



# Introduction to Ansible Engine and Ansible Tower

Markus Koch

Partner Enablement Manager SAP

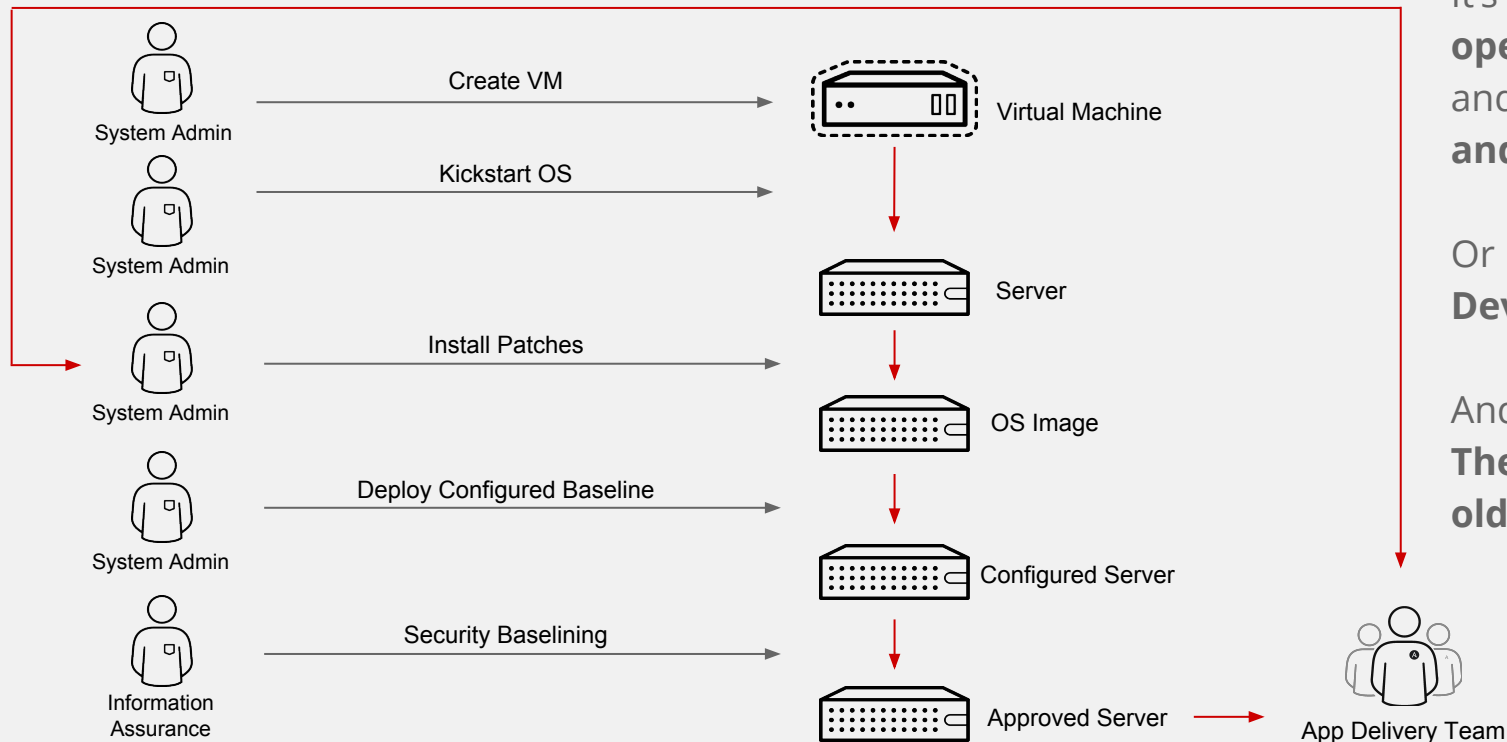




# ~~INSTALL; CONFIGURE - REPEAT~~ AUTOMATE



# THE GOOD OLD DAYS...



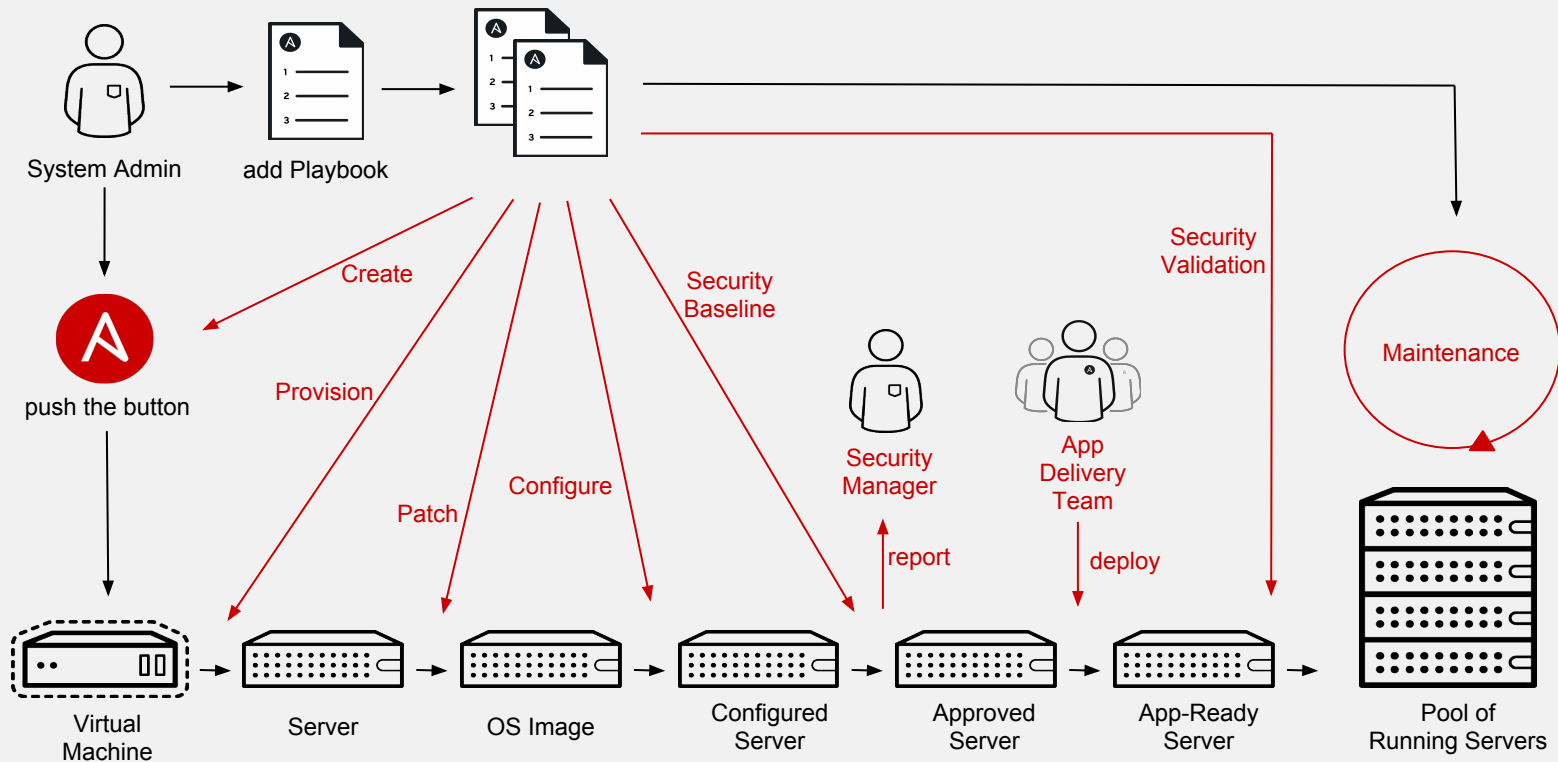
It's hard to **run IT operations this way** and **deliver flexible and agile IT services**.

Or being the **Ops** in **DevOps**.

And yes, you can work **The Cloud** in the **same old way...**



# NOTHING ROUTINE SHOULD BE DONE MANUALLY





# WHAT IS ANSIBLE AUTOMATION?

The Ansible project is an open source community sponsored by Red Hat. It's also a **simple automation language** that perfectly describes IT application environments in **Ansible Playbooks**.

**Ansible Engine** is a **supported product** built from the Ansible community project.

```
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
    http_port: 80

  tasks:
    - name: httpd package is present
      yum:
        name: httpd
        state: latest

    - name: latest index.html file is present
      copy:
        src: files/index.html
        dest: /var/www/html/

    - name: httpd is started
      service:
        name: httpd
        state: started
```



v1 - Set config file to use on boot

1. Write multiple configuration files
  - For each environment/region
2. Inspect metadata on boot and use the matching config file



v1 - Set config file to use on boot

1. Write multiple configuration files
  - For each environment/region
2. Inspect metadata on boot and use the matching config file

**24,000+**

Stars on GitHub

**950+**

Ansible modules

**400,000+**

Downloads a month





## SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Usable by every team

**Get productive quickly**



## POWERFUL

App deployment

Configuration management

Workflow orchestration

Network automation

**Orchestrate the app lifecycle**



## AGENTLESS

Agentless architecture

Uses OpenSSH & WinRM

No agents to exploit or update

Get started immediately

**More efficient & more secure**



# THE ANSIBLE WAY

## CROSS PLATFORM

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

## HUMAN READABLE

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 PLAYS WELL WITH OTHERS

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



# WHAT CAN I DO WITH ANSIBLE?

Automate the deployment and management of your entire IT footprint.

## Do this...

Configuration  
Management

Orchestration

Application  
Deployment

Provisioning

Continuous  
Delivery

Security and  
Compliance

## On these...

Firewalls

Load Balancers

Applications

Containers

Clouds

Servers

Infrastructure

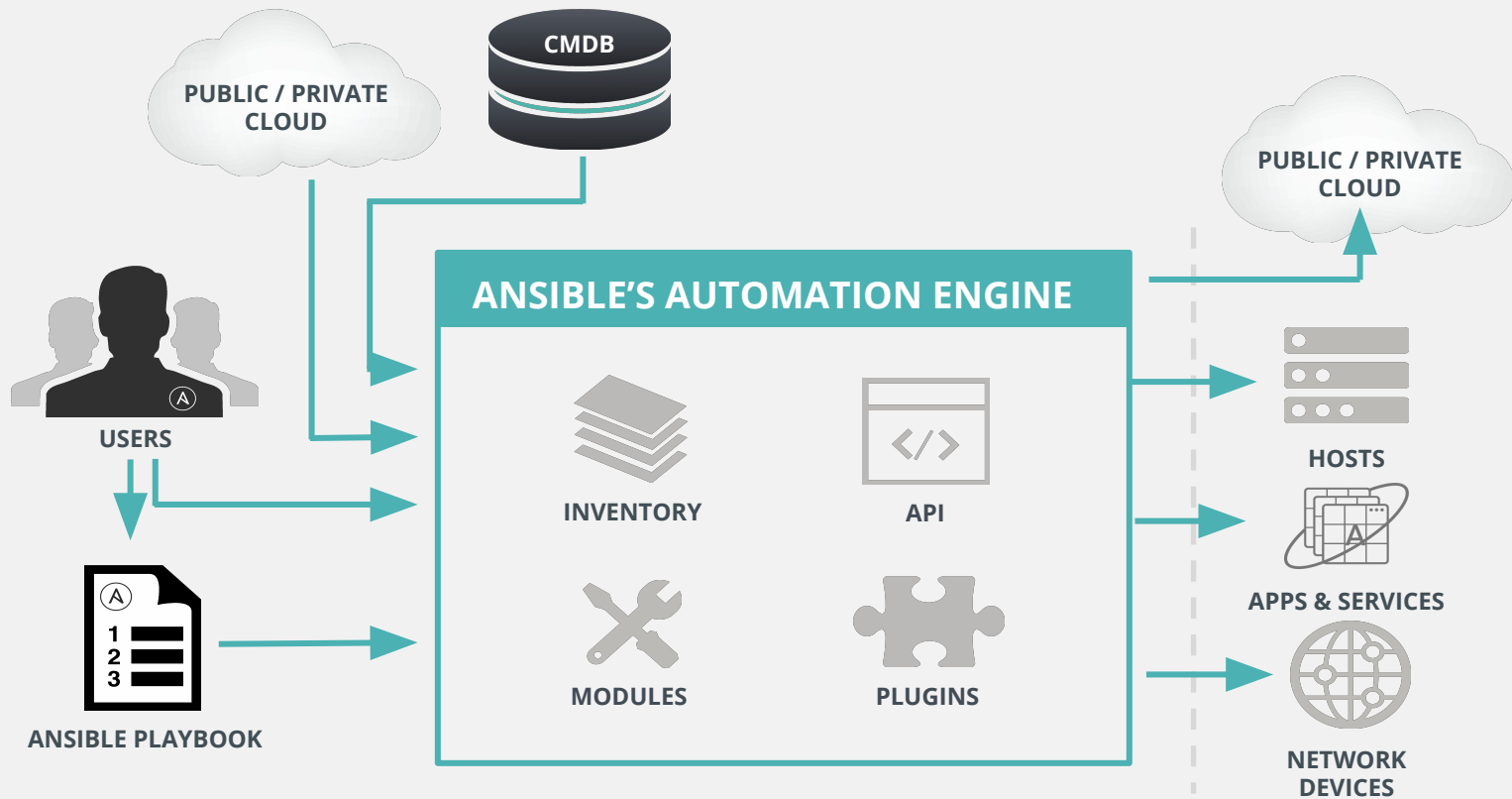
Storage

Network Devices

**And more...**



# HOW ANSIBLE WORKS





## PLAYBOOK EXAMPLE: INSTALL & CONFIGURE APACHE

```
---
- name: install and start apache
  hosts: all
  vars:
    http_port: 80
    max_clients: 200
  become_user: root
  tasks:
    - name: install httpd
      yum: pkg=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
```



# PLAYBOOK EXAMPLE: AWS CLOUD DEPLOYMENT

```
- hosts: localhost
  connection: local
  gather_facts: False
  tasks:
    - name: Provision a set of instances
      ec2:
        key_name: my_key
        group: test
        instance_type: t2.micro
        image: "{{ ami_id }}"
        wait: true
        exact_count: 5
        count_tag:
          Name: Demo
        instance_tags:
          Name: Demo
      register: ec2

    - name: Add all instance public IPs to dynamic host group
      add_host: hostname={{ item.public_ip }} groups=ec2hosts
      with_items: "{{ ec2.instances }}"
```



## PLAYBOOK EXAMPLE: AWS CLOUD DEPLOYMENT

```
- hosts: ec2hosts
  name: configuration play
  user: ec2-user
  gather_facts: true
  tasks:
    - name: Check NTP service
      service:
        name: ntpd
        state: started
```



```
yum install @base xfsprogs libaio net-tools bind-utils gtk2 libicu xulrunner tcsh  
sudo libssh2 expect cairo graphviz iptraf-ng krb5-workstation krb5-libs libpng12  
ntp ntpdate nfs-utils lm_sensors rsyslog openssl098e openssl  
PackageKit-gtk3-module libcanberra-gtk2 libtool-ltdl xorg-x11-xauth numactl
```



```
- name: install required packages  
  yum: state=latest name={{ item }}  
  with_items:  
    - chrony  
    - xfsprogs  
    - libaio  
    - net-tools  
    - bind-utils  
    - ...  
    - numactl  
    - tuned-profiles-sap-hana
```



```
systemctl stop numad  
systemctl disable numad  
systemctl status numad
```



```
- name: disable numad  
  service: name=numad state=stopped enabled=no
```



```
setenforce 0  
sed -i 's/SELINUX=enforcing/SELINUX=permissive/' /etc/selinux/config  
sestatus
```



```
- name: disable selinux  
  selinux: state=disabled
```



```
echo "@sapsys soft nproc unlimited" > /etc/security/limits.d/99-sapsys.conf
echo "@sapsys hard nproc unlimited" > /etc/security/limits.d/99-sapsys.conf
```



```
- name: set number of process to unlimited for sapsys group
  pam_limits:
    domain: "@sapsys"
    limit_item: nproc
    limit_type: "{{ item }}"
    value: unlimited
  with_items:
    - soft
    - hard
```



```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
sed -i '/^GRUB_CMDLINE_LINUX*./ s/\"$/ transparent_hugepage=never\"/'
/etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg
```



```
- name: disable transparent hugepages in grub config
  lineinfile:
    dest: /etc/default/grub
    line: GRUB_CMDLINE_LINUX_DEFAULT="transparent_hugepage=never"
    notify: regenerate grub2 conf

...

handlers:
  - name: regenerate grub2 conf
    shell: grub2-mkconfig -o /boot/grub2/grub.cfg
```



```
echo "vm.swappiness=60" >> /etc/sysctl.d/90-sap_hana.conf
echo "kernel.msgmni=32768" >> /etc/sysctl.d/90-sap_hana.conf
...
sysctl -p /etc/sysctl.d/90-sap_hana.conf
```



```
- name: setting kernel tunables
  sysctl: name={{ item.name }} value={{ item.value }} state=present
  sysctl_set=yes reload=yes
  with_items:
    - { name: kernel.msgmni, value: 32768 }
    ...
    - { name: vm.swappiness, value: 60 }
    ...
```



```
lvcreate -L 1G -n lv_usr_sap /dev/vg00
lvcreate -l +100%FREE -n lv_hana /dev/vg01
mkfs.xfs /dev/vg01/lv_hana
mkfs.xfs /dev/vg00/lv_usr_sap
mkdir /usr/sap
mount /dev/vg00/lv_usr_sap /usr/sap/
mkdir /hana
mount /dev/vg01/lv_hana /hana
```



```
- name: create logical volumes
  lvol: state=present vg=vg00 \
        lv=lv_hana size="100%FREE"

- name: create filesystems
  filesystem:
    dev: /dev/vg01/lv_hana
    fstype: xfs
    force: no
```

```
- name: mount and make fstab entries
  mount:
    name: "/hana"
    fstype: xfs
    opts: defaults
    passno: 4
    src: "/dev/vg01/lv_hana"
    state: mounted
```



# ANSIBLE SHIPS WITH OVER 1250 MODULES (THIS IS WHERE THE MAGIC HAPPENS)

ANSIBLE

## CLOUD

AWS  
Azure  
CenturyLink  
CloudScale  
Digital Ocean  
Docker  
Google  
Linode  
OpenStack  
Rackspace  
**And more...**

## VIRT AND CONTAINER

Docker  
VMware  
RHEV  
OpenStack  
OpenShift  
Atomic  
CloudStack  
**And more...**

## WINDOWS

ACLs  
Files  
Commands  
Packages  
IIS  
Regedits  
Shell  
Shares  
Services  
Configs  
Users  
Domains  
**And more...**

## NETWORK

Arista  
A10  
Cumulus  
Big Switch  
Cisco  
Cumulus  
Dell  
F5  
Juniper  
Palo Alto  
OpenSwitch  
**And more...**

## NOTIFY


HipChat  
IRC  
Jabber  
Email  
RocketChat  
Sendgrid  
Slack  
Twilio  
**And more...**



# ANSIBLE GALAXY CONTAINS MANY READY TO USE ROLES

<http://galaxy.ansible.com>

ANSIBLE

 GALAXY

ABOUTEXPLORESEARCHBROWSE AUTHORSMY CONTENTRHMK

EXPLORE

Most Starred

Name	Stars
<a href="#">carlosbuenosvinos.ansistrano-deploy</a>	1475
<a href="#">dev-sec.os-hardening</a>	736
<a href="#">elastic.elasticsearch</a>	569
<a href="#">jdauphant.nginx</a>	543
<a href="#">geerlingguy.mysql</a>	488
<a href="#">ANXS.postgresql</a>	476
<a href="#">angstwad.docker_ubuntu</a>	428
<a href="#">geerlingguy.jenkins</a>	403
<a href="#">DavidWittman.redis</a>	403
<a href="#">thefinn93.letsencrypt</a>	373

View More

Most Watched

Name	Watchers
<a href="#">elastic.elasticsearch</a>	206
<a href="#">couchbaselabs.couchbase-server</a>	137
<a href="#">CumulusNetworks.cumulus-linux-ansib...</a>	95
<a href="#">cumulus.CumulusLinux</a>	95
<a href="#">ceph.ceph-defaults</a>	86
<a href="#">ceph.ceph-config</a>	73
<a href="#">dev-sec.os-hardening</a>	73
<a href="#">carlosbuenosvinos.ansistrano-deploy</a>	72
<a href="#">cumulus.activedirectory-auth-client</a>	71
<a href="#">CumulusNetworks.activedirectory-auth...</a>	71

View More

Most Downloaded

Name	Downloads
<a href="#">DavidWittman.redis</a>	262134
<a href="#">Stouts.grafana</a>	230617
<a href="#">nickhammond.logrotate</a>	135844
<a href="#">geerlingguy.composer</a>	130095
<a href="#">geerlingguy.java</a>	125573
<a href="#">geerlingguy.redis</a>	112613
<a href="#">geerlingguy.nginx</a>	111275
<a href="#">geerlingguy.mysql</a>	99108
<a href="#">geerlingguy.php</a>	95503
<a href="#">Stouts.mongodb</a>	87277

View More

Top Tags

Tag	# Repos
<a href="#">system</a>	4824
<a href="#">development</a>	2431
<a href="#">web</a>	2049
<a href="#">monitoring</a>	1057

Top Contributors

User	# Repos
<a href="#">andrewrothstein</a>	243
<a href="#">mrlesmithjr</a>	187
<a href="#">softasap</a>	156
<a href="#">debops</a>	134

Newest

Name	Added On
<a href="#">tenequm.security</a>	16 Apr
<a href="#">mimacom.crowd</a>	16 Apr
<a href="#">jaredhocutt.rocketchat</a>	16 Apr
<a href="#">heriet.lookup_ini_sections</a>	16 Apr

Search

22



## HOW & WHERE TO GET ANSIBLE?

Install Ansible Engine on Red Hat Enterprise Linux 7 with any of the following methods:

If you have a Red Hat **Ansible Engine Subscription**, subscribe the system and enable the Ansible Engine repository **rhel-7-server-ansible-2.4-rpms**.

If you are a RHEL only Customer **enable** the **RHEL Extras repository**. The Extras repo release cycle may not update on the same cycle as the Ansible Engine repo.

**RPMs** of Ansible Engine releases are available from **releases.ansible.com**.

*<https://access.redhat.com/articles/3174981>*





**RED HAT**  
**ANSIBLE**  
Tower

# AUTOMATION FOR TEAMS

Ansible Tower technical introduction and overview





**SIMPLE**



**POWERFUL**



**AGENTLESS**

# What is Missing?





## CENTRAL

Central place for everyone

Overview of present and past

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

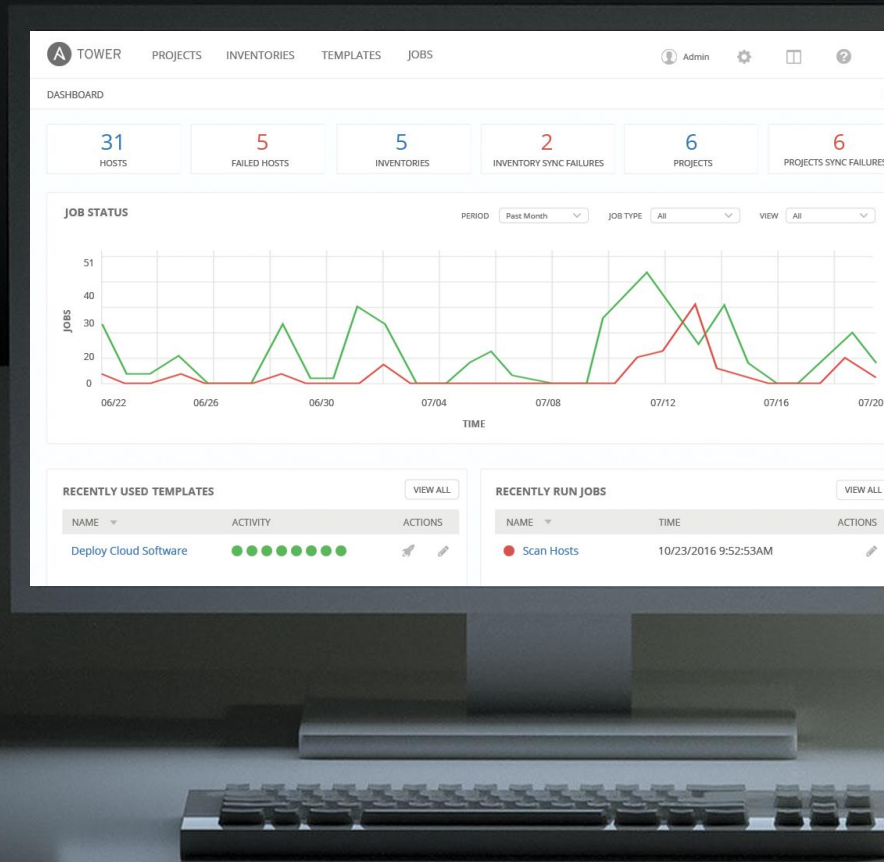
**Separate access and execution**



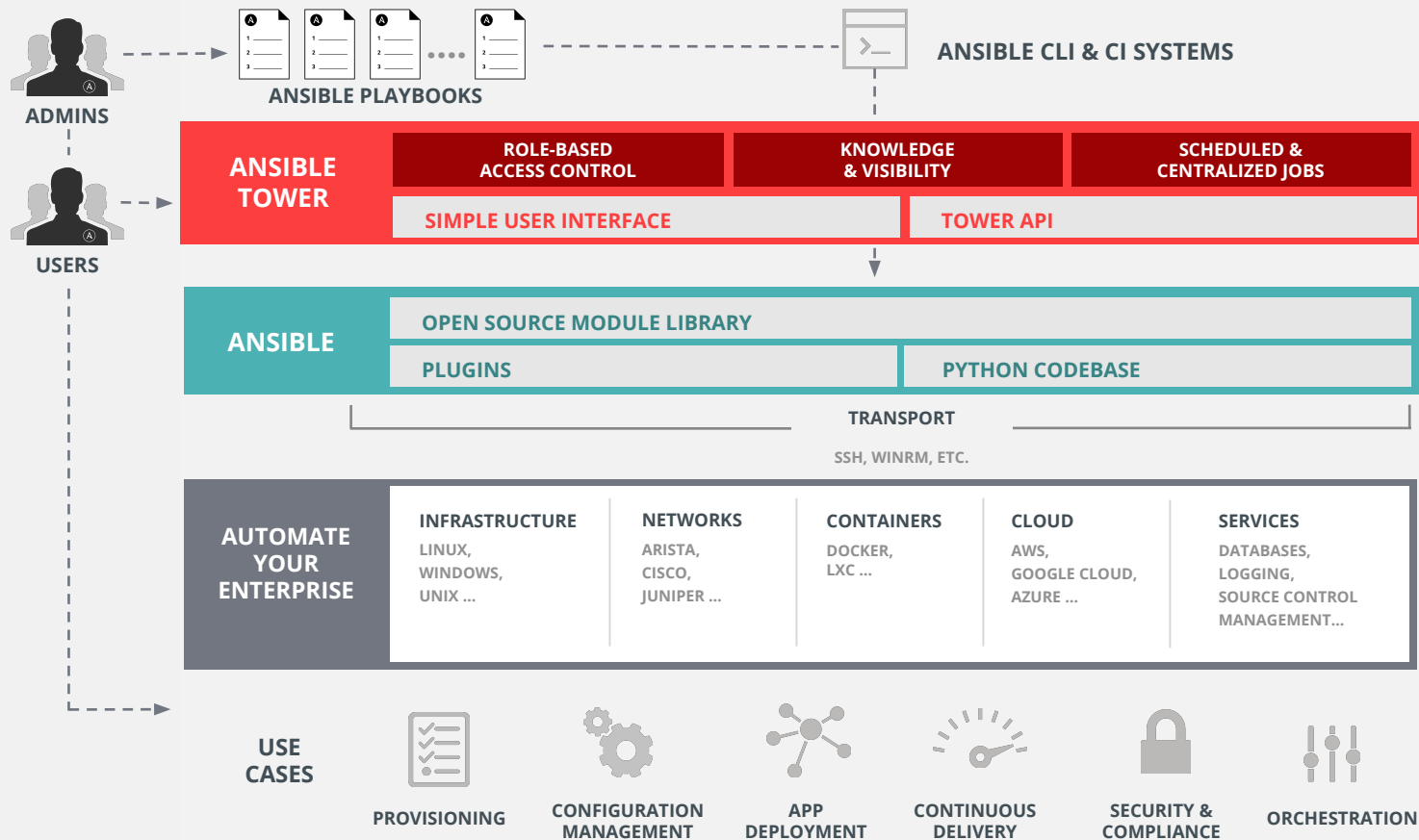
# WHAT IS ANSIBLE TOWER?

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

- **Role-based access control**
- **Deploy** entire applications with **push-button deployment** access
- All automations are **centrally logged**









# ANSIBLE TOWER

## JOB STATUS UPDATE

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

The screenshot displays the Ansible Tower web interface. On the left, the 'DETAILS' sidebar for a job named 'Update License Server' shows a 'Successful' status, completion time of 1:00:12 AM, and various configuration details like 'License' project and 'Store' limit. The main panel, titled 'REMOVE VMWARE HOST', shows a progress bar at 100% and a list of tasks. The tasks include gathering facts, installing packages, updating tools, creating a user, configuring permissions, enabling maintenance, and checking SSH connections, all of which completed successfully. A 'PLAY RECAP' is shown at the bottom of the task list.

Task ID	Task Name	Status	Elapsed Time
1	PLAY [Remove VMware Host]	Successful	00:00:05
2	GATHERING FACTS	Successful	00:00:01
16	TASK: [ansiblelicense   install required packages via yum]	Successful	00:00:01
17	ok: [74.207.226.226]	Successful	
18	ok: [74.207.226.226]	Successful	
20	TASK: [ansiblelicense   update setuptools]	Successful	00:00:01
28	TASK: [ansiblelicense   update pip]	Successful	00:00:01
36	TASK: [ansiblelicense   create unprivileged user for ansiblelicense]	Successful	00:00:01
37	skipping: [74.207.226.226]	Successful	
38	skipping: [74.207.226.226]	Successful	
40	TASK: [ansiblelicense   configure ansiblelicense directory permissions]	Successful	00:00:01
41	changed: [74.207.226.226]	Successful	
42	changed: [74.207.226.226]	Successful	
43	TASK: [ansiblelicense   enable maintenance page]	Successful	00:00:01
44	TASK: [ansiblelicense   check ssh connection to github]	Successful	00:00:01
66	ok: [74.207.226.226]	Successful	
67	PLAY RECAP	Successful	



# ANSIBLE TOWER

The screenshot shows the Ansible Tower web interface. At the top is a navigation bar with links for TOWER, PROJECTS, INVENTORIES, TEMPLATES, and JOBS. On the right of the navigation bar, there is a user profile for 'admin' and several icons for settings, lists, documents, and power. Below the navigation bar is a header for the 'ACTIVITY STREAM' section. The main content area is titled 'ACTIVITY STREAM | ALL ACTIVITY' and includes a 'REFRESH' button and a dropdown menu currently set to 'All Activity'. Below this is a search bar with a dropdown for 'INITIATED BY' and a 'SEARCH' input field. The activity stream is presented as a table with four columns: TIME, INITIATED BY, EVENT, and ACTIONS. The table contains three entries, all initiated by 'admin'.

TIME	INITIATED BY	EVENT	ACTIONS
10/3/2016 5:00:52 PM	admin	created schedule <a href="#">Daily remediation</a>	
10/3/2016 4:51:45 PM	admin	deleted schedule Hourly scan	
10/3/2016 4:51:13 PM	admin	created schedule Hourly scan	

## ACTIVITY STREAM

**Securely** stores every Job that runs, and enables you to view them later, or export details through Tower's API.



# ANSIBLE TOWER

NAME	CAPACITY	RUNNING JOBS
development	82.46%	1
dmz	0%	0
operations	100%	0
test	94.74%	1
tower	90.98%	0

## SCALE-OUT CLUSTERING

Connect multiple Tower nodes into a Tower cluster to add redundancy and capacity to your automation platform.

Add reserved capacity and capacity by organization, and deploy remote execution nodes for additional local capacity.



# ANSIBLE TOWER

**TOWER** PROJECTS INVENTORIES TEMPLATES JOBS admin

JOB TEMPLATES SCHEDULES / JOB TEMPLATE SCHEDULES.EDIT

**DAILY REMEDIATION**

\* NAME: Daily remediation

\* START DATE (MM/DD/YYYY): 10/03/2016

\* START TIME (HH24:MM:SS): 01 : 23 : 45

\* LOCAL TIME ZONE: America/New\_York

\* REPEAT FREQUENCY: Day

**FREQUENCY DETAILS**

\* EVERY: 1 DAYS

\* END: Never

**SCHEDULE DESCRIPTION**

every day

OCCURRENCES (Limited to first 10) DATE FORMAT ☒ LOCAL TIME ☐ UTC

10/03/2016 01:23:45 EDT

## SCHEDULE JOBS

Enables you to any Job now, later, or forever.



# ANSIBLE TOWER

## MANAGE AND TRACK YOUR INVENTORY

Tower's **inventory syncing** and **provisioning callbacks** allow nodes to request configuration on demand, enabling autoscaling.

Smart Inventories allow you to organize and automate hosts across all your providers based on a powerful host fact query engine.

See alerts from Red Hat Insights directly from Tower, and use Insights-provided Playbook Remediation to fix issues in your infrastructure.

The screenshot displays the Ansible Tower web interface. The top navigation bar includes links for TOWER, PROJECTS, INVENTORIES, TEMPLATES, JOBS, and a user profile (admin). The breadcrumb trail indicates the current location: INVENTORIES / MANAGE CLOUD STAGING SERVERS / EDIT.

The main content area is titled 'CLOUD SERVERS' and contains several configuration sections:

- DETAILS / NOTIFICATIONS:** Two tabs are present, with 'DETAILS' currently selected.
- \* NAME:** A text input field containing 'Cloud servers'.
- DESCRIPTION:** An empty text input field.
- SOURCE:** A dropdown menu set to 'Amazon EC2'.
- CLOUD CREDENTIAL:** A search input field containing 'Amazon keys'.
- REGIONS:** A dropdown menu showing 'US East (Northern Virginia)'.
- INSTANCE FILTERS:** A text input field containing 'tag:Name=\*staging\*'.
- ONLY GROUP BY:** An empty text input field.
- UPDATE OPTIONS:** A list of checkboxes: 'Overwrite' (checked), 'Overwrite Variables' (checked), and 'Update on Launch' (unchecked).
- VARIABLES:** Radio buttons for 'YAML' (selected) and 'JSON'.

At the bottom, there is a table with one row containing the number '1'.



# ANSIBLE TOWER

**LAUNCH JOB | DEPLOY SOFTWARE**

INVENTORY

CREDENTIAL

SURVEY

\* ENTER NUMBER OF SERVICE INSTANCES.

\* PLEASE SELECT THE SERVICE OWNER.

Alice

\* ENTER PASSWORD FOR DEPLOYED CERTIFICATE.

SHOW

INVENTORY

Cloud staging servers

CREDENTIAL

Staging ssh key

CANCEL

LAUNCH

## SELF-SERVICE IT

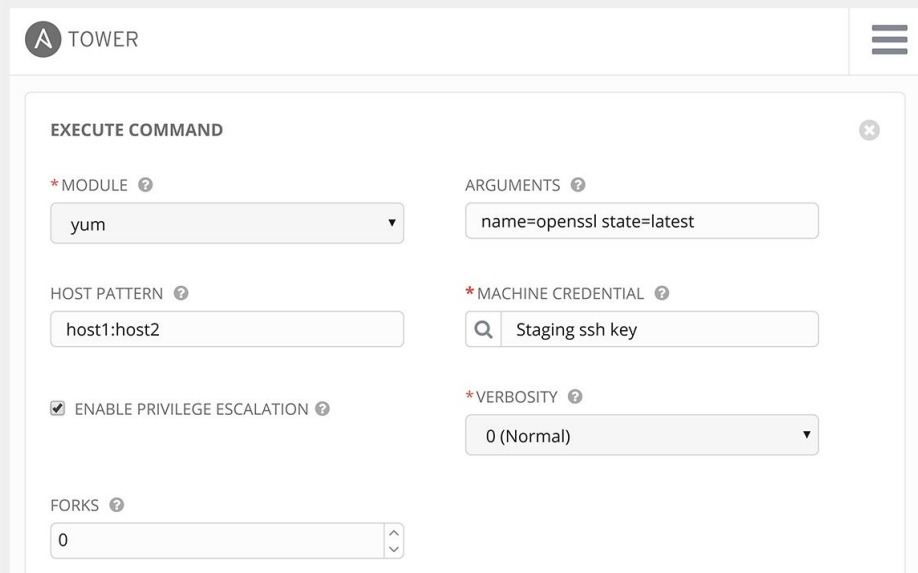
Tower lets you launch Playbooks with just a single click. It can prompt you for variables, let you choose from available secure credentials and monitor the resulting deployments.



# ANSIBLE TOWER

## REMOTE COMMAND EXECUTION

Run simple tasks on any hosts with Tower's **remote command execution**. Add users or groups, reset passwords, restart a malfunctioning service or patch a critical security issue, quickly.



The screenshot shows the 'EXECUTE COMMAND' form in the Ansible Tower web interface. The form is titled 'EXECUTE COMMAND' and includes a close button (X) in the top right corner. It contains several input fields and checkboxes:

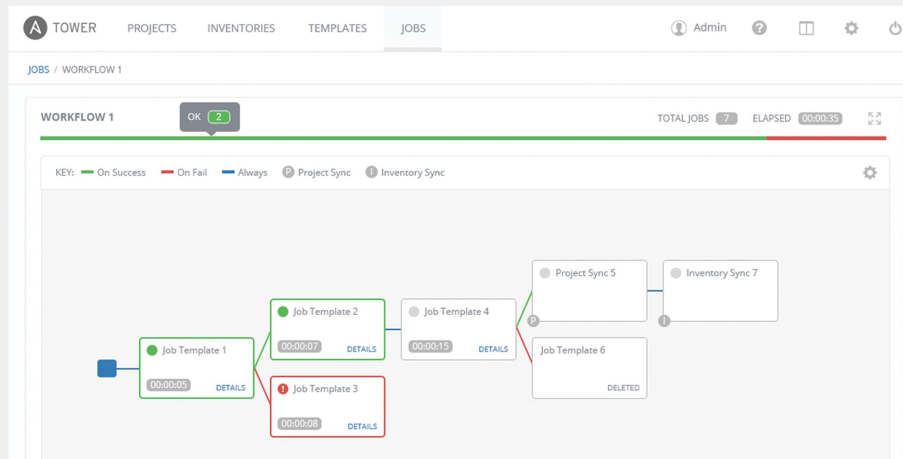
- \*MODULE**: A dropdown menu with 'yum' selected.
- ARGUMENTS**: A text input field containing 'name=openssl state=latest'.
- HOST PATTERN**: A text input field containing 'host1:host2'.
- \*MACHINE CREDENTIAL**: A search input field with a magnifying glass icon and the text 'Staging ssh key'.
- ENABLE PRIVILEGE ESCALATION**: A checked checkbox.
- \*VERBOSITY**: A dropdown menu with '0 (Normal)' selected.
- FORKS**: A text input field with '0' and up/down arrow buttons.



# ANSIBLE TOWER

## MULTI-PLAYBOOK WORKFLOWS

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.






# ANSIBLE TOWER

## INTEGRATED NOTIFICATIONS

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




 **#prodOps Notification**  
Prod Ops Complete!

NOTIFICATION TEMPLATES 1

+ ADD

NAME SEARCH

Q

NAME ^	ACTIONS
<input type="radio"/> Prod Ops Complete	  

ITEMS 1-1 OF 1



# Leverage Ansible from CloudForms

The screenshot displays the Red Hat CloudForms Management Engine interface. The top navigation bar includes a hamburger menu, the title "RED HAT CLOUDFORMS MANAGEMENT ENGINE", and a user profile "Administrator | EVM 41demomasterA". The left sidebar contains navigation links: Cloud Intel, Red Hat Insights, Services, Compute, and Configuration. The main content area is titled "All Configuration Management Providers" and contains a table with the following data:

	Provider Name	URL	Type	Zone	Last Refresh Date	Region Description	Status	Total Configured Systems
<input type="checkbox"/>	Ansible Tower Configuration Manager	https://10.3.48.240/api/v1	Configuration Manager (Ansible Tower)	default	06/22/16 15:10:59 MDT	Region 1	Valid	124
<input type="checkbox"/>	Satellite 6 Configuration Manager	10.3.49.238	Configuration Manager (Red Hat Satellite)	default	06/22/16 15:11:02 MDT	Region 1	Valid	1

Below the table, there are two overlapping windows. The first window, titled "Configuration Script 'JBoss EC2'", shows the "Properties" section with "Name" as "JBoss EC2" and "Region" as "Region 1". The "Variables" section lists "instance\_id" and "ext\_database\_name". The "Surveys" section is empty. The second window, titled "Adding a new Service Catalog Item", shows the "Name / Description" field with "JBoss EC2 Playbook". The "Catalog" dropdown is set to "Hybrid Cloud Automation Items". The "Dialog" dropdown is set to "<No Dialog>". The "Provider" dropdown is set to "Ansible Tower Configuration Manager". The "Ansible Tower Job Template" dropdown is open, showing a list of templates: "AWS ELB", "Add host to HAProxy", "Elastic Deployment", "HA JBoss Deployment", "HAProxy Deployment", "HelloWorld", "HelloWorld EC2", "JBoss Deployment", and "JBoss EC2". The "JBoss EC2" template is selected. The "Provisioning Entry Point" field is set to "AWS ELB". The "Reconfigure Entry Point" field is set to "Elastic Deployment". The "Retirement Entry Point" field is set to "HelloWorld".





<http://people.redhat.com/mkoch/training/>



# 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)