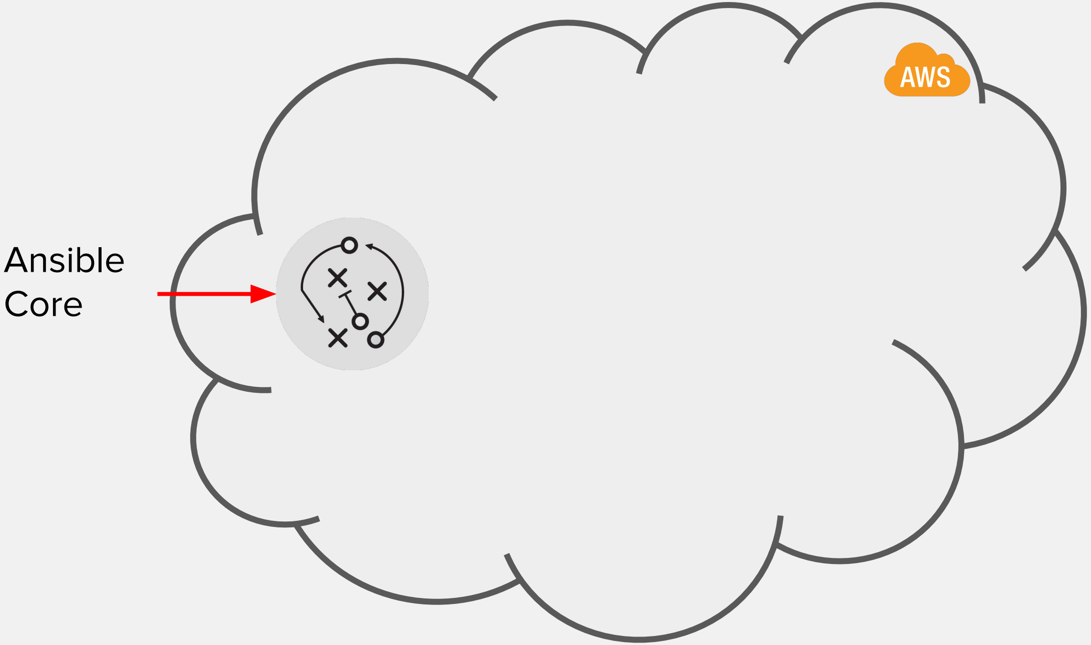
One click to rule them all - Foundation Installation Flow

Ansible
Core



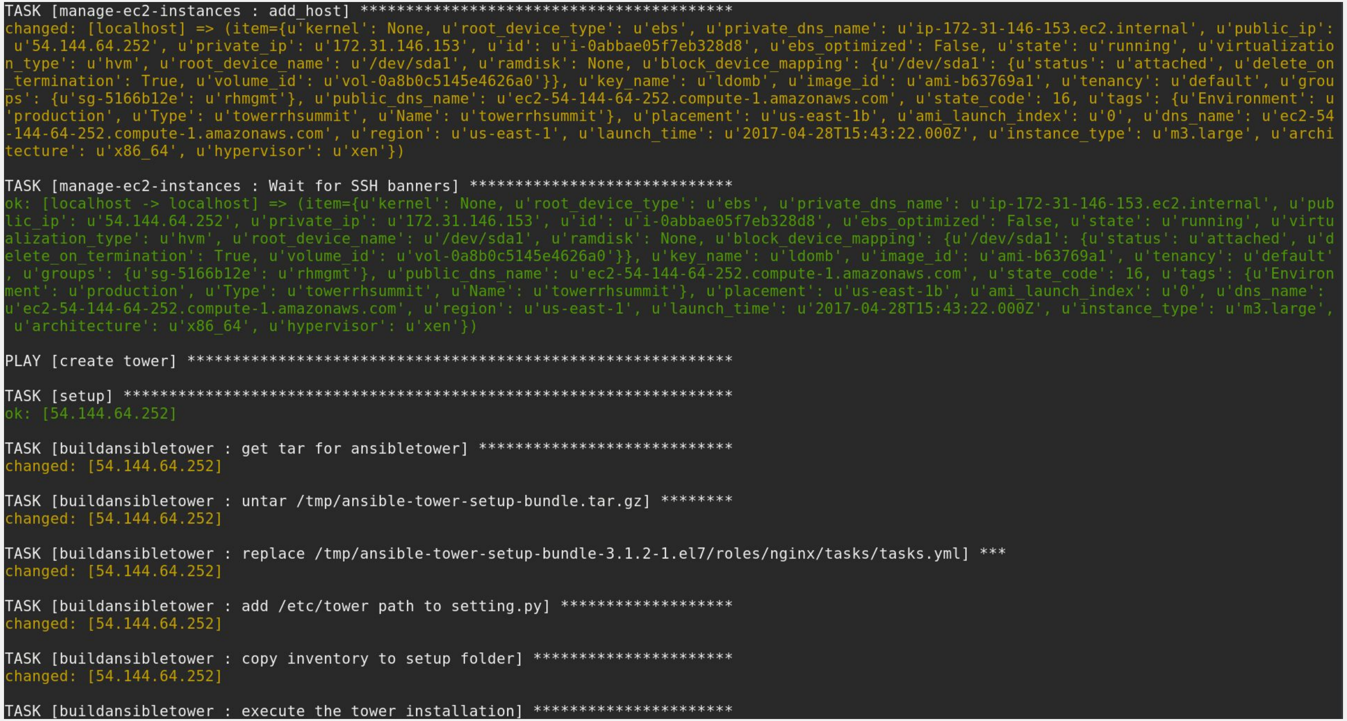
RH-MANAGEMENT ANSIBLE TOWER

One click to rule them all - Foundation Installation Flow



RH-MANAGEMENT TOOLS

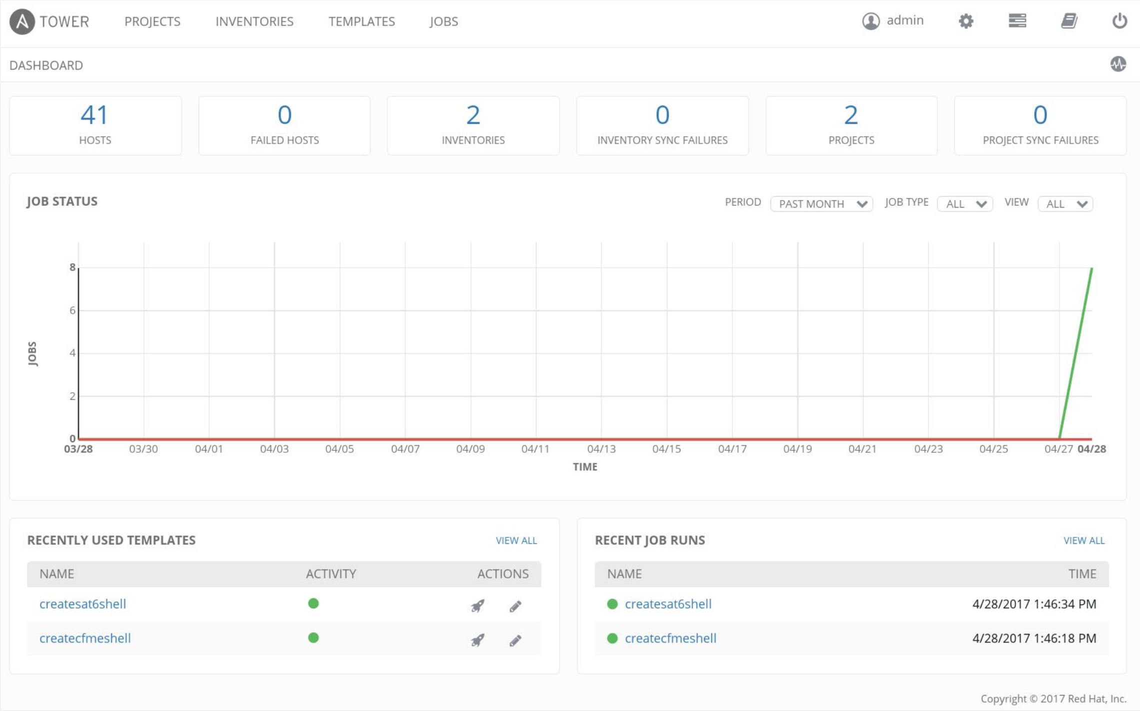
One click to rule them all - Foundation Installation Flow



```
TASK [manage-ec2-instances : add_host] *****************************************
changed: [localhost] => (item={u'kernel': None, u'root_device_type': u'ebs', u'private_dns_name': u'ip-172-31-146-153.ec2.internal', u'public_ip':
 u'54.144.64.252', u'private_ip': u'172.31.146.153', u'id': u'i-0abbae05f7eb328d8', u'ebs_optimized': False, u'state': u'running', u'virtualizatio
n_type': u'hvm', u'root_device_name': u'/dev/sda1', u'ramdisk': None, u'block_device_mapping': {u'/dev/sda1': {u'status': u'attached', u'delete_on
_termination': True, u'volume_id': u'vol-0a8b0c5145e4626a0'}}, u'key_name': u'ldomb', u'image_id': u'ami-b63769a1', u'tenancy': u'default', u'grou
ps': {u'sg-5166b12e': u'rhmgmt'}, u'public_dns_name': u'ec2-54-144-64-252.compute-1.amazonaws.com', u'state_code': 16, u'tags': {u'Environment': u
'production', u'Type': u'towerrhsummit', u'Name': u'towerrhsummit'}, u'placement': u'us-east-1b', u'ami_launch_index': u'0', u'dns_name': u'ec2-54
-144-64-252.compute-1.amazonaws.com', u'region': u'us-east-1', u'launch_time': u'2017-04-28T15:43:22.000Z', u'instance_type': u'm3.large', u'archi
tecture': u'x86_64', u'hypervisor': u'xen'})

TASK [manage-ec2-instances : Wait for SSH banners] *****************************
ok: [localhost -> localhost] => (item={u'kernel': None, u'root_device_type': u'ebs', u'private_dns_name': u'ip-172-31-146-153.ec2.internal', u'pub
lic_ip': u'54.144.64.252', u'private_ip': u'172.31.146.153', u'id': u'i-0abbae05f7eb328d8', u'ebs_optimized': False, u'state': u'running', u'virtu
alization_type': u'hvm', u'root_device_name': u'/dev/sda1', u'ramdisk': None, u'block_device_mapping': {u'/dev/sda1': {u'status': u'attached', u'd
elete_on_termination': True, u'volume_id': u'vol-0a8b0c5145e4626a0'}}, u'key_name': u'ldomb', u'image_id': u'ami-b63769a1', u'tenancy': u'default'
, u'groups': {u'sg-5166b12e': u'rhmgmt'}, u'public_dns_name': u'ec2-54-144-64-252.compute-1.amazonaws.com', u'state_code': 16, u'tags': {u'Environ
ment': u'production', u'Type': u'towerrhsummit', u'Name': u'towerrhsummit'}, u'placement': u'us-east-1b', u'ami_launch_index': u'0', u'dns_name':
u'ec2-54-144-64-252.compute-1.amazonaws.com', u'region': u'us-east-1', u'launch_time': u'2017-04-28T15:43:22.000Z', u'instance_type': u'm3.large',
 u'architecture': u'x86_64', u'hypervisor': u'xen'})

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

TASK [setup] *******************************************************************
ok: [54.144.64.252]

TASK [buildansibletower : get tar for ansibletower] ****************************
changed: [54.144.64.252]

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

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

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

TASK [buildansibletower : copy inventory to setup folder] **********************
changed: [54.144.64.252]

TASK [buildansibletower : execute the tower installation] **********************
```

Red Hat Management Automated

Building Ansible Tower - Gains



41	0	2	0	2	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH JOB TYPE: ALL VIEW: ALL

RECENTLY USED TEMPLATES

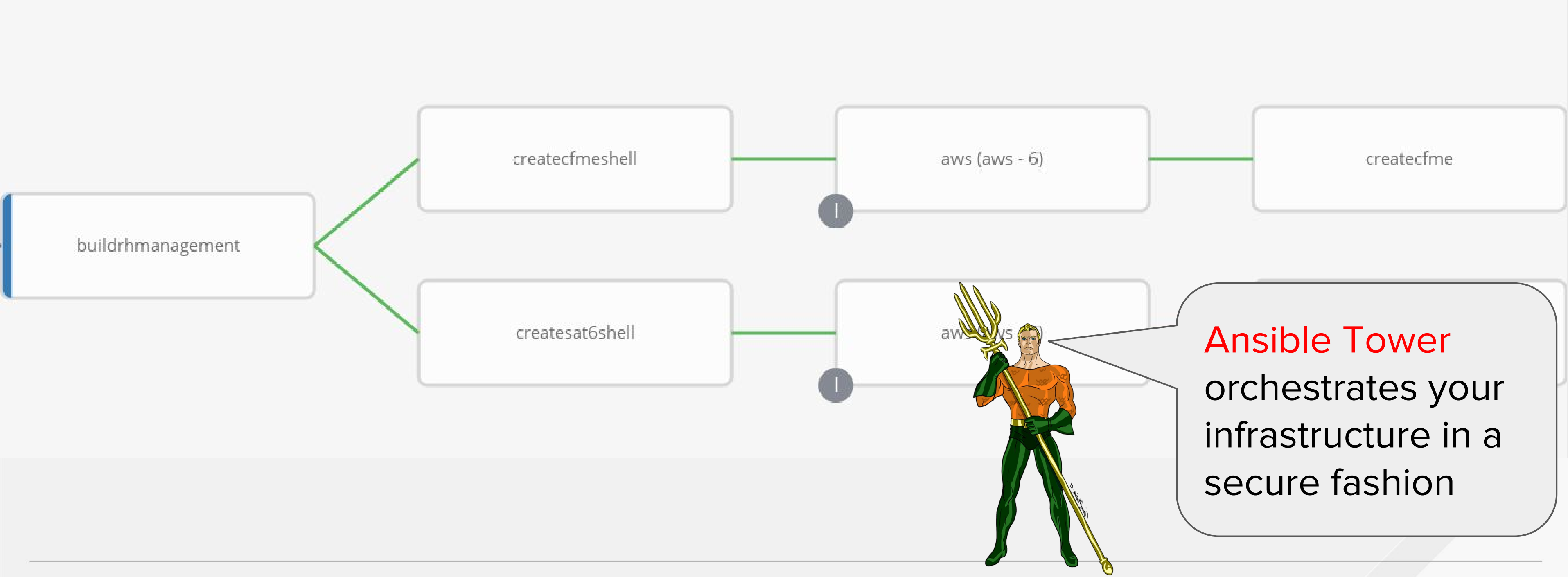
NAME	ACTIVITY	ACTIONS
createsat6shell	●	
createcfmeshell	●	

RECENT JOB RUNS

NAME	TIME
● createsat6shell	4/28/2017 1:46:34 PM
● createcfmeshell	4/28/2017 1:46:18 PM

Red Hat Management Automated

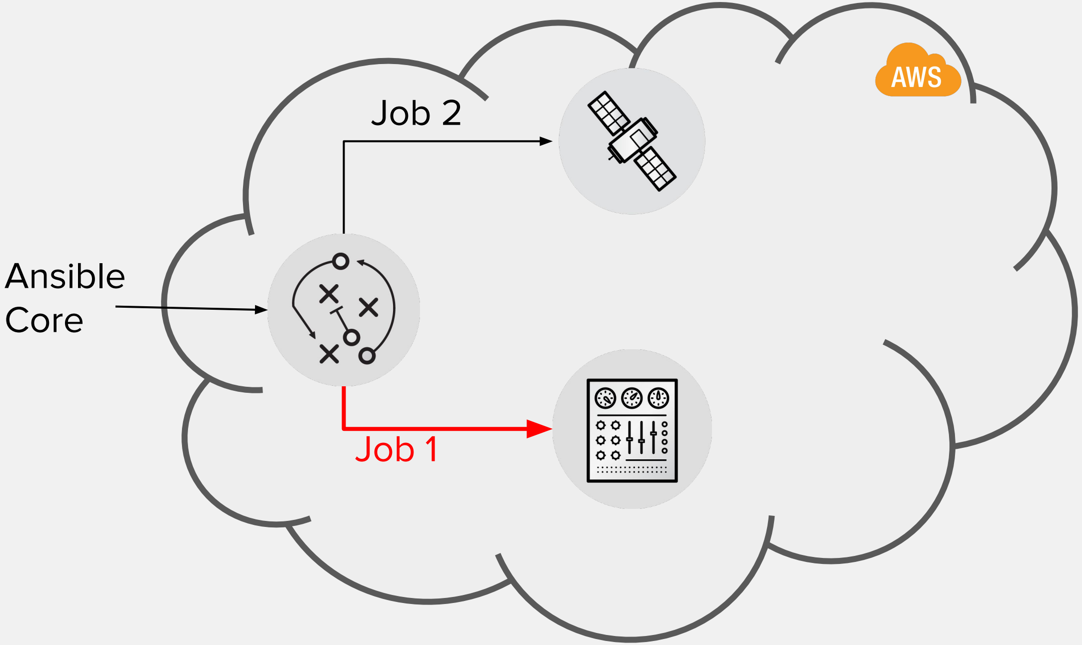
Building Ansible Tower - Gains - Workflow Editor



Ansible Tower orchestrates your infrastructure in a secure fashion

RH-MANAGEMENT SATELLITE 6 / CLOUDFORMS

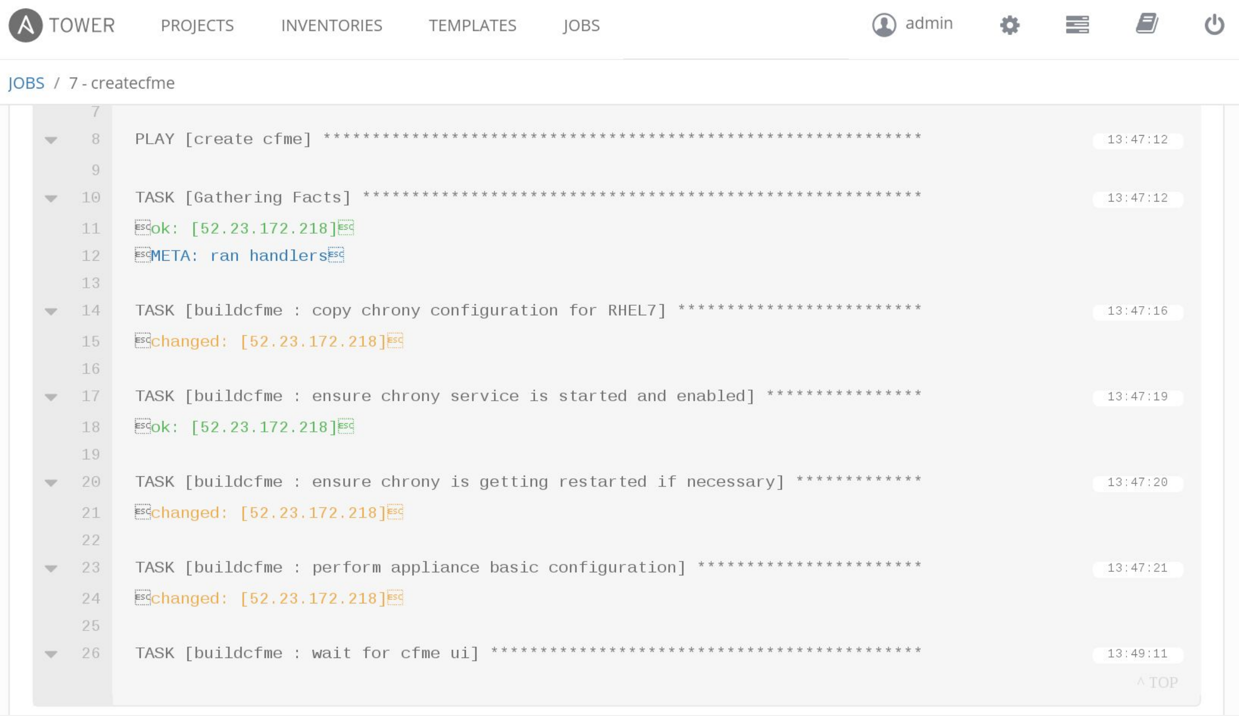
One click to rule them all - Foundation Installation Flow



RH-MANAGEMENT TOOLS

One click to rule them all - Foundation Installation Flow

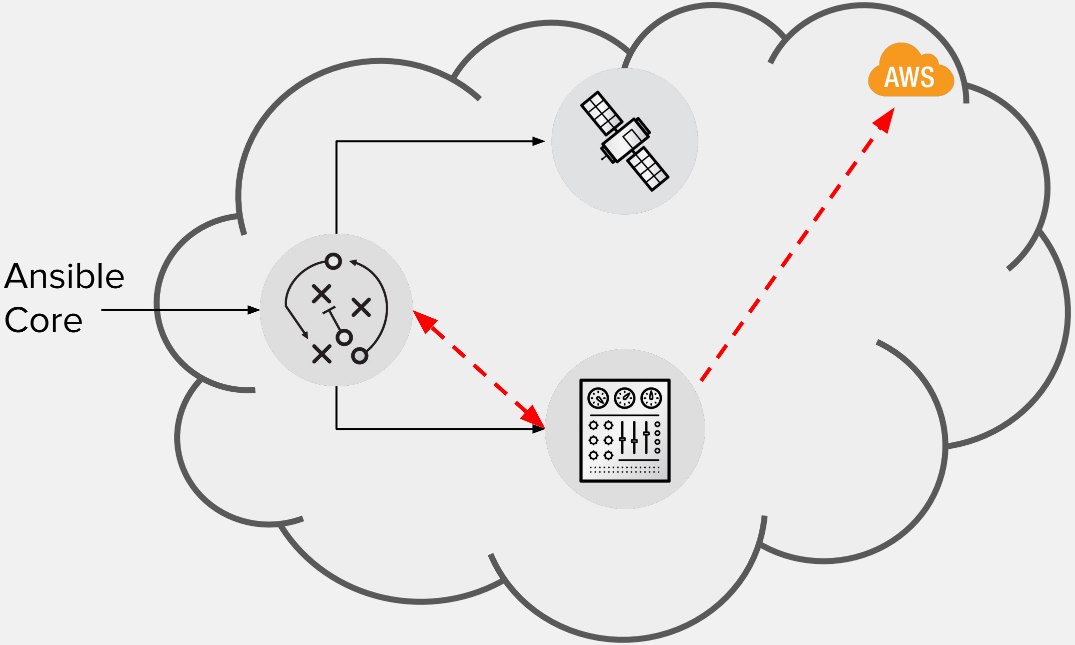
Job 1



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

TASK [Gathering Facts] *********************************************************  13:47:12
␛ok: [52.23.172.218]␛
␛META: ran handlers␛

TASK [buildcfme : copy chrony configuration for RHEL7] *************************  13:47:16
␛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]␛

TASK [buildcfme : wait for cfme ui] ********************************************  13:49:11
```

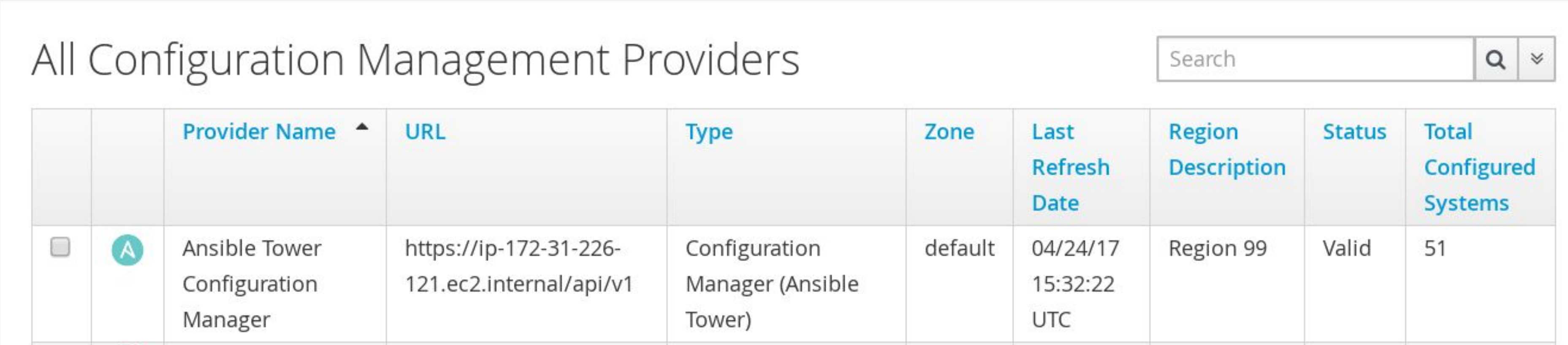
RH-MANAGEMENT CLOUDFORMS

One click to rule them all - Foundation Installation Flow



Red Hat Management Automated

CloudForms - Ansible Tower integration gains

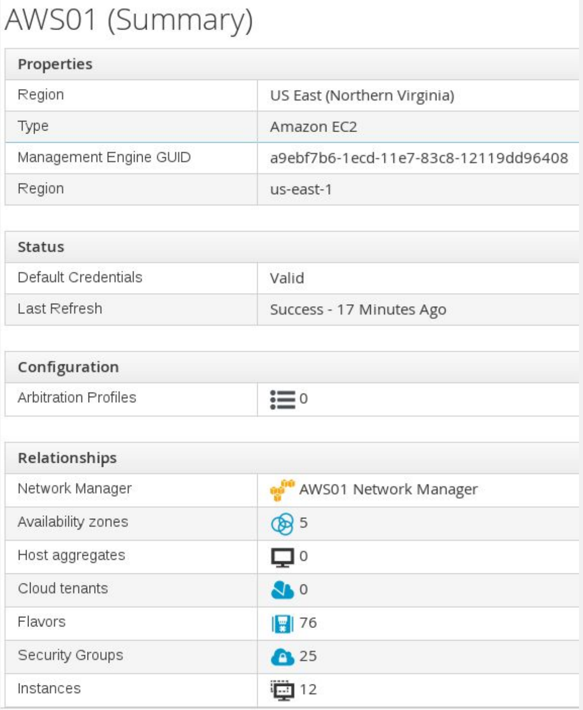


All Configuration Management Providers

		Provider Name ▲	URL	Type	Zone	Last Refresh Date	Region Description	Status	Total Configured Systems
□	A	Ansible Tower Configuration Manager	https://ip-172-31-226-121.ec2.internal/api/v1	Configuration Manager (Ansible Tower)	default	04/24/17 15:32:22 UTC	Region 99	Valid	51

Red Hat Management Automated

CloudForms - Ansible Tower integration gains



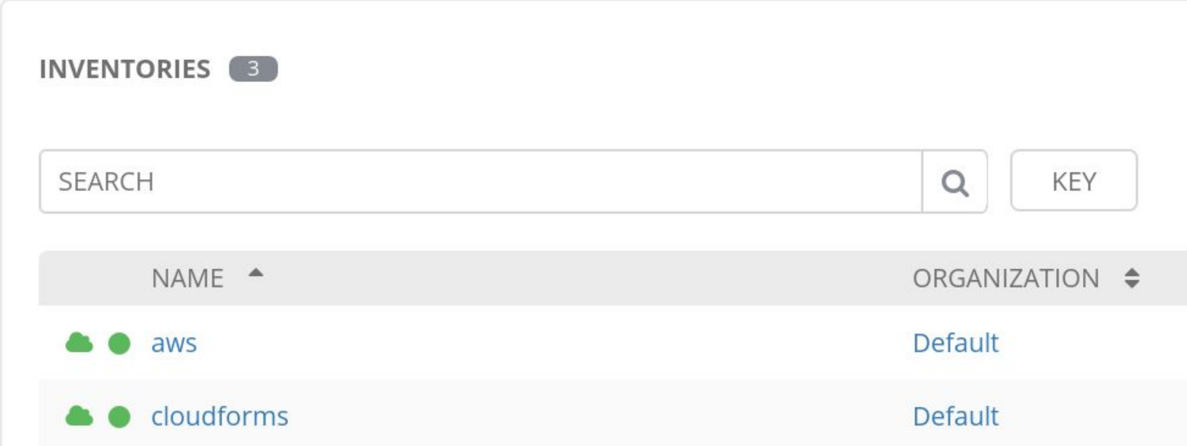
AWS01 (Summary)

Properties	
Region	US East (Northern Virginia)
Type	Amazon EC2
Management Engine GUID	a9ebf7b6-1ecd-11e7-83c8-12119dd96408
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

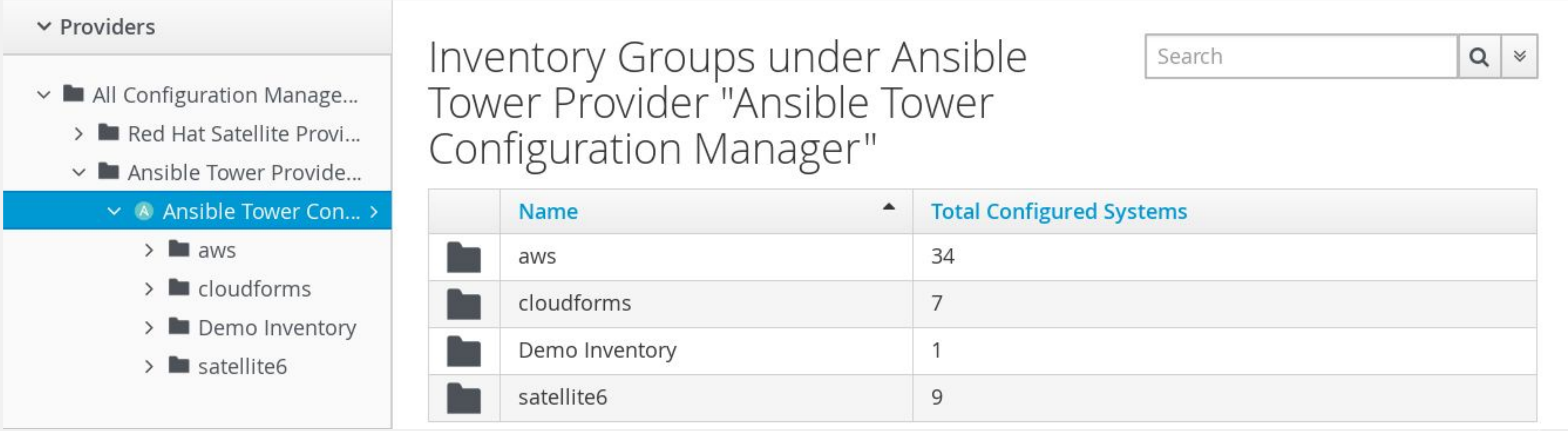


INVENTORIES 3

NAME	ORGANIZATION
aws	Default
cloudforms	Default

Red Hat Management Automated

CloudForms - Ansible Tower integration gains

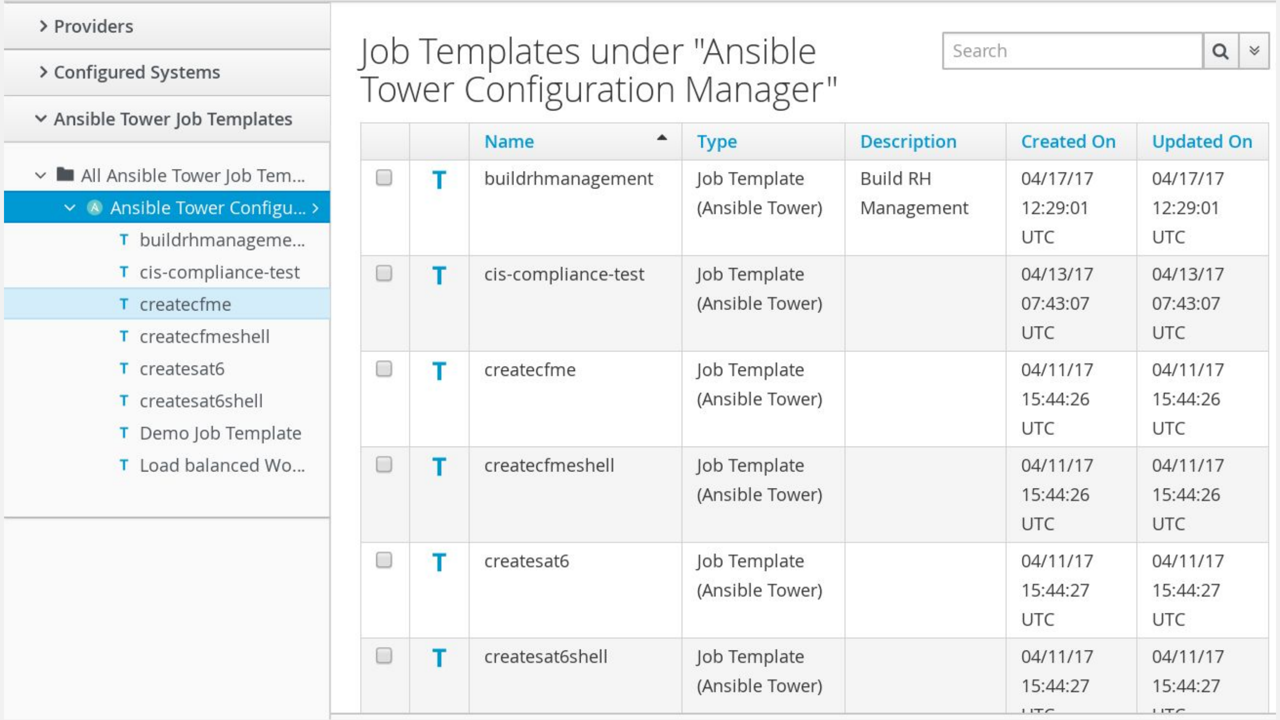


Inventory Groups under Ansible Tower Provider "Ansible Tower Configuration Manager"

	Name	Total Configured Systems
	aws	34
	cloudforms	7
	Demo Inventory	1
	satellite6	9

Red Hat Management Automated

CloudForms - Ansible Tower integration gains

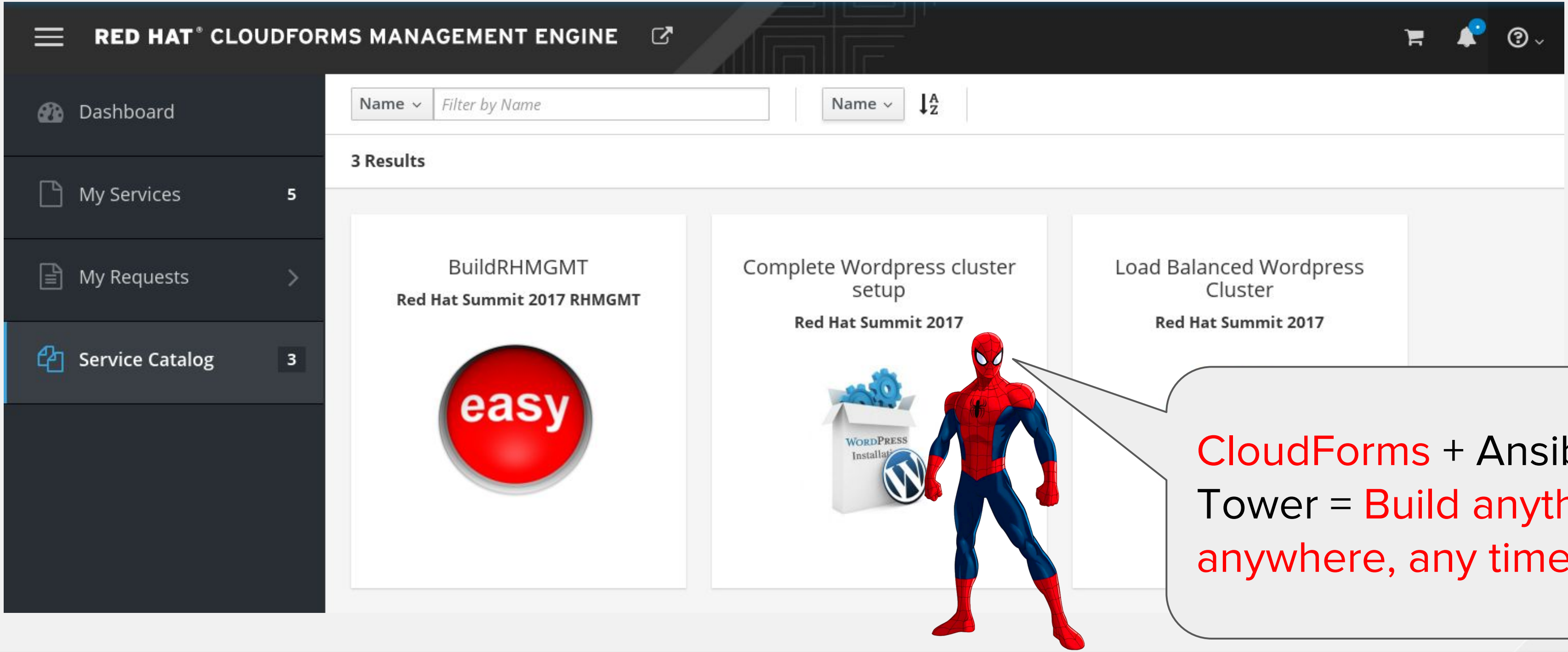


Job Templates under "Ansible Tower Configuration Manager"

		Name	Type	Description	Created On	Updated On
□	T	buildrhmanagement	Job Template (Ansible Tower)	Build RH Management	04/17/17 12:29:01 UTC	04/17/17 12:29:01 UTC
□	T	cis-compliance-test	Job Template (Ansible Tower)		04/13/17 07:43:07 UTC	04/13/17 07:43:07 UTC
□	T	createcfme	Job Template (Ansible Tower)		04/11/17 15:44:26 UTC	04/11/17 15:44:26 UTC
□	T	createcfmeshell	Job Template (Ansible Tower)		04/11/17 15:44:26 UTC	04/11/17 15:44:26 UTC
□	T	createsat6	Job Template (Ansible Tower)		04/11/17 15:44:27 UTC	04/11/17 15:44:27 UTC
□	T	createsat6shell	Job Template (Ansible Tower)		04/11/17 15:44:27 UTC	04/11/17 15:44:27 UTC

Red Hat Management Automated

CloudForms - Ansible Tower integration gains



RED HAT® CLOUDFORMS MANAGEMENT ENGINE

- Dashboard
- My Services 5
- My Requests
- Service Catalog 3

Name ⌄ | Filter by Name | Name ⌄

3 Results

BuildRHMGMT
Red Hat Summit 2017 RHMGMT

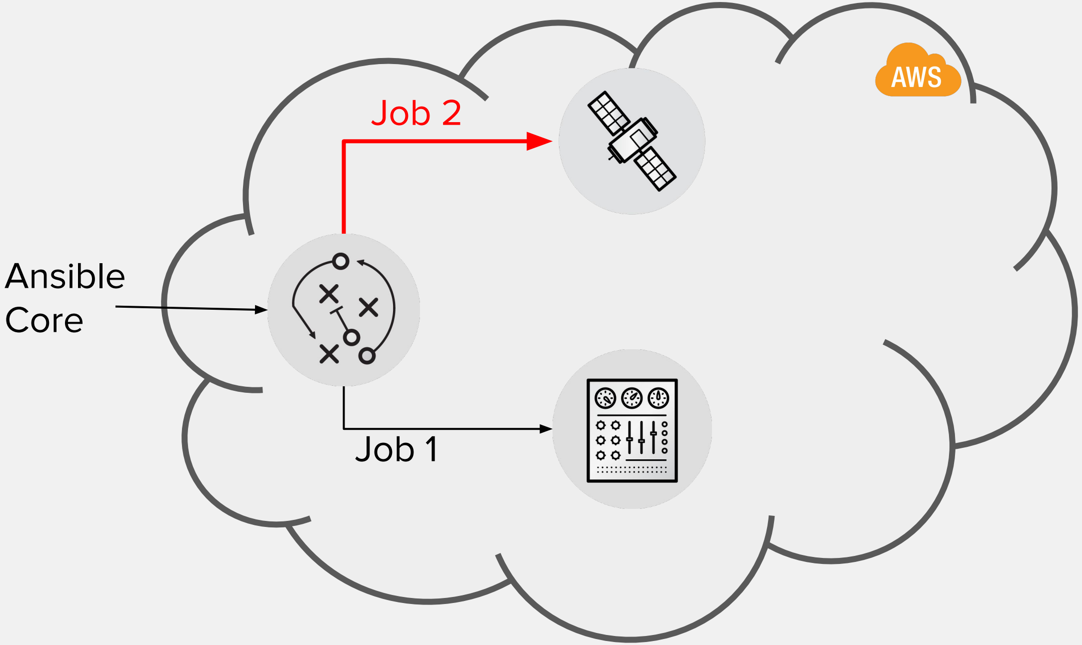
Complete Wordpress cluster setup
Red Hat Summit 2017

Load Balanced Wordpress Cluster
Red Hat Summit 2017

CloudForms + Ansible Tower = Build anything anywhere, any time

RH-MANAGEMENT SATELLITE 6 / CLOUDFORMS

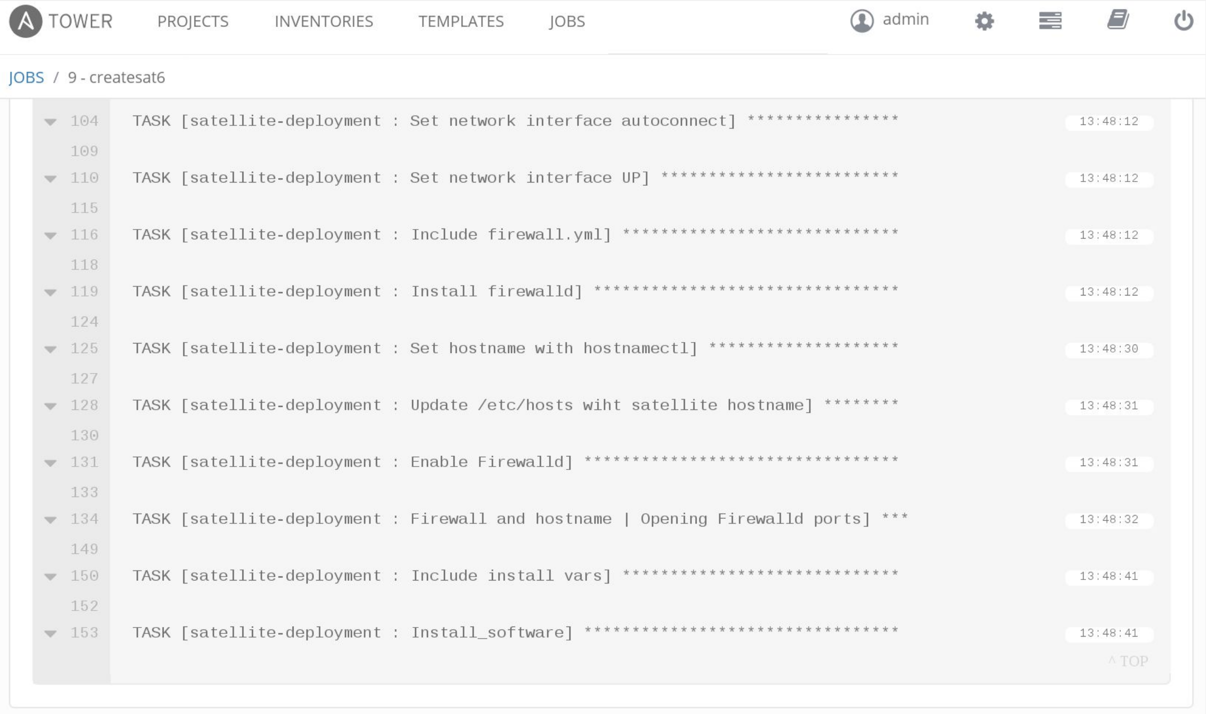
One click to rule them all - Foundation Installation Flow



RH-MANAGEMENT TOOLS

One click to rule them all - Foundation Installation Flow

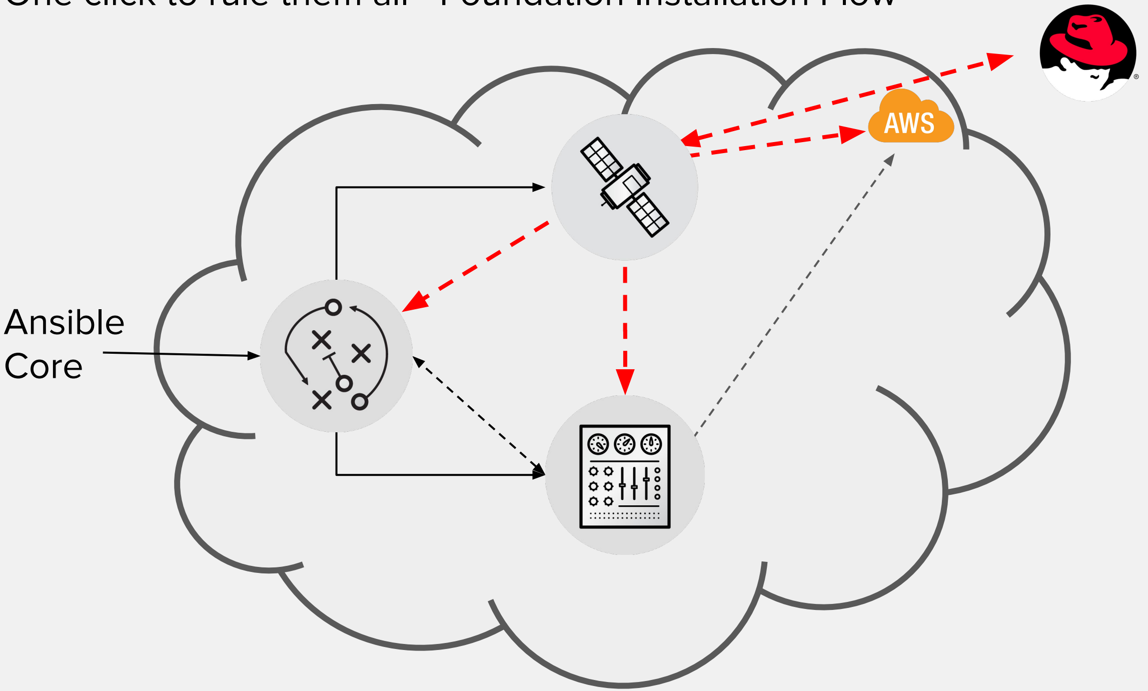
Job 2



```
TASK [satellite-deployment : Set network interface autoconnect] ****************    13:48:12

TASK [satellite-deployment : Set network interface UP] *************************    13:48:12

TASK [satellite-deployment : Include firewall.yml] *****************************    13:48:12

TASK [satellite-deployment : Install firewalld] ********************************    13:48:12

TASK [satellite-deployment : Set hostname with hostnamectl] ********************    13:48:30

TASK [satellite-deployment : Update /etc/hosts wiht satellite hostname] ********    13:48:31

TASK [satellite-deployment : Enable Firewalld] *********************************    13:48:31

TASK [satellite-deployment : Firewall and hostname | Opening Firewalld ports] ***   13:48:32

TASK [satellite-deployment : Include install vars] *****************************    13:48:41

TASK [satellite-deployment : Install_software] *********************************    13:48:41
```

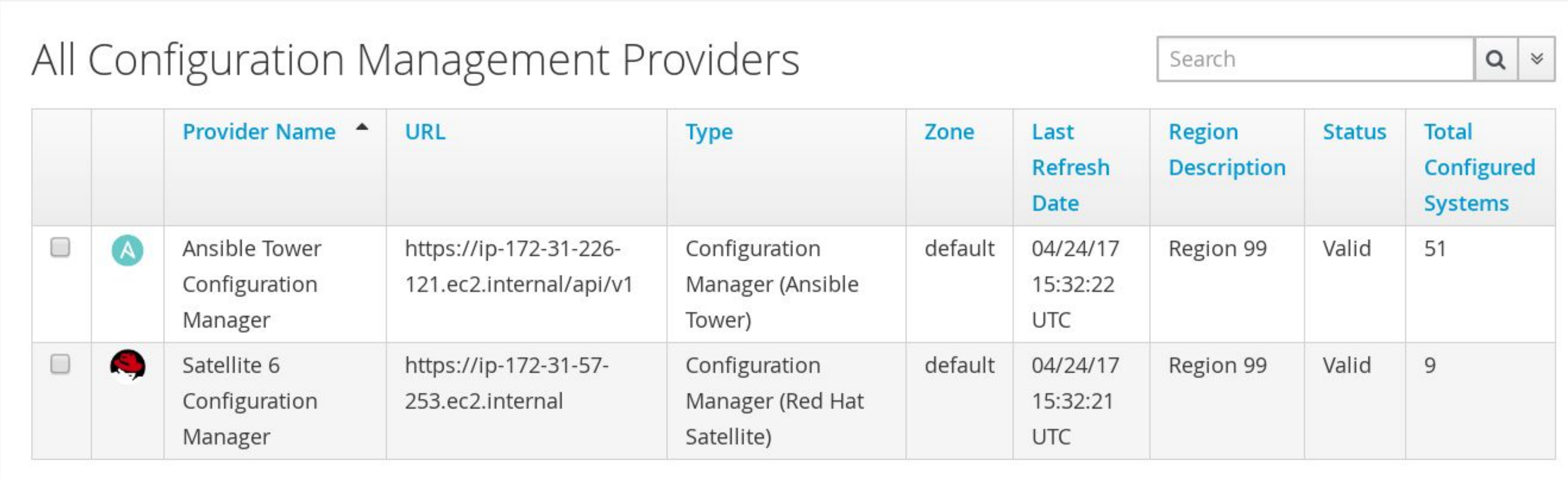
RH-MANAGEMENT SATELLITE 6

One click to rule them all - Foundation Installation Flow



Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains

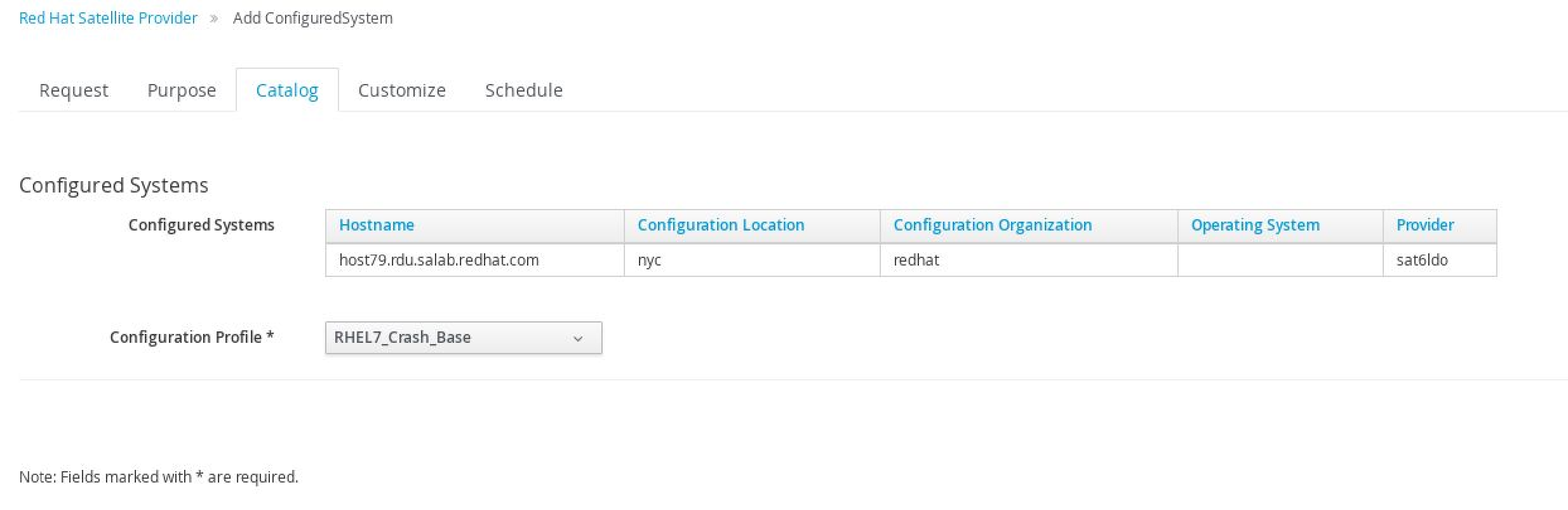


All Configuration Management Providers

		Provider Name	URL	Type	Zone	Last Refresh Date	Region Description	Status	Total Configured Systems
□		Ansible Tower Configuration Manager	https://ip-172-31-226-121.ec2.internal/api/v1	Configuration Manager (Ansible Tower)	default	04/24/17 15:32:22 UTC	Region 99	Valid	51
□		Satellite 6 Configuration Manager	https://ip-172-31-57-253.ec2.internal	Configuration Manager (Red Hat Satellite)	default	04/24/17 15:32:21 UTC	Region 99	Valid	9

Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains



Configured Systems

Configured Systems

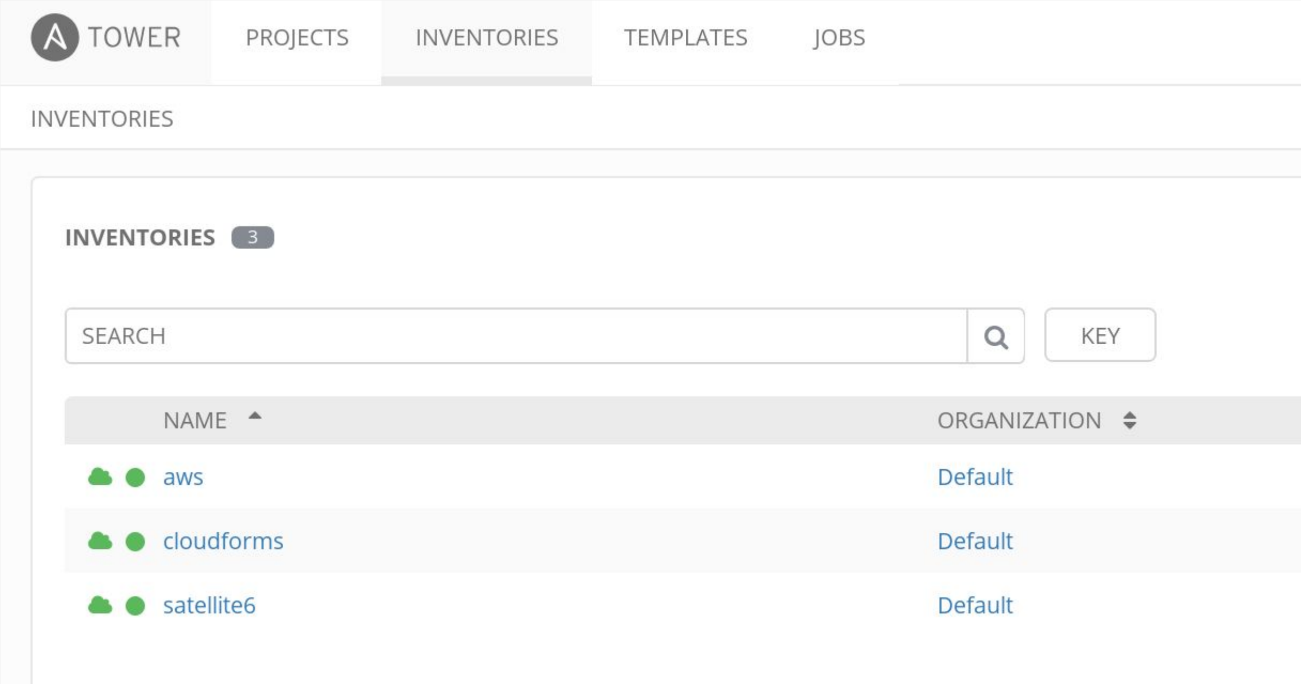
Hostname	Configuration Location	Configuration Organization	Operating System	Provider
host79.rdu.salab.redhat.com	nyc	redhat		sat6ldo

Configuration Profile * RHEL7_Crash_Base

Note: Fields marked with * are required.

Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains



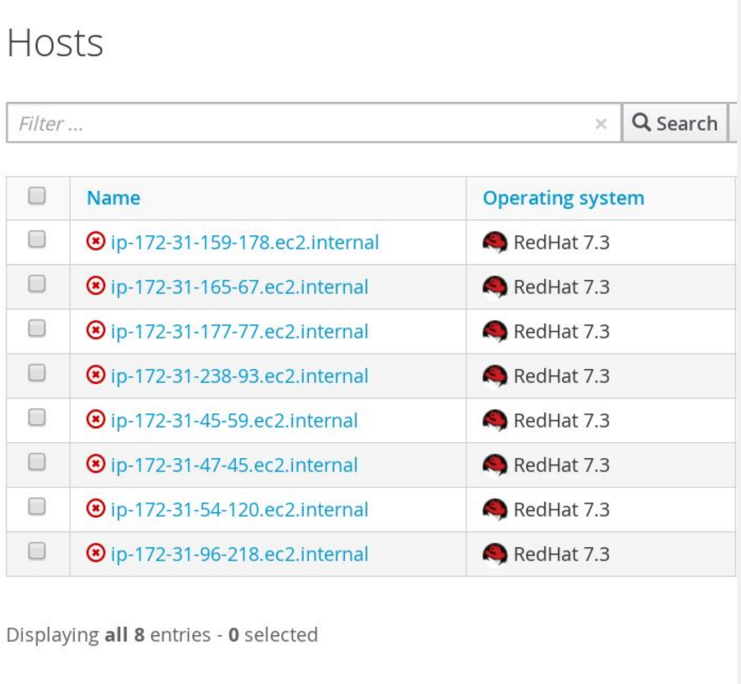
INVENTORIES 3

NAME	ORGANIZATION
aws	Default
cloudforms	Default
satellite6	Default

Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains

Satellite 6



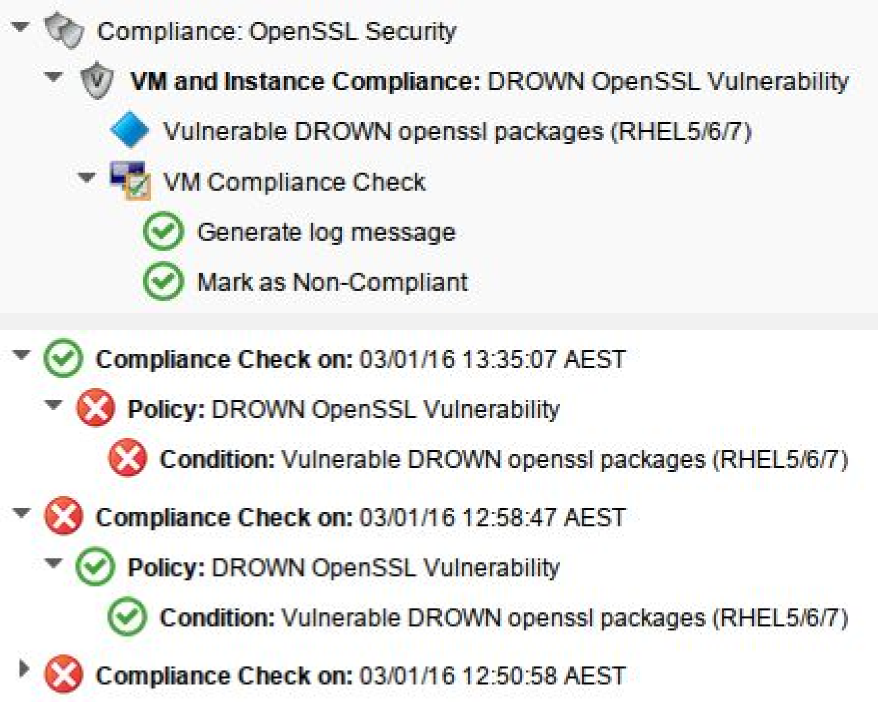
Hosts

Filter ... × | Search

□	Name	Operating system
□	⊗ ip-172-31-159-178.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-165-67.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-177-77.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-238-93.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-45-59.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-47-45.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-54-120.ec2.internal	RedHat 7.3
□	⊗ ip-172-31-96-218.ec2.internal	RedHat 7.3

Displaying **all 8** entries - **0** selected

CloudForms



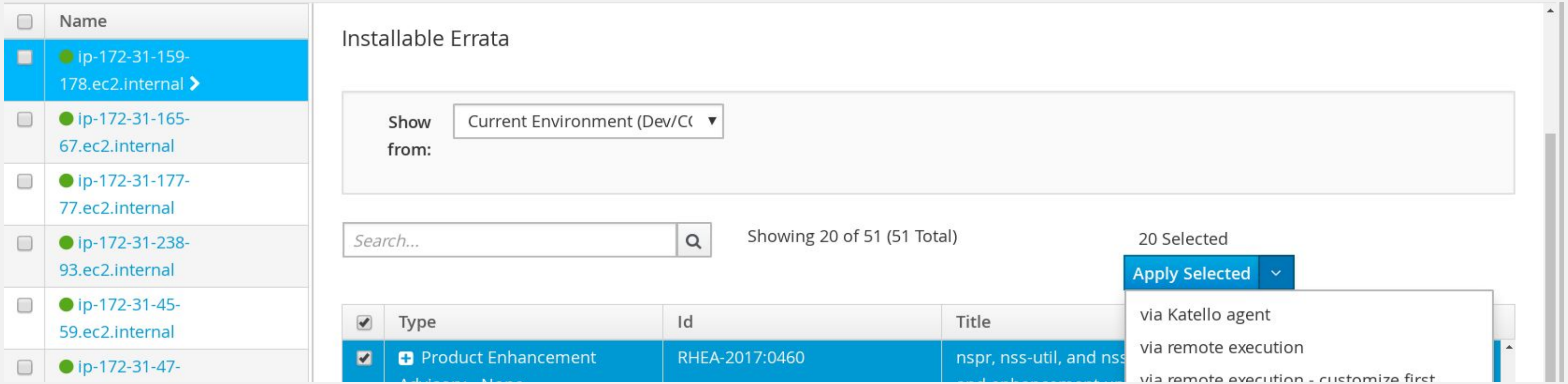
- Compliance: OpenSSL Security
  - **VM and Instance Compliance:** DROWN OpenSSL Vulnerability
    - Vulnerable DROWN openssl packages (RHEL5/6/7)
    - VM Compliance Check
      - ✓ Generate log message
      - ✓ Mark as Non-Compliant

- ✓ **Compliance Check on:** 03/01/16 13:35:07 AEST
  - ✗ **Policy:** DROWN OpenSSL Vulnerability
    - ✗ **Condition:** Vulnerable DROWN openssl packages (RHEL5/6/7)
- ✗ **Compliance Check on:** 03/01/16 12:58:47 AEST
  - ✓ **Policy:** DROWN OpenSSL Vulnerability
    - ✓ **Condition:** Vulnerable DROWN openssl packages (RHEL5/6/7)
- ✗ **Compliance Check on:** 03/01/16 12:50:58 AEST

Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains

Satellite 6



□	Name
□	ip-172-31-159-178.ec2.internal
□	ip-172-31-165-67.ec2.internal
□	ip-172-31-177-77.ec2.internal
□	ip-172-31-238-93.ec2.internal
□	ip-172-31-45-59.ec2.internal
□	ip-172-31-47-

Installable Errata

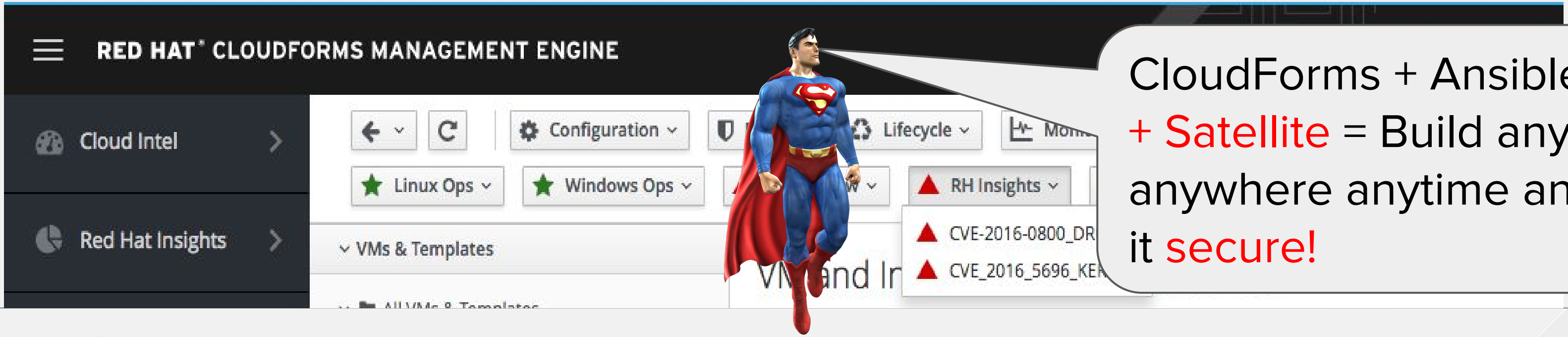
Show from: Current Environment (Dev/C(

Showing 20 of 51 (51 Total)

20 Selected

☑	Type	Id	Title
☑	Product Enhancement	RHEA-2017:0460	nspr, nss-util, and nss

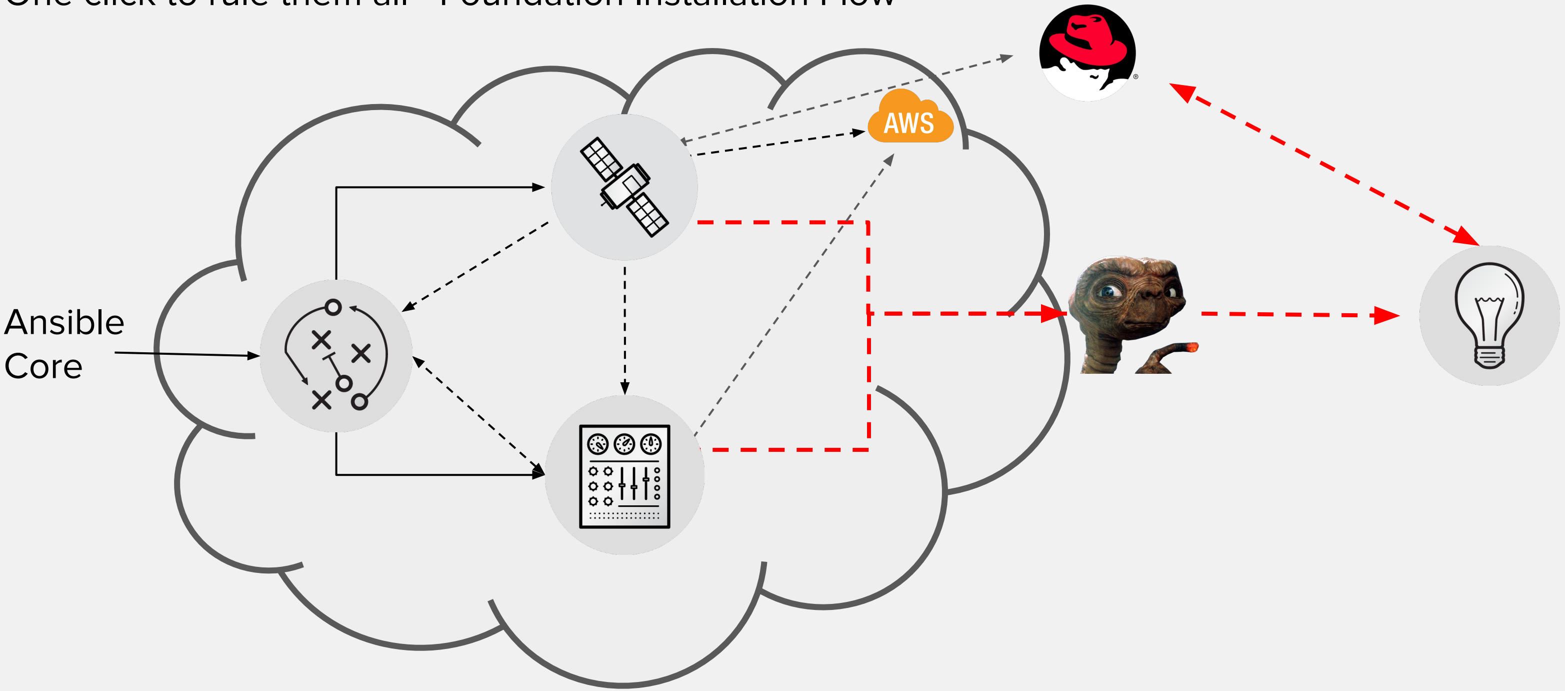
CloudForms



CloudForms + Ansible Tower + Satellite = Build anything anywhere anytime and make it secure!

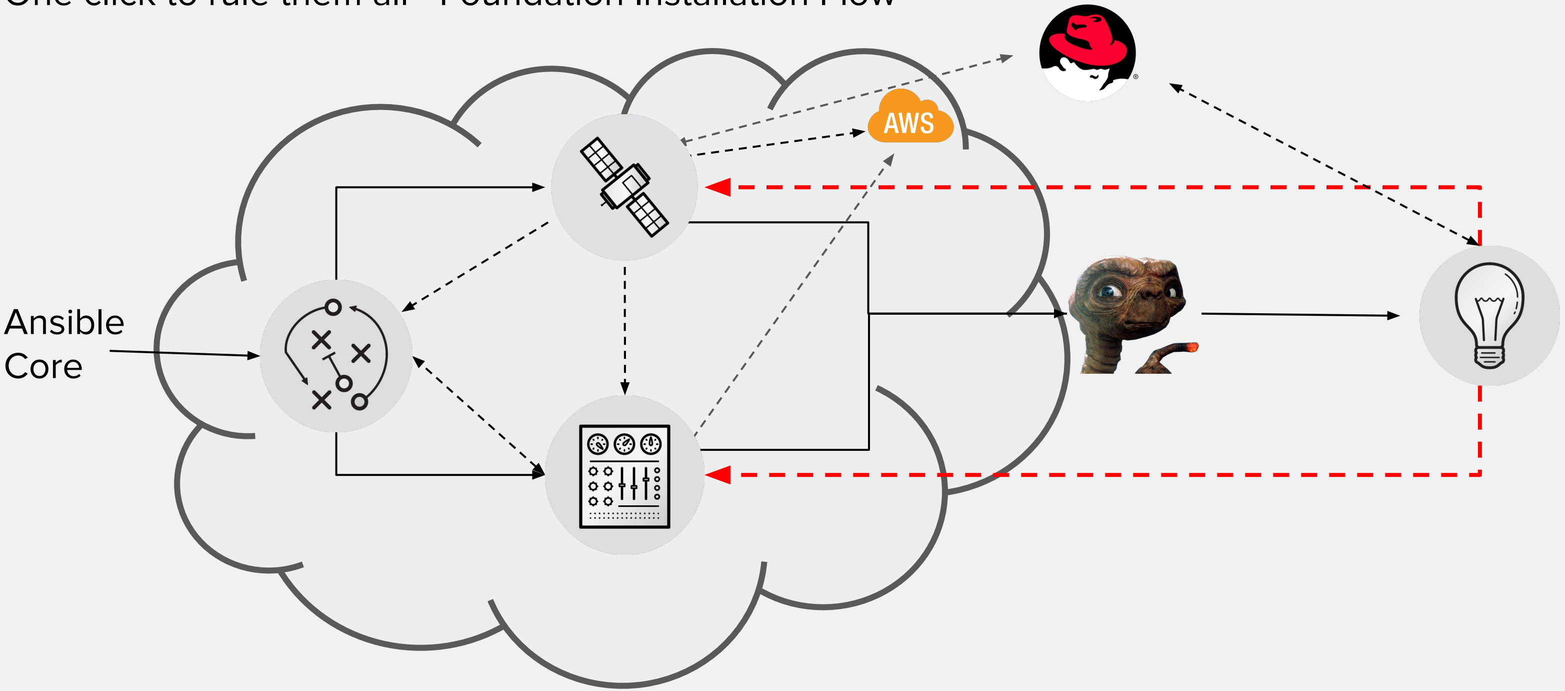
RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow



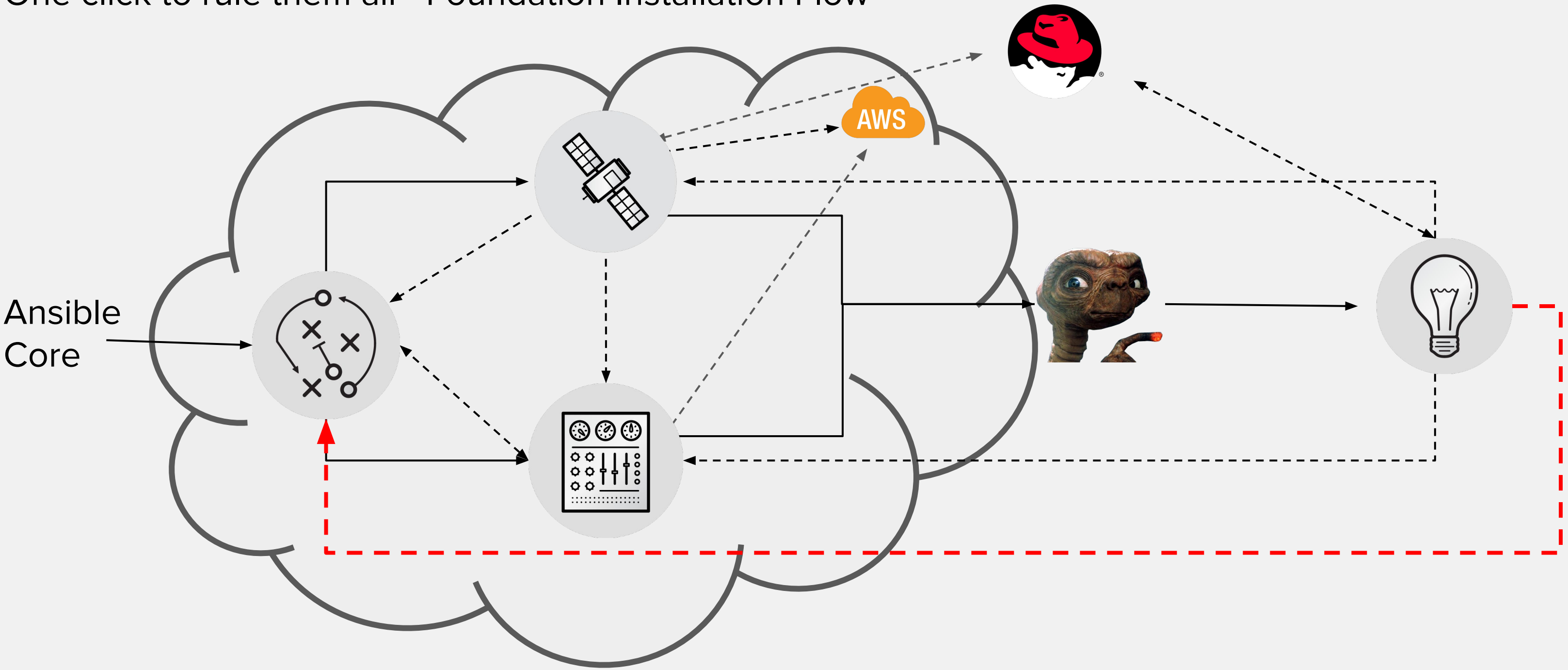
RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow



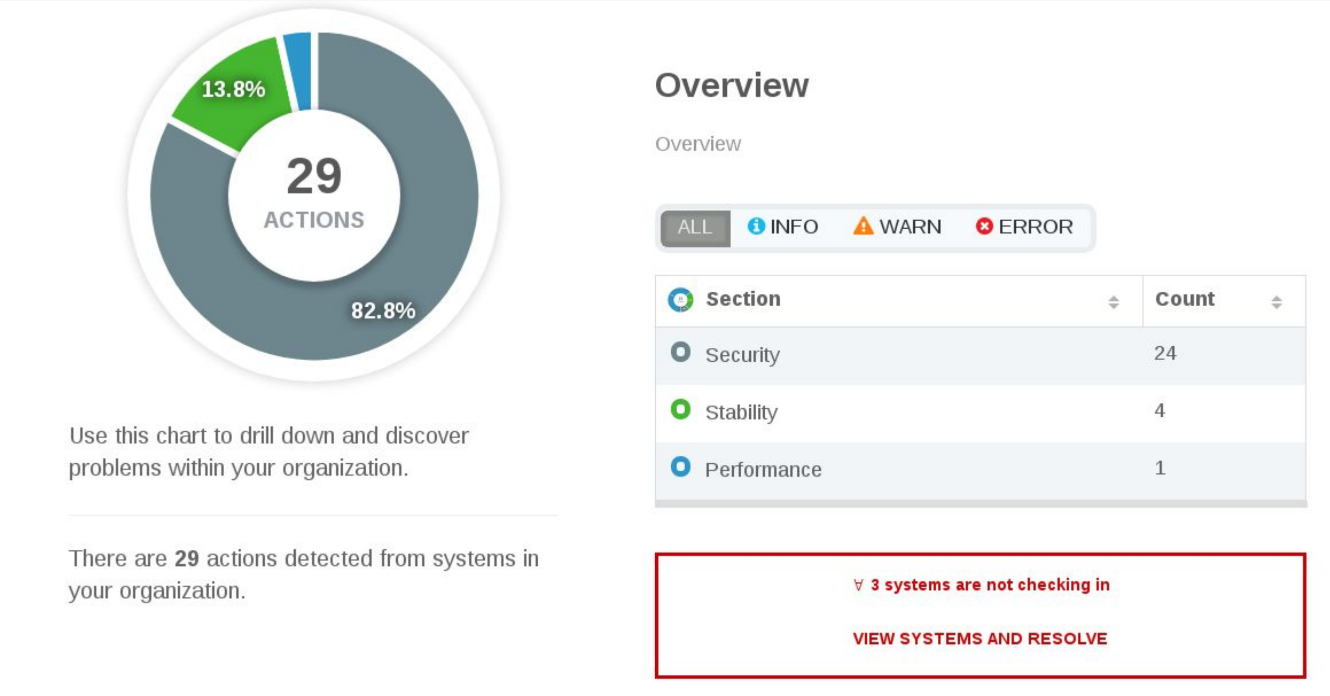
RH-MANAGEMENT INSIGHTS

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

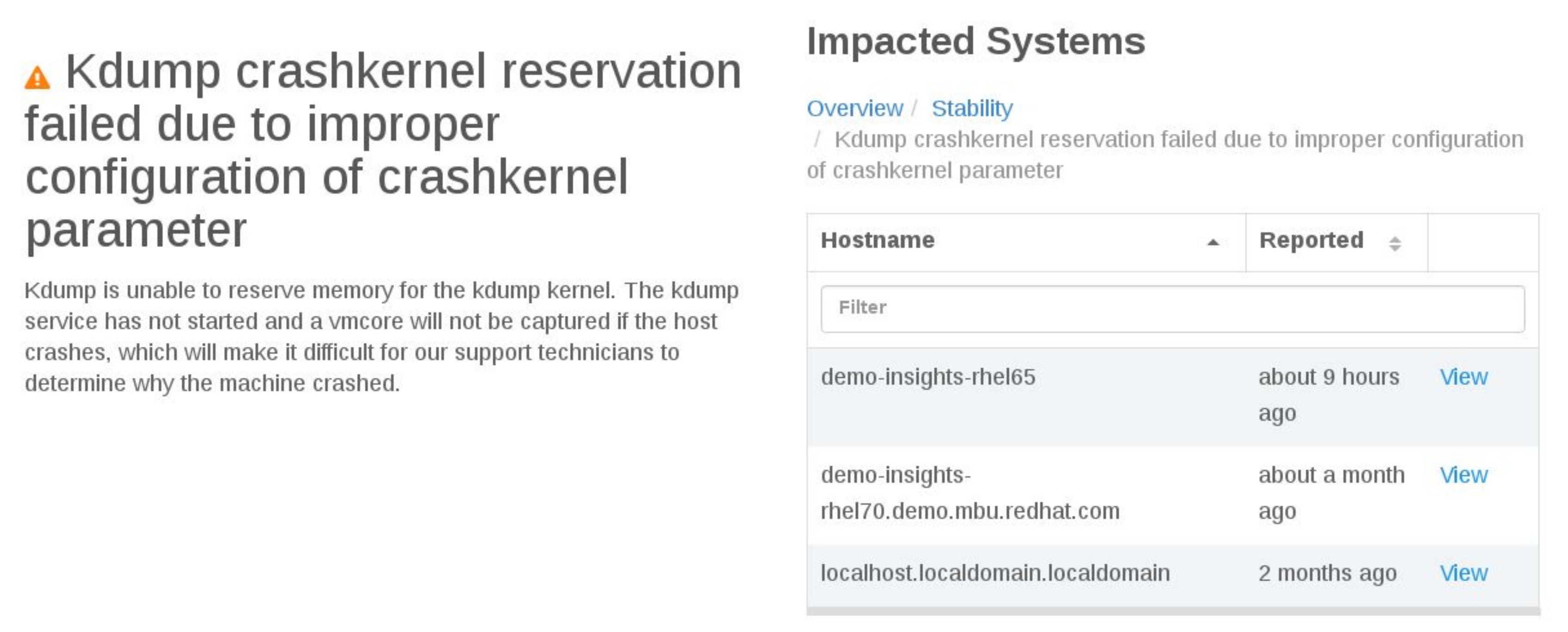
Overview

Section	Count
Security	24
Stability	4
Performance	1

∀ 3 systems are not checking in

Red Hat Management Automated

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



⚠ 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

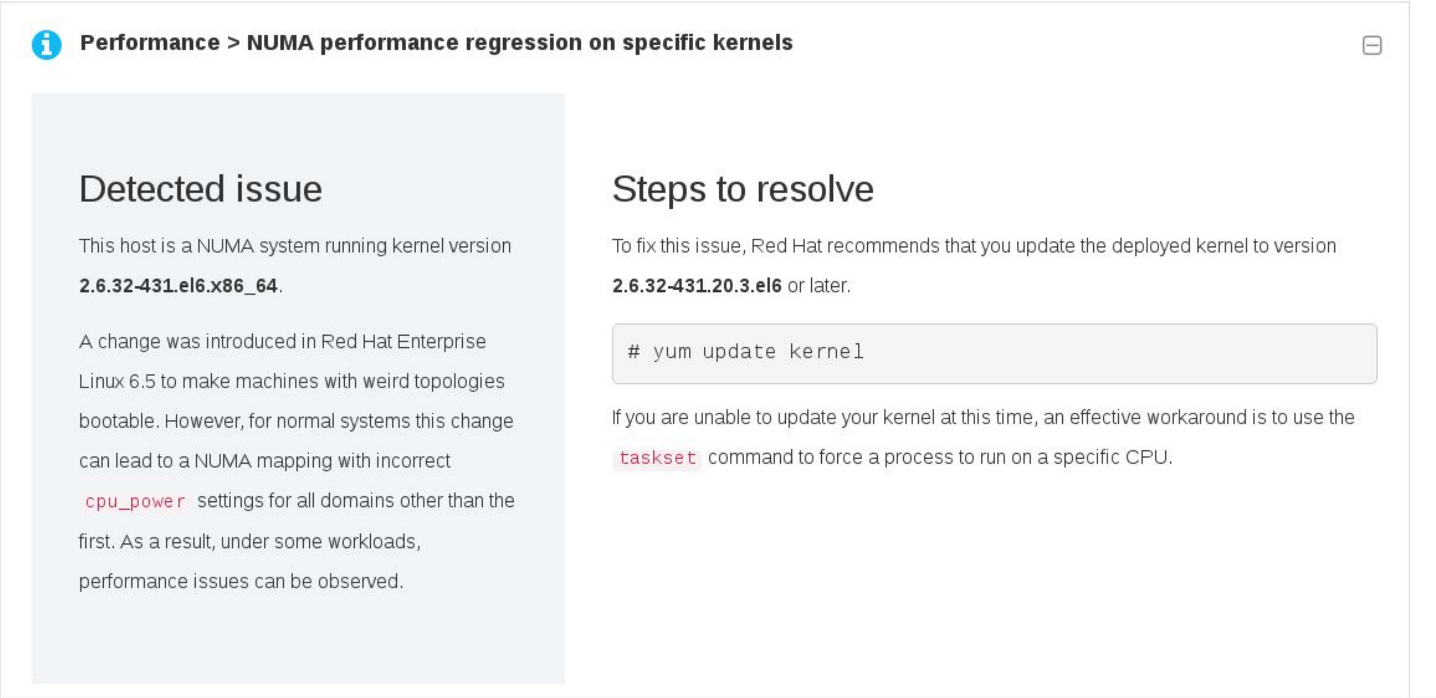
Overview / Stability / Kdump crashkernel reservation failed due to improper configuration of crashkernel parameter

Filter

Hostname ▲	Reported ⬍	
demo-insights-rhel65	about 9 hours ago	View
demo-insights-rhel70.demo.mbu.redhat.com	about a month ago	View
localhost.localdomain.localdomain	2 months ago	View

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

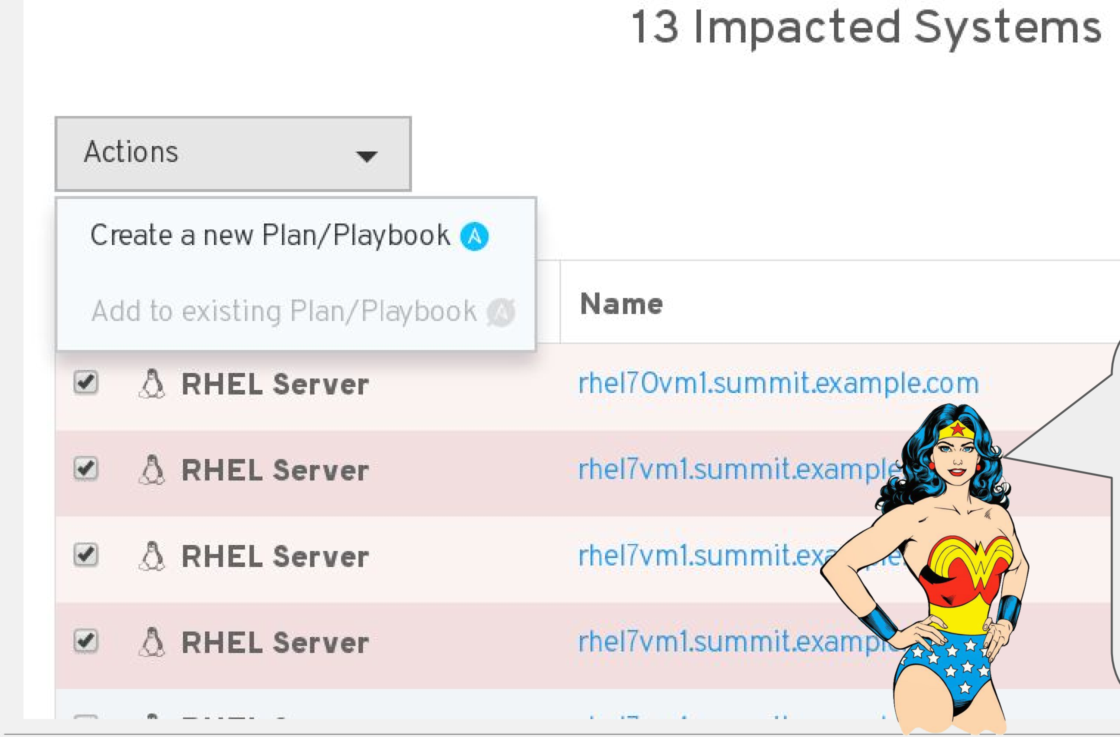
To fix this issue, Red Hat recommends that you update the deployed kernel to version **2.6.32-431.20.3.el6** or later.

```
# 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.

Red Hat Management Automated

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



13 Impacted Systems

Actions ▾

		Name	Reported
☑	RHEL Server	rhel70vm1.summit.example.com	
☑	RHEL Server	rhel7vm1.summit.example	
☑	RHEL Server	rhel7vm1.summit.exa	
☑	RHEL Server	rhel7vm1.summit.examp	

CloudForms + Ansible Tower + Satellite + Insights = Build anything, anywhere, anytime, make it secure and keep a piece of mind that you’ve done it right

RH-MANAGEMENT SUPERPOWERS TEAM

SATELLITE 6

INSIGHTS

ANSIBLE TOWER

CLOUDFORMS

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

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

https://youtu.be/dLuuZCIUkqg

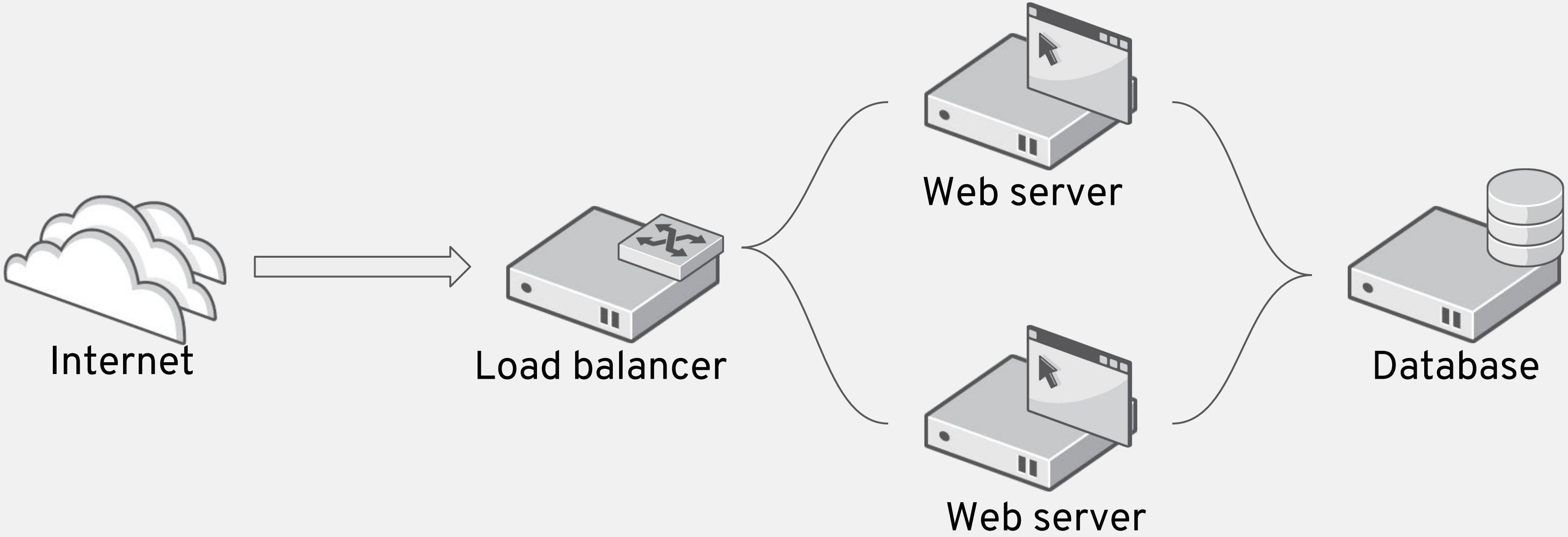
Here’s a practical example, kids!

APPLICATION ANATOMY

Or: whose critical application only runs on a single server?

- 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



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

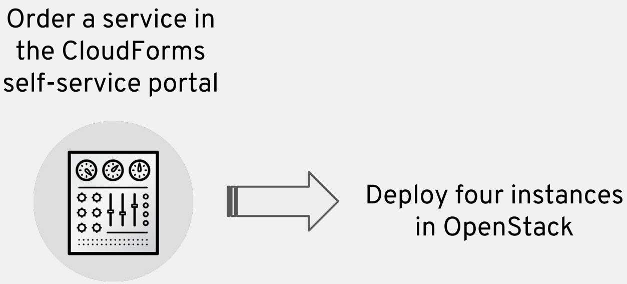
CLOUDFORMS SELF-SERVICE

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

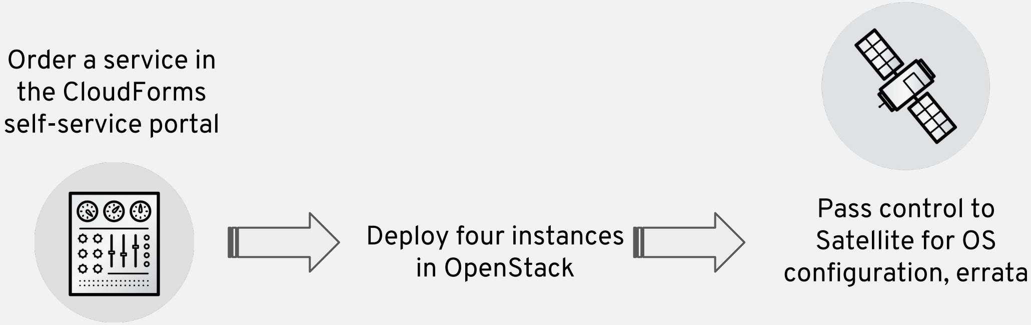


Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

CLOUDFORMS SELF-SERVICE

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



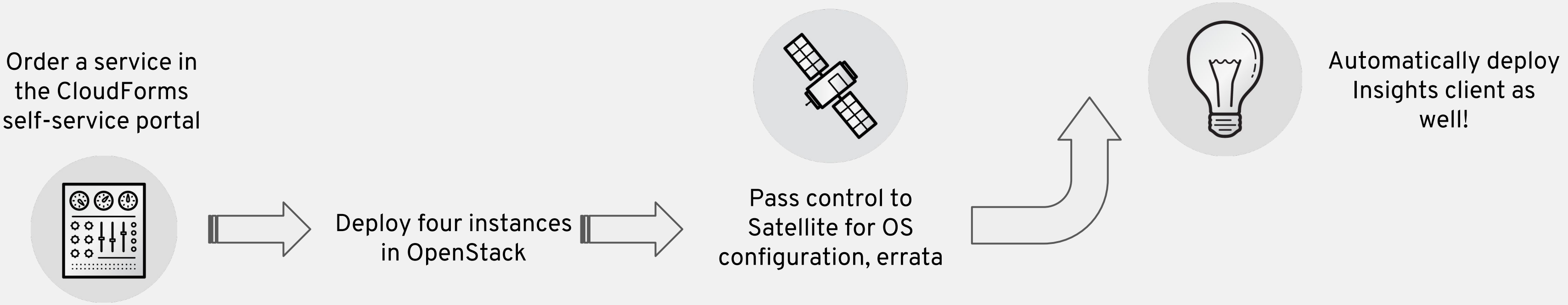
Order a service in the CloudForms self-service portal

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

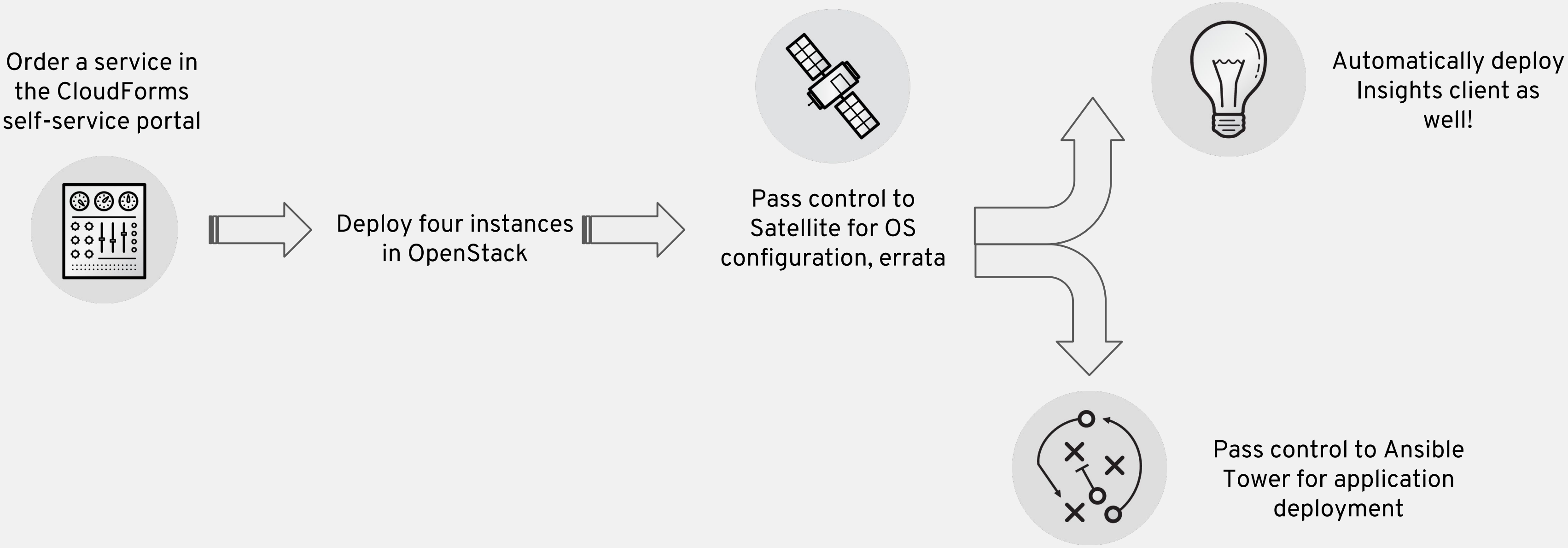
Deploy four instances in OpenStack

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

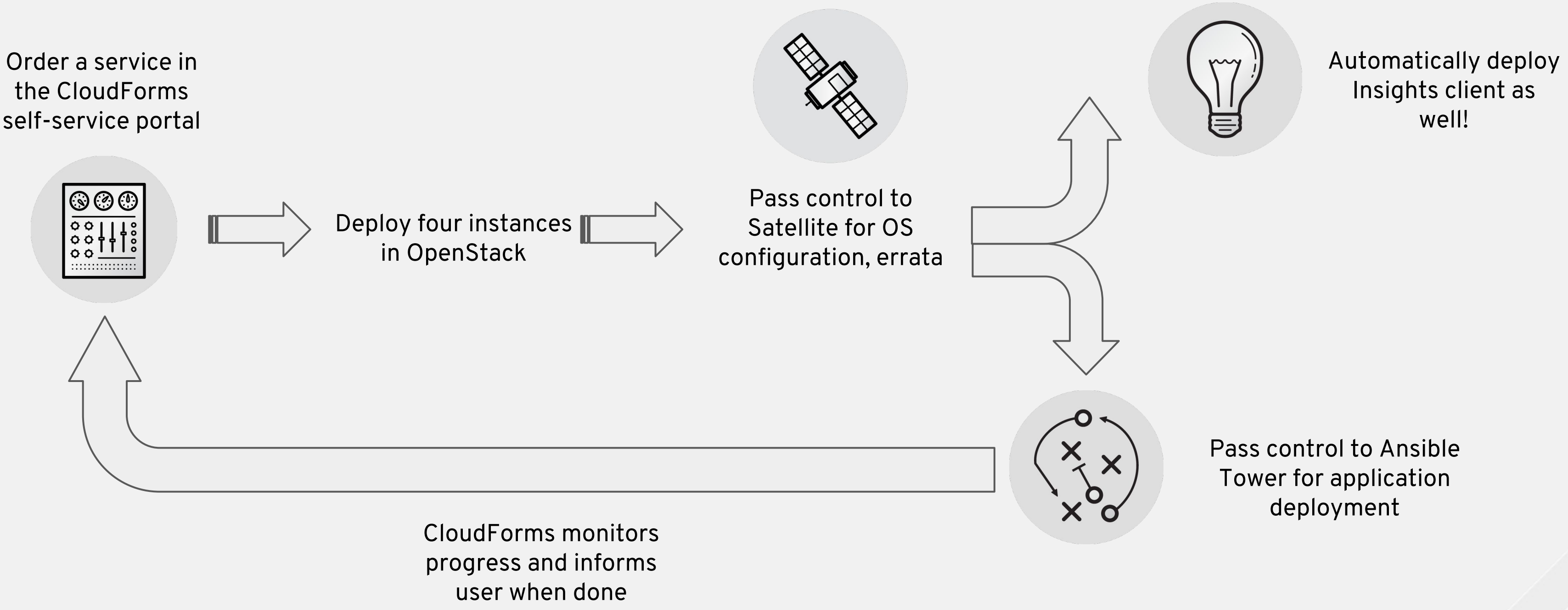
Pass control to Satellite for OS configuration, errata

Automatically deploy Insights client as well!

Pass control to Ansible Tower for application deployment

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

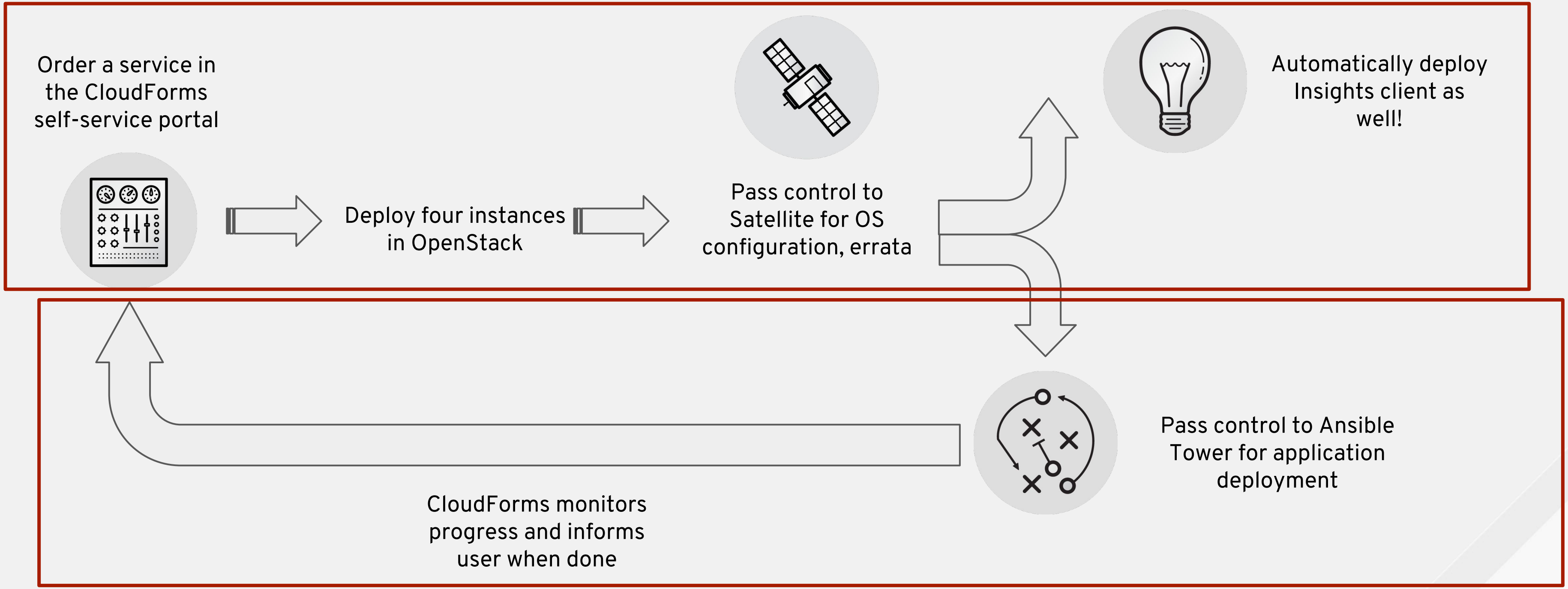
Automatically deploy Insights client as well!

Pass control to Ansible Tower for application deployment

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

Automatically deploy Insights client as well!

Pass control to Ansible Tower for application deployment

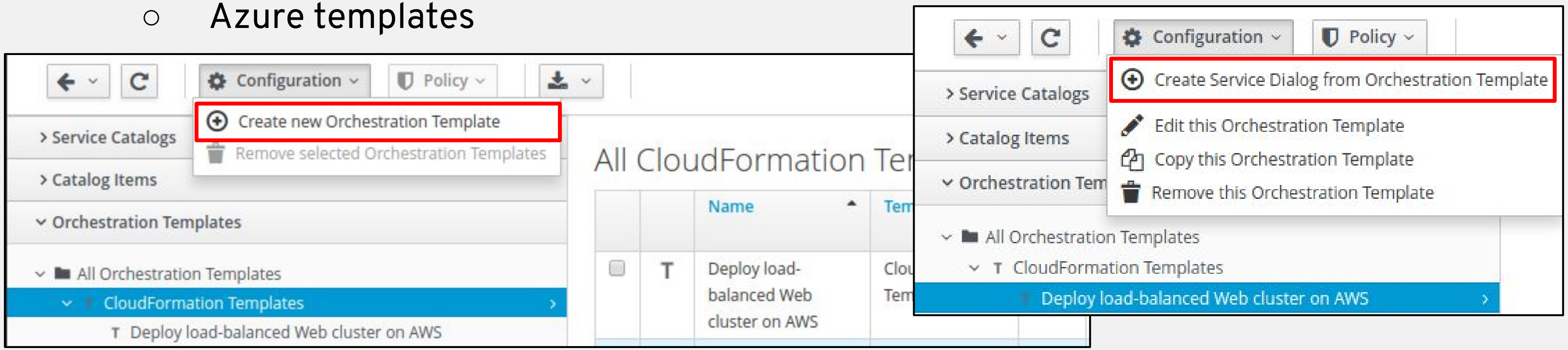
CloudForms monitors progress and informs user when done

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

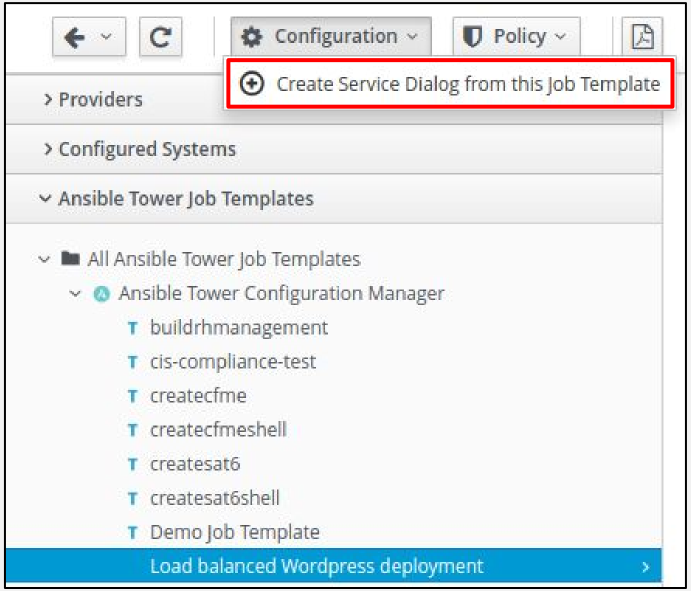
- CloudForms can automatically create dialogs from the parameters in those templates


Configuration
Policy
Create new Orchestration Template
Remove selected Orchestration Templates
Service Catalogs
Catalog Items
Orchestration Templates
All Orchestration Templates
CloudFormation Templates
Deploy load-balanced Web cluster on AWS
All CloudFormation Ter
Name
Tem
Deploy load-balanced Web cluster on AWS
Clou
Tem
Configuration
Policy
Create Service Dialog from Orchestration Template
Edit this Orchestration Template
Copy this Orchestration Template
Remove this Orchestration Template
Service Catalogs
Catalog Items
Orchestration Tem
All Orchestration Templates
CloudFormation Templates
Deploy load-balanced Web cluster on AWS

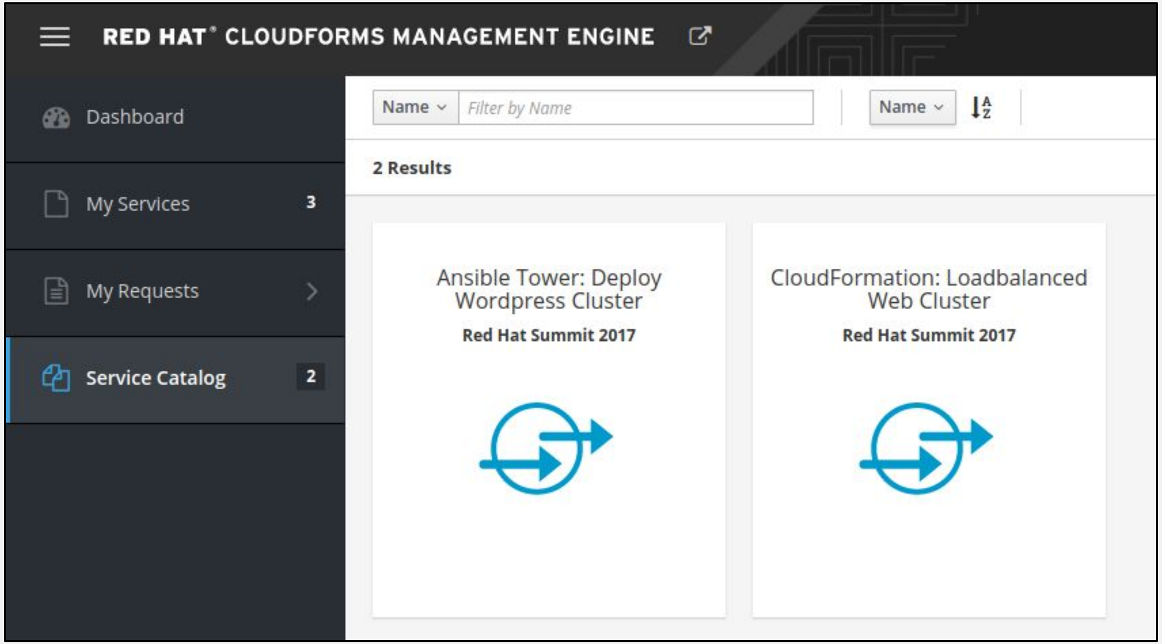
How hard is that?

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


Configuration
Policy
Create Service Dialog from this Job Template
Providers
Configured Systems
Ansible Tower Job Templates
All Ansible Tower Job Templates
Ansible Tower Configuration Manager
buildrhmanagement
cis-compliance-test
createcfme
createcfmeshell
createsat6
createsat6shell
Demo Job Template
Load balanced Wordpress deployment

So now we have two self service items



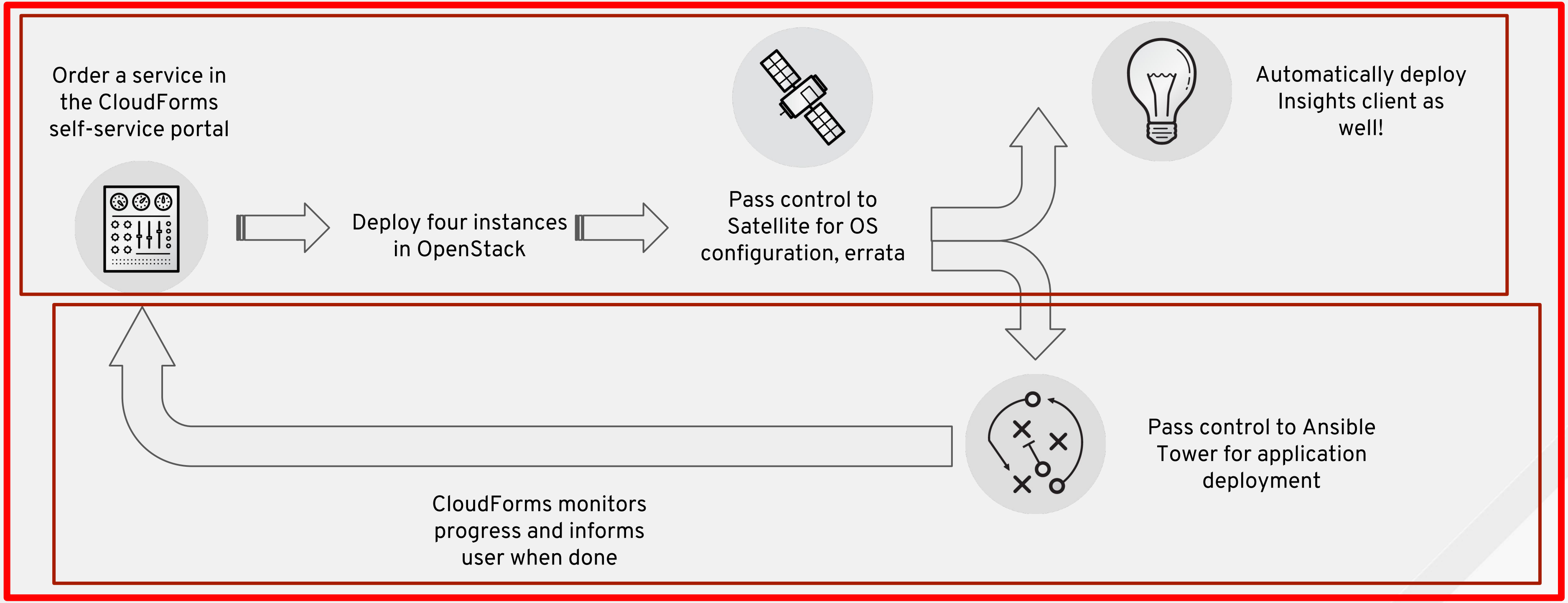
2 Results

Ansible Tower: Deploy Wordpress Cluster
Red Hat Summit 2017

CloudFormation: Loadbalanced Web Cluster
Red Hat Summit 2017

This calls for a bundle!

BUNDLING CATALOG ITEMS



Order a service in the CloudForms self-service portal

Deploy four instances in OpenStack

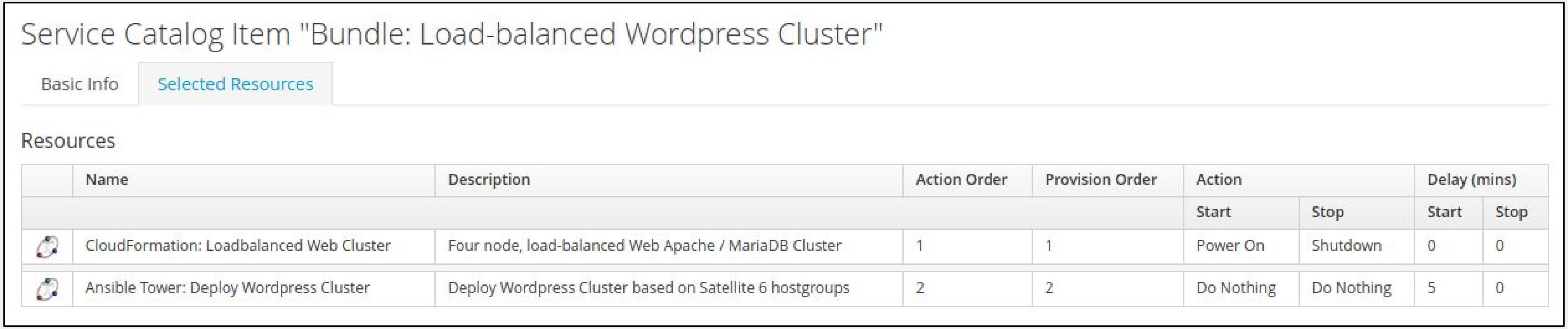
Pass control to Satellite for OS configuration, errata

Automatically deploy Insights client as well!

Pass control to Ansible Tower for application deployment

CloudForms monitors progress and informs user when done

A Catalog Bundle!

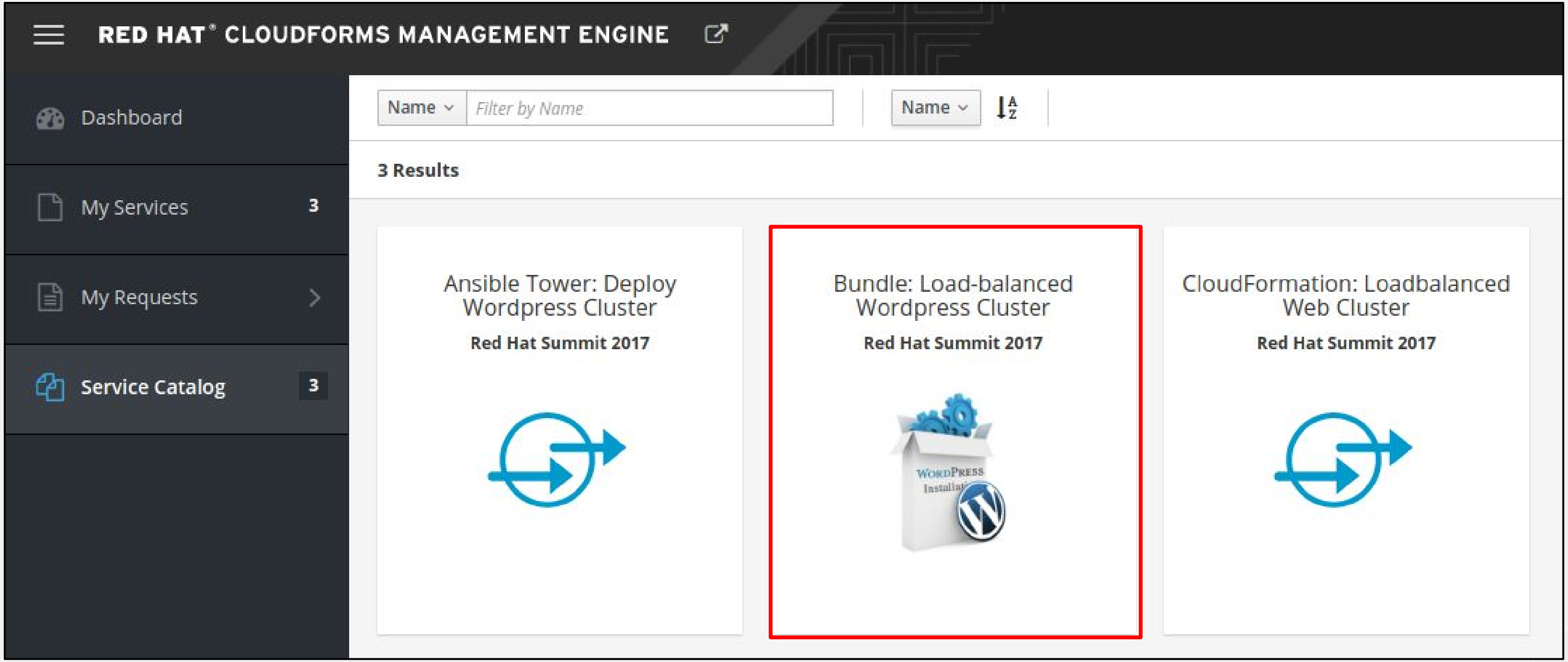


Service Catalog Item "Bundle: Load-balanced Wordpress Cluster"

Resources

	Name	Description	Action Order	Provision Order	Action		Delay (mins)	
					Start	Stop	Start	Stop
	CloudFormation: Loadbalanced Web Cluster	Four node, load-balanced Web Apache / MariaDB Cluster	1	1	Power On	Shutdown	0	0
	Ansible Tower: Deploy Wordpress Cluster	Deploy Wordpress Cluster based on Satellite 6 hostgroups	2	2	Do Nothing	Do Nothing	5	0

A CatalogBundle!



RED HAT® CLOUDFORMS MANAGEMENT ENGINE

- Dashboard
- My Services 3
- My Requests
- Service Catalog 3

Name | Filter by Name | Name

3 Results

Ansible Tower: Deploy Wordpress Cluster
Red Hat Summit 2017

Bundle: Load-balanced Wordpress Cluster
Red Hat Summit 2017

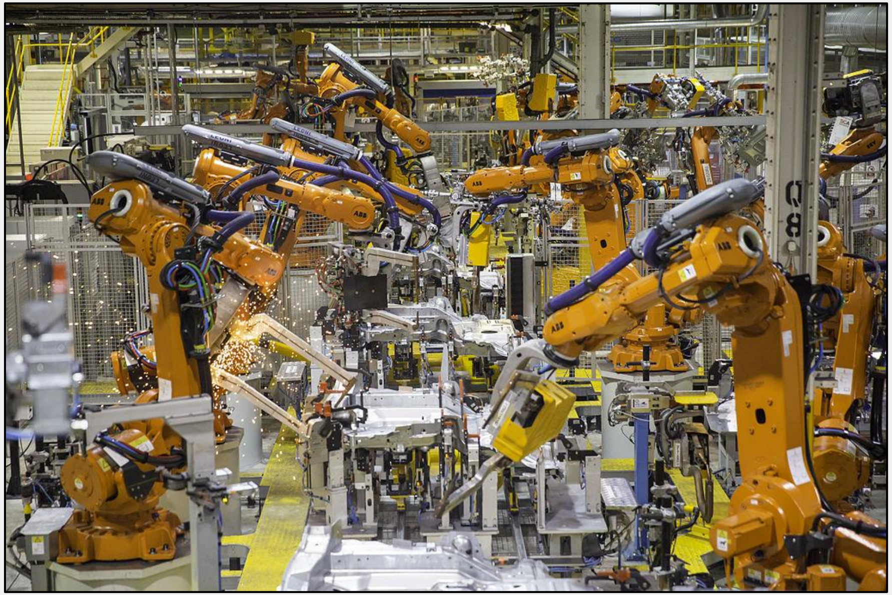
CloudFormation: Loadbalanced Web Cluster
Red Hat Summit 2017

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

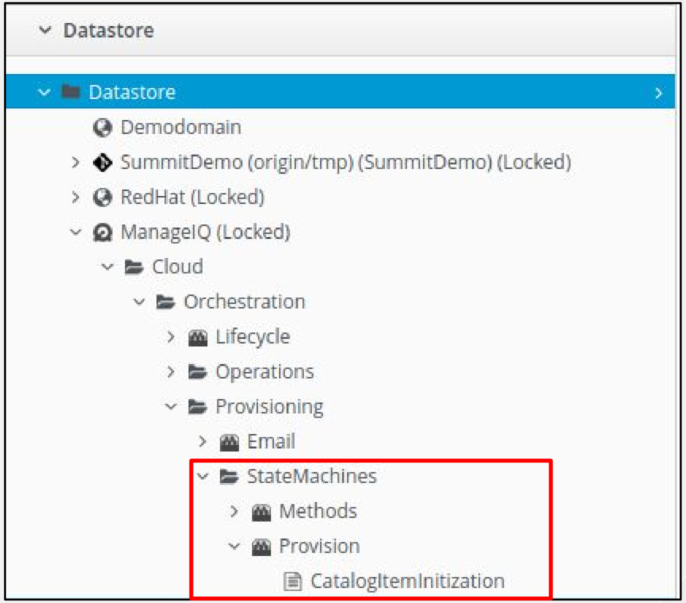
- Saving of the hostnames Happens in same script as previous customization example
- Customized method to start the Ansible Tower Job Template:

  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 :)



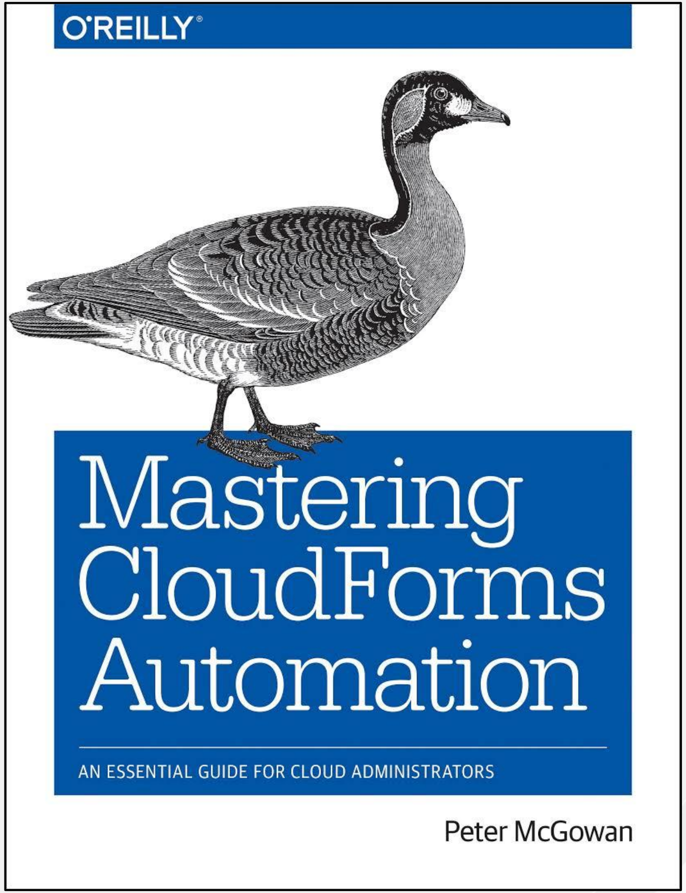
Datastore

- Datastore
  - Demodomain
  - SummitDemo (origin/tmp) (SummitDemo) (Locked)
  - RedHat (Locked)
  - ManageIQ (Locked)
    - Cloud
      - Orchestration
        - Lifecycle
        - Operations
        - Provisioning
          - Email
          - StateMachines
            - Methods
            - Provision
              - CatalogItemInitization

Want to learn more?

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

- http://red.ht/2oYQttJ


O'REILLY®
Mastering CloudForms Automation
AN ESSENTIAL GUIDE FOR CLOUD ADMINISTRATORS
Peter McGowan

DEMO

- 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 use 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!

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

RED HAT
SUMMIT

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)