RED HAT® ANSIBLE® Tower

AUTOMATION ACROSS THE ENTERPRISE

redhat.

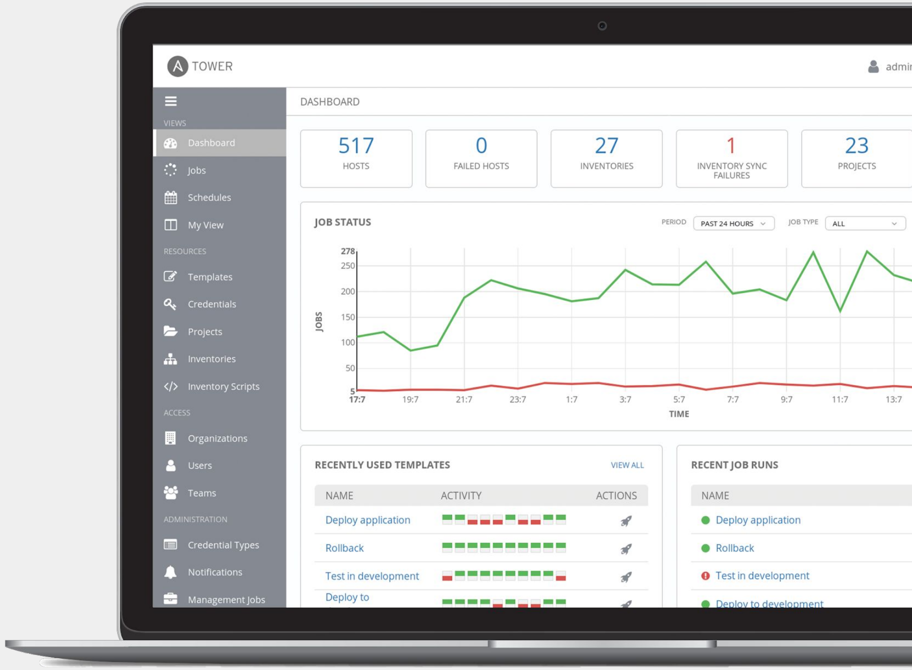
WHAT WILL YOU LEARN?

- What is Ansible Tower
- How Ansible Tower Works
- Installing Ansible Tower
- Key Features

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



517	0	27	1	23
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS

RECENTLY USED TEMPLATES

NAME	ACTIVITY	ACTIONS
Deploy application		
Rollback		
Test in development		
Deploy to		

RECENT JOB RUNS

NAME
● Deploy application
● Rollback
● Test in development
● Deploy to development

redhat.

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.

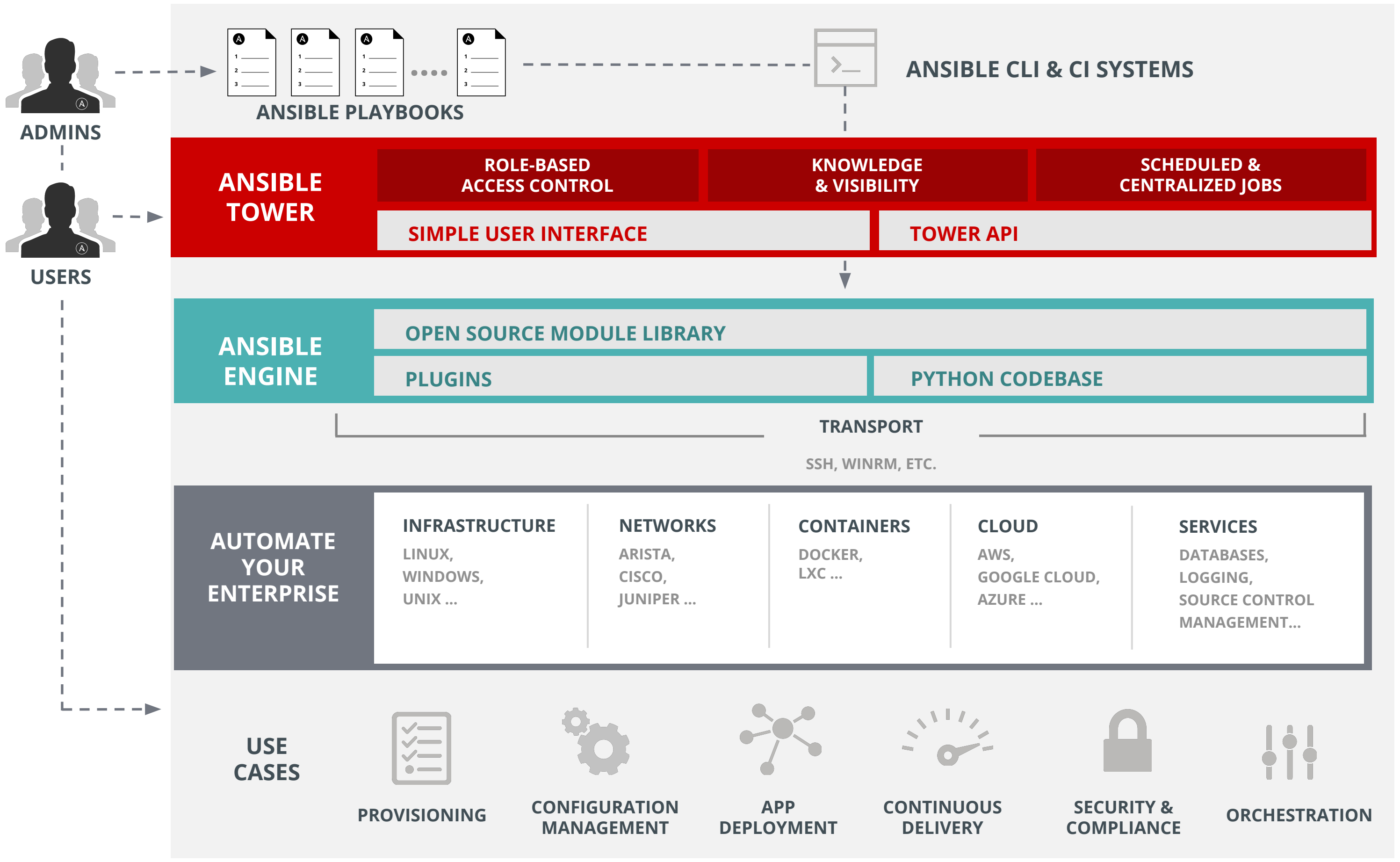
ENTERPRISE INTEGRATIONS

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.

redhat.


ADMINS
ANSIBLE PLAYBOOKS
ANSIBLE CLI & CI SYSTEMS
USERS
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

INSTALLING ANSIBLE TOWER

```
# the most common and preferred way of
# installation for Red Hat Enterprise Linux
$ wget https://bit.ly/ansibletower

# bundled installer can be downloaded for
# Red Hat Enterprise Linux
$ wget https://bit.ly/ansibletowerbundle

# looking for a specific version? navigate to
# http://releases.ansible.com/ansible-tower
# to see all the versions available for download
```

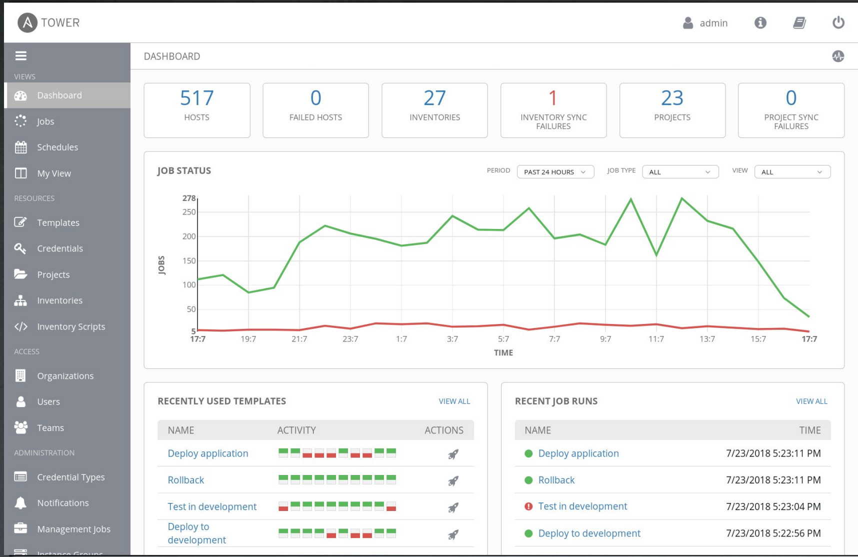
SERVER REQUIREMENTS

- Red Hat Enterprise Linux (RHEL) 7 (and select derivatives), Ubuntu 14.04 64-bit, and Ubuntu 16.04 LTS 64-bit support required (kernel and runtime).
- A currently supported version of Mozilla Firefox or Google Chrome.
- 2 GB RAM minimum (4+ GB RAM highly recommended)
- 20 GB of dedicated hard disk space

RED HAT® ANSIBLE® Tower

FEATURE OVERVIEW:

TOWER CONCEPTS



517 HOSTS | 0 FAILED HOSTS | 27 INVENTORIES | 1 INVENTORY SYNC FAILURES | 23 PROJECTS | 0 PROJECT SYNC FAILURES

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

RECENTLY USED TEMPLATES

NAME	ACTIVITY	ACTIONS
Deploy application		
Rollback		
Test in development		
Deploy to development		

RECENT JOB RUNS

NAME	TIME
● Deploy application	7/23/2018 5:23:11 PM
● Rollback	7/23/2018 5:23:11 PM
● Test in development	7/23/2018 5:23:04 PM
● Deploy to development	7/23/2018 5:22:56 PM

redhat.

USER MANAGEMENT

- A **user** is an account to access Ansible Tower and its services given the permissions granted to it.
- An **organization** is a logical collection of users, teams, projects, inventories and more. All entities belong to an organization with the exception of users.
- **Teams** provide a means to implement role-based access control schemes and delegate responsibilities across organizations.

CREDENTIALS

Credentials are utilized by Ansible Tower for authentication with various external resources:

- Connecting to remote machines to run jobs
- Syncing with inventory sources
- Importing project content from version control systems
- Connecting to and managing network devices

Centralized management of various credentials allows end users to leverage a secret without ever exposing that secret to them.

INVENTORY

Inventory is a collection of hosts (nodes) with associated data and groupings that Ansible Tower can connect to and manage.

- Hosts (nodes)
- Groups
- Inventory-specific data (variables)
- Static or dynamic sources

PROJECTS

A Project is a logical collection of Ansible Playbooks, represented in Ansible Tower.

You can manage Playbooks and Playbook directories by placing them in a **source code management system** supported by Ansible Tower, including Git, Subversion, and Mercurial.

JOB TEMPLATES

A job template is a definition and set of parameters for running an Ansible Playbook.

Job templates are useful to **execute** the same job many times and encourage the **reuse** of Ansible Playbook content and collaboration between teams.

JOBS

A job is an instance of Ansible Tower launching an Ansible Playbook against an inventory of hosts.

- Job results can be easily viewed
- View the standard out for a more in-depth look

ROLE BASED ACCESS CONTROL (RBAC)

Role-Based Access Controls (RBAC) are built into Ansible Tower and allow administrators to **delegate access** to server inventories, organizations, and more. These controls allow Ansible Tower to help you **increase security** and **streamline management** of your Ansible automation.

ROLE BASED ACCESS CONTROL (RBAC)

Role-Based Access Controls (RBAC) are built into Ansible Tower and allow administrators to **delegate access** to server inventories, organizations, and more. These controls allow Ansible Tower to help you **increase security** and **streamline management** of your Ansible automation.

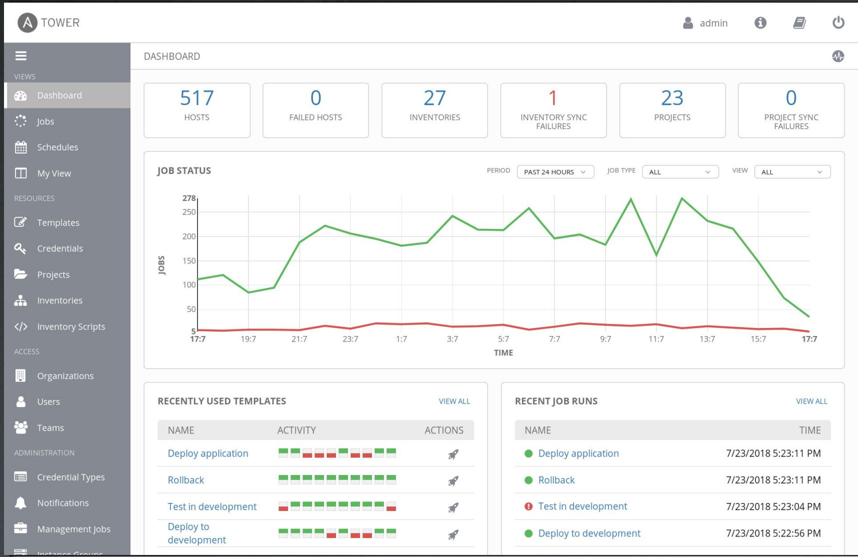
DYNAMIC INVENTORY

Dynamic inventory is a script that queries a service, like a cloud provider API or a management application. This data is formatted in an Ansible-specific JSON data structure and is used in lieu of static inventory files.

- Groups are generated based on host metadata
- Single source of truth saves time, avoids duplication and reduces human error
- Dynamic and static inventory sources can be used together

RED HAT® ANSIBLE® Tower

FEATURE OVERVIEW: CONTROL



517	0	27	1	23	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

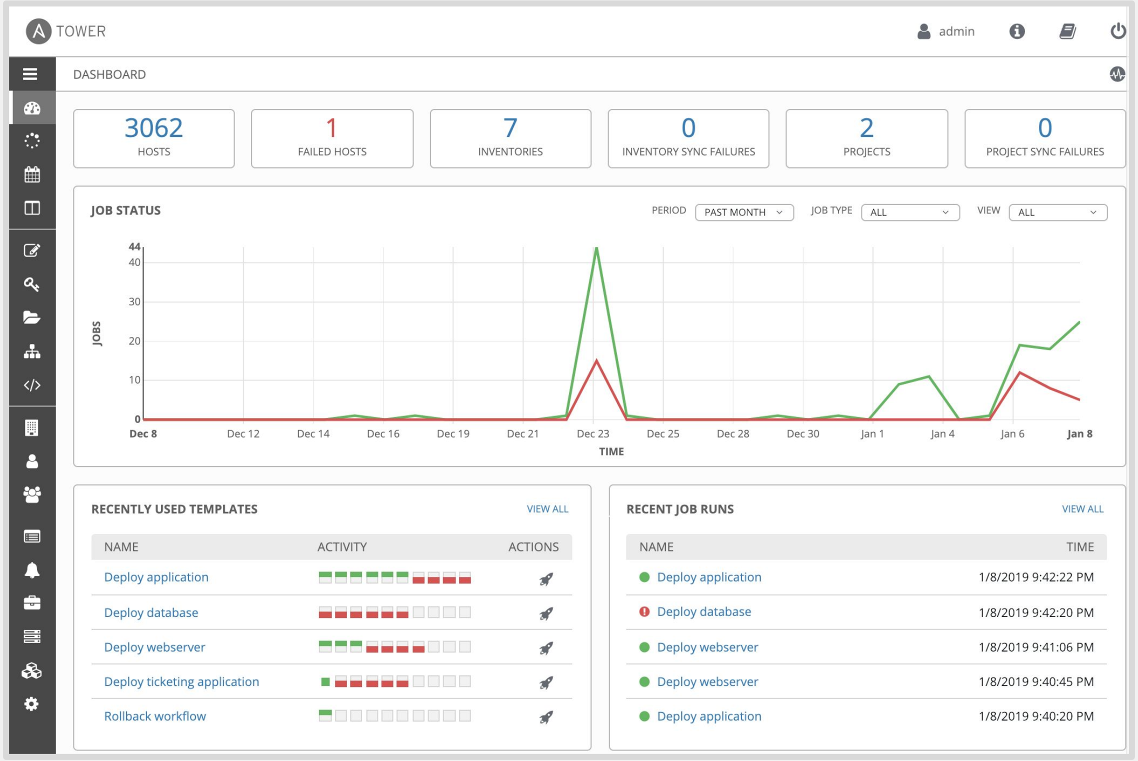
NAME	ACTIVITY	ACTIONS
Deploy application	🟩🟩🟥🟥🟥🟩🟥🟥🟩🟩	
Rollback	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	
Test in development	🟥🟩🟩🟩🟩🟩🟩🟩🟩🟥	
Deploy to development	🟩🟩🟩🟩🟥🟩🟥🟥🟩🟩	

RECENT JOB RUNS VIEW ALL

NAME	TIME
● Deploy application	7/23/2018 5:23:11 PM
● Rollback	7/23/2018 5:23:11 PM
● Test in development	7/23/2018 5:23:04 PM
● Deploy to development	7/23/2018 5:22:56 PM

redhat.

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

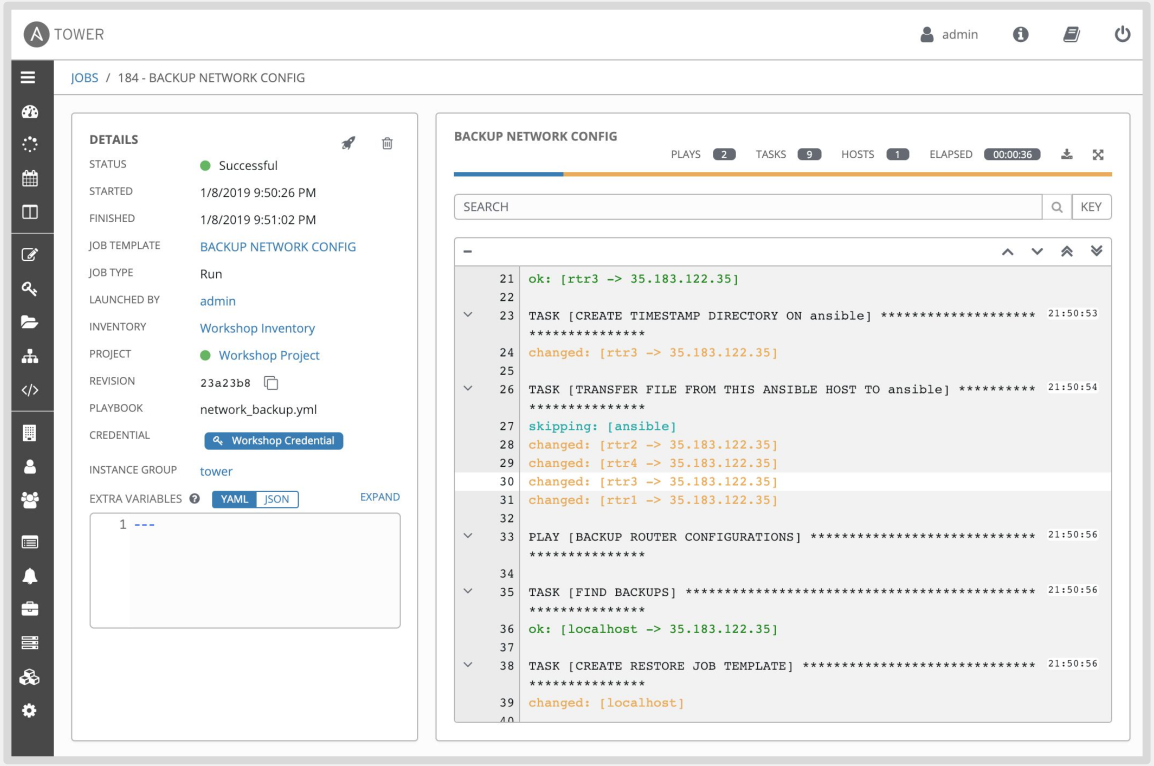
RECENTLY USED TEMPLATES — VIEW ALL

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



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

```
---
```

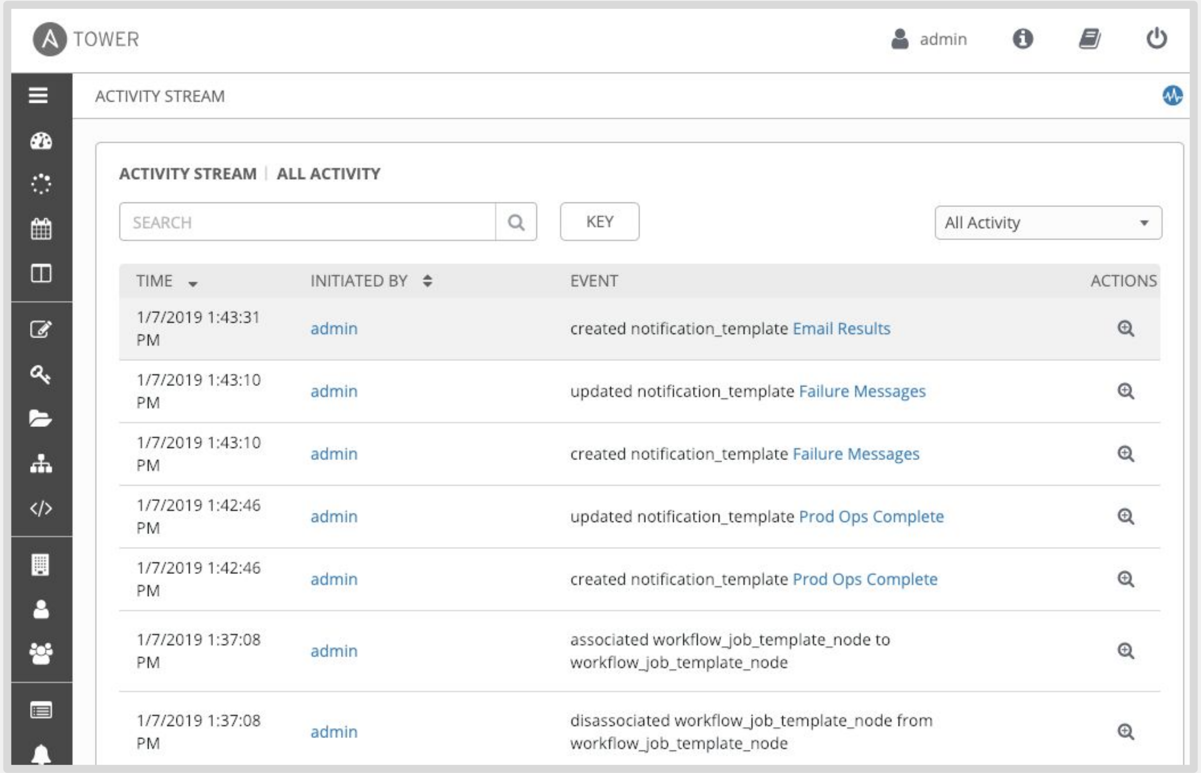
BACKUP NETWORK CONFIG

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

SEARCH KEY

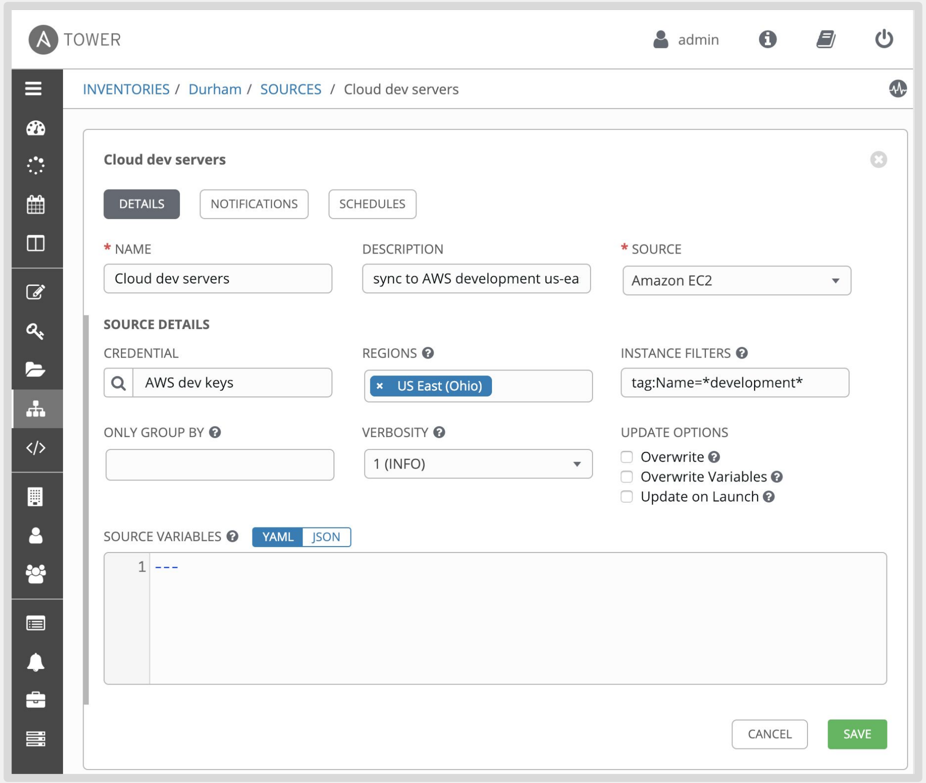
```
ok: [rtr3 -> 35.183.122.35]

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

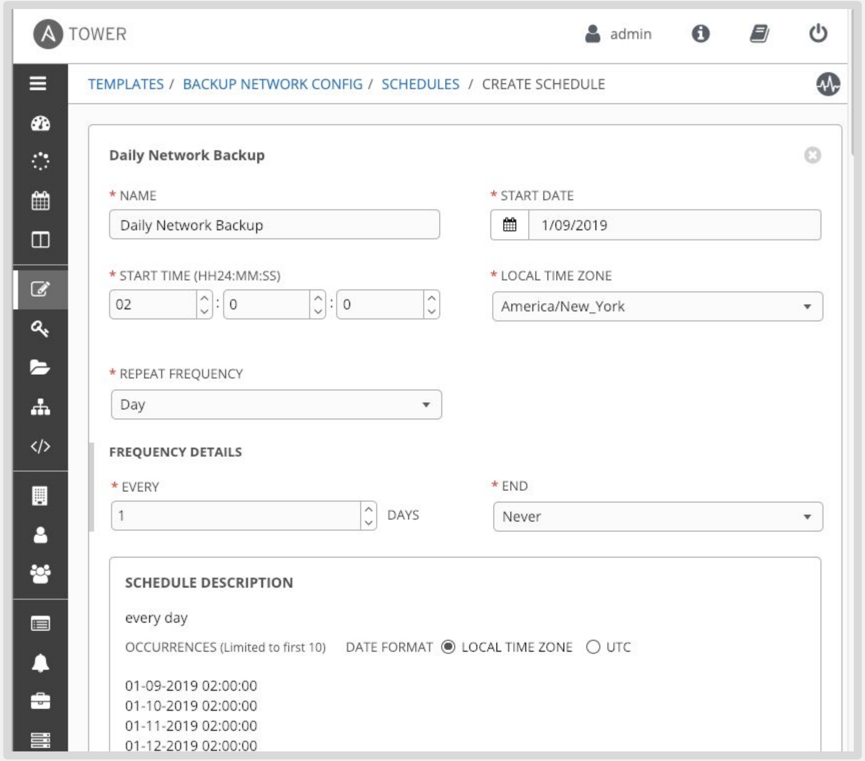
SOURCE DETAILS

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

SOURCE VARIABLES [✓ YAML] JSON

```
---
```

ANSIBLE TOWER FEATURES: SCHEDULE JOBS



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

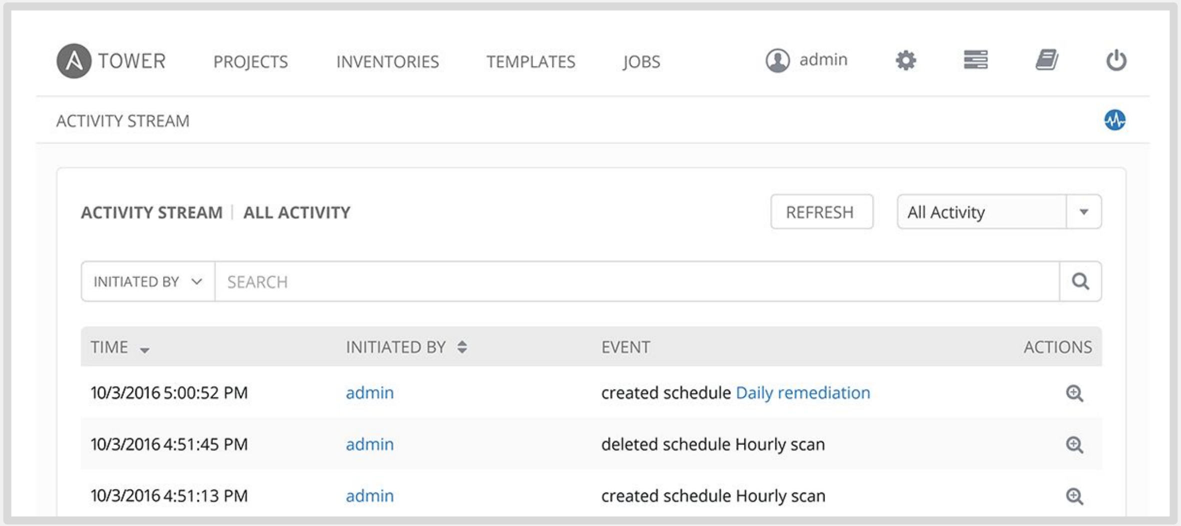
SCHEDULE DESCRIPTION

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

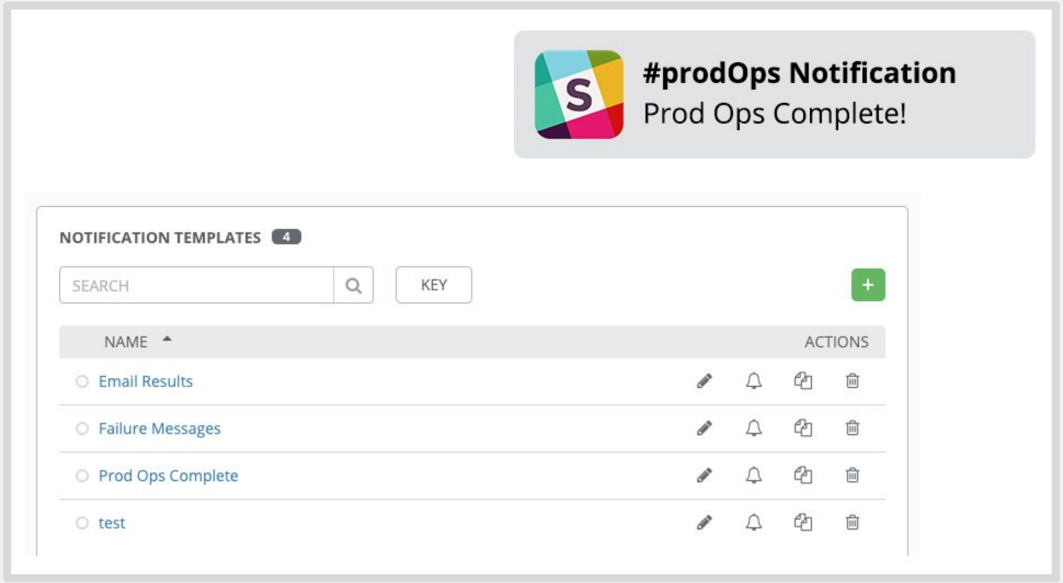
ANSIBLE TOWER FEATURES: EXTERNAL LOGGING



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: INTEGRATED NOTIFICATIONS


#prodOps Notification
Prod Ops Complete!

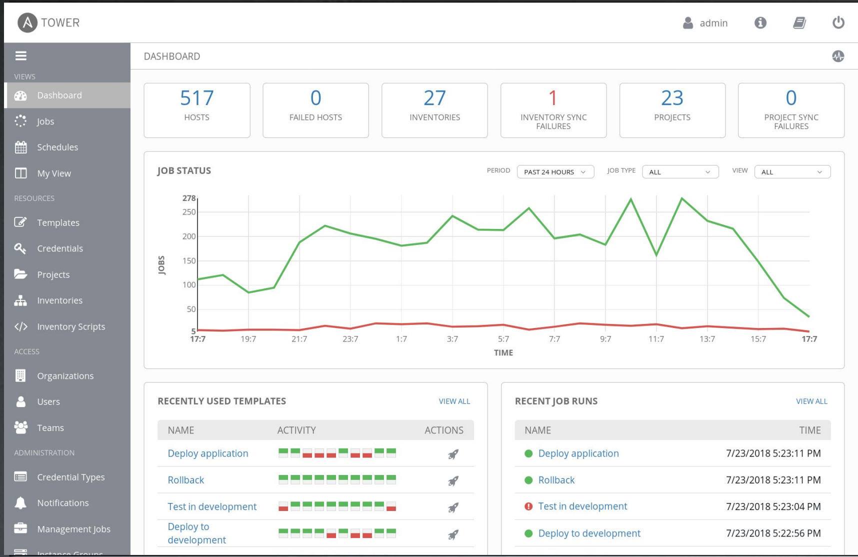
NOTIFICATION TEMPLATES 4

NAME	ACTIONS
Email Results	
Failure Messages	
Prod Ops Complete	
test	

RED HAT® ANSIBLE® Tower

FEATURE OVERVIEW:

DELEGATION



517	0	27	1	23	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

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

RECENTLY USED TEMPLATES

NAME	ACTIVITY	ACTIONS
Deploy application		
Rollback		
Test in development		
Deploy to development		

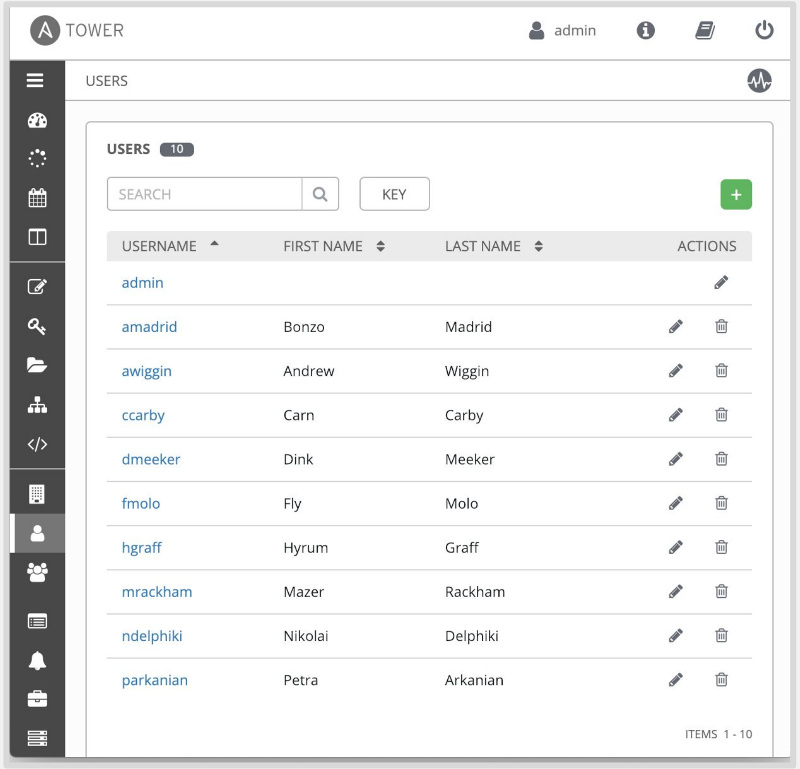
RECENT JOB RUNS

NAME	TIME
● Deploy application	7/23/2018 5:23:11 PM
● Rollback	7/23/2018 5:23:11 PM
● Test in development	7/23/2018 5:23:04 PM
● Deploy to development	7/23/2018 5:22:56 PM

redhat.

ANSIBLE TOWER FEATURES: ROLE BASED ACCESS CONTROL

USERS



TOWER admin

USERS

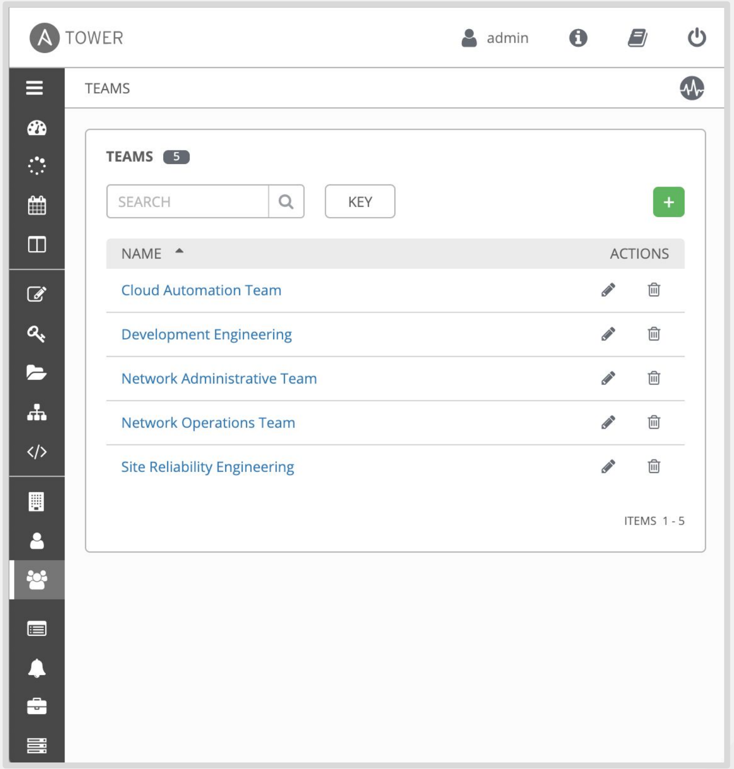
USERS 10

SEARCH KEY +

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



TOWER admin

TEAMS

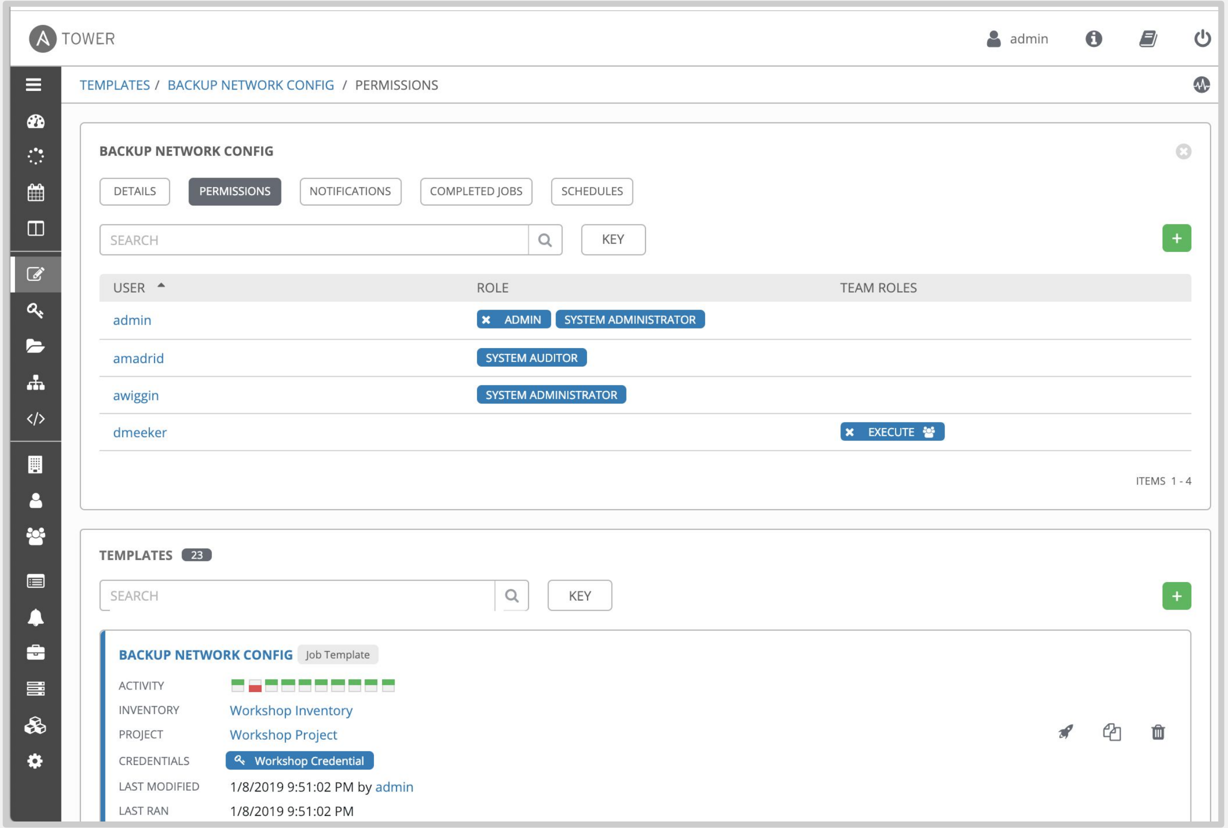
TEAMS 5

SEARCH KEY +

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

ITEMS 1 - 5

ANSIBLE TOWER FEATURES: ROLE BASED ACCESS CONTROL



BACKUP NETWORK CONFIG

DETAILS | PERMISSIONS | NOTIFICATIONS | COMPLETED JOBS | SCHEDULES

SEARCH | KEY

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

ITEMS 1 - 4

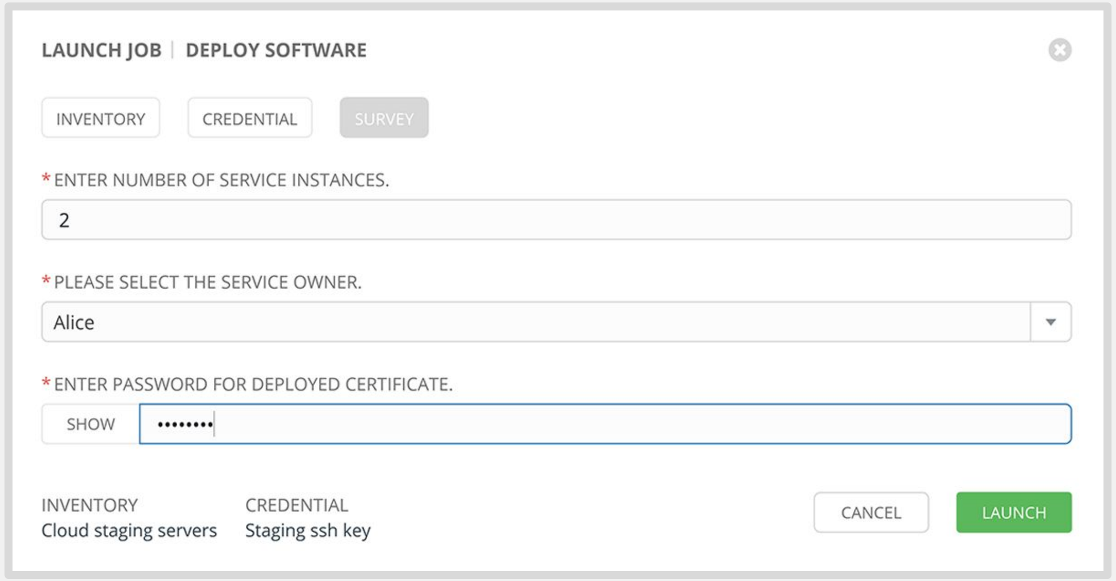
TEMPLATES 23

SEARCH | KEY

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: SELF-SERVICE I.T.



* ENTER NUMBER OF SERVICE INSTANCES.

2

* PLEASE SELECT THE SERVICE OWNER.

Alice

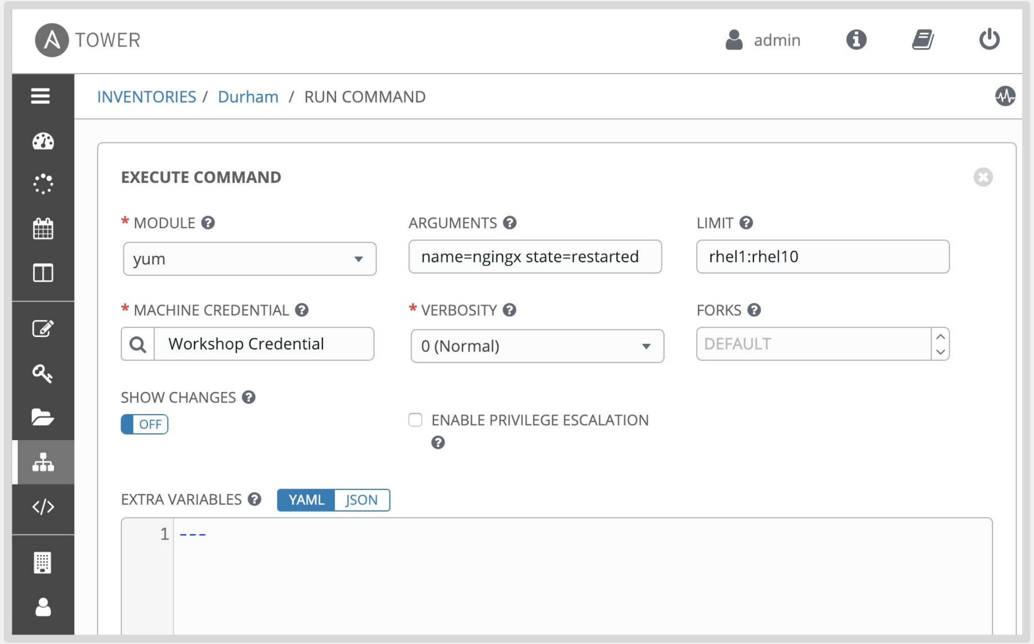
* ENTER PASSWORD FOR DEPLOYED CERTIFICATE.

SHOW ••••••••

INVENTORY
Cloud staging servers

CREDENTIAL
Staging ssh key

ANSIBLE TOWER FEATURES: REMOTE COMMAND EXECUTION



EXECUTE COMMAND

* MODULE: yum

ARGUMENTS: name=ngingx state=restarted

LIMIT: rhel1:rhel10

* MACHINE CREDENTIAL: Workshop Credential

* VERBOSITY: 0 (Normal)

FORKS: DEFAULT

SHOW CHANGES: OFF

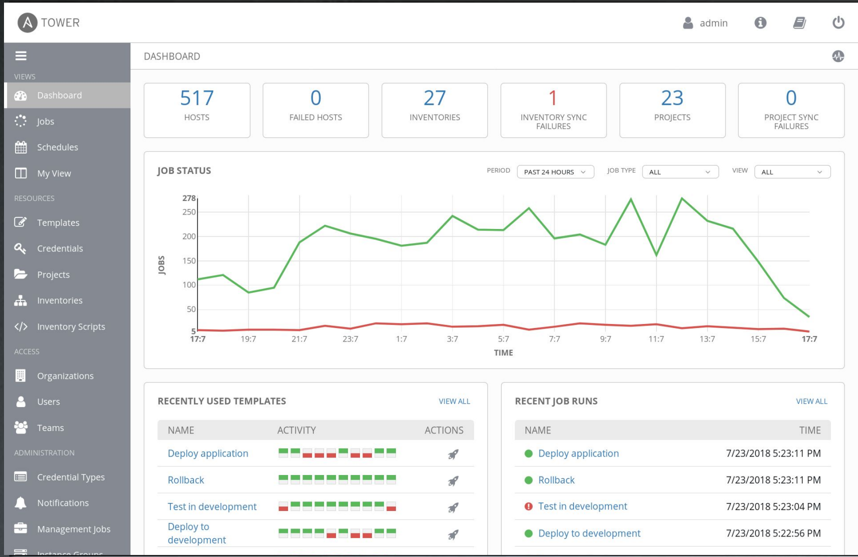
ENABLE PRIVILEGE ESCALATION

EXTRA VARIABLES [✓ YAML] JSON

```
---
```

RED HAT® ANSIBLE® Tower

FEATURE OVERVIEW: SCALE



DASHBOARD

517	0	27	1	23	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

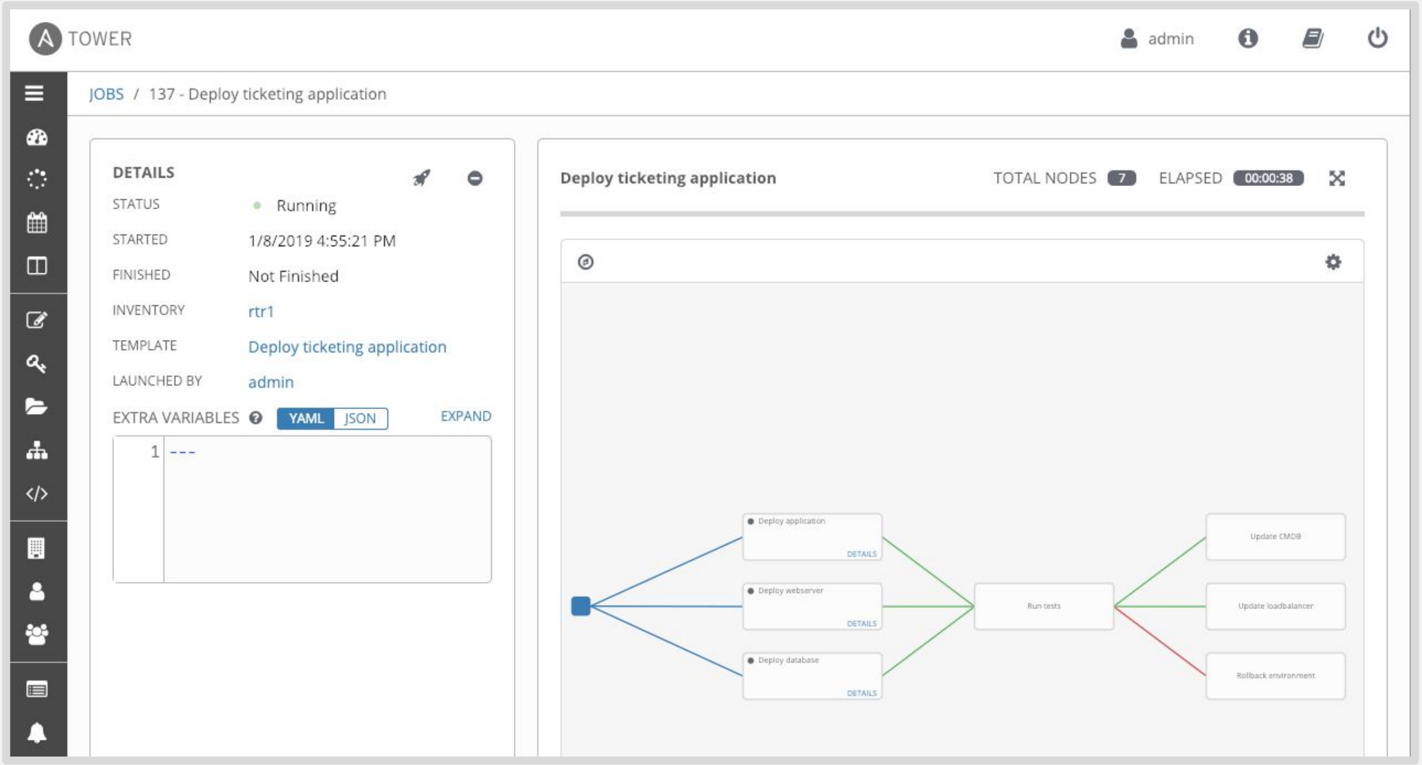
RECENTLY USED TEMPLATES

NAME	ACTIVITY	ACTIONS
Deploy application		
Rollback		
Test in development		
Deploy to development		

RECENT JOB RUNS

NAME	TIME
● Deploy application	7/23/2018 5:23:11 PM
● Rollback	7/23/2018 5:23:11 PM
● Test in development	7/23/2018 5:23:04 PM
● Deploy to development	7/23/2018 5:22:56 PM

ANSIBLE TOWER FEATURES: CREATE AUTOMATION WORKFLOWS



DETAILS

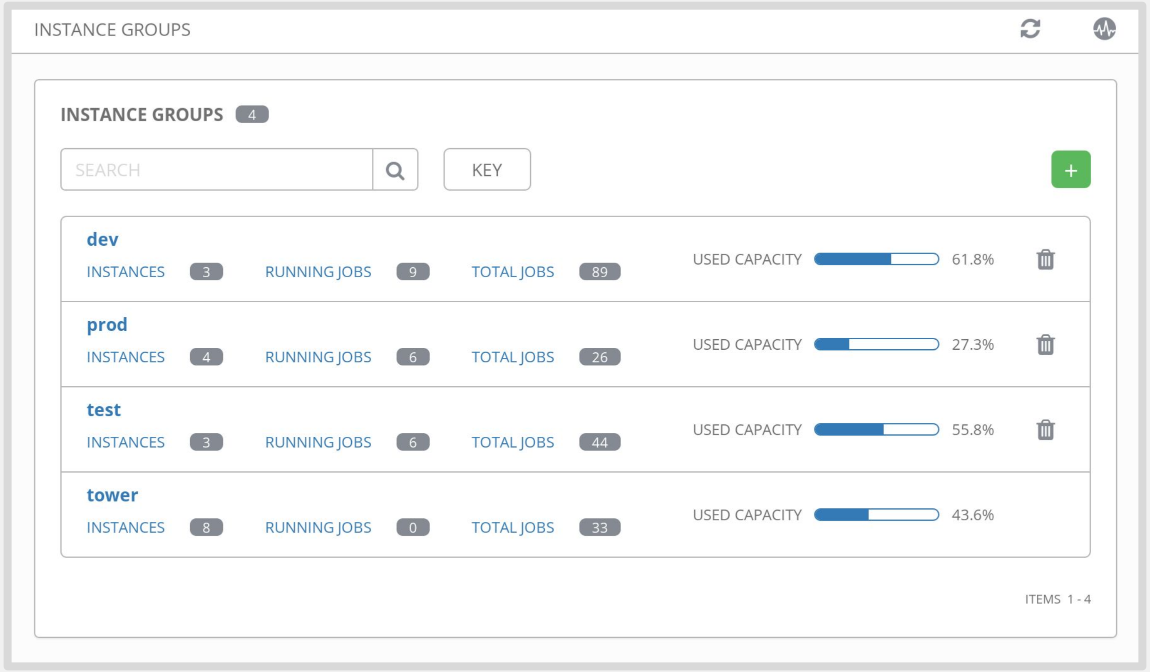
STATUS	● Running
STARTED	1/8/2019 4:55:21 PM
FINISHED	Not Finished
INVENTORY	rtr1
TEMPLATE	Deploy ticketing application
LAUNCHED BY	admin

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

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

ITEMS 1 - 4

NEXT STEPS

GET STARTED

ansible.com/get-started

ansible.com/tower-trial

JOIN THE COMMUNITY

ansible.com/community

WORKSHOPS & TRAINING

ansible.com/workshops

Red Hat Training

SHARE YOUR STORY

Follow us @Ansible

Friend us on Facebook

redhat.