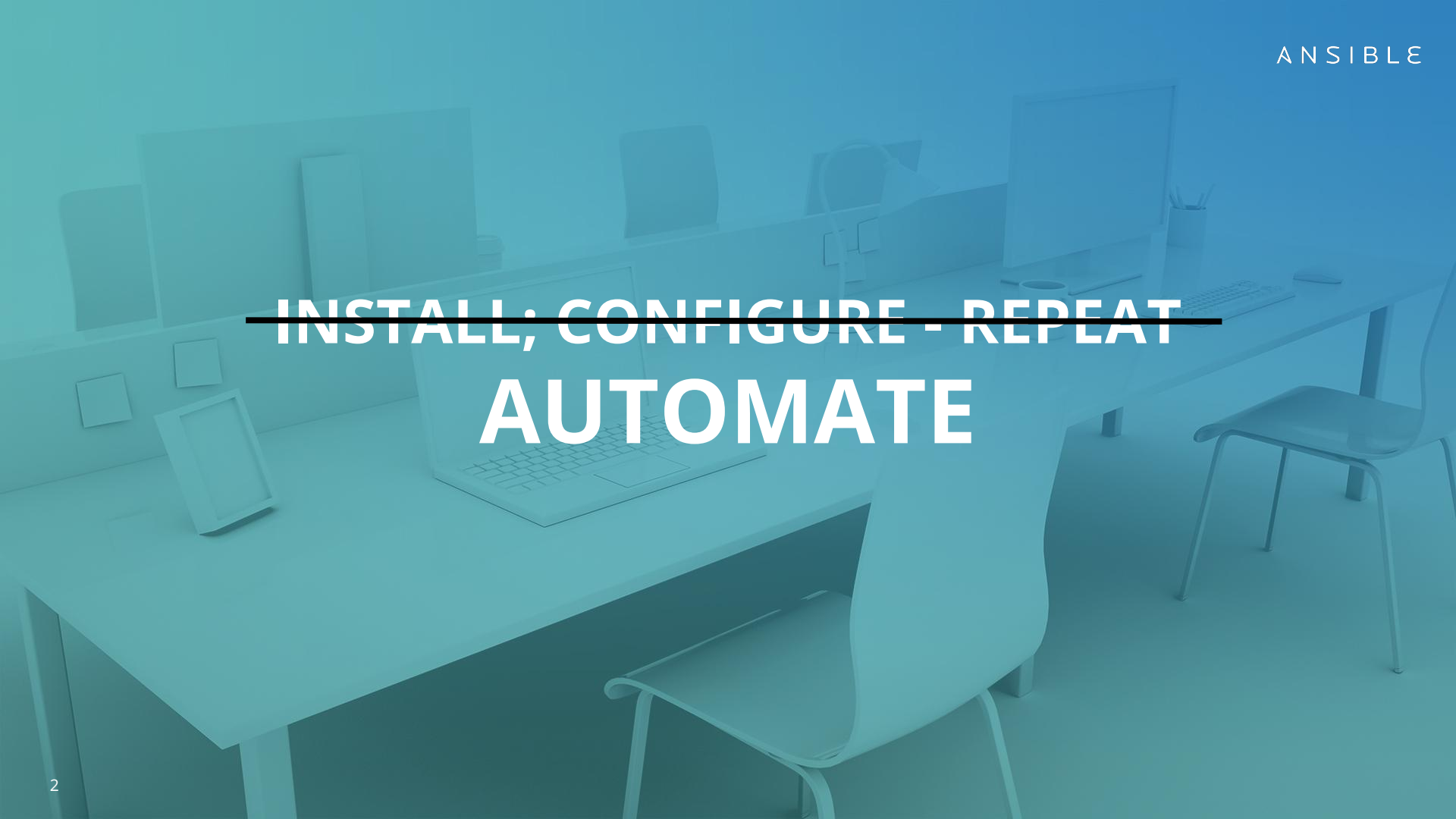
redhat.

Introduction to Ansible Engine and Ansible Tower

Markus Koch

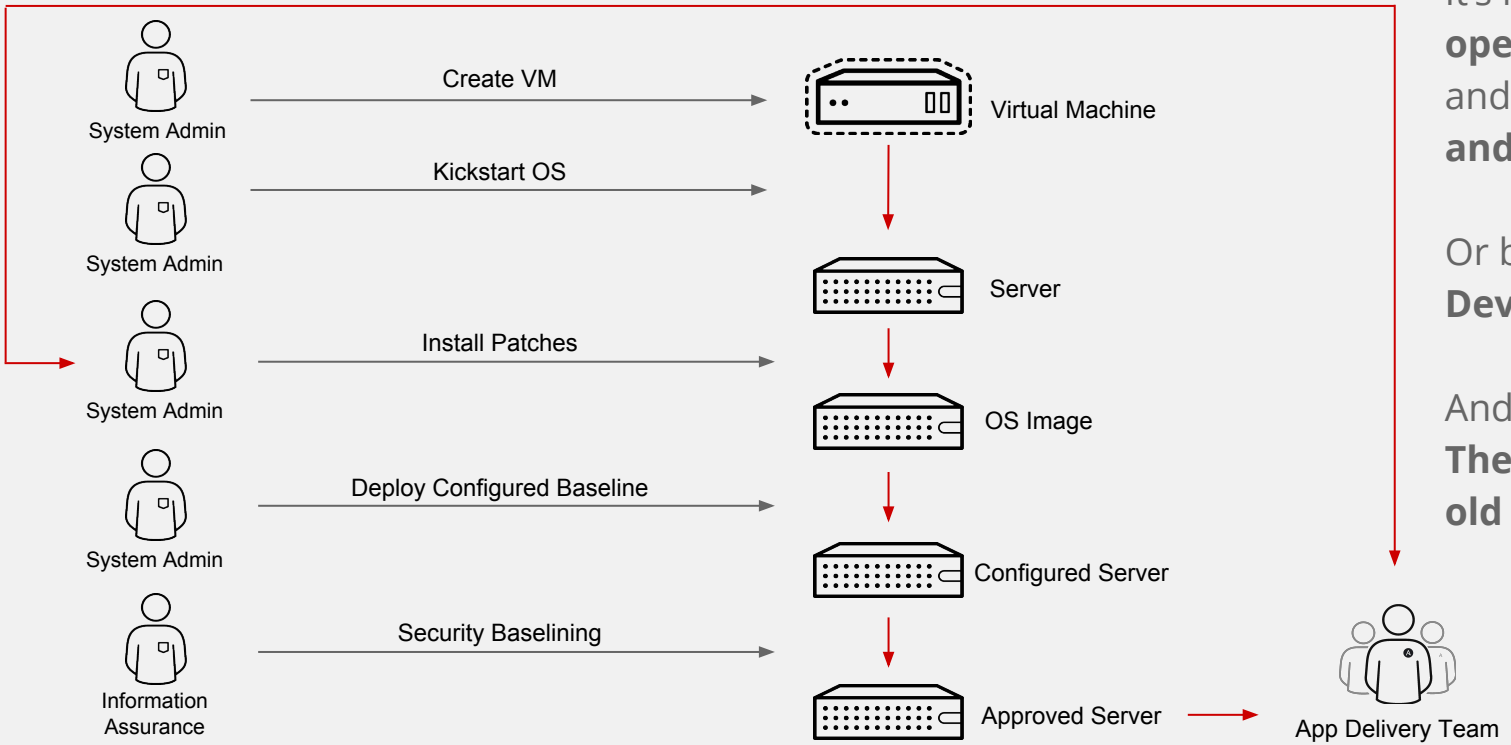
Partner Enablement Manager SAP



~~INSTALL; CONFIGURE - REPEAT~~

AUTOMATE

THE GOOD OLD DAYS…

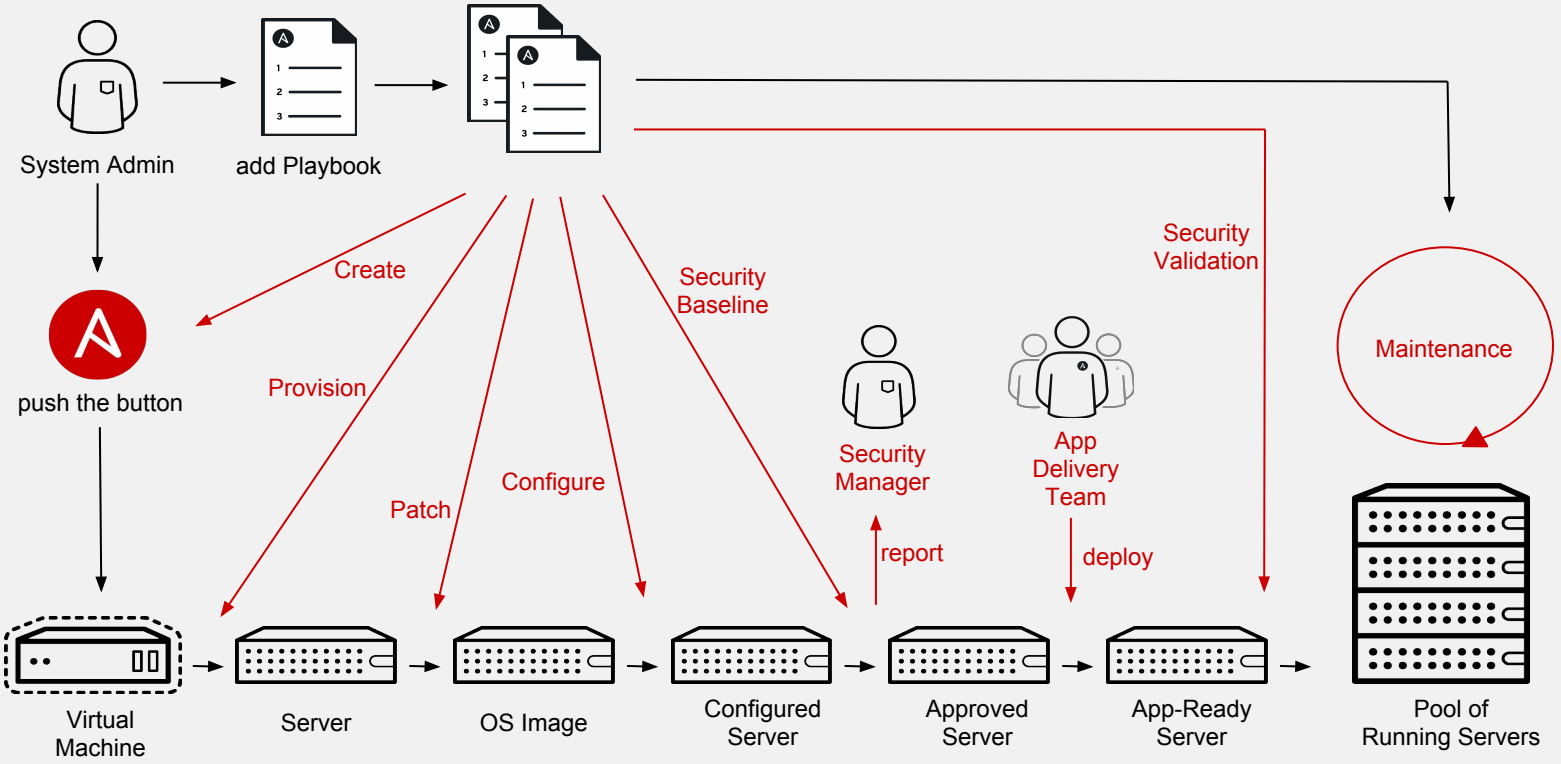


It's hard to **run IT operations this way** and **deliver flexible and agile IT** services.

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
```


ANSIBLE
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

WHY ANSIBLE?

ANSIBLE

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.

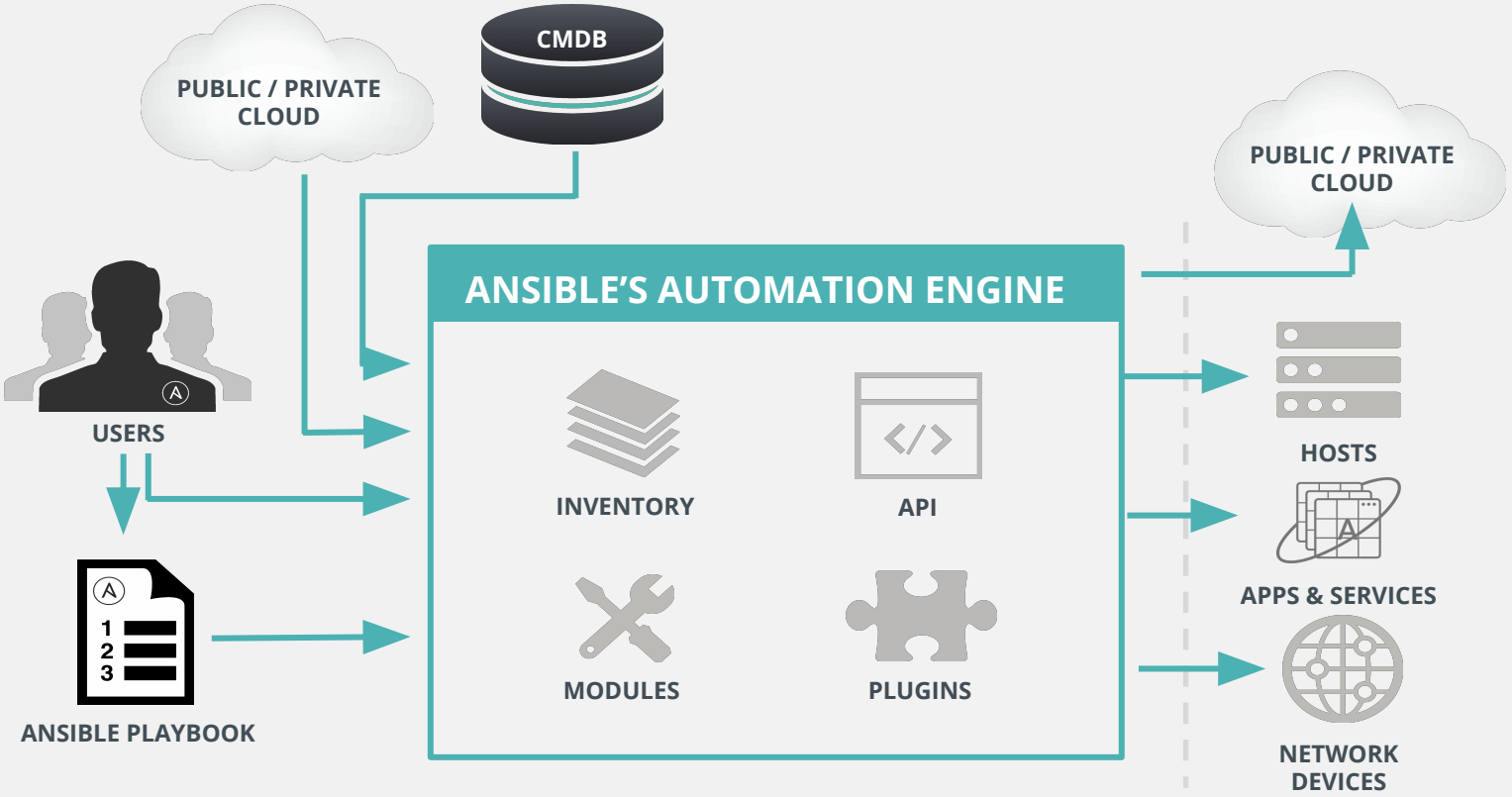
Do this...

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
```

SAP HANA standard installation process

- On Site construction
- Installation & Configuration
- Installation & Configuration
- Validation & Customization
- Maintenance & Updates

Individually for each server and environment!

SAP HANA DEPLOYMENT WITH ANSIBLE

```
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
```

SAP HANA DEPLOYMENT WITH ANSIBLE

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

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

SAP HANA DEPLOYMENT WITH ANSIBLE

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

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

SAP HANA DEPLOYMENT WITH ANSIBLE

```
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
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
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
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
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 }
       ...
```

SAP HANA DEPLOYMENT WITH ANSIBLE

```
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...

SAP HANA optimized installation process

- On Site construction
- Installation & Configuration
- Installation & Configuration
- Validation & Customization
- Maintenance & Updates

Automation for the whole environment

the ansible roles under the hood
Supported deployment scenarios

hana scale-up

- one instance
- multi instance and multi container installation
- hana scale-up system-replication (one and multi-instance)

hana scale-out

- one instance
- multi instance and multi container installation
- hana scale-out system-replication (one and multi-instance)

the ansible roles under the hood

Roles overview

The ansible roles covered in this project

- preconfigure
- deployment
- hsr

the ansible roles under the hood
Roles overview

The ansible roles covered in this project

- preconfigure

  check installation media and version

  runs the checks & configures according to SAP Notes

- deployment
- hsr

the ansible roles under the hood

Roles overview

The ansible roles covered in this project

- preconfigure
- deployment

  kicks off the final deployment according to configuration file

  configuration file centrally stored for easier reproduction

- hsr

the ansible roles under the hood

Roles overview

The ansible roles covered in this project

- base-host-setup
- preconfigure
- deployment
- hsr

configure HANA system replication between formerly deployed systems

the ansible roles under the hood
example playbook

```
- name: Install SAP HANA
  hosts: hana.example.com
  become: yes

  vars:
              # SAP Precoonfigure role
              # SAP-Media Check
              install_nfs: "nfssrv:/export"
              installroot: /install
              hana_installdir: /install/HANA_EXPRESS_20

              hana_pw_hostagent_ssl: "MyS3cret!"
              id_user_sapadm: "30200"
              id_group_shm: "30220"
              id_group_sapsys: "30200"
              pw_user_sapadm_clear: "MyS3cret!"

  roles:
              - { role: mk-ansible-roles.saphana-preconfigure }
              - { role: mk-ansible-roles.saphana-deploy }
```

the ansible roles under the hood

example host configuration file for hana.example.com

```
---
hostname: "{{ ansible_hostname }}"
deployment_instance: true

instances:
  hxe:
    id_user_sidadm: "30210"
    pw_user_sidadm: "Adm12356"
    hana_pw_system_user_clear: "System123"
    hana_components: "client,server"
    hana_system_type: "Master"
    id_group_shm: "30220"
    hana_instance_hostname: "{{ ansible_hostname }}"
    hana_addhosts:
    hana_sid: HXE
    hana_instance_number: 90
    hana_system_usage: custom
...
```

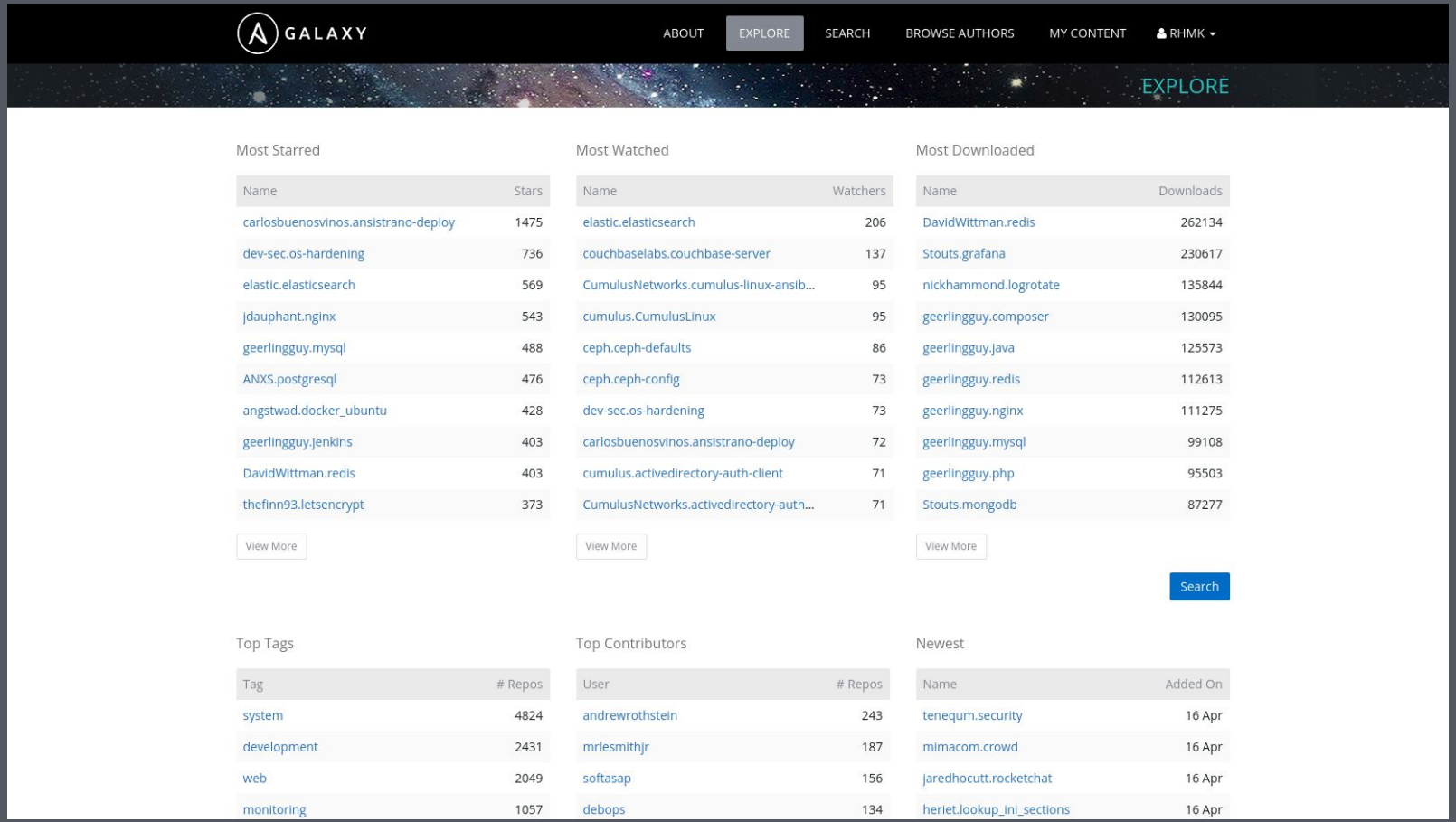
the ansible roles under the hood
example host configuration file for hana.example.com

- some more variables for setting up HSR
- visit http://github.com/mk-ansible-roles to read more
- visit http://people.redhat.com/mkoch/training to learn how to use the roles and create your own playbooks

ANSIBLE GALAXY CONTAINS MANY READY TO USE ROLES

http://galaxy.ansible.com

ANSIBLE



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

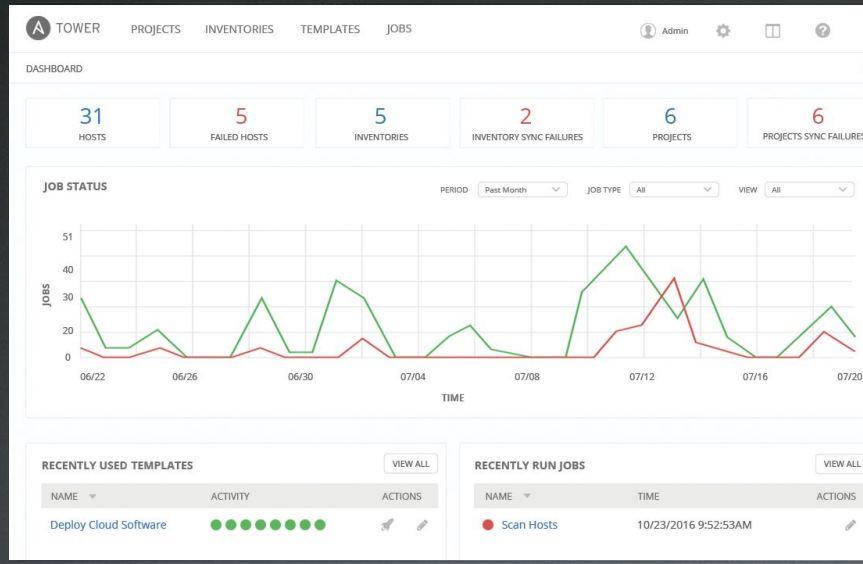
Assign access to unprivileged

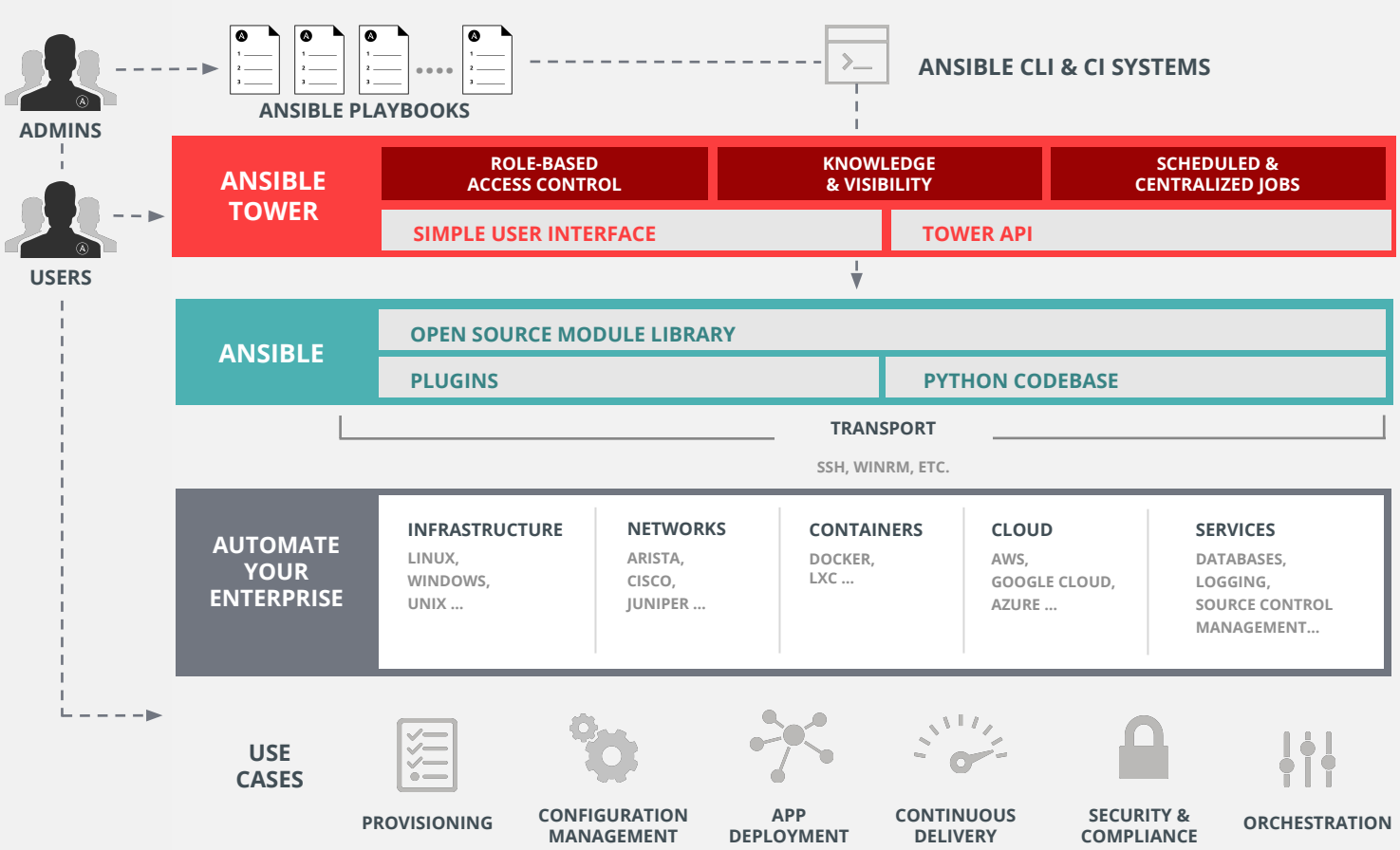
Separate access and execution

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**

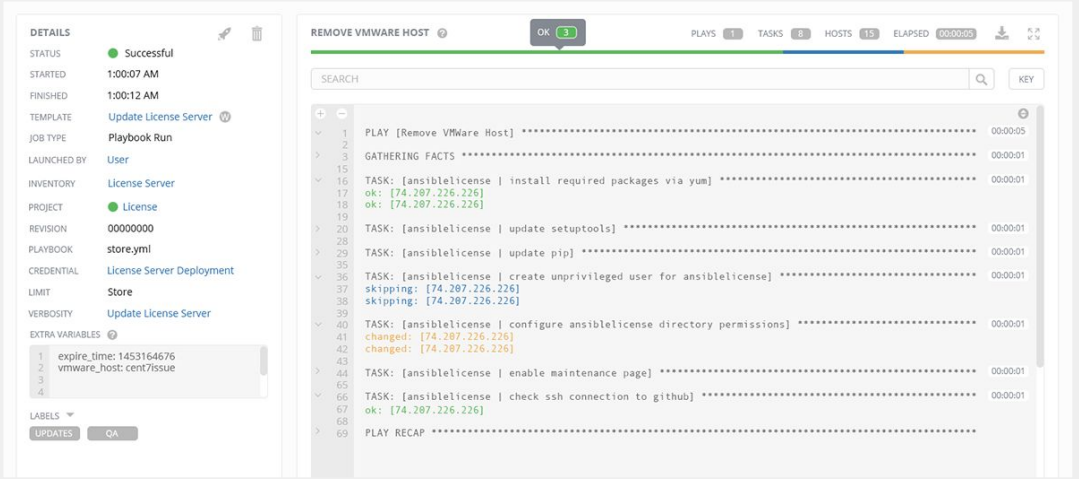



ADMINS
USERS
ANSIBLE PLAYBOOKS
ANSIBLE CLI & CI SYSTEMS
ANSIBLE TOWER
ROLE-BASED ACCESS CONTROL
KNOWLEDGE & VISIBILITY
SCHEDULED & CENTRALIZED JOBS
SIMPLE USER INTERFACE
TOWER API
ANSIBLE
OPEN SOURCE MODULE LIBRARY
PLUGINS
PYTHON CODEBASE
TRANSPORT
SSH, WINRM, ETC.
AUTOMATE YOUR ENTERPRISE
INFRASTRUCTURE
LINUX, WINDOWS, UNIX ...
NETWORKS
ARISTA, CISCO, JUNIPER ...
CONTAINERS
DOCKER, LXC ...
CLOUD
AWS, GOOGLE CLOUD, AZURE ...
SERVICES
DATABASES, LOGGING, SOURCE CONTROL MANAGEMENT...
USE CASES
PROVISIONING
CONFIGURATION MANAGEMENT
APP DEPLOYMENT
CONTINUOUS DELIVERY
SECURITY & COMPLIANCE
ORCHESTRATION

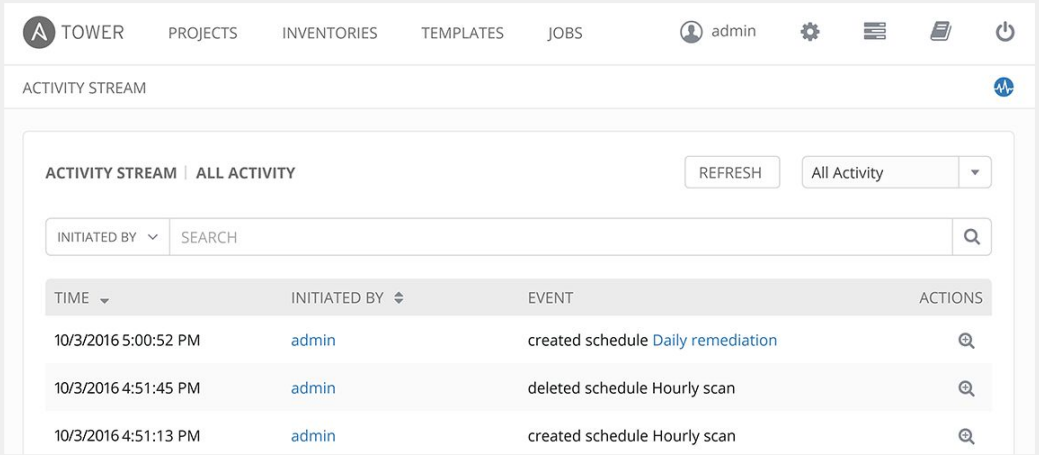
ANSIBLE TOWER

JOB STATUS UPDATE

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



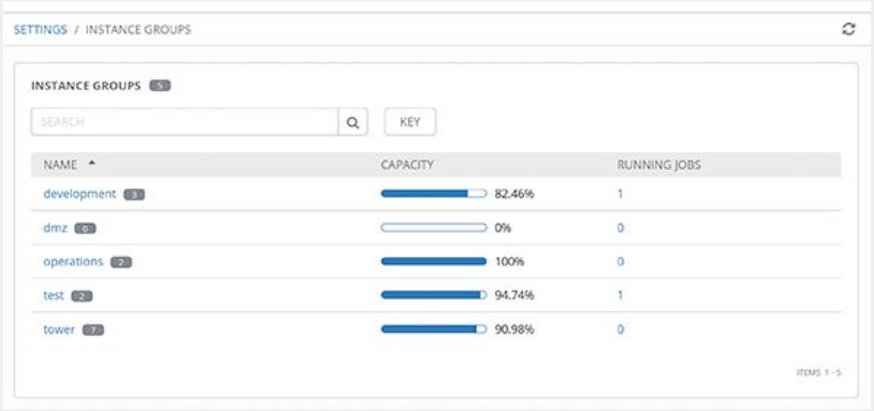
ANSIBLE TOWER



ACTIVITY STREAM

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

ANSIBLE TOWER

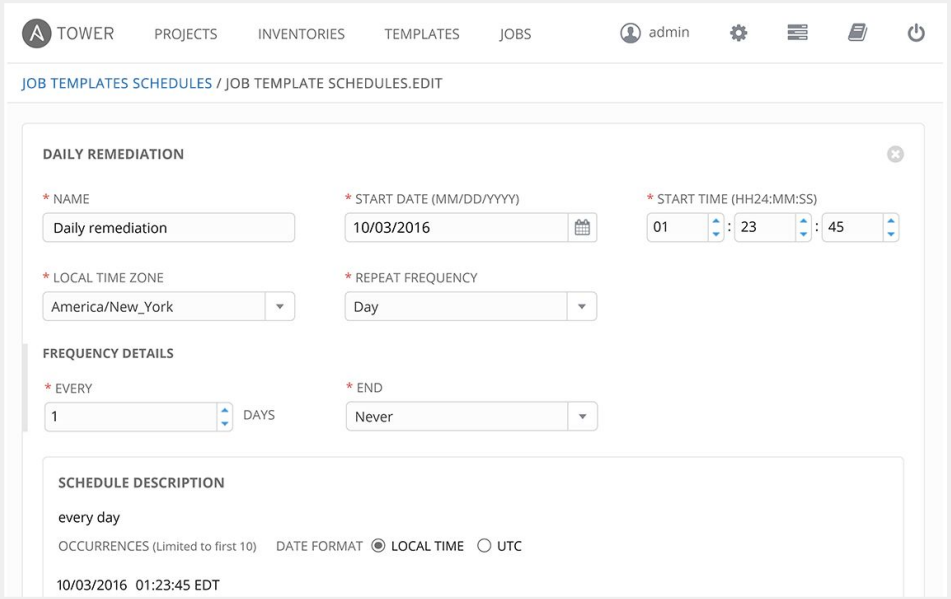


SCALE-OUT CLUSTERING

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



SCHEDULE JOBS

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

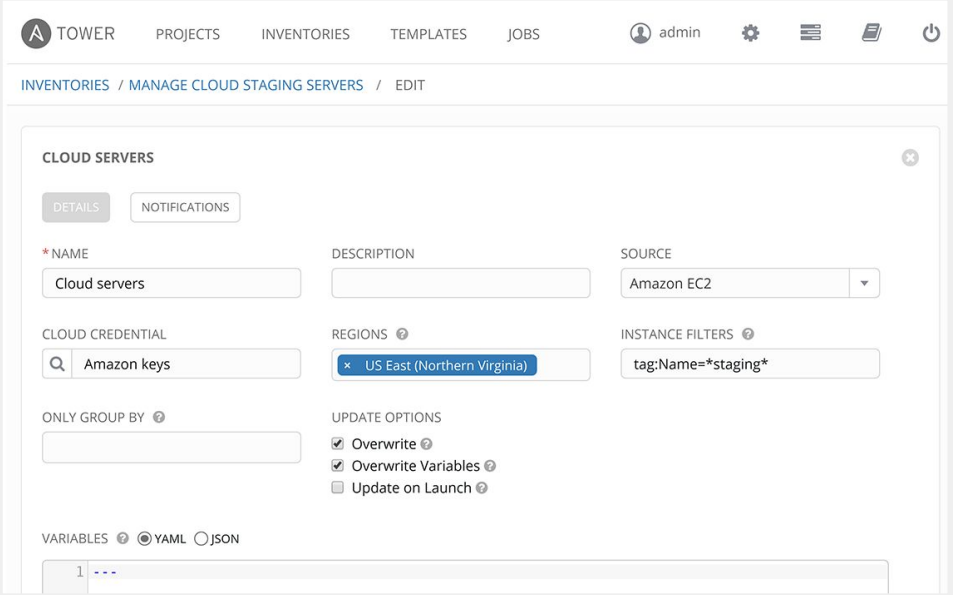
ANSIBLE TOWER

MANAGE AND TRACK YOUR INVENTORY

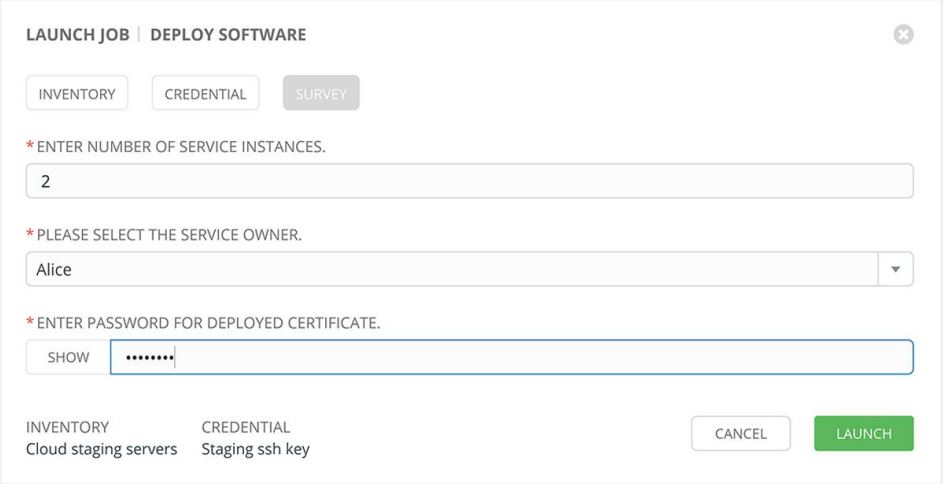
Tower's **inventory syncing** and **provisioning callbacks** allow nodes to request configuration on demand, enabling autoscaling.

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.



ANSIBLE TOWER



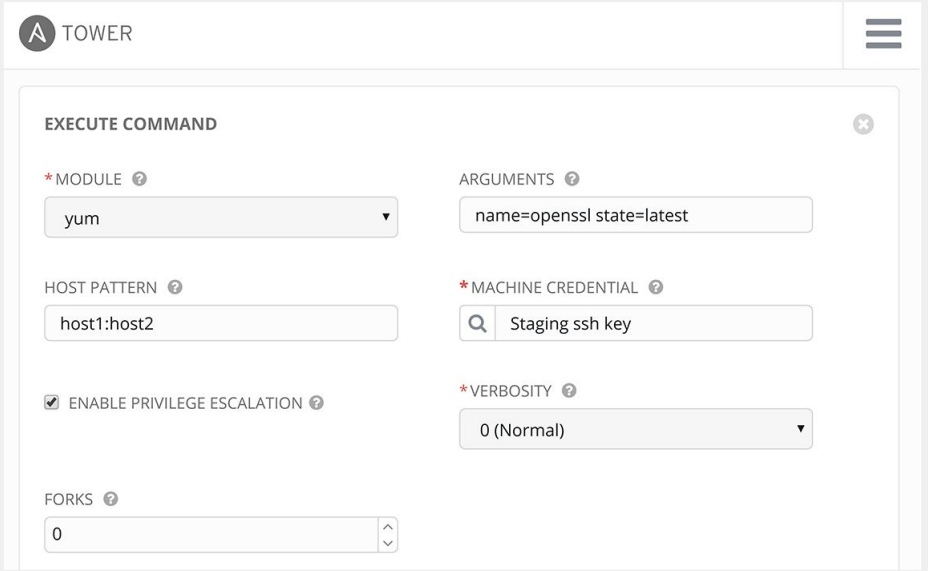
SELF-SERVICE IT

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

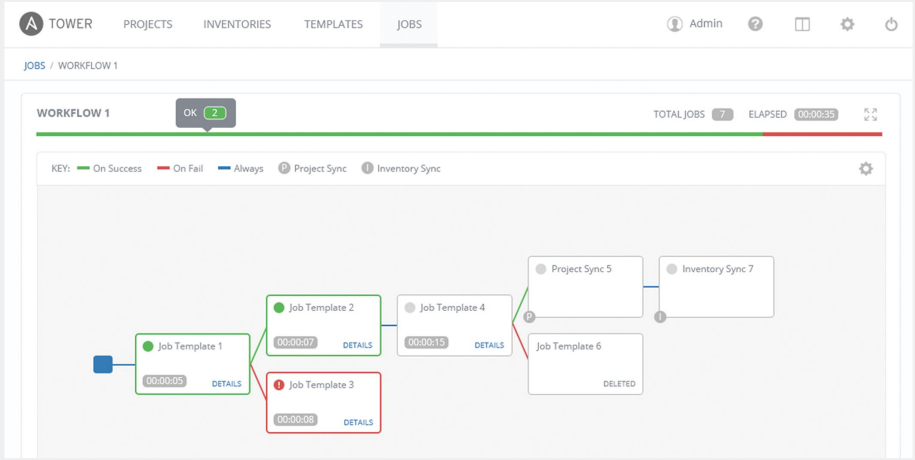
Run simple tasks on any hosts with Tower's **remote command execution**. Add users or groups, reset passwords, restart a malfunctioning service or patch a critical security issue, quickly.



ANSIBLE TOWER

MULTI-PLAYBOOK WORKFLOWS

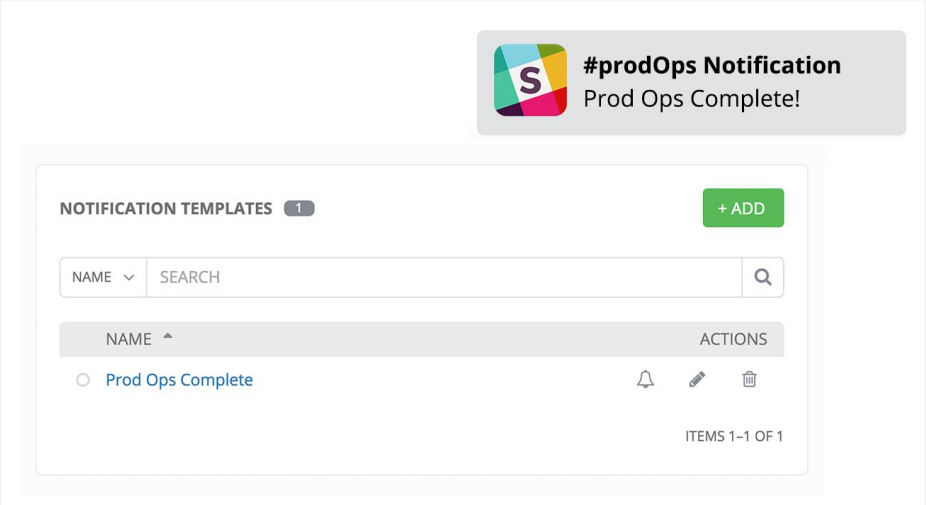
Tower's multi-Playbook workflows chains any number of Playbooks together to create a single workflow. Different Jobs can be run depending on success or failure of the prior Playbook.



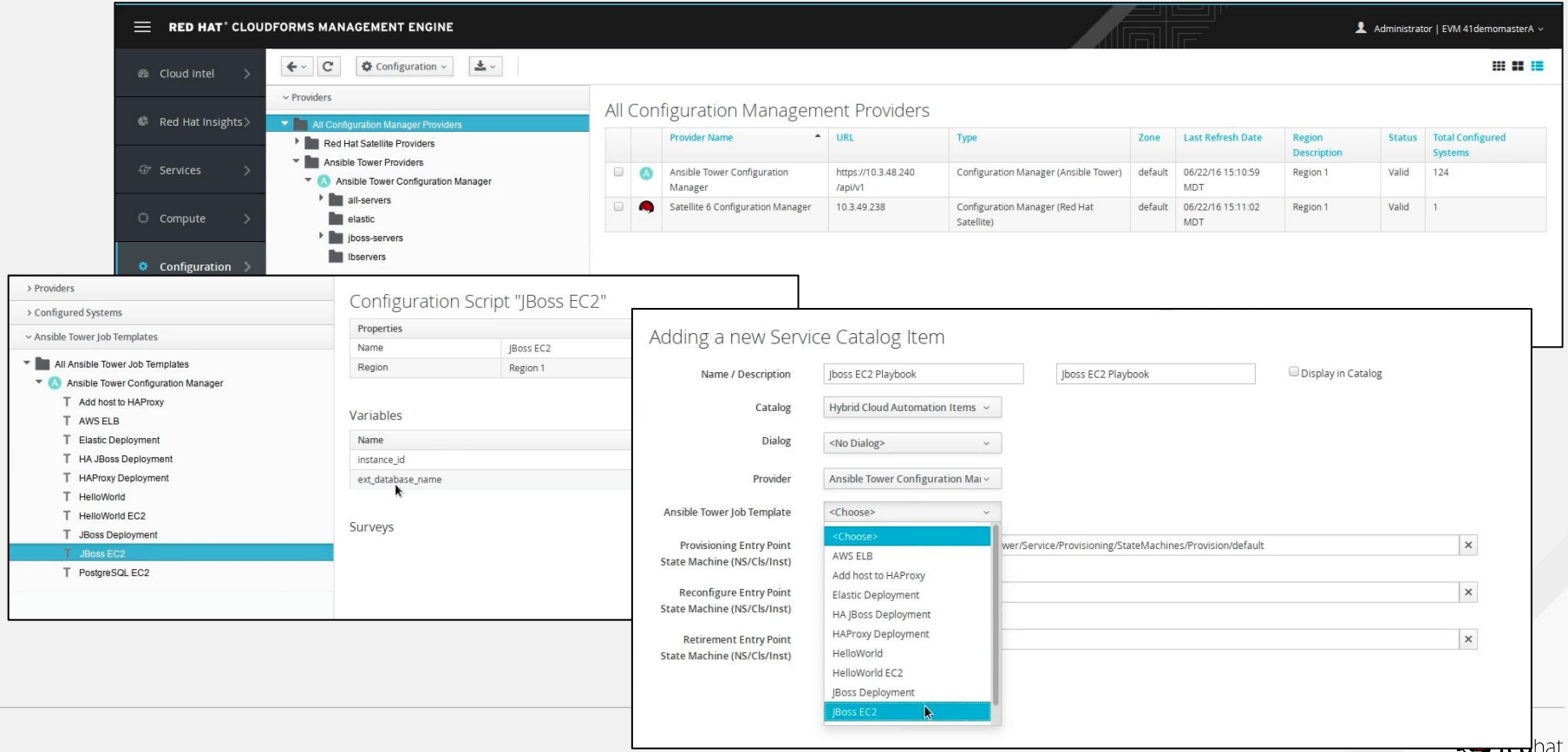
ANSIBLE TOWER

INTEGRATED NOTIFICATIONS

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



Leverage Ansible from CloudForms



http://people.redhat.com/mkoch

=> Training May 2018 Walldorf@HPE

THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/RedHatNews


youtube.com/user/RedHatVideos