



Red Hat

Ansible Automation Platform

Please mute your microphone and video
Presentation will begin at 1:03 pm Eastern
Please post your questions in the chat

Automation for all

Ansible technical introduction and overview

Peter Gervase
Solutions Architect, RHCA
pgervase@redhat.com



Red Hat
Ansible Automation
Platform

Automation for all

Ansible technical introduction and overview

Peter Gervase
Solutions Architect, RHCA
pgervase@redhat.com



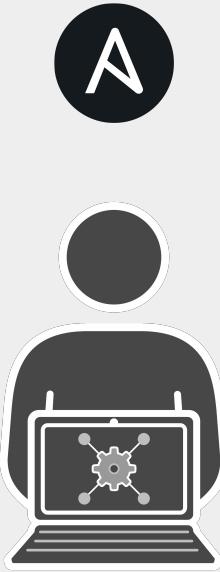
Red Hat

Ansible Automation Platform

Agenda

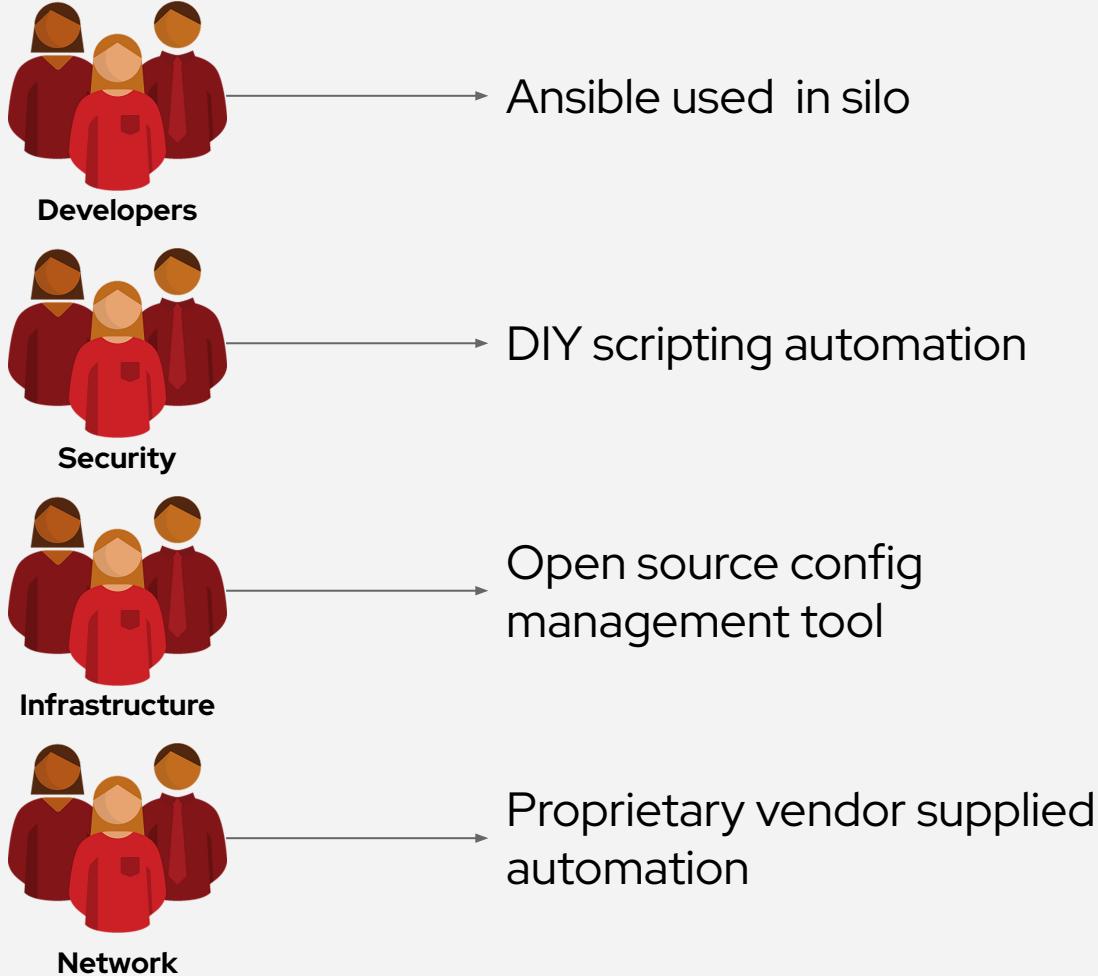
- What is automation?
- Why Ansible?
- Ansible Tower
- Red Hat's cloud.redhat.com
- Demonstrations





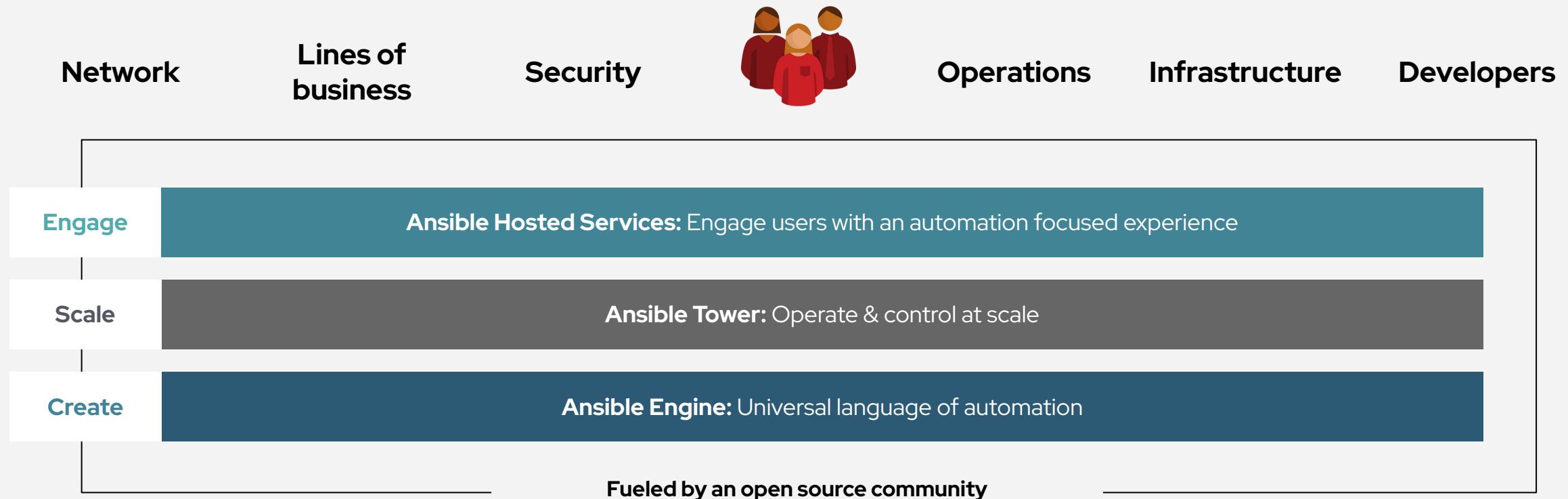
Automation happens when one person meets a
problem they never want to solve again

Ad-hoc Automation is happening in silos



Is organic
automation enough?

Red Hat Ansible Automation Platform



Teams are automating...



Lines Of Business



Network



Security



Operations



Developers



Infrastructure

Why 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



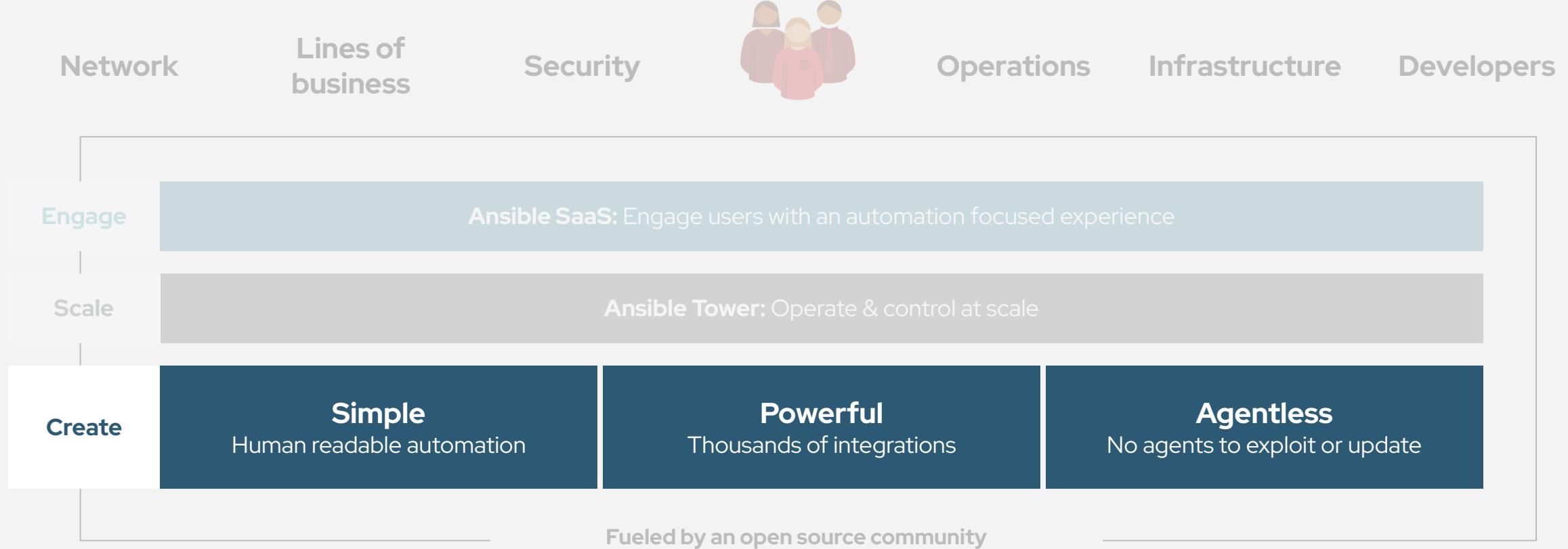
Red Hat

Ansible Automation Platform

Red Hat Ansible Engine:
Universal language
of automation



Red Hat Ansible Automation Platform



Red Hat Ansible Engine

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

Orchestration plays well with others: ServiceNow, Infoblox, AWS, Terraform, Cisco ACI and more



Ansible Engine Demo

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

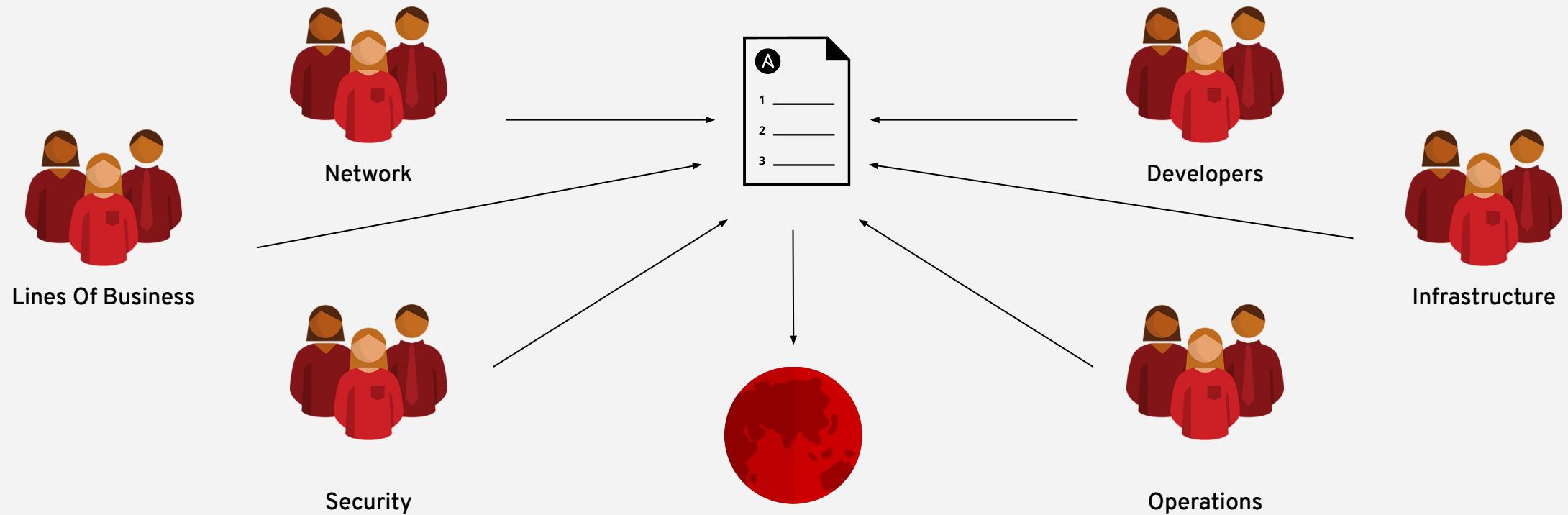
And more...

Ansible automates technologies you use

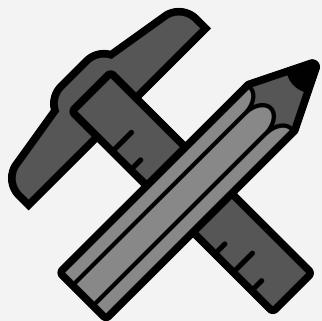
Time to automate is measured in minutes

Cloud	Virt & Container	Windows	Network	Security	Monitoring
AWS	Docker	ACLs	A10	Checkpoint	Dynatrace
Azure	VMware	Files	Arista	Cisco	Datadog
Digital Ocean	RHV	Packages	Aruba	CyberArk	LogicMonitor
Google	OpenStack	IIS	Cumulus	F5	New Relic
OpenStack	OpenShift	Regedits	Bigswitch	Fortinet	Sensu
Rackspace	+more	Shares	Cisco	Juniper	+more
+more		Services	Dell	IBM	
Operating Systems	Storage	Configs	Extreme	Palo Alto	Devops
RHEL	Netapp	Users	F5	Snort	Jira
Linux	Red Hat Storage	Domains	Lenovo	+more	GitHub
Windows	Infinidat	+more	MikroTik		Vagrant
+more	+more		Juniper		Jenkins
			OpenSwitch		Slack
			+more		+more

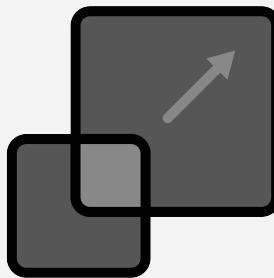
When automation crosses teams, you need an automation platform



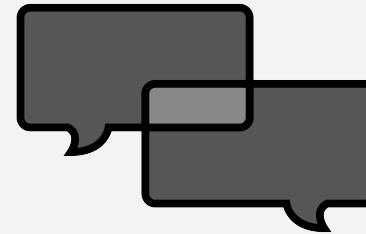
A platform can help you:



Create



Scale



Engage



Red Hat

Ansible Automation Platform

Red Hat Ansible Tower:
Operate and
control at scale



Red Hat Ansible Tower

by the numbers:

94%

Reduction in recovery time
following a security incident

84%

Savings by deploying workloads
to generic systems appliances
using Ansible Tower

67%

Reduction in man hours required
for customer deliveries

Financial summary:

146%

ROI on Ansible Tower

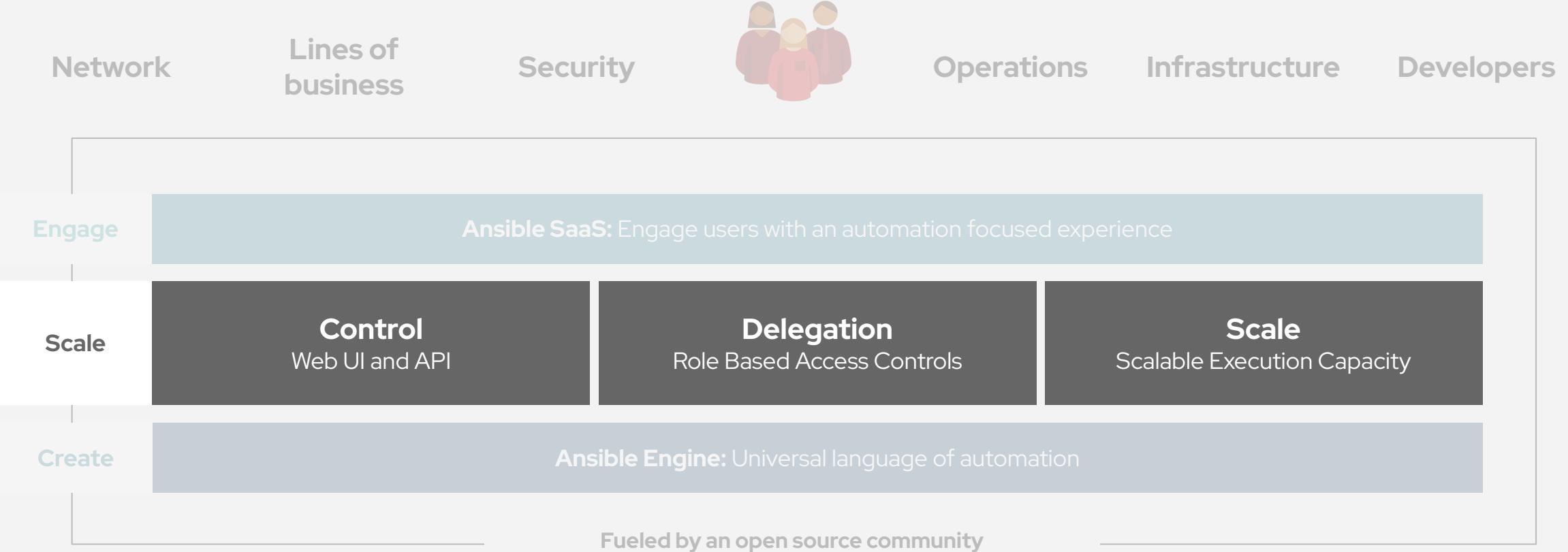
<3 MONTHS

Payback on Ansible Tower

SOURCE: "The Total Economic Impact™ Of Red Hat Ansible Tower, a June 2018 commissioned study conducted by Forrester Consulting on behalf of Red Hat."
redhat.com/en/engage/total-economic-impact-ansible-tower-20180710



Red Hat Ansible Automation Platform



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



Red Hat Ansible Tower

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.

RBAC

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

Enterprise integrations

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

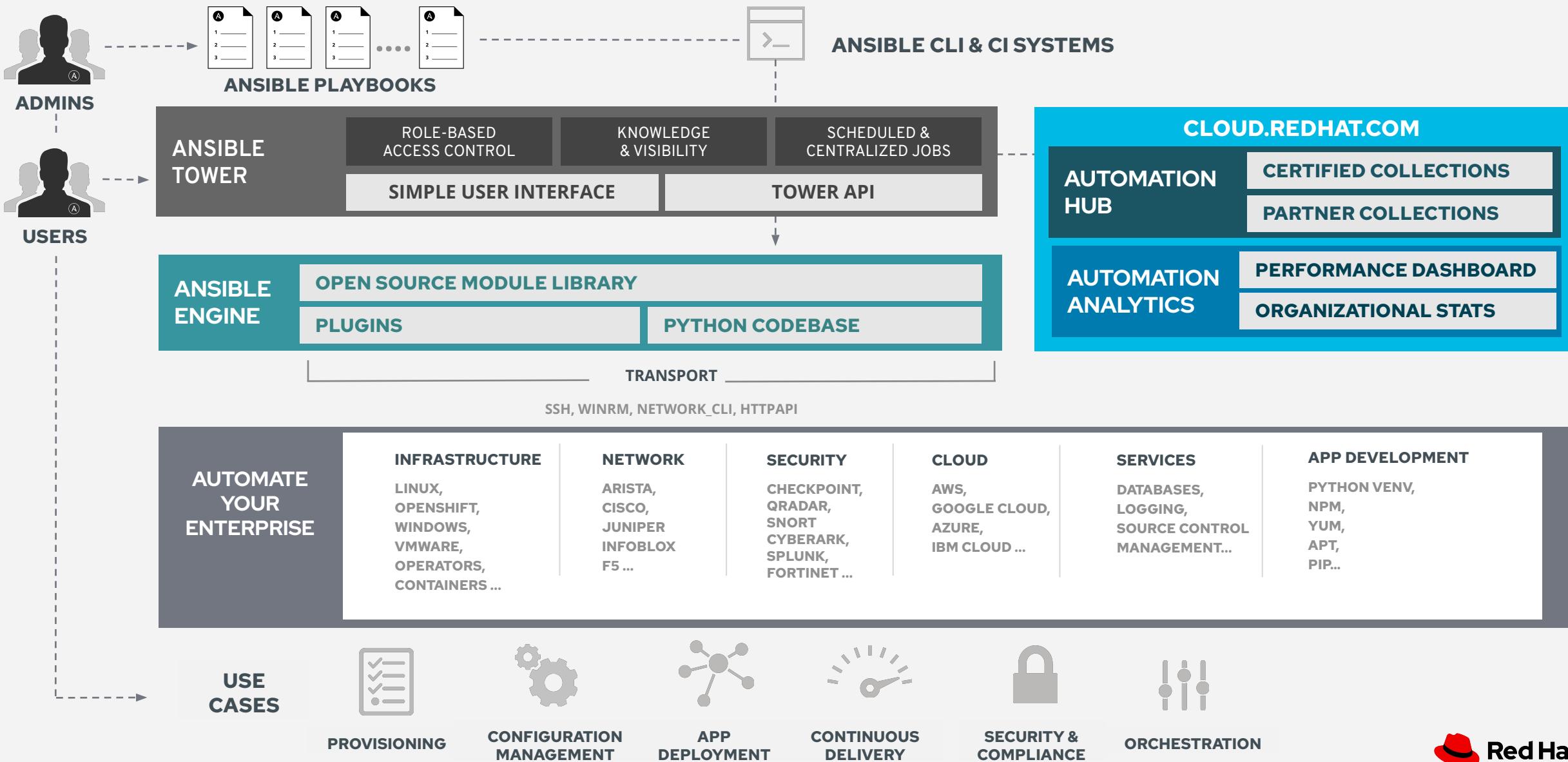
Centralized logging

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.

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.

Ansible Automation Platform



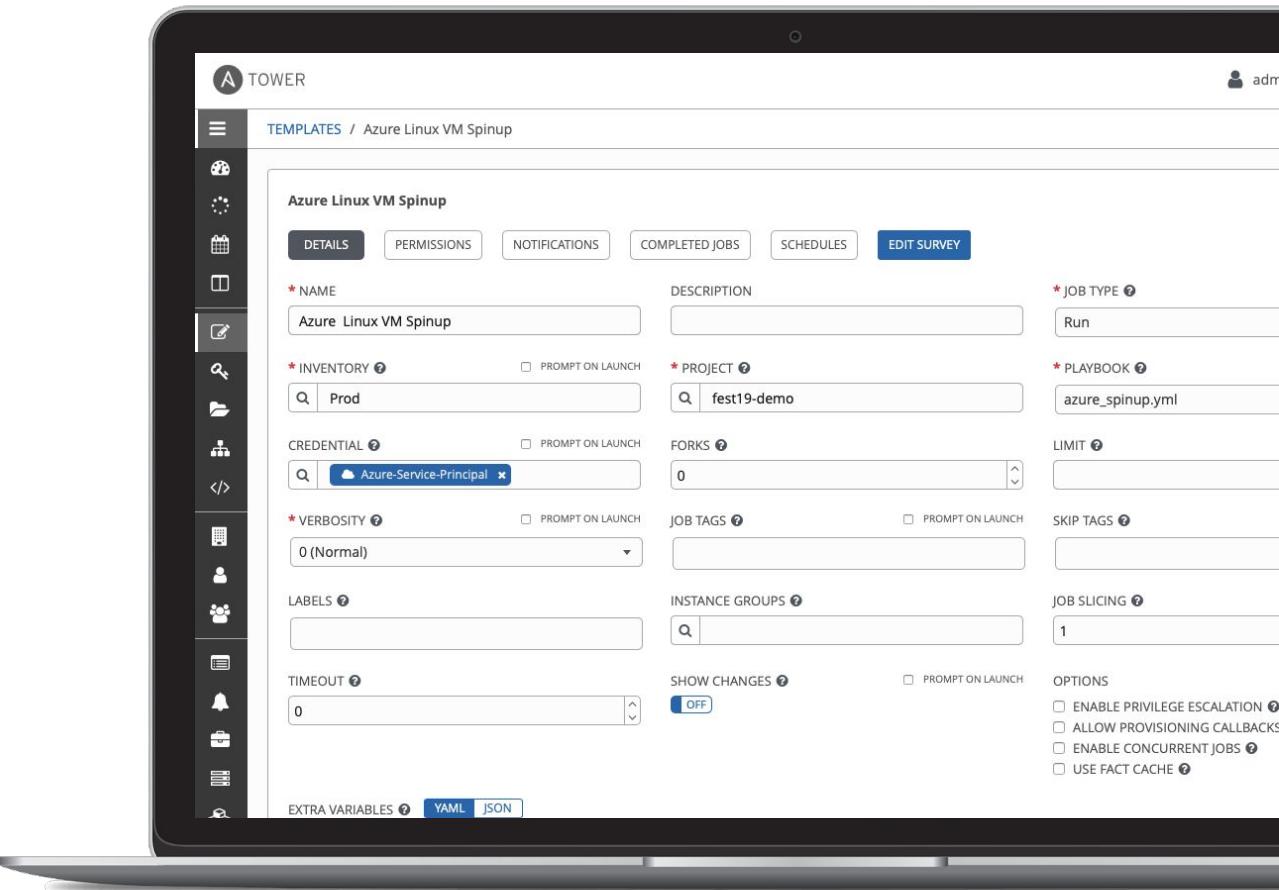
Job Templates

Everything in Ansible Tower revolves around the concept of a **Job Template**. Job Templates allow Ansible Playbooks to be controlled, delegated, and scaled for an organization.

Job templates also encourage the reuse of Ansible Playbook content and collaboration between teams.

A **Job Template** requires:

- An **Inventory** to run the job against
- A **Credential** to login to devices.
- A **Project** which contains Ansible Playbooks



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

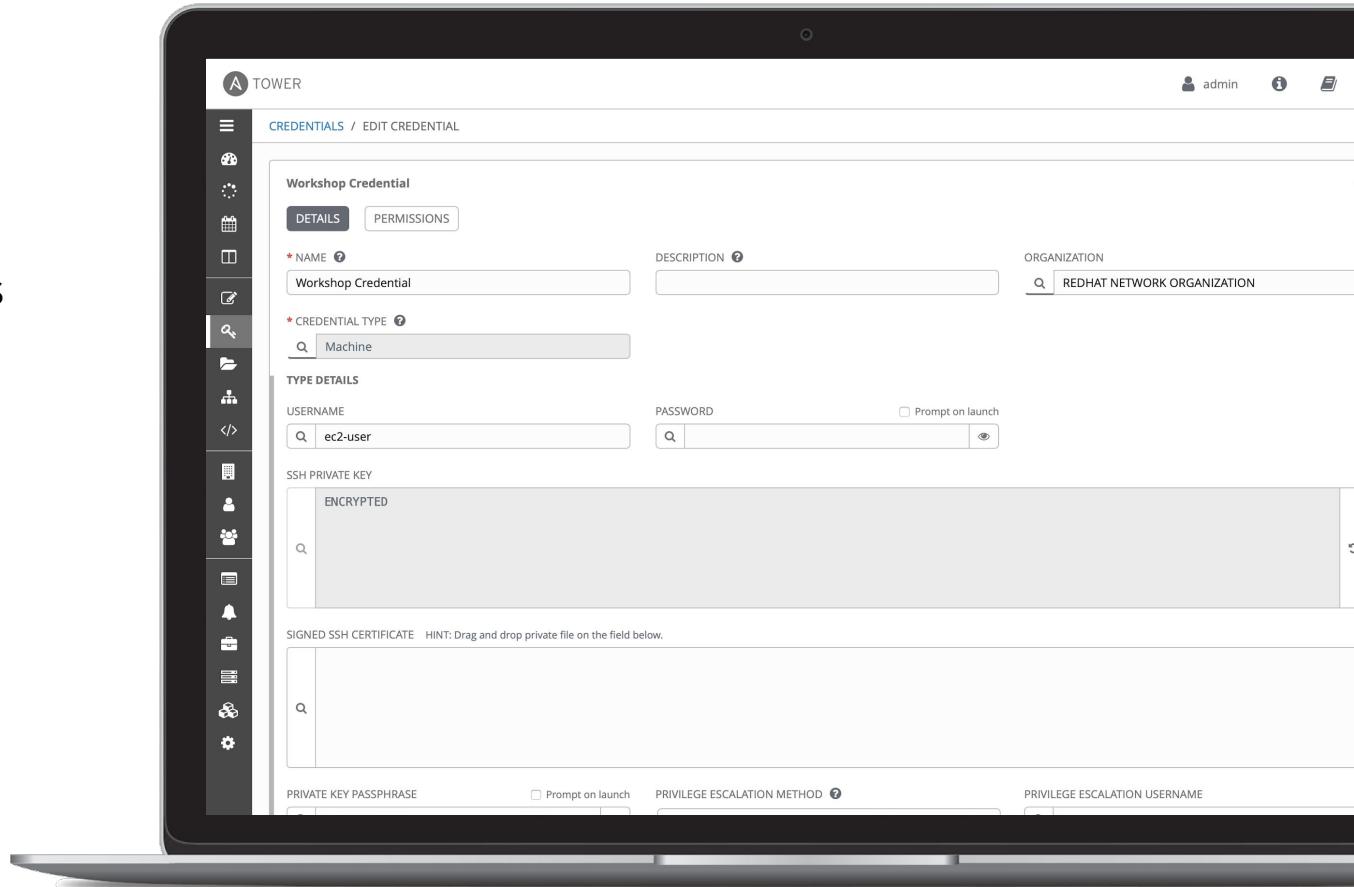
The screenshot shows the Ansible Tower interface on a laptop screen. The main window displays the 'Workshop Inventory' host list. The left sidebar has a dark theme with various icons for navigation. The top navigation bar shows 'INVENTORIES / Workshop Inventory / HOSTS'. The main content area has tabs for 'DETAILS', 'PERMISSIONS', 'GROUPS', 'HOSTS' (which is selected), 'SOURCES', and 'COMPLETED JOBS'. A search bar and a 'KEY' button are also present. The host list table has columns for 'HOSTS' (with checkboxes and status indicators like 'ON'), 'NAME', 'TYPE', and 'ORGANIZATION'. Under 'HOSTS', there are five entries: 'ansible', 'rtr1', 'rtr2', 'rtr3', and 'rtr4'. To the right of each host entry are 'RELATED GROUPS' buttons, such as 'control', 'cisco', 'dc1', 'arista', 'dc2', 'dc1', 'juniper', 'arista', and 'dc2'. The bottom of the screen shows another part of the interface with tabs for 'INVENTORIES' and 'HOSTS', a search bar, and filters for 'NAME', 'TYPE', and 'ORGANIZATION'.

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

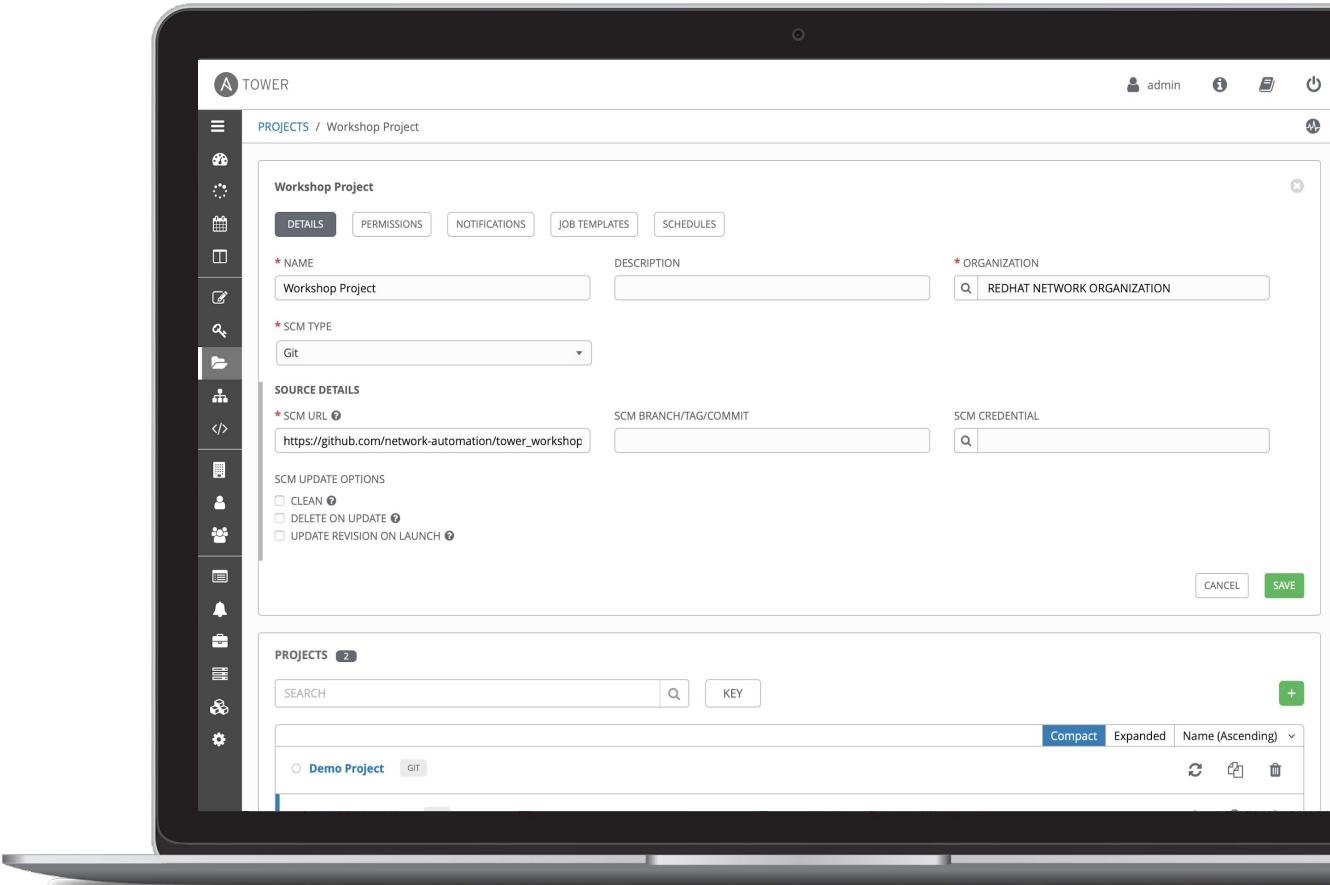
Centralized management of various credentials allows end users to leverage a secret without ever exposing that secret to them.



Project

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

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



RESTful API

Fully browsable API,
everything within the Web UI
can be accessed via the API
for programmatic access

The screenshot shows a web-based API browser for the Tower REST API. At the top, there's a header with the Tower logo, the text "TOWER REST API", a user icon labeled "admin", and a "Log out" button. To the right are standard browser navigation icons. Below the header, the URL "REST API / Version 2" is displayed. The main content area has a title "Version 2" with a question mark icon. Underneath, a "GET /api/v2/" row includes "OPTIONS" and "GET" buttons. The response section starts with "HTTP 200 OK" and lists various headers: Allow, Content-Type, Vary, X-API-Node, and X-API-Time. Below the headers is a large JSON object representing the API's root resources:

```
{  
    "ping": "/api/v2/ping/",  
    "instances": "/api/v2/instances/",  
    "instance_groups": "/api/v2/instance_groups/",  
    "config": "/api/v2/config/",  
    "settings": "/api/v2/settings/",  
    "me": "/api/v2/me/",  
    "dashboard": "/api/v2/dashboard/",  
    "organizations": "/api/v2/organizations/",  
    "users": "/api/v2/users/",  
    "projects": "/api/v2/projects/",  
    "project_updates": "/api/v2/project_updates/",  
    "teams": "/api/v2/teams/",  
    "credentials": "/api/v2/credentials/",  
}
```

A callout bubble points to the JSON content with the text: "This structured JSON output contains clickable links".

Role Based Access Control (RBAC)

Job Templates, Inventory, Credentials and Projects can be assigned to specific Users and Teams.

Clicking the USERS or TEAMS buttons shows available options

The screenshot shows a user interface for managing RBAC permissions. At the top, there's a navigation bar with 'TOWER' and a user icon 'admin'. Below it is a sidebar with 'VIEWS' and several resource categories: 'Dashboard', 'Jobs', 'Schedules', 'My View', 'TEMPLATES' (which is selected), 'Credentials', 'Projects', 'Inventories', 'Inventory Scans', and 'Organizations'. A modal window titled 'CONFIGURE RED HAT ENTERPRISE LINUX WEB SERVERS | ADD USERS / TEAMS' is open. It contains a step indicator '1 Please select Users / Teams from the lists below.' and two buttons: 'USERS' (highlighted with a red box) and 'TEAMS'. Below these buttons is a search bar with a magnifying glass icon and a 'KEY' button. The main area displays a table with columns 'USERNAME', 'FIRST NAME', and 'LAST NAME'. Two users are listed: 'ewiggin' (First Name: Ender, Last Name: Wiggin) and 'mrackham' (First Name: Mazer, Last Name: Rackham). Each user has a checkbox next to their name. At the bottom right of the modal are 'CANCEL' and 'SAVE' buttons.

USERNAME	FIRST NAME	LAST NAME
<input type="checkbox"/> ewiggin	Ender	Wiggin
<input type="checkbox"/> mrackham	Mazer	Rackham

Enterprise Authentication

Use your existing enterprise authentication including:

- Azure AD
- Github
- Google OAuth2
- LDAP
- Radius
- SAML
- TACACS+

The screenshot shows the Ansible Tower interface with the 'SETTINGS / AUTHENTICATION' tab selected. On the left, there's a sidebar with 'VIEWS' and 'RESOURCES' sections. Under 'RESOURCES', 'Credentials' is highlighted. The main panel displays the 'AUTHENTICATION' configuration. A horizontal row of buttons at the top allows switching between different authentication methods: AZURE AD, GITHUB, GOOGLE OAUTH2, LDAP, RADIUS, SAML, and TACACS+. The 'TACACS+' button is highlighted with a red box. Below these buttons are configuration fields for 'TACACS+ SERVER', 'TACACS+ PORT', 'TACACS+ SECRET', 'TACACS+ AUTH SESSION TIMEOUT', and 'TACACS+ AUTHENTICATION PROTOCOL'. At the bottom right are 'CANCEL' and 'SAVE' buttons.

Multiple supported enterprise authentication methods are easily integrated with Ansible Tower

Centralized Logging

Ansible Tower creates a centralized control point for Ansible Automation. If desired Ansible Tower can integrated with existing log aggregation services.

The screenshot shows the Ansible Tower interface with the sidebar menu open. The main area is titled 'SETTINGS / SYSTEM' and has a 'SYSTEM' tab selected. Under the 'LOGGING' tab, there are several configuration options:

- 'ENABLE EXTERNAL LOGGING' is set to 'OFF'.
- 'LOGGING AGGREGATOR' is set to 'log.eros.rhdemo.io'.
- 'LOGGING AGGREGATOR PORT' is empty.
- 'LOGGING AGGREGATOR USERNAME' is set to 'ender'.
- 'LOGGING AGGREGATOR PASSWORD/TOKEN' is empty, with a 'SHOW' button.
- 'LOG SYSTEM TRACKING FACTS INDIVIDUALLY' is set to 'OFF'.
- 'LOGGING AGGREGATOR LEVEL' is set to 'INFO'.
- 'LOGGING AGGREGATOR PROTOCOL' is set to 'HTTPS/HTTP'.
- 'ENABLE/DISABLE HTTPS CERTIFICATE' is empty.

A callout bubble points to the 'LOGGING AGGREGATOR TYPE' dropdown, which contains a list of supported external logging methods. The 'splunk' option is highlighted with a grey background and a red border around the entire dropdown area.

Multiple supported 3rd party external logging methods are easily integrated with Ansible Tower

- splunk
- logstash
- splunk
- loggly
- sumologic
- other

5

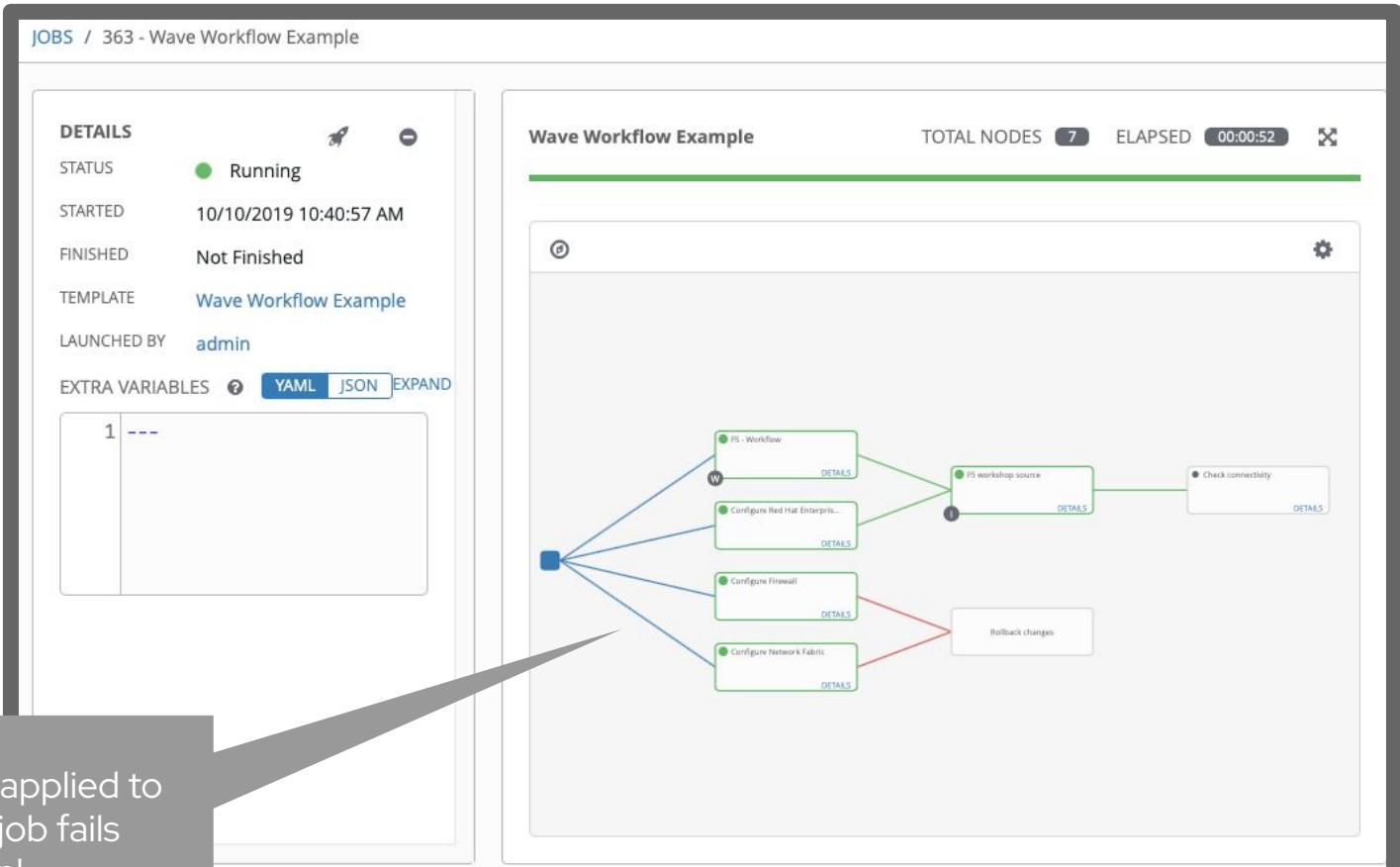
Workflows

Create powerful holistic automation using Ansible Workflows.

Orchestration can easily be configured by linking Job Templates.

Workflow approvals allow Workflows to pause and wait for human interaction

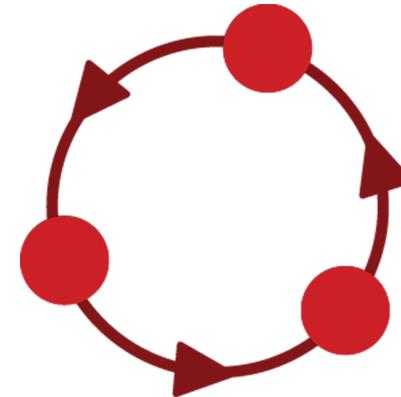
Flow logic can be applied to workflows. If this job fails this next Job is run!



Webhooks - Enabling GitOps

Trigger Job Templates or Workflows straight via
configurable webhooks

Automatically provision, update, configure, and
apply based on pushes to your source control.

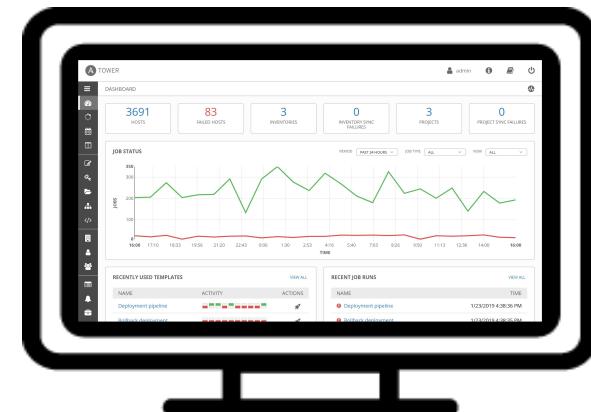


Scale

Ansible Tower clusters add redundancy and capacity, allowing you to scale Ansible automation across your enterprise.

- Unifying task execution across execution nodes
- Leverage Kubernetes and OpenShift to spin up execution capacity at runtime
- Expand execution to be able to pull jobs from a central Ansible Tower infrastructure

Ansible Tower





Red Hat

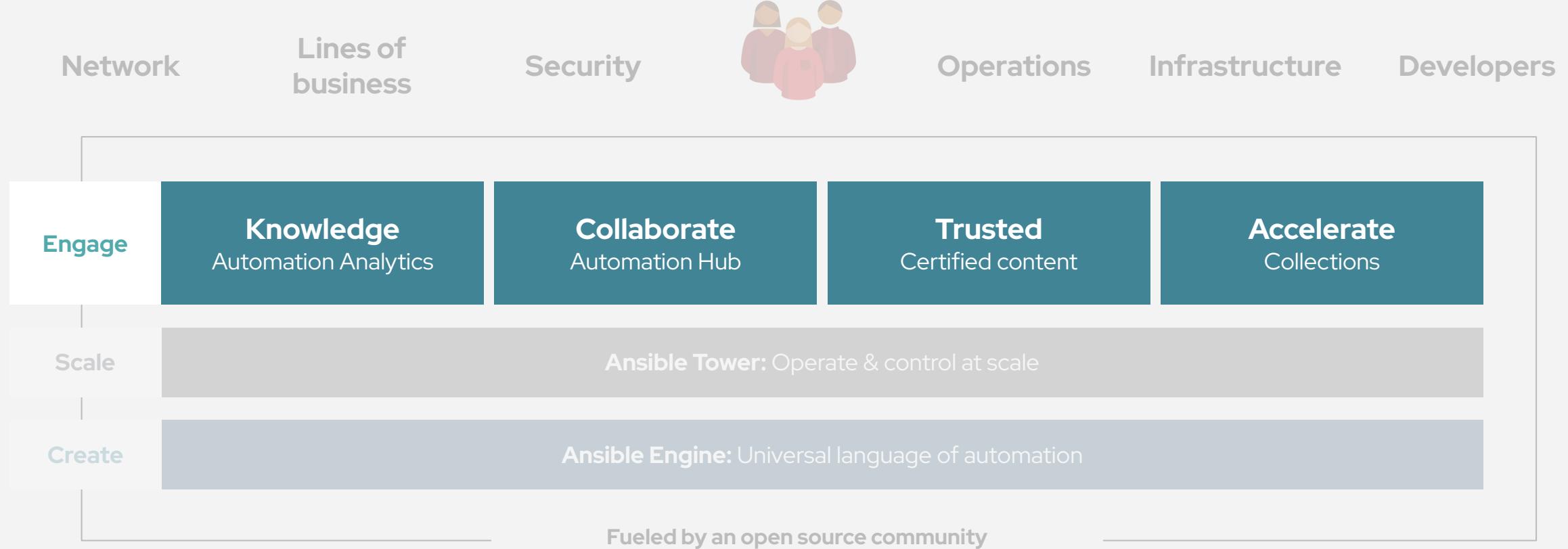
Ansible Automation
Platform

CLOUD.REDHAT.COM

Engage users with
an automation
focused experience



Red Hat Ansible Automation Platform



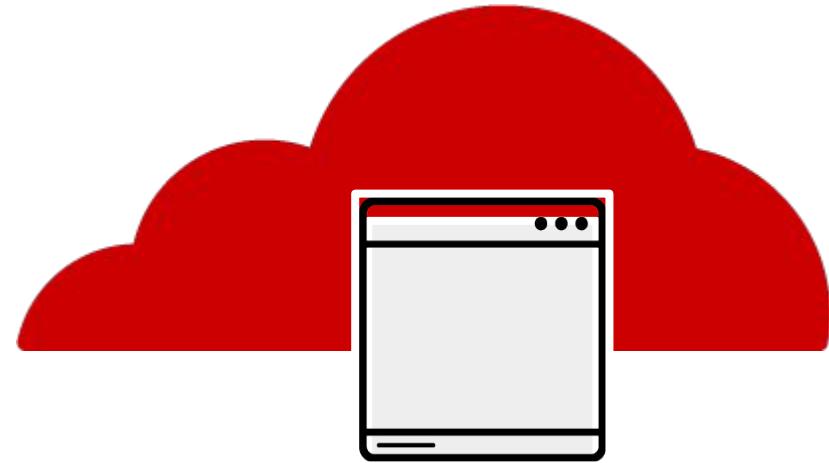
Automation Analytics: What is it?

SaaS (Software as a Service) on cloud.redhat.com

Analytics for all Ansible Tower clusters for an organization

Includes:

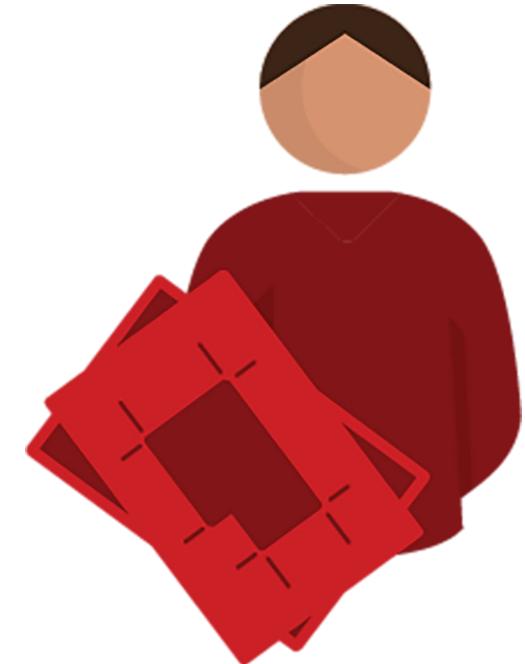
- visual dashboard
- health notifications
- organization statistics



Automation Analytics: What does it provide?

Enables an Automation Center of Excellence

- View information about automation health, usage and performance across your enterprise.
- Gain information about automation in your enterprise:
 - Which organizations are using the most automation?
 - Utilization rates
 - Enterprise-wide success and failure rates for automation





Red Hat

Ansible Automation Platform

Demonstrations

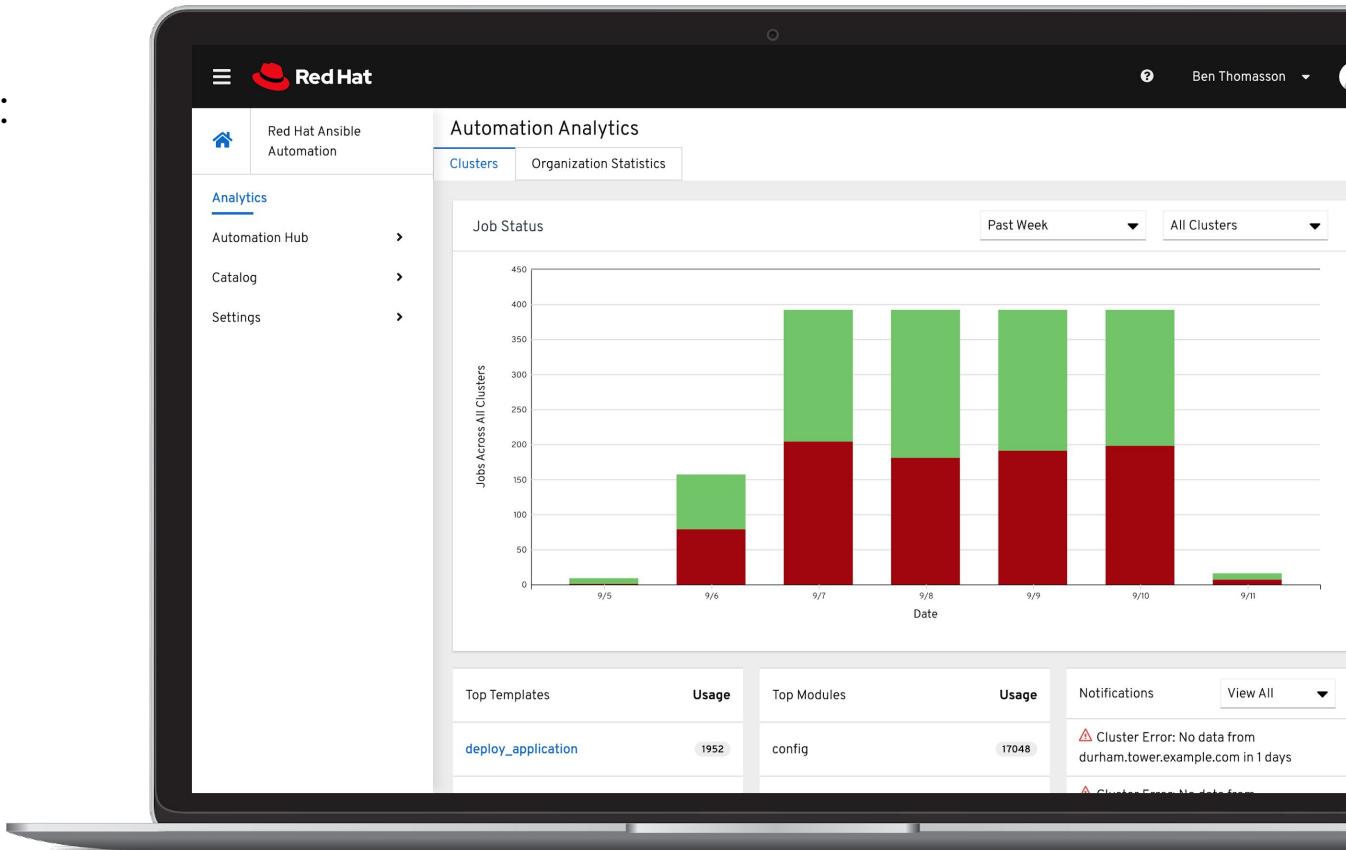
- Red Hat's cloud.redhat.com
- Ansible Tower



Analytics dashboard

Information across all clusters for an enterprise:

- Job Status graph
- Top Job Templates
- Top Modules



Health notifications

- Ansible Tower Cluster is down
- Node (within a cluster) is down
- Last time data was updated
- Near license count

Notifications View All ▾

⚠ Cluster Error: No data from durham.tower.example.com in 1 days

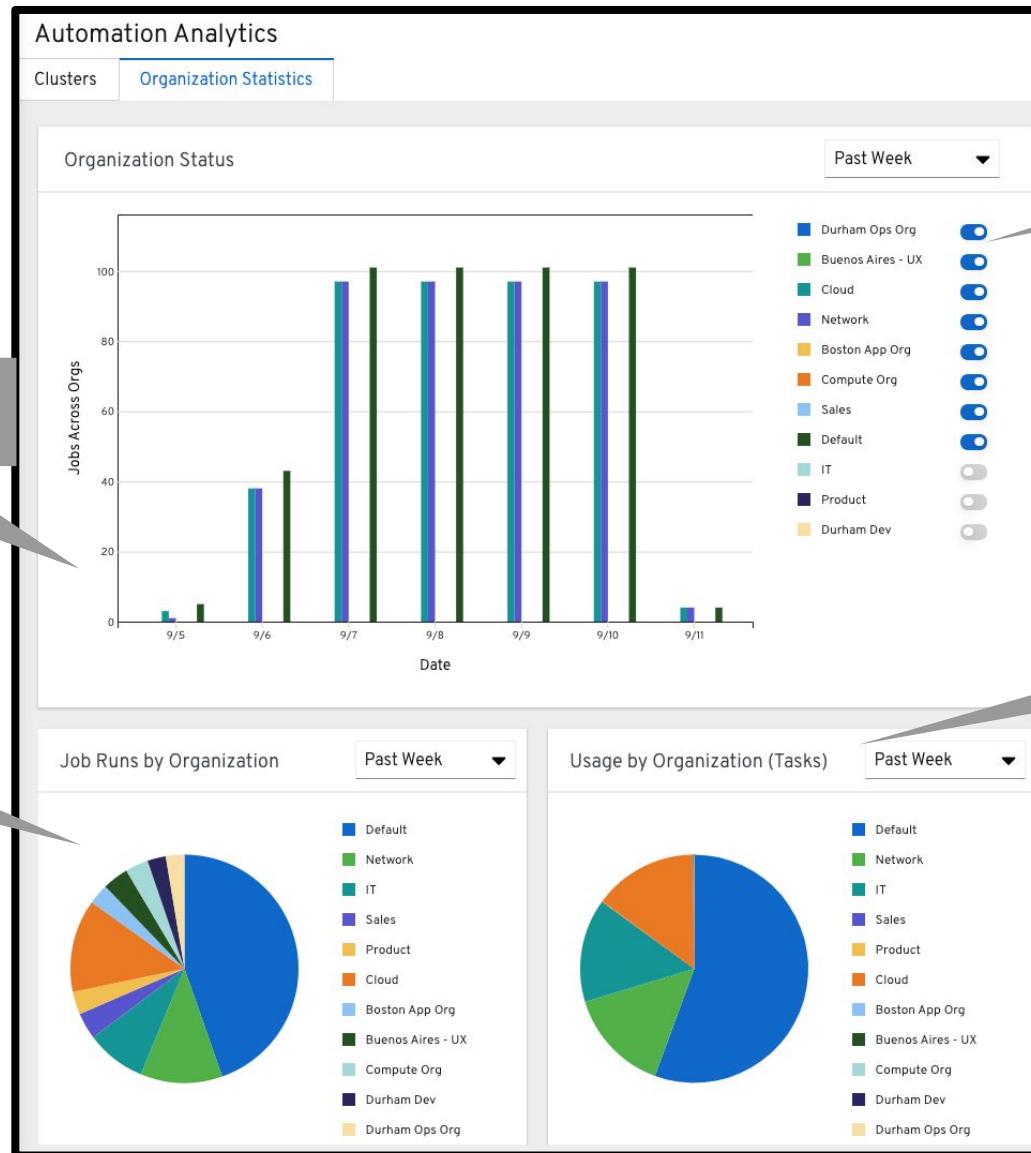
⚠ Cluster Error: No data from madrid.tower.example.com in 1 days

Notifications last updated 2019-09-11 07:42:12 UTC

Organizational statistics

Job Status by Organization

Job Runs by Organization



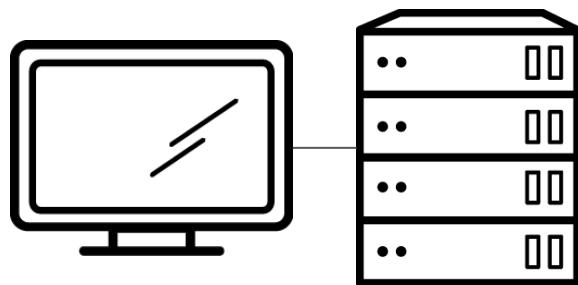
Filter by Organization

Usage by Organization

Dashboard comparison

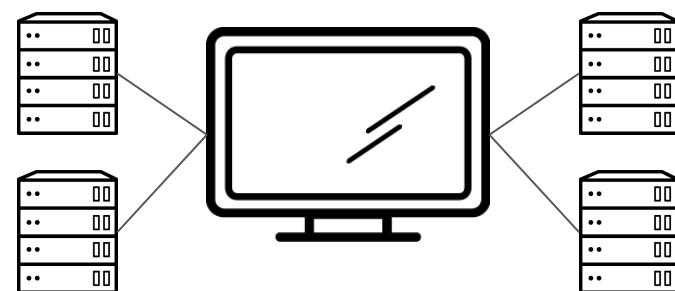
Ansible Tower

- Recent job templates
- No module data
- One cluster



Automation Analytics

- Top job templates
- Top modules
- All clusters
- Filter by cluster



Ansible Content Collections

Simplified and consistent content delivery

Provides quick benefit by lowering barriers to automation

Streamlines tech partners providing direct-to-user automation

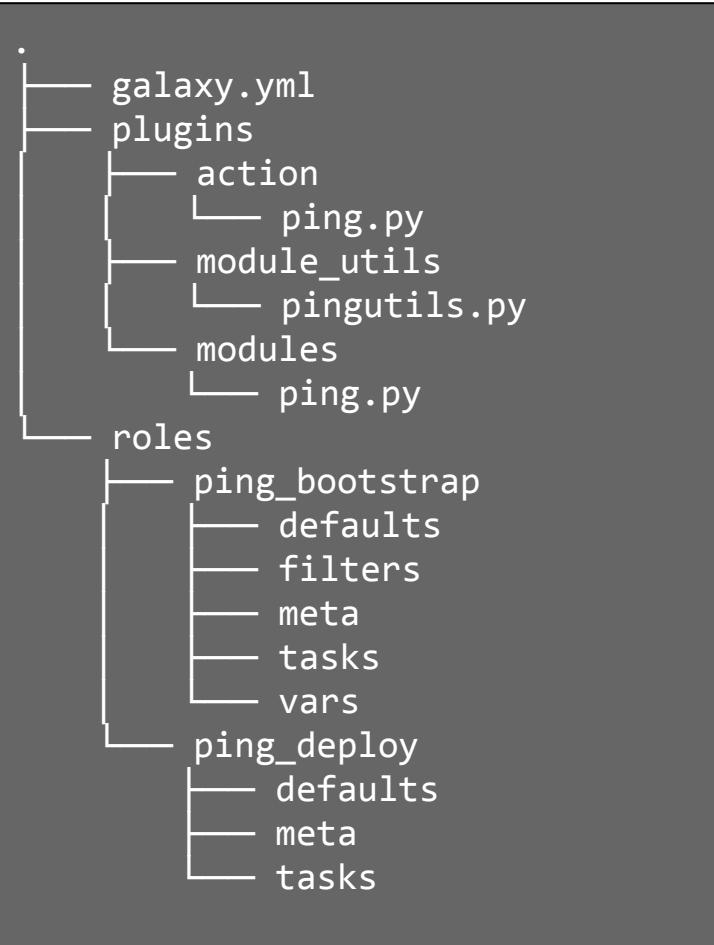
Simplifies internal collaboration, distribution, versioning

Ability to distribute, share and consume content at your own pace



Ansible Content Collection example

Directory Layout



In a playbook

```
hosts: somehosts
collections:
  - custom.pinger
  - redhat.open_ping

tasks:
  - custom.pinger.ping:

  - ansible.builtin.ping: # use only the ping packaged in core

  - ansible.legacy.ping: # use core or library/etc)/ping.py
    when: thing | custom.pinger.filter == 42

  - ping: # searches collections "path" otherwise...
    # still works, == ansible.legacy.ping:
```

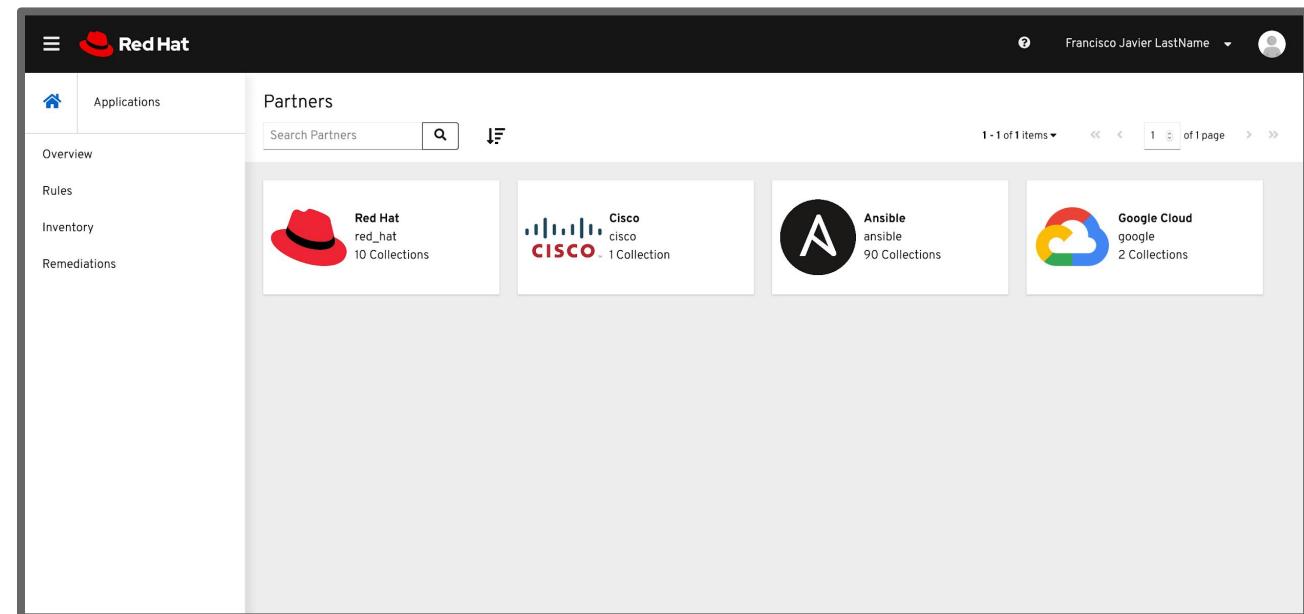
Automation Hub

Discover, publish, and manage Collections

Quickly discover available Red Hat and certified content through Collections.

Manage and test your organization's view of available content.*

Publish your locally available automation via on-premise.*



Next steps:

Get started

ansible.com/get-started

ansible.com/tower-trial

Workshops, training & services

ansible.com/workshops

[Red Hat Training](#)

[Red Hat Services: Automation Adoption Journey](#)

Join the community

ansible.com/community

Share your story

[Follow us @Ansible](#)

[Friend us on Facebook](#)



Red Hat
Ansible Automation
Platform

USE CASE:
WINDOWS AUTOMATION

WINDOWS AUTOMATION

90+

Windows
Modules

1,300+

Powershell DSC
resources

ansible.com/windows

Thank you



linkedin.com/company/red-hat



youtube.com/AnsibleAutomation



facebook.com/ansibleautomation



twitter.com/ansible



github.com/ansible