

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

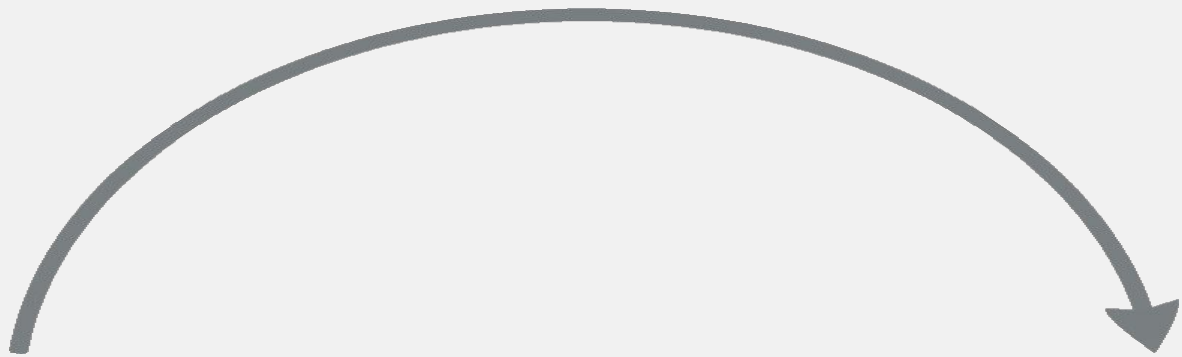
# Intro

Ansible Tower, CloudForms, Insights, Satellite 6

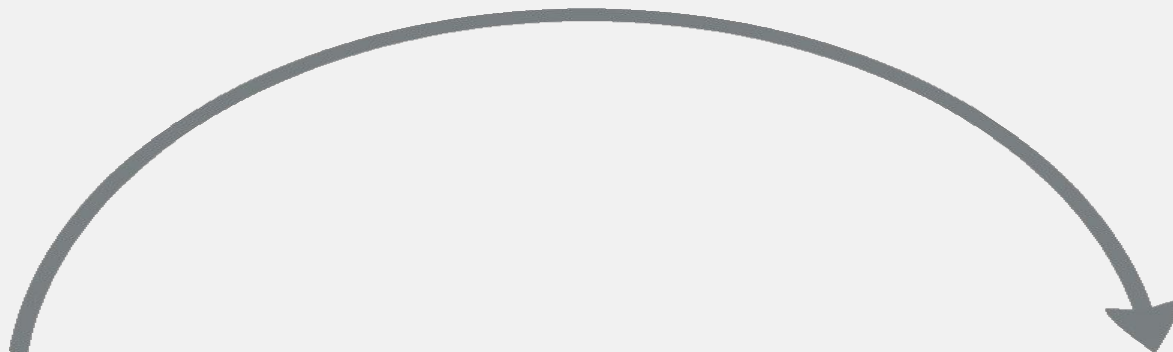
- Why do we care?
- What problems are we solving?
- How do the Red Hat tools address this?
- What does it look like in action?







**CODE**



**PROD**



{THE HOW}

CODE



{THE WHAT}

PROD



{THE WHERE}





{THE HOW}

CODE



{THE WHAT}

PROD



{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

{THE HOW}

CODE



{THE WHAT}

“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**



PROD

{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/lomb/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

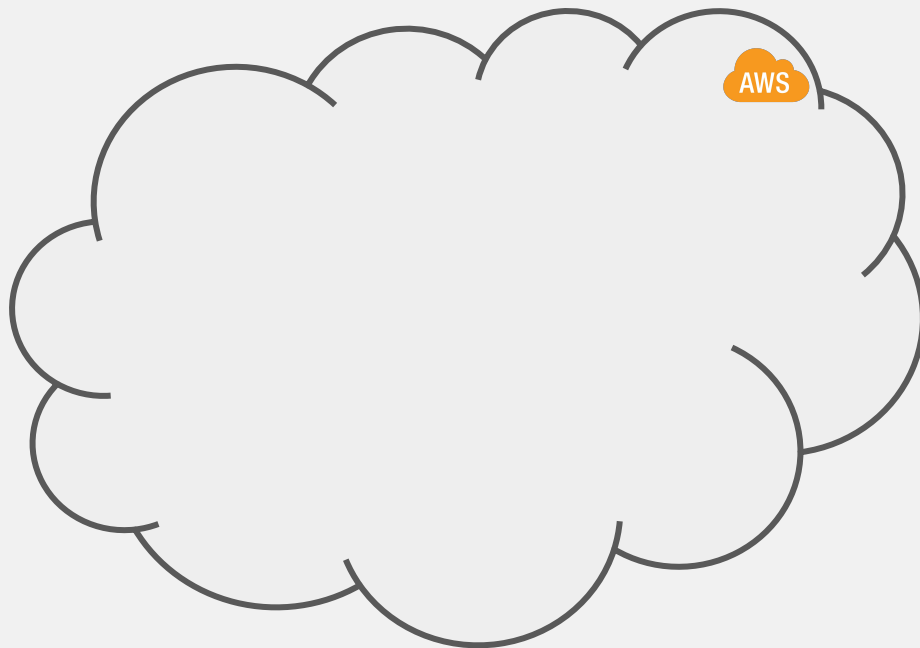
```
summit2017$ ansible-playbook buildrhmgmt.yaml --private-key=ldomb.pem  
--vault-password-file=../vaultpass -vv
```



# RH-MANAGEMENT CORE

One click to rule them all - Foundation Installation Flow

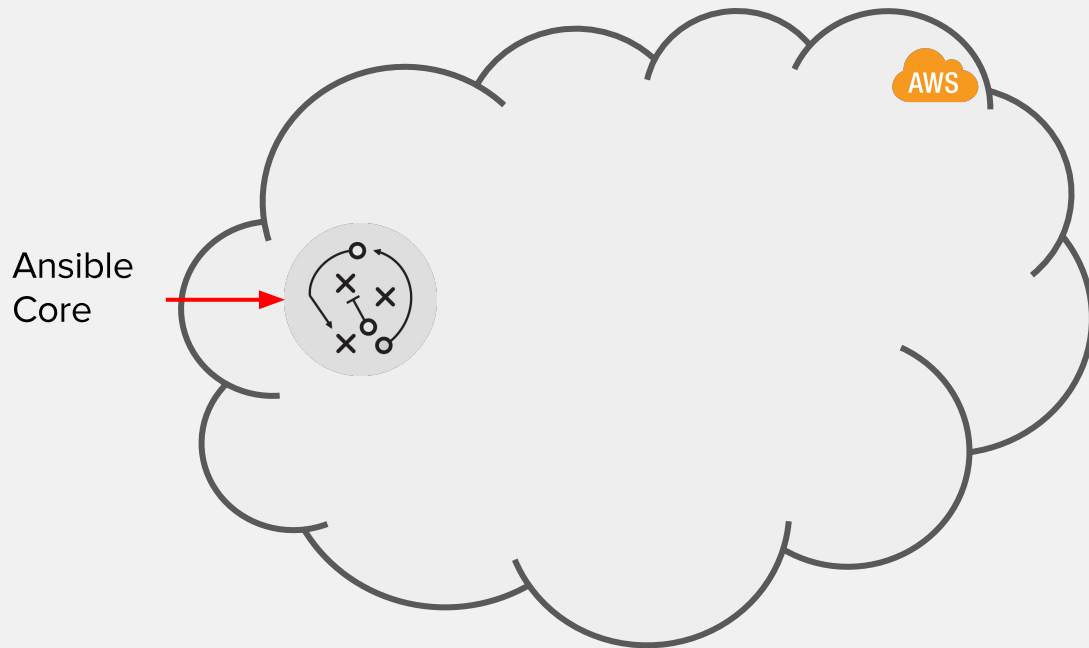
Ansible  
Core





# 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'efs', 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-0abb05f7eb328d8', u'efs 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'efs', 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-0abb05f7eb328d8', u'efs 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]

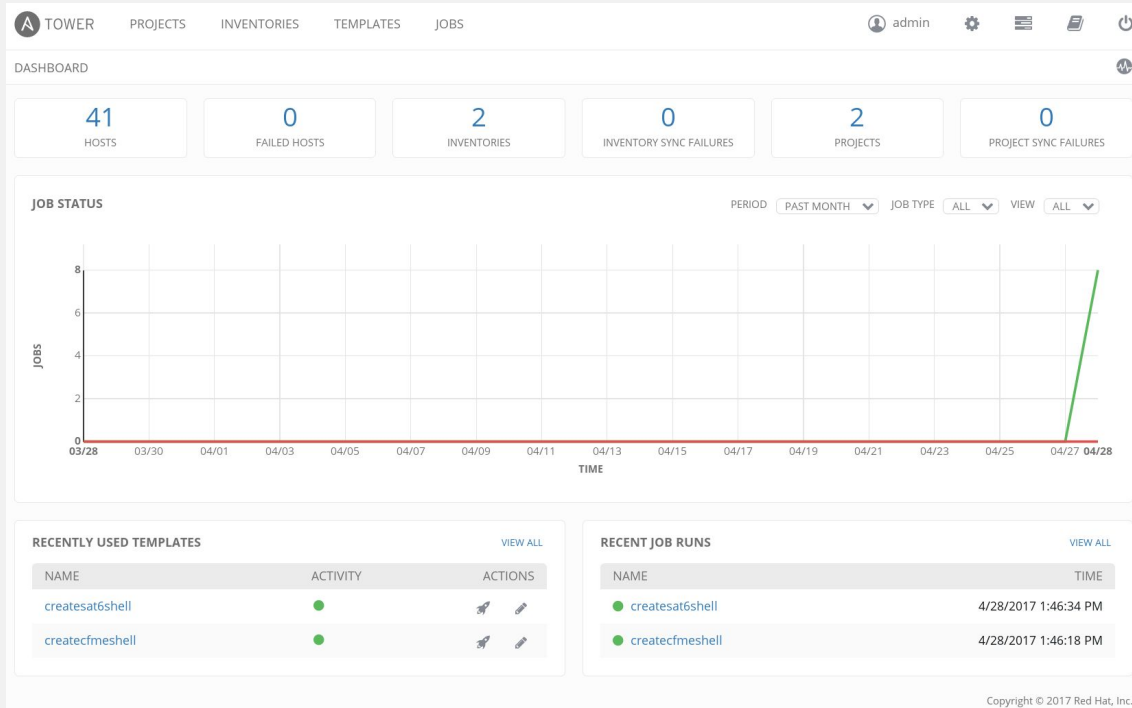
TASK [buildansibletower : add /etc/tower path to setting.py] *****
changed: [54.144.64.252]

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

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

# Red Hat Management Automated

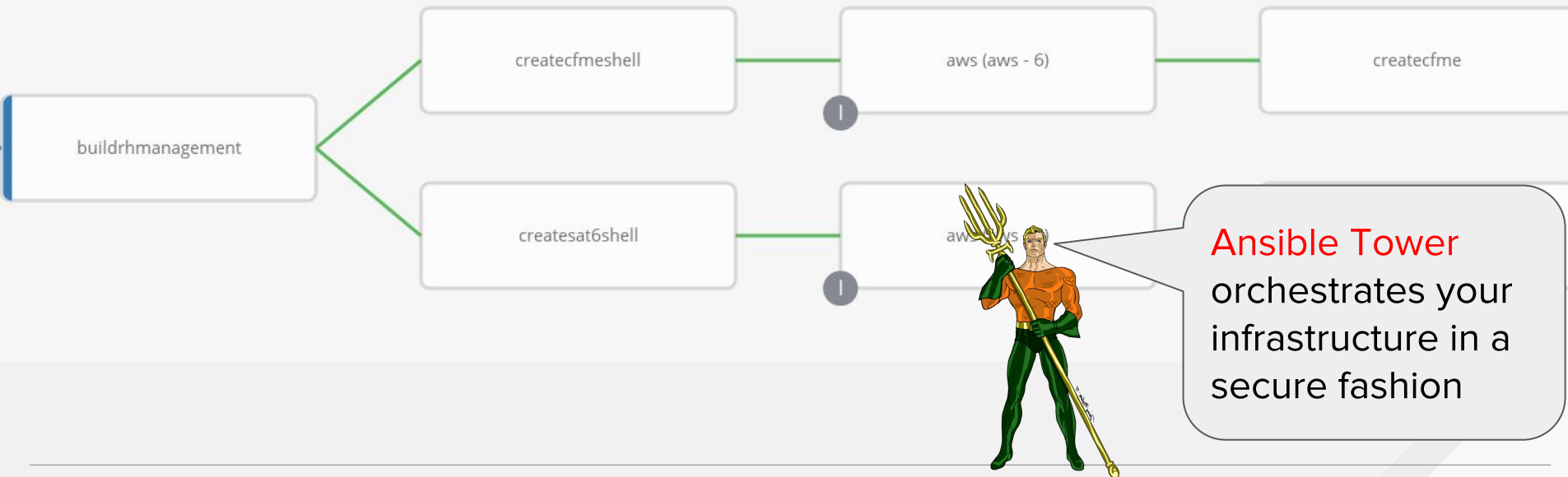
## Building Ansible Tower - Gains



Copyright © 2017 Red Hat, Inc.

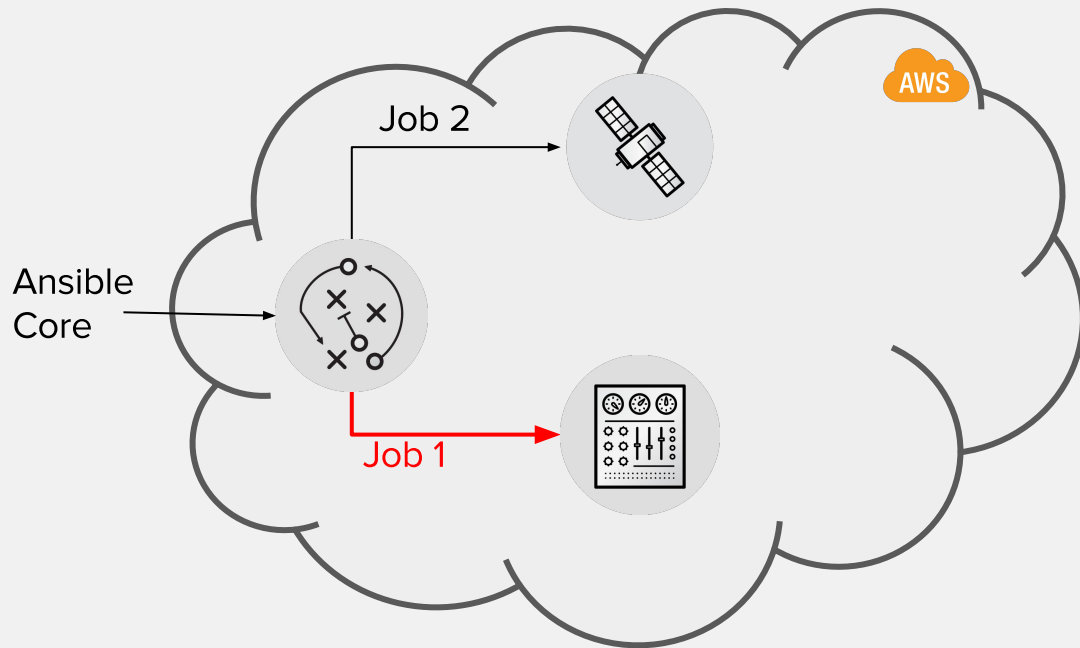
# Red Hat Management Automated

Building Ansible Tower - Gains - Workflow Editor



# 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

TOWER PROJECTS INVENTORIES TEMPLATES JOBS admin ⚙️ ☰ 📄 🔌

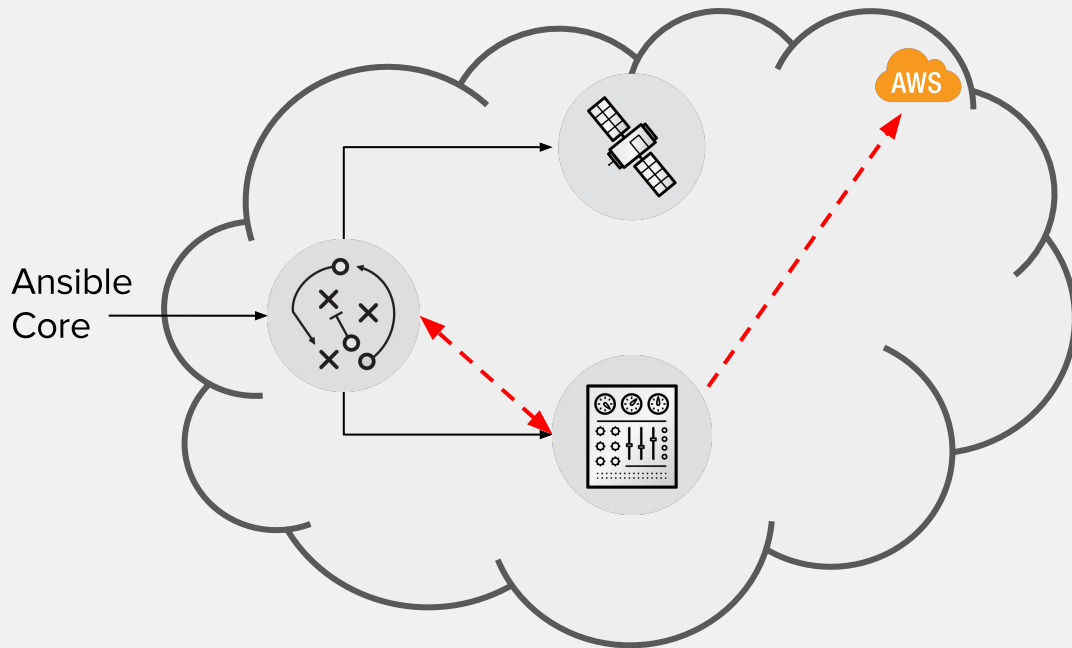
JOBS / 7 - createcfme

7			
8	PLAY [create cfme] *****		13:47:12
9			
10	TASK [Gathering Facts] *****		13:47:12
11	ok: [52.23.172.218]		
12	META: ran handlers		
13			
14	TASK [buildcfme : copy chrony configuration for RHEL7] *****		13:47:16
15	changed: [52.23.172.218]		
16			
17	TASK [buildcfme : ensure chrony service is started and enabled] *****		13:47:19
18	ok: [52.23.172.218]		
19			
20	TASK [buildcfme : ensure chrony is getting restarted if necessary] *****		13:47:20
21	changed: [52.23.172.218]		
22			
23	TASK [buildcfme : perform appliance basic configuration] *****		13:47:21
24	changed: [52.23.172.218]		
25			
26	TASK [buildcfme : wait for cfme ui] *****		13:49:11
			^ TOP

Job 1

# 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 <sup>▲</sup>	URL	Type	Zone	Last Refresh Date	Region Description	Status	Total Configured Systems
<input type="checkbox"/>		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

SEARCH	Q	KEY
NAME		ORGANIZATION
aws		Default
cloudforms		Default

# Red Hat Management Automated

CloudForms - Ansible Tower integration gains

Providers

- ▼ All Configuration Manage...
  - > Red Hat Satellite Provi...
  - ▼ Ansible Tower Provide...
    - ▼ **Ansible Tower Con... >**
      - > aws
      - > cloudforms
      - > Demo Inventory
      - > satellite6

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

> Providers

> Configured Systems

▼ Ansible Tower Job Templates

▼ All Ansible Tower Job Tem...

▼ Ansible Tower Configu... >

- T buildrhmanageme...
- T cis-compliance-test
- T createcfme
- T createcfmeshell
- T createsat6
- T createsat6shell
- T Demo Job Template
- T Load balanced Wo...

Job Templates under "Ansible Tower Configuration Manager"

Search  🔍

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

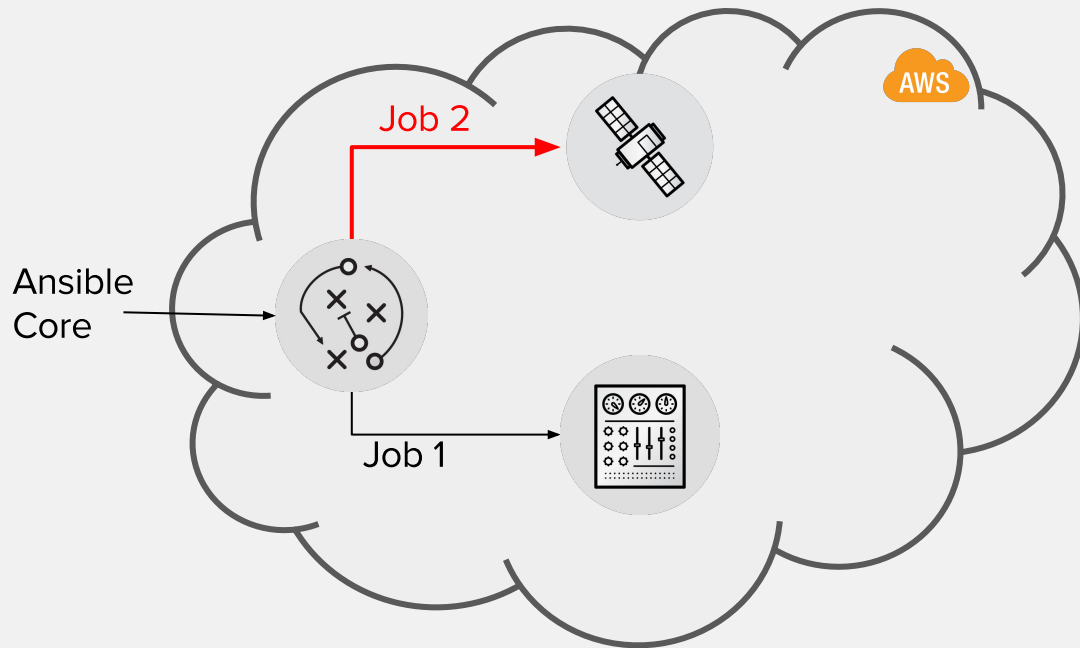
The screenshot displays the Red Hat CloudForms Management Engine interface. The top navigation bar includes the title "RED HAT® CLOUDFORMS MANAGEMENT ENGINE" and a search icon. The left sidebar contains navigation options: "Dashboard", "My Services" (5 items), "My Requests" (1 item), and "Service Catalog" (3 items). The main content area shows a list of services under the heading "3 Results".

Name	Filter by Name	Name	Sort
BuildRHMGMT	Red Hat Summit 2017 RHMGMT	Complete Wordpress cluster setup	Red Hat Summit 2017
		Load Balanced Wordpress Cluster	Red Hat Summit 2017

The "BuildRHMGMT" service is highlighted with a red circular icon containing the word "easy". The "Complete Wordpress cluster setup" service features an illustration of Spider-Man standing next to a box labeled "WORDPRESS Install" with a gear icon. A speech bubble points to Spider-Man with the text: "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

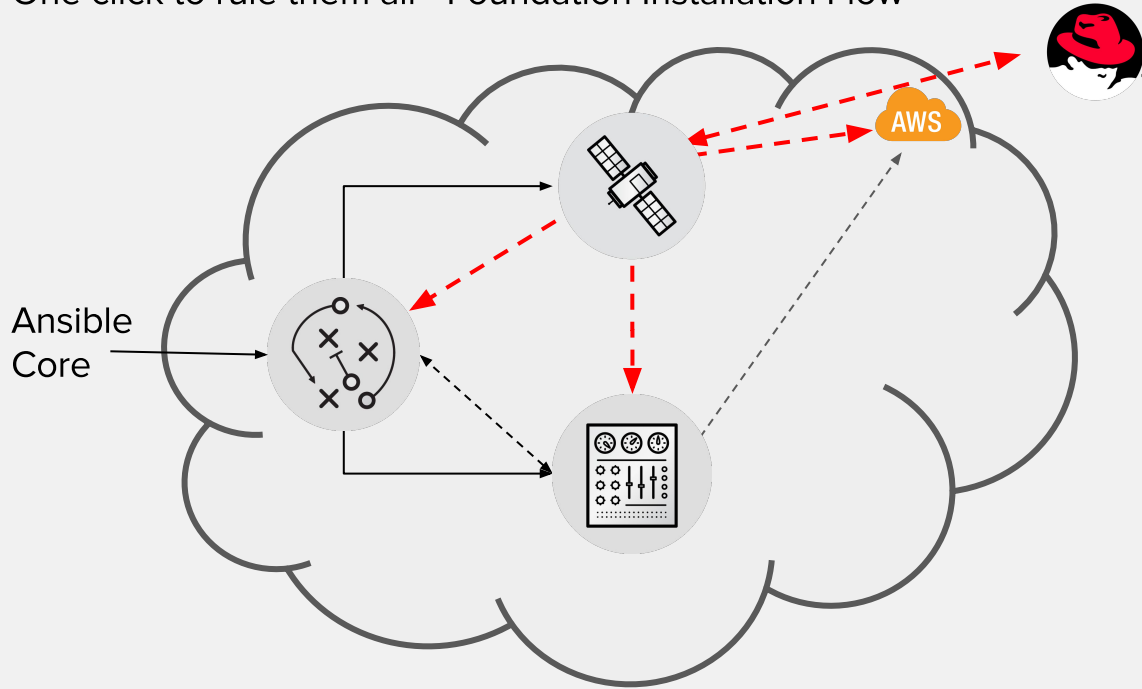
The screenshot shows the Tower web interface with a navigation bar at the top containing 'TOWER', 'PROJECTS', 'INVENTORIES', 'TEMPLATES', and 'JOBS'. A user profile 'admin' and several utility icons are also present. Below the navigation bar, the breadcrumb 'JOBS / 9 - createsat6' is visible. The main content area displays a list of job tasks with their IDs, descriptions, and completion times. The tasks are as follows:

Task ID	Task Description	Completion Time
104	TASK [satellite-deployment : Set network interface autoconnect] *****	13:48:12
109		
110	TASK [satellite-deployment : Set network interface UP] *****	13:48:12
115		
116	TASK [satellite-deployment : Include firewall.yml] *****	13:48:12
118		
119	TASK [satellite-deployment : Install firewalld] *****	13:48:12
124		
125	TASK [satellite-deployment : Set hostname with hostnamectl] *****	13:48:30
127		
128	TASK [satellite-deployment : Update /etc/hosts with satellite hostname] *****	13:48:31
130		
131	TASK [satellite-deployment : Enable Firewalld] *****	13:48:31
133		
134	TASK [satellite-deployment : Firewall and hostname   Opening Firewalld ports] ***	13:48:32
149		
150	TASK [satellite-deployment : Include install vars] *****	13:48:41
152		
153	TASK [satellite-deployment : Install software] *****	13:48:41

An '^ TOP' link is located at the bottom right of the task list.

# 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
<input type="checkbox"/>		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
<input type="checkbox"/>		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

[Red Hat Satellite Provider](#) » Add ConfiguredSystem

Request Purpose **Catalog** Customize Schedule

## 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

The screenshot displays the Ansible Tower web interface. At the top, there is a navigation bar with tabs for TOWER, PROJECTS, INVENTORIES (which is selected), TEMPLATES, and JOBS. Below the navigation bar, the page title is 'INVENTORIES'. The main content area shows a section titled 'INVENTORIES' with a count of 3. There is a search bar with the placeholder text 'SEARCH' and a search icon, followed by a 'KEY' button. Below the search bar is a table with two columns: 'NAME' and 'ORGANIZATION'. The table contains three rows of inventory data.

NAME	ORGANIZATION
aws	Default
cloudforms	Default
satellite6	Default

# Red Hat Management Automated

Satellite 6 - CloudForms - Ansible Tower - Integration Gains

Satellite 6

Hosts

Filter ... x Search

<input type="checkbox"/>	Name	Operating system
<input type="checkbox"/>	<a href="#">ip-172-31-159-178.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-165-67.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-177-77.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-238-93.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-45-59.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-47-45.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-54-120.ec2.internal</a>	RedHat 7.3
<input type="checkbox"/>	<a href="#">ip-172-31-96-218.ec2.internal</a>	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

Installable Errata

Show from: Current Environment (Dev/Ct)

Search... Showing 20 of 51 (51 Total) 20 Selected

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

- Apply Selected
  - via Katello agent
  - via remote execution
  - via remote execution - customize first

CloudForms

RED HAT® CLOUDFORMS MANAGEMENT ENGINE

Cloud Intel

Red Hat Insights

Configuration Lifecycle

Linux Ops Windows Ops

VMs & Templates

RH Insights

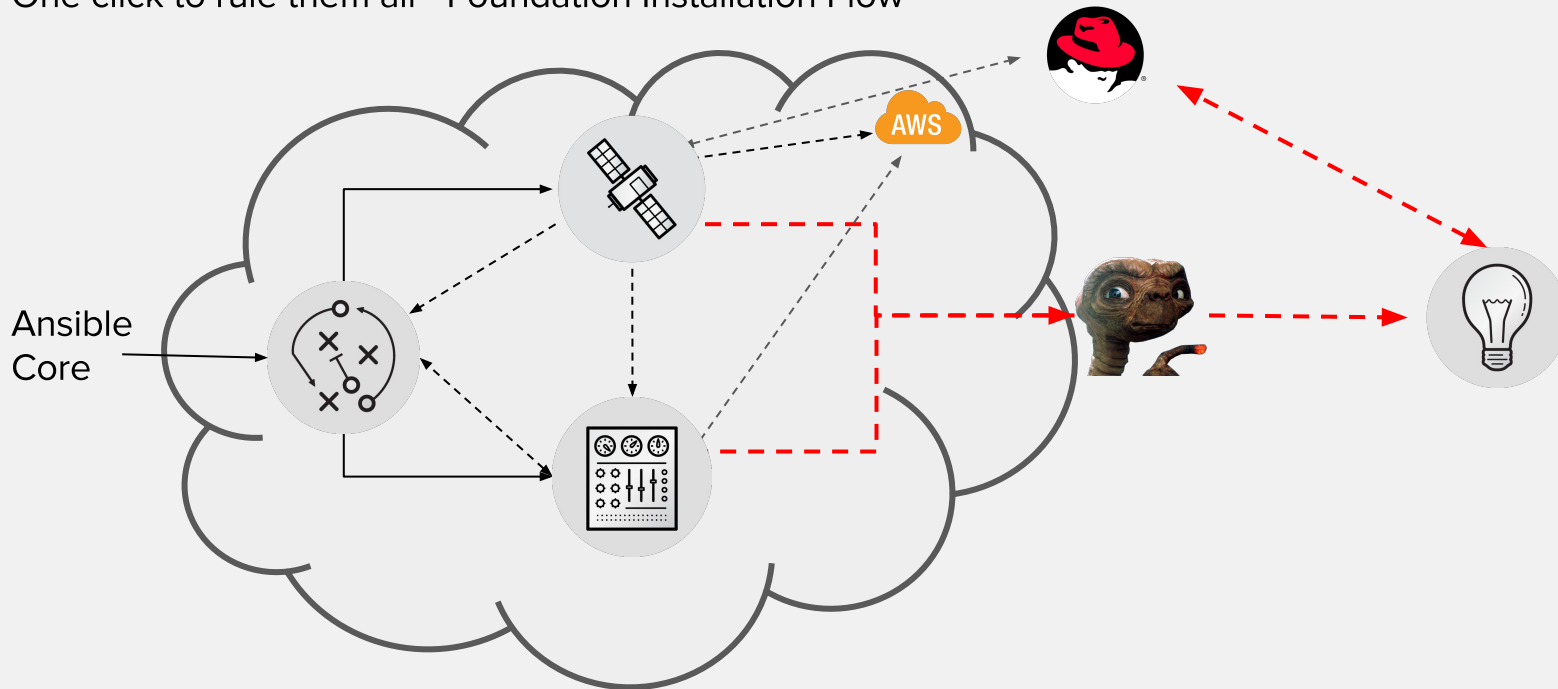
CVE-2016-0800\_DR

CVE\_2016\_5696\_KEN

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

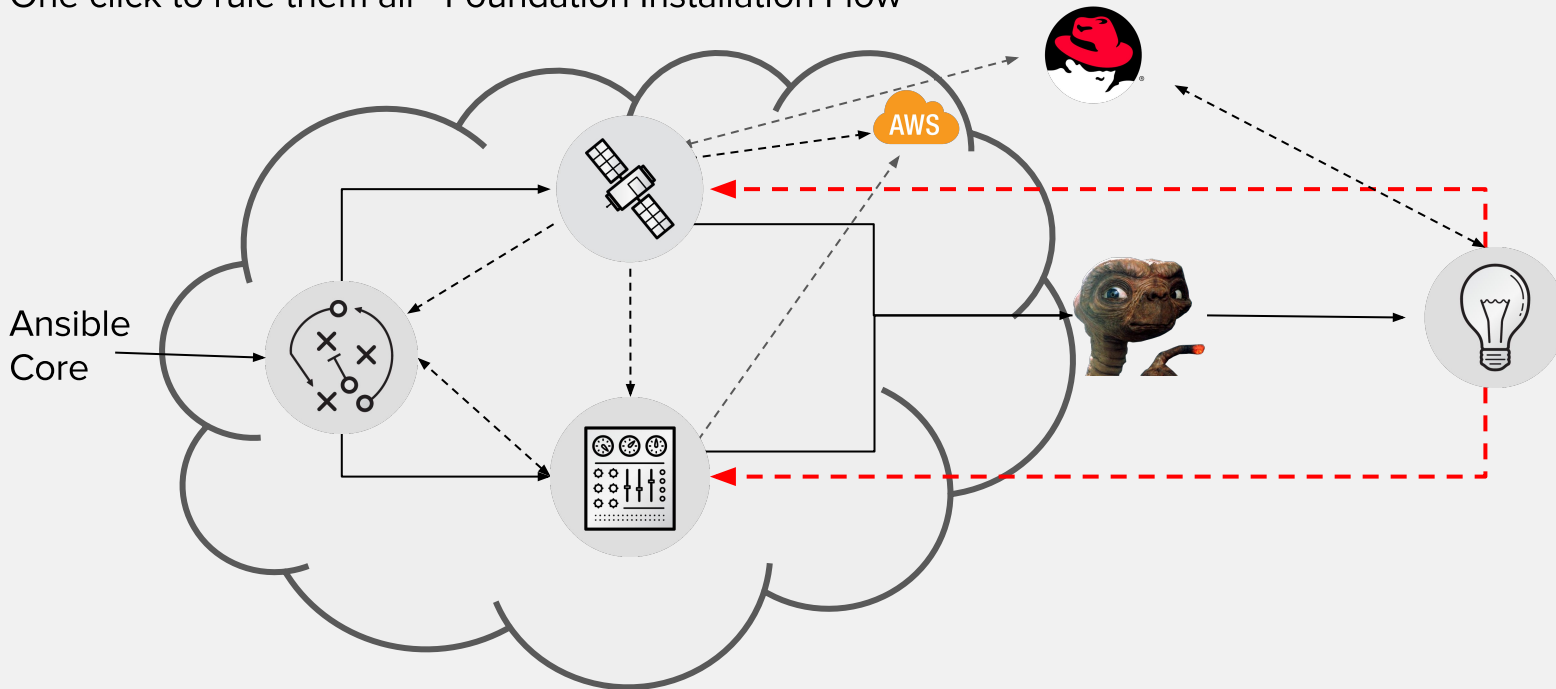
# RH-MANAGEMENT INSIGHTS

One click to rule them all - Foundation Installation Flow



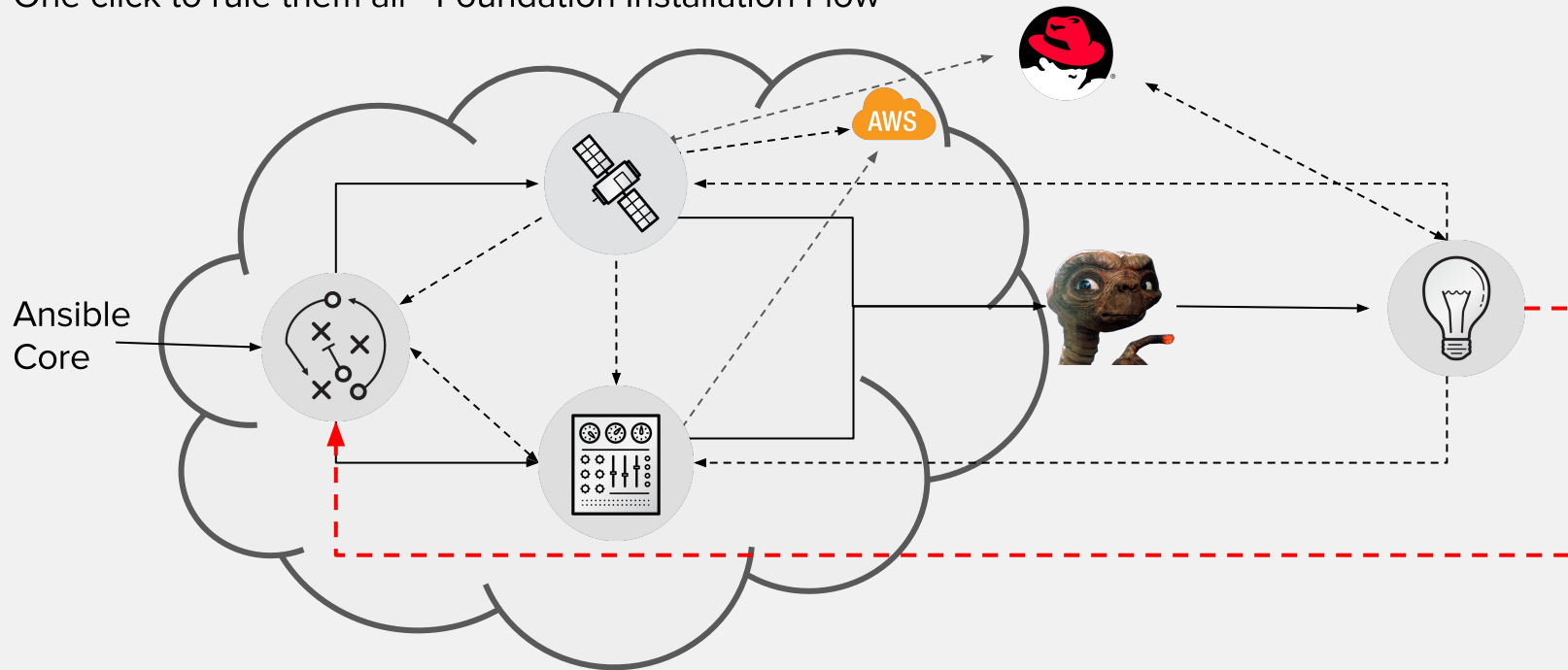
# RH-MANAGEMENT INSIGHTS

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

ALL INFO WARN ERROR

Section	Count
Security	24
Stability	4
Performance	1

3 systems are not checking in

[VIEW SYSTEMS AND RESOLVE](#)



# 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

Hostname ▲	Reported ⇅	
<input type="text" value="Filter"/>		
demo-insights-rhel65	about 9 hours ago	<a href="#">View</a>
demo-insights-rhel70.demo.mbu.redhat.com	about a month ago	<a href="#">View</a>
localhost.localdomain.localdomain	2 months ago	<a href="#">View</a>

# 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

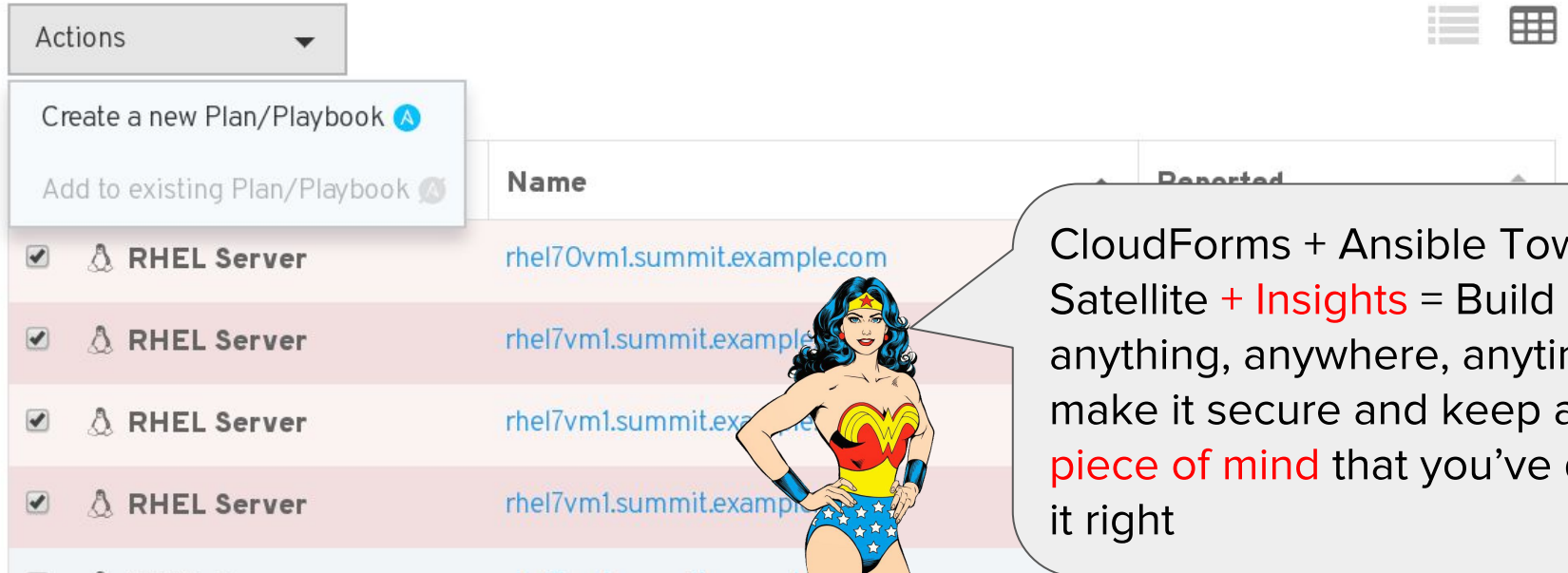
To fix this issue, Red Hat recommends that you update the deployed kernel to version **2.6.32-431.20.3.el6** or later.

```
# 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



The screenshot shows a management interface with a dropdown menu for 'Actions' and a table of systems. The dropdown menu includes options: 'Create a new Plan/Playbook' and 'Add to existing Plan/Playbook'. The table has columns for 'Name' and 'Reported'. The first four rows of the table are highlighted in pink and each contains a checkmark, a penguin icon, and the text 'RHEL Server'.

	Name	Reported
<input checked="" type="checkbox"/>	RHEL Server	rhel70vm1.summit.example.com
<input checked="" type="checkbox"/>	RHEL Server	rhel7vm1.summit.example.com
<input checked="" type="checkbox"/>	RHEL Server	rhel7vm1.summit.example.com
<input checked="" type="checkbox"/>	RHEL Server	rhel7vm1.summit.example.com



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



<http://bit.ly/2oQwxxF>

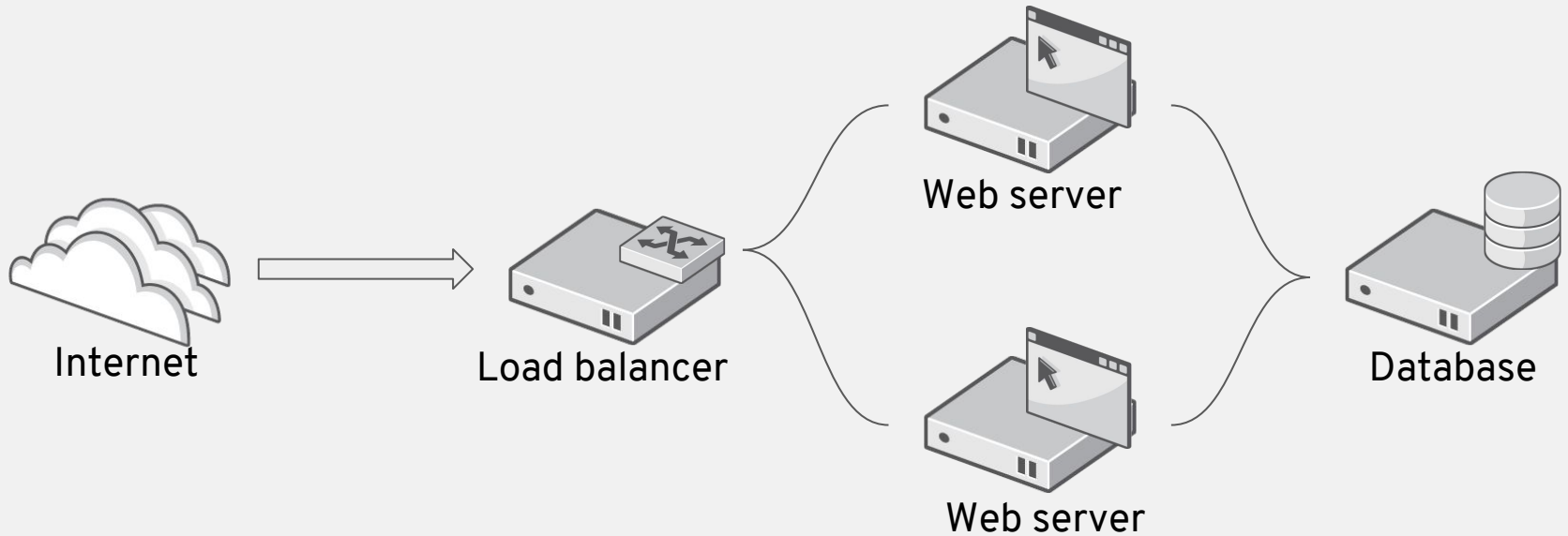
**Here's a practical example, kids!**

# 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'

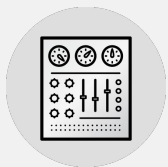
# HOW DOES THAT WORK?

- 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

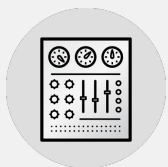
Order a service in  
the CloudForms  
self-service portal



# CLOUDFORMS SELF-SERVICE

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

Order a service in  
the CloudForms  
self-service portal

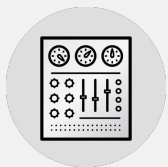


Deploy four instances  
in OpenStack

# CLOUDFORMS SELF-SERVICE

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

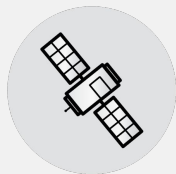
Order a service in  
the CloudForms  
self-service portal



Deploy four instances  
in OpenStack



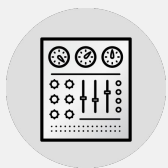
Pass control to  
Satellite for OS  
configuration, errata



# CLOUDFORMS SELF-SERVICE

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

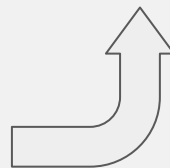
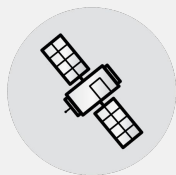
Order a service in the CloudForms self-service portal



Deploy four instances in OpenStack



Pass control to Satellite for OS configuration, errata

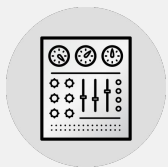


Automatically deploy Insights client as well!

# CLOUDFORMS SELF-SERVICE

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

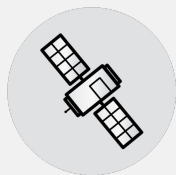
Order a service in the CloudForms self-service portal



Deploy four instances in OpenStack



Pass control to Satellite for OS configuration, errata



Pass control to Ansible Tower for application deployment

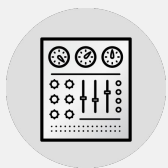


Automatically deploy Insights client as well!

# CLOUDFORMS SELF-SERVICE

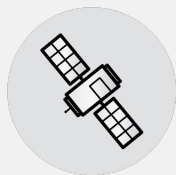
How CloudForms ties self-service, system deployment and configuration together

Order a service in the CloudForms self-service portal



Deploy four instances in OpenStack

Pass control to Satellite for OS configuration, errata



Automatically deploy Insights client as well!



Pass control to Ansible Tower for application deployment

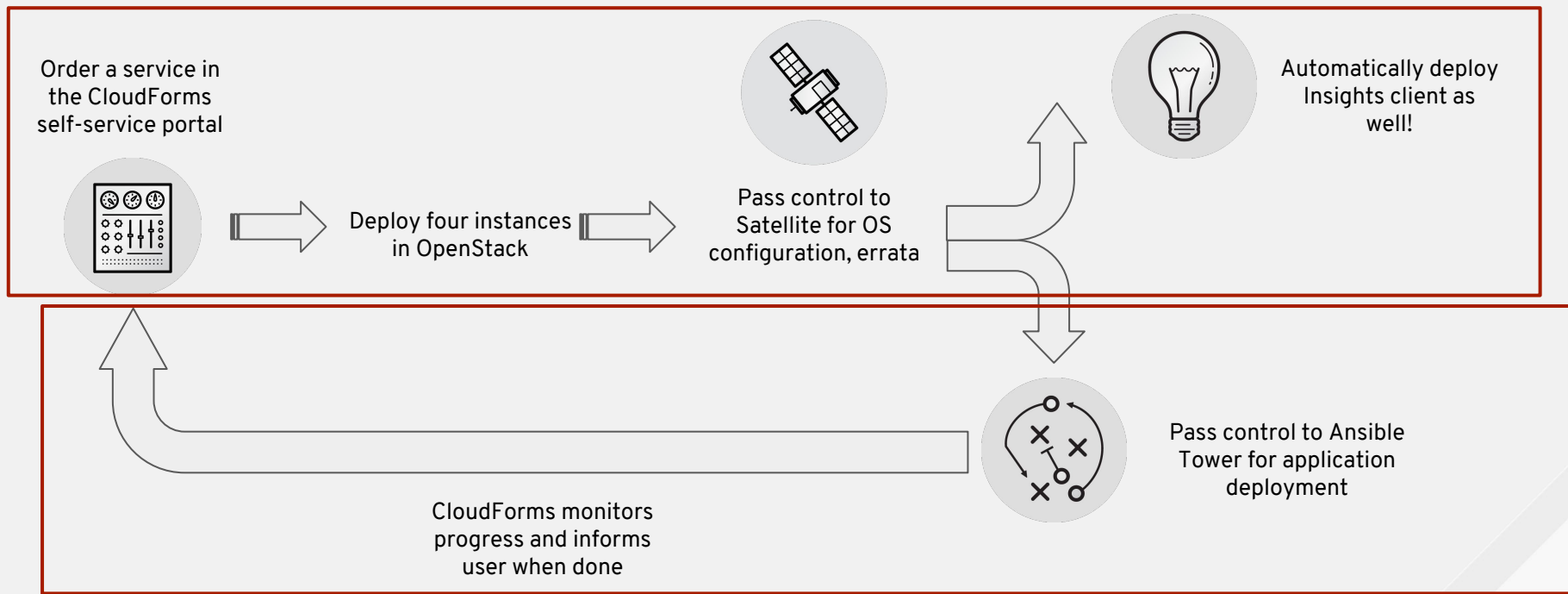


CloudForms monitors progress and informs user when done



# CLOUDFORMS SELF-SERVICE

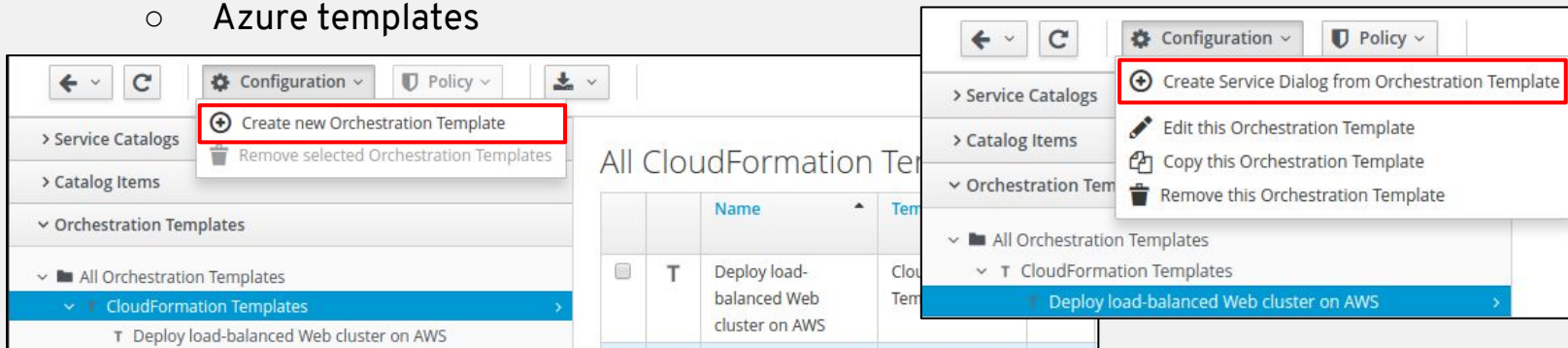
How CloudForms ties self-service, system deployment and configuration together



# How hard is that?

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

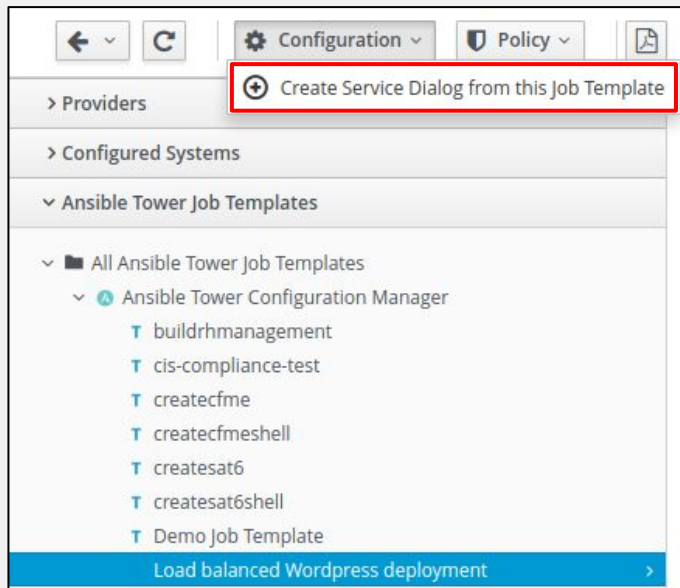
- CloudForms can consume and store:
  - Heat templates
  - CloudFormation templates
  - Azure templates
- CloudForms can automatically create dialogs from the parameters in those templates



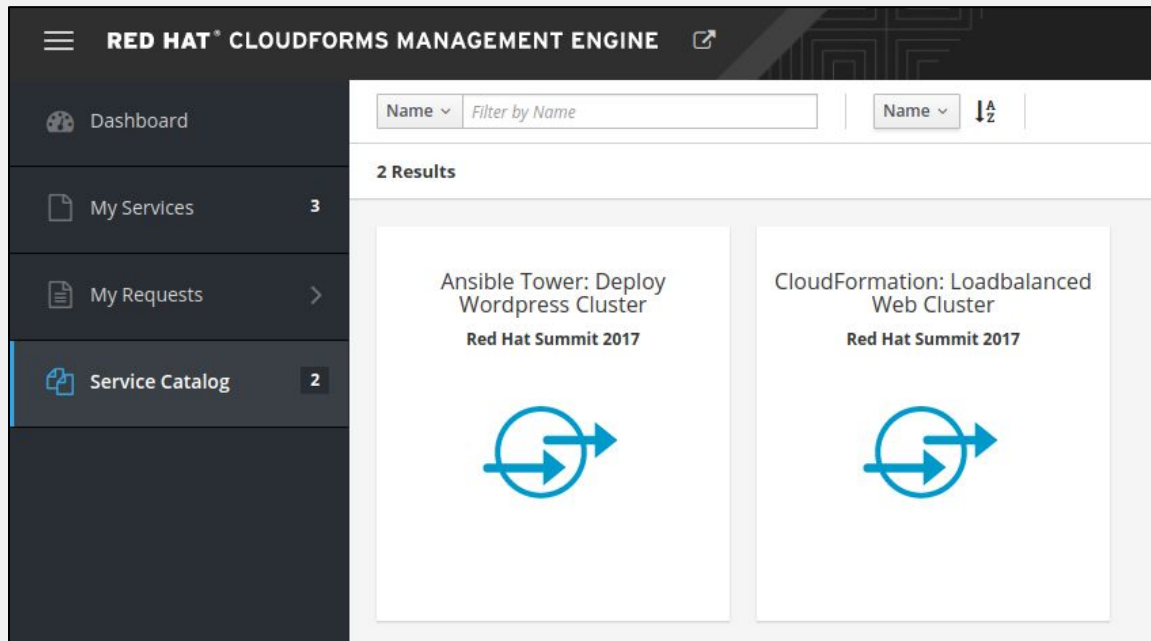
# How hard is that?

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

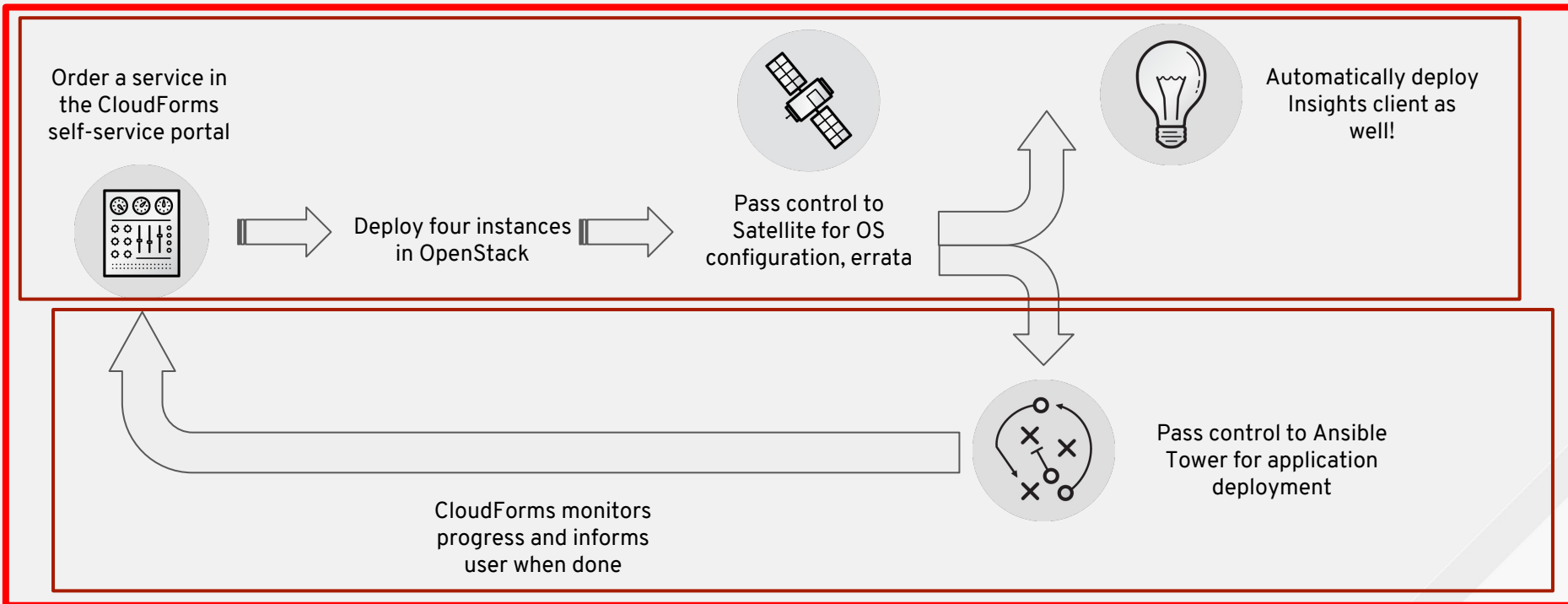


# So now we have two self service items



## This calls for a bundle!

# BUNDLING CATALOG ITEMS



# A Catalog Bundle!

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

Basic Info **Selected Resources**

### 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

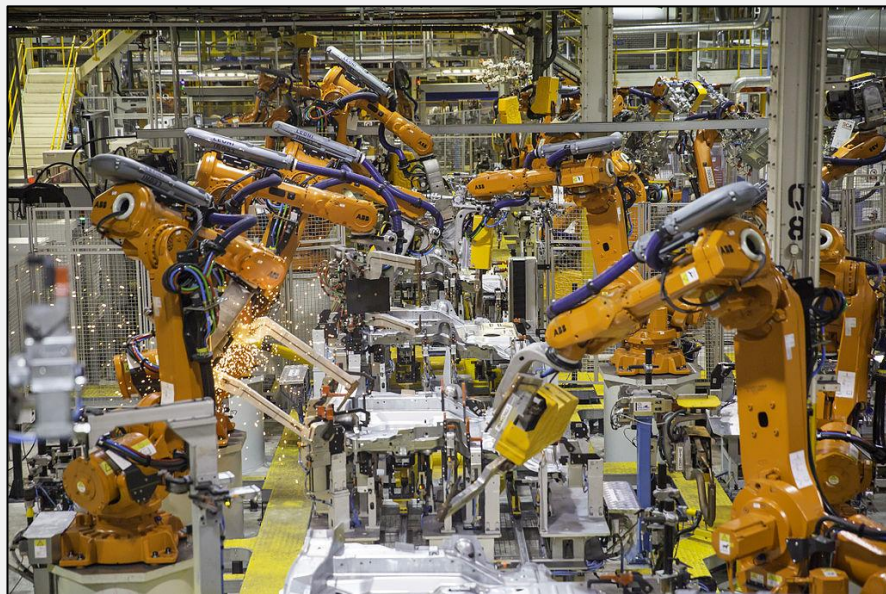
# A CatalogBundle!

The screenshot displays the Red Hat CloudForms Management Engine interface. On the left is a dark sidebar with navigation options: Dashboard, My Services (3), My Requests (>), and Service Catalog (3). The main content area has a header with a search filter 'Name Filter by Name' and a sort dropdown 'Name' with a down arrow icon. Below the header, it shows '3 Results'. Three service cards are listed:

- Ansible Tower: Deploy Wordpress Cluster**  
Red Hat Summit 2017  
Icon: A blue circular arrow with a right-pointing arrow.
- Bundle: Load-balanced Wordpress Cluster** (highlighted with a red border)  
Red Hat Summit 2017  
Icon: A white box labeled 'WORDPRESS Installation' with blue gears and the WordPress logo.
- CloudFormation: Loadbalanced Web Cluster**  
Red Hat Summit 2017  
Icon: A blue circular arrow with a right-pointing arrow.

# 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>

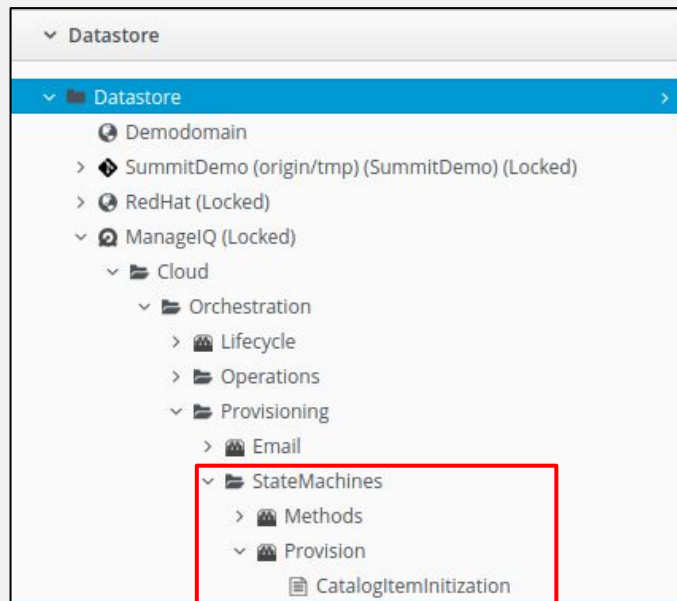
# Customizing state machines: example 2

- 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>

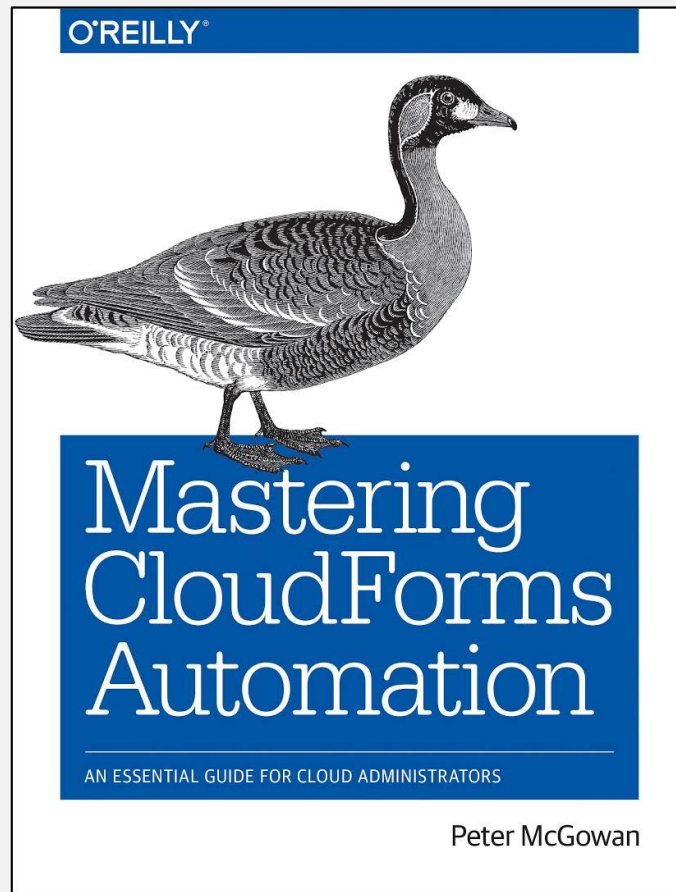
# How hard is customizing state machines?

- 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 :)



# Want to learn more?

- There is an excellent book on CloudForms automation
- It's freely available on our website
- <http://red.ht/2oYQttJ>



# 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 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!

RED HAT  
**SUMMIT**

# THANK YOU



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)

The logo consists of a red speech bubble shape pointing downwards, containing the text "RED HAT" in a smaller font above "SUMMIT" in a larger, bold font, both in white.

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)