Red Hat
Ansible
Automation

Ansible Network Automation

Introduction to Ansible for network engineers and operators

Mauricio Santacruz Delgado
Senior Solution Architect
Red Hat

Red Hat | intel

What are we going to talk about

Automatización	Ansible	Dev/Net/Sec Ops
Silos	Networking	Tower
Playbooks	Playbooks	etc

Automation happens when one person meets a problem they never want to solve again

Introduction

Topics Covered:

- Why Network Automation?
- How Ansible Network Automation works

Red Hat Ansible Automation

Red Hat intel

71%

of networks are still driven manually via CLI

Source: Gartner, *Look Beyond Network Vendors for Innovation.* January 2018

THE WORLD IS AUTOMATING

Those who succeed in automation will win

ACCELERATE

INTEGRATE

COLLABORATE

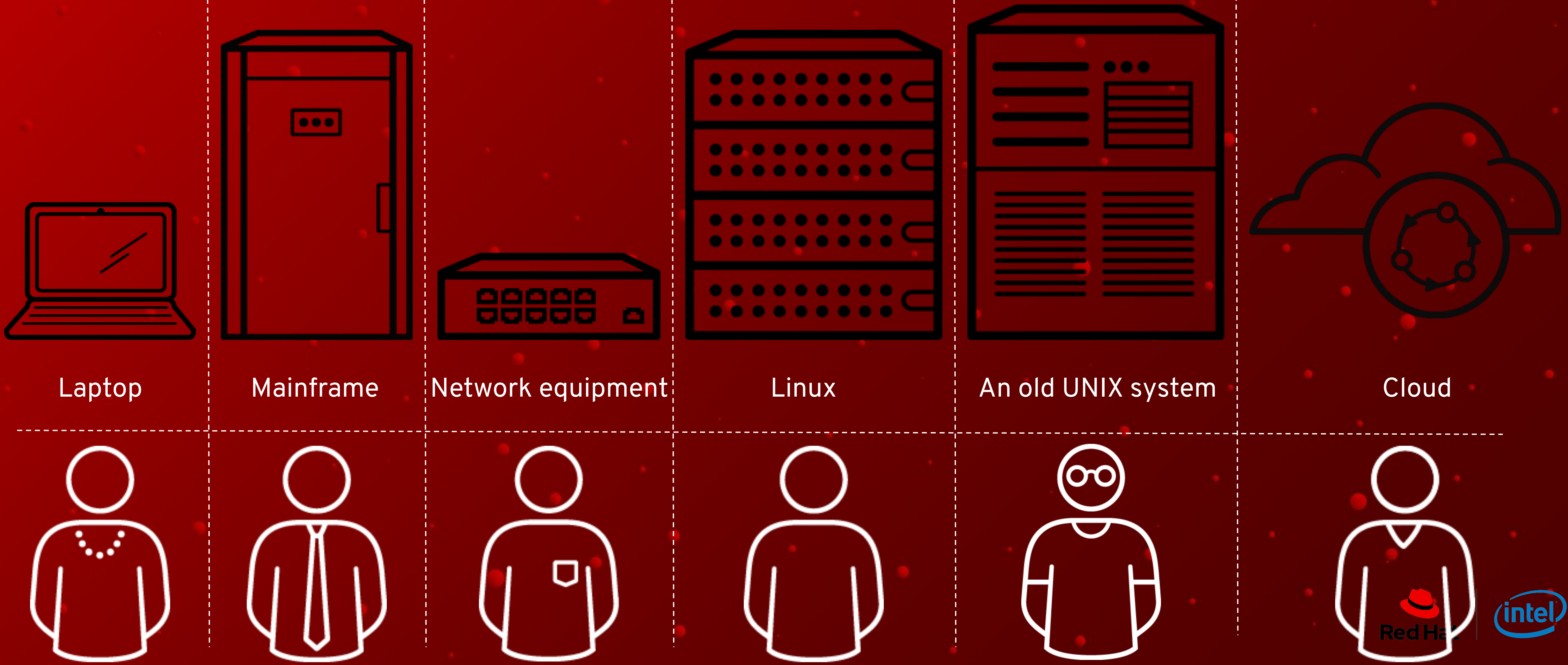
NOT AS SIMPLE ANYMORE

What do you need to automate today?

What do you need to automate today?

Laptop	Mainframe	Network equipment	Linux	An old UNIX system	Cloud
System A Language Z	System B Language Y	System C Language X	System D Language W	System E Language V	System F Language U

What do you need to automate today?



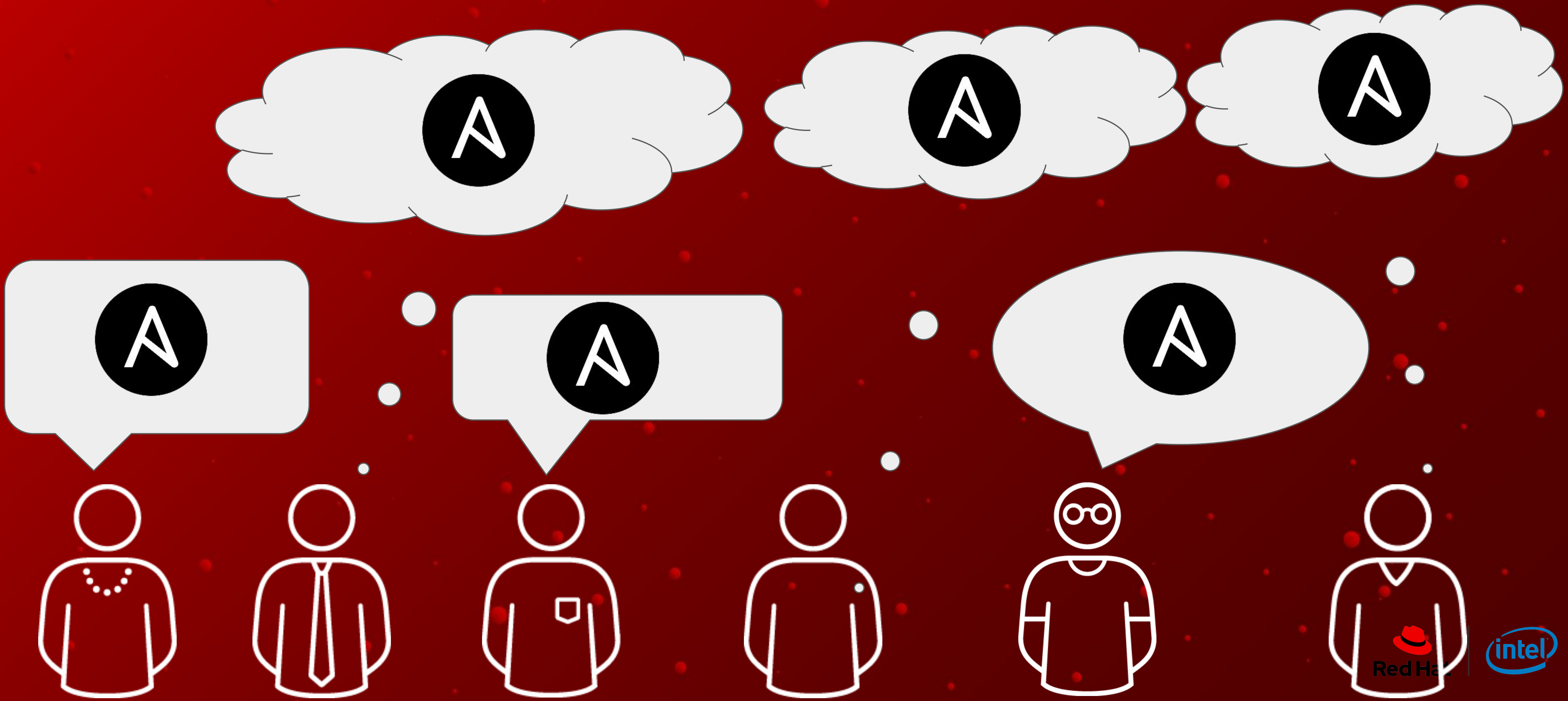
Who do you need to collaborate with today?

Världen är platt.
El mundo es redondo.
Jorden er flat.
Die Welt ist rund.
Verden er trekantet.
Maa on pannukakku.

What do you need to learn today?

What do you need to learn today?

Who do you need to collaborate with today?

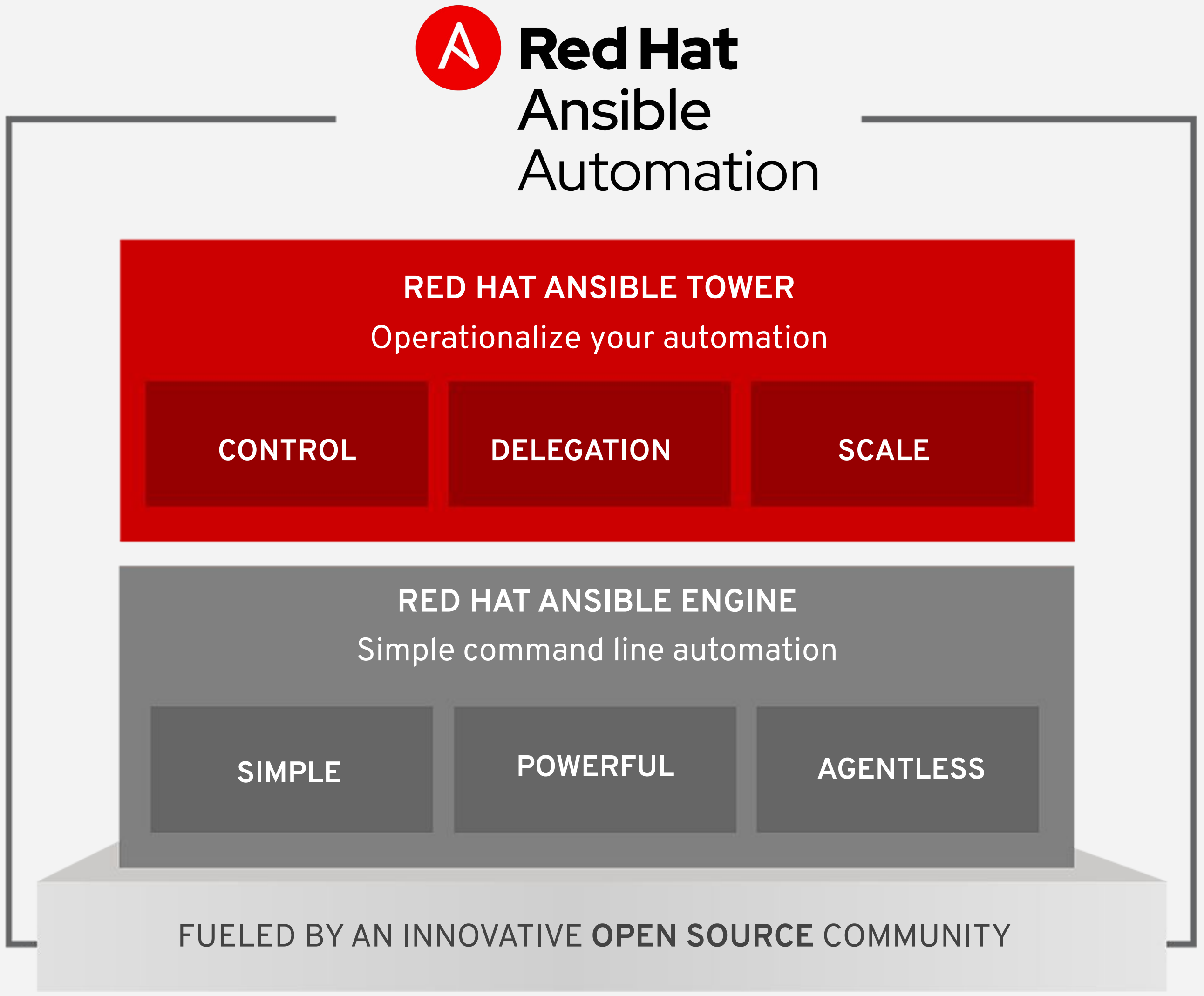


What is Ansible Automation?

Ansible Automation is the enterprise **framework** for automating across IT operations.

Ansible Engine runs Ansible Playbooks, the automation **language** that can perfectly describe an IT application infrastructure.

Ansible Tower allows you **scale** IT automation, manage complex deployments and speed productivity.



Red Hat Ansible Automation
RED HAT ANSIBLE TOWER
Operationalize your automation
CONTROL
DELEGATION
SCALE
RED HAT ANSIBLE ENGINE
Simple command line automation
SIMPLE
POWERFUL
AGENTLESS
FUELED BY AN INNOVATIVE OPEN SOURCE COMMUNITY

WHY ANSIBLE?
(for networks)

SIMPLE

For operators, not developers

Download and go

Existing knowledge reuse

POWERFUL

Connect via Plugins

Easy platform enablement

Leverage Linux tools

AGENTLESS

Ideal for network gear

No agents to exploit or update

Standards-based SSH

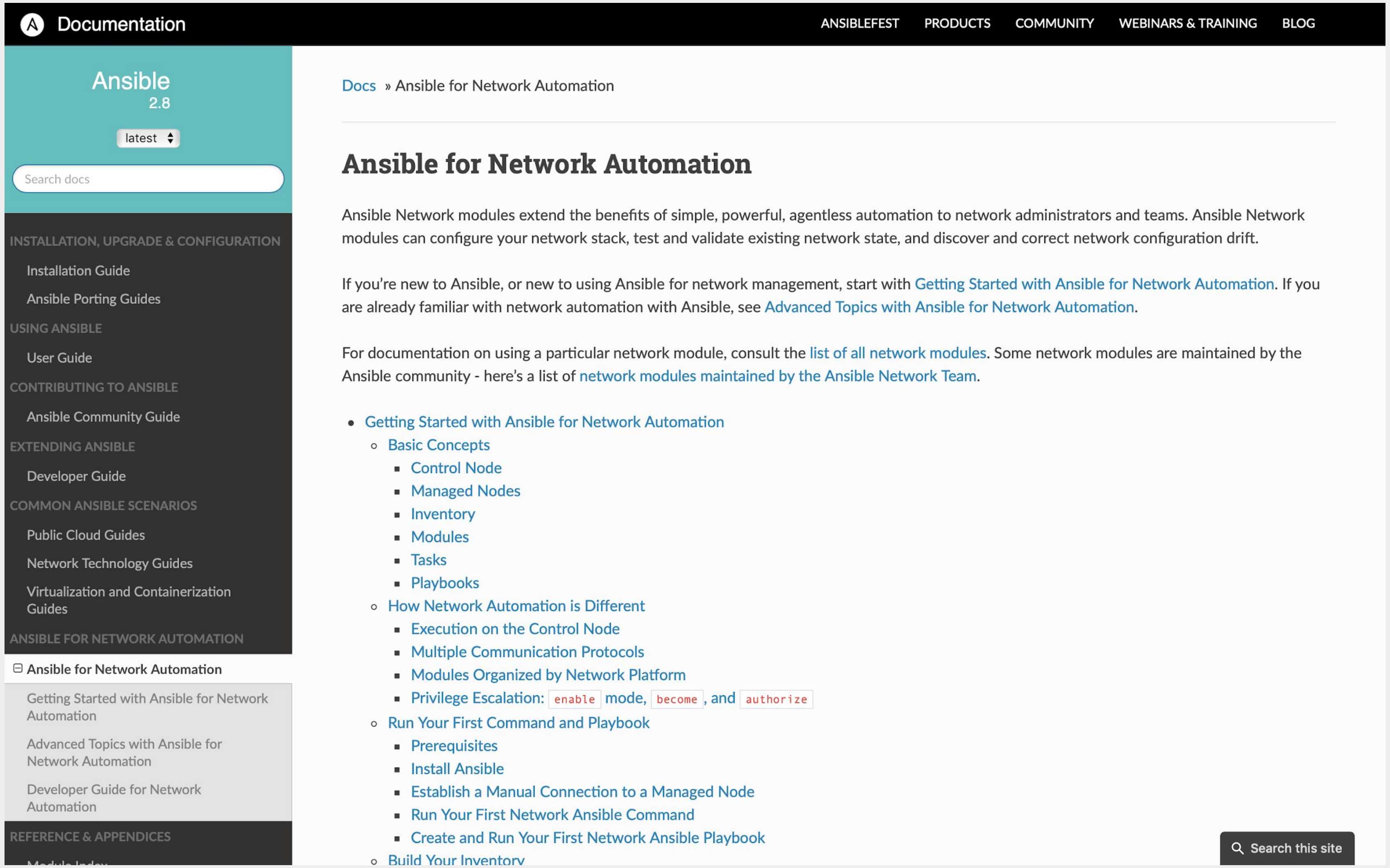
ANSIBLE NETWORK AUTOMATION

65+	1000+	15*
Network Platforms	Network Modules	Galaxy Network Roles

ansible.com/for/networks
galaxy.ansible.com/ansible-network

Roles developed and maintained by Ansible Network Engineering

“Ansible for Network Automation” Documentation



Documentation

ANSIBLEFEST PRODUCTS COMMUNITY WEBINARS & TRAINING BLOG

Ansible 2.8

latest

Search docs

INSTALLATION, UPGRADE & CONFIGURATION
- Installation Guide
- Ansible Porting Guides

USING ANSIBLE
- User Guide

CONTRIBUTING TO ANSIBLE
- Ansible Community Guide

EXTENDING ANSIBLE
- Developer Guide

COMMON ANSIBLE SCENARIOS
- Public Cloud Guides
- Network Technology Guides
- Virtualization and Containerization Guides

ANSIBLE FOR NETWORK AUTOMATION
- ⊟ Ansible for Network Automation
  - Getting Started with Ansible for Network Automation
  - Advanced Topics with Ansible for Network Automation
  - Developer Guide for Network Automation

REFERENCE & APPENDICES

Docs » Ansible for Network Automation

Ansible for Network Automation

Ansible Network modules extend the benefits of simple, powerful, agentless automation to network administrators and teams. Ansible Network modules can configure your network stack, test and validate existing network state, and discover and correct network configuration drift.

If you’re new to Ansible, or new to using Ansible for network management, start with Getting Started with Ansible for Network Automation. If you are already familiar with network automation with Ansible, see Advanced Topics with Ansible for Network Automation.

For documentation on using a particular network module, consult the list of all network modules. Some network modules are maintained by the Ansible community - here’s a list of network modules maintained by the Ansible Network Team.

Search this site

http://bit.ly/AnsibleNetwork

What can I do using Ansible?

Automate the deployment and management of your entire IT footprint.

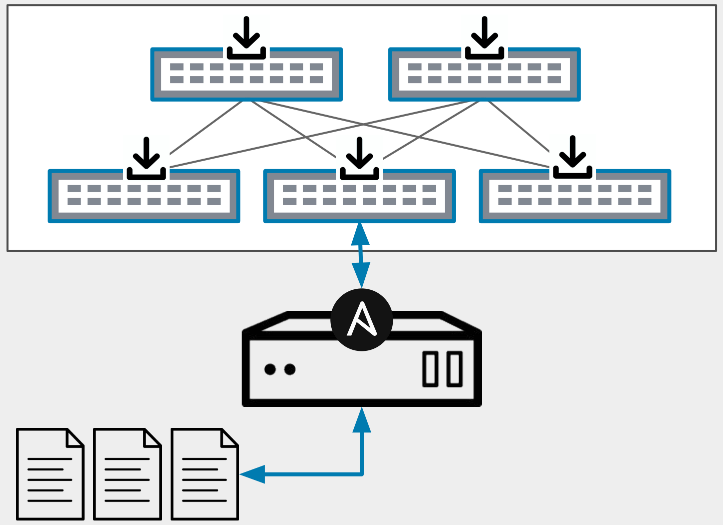
Do this...

Orchestration	Configuration Management	Application Deployment	Provisioning	Continuous Delivery	Security and Compliance

On these...

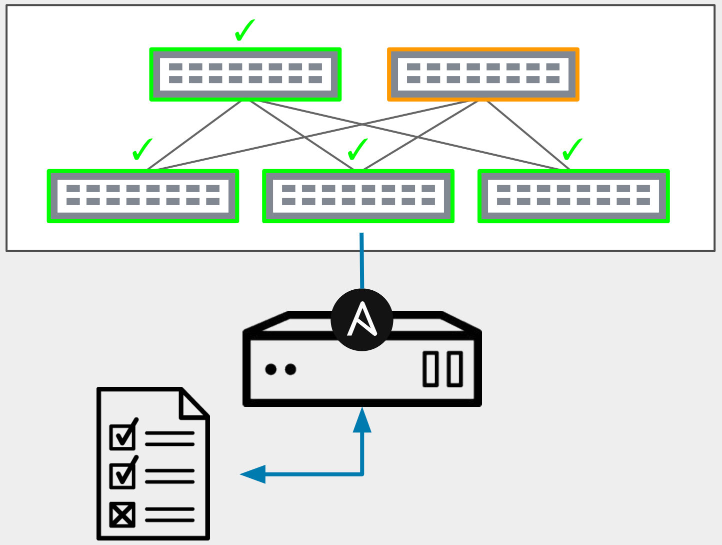
Firewalls	Load Balancers	Applications	Containers	Clouds
Servers	Infrastructure	Storage	Network Devices	And more...

Common use cases



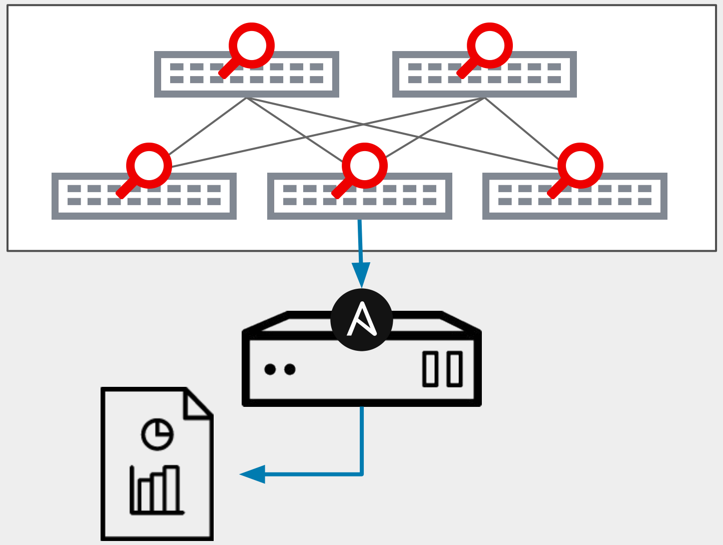
Backup and Restore

- Schedule backups
- Restore from any timestamp
- Build workflows that rollback



Configuration Compliance

- Check configuration standards
- Track configuration drift
- Enforce configuration policy

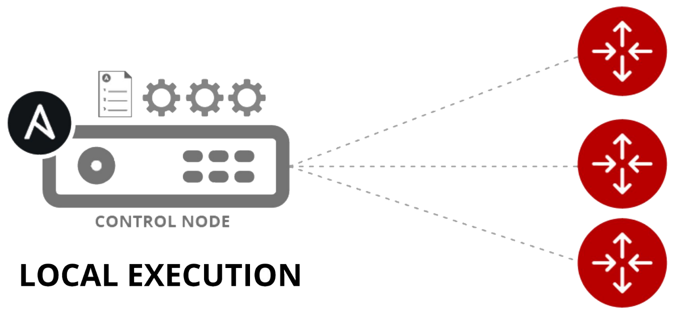


Dynamic Documentation

- Build reports
- Grab software versions, MTU, interfaces status
- Audit system services and other common config

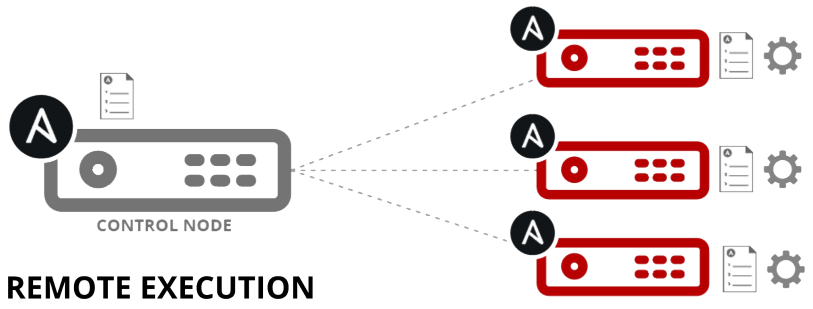
How Ansible Network Automation works

Module code is executed locally on the control node



NETWORKING DEVICES

Module code is copied to the managed node, executed, then removed



LINUX/WINDOWS HOSTS

Ansible automates technologies you use

Time to automate is measured in minutes

Cloud

AWS
Azure
Digital Ocean
Google
OpenStack
Rackspace
+more

Operating Systems

Rhel And Linux
Unix
Windows
+more

Virt & Container

Docker
VMware
RHV
OpenStack
OpenShift
+more

Storage

Netapp
Red Hat Storage
Infinidat
+more

Windows

ACLs
Files
Packages
IIS
Regedits
Shares
Services
Configs
Users
Domains
+more

Network

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

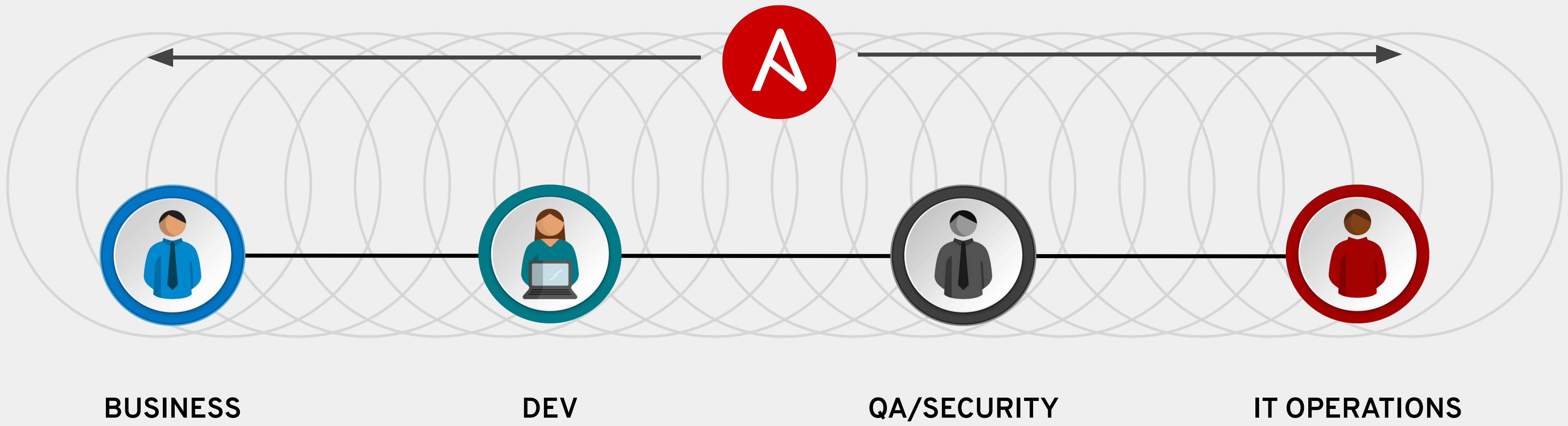
Devops

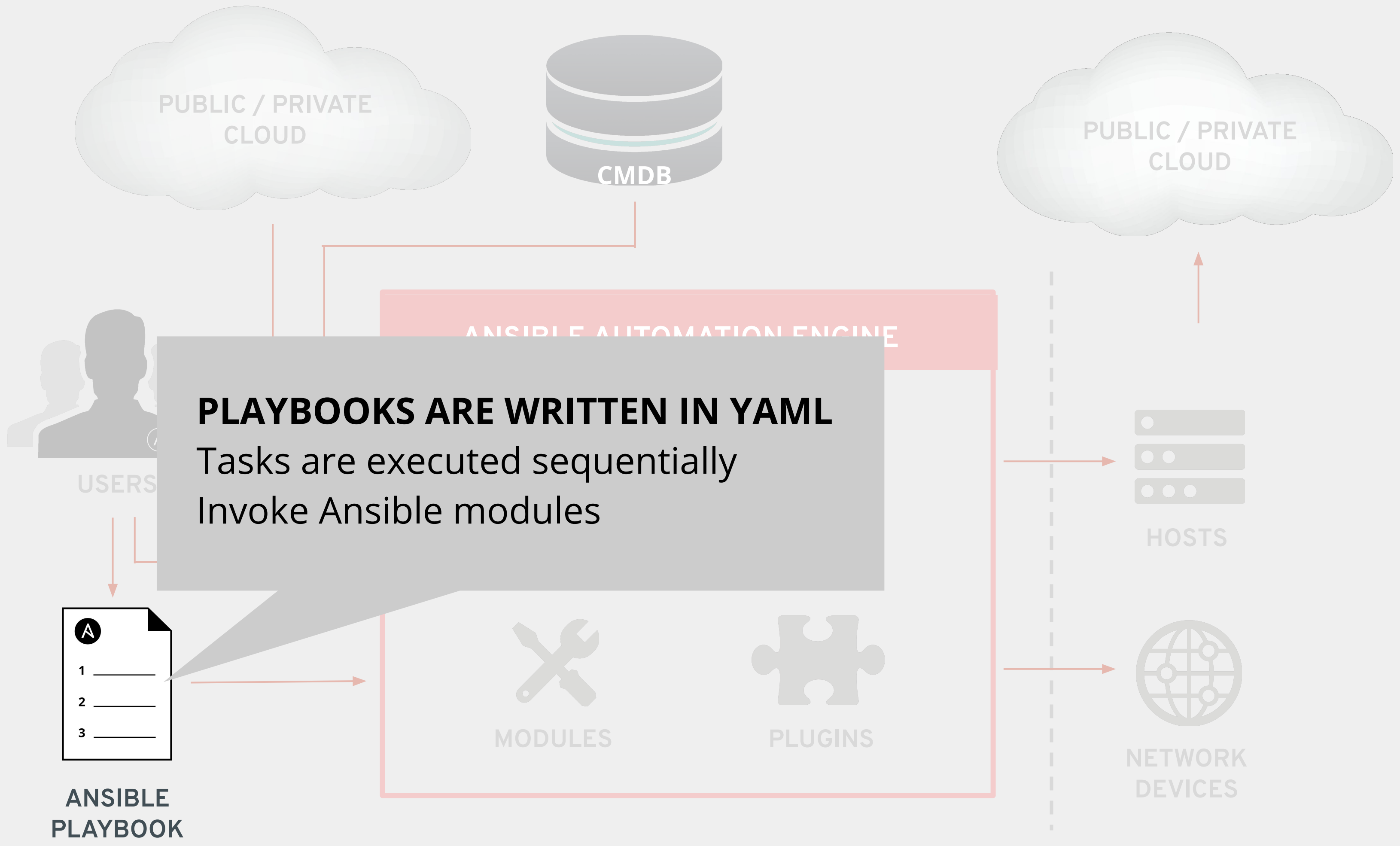
Jira
GitHub
Vagrant
Jenkins
Bamboo
Atlassian
Subversion
Slack
Hipchat
+more

Monitoring

Dynatrace
Airbrake
BigPanda
Datadog
LogicMonitor
Nagios
New Relic
PagerDuty
Sensu
StackDriver
Zabbix
+more

ANSIBLE IS THE UNIVERSAL LANGUAGE

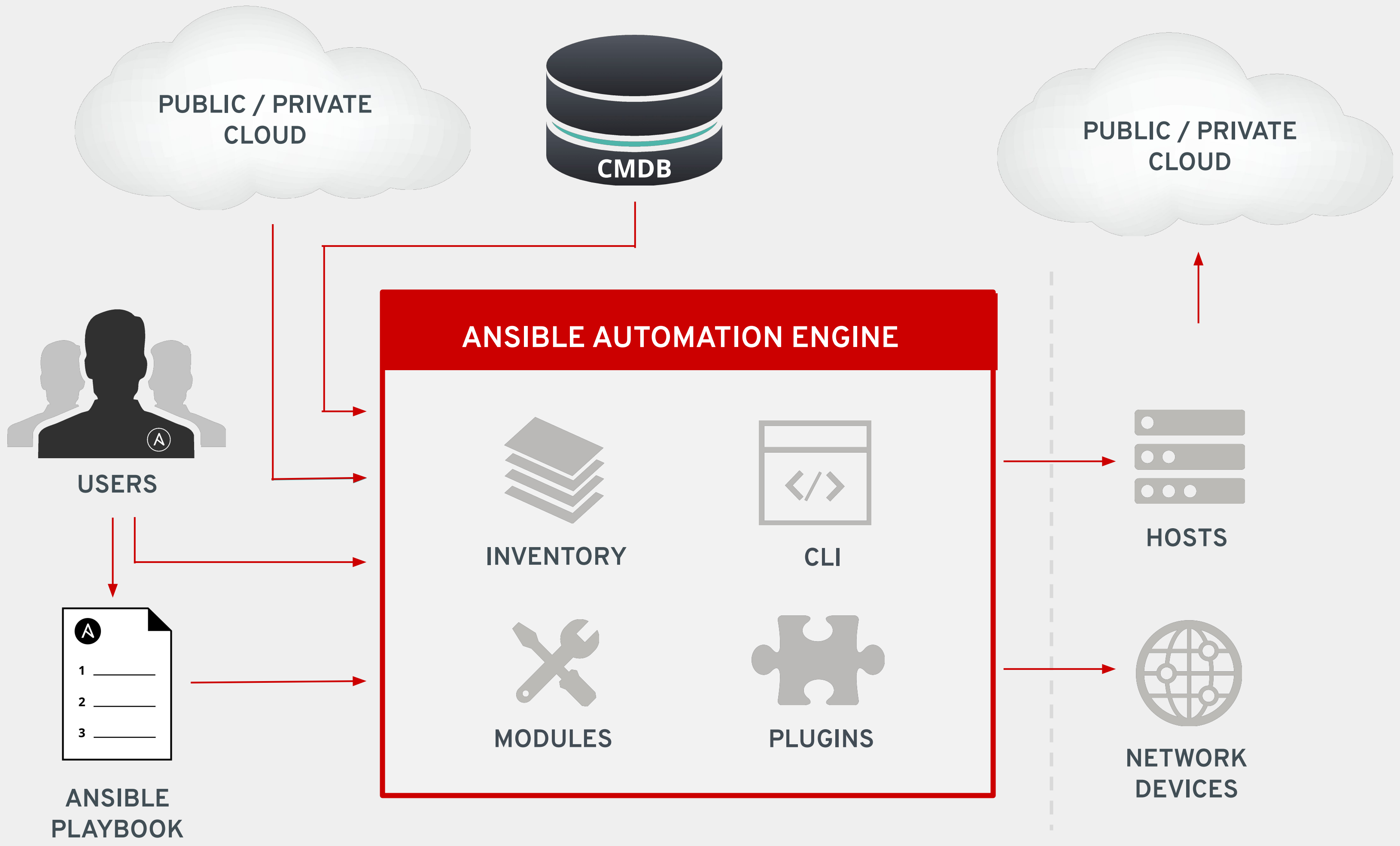




PLAYBOOKS ARE WRITTEN IN YAML

Tasks are executed sequentially

Invoke Ansible modules


PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE CLOUD
USERS
ANSIBLE AUTOMATION ENGINE
INVENTORY
CLI
MODULES
PLUGINS
HOSTS
NETWORK DEVICES
1
2
3
ANSIBLE PLAYBOOK

Red Hat

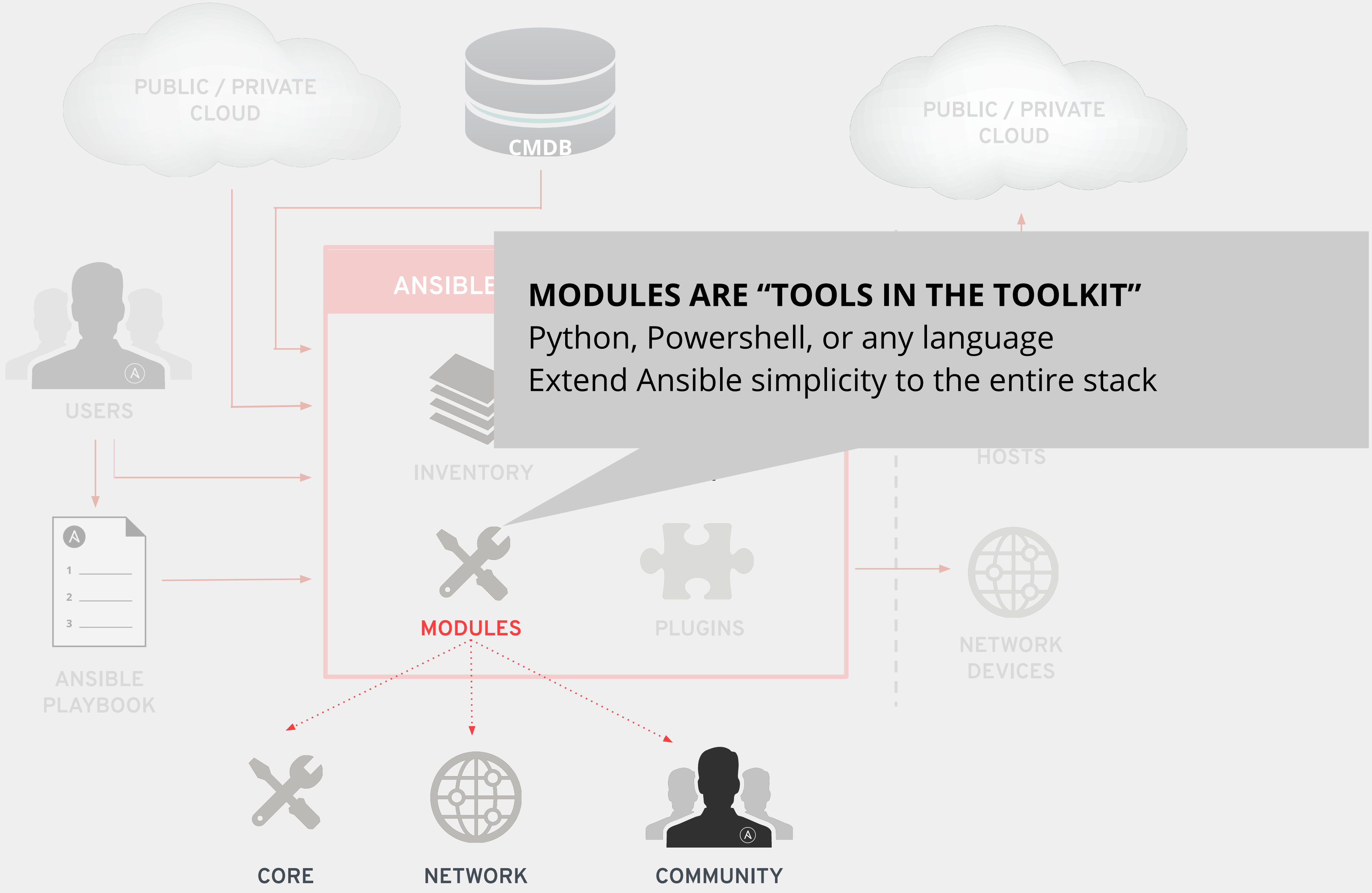
```yaml
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
   http_port: 80

  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
```

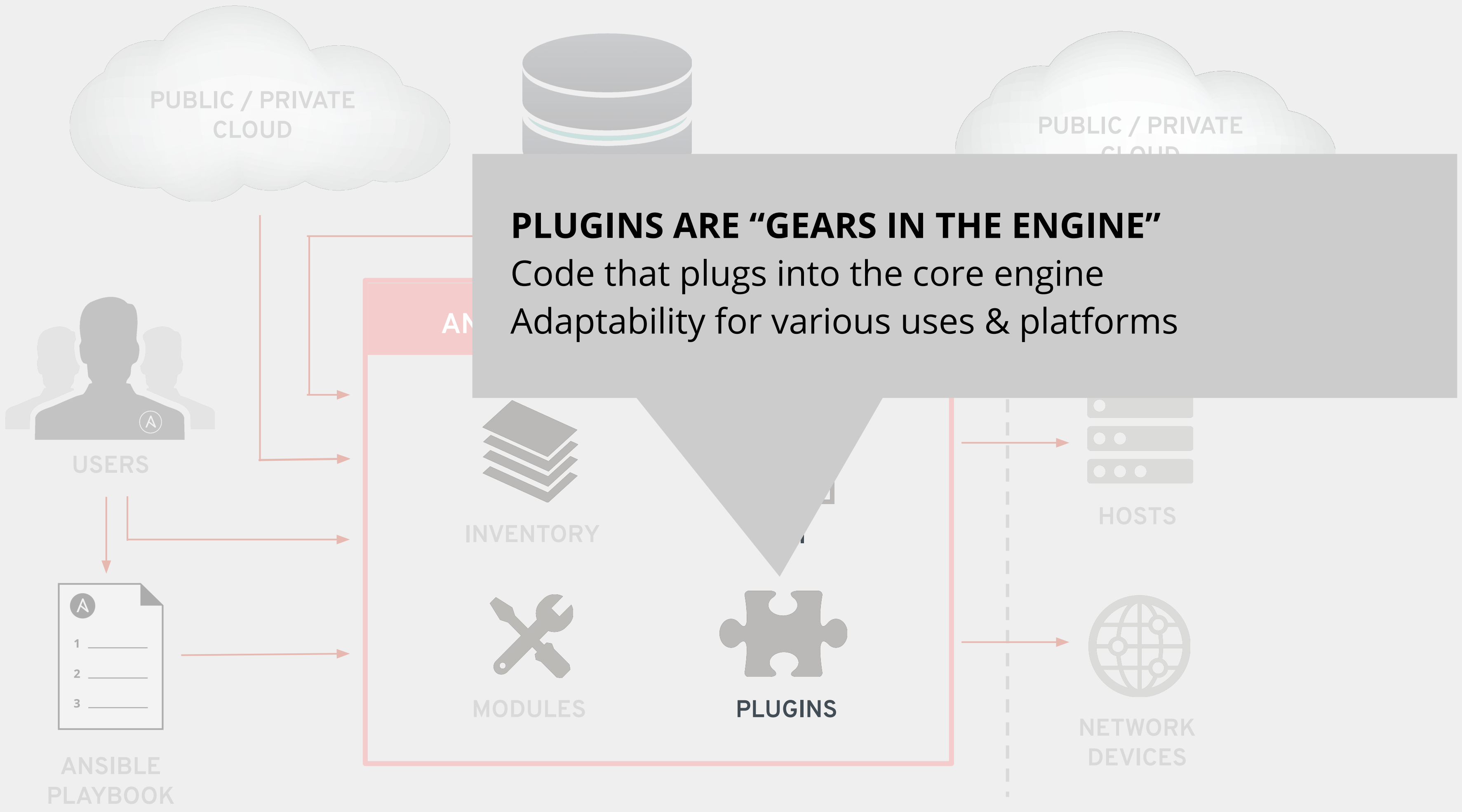
Red Hat

MODULES ARE “TOOLS IN THE TOOLKIT”

Python, Powershell, or any language

Extend Ansible simplicity to the entire stack

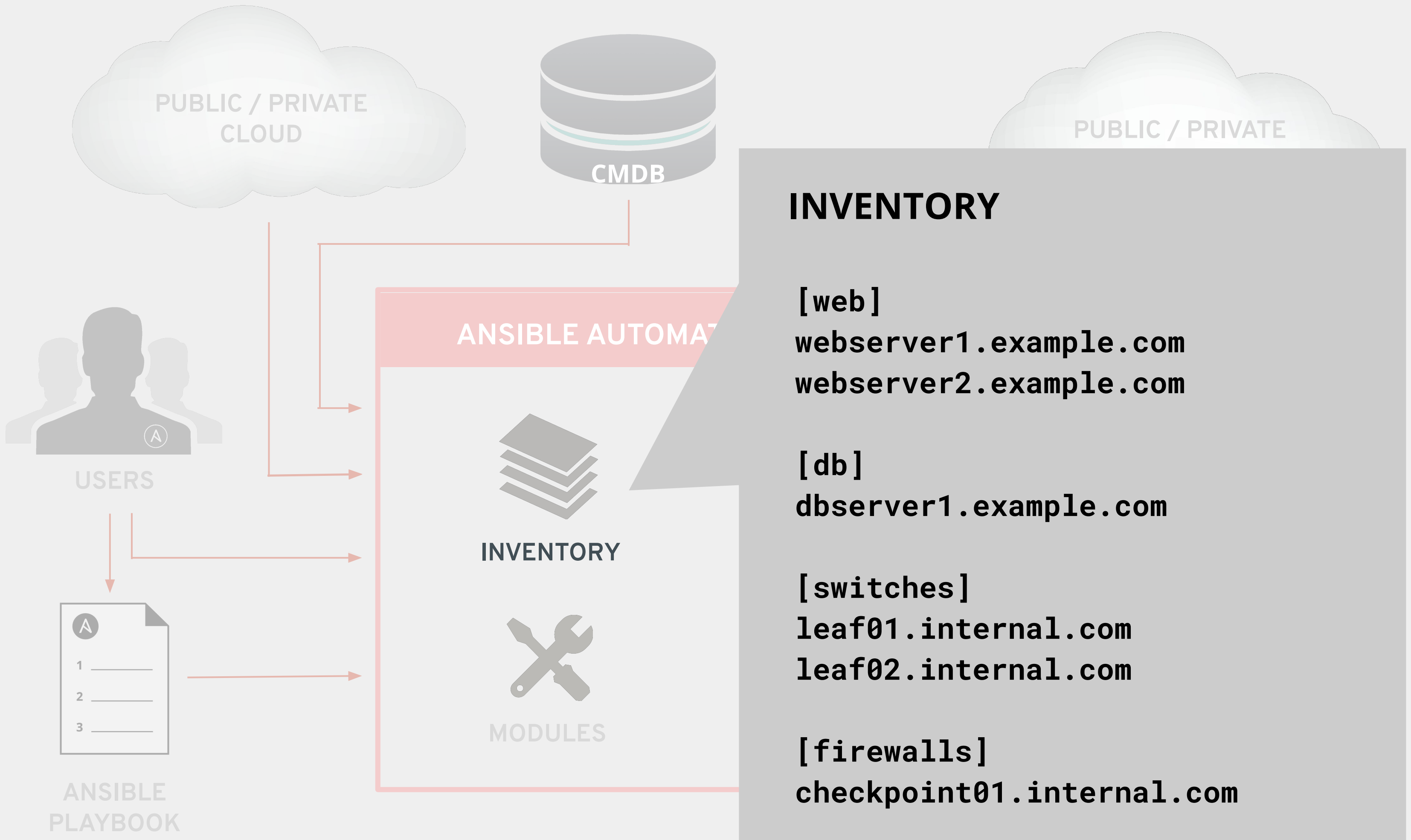




PLUGINS ARE “GEARS IN THE ENGINE”

Code that plugs into the core engine

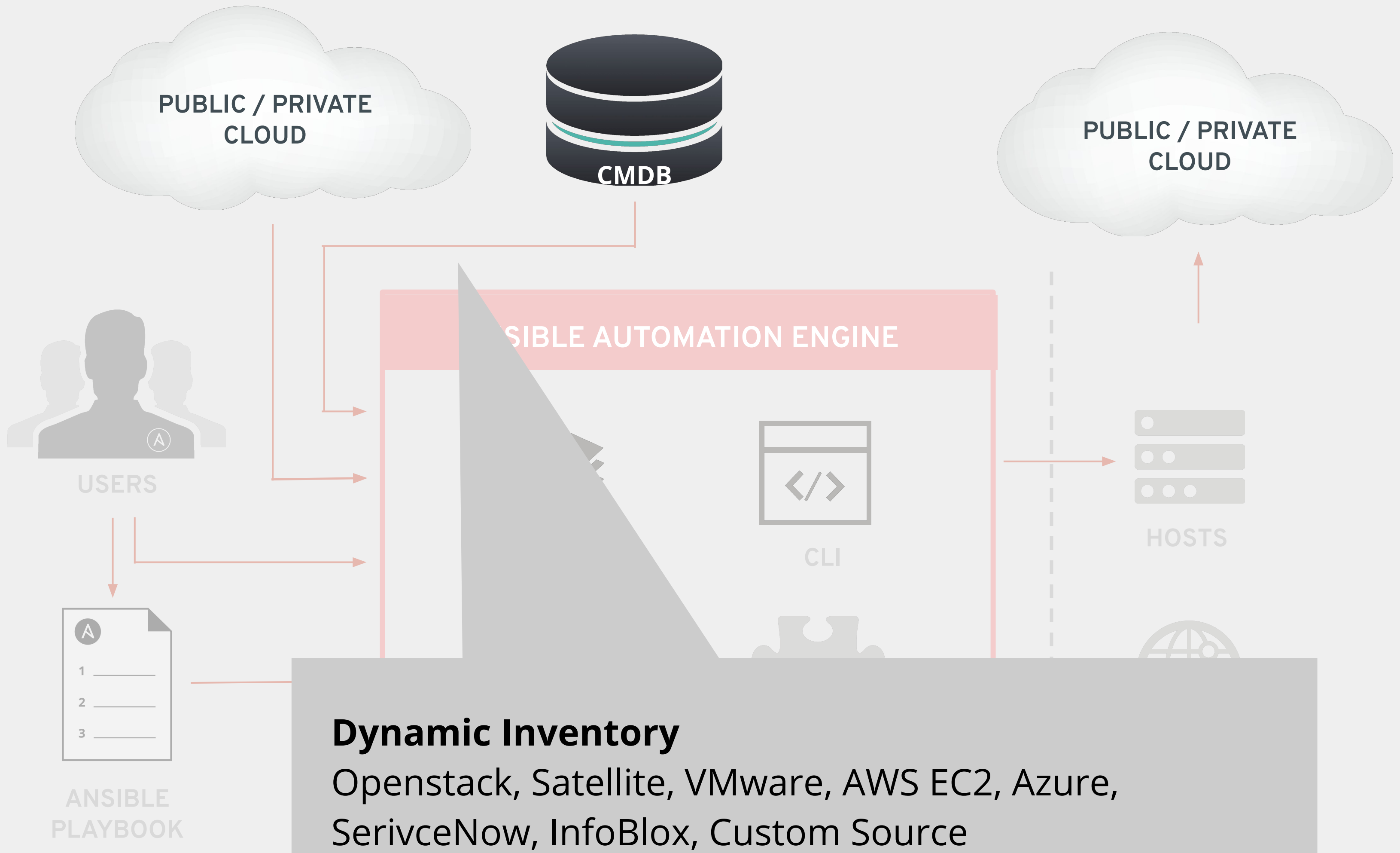
Adaptability for various uses & platforms



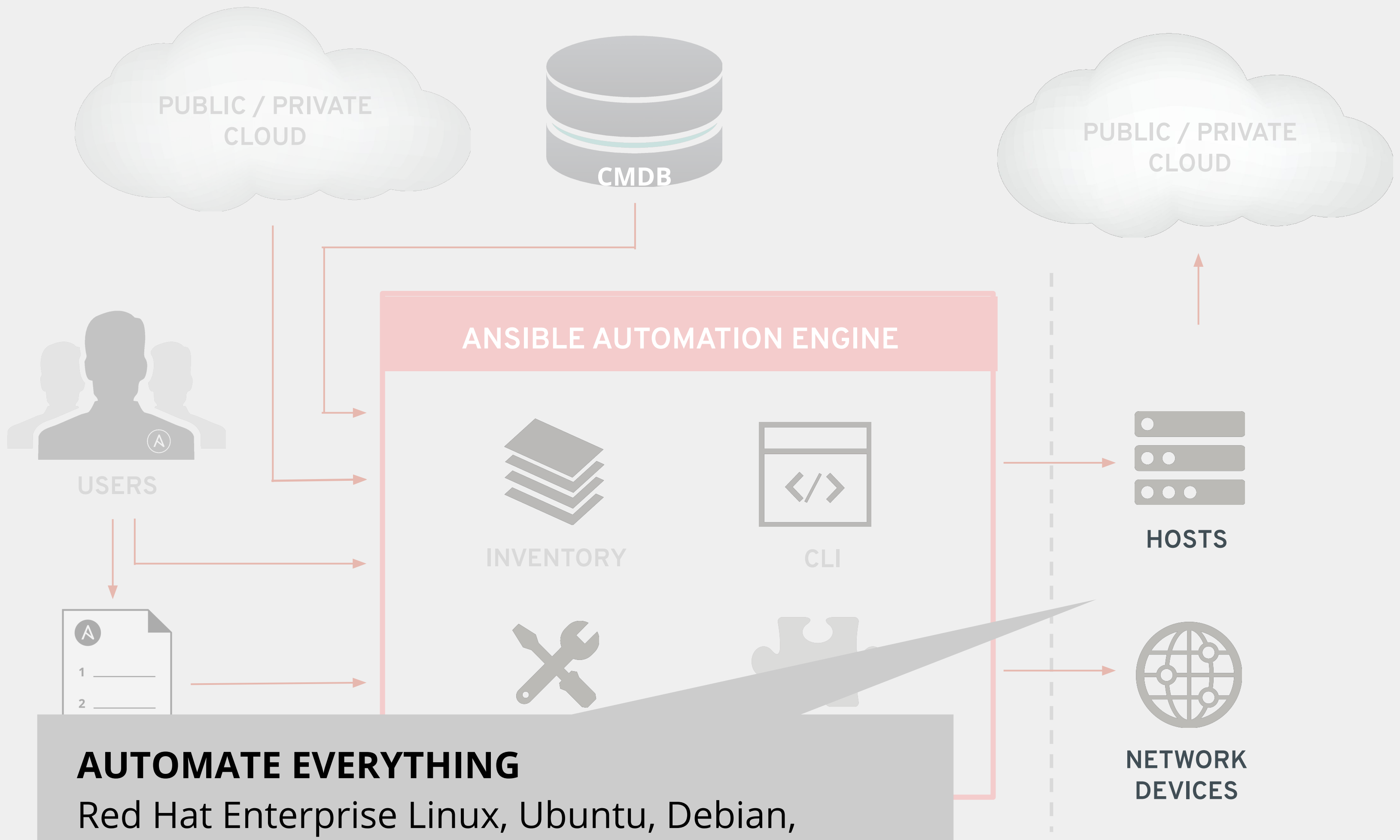
INVENTORY

```
[web]
webserver1.example.com
webserver2.example.com

[db]
dbserver1.example.com

[switches]
leaf01.internal.com
leaf02.internal.com

[firewalls]
checkpoint01.internal.com

[lb]
f5-01.internal.com
```

Red Hat



Dynamic Inventory
Openstack, Satellite, VMware, AWS EC2, Azure, SerivceNow, InfoBlox, Custom Source



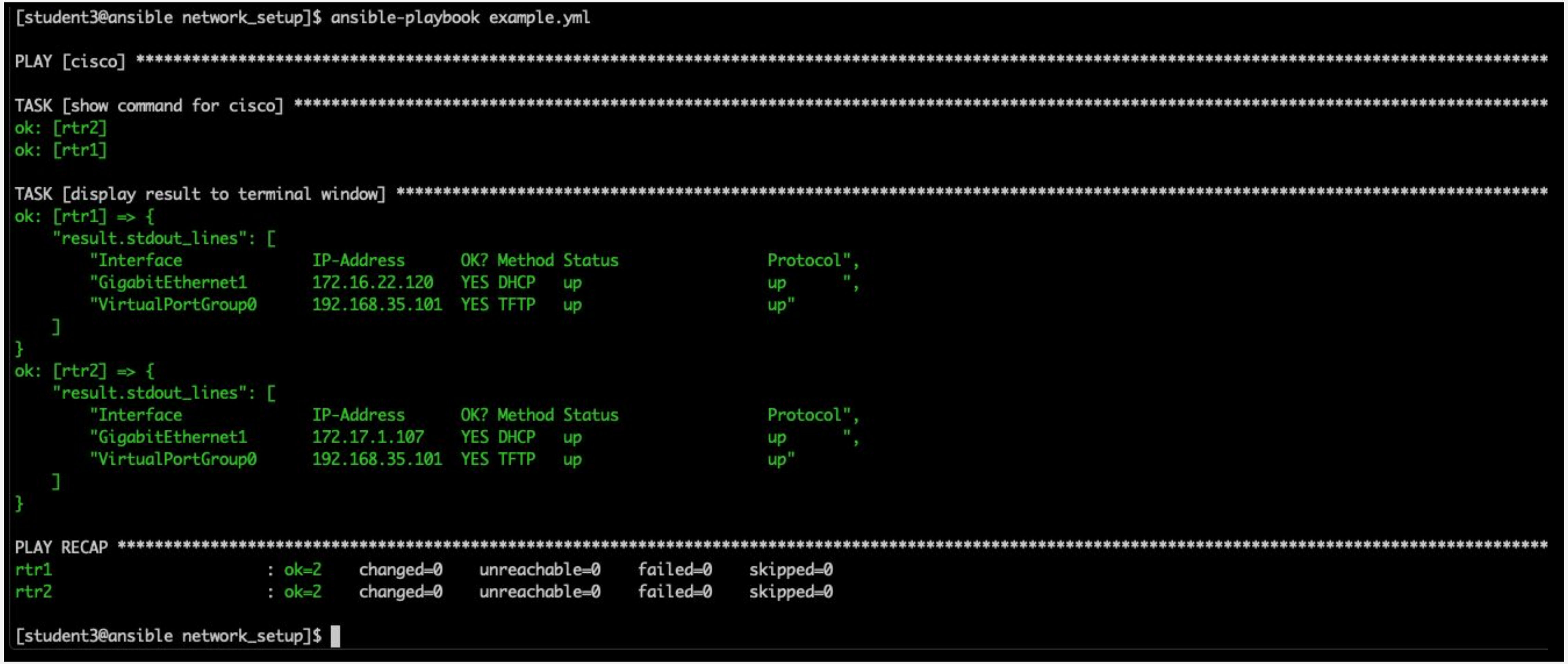
AUTOMATE EVERYTHING

Red Hat Enterprise Linux, Ubuntu, Debian,
Cisco routers, Arista switches, Juniper routers,
Windows hosts, Checkpoint firewalls and more

AUTOMATION FOR EVERYONE: **NETWORK ENGINEERS**

```
---
- hosts: cisco
  gather_facts: false
  connection: network_cli

  tasks:
   - name: show command for cisco
     cli_command:
        command: show ip int br
     register: result

   - name: display result to terminal window
     debug:
        var: result.stdout_lines
```

AUTOMATION FOR EVERYONE: **PLAYBOOK RESULTS**



```
[student3@ansible network_setup]$ ansible-playbook example.yml

PLAY [cisco] ***********************************************************************************************************

TASK [show command for cisco] ******************************************************************************************
ok: [rtr2]
ok: [rtr1]

TASK [display result to terminal window] *******************************************************************************
ok: [rtr1] => {
    "result.stdout_lines": [
        "Interface              IP-Address      OK? Method Status                Protocol",
        "GigabitEthernet1       172.16.22.120   YES DHCP   up                    up      ",
        "VirtualPortGroup0      192.168.35.101  YES TFTP   up                    up"
    ]
}
ok: [rtr2] => {
    "result.stdout_lines": [
        "Interface              IP-Address      OK? Method Status                Protocol",
        "GigabitEthernet1       172.17.1.107    YES DHCP   up                    up      ",
        "VirtualPortGroup0      192.168.35.101  YES TFTP   up                    up"
    ]
}

PLAY RECAP *************************************************************************************************************
rtr1                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rtr2                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0

[student3@ansible network_setup]$ 
```

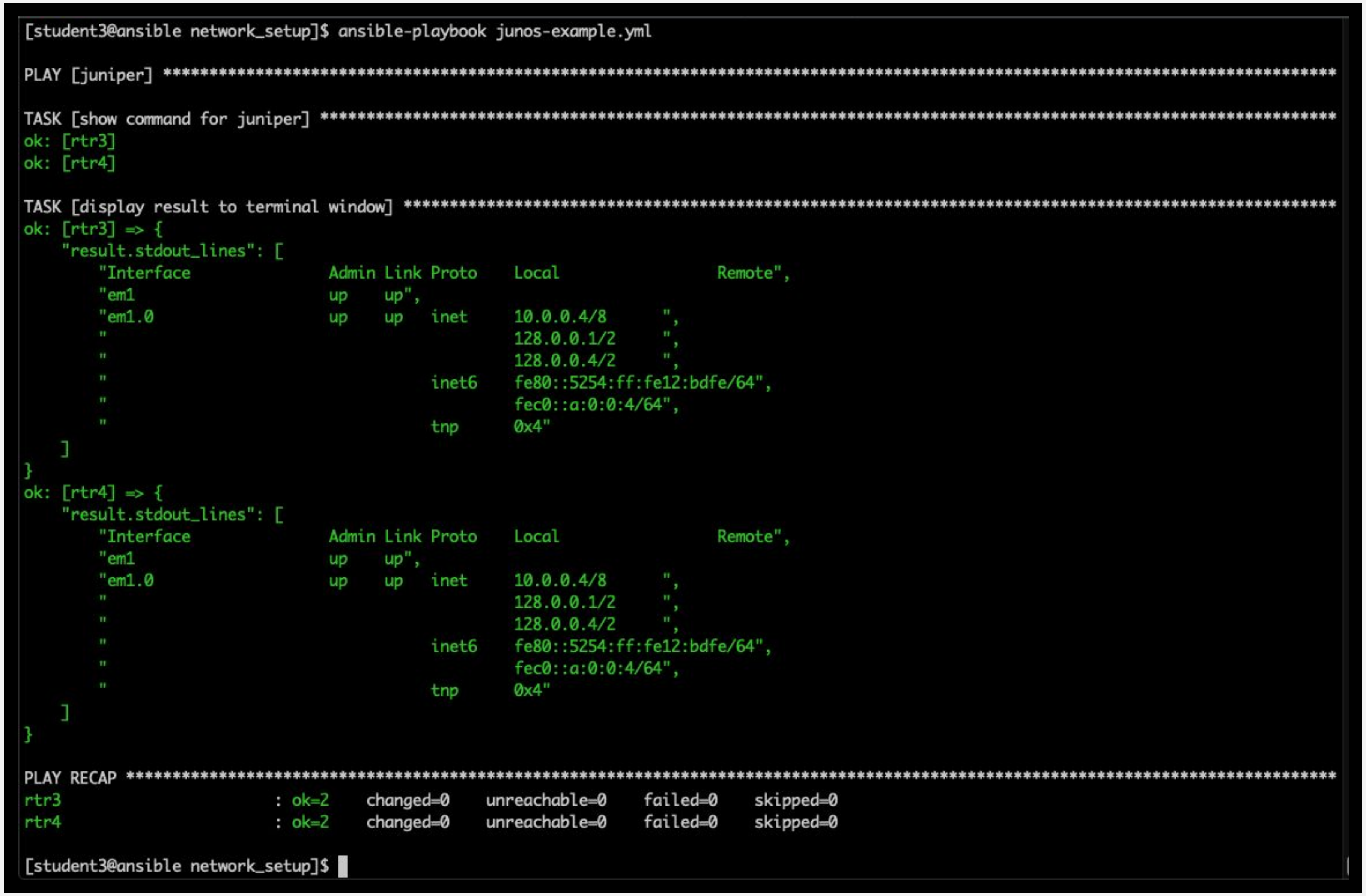
Red Hat

AUTOMATION FOR EVERYONE: **NETWORK ENGINEERS**

```
---
- hosts: juniper
  gather_facts: false
  connection: network_cli

  tasks:
   - name: show command for juniper
     cli_command:
        command: show interfaces terse em1
     register: result

   - name: display result to terminal window
     debug:
        var: result.stdout_lines
```

Red Hat

AUTOMATION FOR EVERYONE: **PLAYBOOK RESULTS**



```
[student3@ansible network_setup]$ ansible-playbook junos-example.yml

PLAY [juniper] *****************************************************************************************************

TASK [show command for juniper] ************************************************************************************
ok: [rtr3]
ok: [rtr4]

TASK [display result to terminal window] ***************************************************************************
ok: [rtr3] => {
    "result.stdout_lines": [
        "Interface               Admin Link Proto    Local                 Remote",
        "em1                     up    up",
        "em1.0                   up    up   inet     10.0.0.4/8      ",
        "                                            128.0.0.1/2     ",
        "                                            128.0.0.4/2     ",
        "                                   inet6    fe80::5254:ff:fe12:bdfe/64",
        "                                            fec0::a:0:0:4/64",
        "                                   tnp      0x4"
    ]
}
ok: [rtr4] => {
    "result.stdout_lines": [
        "Interface               Admin Link Proto    Local                 Remote",
        "em1                     up    up",
        "em1.0                   up    up   inet     10.0.0.4/8      ",
        "                                            128.0.0.1/2     ",
        "                                            128.0.0.4/2     ",
        "                                   inet6    fe80::5254:ff:fe12:bdfe/64",
        "                                            fec0::a:0:0:4/64",
        "                                   tnp      0x4"
    ]
}

PLAY RECAP *********************************************************************************************************
rtr3                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rtr4                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0

[student3@ansible network_setup]$
```

A Sample Ansible Playbook

```
---
- name: deploy vlans
  hosts: cisco
  gather_facts: no

  tasks:
   - name: ensure vlans exist
     nxos_vlan:
       vlan_id: 100
       admin_state: up
       name: WEB
```

- Playbook is a list of plays.
- Each play is a list of tasks.
- Tasks invoke modules.
- A playbook can contain more than one play.

Red Hat
Ansible
Automation

USE CASE:

Cloud automation

Red Hat

PLAYBOOK EXAMPLE: **RED HAT OPENSTACK**

```
---
- name: openstack playbook
  hosts: localhost
  connection: local

  tasks:
    - name: launch an instance
     os_server:
          name: vm1
          cloud: mordred
          region_name: ams01
          image: Red Hat Enterprise Linux 7.4
          flavor_ram: 4096
```

Tower Introduction

Red Hat Ansible Tower

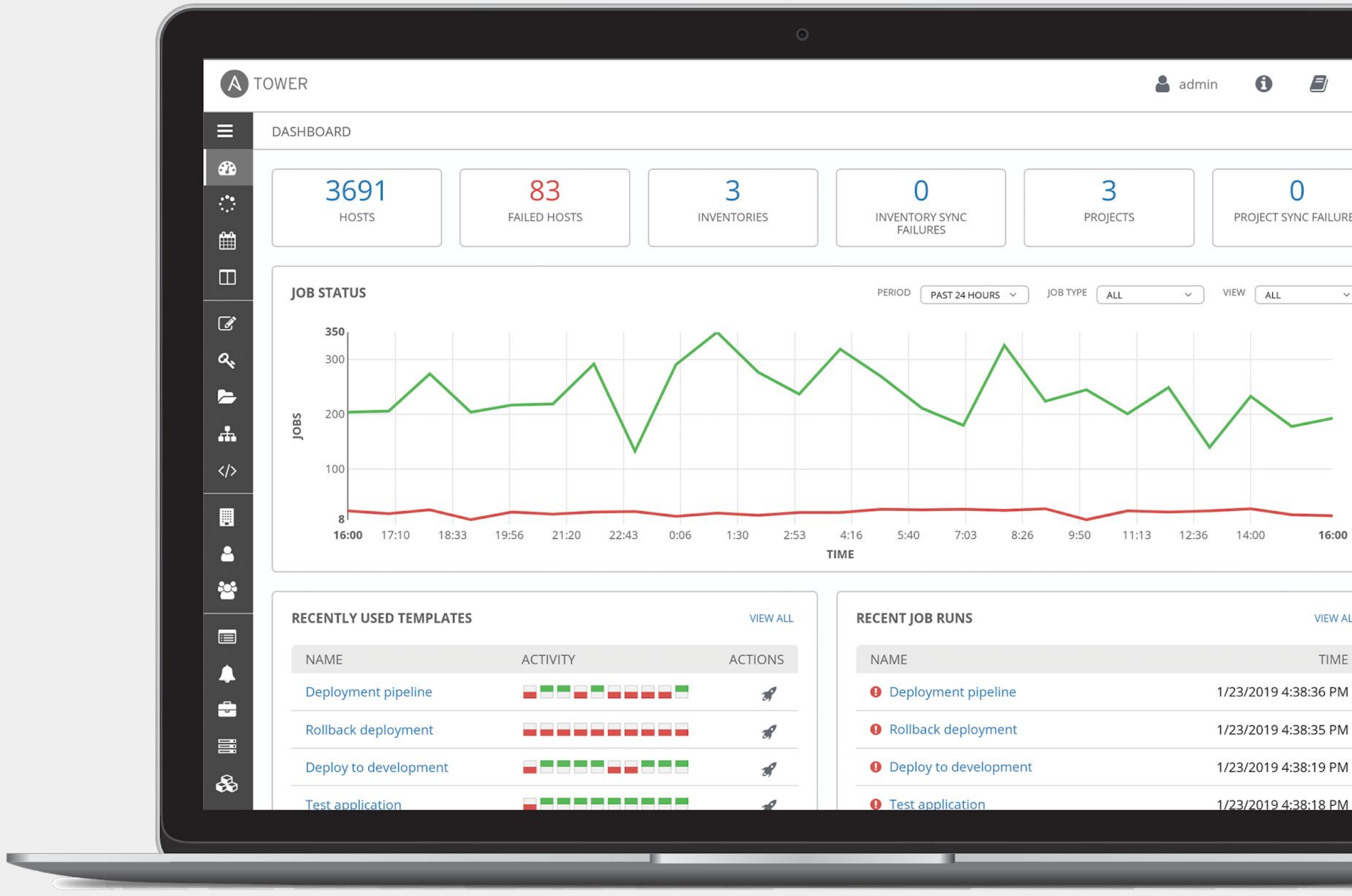
Topics Covered:

- What is Ansible Tower?
- Job Templates
    - Inventory
    - Credentials
    - Projects

What is Ansible Tower?

Ansible Tower is a UI and RESTful API allowing you to scale IT automation, manage complex deployments and speed productivity.

- Role-based access control
- Deploy entire applications with push-button deployment access
- All automations are centrally logged
- Powerful workflows match your IT processes



3691	83	3	0	3	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURE

PERIOD PAST 24 HOURS JOB TYPE ALL VIEW ALL

RECENTLY USED TEMPLATES

NAME	ACTIVITY	ACTIONS
Deployment pipeline	🟥🟩🟩🟥🟩🟥🟥🟥🟥🟩	
Rollback deployment	🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥	
Deploy to development	🟥🟩🟩🟩🟩🟥🟥🟩🟩🟩	
Test application	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟩	

RECENT JOB RUNS

NAME	TIME
Deployment pipeline	1/23/2019 4:38:36 PM
Rollback deployment	1/23/2019 4:38:35 PM
Deploy to development	1/23/2019 4:38:19 PM
Test application	1/23/2019 4:38:18 PM

Red Hat

Red Hat Ansible Tower

RBAC

Allow restricting playbook access to authorized users. One team can use playbooks in check mode (read-only) while others have full administrative abilities.

Push button

An intuitive user interface experience makes it easy for novice users to execute playbooks you allow them access to.

RESTful API

With an API first mentality every feature and function of Tower can be API driven. Allow seamless integration with other tools like ServiceNow and Infoblox.

Workflows

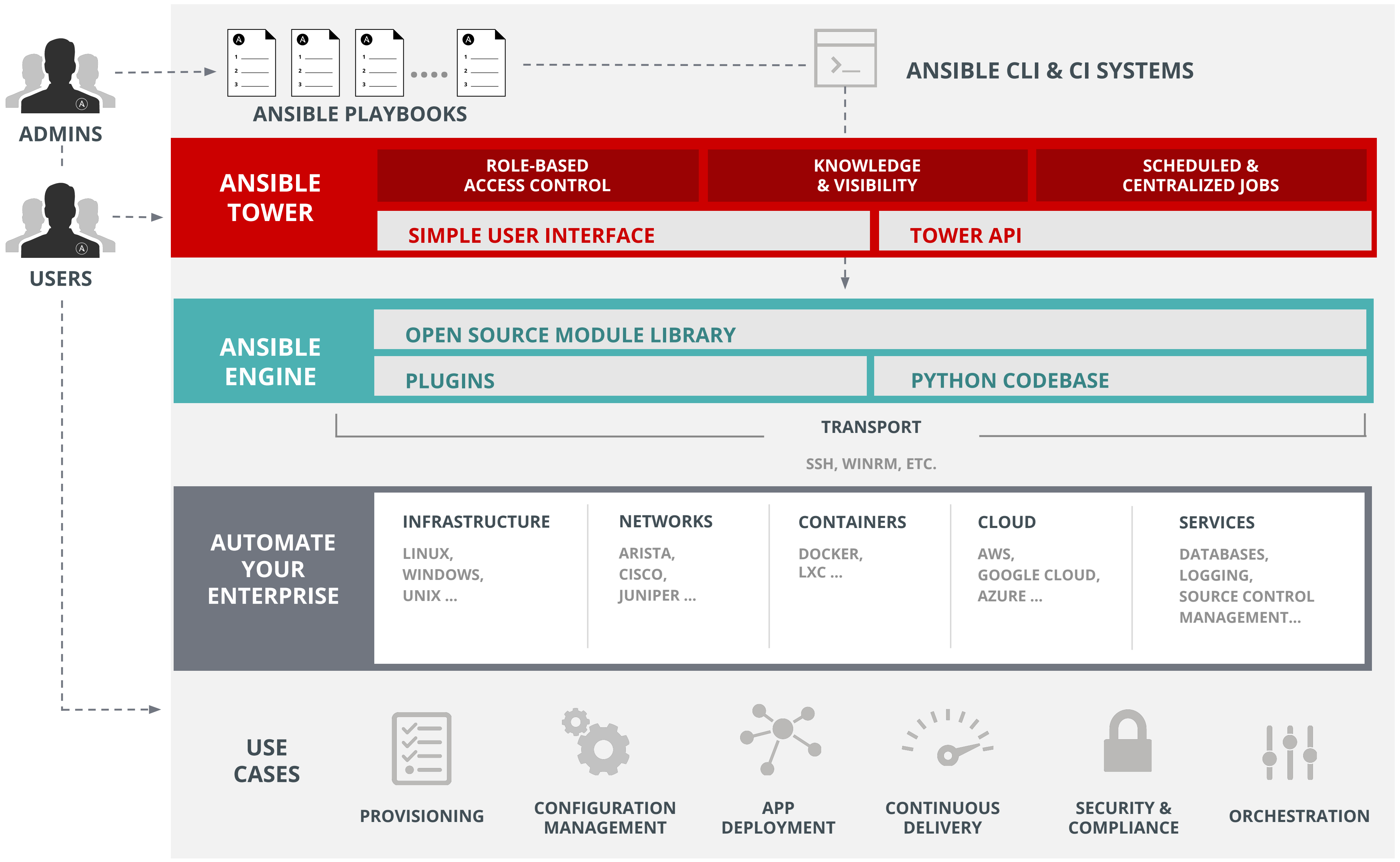
Ansible Tower's multi-playbook workflows chain any number of playbooks, regardless of whether they use different inventories, run as different users, run at once or utilize different credentials.

Enterprise integrations

Integrate with enterprise authentication like TACACS+, RADIUS, Azure AD. Setup token authentication with OAuth 2. Setup notifications with PagerDuty, Slack and Twilio.

Centralized logging

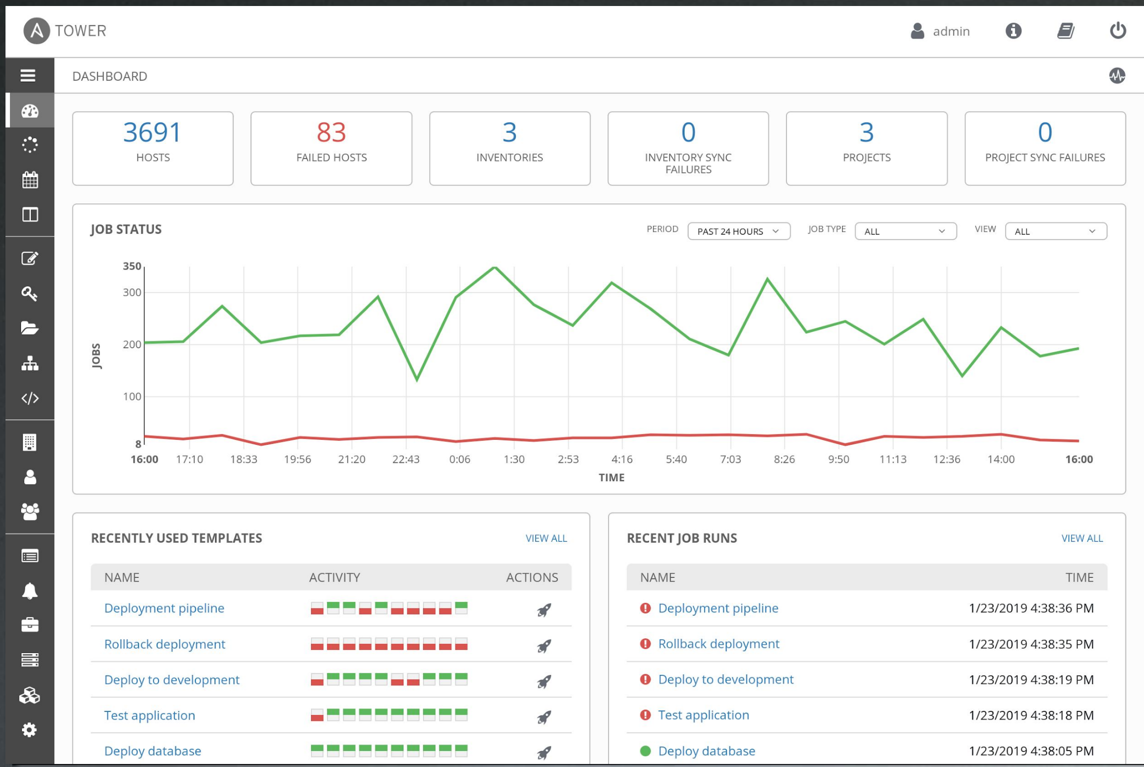
All automation activity is securely logged. Who ran it, how they customized it, what it did, where it happened - all securely stored and viewable later, or exported through Ansible Tower's API.


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 ENGINE
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

Red Hat Ansible Tower

FEATURE OVERVIEW:

Control



3691	83	3	0	3	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST 24 HOURS · JOB TYPE: ALL · VIEW: ALL

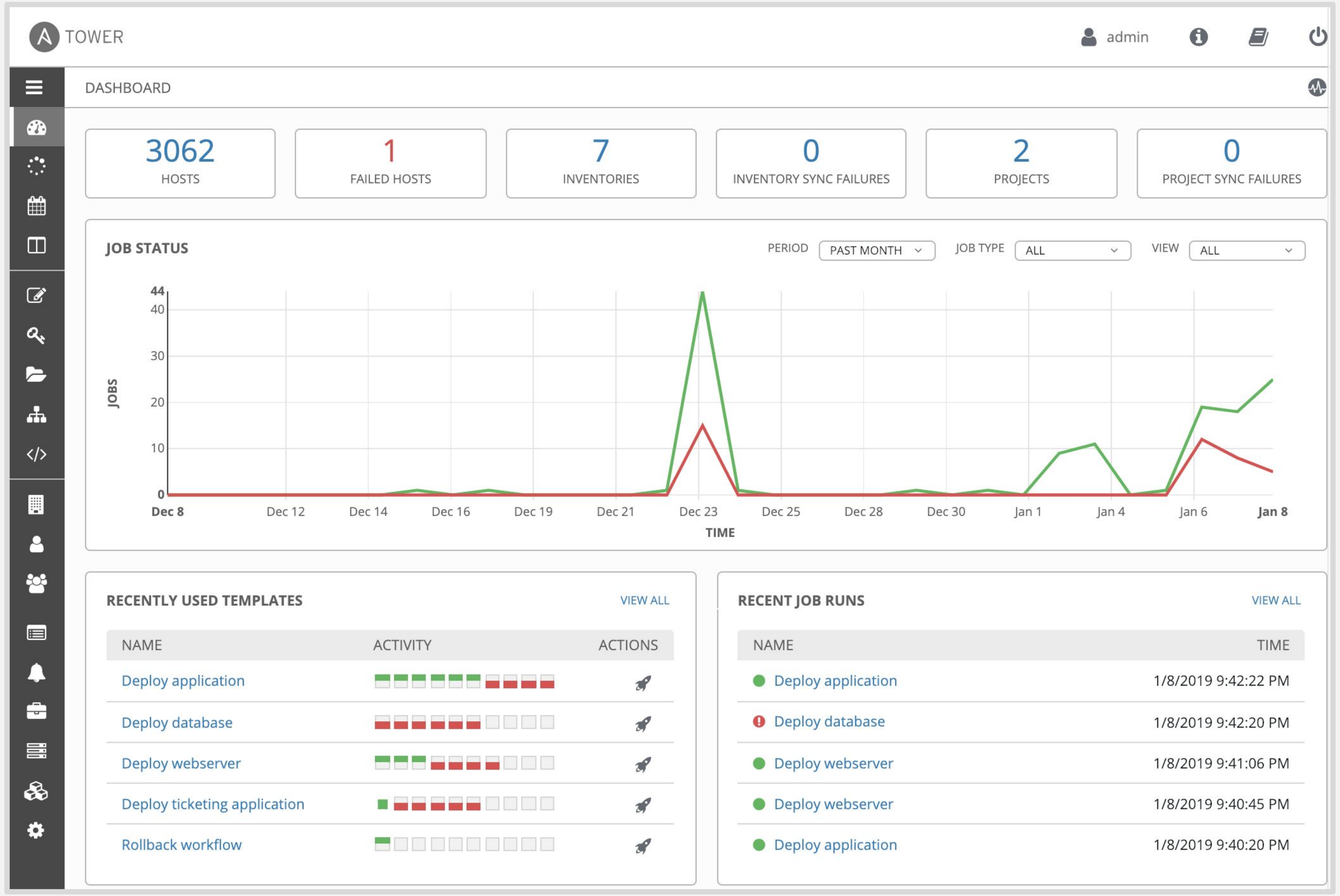
RECENTLY USED TEMPLATES · VIEW ALL

NAME	ACTIVITY	ACTIONS
Deployment pipeline	🟥🟩🟩🟥🟩🟥🟥🟥🟥🟩	
Rollback deployment	🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥	
Deploy to development	🟥🟩🟩🟩🟩🟥🟥🟩🟩🟩	
Test application	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟩	
Deploy database	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	

RECENT JOB RUNS · VIEW ALL

NAME	TIME
● Deployment pipeline	1/23/2019 4:38:36 PM
● Rollback deployment	1/23/2019 4:38:35 PM
● Deploy to development	1/23/2019 4:38:19 PM
● Test application	1/23/2019 4:38:18 PM
● Deploy database	1/23/2019 4:38:05 PM

ANSIBLE TOWER FEATURES: YOUR ANSIBLE DASHBOARD



3062	1	7	0	2	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH — JOB TYPE: ALL — VIEW: ALL

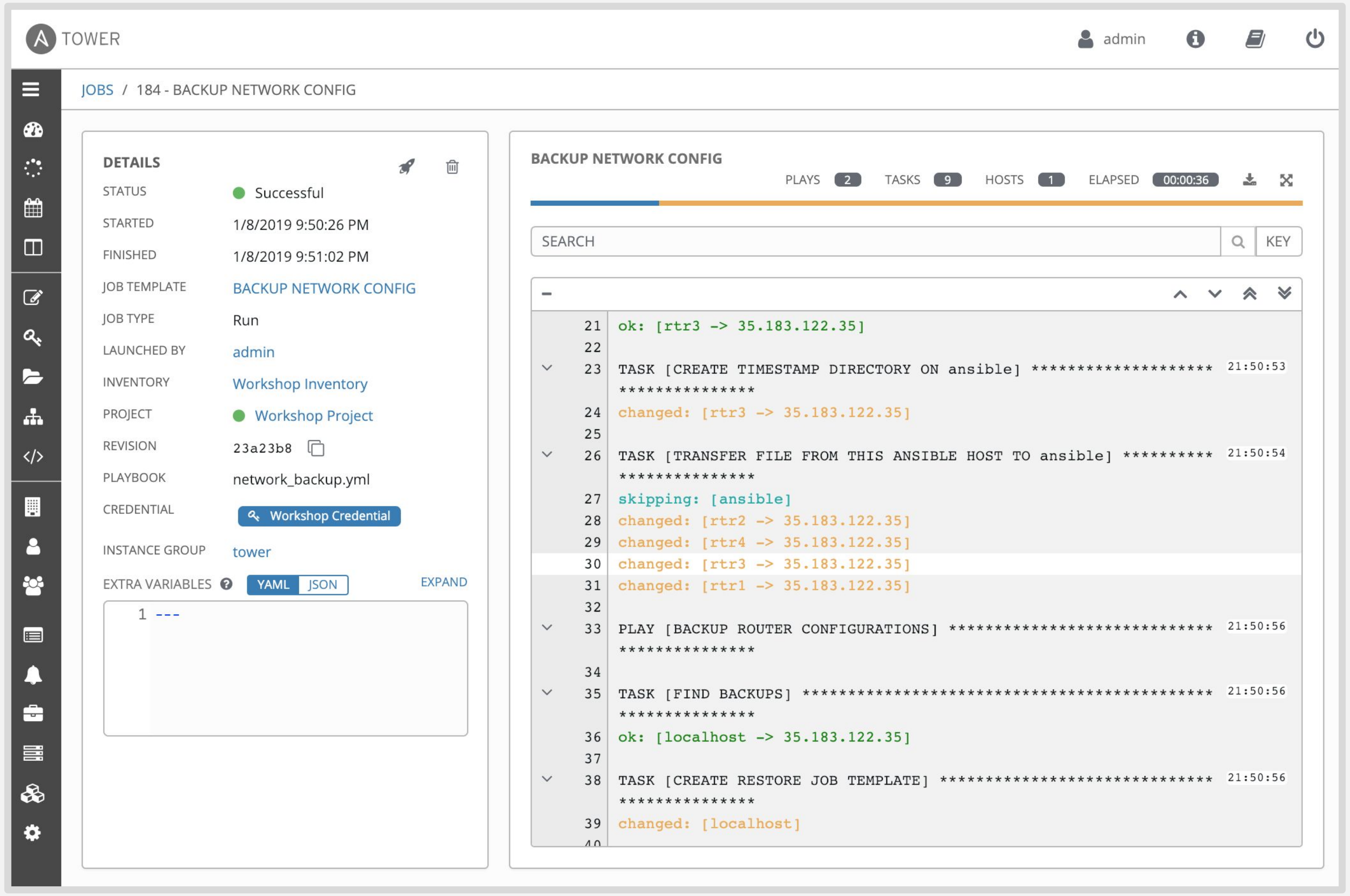
RECENTLY USED TEMPLATES — VIEW ALL

NAME	ACTIVITY	ACTIONS
Deploy application	🟩🟩🟩🟩🟩🟩🟥🟥🟥🟥	
Deploy database	🟥🟥🟥🟥🟥🟥⬜⬜⬜⬜	
Deploy webserver	🟩🟩🟩🟥🟥🟥🟥⬜⬜⬜	
Deploy ticketing application	🟩🟥🟥🟥🟥🟥⬜⬜⬜⬜	
Rollback workflow	🟩⬜⬜⬜⬜⬜⬜⬜⬜⬜	

RECENT JOB RUNS — VIEW ALL

NAME	TIME
● Deploy application	1/8/2019 9:42:22 PM
! Deploy database	1/8/2019 9:42:20 PM
● Deploy webserver	1/8/2019 9:41:06 PM
● Deploy webserver	1/8/2019 9:40:45 PM
● Deploy application	1/8/2019 9:40:20 PM

ANSIBLE TOWER FEATURES: JOB STATUS UPDATE



JOBS / 184 - BACKUP NETWORK CONFIG

DETAILS

STATUS	● Successful
STARTED	1/8/2019 9:50:26 PM
FINISHED	1/8/2019 9:51:02 PM
JOB TEMPLATE	BACKUP NETWORK CONFIG
JOB TYPE	Run
LAUNCHED BY	admin
INVENTORY	Workshop Inventory
PROJECT	● Workshop Project
REVISION	23a23b8
PLAYBOOK	network_backup.yml
CREDENTIAL	Workshop Credential
INSTANCE GROUP	tower
EXTRA VARIABLES	YAML JSON

EXPAND

```
---
```

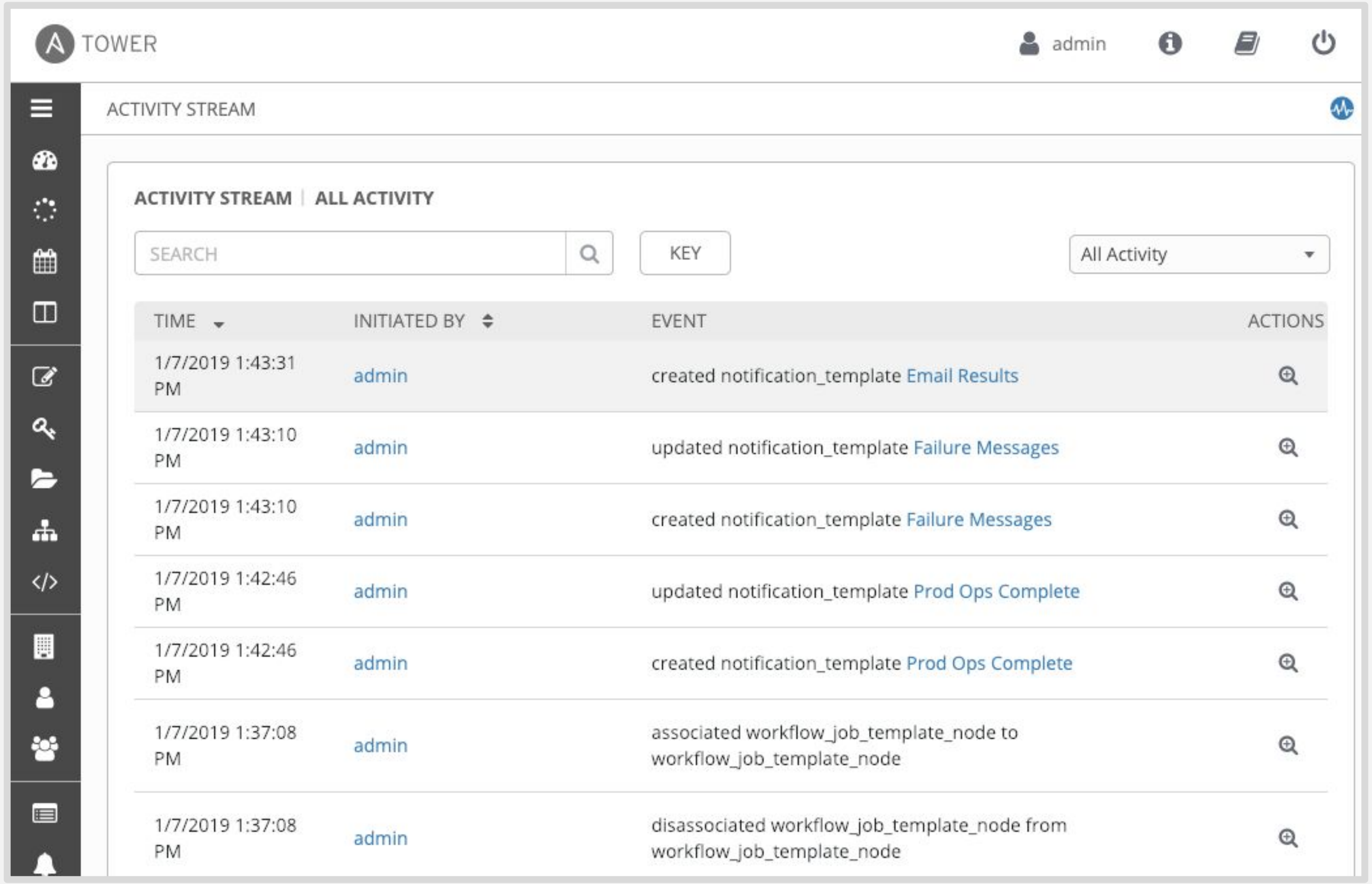
BACKUP NETWORK CONFIG

PLAYS 2 TASKS 9 HOSTS 1 ELAPSED 00:00:36

SEARCH KEY

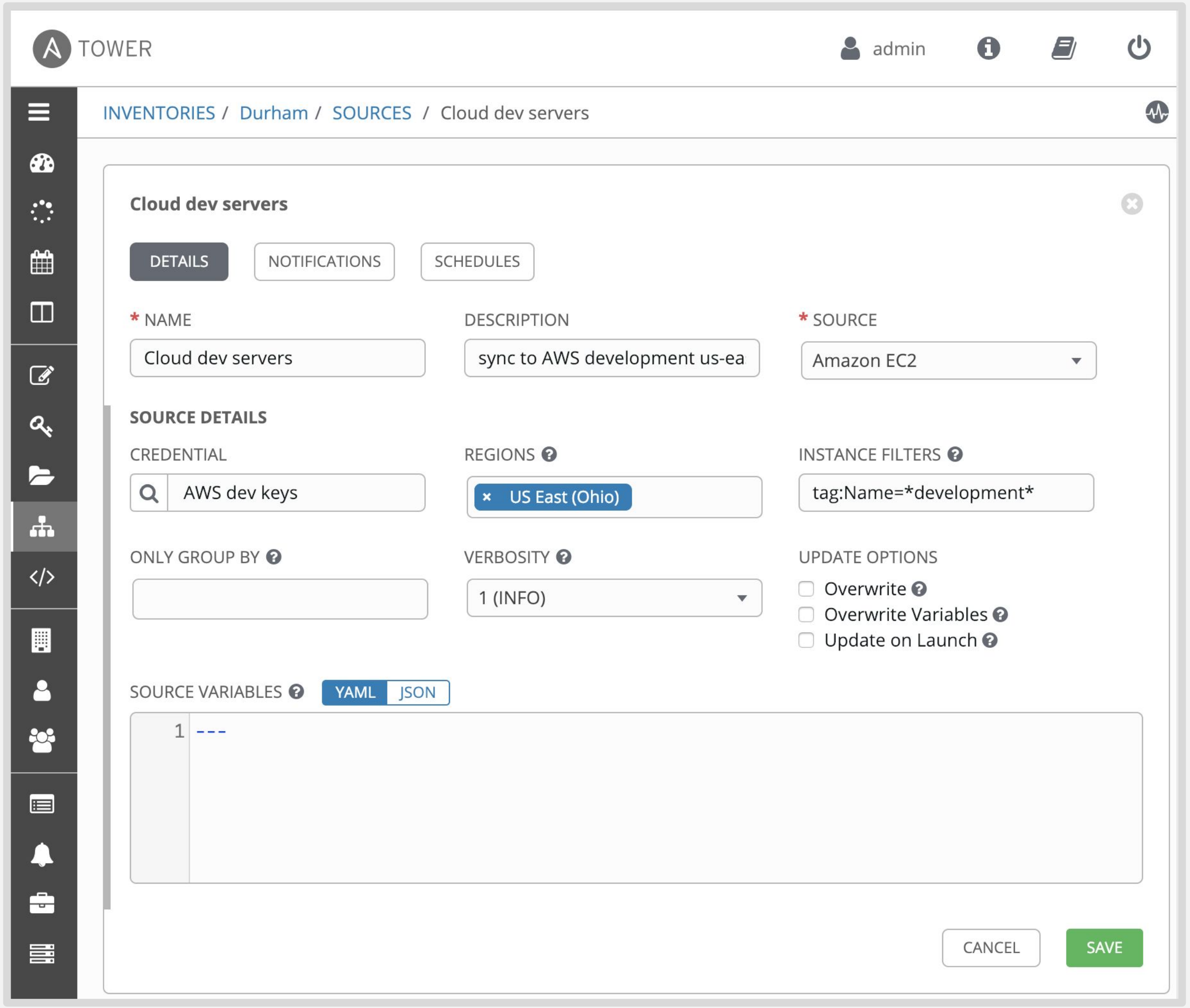
```
ok: [rtr3 -> 35.183.122.35]

TASK [CREATE TIMESTAMP DIRECTORY ON ansible] *******************************************  21:50:53
changed: [rtr3 -> 35.183.122.35]

TASK [TRANSFER FILE FROM THIS ANSIBLE HOST TO ansible] ************************  21:50:54
skipping: [ansible]
changed: [rtr2 -> 35.183.122.35]
changed: [rtr4 -> 35.183.122.35]
changed: [rtr3 -> 35.183.122.35]
changed: [rtr1 -> 35.183.122.35]

PLAY [BACKUP ROUTER CONFIGURATIONS] ********************************************  21:50:56

TASK [FIND BACKUPS] ************************************************************  21:50:56
ok: [localhost -> 35.183.122.35]

TASK [CREATE RESTORE JOB TEMPLATE] *********************************************  21:50:56
changed: [localhost]
```

ANSIBLE TOWER FEATURES: ACTIVITY STREAM



TIME	INITIATED BY	EVENT	ACTIONS
1/7/2019 1:43:31 PM	admin	created notification_template Email Results	
1/7/2019 1:43:10 PM	admin	updated notification_template Failure Messages	
1/7/2019 1:43:10 PM	admin	created notification_template Failure Messages	
1/7/2019 1:42:46 PM	admin	updated notification_template Prod Ops Complete	
1/7/2019 1:42:46 PM	admin	created notification_template Prod Ops Complete	
1/7/2019 1:37:08 PM	admin	associated workflow_job_template_node to workflow_job_template_node	
1/7/2019 1:37:08 PM	admin	disassociated workflow_job_template_node from workflow_job_template_node	

ANSIBLE TOWER FEATURES: MANAGE AND TRACK YOUR INVENTORY



INVENTORIES / Durham / SOURCES / Cloud dev servers

Cloud dev servers

* NAME	DESCRIPTION	* SOURCE
Cloud dev servers	sync to AWS development us-ea	Amazon EC2

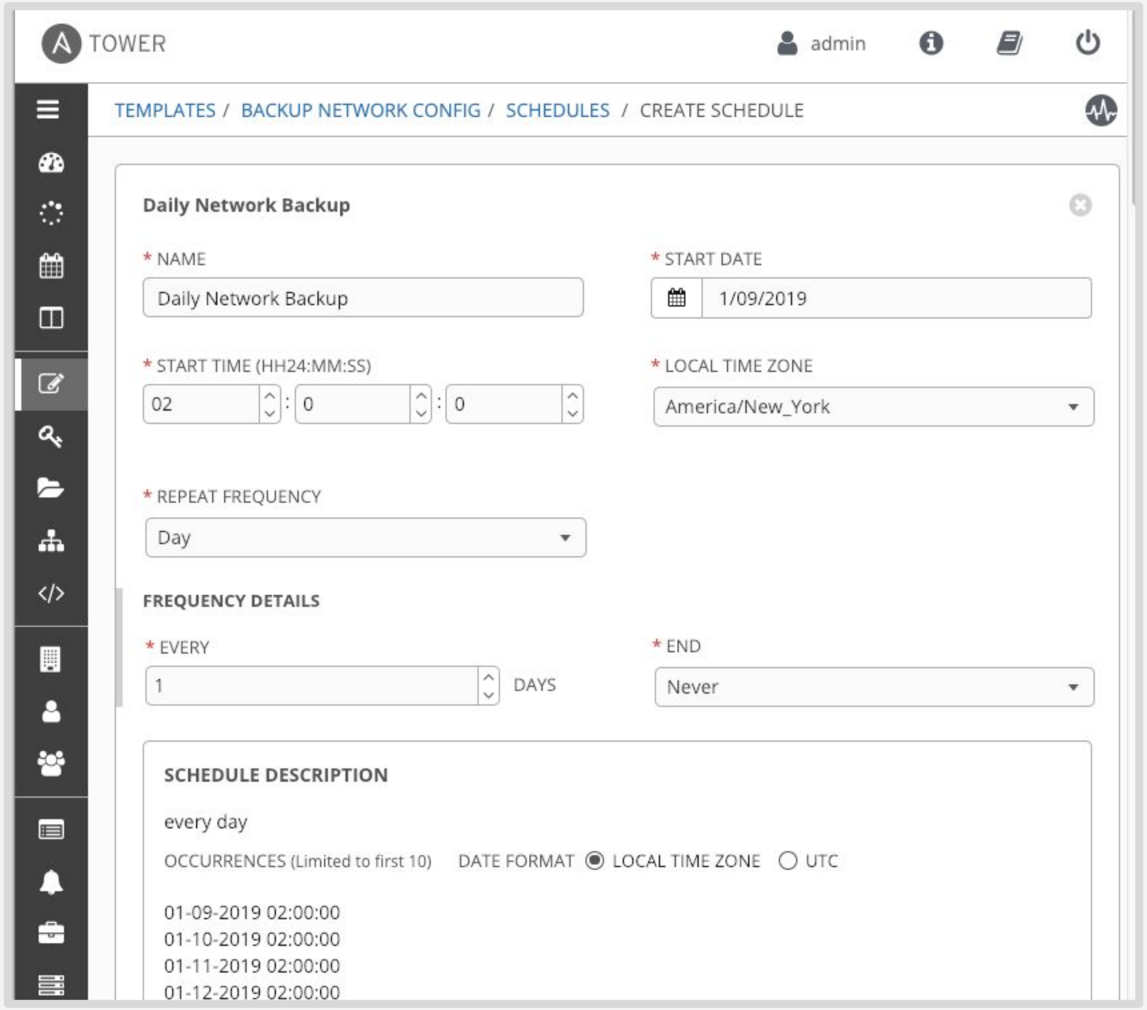
SOURCE DETAILS

CREDENTIAL	REGIONS	INSTANCE FILTERS
AWS dev keys	US East (Ohio)	tag:Name=*development*
ONLY GROUP BY	VERBOSITY	UPDATE OPTIONS
	1 (INFO)	☐ Overwrite ☐ Overwrite Variables ☐ Update on Launch

SOURCE VARIABLES [✓ YAML] JSON

```
---
```

ANSIBLE TOWER FEATURES: SCHEDULE JOBS


TOWER
admin
TEMPLATES / BACKUP NETWORK CONFIG / SCHEDULES / CREATE SCHEDULE

Daily Network Backup

* NAME	Daily Network Backup
* START DATE	1/09/2019
* START TIME (HH24:MM:SS)	02 : 0 : 0
* LOCAL TIME ZONE	America/New_York
* REPEAT FREQUENCY	Day

FREQUENCY DETAILS

* EVERY	1 DAYS
* END	Never

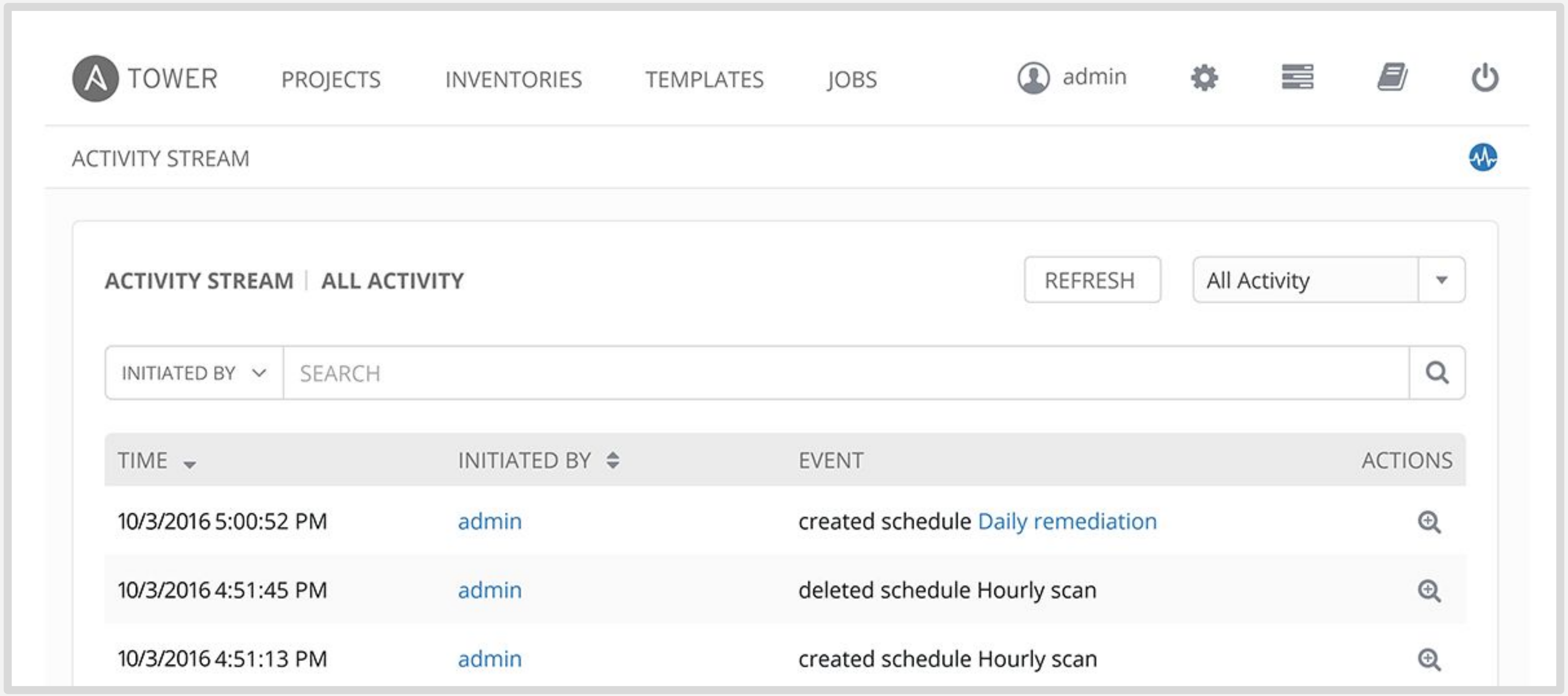
SCHEDULE DESCRIPTION

every day

OCCURRENCES (Limited to first 10) DATE FORMAT [✓ LOCAL TIME ZONE] UTC

01-09-2019 02:00:00
01-10-2019 02:00:00
01-11-2019 02:00:00
01-12-2019 02:00:00

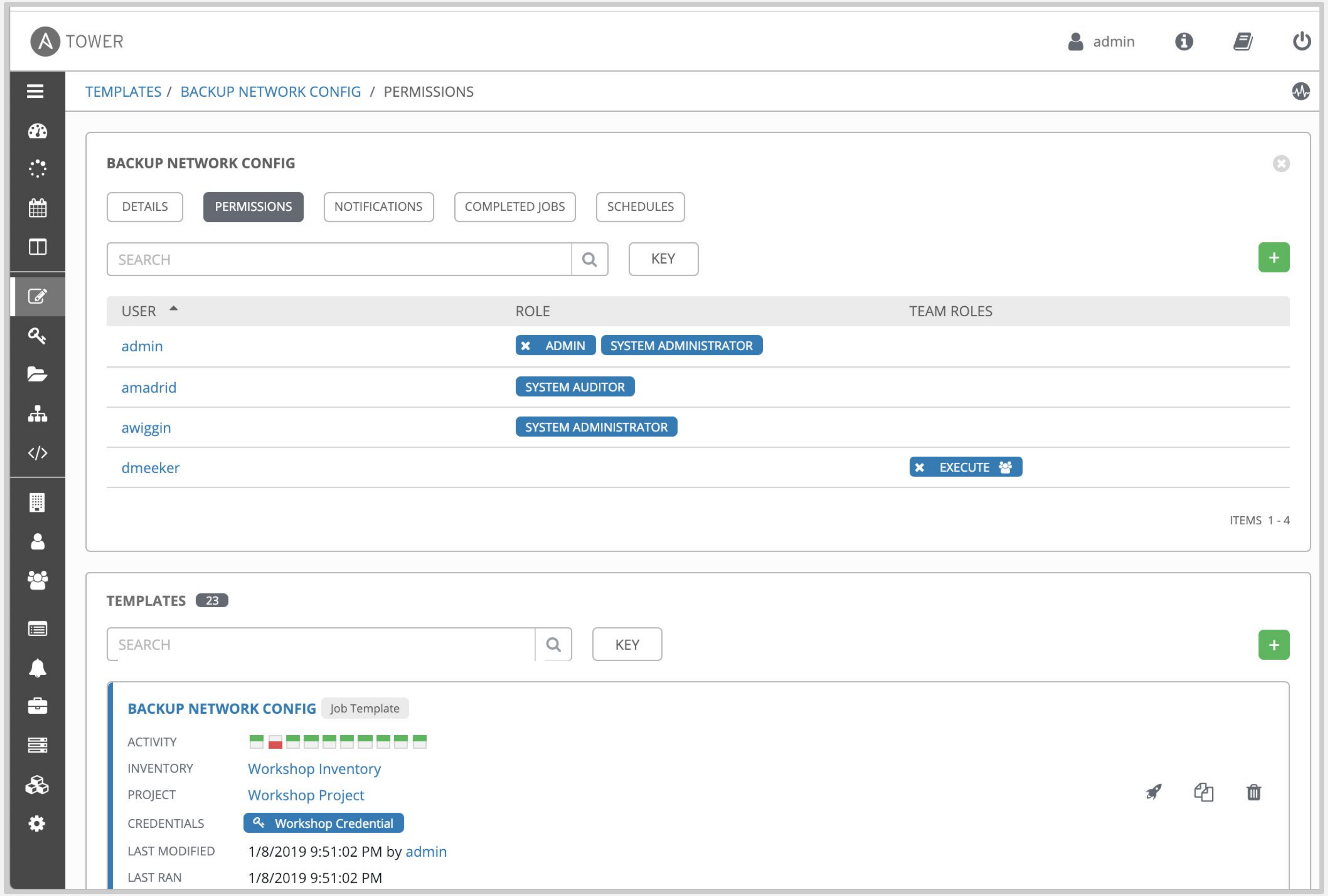
ANSIBLE TOWER FEATURES: EXTERNAL LOGGING



ACTIVITY STREAM | ALL ACTIVITY

TIME	INITIATED BY	EVENT	ACTIONS
10/3/2016 5:00:52 PM	admin	created schedule Daily remediation	
10/3/2016 4:51:45 PM	admin	deleted schedule Hourly scan	
10/3/2016 4:51:13 PM	admin	created schedule Hourly scan	

ANSIBLE TOWER FEATURES: ROLE BASED ACCESS CONTROL



BACKUP NETWORK CONFIG

USER	ROLE	TEAM ROLES
admin	ADMIN, SYSTEM ADMINISTRATOR	
amadrid	SYSTEM AUDITOR	
awiggin	SYSTEM ADMINISTRATOR	
dmeeker		EXECUTE

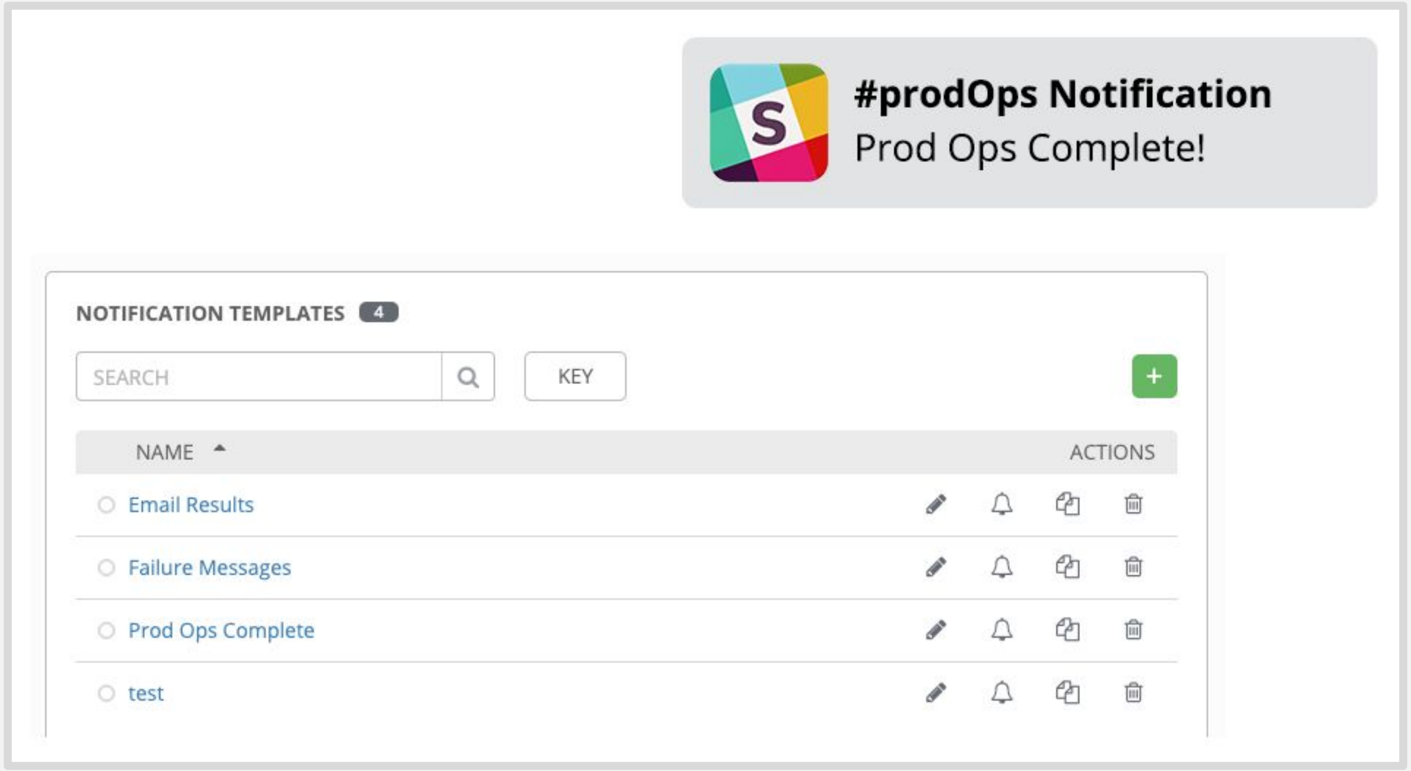
ITEMS 1 - 4

TEMPLATES 23

BACKUP NETWORK CONFIG Job Template

ACTIVITY	🟩🟥🟩🟩🟩🟩🟩🟩🟩🟩
INVENTORY	Workshop Inventory
PROJECT	Workshop Project
CREDENTIALS	Workshop Credential
LAST MODIFIED	1/8/2019 9:51:02 PM by admin
LAST RAN	1/8/2019 9:51:02 PM

ANSIBLE TOWER FEATURES: INTEGRATED NOTIFICATIONS



#prodOps Notification
Prod Ops Complete!

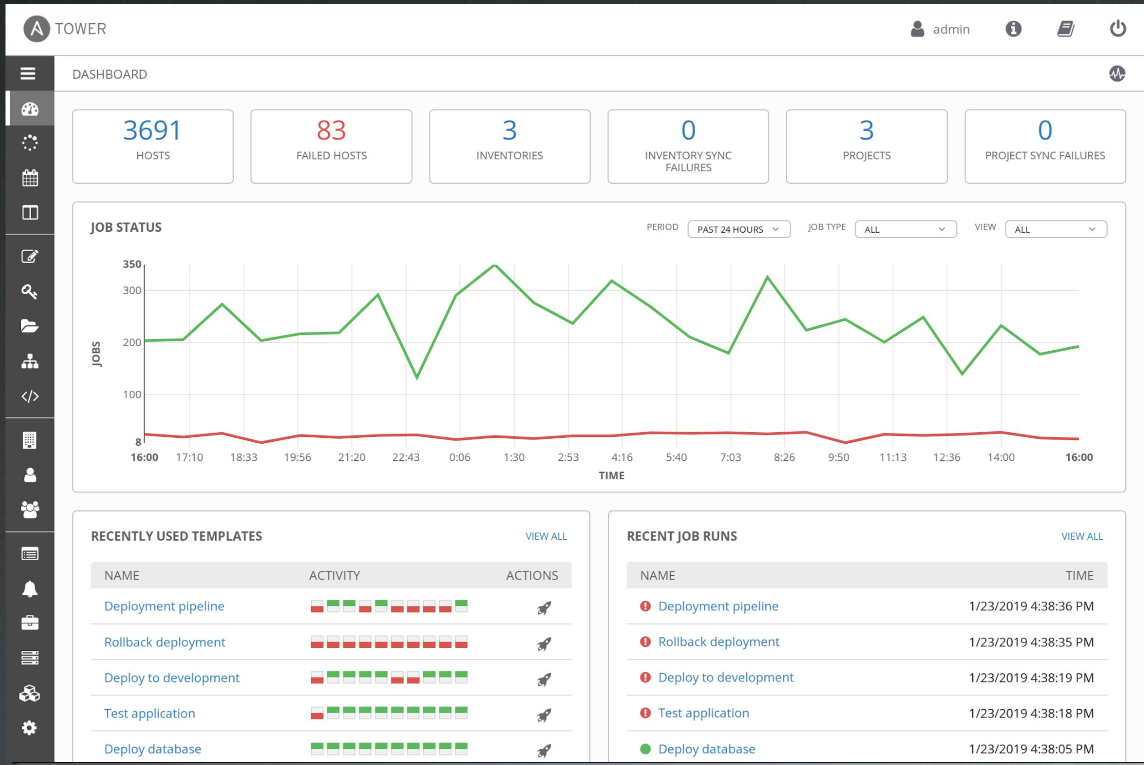
NOTIFICATION TEMPLATES 4

NAME	ACTIONS
Email Results	
Failure Messages	
Prod Ops Complete	
test	

Red Hat
Ansible
Tower

FEATURE OVERVIEW:

Delegation



3691	83	3	0	3	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

RECENTLY USED TEMPLATES — VIEW ALL

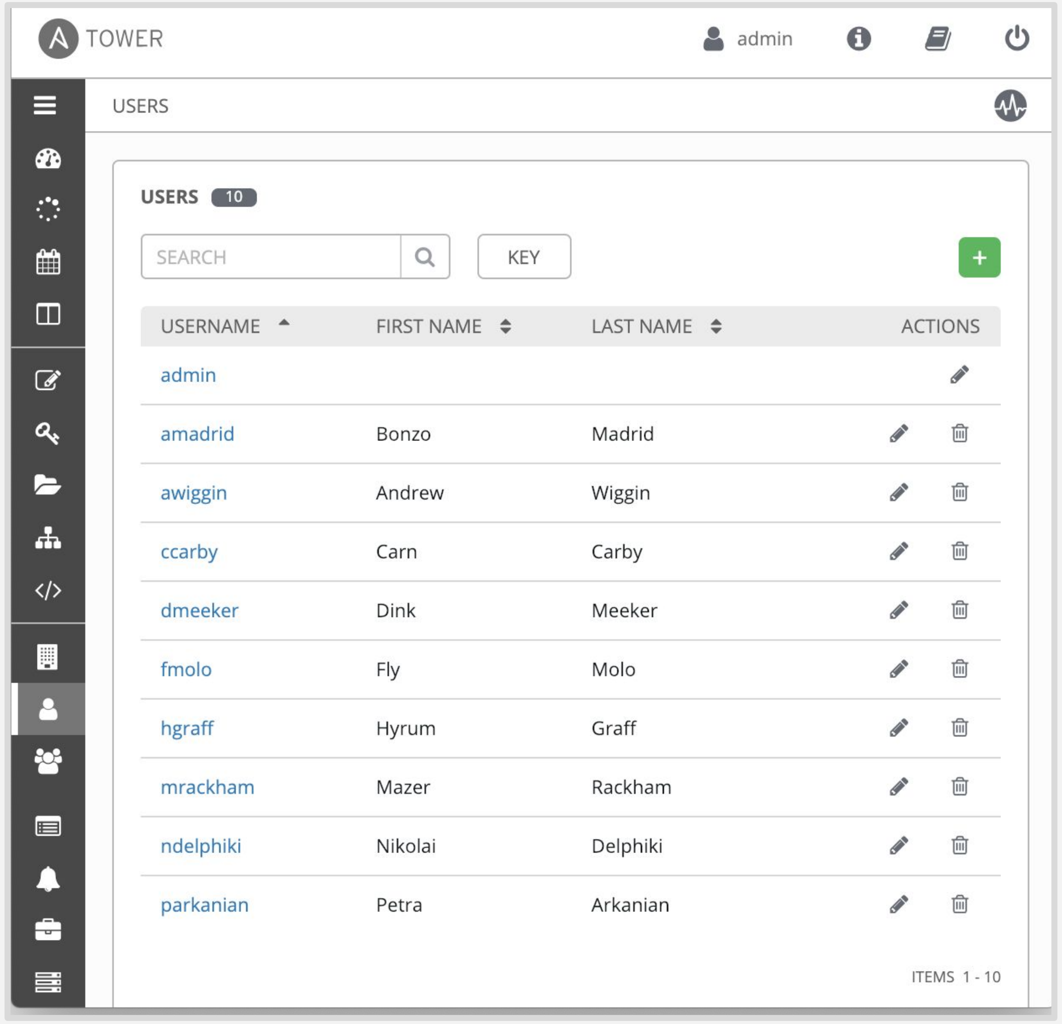
NAME	ACTIVITY	ACTIONS
Deployment pipeline	🟥🟩🟩🟥🟩🟥🟥🟥🟥🟩	
Rollback deployment	🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥	
Deploy to development	🟥🟩🟩🟩🟩🟥🟥🟩🟩🟩	
Test application	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟩	
Deploy database	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	

RECENT JOB RUNS — VIEW ALL

NAME	TIME
Deployment pipeline	1/23/2019 4:38:36 PM
Rollback deployment	1/23/2019 4:38:35 PM
Deploy to development	1/23/2019 4:38:19 PM
Test application	1/23/2019 4:38:18 PM
Deploy database	1/23/2019 4:38:05 PM

ANSIBLE TOWER FEATURES: ROLE BASED ACCESS CONTROL

USERS

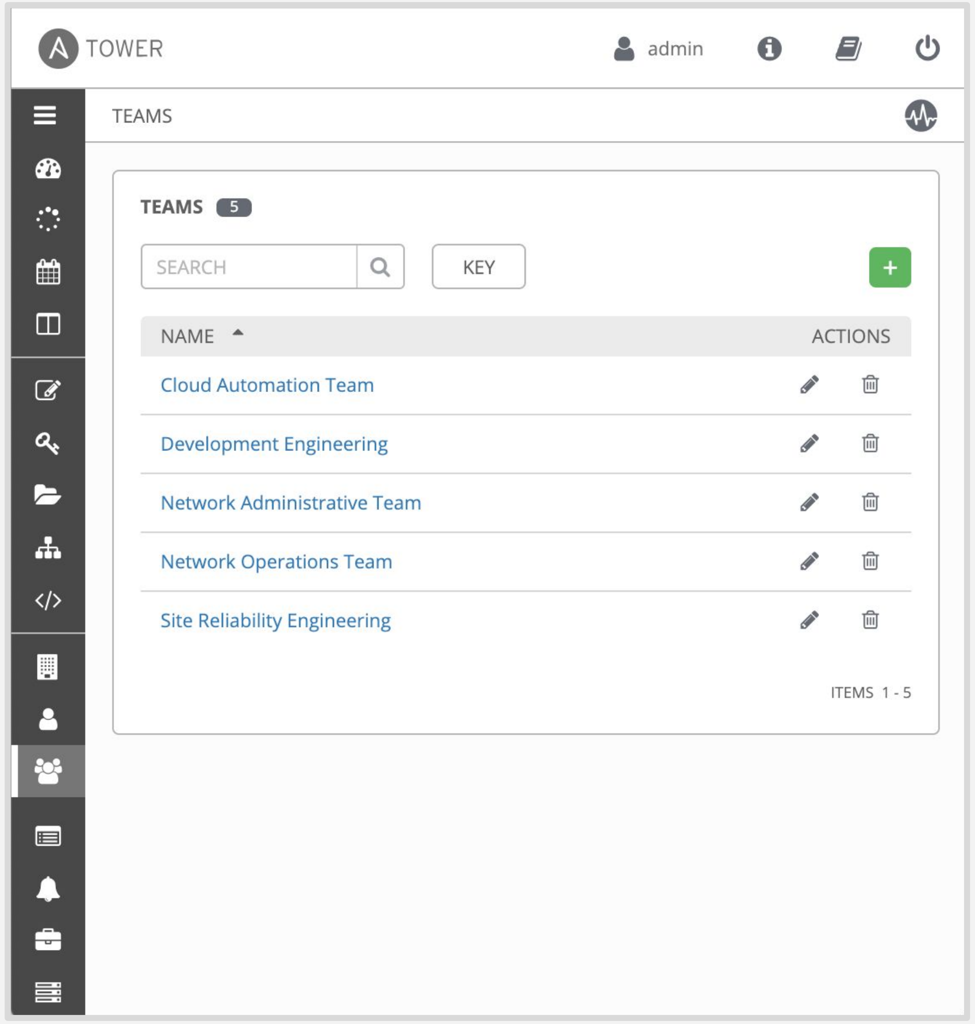


USERS 10

USERNAME	FIRST NAME	LAST NAME	ACTIONS
admin			
amadrid	Bonzo	Madrid	
awiggin	Andrew	Wiggin	
ccarby	Carn	Carby	
dmeeker	Dink	Meeker	
fmolo	Fly	Molo	
hgraff	Hyrum	Graff	
mrackham	Mazer	Rackham	
ndelphiki	Nikolai	Delphiki	
parkanian	Petra	Arkanian	

ITEMS 1 - 10

TEAMS

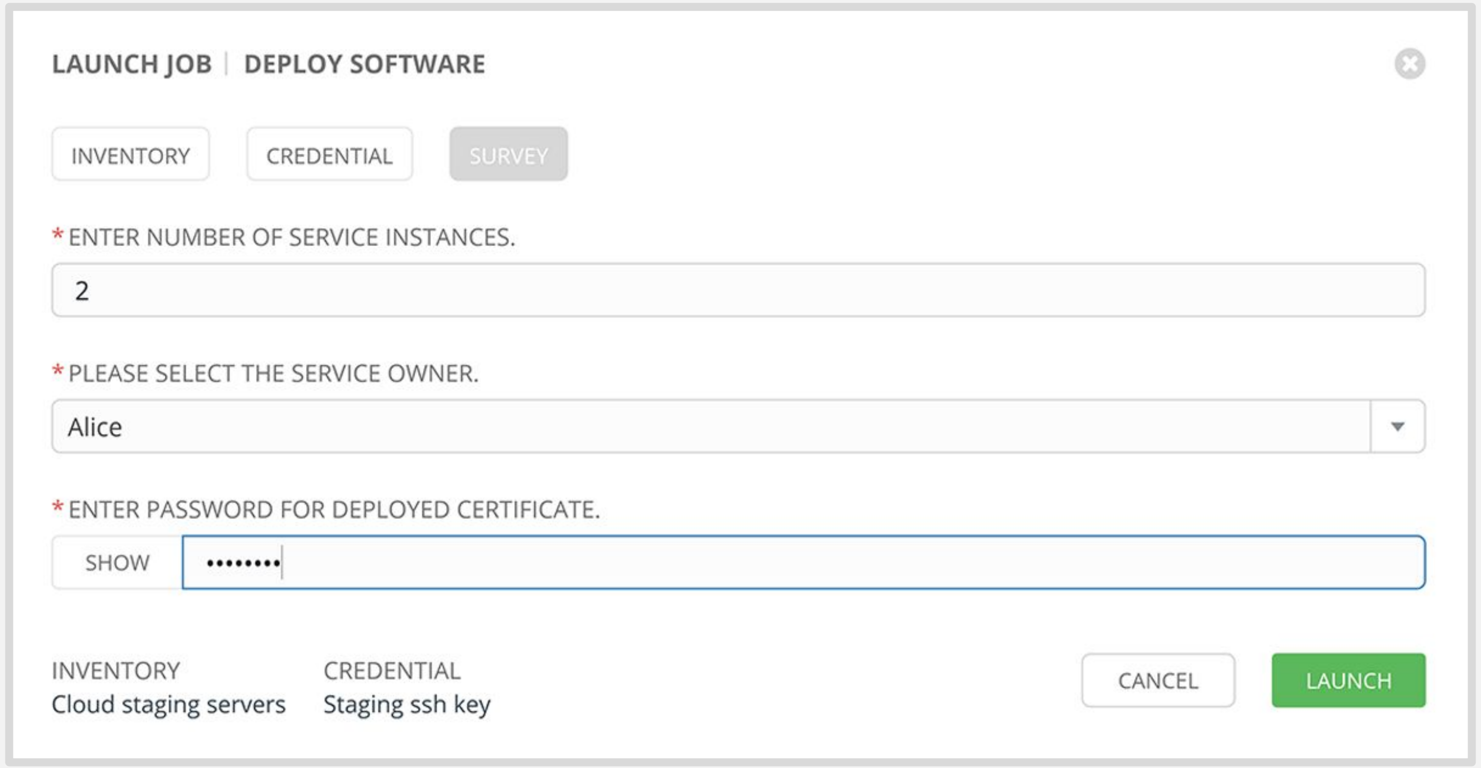


TEAMS 5

NAME	ACTIONS
Cloud Automation Team	
Development Engineering	
Network Administrative Team	
Network Operations Team	
Site Reliability Engineering	

ITEMS 1 - 5

ANSIBLE TOWER FEATURES: SELF-SERVICE I.T.


LAUNCH JOB | DEPLOY SOFTWARE

INVENTORY CREDENTIAL SURVEY

* ENTER NUMBER OF SERVICE INSTANCES.

2

* PLEASE SELECT THE SERVICE OWNER.

Alice

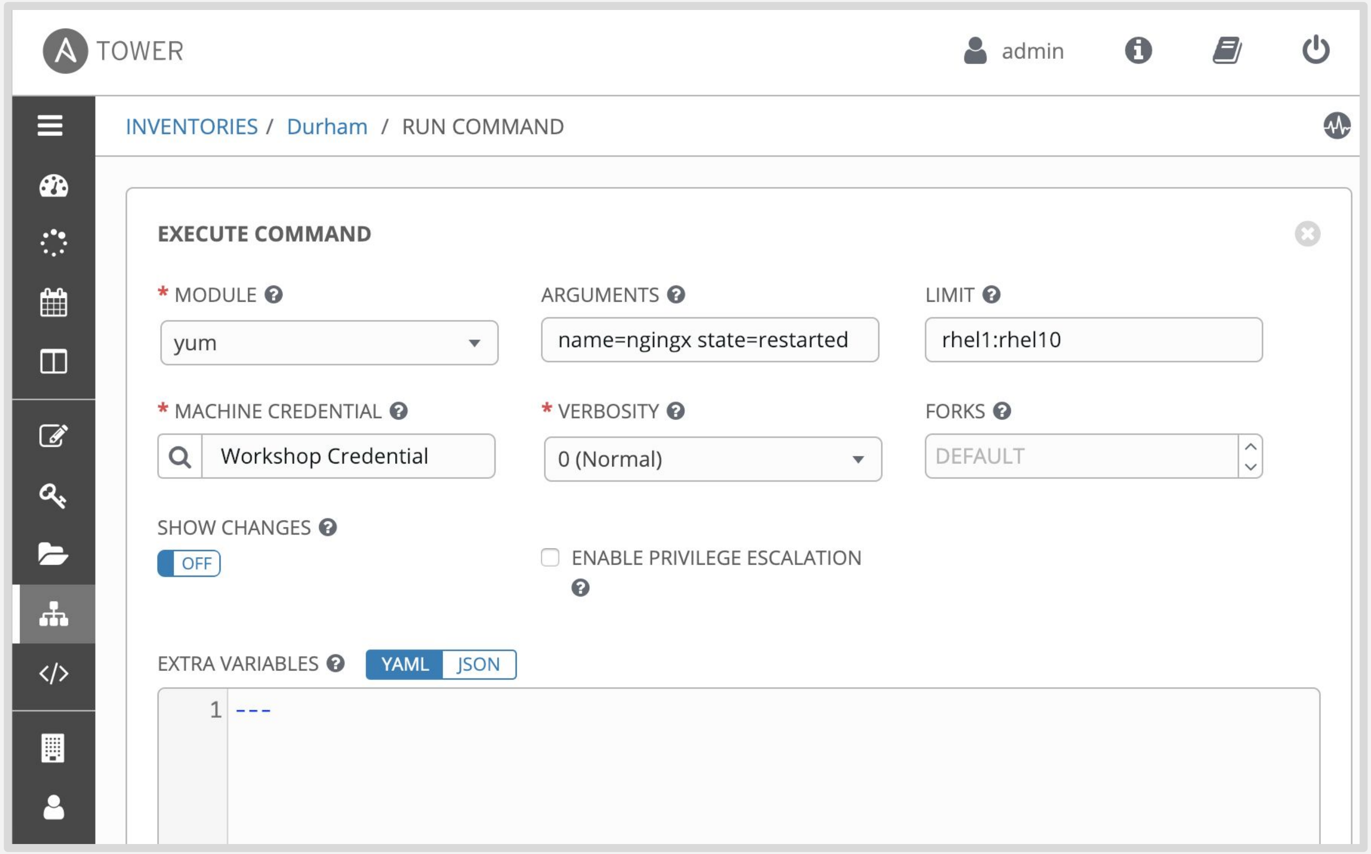
* ENTER PASSWORD FOR DEPLOYED CERTIFICATE.

SHOW •••••••••

INVENTORY	CREDENTIAL
Cloud staging servers	Staging ssh key

CANCEL LAUNCH

ANSIBLE TOWER FEATURES: REMOTE COMMAND EXECUTION



EXECUTE COMMAND

* MODULE	ARGUMENTS	LIMIT
yum	name=ngingx state=restarted	rhel1:rhel10
* MACHINE CREDENTIAL	* VERBOSITY	FORKS
Workshop Credential	0 (Normal)	DEFAULT

SHOW CHANGES: OFF

☐ ENABLE PRIVILEGE ESCALATION

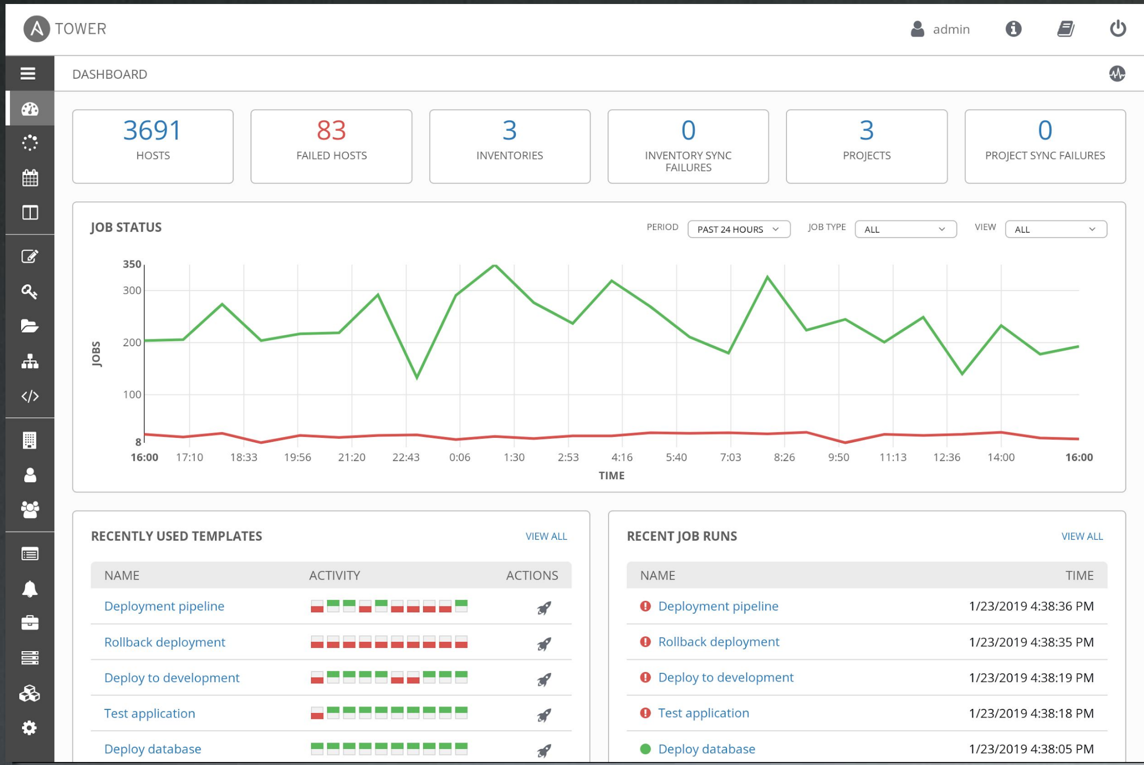
EXTRA VARIABLES [✓ YAML] JSON

```
---
```

Red Hat Ansible Tower

FEATURE OVERVIEW:

Scale



3691	83	3	0	3	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

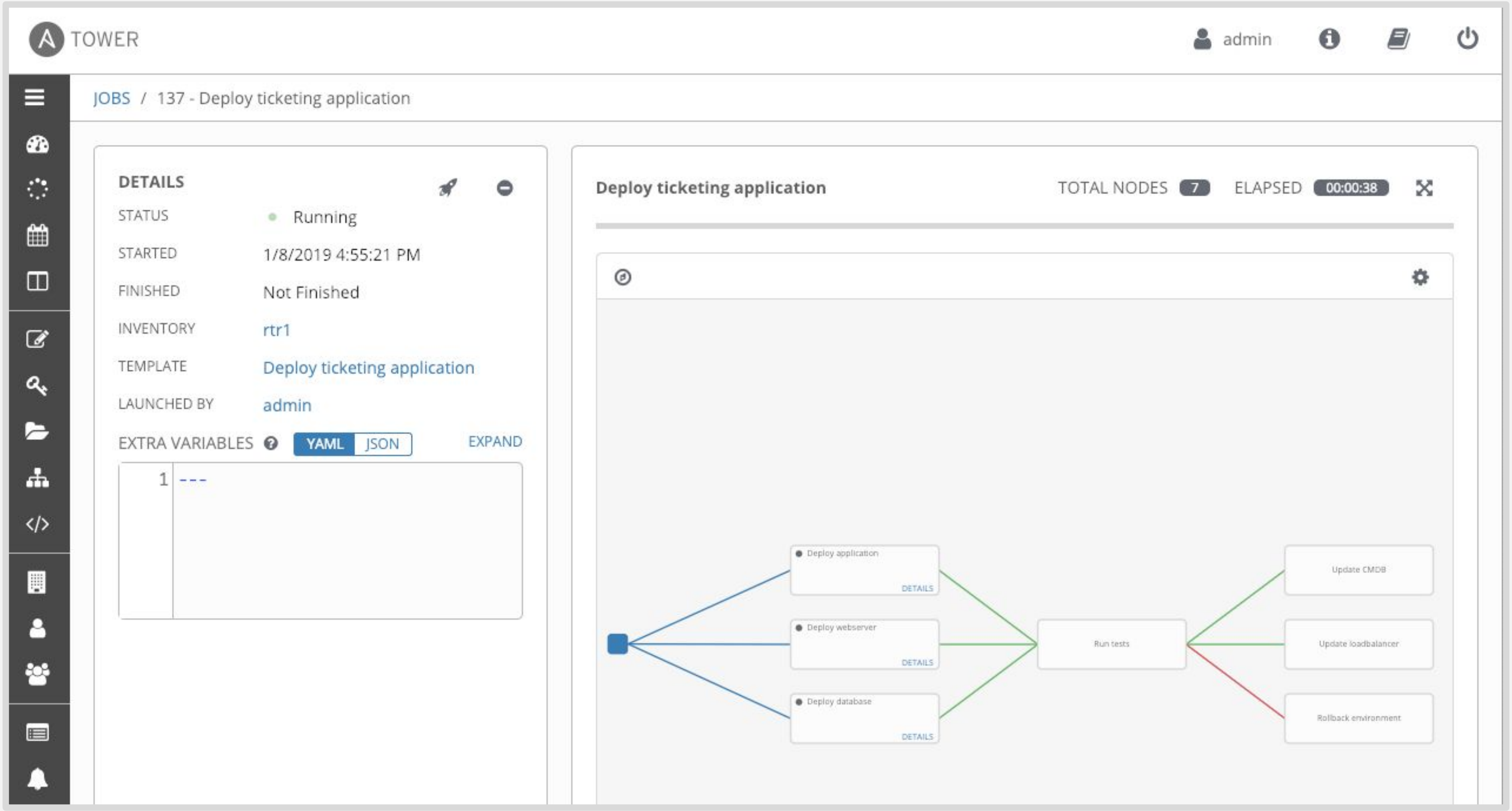
RECENTLY USED TEMPLATES VIEW ALL

NAME	ACTIVITY	ACTIONS
Deployment pipeline	🟥🟩🟩🟥🟩🟥🟥🟥🟥🟩	
Rollback deployment	🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥	
Deploy to development	🟥🟩🟩🟩🟩🟥🟥🟩🟩🟩	
Test application	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟩	
Deploy database	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	

RECENT JOB RUNS VIEW ALL

NAME	TIME
Deployment pipeline	1/23/2019 4:38:36 PM
Rollback deployment	1/23/2019 4:38:35 PM
Deploy to development	1/23/2019 4:38:19 PM
Test application	1/23/2019 4:38:18 PM
Deploy database	1/23/2019 4:38:05 PM

ANSIBLE TOWER FEATURES: CREATE AUTOMATION WORKFLOWS



DETAILS

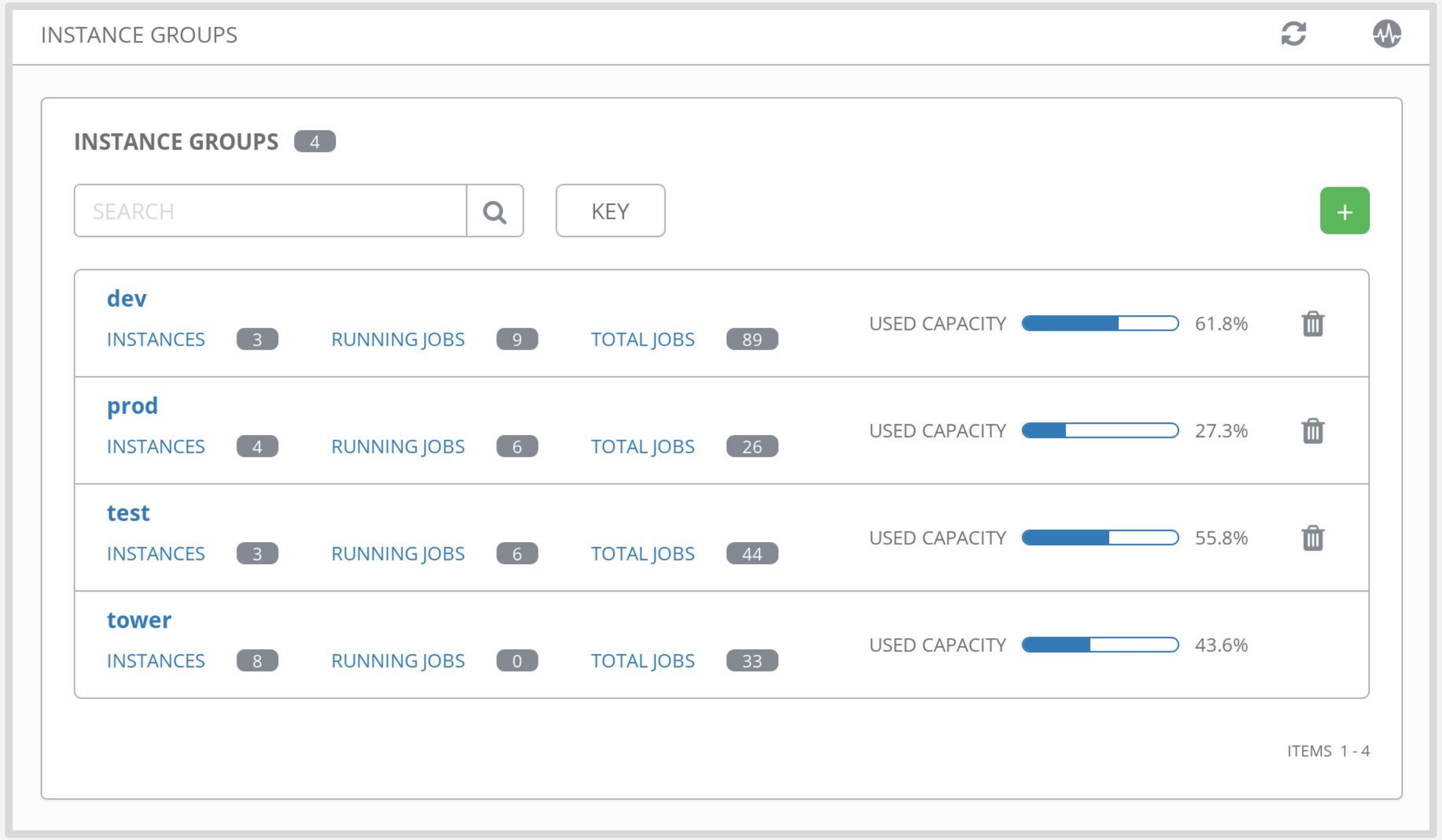
STATUS	● Running
STARTED	1/8/2019 4:55:21 PM
FINISHED	Not Finished
INVENTORY	rtr1
TEMPLATE	Deploy ticketing application
LAUNCHED BY	admin

EXTRA VARIABLES YAML JSON EXPAND

```
---
```

Deploy ticketing application TOTAL NODES 7 ELAPSED 00:00:38

ANSIBLE TOWER FEATURES: SCALE OUT CLUSTERING


INSTANCE GROUPS

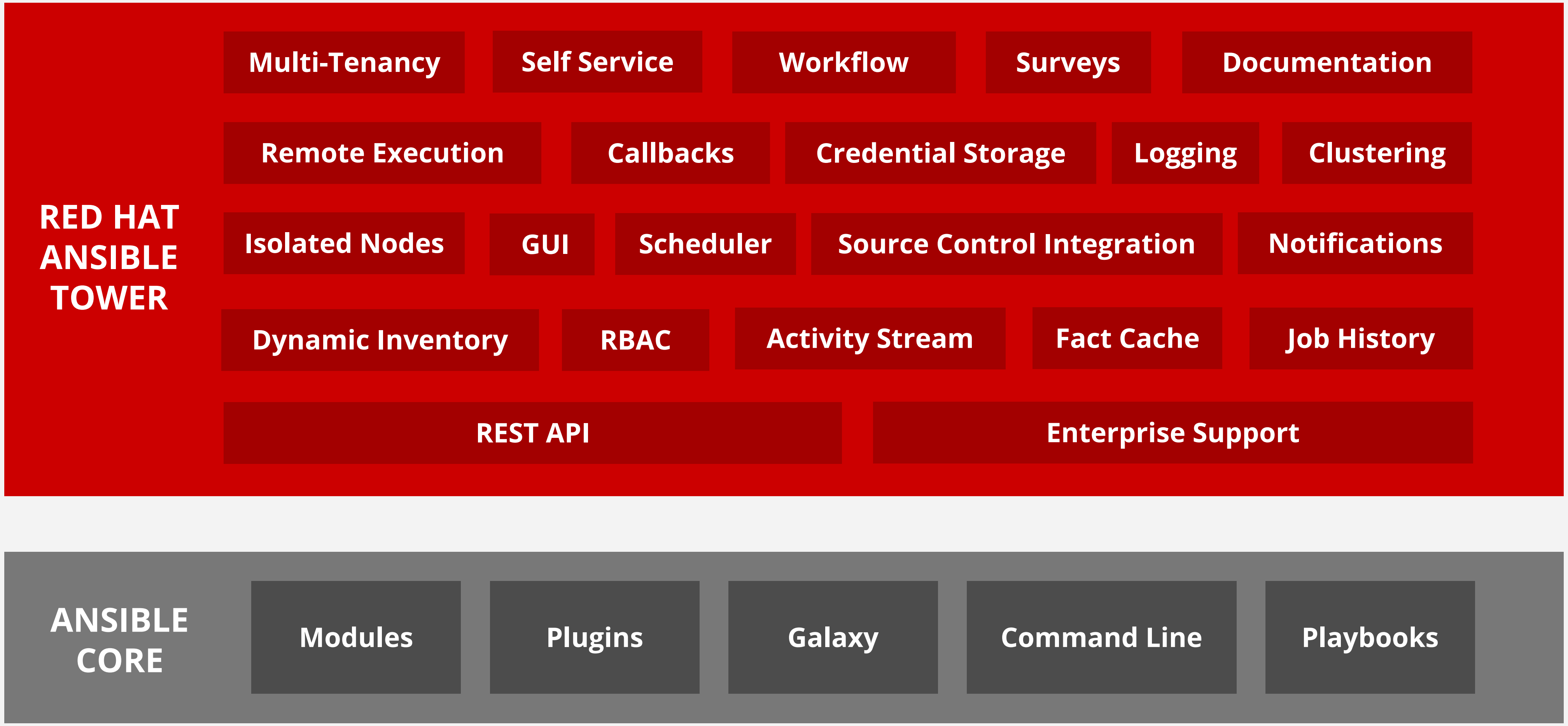
INSTANCE GROUPS 4

SEARCH KEY

Instance group	INSTANCES	RUNNING JOBS	TOTAL JOBS	USED CAPACITY
dev	3	9	89	61.8%
prod	4	6	26	27.3%
test	3	6	44	55.8%
tower	8	0	33	43.6%

ITEMS 1 - 4

Red Hat Ansible Tower



Thank you!

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat