



redhat.



Data Center Automation Made Easy With Ansible

Steffen Frömer, Technical Account Manager
Martin Reinke, Senior Solution Architect

Customer Community Meeting
Hamburg, 6 July 2017

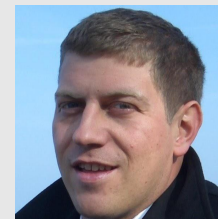


Steffen Frömer (*"coding brain"*)

Technical Account Manager @ Red Hat
Global Support Services (part of Engineering)

- Living nearby Braunschweig
Have fun with Garden, House and Barbecue
- Starting professional IT 2006 at German Aerospace Center (DLR)
- Joined Red Hat May 2016
- You can find me on Xing, LinkedIn, GitHub (Knumskull)

Who we are?



Martin Reinke (*"presenter"*)

Senior Solution Architect @ Red Hat
Manufacturing Vertical Germany

- Living nearby Hamburg (Glinde)
Loves music, travelling, sailing and have fun /w family (2 kids)
- My IT Life Started 1993 at IBM
- Infrastructure / IT Mgmt Consulting
- Software Solution Sales / Pre-sales
- Joined Red Hat Oct 2014
- I'm on Xing, LinkedIn, Twitter...

Agenda

Data Center Automation

Ansible kurze Einführung

Virtualization Use Case

- Hypervisor Management
- Virtual Machine Management

Use Case Demo

Data Center

+

Automation



Nothing routine in IT should be done manually!

Agenda

Data Center Automation

Ansible kurze Einführung

Virtualization Use Case

- Hypervisor Management
- Virtual Machine Management

Use Case Demo

WHAT IS ANSIBLE?

It's a **simple automation language** that can perfectly describe an IT application infrastructure in Ansible Playbooks.

It's an **automation engine** that runs Ansible Playbooks.

Ansible Tower is an enterprise framework for **controlling, securing and managing** your Ansible automation with a **UI, a restful API and workflow management**.



Automate the deployment and management of your entire IT footprint

DO THIS...

- Orchestration
- Configuration management
- Application deployment
- Provisioning
- Continuous delivery
- Security and compliance

ON THESE...

- Firewalls
- Load Balancers
- Applications
- Containers
- Clouds
- Servers
- Infrastructure
- Network devices
- (And more)

CLOUD

AWS
Azure
Centurylink
Cloudscale
Digital Ocean
Docker
Google
Linode
Openstack
Rackspace
+more

Ansible ships with 900+ modules

VIRT AND CONTAINER

Docker
VMware
RHEV
Openstack
OpenShift
Atomic
Cloudstack
+more

WINDOWS

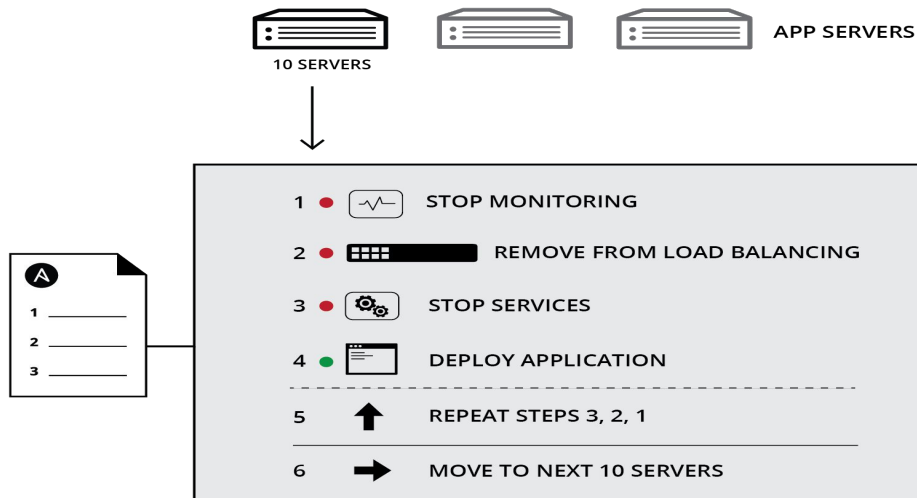
ACLs
Files
Commands
Packages
IIS
Regedit
Shell
Shares
Services
Configs
Users
Domains
+more

NETWORK

Arista
A10
Cumulus
Bigswitch
Cisco
Cumulus
Dell
F5
Juniper
Palo Alto
OpenSwitch
+more

NOTIFY

HipChat
IRC
Jabber
Email
Rocketchat
Sendgrid
Slack
Twilio
+more



Combination of configuration **and** order

Configuration Management alone is not sufficient for app deployment

Allows you to automate entire sets of process

CROSS PLATFORM – Linux, Windows, UNIX, Networks, Clouds

Agentless support for all major OS variants, **physical, virtual, cloud** and **network devices**

HUMAN READABLE – YAML

Perfectly describe and document every aspect of your application environment

PERFECT DESCRIPTION OF APPLICATION

Every change can be made by playbooks, ensuring everyone is on the same page

VERSION CONTROLLED

Playbooks are plain-text. Treat them like code in your existing version control.

DYNAMIC INVENTORIES

Capture all the servers 100% of the time, regardless of infrastructure, location, etc.

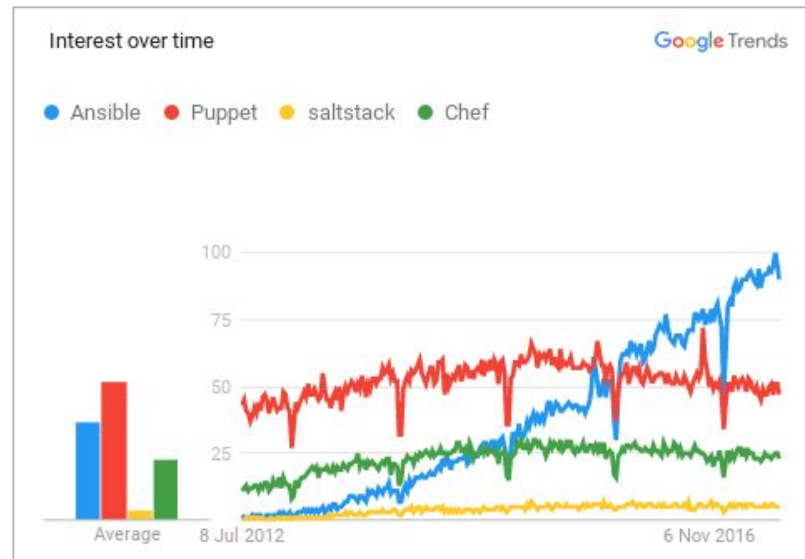
ORCHESTRATION THAT PLAYS WELL WITH OTHERS – HP SA, Puppet, Jenkins, RHNS, etc.

Homogenize existing environments by leveraging current toolsets and update mechanisms.

THE MOST POPULAR OPEN-SOURCE AUTOMATION COMMUNITY ON GITHUB

AS OF FEB 2017

- 21,000+ stars & 6,800+ forks on GitHub
- 2500+ Contributors (one of the largest open source projects communities)
- Ansible Galaxy: over 10,000 Roles
- 400,000+ downloads a month
- AnsibleFests in San Francisco, London
- Automates events globally



Source: https://twitter.com/reinke_m/status/795566055175442432



SIMPLE

Human readable automation
No special coding skills needed
Tasks executed in order
Get productive quickly



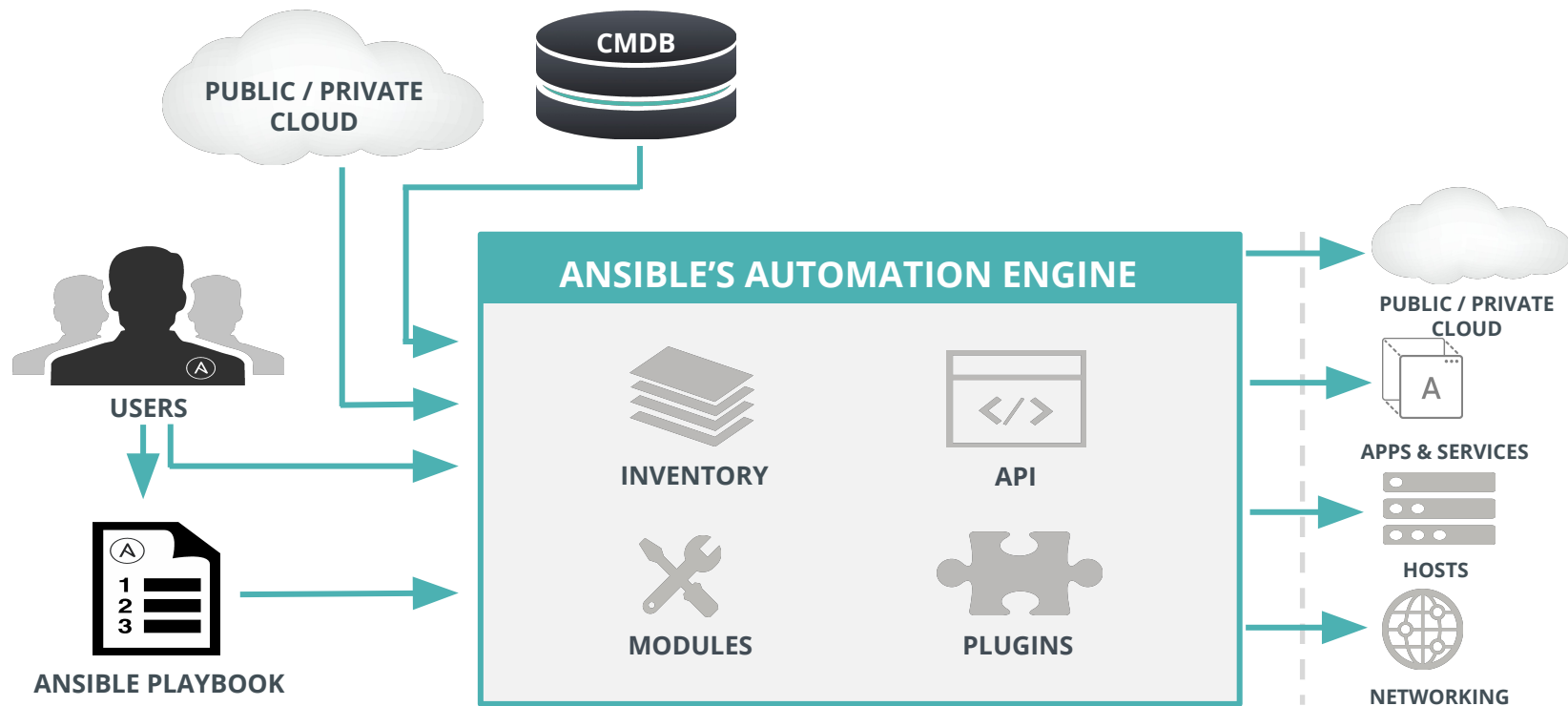
POWERFUL

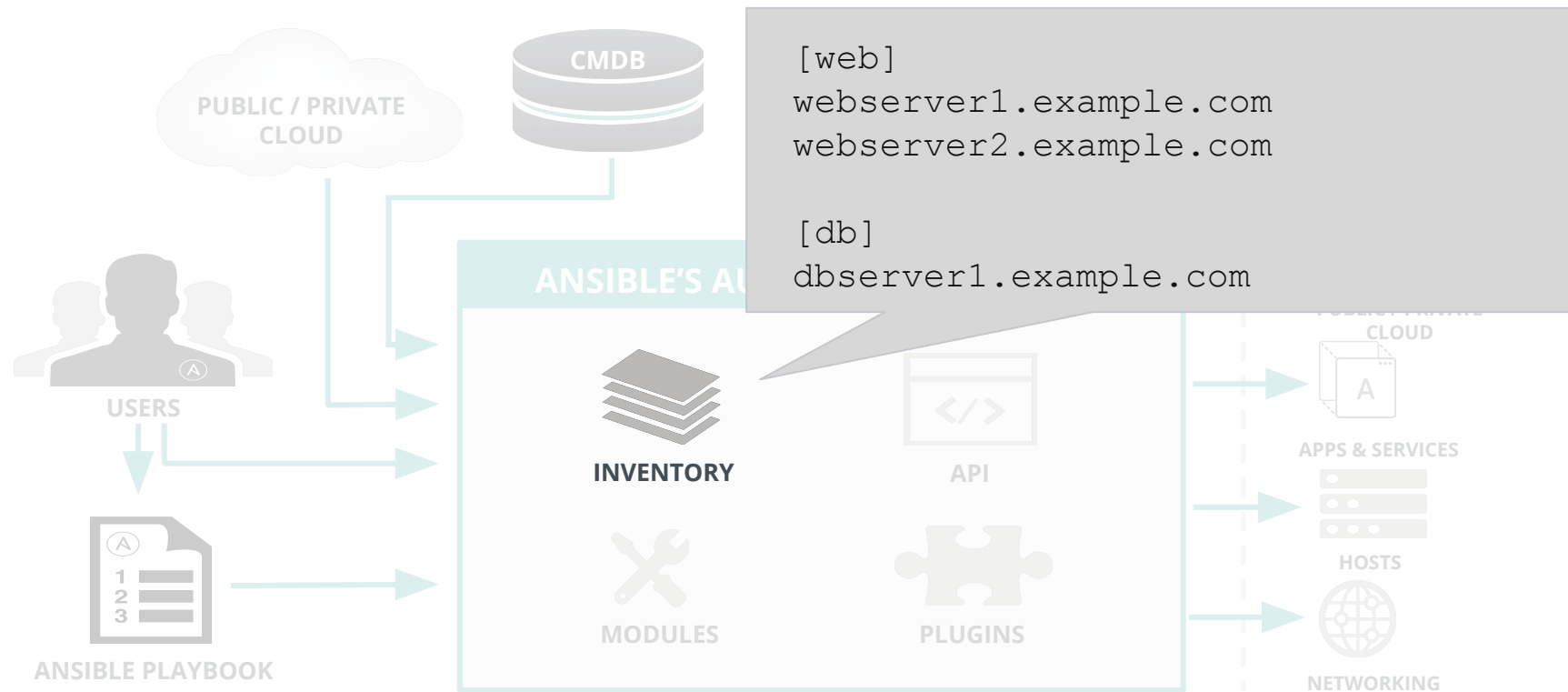
App deployment
Configuration management
Workflow orchestration
Orchestrate the app lifecycle

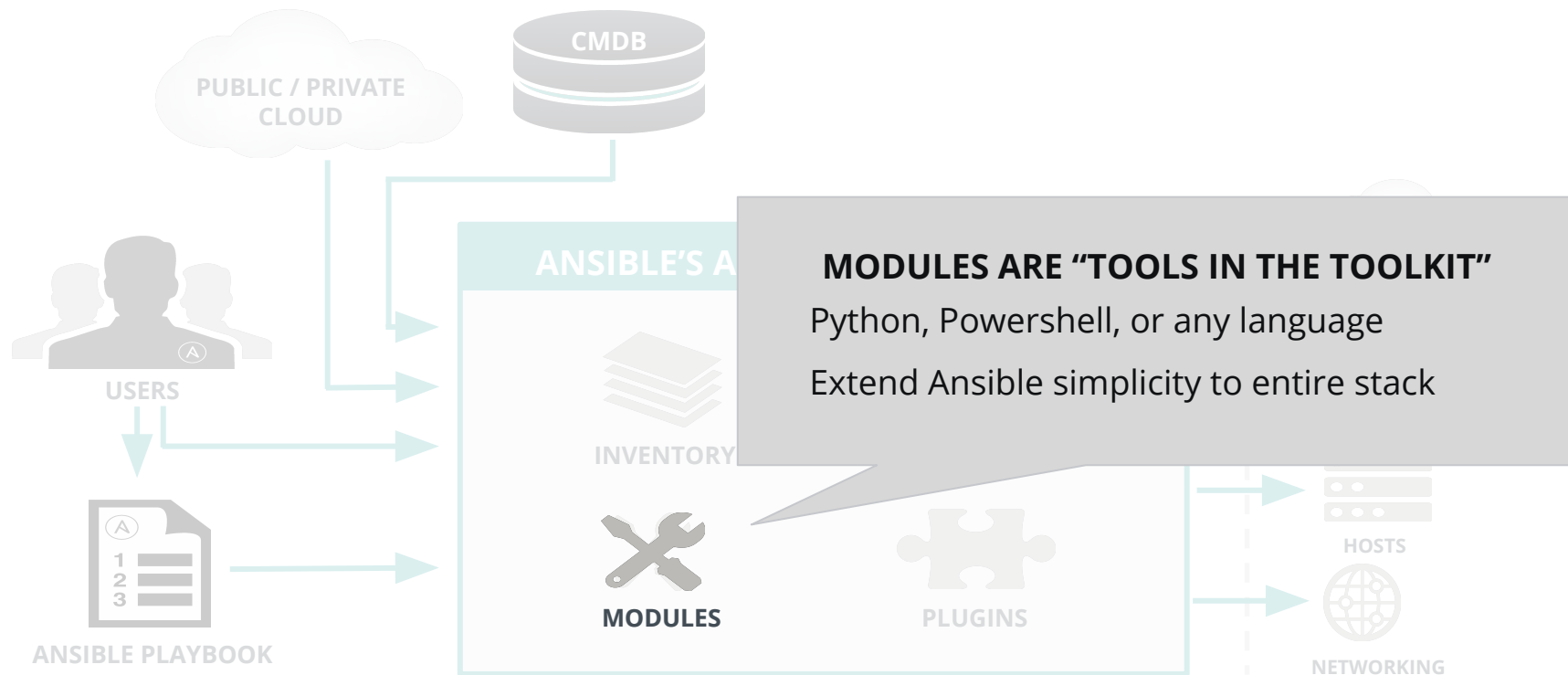


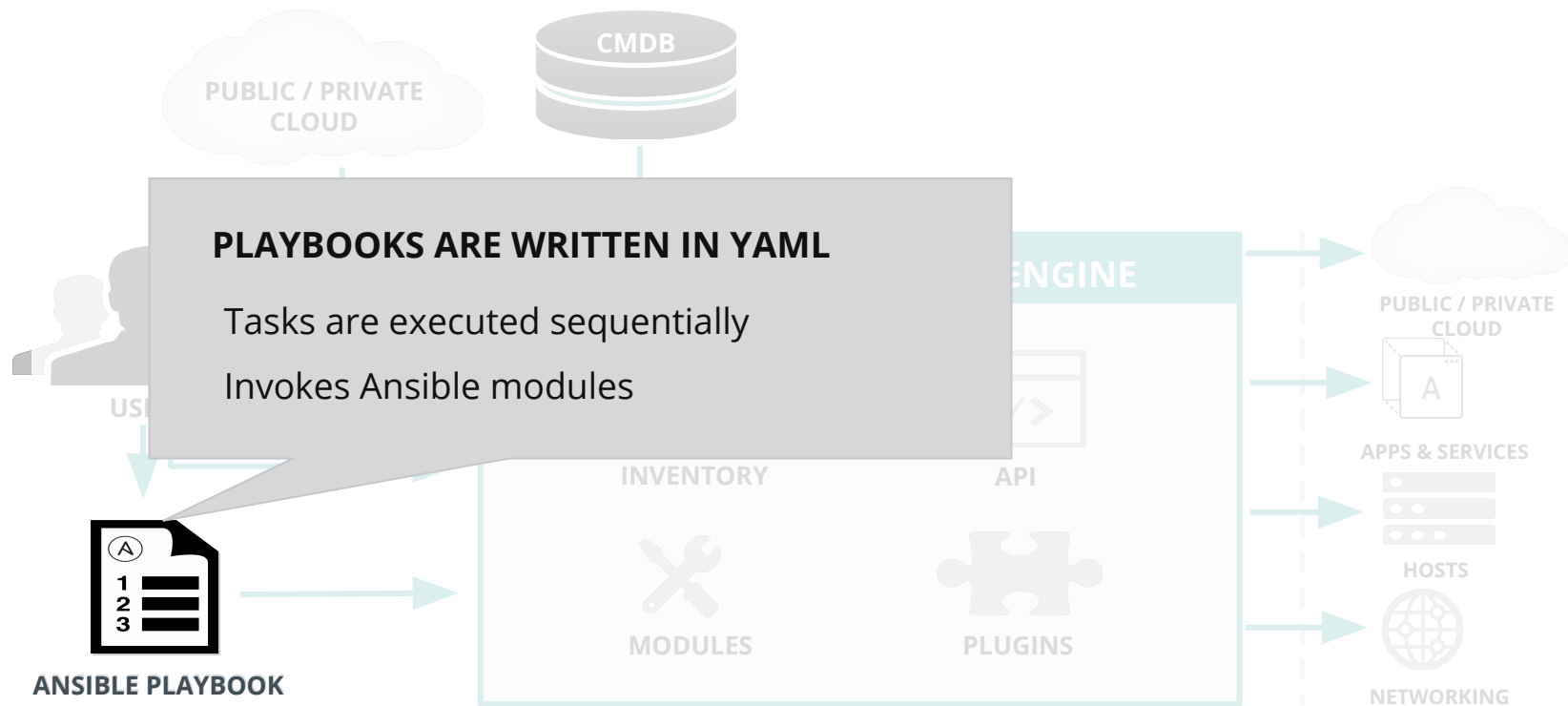
AGENTLESS

Agentless architecture
Uses OpenSSH & WinRM
No agents to exploit or update
More efficient & more secure










```
---
- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root

  tasks:
    - name: install httpd
      package: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    - name: start httpd
      service: name=httpd state=running
```

```
---  
- name: install and start apache  
  hosts: all  
  vars:  
    http_port: 80  
    max_clients: 200  
  remote_user: root  
  
  tasks:  
    - name: install httpd  
      package: name=httpd state=latest  
    - name: write the apache config file  
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf  
    - name: start httpd  
      service: name=httpd state=running
```

```
---
- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root

  tasks:
    - name: install httpd
      package: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    - name: start httpd
      service: name=httpd state=running
```

```
---
- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root

  tasks:
  - name: install httpd
    package: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
  - name: start httpd
    service: name=httpd state=running
```

What is missing?



SIMPLE



POWERFUL



AGENTLESS



CENTRAL

Central place for everyone

Overview of present and past

Create workflows, schedule jobs

Have one common view



INTEGRATION

Simple, powerful API

Uses REST for quick adoption

No special agents or lib needed

Integrate with everything



ACCESS

Teams and users enable RBAC

Deposit credentials securely

Assign access to unprivileged

Separate access and execution

WHY ANSIBLE TOWER

ANSIBLE



Building, managing dynamic inventory

Organizing admin control with users and teams

Leverage Ansible Workflows to break up tasks

- new in 3.1

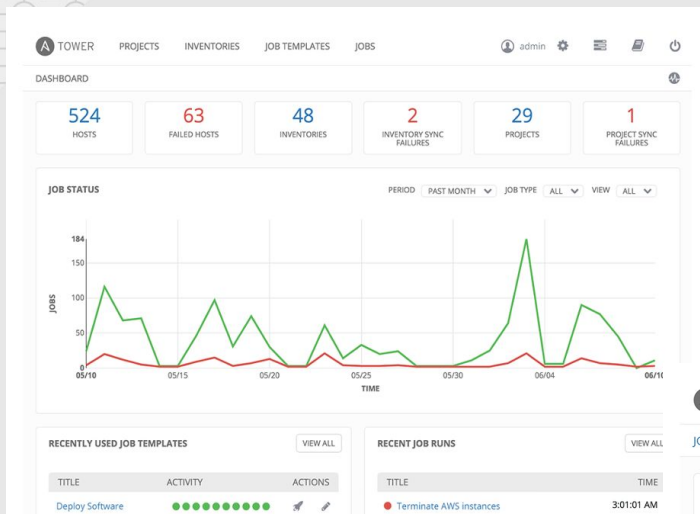
Ongoing compliance

- compare running configs to golden masters on schedules

Utilize the RESTful API for anything



DASHBOARD & LIVE VIEW



See **real-time output of your automation jobs**, and easily search through output to find exactly what you need.

Heads-up NOC-style **automation dashboard** displays everything going on in your Ansible environment.

The screenshot shows the 'RESULTS' section of a job run titled '292 - DEPLOY SOFTWARE'. The job is marked as 'Successful'. Key details include:

- STARTED: 10/3/16 16:37:21 PM
- FINISHED: 10/3/16 16:40:36 PM
- ELAPSED: 00:03:14
- PROJECT: Software deployment
- MACHINE CREDENTIAL: Staging ssh key
- TEMPLATE: Deploy software
- JOB TYPE: Run
- LAUNCHED BY: admin
- INVENTORY: Cloud staging servers
- PLAYBOOK: deploy.yml
- VERBOSITY: Default

The 'STANDARD OUT' section displays the output of the job, including task results and inventory information. The output shows that the job was successful and that the inventory was updated. The output is truncated with '...' indicating that there is more content.

SCHEDULING & NOTIFICATIONS

The screenshot shows the 'JOB TEMPLATES SCHEDULES / JOB TEMPLATE SCHEDULES.EDIT' page in the Tower interface. It features a 'DAILY REMEDIATION' section with fields for NAME (Daily remediation), START DATE (10/03/2016), START TIME (01:23:45), LOCAL TIME ZONE (America/New_York), and REPEAT FREQUENCY (Day). Below this is a 'FREQUENCY DETAILS' section with 'EVERY' (1) DAYS and 'END' (Never). A 'SCHEDULE DESCRIPTION' section shows 'every day' and a list of occurrences: 10/03/2016 01:23:45 EDT, 10/04/2016 01:23:45 EDT, and 10/05/2016 01:23:45 EDT.

Scheduled Jobs inside Tower enable you to any Job now, later, or forever.

Stay informed of your automation status via **integrated notifications**. Connect Slack, Hipchat, SMS, email, and more.



#prodOps Notification
Prod Ops Complete!

The screenshot shows the 'NOTIFICATION TEMPLATES' management page. It includes a '+ ADD' button, a search bar with 'NAME' and 'SEARCH' fields, and a table with columns for 'NAME' and 'ACTIONS'. One notification template is listed: 'Prod Ops Complete', with icons for a bell, edit, and delete. The page footer indicates 'ITEMS 1-1 OF 1'.

SELF SERVICE & WORKFLOWS

LAUNCH JOB | DEPLOY SOFTWARE

INVENTORY CREDENTIAL SURVEY

*ENTER NUMBER OF SERVICE INSTANCES.
2

*PLEASE SELECT THE SERVICE OWNER.
Alice

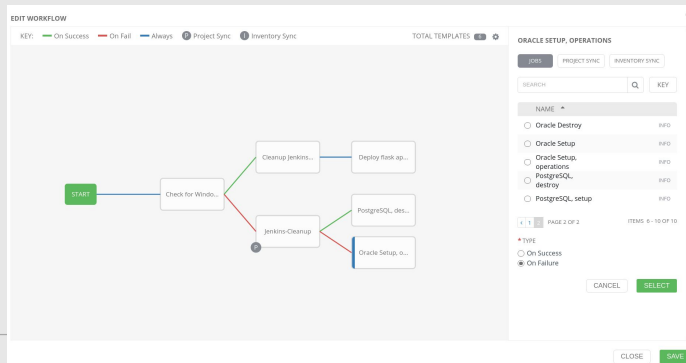
*ENTER PASSWORD FOR DEPLOYED CERTIFICATE.
SHOW [REDACTED]

INVENTORY CREDENTIAL
Cloud staging servers Staging ssh key

CANCEL LAUNCH

Tower's **multi-Playbook workflows** chains any number of Playbooks together to create a single workflow. Different Jobs can be run depending on success or failure of the prior Playbook.

With **self-service**, Tower lets you launch Playbooks with just a single click. It can prompt you for variables, let you choose from available secure credentials and monitor the resulting deployments.



Agenda

Data Center Automation

Ansible kurze Einführung

Virtualization Use Case & Demo

- Hypervisor Management
- Virtual Machine Management

Virtualization Use Case

Hypervisor Management for RHEV/RHV

Planning and operations automation tasks


- Installation
- Configuration
- Upgrade of hypervisors (not RHEV-M)

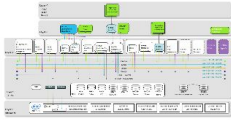
Go further beyond hypervisors and virtual server (outlook)


Introduction to the demo environment



- internal environment to test, run and demonstrate Red Hat multi product integrated solutions
- Wide area of cloud usages scenario's
 - RHEL/RHV, OpenStack, OpenShift, JBOSS Middleware, CloudForms, Ansible Tower, Satellite ++
- chosen for our demo purposes today:
 - Red Hat Virtualization (part only) with some Ansible Automation

PROJECT 'HAILSTORM'
Take your customer on a journey: A day in the life with Red Hat Cloud

DEMO STORYLINE AND COLLATERALS


SOLUTION ARCHITECTURE


ANSIBLE AUTOMATION


 3 

CONFIDENTIAL - FOR INTERNAL USE ONLY

You can see/follow our entire project on GitHub
<https://github.com/wrichter/hailstorm>

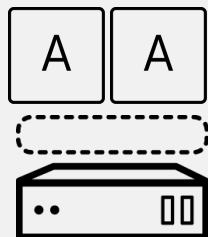
What is Red Hat Virtualization (RHV)?

What is RHV?

More than just KVM

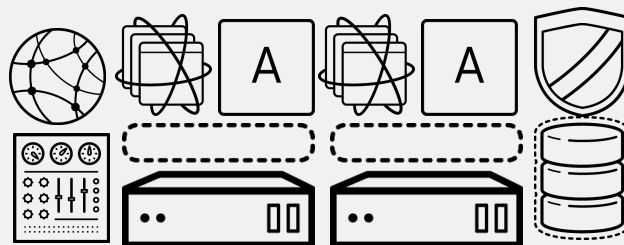
RHEL + KVM

- Basic support for KVM hypervisor
- No enterprise virtualization management features
- Limited number of VMs allowed



Red Hat Virtualization

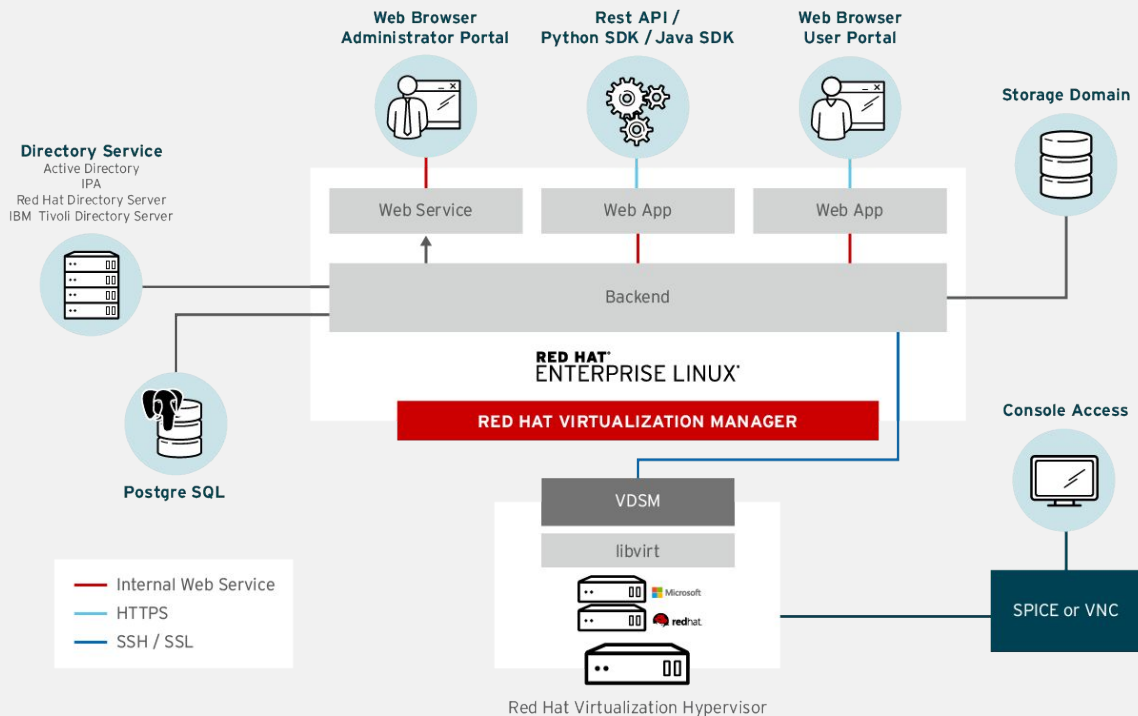
- Build on RHEL+KVM
- Centralized Management for the KVM hypervisor as well as compute, network, and storage resources
- Enterprise features to support mission critical applications



https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html-single/technical_reference/

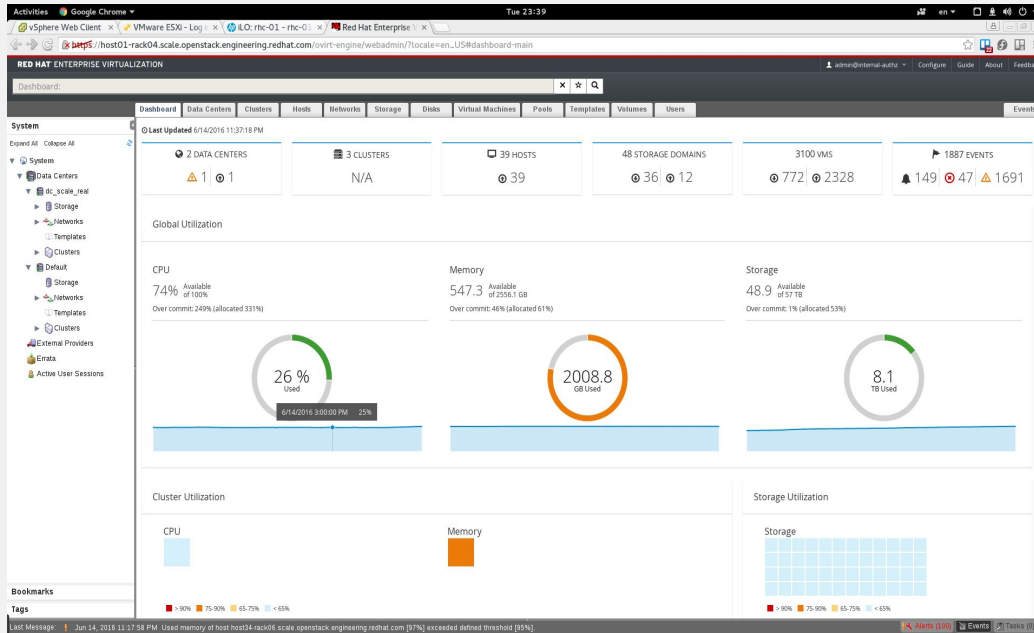
What is RHV?

Overview of RHV



RHV Features

RHV Manager (RHV-M)



- Centralized management of virtualized infrastructure (hosts / servers, VMs, templates, networks, data warehouses)
- Suitable for large installations (500+ hosts and 5,000+ VMs)
- Intuitive dashboard with detailed information

Some more RHV Features

Not comprehensive overview

Core	Live Migration	HA VMs	Affinity / Anti-Affinity	CPU Pinning
	NUMA Support	Resource Reservation	Large Page support	Memory page sharing
	VM templates	Hot Add Memory & CPU	Migrate/Import VMs	Overcommit
Network	VLAN tagging	PCI Passthrough	OVS (tech preview)	Network QoS
	NIC bonding	Jumbo frames	IPV6 (guest)	Network labels
Storage	Storage Live Migration	Thin/Thick Provisioning	Live Snapshots/Merge	SCSI, NFS, FC, GlusterFS
Security	RBAC & Tiered Access	Firewall/SELinux		
Mgmt & APIs	Browser Based Mgmt	Python & Java SDKs	REST APIs	

Overview of the demo

RHV environment setup steps

1. Provision base RHEL Server for hypervisors
Provision RHEL Server for RHEV Manager
2. Install RHEV Manager
3. Provision Hypervisors, Cluster, Storage etc.
4. Provision Virtual Machines in Hypervisors
5. Do some updates on VMs

Already done (prep work)

Show Ansible playbooks at work

Show manual Ansible work
using ad-hoc commands

Demo introduction

Some screenshots from the prep work

- Joined re-work of ansible automation of the RHEV environment from scratch
- Short / limited time for writing the playbooks
→ done by Steffen in 4 h (+ reading / understanding >1 day)
- ½ day internal dry run
- and ... Here we are !

Before - no RHV environment exists

Before Install

```
root@storm2:~  
File Edit View Search Terminal Help  
[mreinke@mreinke-t540 ansible]$ ssh -i binary/hailstorm root@storm2.coe.muc.redhat.com  
Last login: Mon Jul 3 15:08:18 2017 from ovpn-117-185.ams2.redhat.com  
Kickstarted on 2016-05-19  
[root@storm2 ~]# virsh list  
-----  
Id Name State  
-----  
1 satellite running  
3 ipa running  
5 infrastructure running  
17 ceph1 running  
18 ceph0 running  
19 ceph2 running  
34 storage-console running  
35 tower running  
65 efk running  
97 ose3-lb running  
98 ose3-master2 running  
99 ose3-master3 running  
100 ose3-master1 running  
101 ose3-node2 running  
102 ose3-infranode1 running  
103 ose3-node1 running  
105 rhosp-director running  
117 rhosp-computel running  
118 rhosp-control1 running  
119 rhosp-compute3 running  
120 rhosp-control2 running  
121 rhosp-control3 running  
124 dev-client running  
[root@storm2 ~]# virsh list | grep rhev*  
[root@storm2 ~]#
```

After Install

```
root@storm2:~  
File Edit View Search Terminal Help  
[root@storm2 ~]# virsh list  
-----  
Id Name State  
-----  
1 satellite running  
3 ipa running  
5 infrastructure running  
17 ceph1 running  
18 ceph0 running  
19 ceph2 running  
34 storage-console running  
35 tower running  
65 efk running  
97 ose3-lb running  
98 ose3-master2 running  
99 ose3-master3 running  
100 ose3-master1 running  
101 ose3-node2 running  
102 ose3-infranode1 running  
103 ose3-node1 running  
105 rhosp-director running  
117 rhosp-computel running  
118 rhosp-control1 running  
119 rhosp-compute3 running  
120 rhosp-control2 running  
121 rhosp-control3 running  
124 dev-client running  
193 rhevm running  
194 rhevh3 running  
195 rhevh1 running  
196 rhevh2 running  
[root@storm2 ~]# virsh list | grep rhev  
193 rhevm running  
194 rhevh3 running  
195 rhevh1 running  
196 rhevh2 running  
[root@storm2 ~]#
```

Setup plain RHEL systems for 3 HVs and 1 RHEV-M

- Pre-configured RHELs
- Registered at Satellite
- Repo config done

```
File Virtual Machine View Send Key
[ OK ] Reached target Timers.
[ OK ] Reached target Paths.
[ OK ] Reached target Basic System.
Starting Dump dmesg to /var/log/dmesg...
[ OK ] Started D-Bus System Message Bus.
Starting D-Bus System Message Bus...
Starting Authorization Manager...
Starting NTP client/server...
Starting Login Service...
Starting Permit User Sessions...
Starting OpenSSH Server Key Generation...
[ OK ] Started irbalance daemon.
Starting irbalance daemon...
[ OK ] Started Dump dmesg to /var/log/dmesg.
[ OK ] Started Permit User Sessions.
Starting Terminate Plymouth Boot Screen...
[ OK ] Started Command Scheduler.
Starting Command Scheduler...
Starting Wait for Plymouth Boot Screen to Quit...
[ OK ] Started Login Service.
[ OK ] Started Authorization Manager.
Starting firewalld - dynamic firewall daemon...
[ OK ] Started iptables (C) 2000-2006 Netfilter Core Team
13.997437] Ebttables v2.0 registered
14.032789] nf_conntrack version 0.5.0 (65536 buckets, 262144 max)
14.034937] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
14.037283] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
14.040414] IPv6: ADDRCONF(NETDEV_UP): eth2: link is not ready
[ OK ] Sta
14.042396] IPv6: ADDRCONF(NETDEV_UP): eth3: link is not ready
Sta
14.044614] IPv6: ADDRCONF(NETDEV_UP): eth4: link is not ready
[ OK ] Sta
14.089597] bridge: automatic filtering via arp/ip/iptables has been deprecated. Update your scr
pts to load br_netfilter if you need this.
Sta
14.104292] Netfilter messages via NETLINK v0.30.
Sta
14.107732] ip_set: protocol 6
Sta
Sta Red Hat Enterprise Linux Server 7.3 (Maipo)
Sta Kernel 3.10.0-514.el7.x86_64 on an x86_64
rhev1 login: [
[ OK ] Sta
[ OK ] Sta r/authorization manager.
[ OK ] Started Login Service.
Starting firewalld - dynamic firewall daemon...
Starting Terminate Plymouth Boot Screen...
[ OK ] Started Command Scheduler.
Starting Command Scheduler...
Starting Wait for Plymouth Boot Screen to Quit...
[ OK ] Started NTP client/server.
[ OK ] Started Login Service.
Starting Authorization Manager.
Starting firewalld - dynamic firewall daemon...
[ OK ] Started iptables (C) 2000-2006 Netfilter Core Team
13.997437] Ebttables v2.0 registered
14.032789] nf_conntrack version 0.5.0 (65536 buckets, 262144 max)
14.034937] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
14.037283] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
14.040414] IPv6: ADDRCONF(NETDEV_UP): eth2: link is not ready
14.042396] IPv6: ADDRCONF(NETDEV_UP): eth3: link is not ready
14.114752] IPv6: ADDRCONF(NETDEV_UP): eth3: link is not ready
14.118007] IPv6: ADDRCONF(NETDEV_UP): eth4: link is not ready
14.159425] bridge: automatic filtering via arp/ip/iptables has been deprecated. Update your scr
pts to load br_netfilter if you need this.
14.170657] Netfilter messages via NETLINK v0.30.
14.174205] ip_set: protocol 6
Sta
Sta Red Hat Enterprise Linux Server 7.3 (Maipo)
Kernel 3.10.0-514.el7.x86_64 on an x86_64
rhev2 login: [
[ OK ] Started Command Scheduler.
Starting Command Scheduler...
Starting Terminate Plymouth Boot Screen...
[ OK ] Started NTP client/server.
[ OK ] Started Login Service.
Starting Authorization Manager.
Starting firewalld - dynamic firewall daemon...
[ OK ] Started iptables (C) 2000-2006 Netfilter Core Team
13.991912] Ebttables v2.0 registered
13.963815] nf_conntrack version 0.5.0 (65536 buckets, 262144 max)
12.993223] nf_conntrack version 0.5.0 (65536 buckets, 262144 max)
14.004028] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
14.008516] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
14.012070] IPv6: ADDRCONF(NETDEV_UP): eth2: link is not ready
14.015763] IPv6: ADDRCONF(NETDEV_UP): eth3: link is not ready
14.019651] IPv6: ADDRCONF(NETDEV_UP): eth4: link is not ready
14.047195] bridge: automatic filtering via arp/ip/iptables has been deprecated. Update your scr
pts to load br_netfilter if you need this.
14.064918] Netfilter messages via NETLINK v0.30.
14.069984] ip_set: protocol 6
Sta
Sta Red Hat Enterprise Linux Server 7.3 (Maipo)
Kernel 3.10.0-514.el7.x86_64 on an x86_64
rhev3 login: [
```

Install RHEV-M

- Create Installation answer file
- Run setup
- after RHEV-M Installation Management UI and API's are working

```
mreinke@mreinke-t540:VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
File Edit View Search Terminal Help
[mreinke@mreinke-t540 ansible]$ pwd
/VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
[mreinke@mreinke-t540 ansible]$ ansible-playbook -i hosts -e @config/hailstorm_co
nfig.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.muc.redhat.co
m.yml rhv-setup.yml

PLAY [rhvm] *****

TASK [layer2_rhvm_ansible : Copy Answer File] *****
changed: [rhvm]

TASK [layer2_rhvm_ansible : Run RHV setup] *****
changed: [rhvm]

TASK [layer2_rhvm_ansible : create .ovirtshellrc (for RHEV 4.x)] *****
changed: [rhvm]

PLAY RECAP *****
rhvm                               : ok=3    changed=3    unreachable=0    failed=0

[mreinke@mreinke-t540 ansible]$
[mreinke@mreinke-t540 ansible]$
```

Blank RHEV Web User Interface

but no hypervisors are managed by it yet

The screenshot displays the RHEV Web User Interface. The top navigation bar includes the title "RED HAT VIRTUALIZATION" and a user profile for "admin@internal-authz". The main dashboard area shows a "Last Updated" timestamp of "7/3/2017, 5:31:44 PM GMT+2". A row of six summary cards displays the following metrics: 1 Data Centers (with a warning icon), 1 Clusters (N/A), 0 Hosts (with a green checkmark), 0 Data Storage Domains (with a green checkmark), 0 Virtual Machines (with a green checkmark), and 0 Events (with a green checkmark). Below this is a "Global Utilization" section with three sub-cards for CPU, Memory, and Storage, all showing 0.0% or 0.0 MIB Used. The bottom of the dashboard features "Cluster Utilization" and "Storage Utilization" sections, which are currently empty. The left sidebar contains a navigation menu with categories like System, Data Centers, Clusters, and External Providers. The bottom status bar shows a "Last Message" from "User admin@internal-authz" and notification icons for Alerts, Events, and Tasks.



Sounds like a piece of cake

YES!

...but as always, there were some obstacles overcome...

- **Ansible version updates**
(always check for, if you want to use latest & greatest)
- **Environment related problems**
(libvirtd restart required due to nested virtualization setup)
- **Subscriptions expiration**
(needed to renew most of the subscriptions on monday)

LET'S START TO DO DEMO ...

SHOWTIME

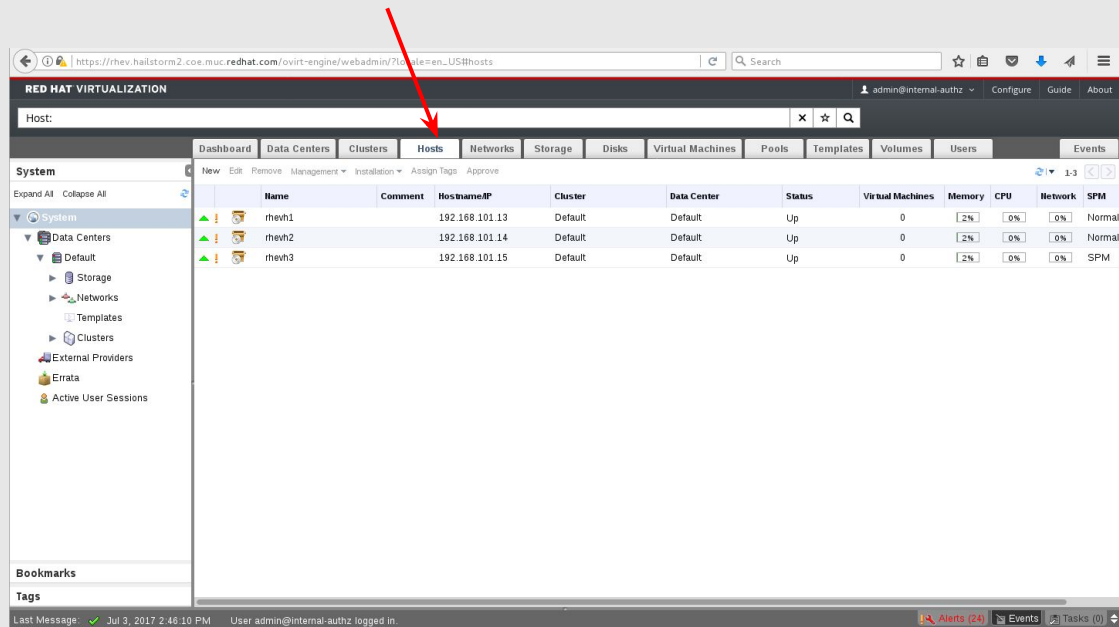


Successfully create RHV environment with 3 HVs

Make 3 RHEL servers connected into RHEV-M as new hypervisors,

Show:

1. RHEV documentation [1]
2. Ansible Playbooks



The screenshot shows the Red Hat Virtualization (RHV) web interface. The browser address bar displays the URL: https://rhev-halstorm2.coe.muc.redhat.com/ovirt-engine/webadmin/?locale=en_US&hosts. The interface is titled "RED HAT VIRTUALIZATION" and shows the user "admin@internal-authz". The "Hosts" tab is selected, displaying a table of hosts. A red arrow points to the "Hosts" tab in the navigation menu.

Name	Comment	Hostname/IP	Cluster	Data Center	Status	Virtual Machines	Memory	CPU	Network	SPM
rhev1		192.168.101.13	Default	Default	Up	0	2%	0%	0%	Normal
rhev2		192.168.101.14	Default	Default	Up	0	2%	0%	0%	Normal
rhev3		192.168.101.15	Default	Default	Up	0	2%	0%	0%	SPM

Links:

[1] - https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html-single/installation_guide/

Successfully create RHV environment with 3 HVs

Ansible Playbook and sample output

```
mreinke@mreinke-t540/VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
File Edit View Search Terminal Help
[mreinke@mreinke-t540 ansible]$ cat 03-provision-rhv-hv
#!/bin/bash

# Setup Datacenter and Clusterconfiguration
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.m
uc.redhat.com.yml rhv-deploy-cluster.yml

# Add Hypervisors to RHV cluster
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.m
uc.redhat.com.yml rhv-deploy-hv.yml

# Setup Datacenter and Clusterconfiguration
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.m
uc.redhat.com.yml rhv-deploy-storage.yml

# Setup Datacenter and Clusterconfiguration
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.m
uc.redhat.com.yml rhv-deploy-templates.yml
[mreinke@mreinke-t540 ansible]$ cat rhv-deploy-cluster.yml
...
- hosts: rhvm
  gather_facts: false

vars:
  engine_url: "https://{{ rhv_dns_domain }}/ovirt-engine/api"
  username: admin@internal

  engine_cafile: /etc/pki/ovirt-engine/ca.pem
  datacenter: Default
  cluster: Default

tasks:
- block:
  - name: obtain SSO token
    no_log: true
    oVirt_auth:
      url: "{{ engine_url }}"
      username: "{{ username }}"
      password: "{{ root_password }}"
      ca_file: "{{ engine_cafile }}"

  - name: Ensure Datacenter "{{ datacenter }}" exists
    ovirt_datacenters:
      auth: "{{ ovirt_auth }}"
      name: "{{ datacenter }}"
      comment: "The primary datacenter"
      compatibility version: 4.1
      quota_mode: enabled
      local: False

  - name: Ensure cluster "{{ cluster }}" exists
    ovirt_clusters:
      auth: "{{ ovirt_auth }}"
```

```
mreinke@mreinke-t540/VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
File Edit View Search Terminal Help
[mreinke@mreinke-t540 ansible]$ ./03-provision-rhv-hv
[WARNING]: Found both group and host with same name: storage-console
[WARNING]: Found both group and host with same name: lookbusy-rhev
[WARNING]: Found both group and host with same name: cloudforms
[WARNING]: Found both group and host with same name: ipa
[WARNING]: Found both group and host with same name: test-rhel6
[WARNING]: Found both group and host with same name: dev-client
[WARNING]: Found both group and host with same name: infrastructure
[WARNING]: Found both group and host with same name: rhvm
[WARNING]: Found both group and host with same name: satellite
[WARNING]: Found both group and host with same name: ose3-lb
[WARNING]: Found both group and host with same name: lookbusy-osp
[WARNING]: Found both group and host with same name: layer1
[WARNING]: Found both group and host with same name: efk
[WARNING]: Found both group and host with same name: tower
[WARNING]: Found both group and host with same name: rhosp-director
[WARNING]: Found both group and host with same name: test-rhel7

PLAY [rhvm] *****
TASK [obtain SSO token] *****
ok: [rhvm]

TASK [Ensure Datacenter "Default" exists] *****
changed: [rhvm]

TASK [Ensure cluster "Default" exists] *****
changed: [rhvm]

TASK [Revoke the SSO token] *****
ok: [rhvm]

PLAY RECAP *****
rhvm : ok=4 changed=2 unreachable=0 failed=0

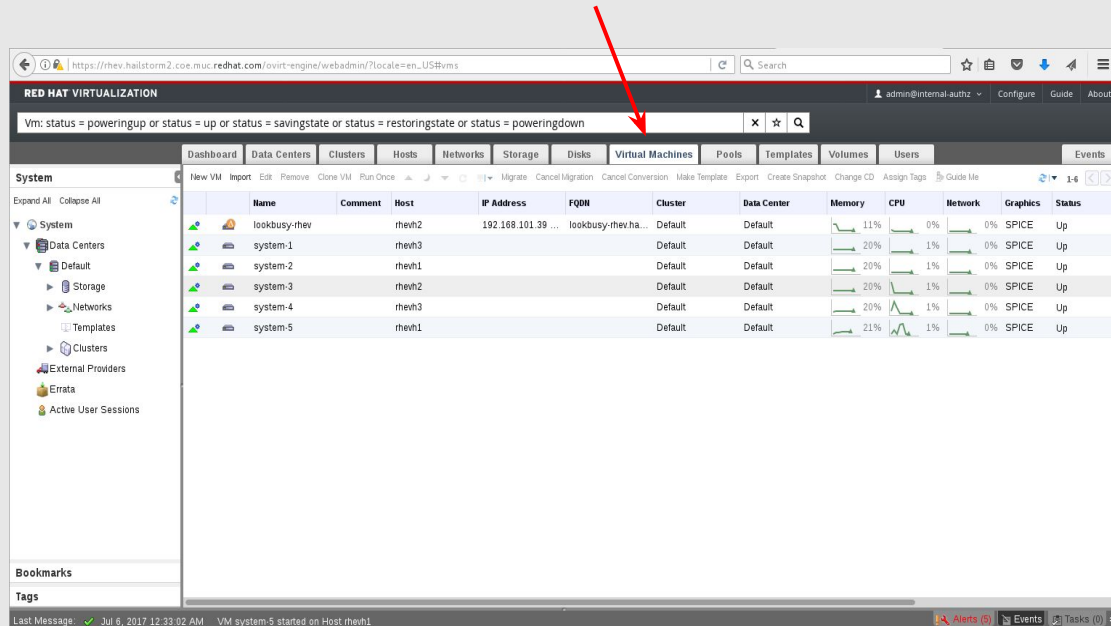
[WARNING]: Found both group and host with same name: storage-console
```

Create Virtual Machines within the RHV environment

Create a number of new RHEL virtual machines on the RHEV hypervisors,

Show:

1. RHEV VM mgmt guide [1]
2. RHEV Admin Guide [2]
3. Ansible Playbooks



The screenshot displays the Red Hat Virtualization (RHV) web console interface. The browser address bar shows the URL: `https://rhev.halstorm2.coe.muc.redhat.com/ovirt-engine/webadmin/?locale=en_US#vms`. The page title is "RED HAT VIRTUALIZATION". The navigation menu includes: Dashboard, Data Centers, Clusters, Hosts, Networks, Storage, Disks, Virtual Machines, Pools, Templates, Volumes, Users, and Events. The "Virtual Machines" tab is selected. The main content area shows a table of virtual machines with columns: Name, Comment, Host, IP Address, FQDN, Cluster, Data Center, Memory, CPU, Network, Graphics, and Status. The table lists five VMs: lookbusy-rhev, system-1, system-2, system-3, system-4, and system-5. A red arrow points to the search bar at the top of the page, which contains the text: "Vm: status = poweringup or status = up or status = savingstate or status = restoringstate or status = poweringdown".

Name	Comment	Host	IP Address	FQDN	Cluster	Data Center	Memory	CPU	Network	Graphics	Status
lookbusy-rhev		rhev2	192.168.101.39...	lookbusy-rhev.ha...	Default	Default	11%	0%	0%	0%	SPICE Up
system-1		rhev3			Default	Default	20%	1%	0%	0%	SPICE Up
system-2		rhev1			Default	Default	20%	1%	0%	0%	SPICE Up
system-3		rhev2			Default	Default	20%	1%	0%	0%	SPICE Up
system-4		rhev3			Default	Default	20%	1%	0%	0%	SPICE Up
system-5		rhev1			Default	Default	21%	1%	0%	0%	SPICE Up

Links:

- [1] - https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html-single/virtual_machine_management_guide/
- [2] - https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.1/html-single/administration_guide/

Create Virtual Machines within the RHV environment

Ansible Playbook and sample output

```
mreinke@mreinke-t540:VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
File Edit View Search Terminal Help
[mreinke@mreinke-t540 ansible]$ cat 04-provision-rhev-vm
#!/bin/bash

# Setup Datacenter and Clusterconfiguration
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.uc.redhat.com.yml create-04-democontent.yml --tags layer3,lookbusy,lookbusy-rhev

# Setup Datacenter and Clusterconfiguration
ansible-playbook -i hosts -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.uc.redhat.com.yml rhv-vm-deploy.yml

[mreinke@mreinke-t540 ansible]$ cat create-04-democontent.yml
- hosts: rhvm
  remote_user: root
  gather_facts: false
  roles:
    - { role: layer2_rhevml_demospermissions, tags: [ 'layer2', 'rhev', 'rhevml', 'rhevml_dap' ], mode: create }

- hosts: lookbusy-rhev
  remote_user: root
  gather_facts: false
  roles:
    - { role: layer3_vm_on_rhev, tags: [ 'layer3', 'lookbusy', 'lookbusy-rhev', 'vm' ], mode: create }
    - { role: layer2_rhel_reconfigure_dns, tags: [ 'layer3', 'lookbusy', 'lookbusy-rhev', 'dns' ], mode: create, names
server: '{{ hostvars[ipa].vm_nics[0].ip }}' }
    - { role: layer2_rhel, tags: [ 'layer3', 'lookbusy', 'lookbusy-rhev', 'rhel' ], mode: create }
    - { role: layer3_lookbusy, tags: [ 'layer3', 'lookbusy', 'lookbusy-rhev' ], mode: create }

- hosts: lookbusy-osp
  remote_user: root
  gather_facts: false
  roles:
    - { role: layer3_vm_on_ubuntu, tags: [ 'layer3', 'lookbusy', 'lookbusy-osp', 'vm' ], mode: create }
    - { role: layer2_rhel, tags: [ 'layer3', 'lookbusy', 'lookbusy-osp', 'rhel' ], mode: create }
    - { role: layer3_lookbusy, tags: [ 'layer3', 'lookbusy', 'lookbusy-osp' ], mode: create }

- hosts: rhosp-director
  remote_user: root
  gather_facts: false
  roles:
    - { role: layer2_rhosp_overcloud_demo_content, tags: [ 'layer2', 'rhosp', 'overcloud', 'democontent' ], mode: create }

- hosts: cloudforms
  remote_user: root
  gather_facts: false
  roles:
    - { role: layer3_cloudforms_content, tags: [ 'layer3', 'cf', 'cfme-content' ], mode: create, content_location: "*/CloudForms-Internals*"

- hosts: ose3-installer
  remote_user: root
  gather_facts: false
  roles:
    - { role: layerX_openshift_devops_tools, tags: [layer2, 'ose3', 'ose3-devops'], mode: create }
    - { role: layerX_openshift_demo_monster, tags: [layer2, 'ose3', 'ose3-demo', 'ose3-demo-ticketmonster'], mode: create }
    - { role: layerX_openshift_demo_wordpress, tags: [layer2, 'ose3', 'ose3-demo', 'ose3-demo-wordpress'], mode: create }

- { role: layerX_openshift_demo_amq, tags: [layer2, 'ose3', 'ose3-demo-amq'], mode: create }
- { role: layerX_openshift_cfme_mmgr, tags: [layer2, 'ose3', 'ose3-cfme-mmgr'], mode: create }
```

```
mreinke@mreinke-t540:VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible
File Edit View Search Terminal Help
[mreinke@mreinke-t540 ansible]$ ./04-provision-rhev-vm
[WARNING]: Found both group and host with same name: storage-console

[WARNING]: Found both group and host with same name: lookbusy-rhev
[WARNING]: Found both group and host with same name: cloudforms
[WARNING]: Found both group and host with same name: ipa
[WARNING]: Found both group and host with same name: test-rhel6
[WARNING]: Found both group and host with same name: dev-client
[WARNING]: Found both group and host with same name: infrastructure
[WARNING]: Found both group and host with same name: rhvm
[WARNING]: Found both group and host with same name: satellite
[WARNING]: Found both group and host with same name: ose3-lb
[WARNING]: Found both group and host with same name: lookbusy-osp
[WARNING]: Found both group and host with same name: layer1
[WARNING]: Found both group and host with same name: efk
[WARNING]: Found both group and host with same name: tower
[WARNING]: Found both group and host with same name: rhosp-director
[WARNING]: Found both group and host with same name: test-rhel7

PLAY [rhvm] *****
PLAY [lookbusy-rhev] *****
TASK [layer3_vm_on_rhev : include] *****
included: /home/mreinke/VirtualMachines/git/hail-storm-temp/hailstorm-mreinke/ansible/roles/Layer3_vm_on_rhev/tasks/instantiate_vm_rhv4.yml for lookbusy-rhev
TASK [layer3_vm_on_rhev : find existing VM] *****
ok: [lookbusy-rhev -> 192.168.103.12]
TASK [layer3_vm_on_rhev : debug] *****
ok: [lookbusy-rhev => {
  "rhev_image": "RHEL7"
}]
TASK [layer3_vm_on_rhev : create VM from template] *****
skipping: [lookbusy-rhev]
TASK [layer3_vm_on_rhev : wait for VM to unlock] *****
skipping: [lookbusy-rhev]
TASK [layer3_vm_on_rhev : get nic info] *****
skipping: [lookbusy-rhev]
TASK [layer3_vm_on_rhev : debug] *****
skipping: [lookbusy-rhev]
TASK [layer3_vm_on_rhev : reconfigure improperly configured nics] *****
```

Simple adhoc management tasks

```
$: cat ovirt-credentials.yml
ovirt_auth:
  url: "https://{{ rhv_dns_domain }}/ovirt-engine/api"
  username: "admin@internal"
  password: "{{ root_password }}"
  ca_file: "/etc/pki/ovirt-engine/ca.pem"

$: alias ansible-h2
alias ansible-h2='ansible -e @config/hailstorm_config.yml -e @config/infrastructure_config.yml -e @config/storm2.coe.muc.redhat.com.yml'

# shutdown VM system-1
$: ansible-h2 rhevm -e @ovirt-credentials.yml -m ovirt_vms --args='auth={{ ovirt_auth }} name=system-1 state=stopped'

# remove VM system-1
$: ansible-h2 rhevm -e @ovirt-credentials.yml -m ovirt_vms --args='auth={{ ovirt_auth }} name=system-1 state=absent'

# migrate VM system-2 to Hypervisor rhevh1
$: ansible-h2 rhevm -e @ovirt-credentials.yml -m ovirt_vms --args='auth={{ ovirt_auth }} name=system-2 host=rhevh1'

# put HV rhevh3 into maintenance
$: ansible-h2 rhevm -e @ovirt-credentials.yml -m ovirt_hosts --args='auth={{ ovirt_auth }} name=rhevh3 state=maintenance'

# create VM with cloud-init
$: cat cloudinit.yml
cloudinit:
  user_name: root
  root_password: "{{ root_password }}"

$: ansible-h2 rhevm -e @cloudinit.yml -e @ovirt-credentials.yml -m ovirt_vms --args='auth={{ ovirt_auth }} name=system-6 cluster=Default
template=RHEL7 cloud_init={{ cloudinit }}'
```

Further story

- Network automation solution
 - see inside and at [1]
- Openstack & wordpress Application sample
 - see link from Keith Tenzer blog [2]
- Red Hat Cloud Solution
 - Container / Openshift Installer
 - Hyper Converged Infrastructure (HCI) Installer
- Training
 - Free starter training (see next slides)



Links:

[1] - <https://www.ansible.com/networking>

[2] - <https://keithtenzer.com/2016/05/09/openstack-heat-and-ansible-automation-born-in-the-cloud/>

WHERE DO I BEGIN?

Learn Ansible

- Join existing Ansible network automation communities
- Take Ansible training courses from Red Hat or elsewhere

Start small!

- Create Playbooks that read or check only
- Create simple jobs that eliminate the most annoying tasks
- Leverage existing knowledge internally

Develop success criteria

- Create specific goals that require planning, tailored to your organization
- Create phases to ensure people and processes aren't alienated

Startup Training - Ansible Essentials (Check for free)

goo.gl/qBtFxQ (shortened URL)

redhat.com/rhtapps/promo-do007 (direct register link to start)

Download Ansible v2.3:

releases.ansible.com/ansible/

Evaluate Ansible Tower:

ansible.com/tower-trial/

Join the Community

Users list: [ansible-project](#)

Development list: [ansible-devel](#)

Announcement

list: [ansible-announce](#) (*read only*)

irc.freenode.net: #ansible



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos