



Red Hat

Ansible Automation
Platform

Automation for all

Ansible technical introduction and overview



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

Teams are automating...



Lines Of Business



Network



Security



Operations

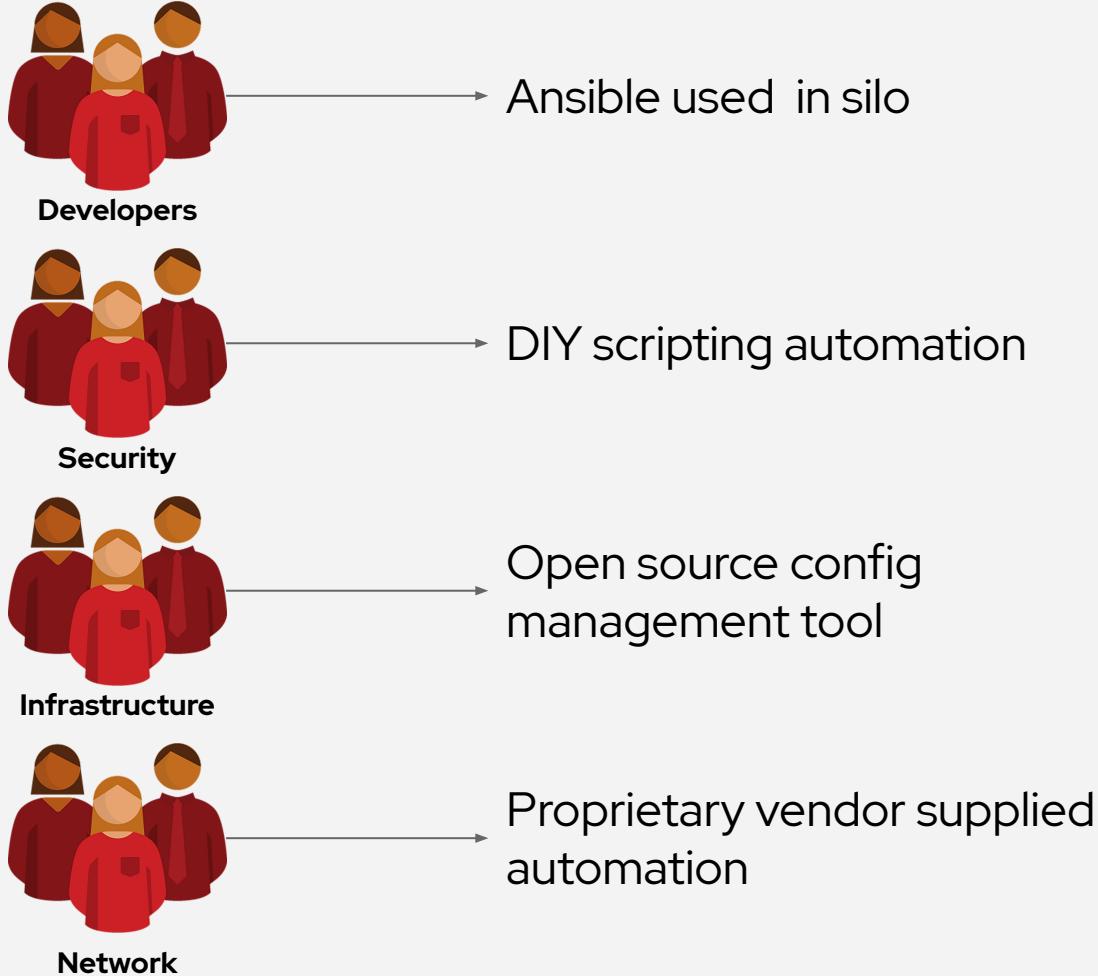


Developers



Infrastructure

Ad-hoc Automation is happening in silos



Is organic
automation enough?

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

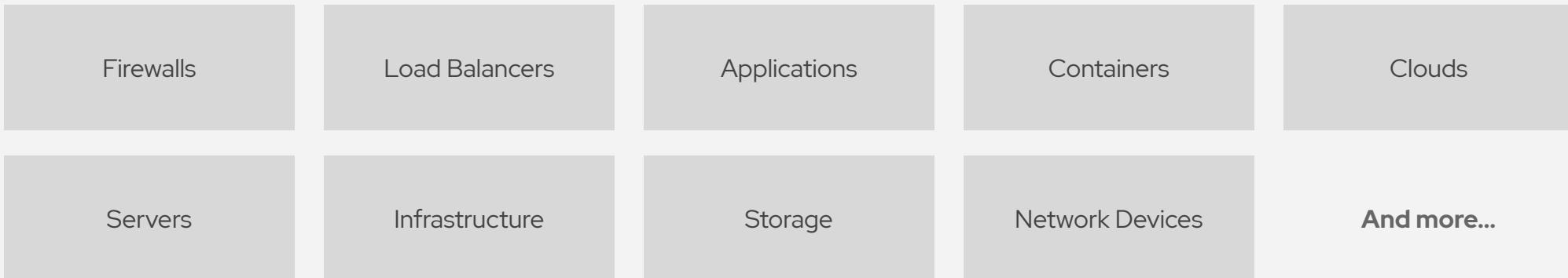
What can I do using Ansible?

Automate the deployment and management of your entire IT footprint.

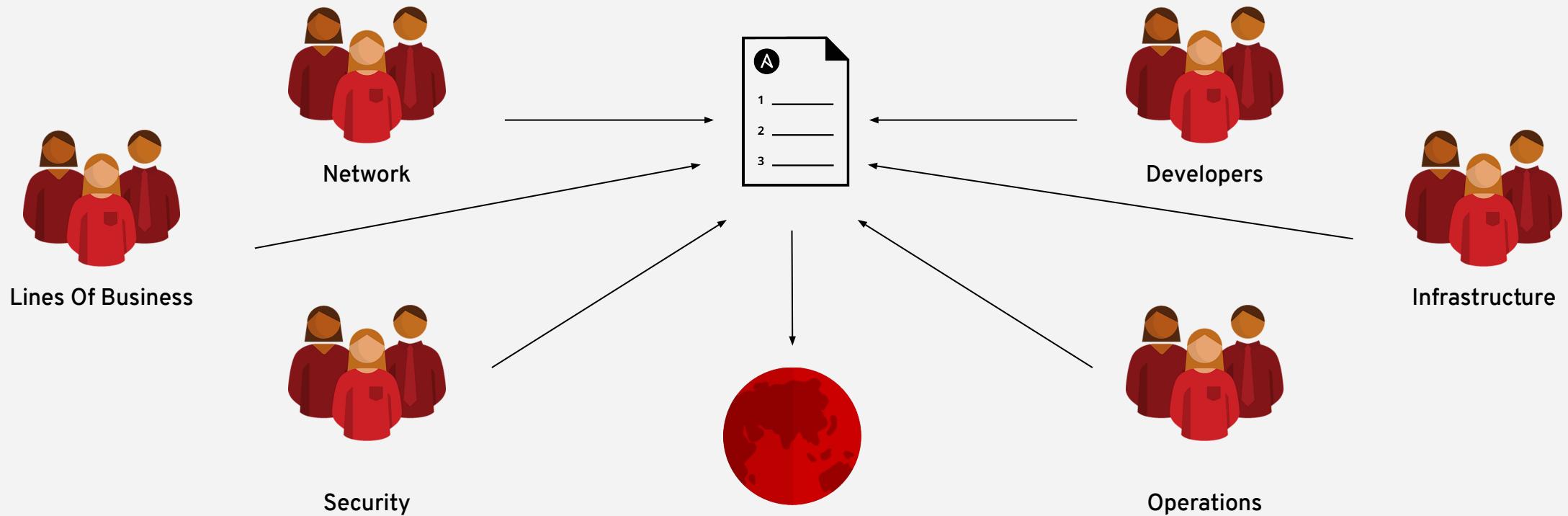
Do this...



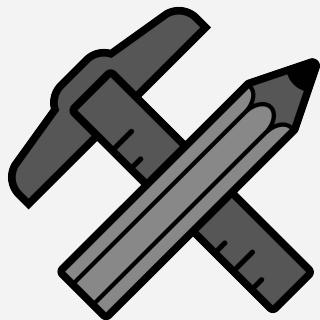
On these...



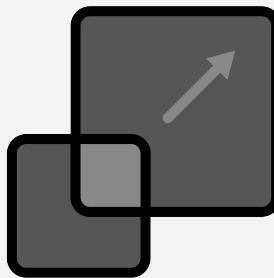
When automation crosses teams, you need an automation platform



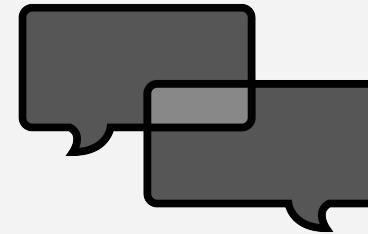
A platform can help you:



Create

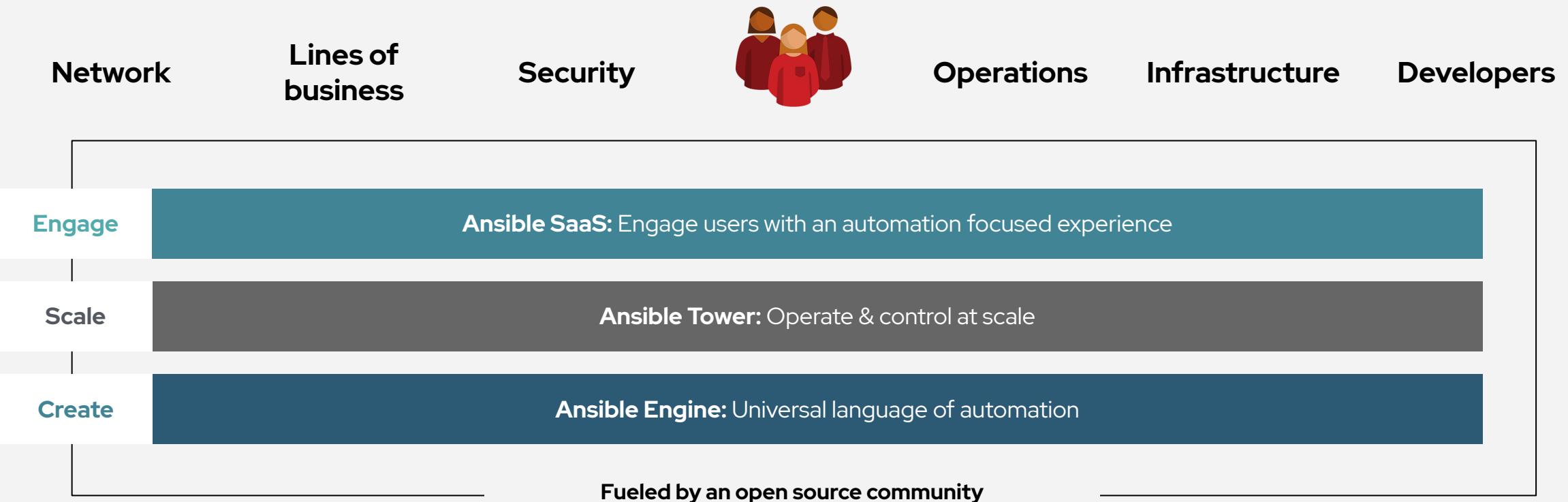


Scale



Engage

Red Hat Ansible Automation Platform



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

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





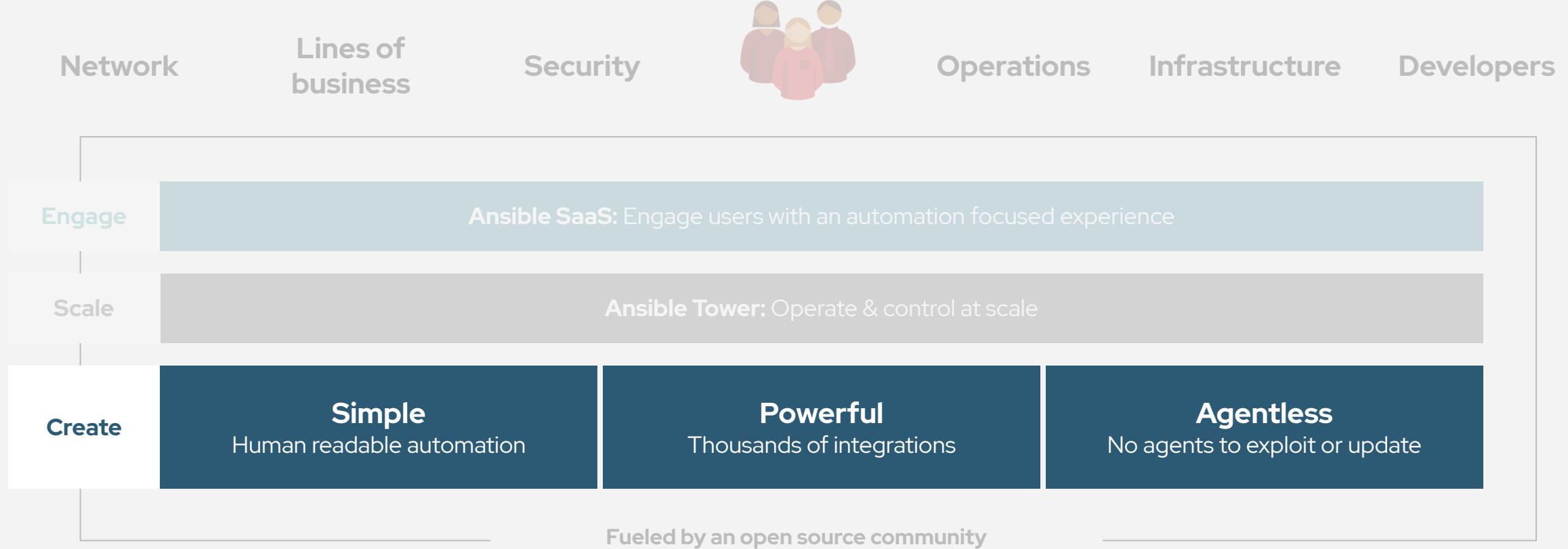
Red Hat

Ansible Automation Platform

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.

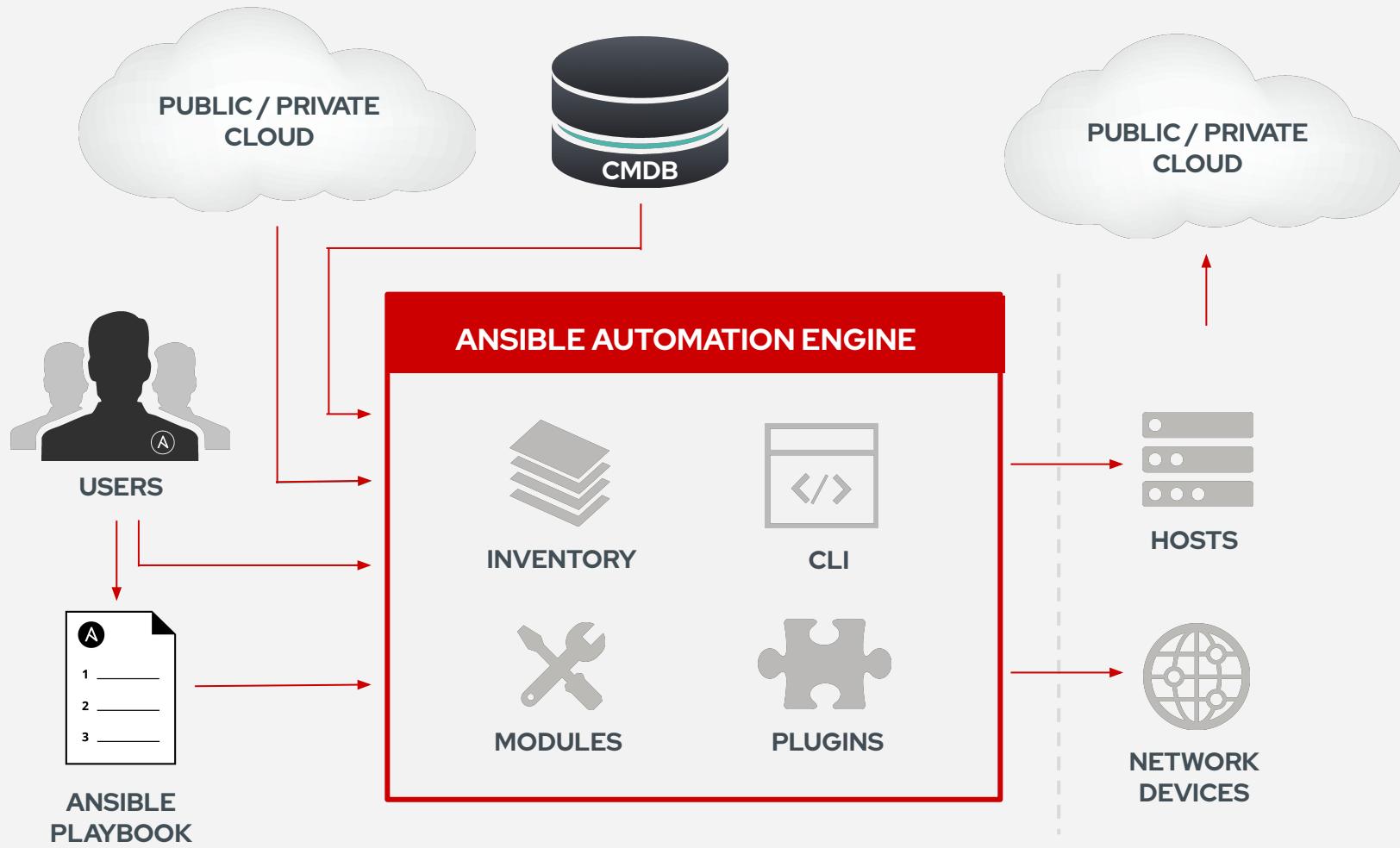
Dynamic inventories

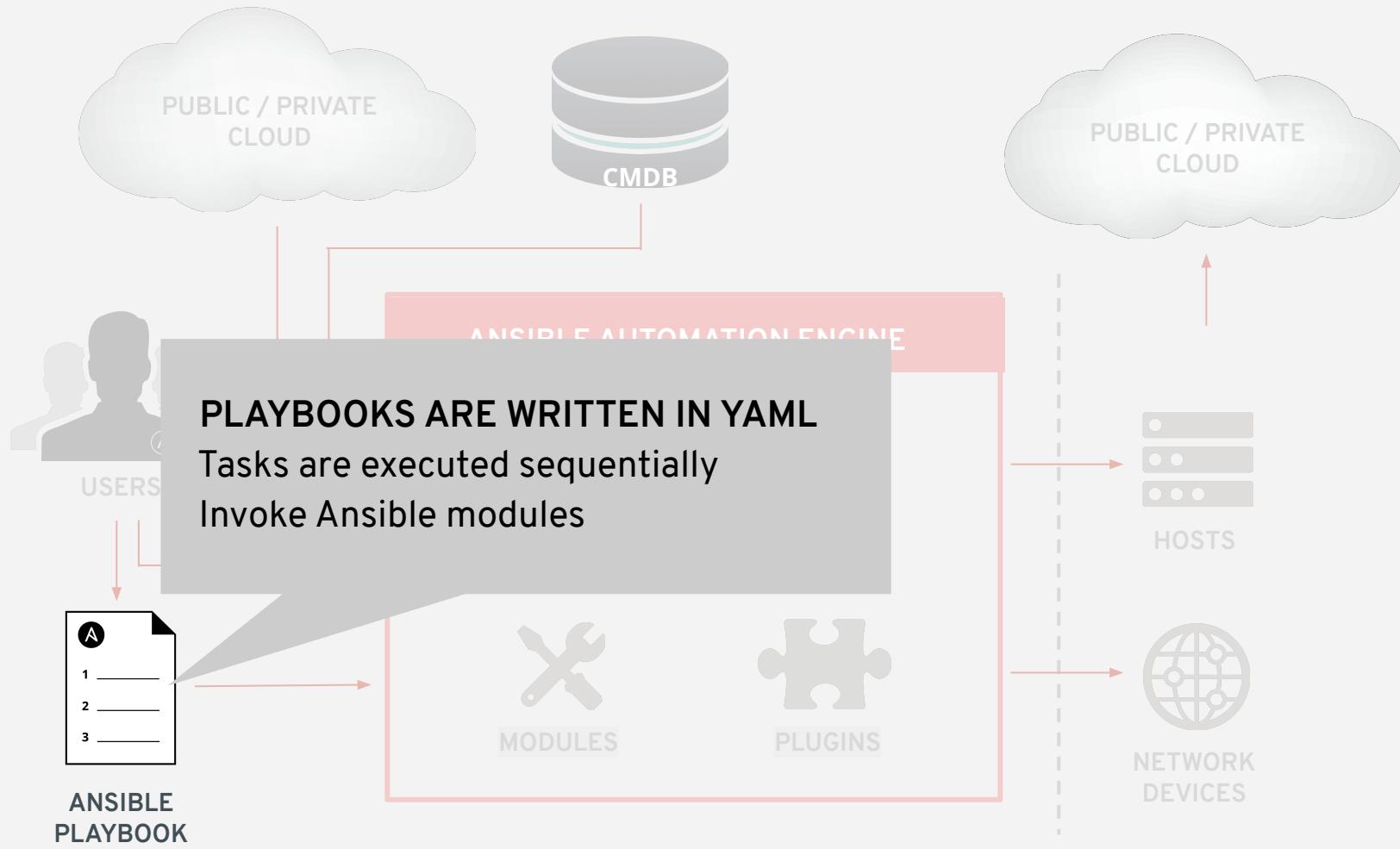
Capture all the servers 100% of the time, regardless of infrastructure, location, etc.

Orchestration plays well with others

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

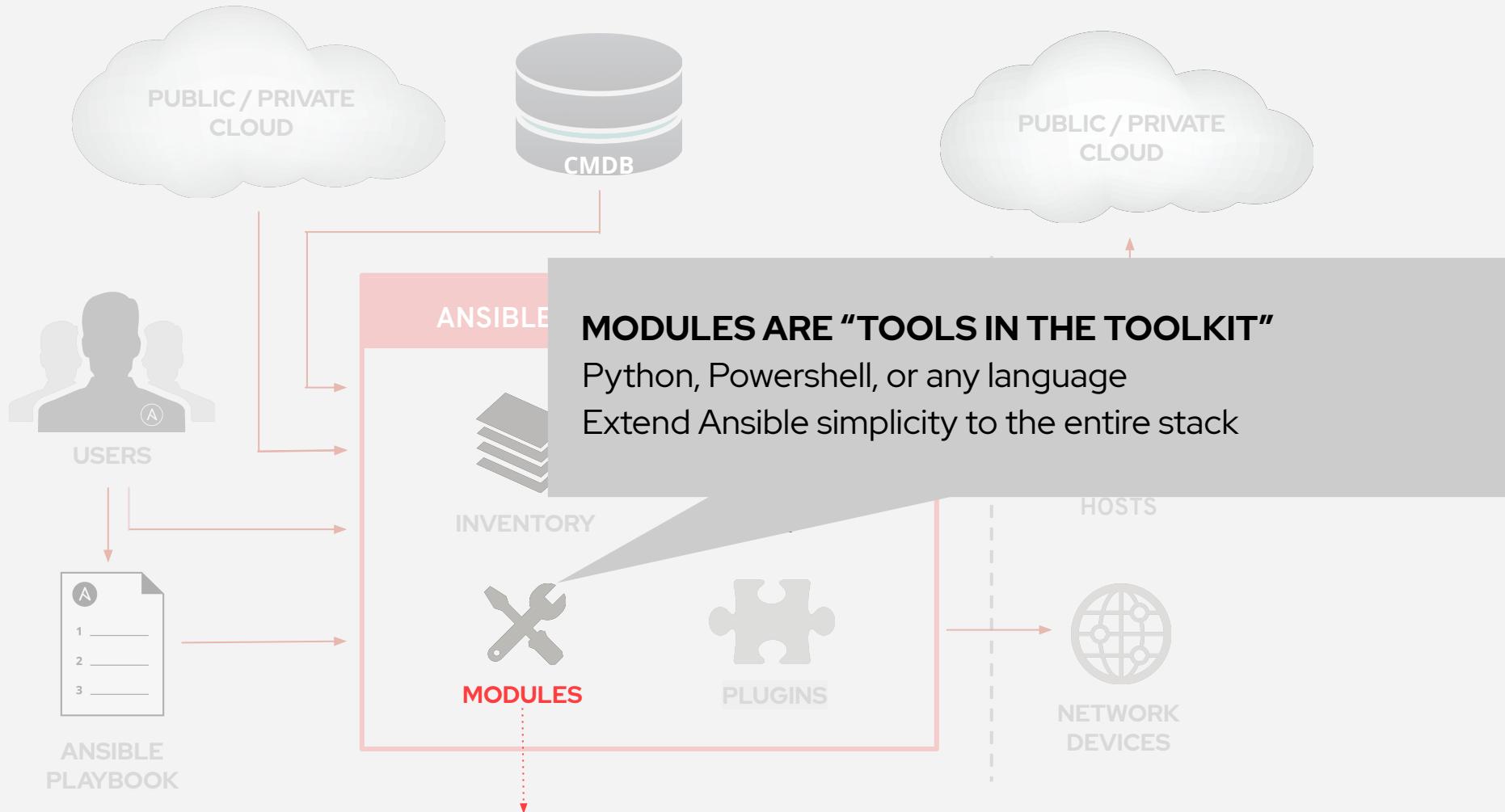




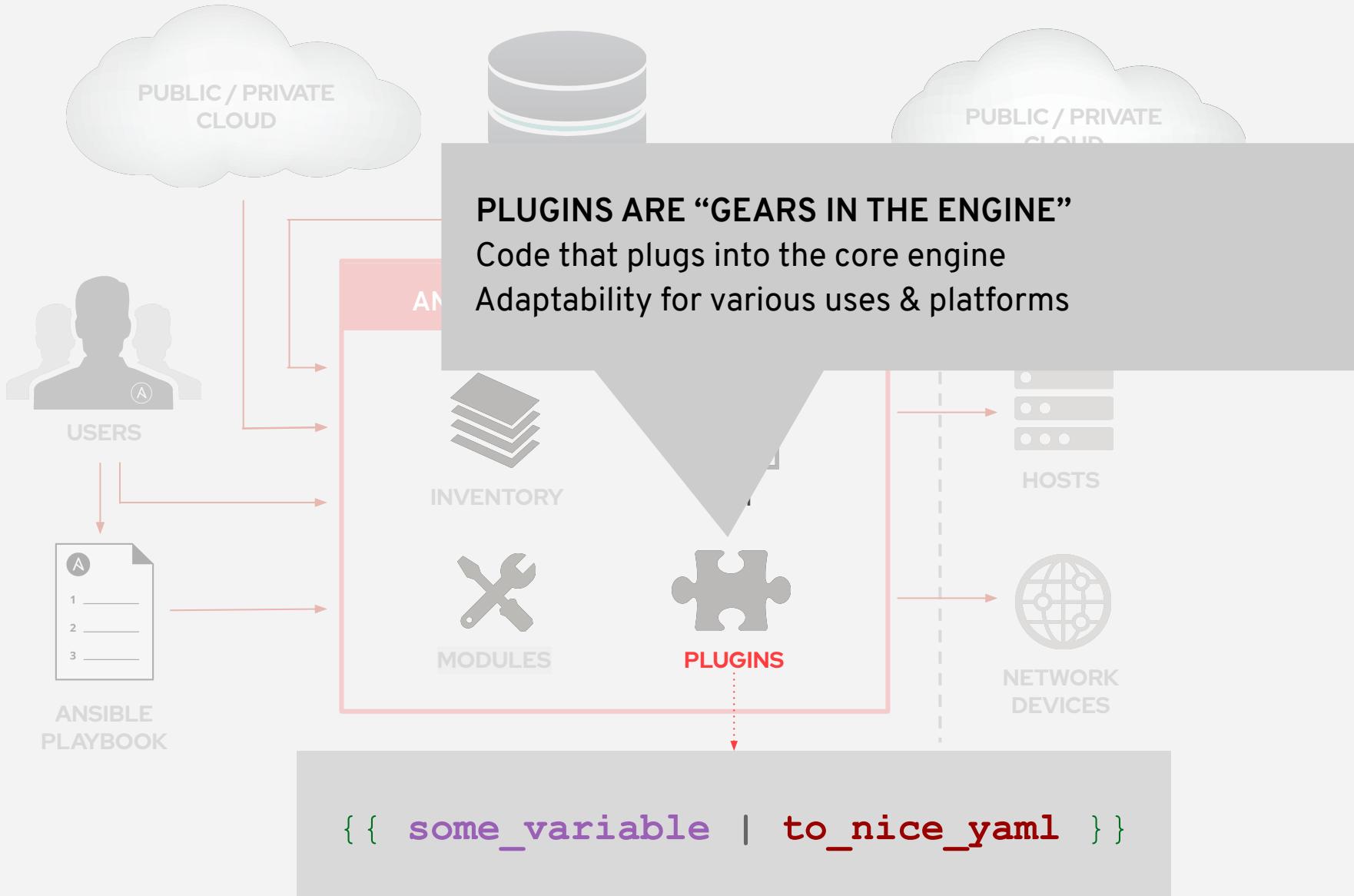


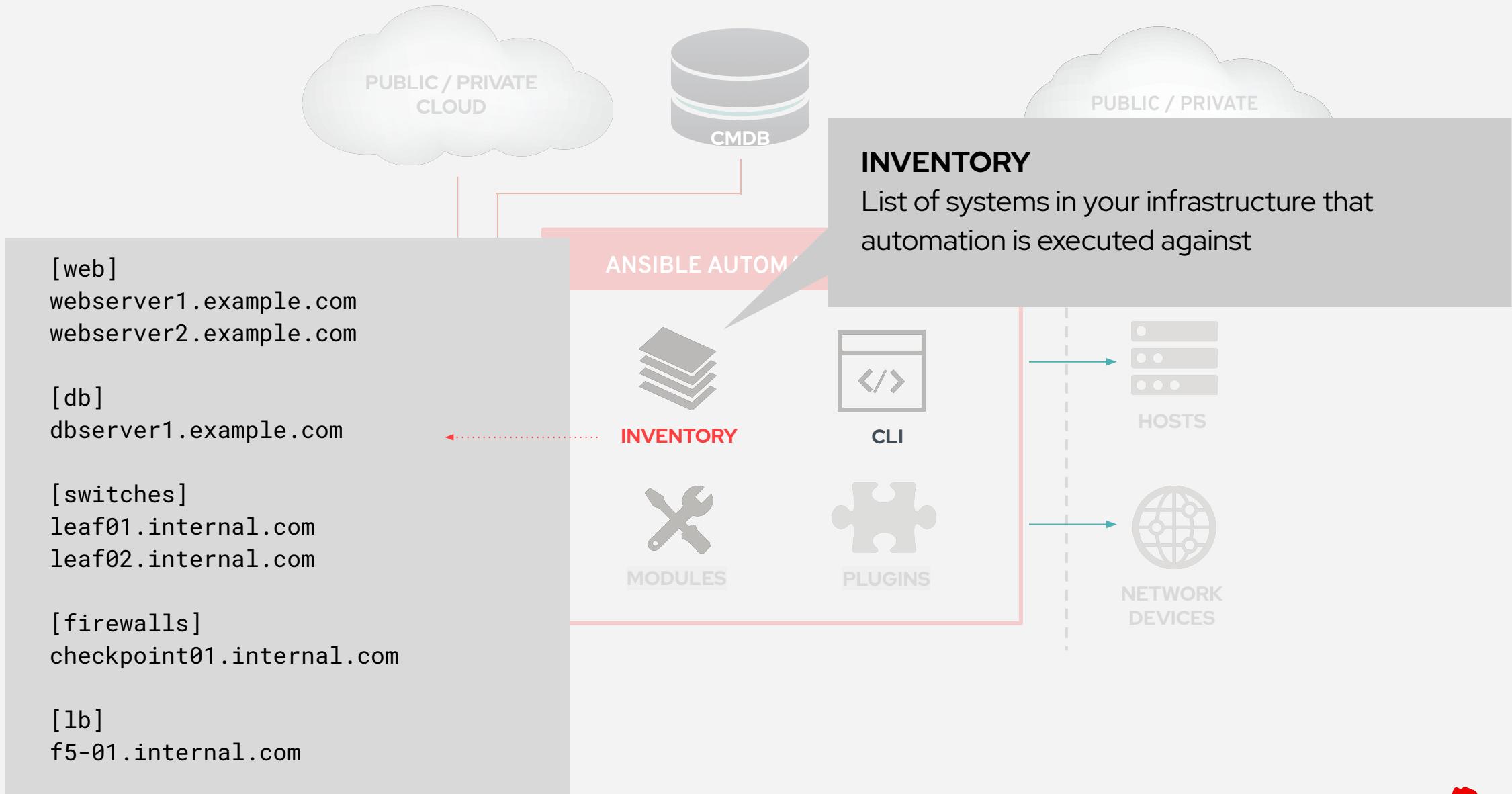
```
---
```

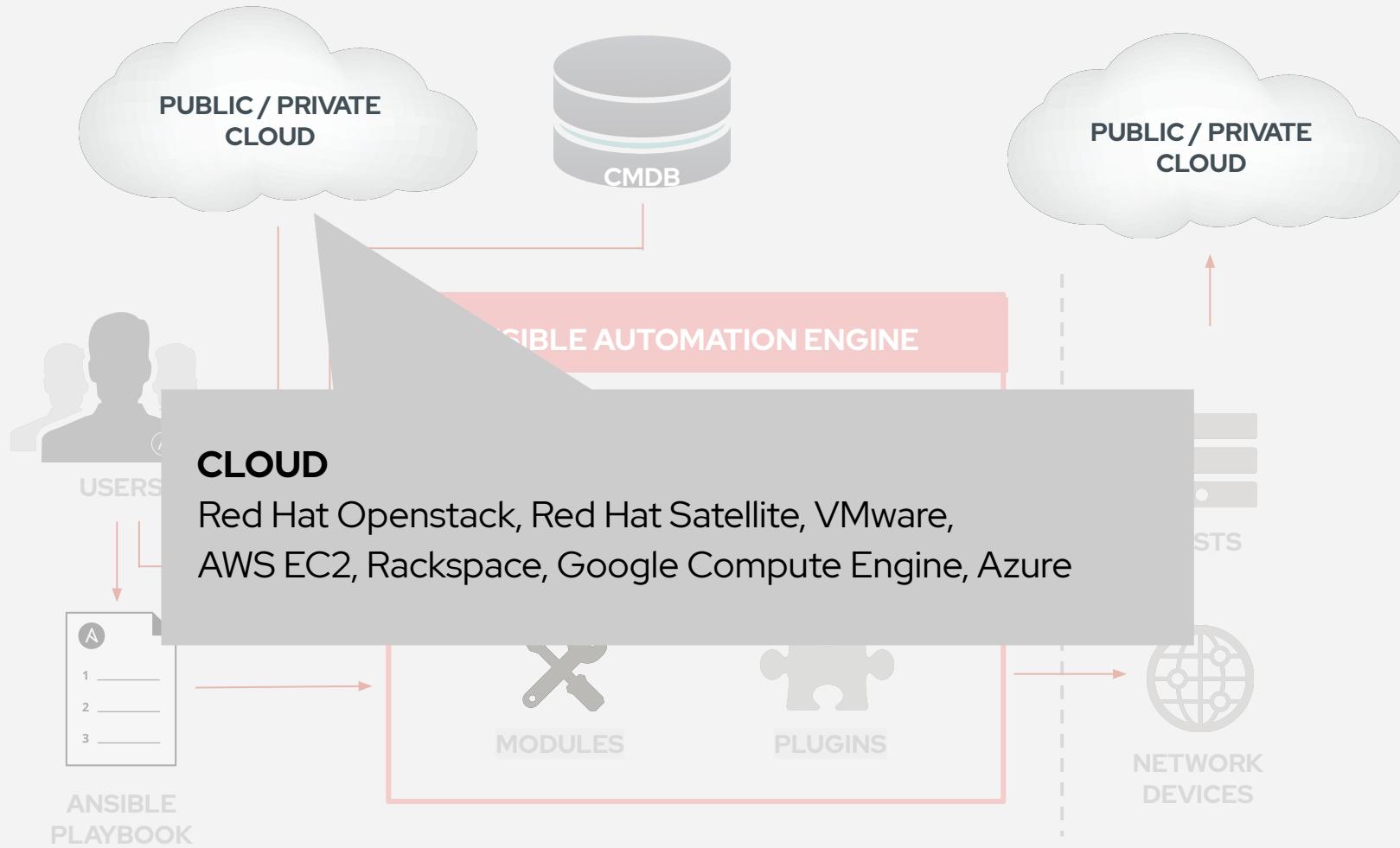
- **name: install and start apache**
 - hosts:** web
 - become:** yes
 - vars:**
 - http_port:** 80
- tasks:**
 - **name: httpd package is present**
 - yum:**
 - name:** httpd
 - state:** latest
 - **name: latest index.html file is present**
 - template:**
 - src:** files/index.html
 - dest:** /var/www/html/
 - **name: httpd is started**
 - service:**
 - name:** httpd
 - state:** started

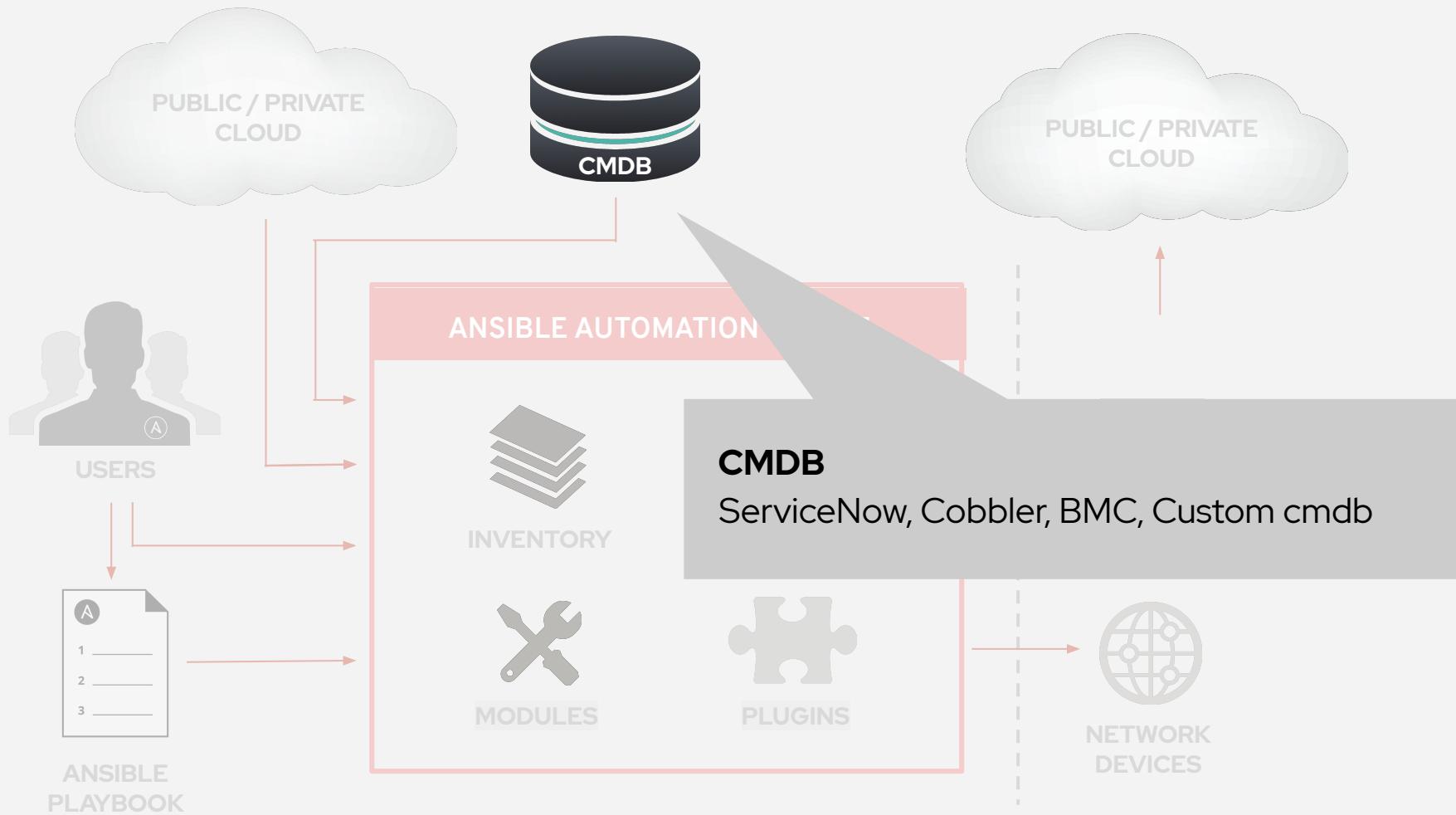


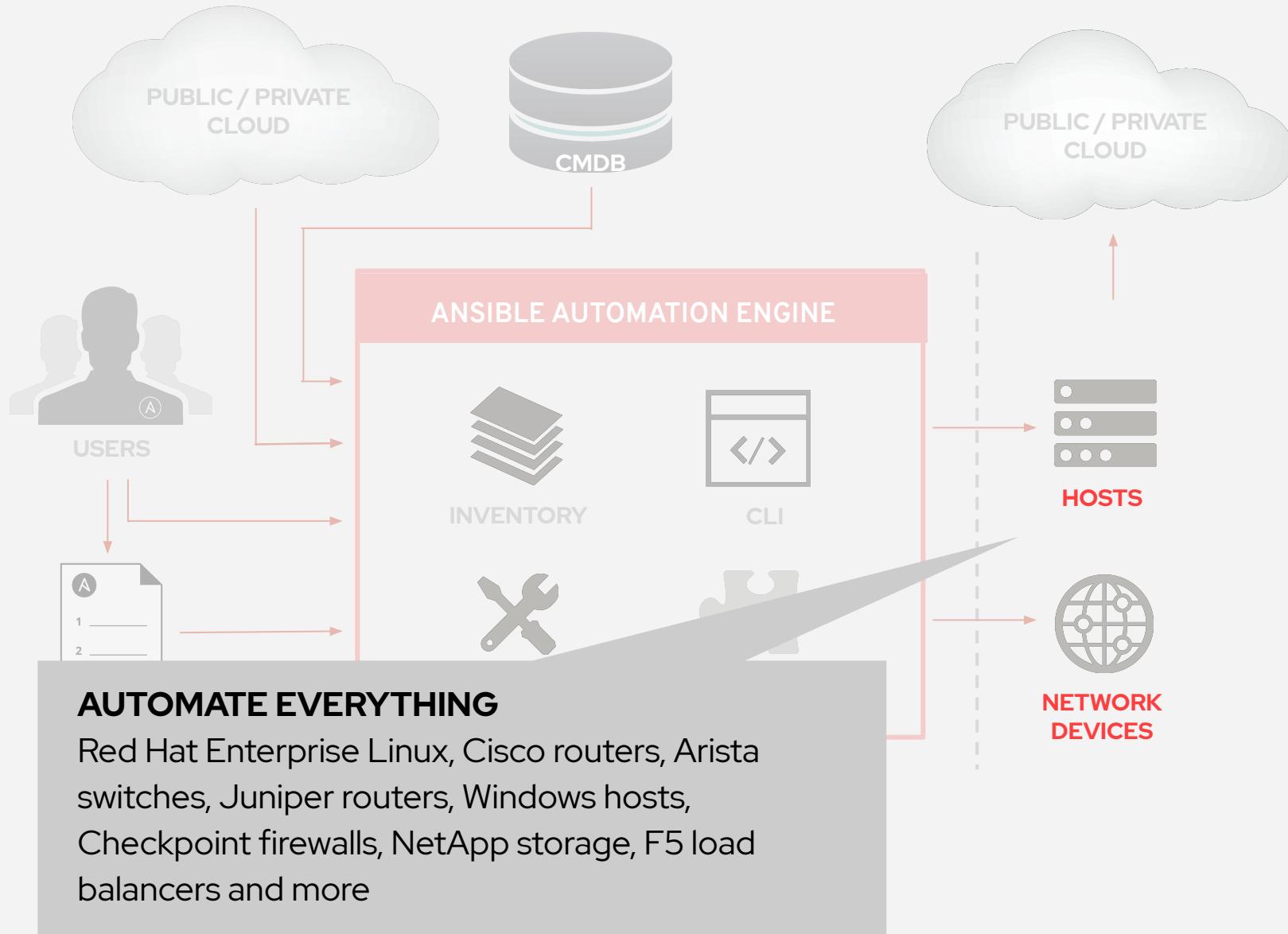
```
- name: latest index.html file is present
  template:
    src: files/index.html
    dest: /var/www/html/
```











Ansible Playbook examples:

GITHUB EXAMPLES

github.com/ansible/ansible-examples
github.com/ansible/workshops

LAMP + HAProxy + NAGIOS

bit.ly/lamp_haproxy

WINDOWS

bit.ly/ansible_windows

COMPLIANCE

bit.ly/ansible_compliance

NETWORK

github.com/network-automation

SECURITY

github.com/ansible-security/



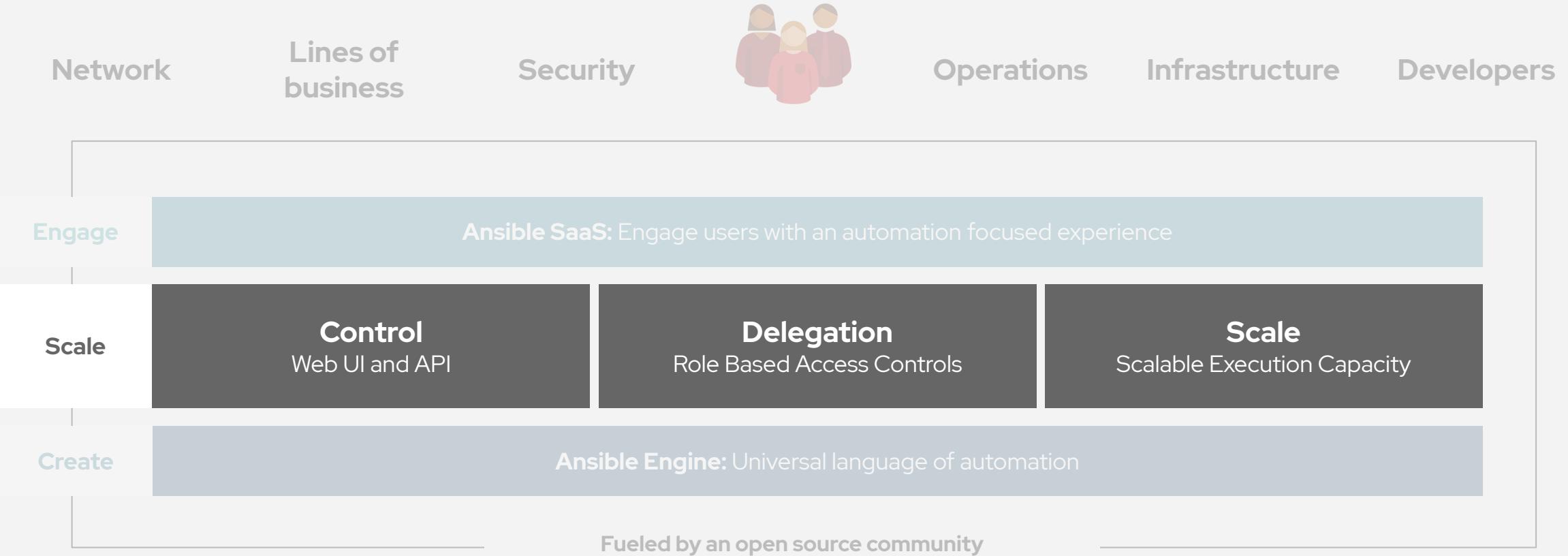
Red Hat

Ansible Automation Platform

Red Hat Ansible Tower:
Operate and
control at scale



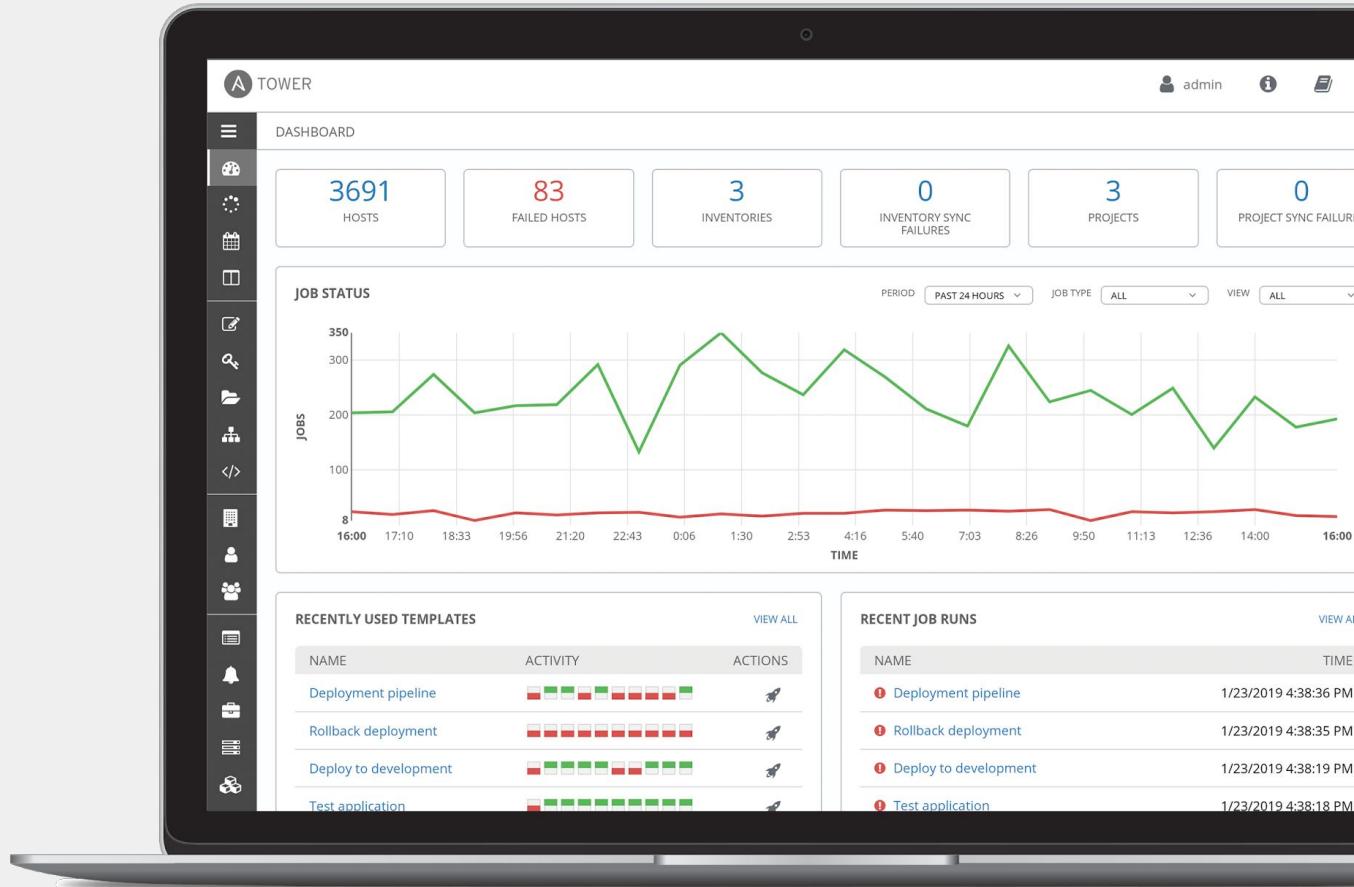
Red Hat Ansible Automation Platform



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.

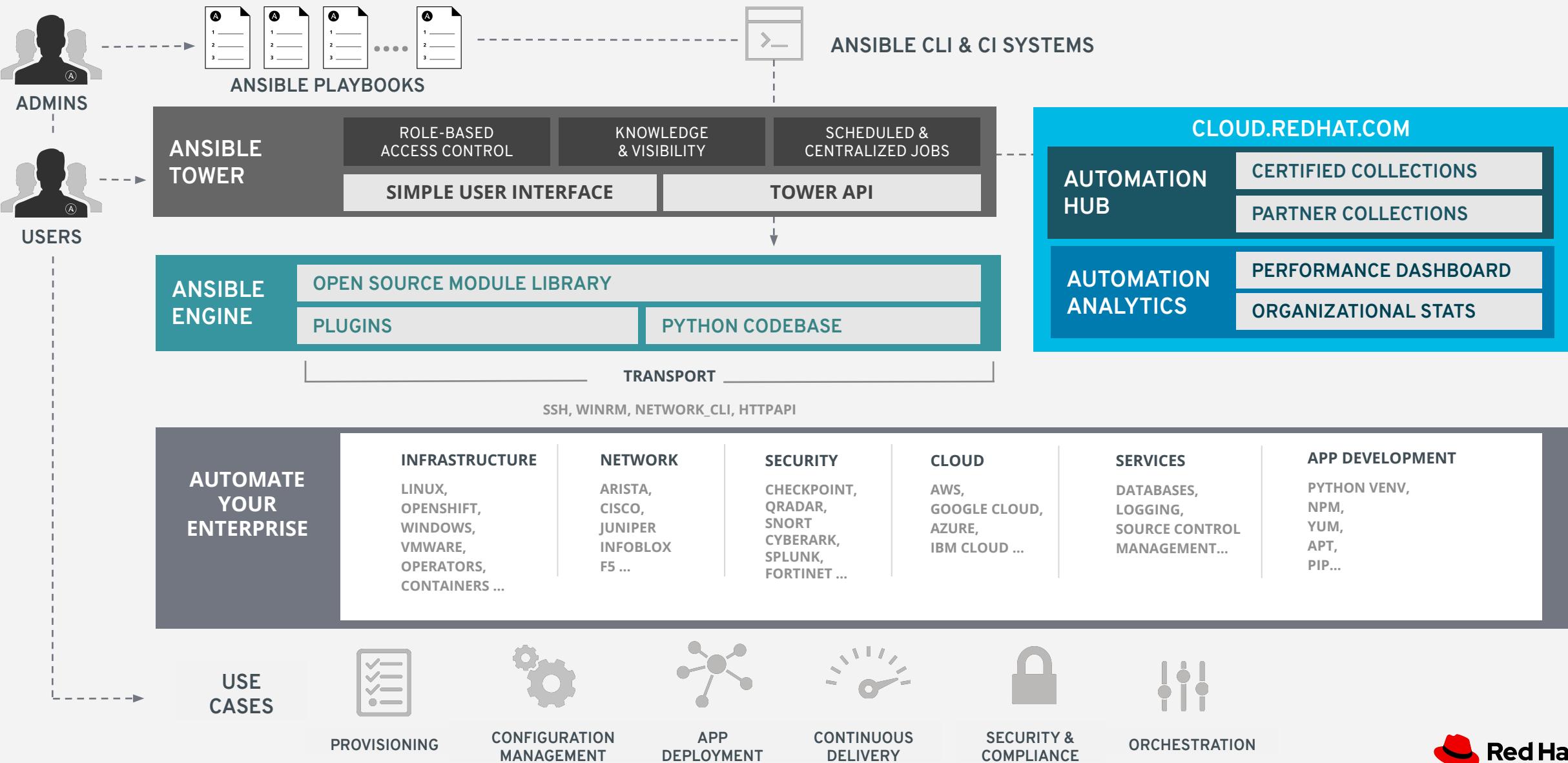
Centralized logging

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



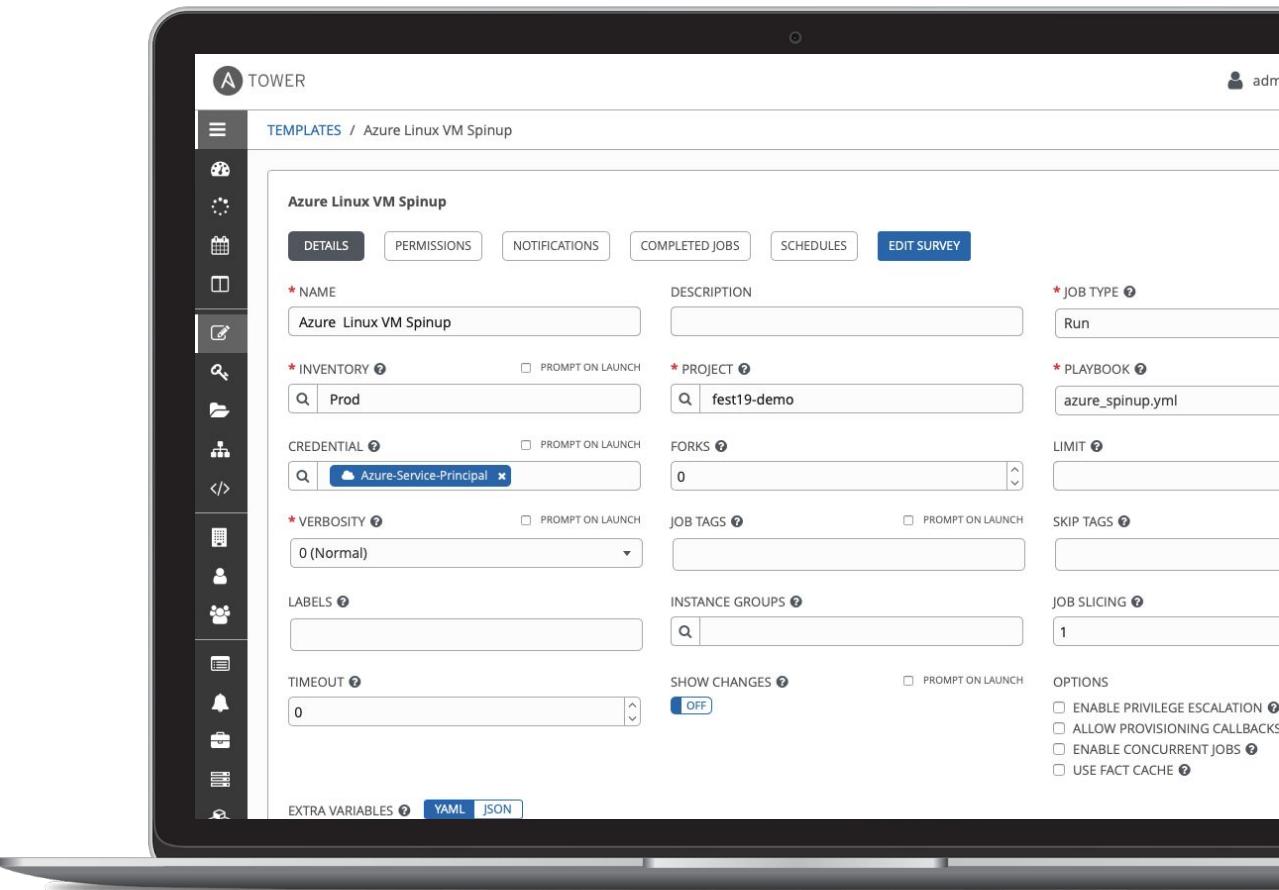
Job Templates

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:

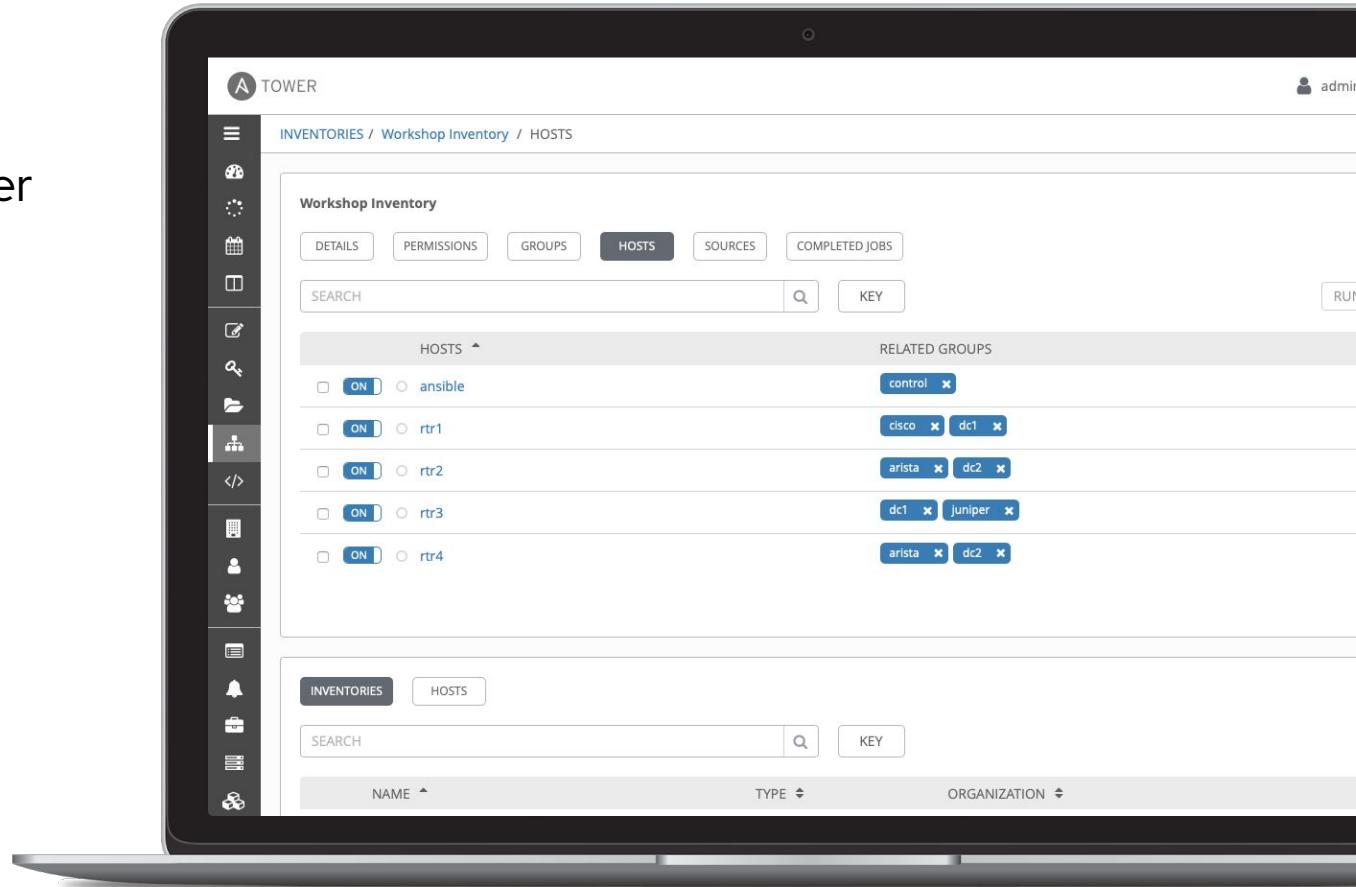
- An **Inventory** to run the job against
- A **Credential** to login to devices.
- A **Project** which contains Ansible Playbooks



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

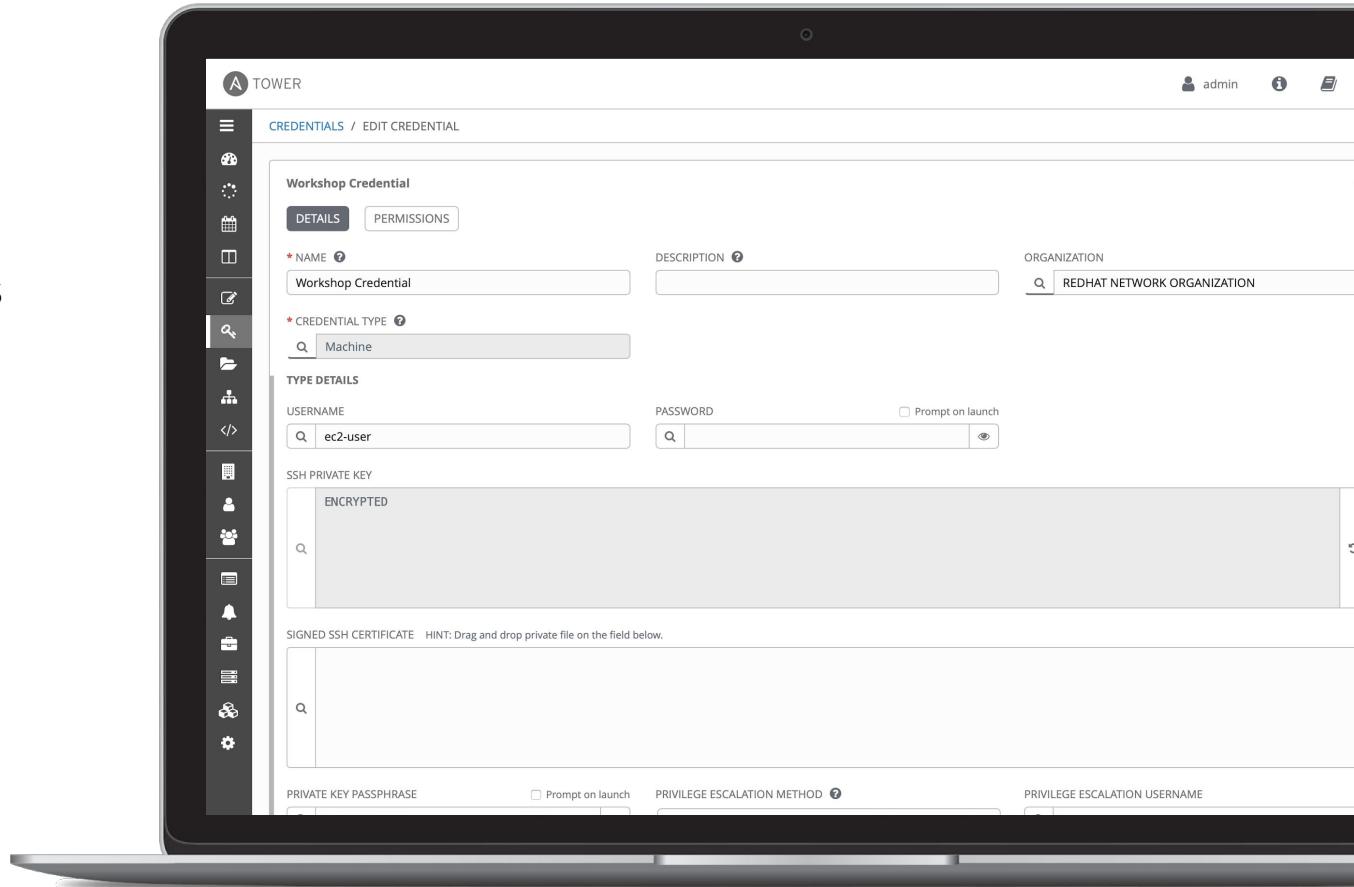


Credentials

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

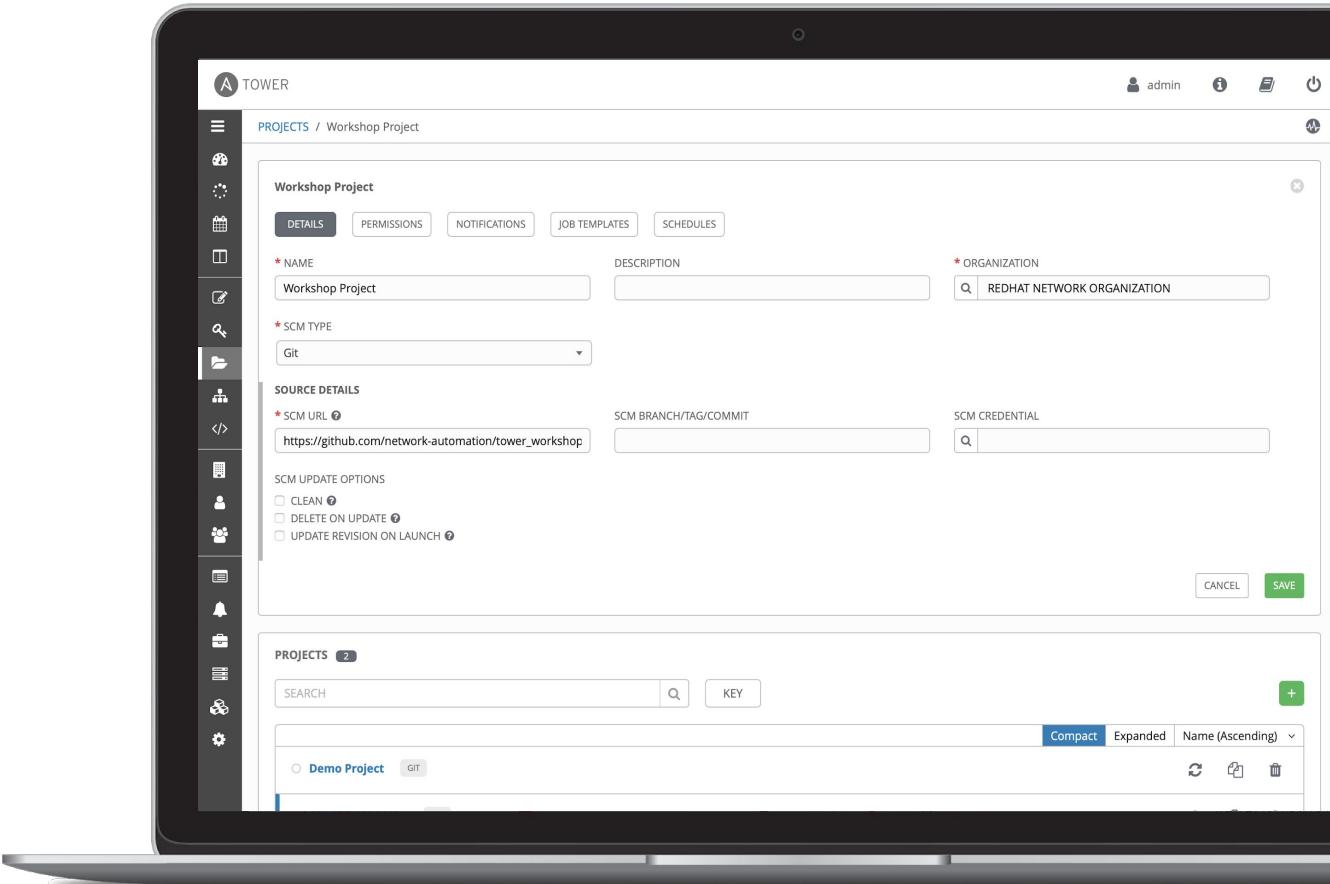
Centralized management of various credentials allows end users to leverage a secret without ever exposing that secret to them.



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

A TOWER REST API

admin Log out ? ← ↗

REST API / Version 2

Version 2^②

OPTIONS GET ▾

GET /api/v2/

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
X-API-Node: localhost
X-API-Time: 0.019s

```
{  
    "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/",  
}
```

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 permissions on Red Hat Enterprise Linux web servers. The top navigation bar includes 'TOWER' (with a logo), 'admin', and an information icon. Below the navigation is a sidebar with 'VIEWS' and several resource categories: 'Dashboard', 'Jobs', 'Schedules', 'My View', 'TEMPLATES' (selected), 'Credentials', 'Projects', 'Inventories', 'Inventory Scans', 'ACCESS', and 'Organizations'. The main content area has a title 'CONFIGURE RED HAT ENTERPRISE LINUX WEB SERVERS | ADD USERS / TEAMS'. A step-by-step guide says '1 Please select Users / Teams from the lists below.' Two buttons are highlighted with red boxes: 'USERS' (selected) and 'TEAMS'. Below these buttons is a search bar with a magnifying glass icon and a 'KEY' button. The main table lists users with columns for '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). At the bottom right 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. At the top, there are several tabs: AZURE AD, GITHUB, GOOGLE OAUTH2, LDAP, RADIUS, SAML, and TACACS+. The 'TACACS+' tab is currently active and highlighted with a red border. Below the tabs, there are configuration fields for 'TACACS+ SERVER', 'TACACS+ PORT', 'TACACS+ SECRET', 'TACACS+ AUTH SESSION TIMEOUT', and 'TACACS+ AUTHENTICATION PROTOCOL'. A callout bubble points to the 'TACACS+' tab with the text: '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' section, there is a 'LOGGING AGGREGATOR' field containing 'log.eros.rhdemo.io'. Below it is a 'LOGGING AGGREGATOR USERNAME' field with 'ender'. To the right, there is a 'LOGGING AGGREGATOR PORT' field and a 'LOGGING AGGREGATOR PASSWORD/TOKEN' field with 'SHOW' and a redacted password. Further down are fields for 'LOG SYSTEM TRACKING FACTS INDIVIDUALLY' (set to 'OFF') and 'LOGGING AGGREGATOR LEVEL' (set to 'INFO'). On the far right, there is a 'ENABLE/DISABLE HTTPS CERTIFICATE' button. A large callout bubble points to the 'LOGGING AGGREGATOR TYPE' dropdown, which is currently set to 'splunk'. This dropdown is highlighted with a red border. The bubble contains the text: 'Multiple supported 3rd party external logging methods are easily integrated with Ansible Tower'.

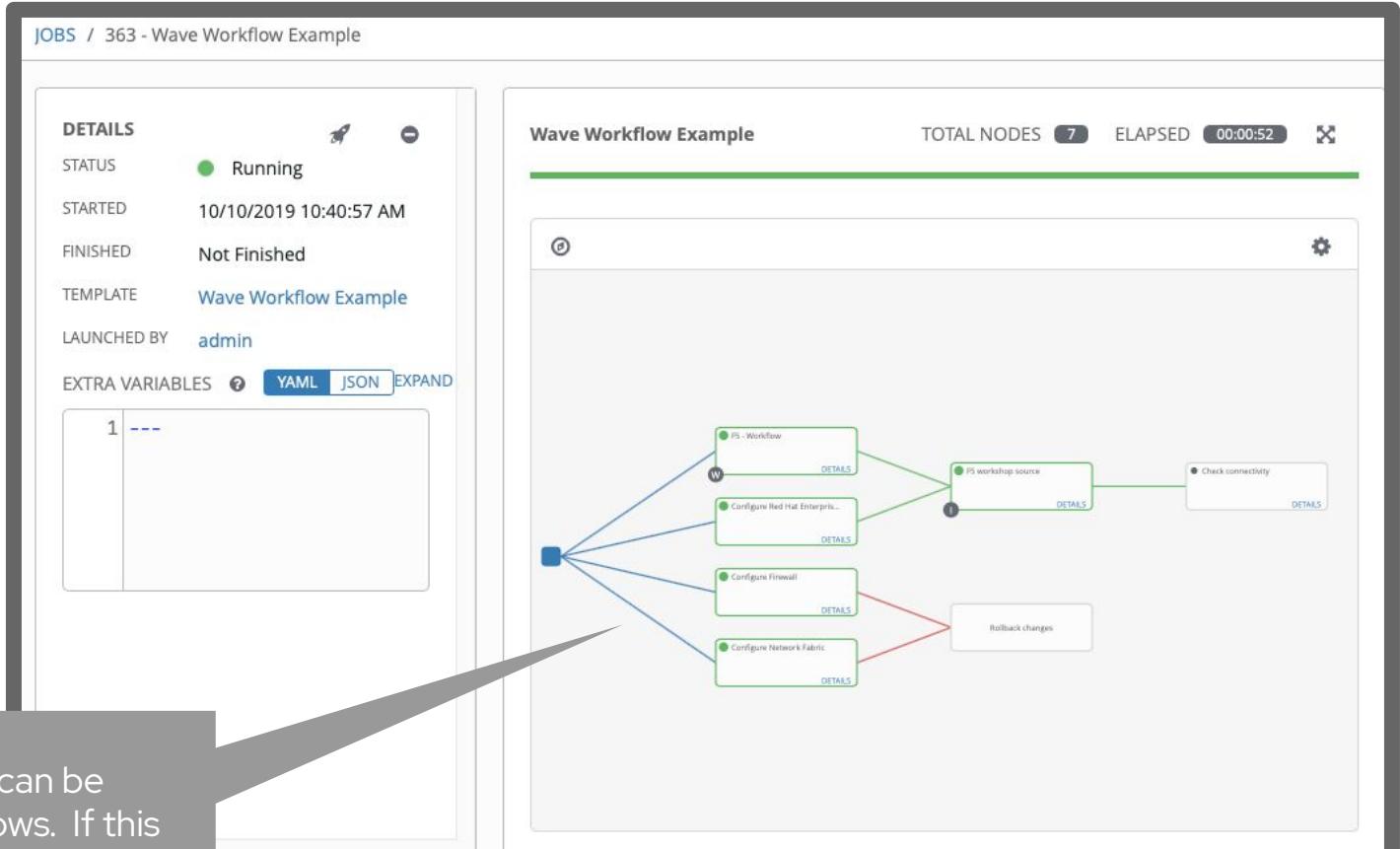
Workflows

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

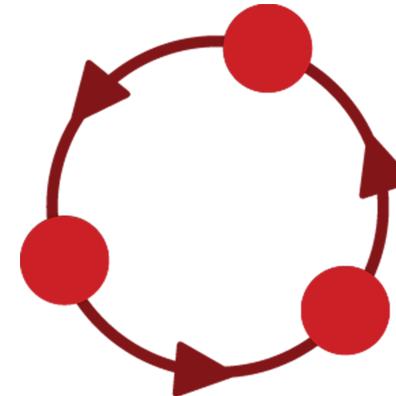
Conditional logic can be applied to workflows. If this job fails this next Job is run!



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.

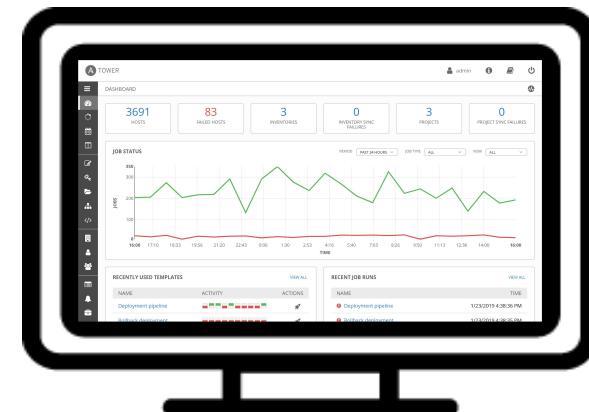


Scale

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

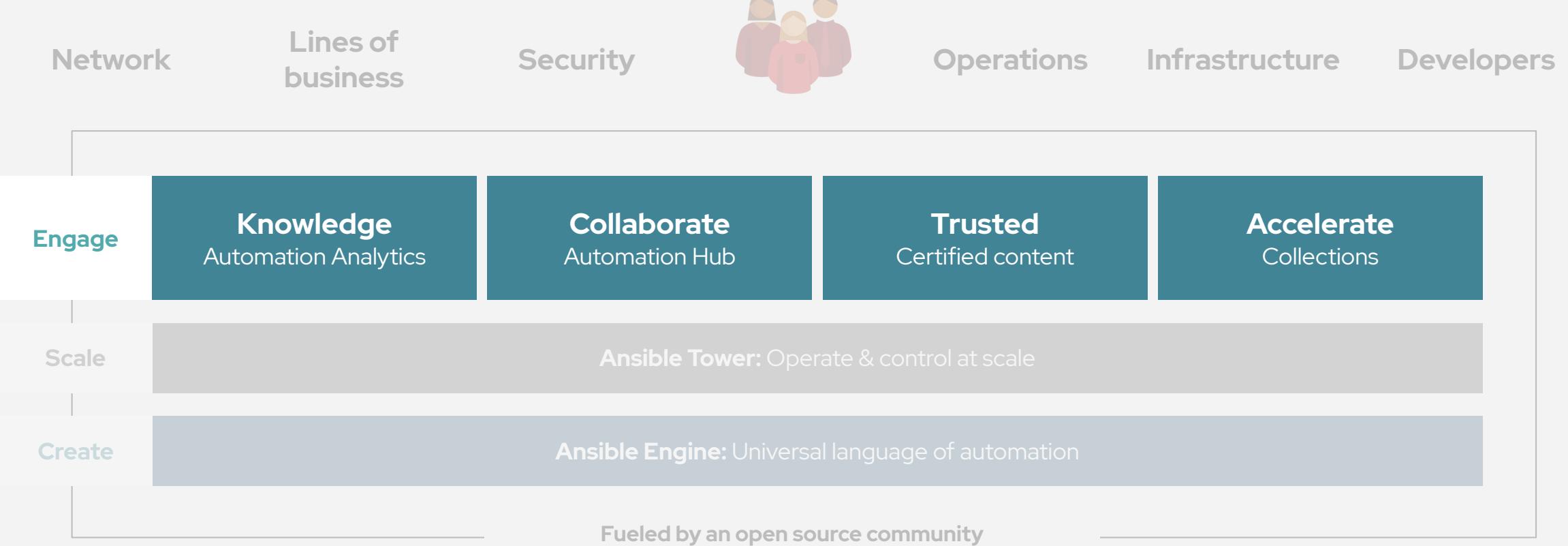
Ansible Automation Platform

CLOUD.REDHAT.COM

Engage users with
an automation
focused experience



Red Hat Ansible Automation Platform



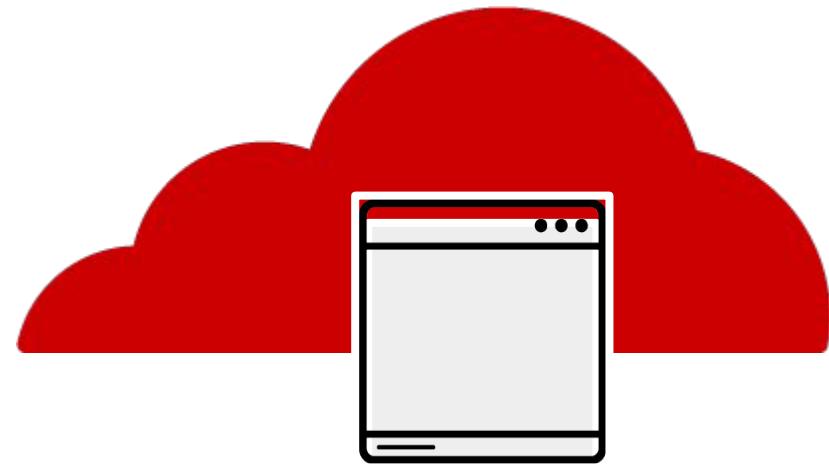
Automation Analytics: What is it?

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

Analytics for all Ansible Tower clusters for an organization

Includes:

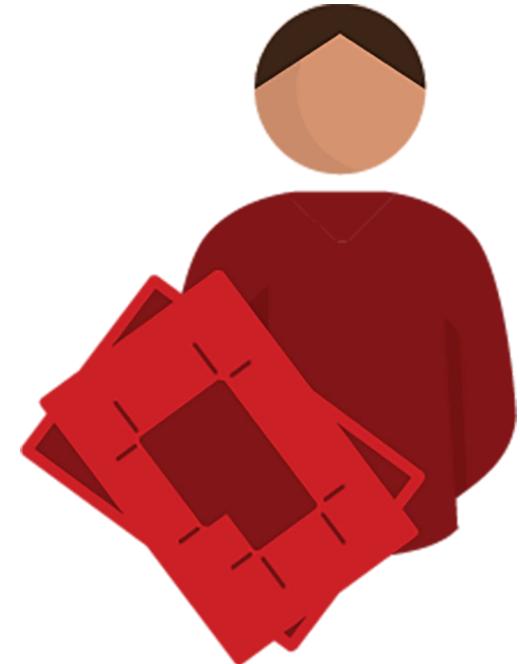
- visual dashboard
- health notifications
- organization statistics



Automation Analytics: What does it provide?

Enables an Automation Center of Excellence

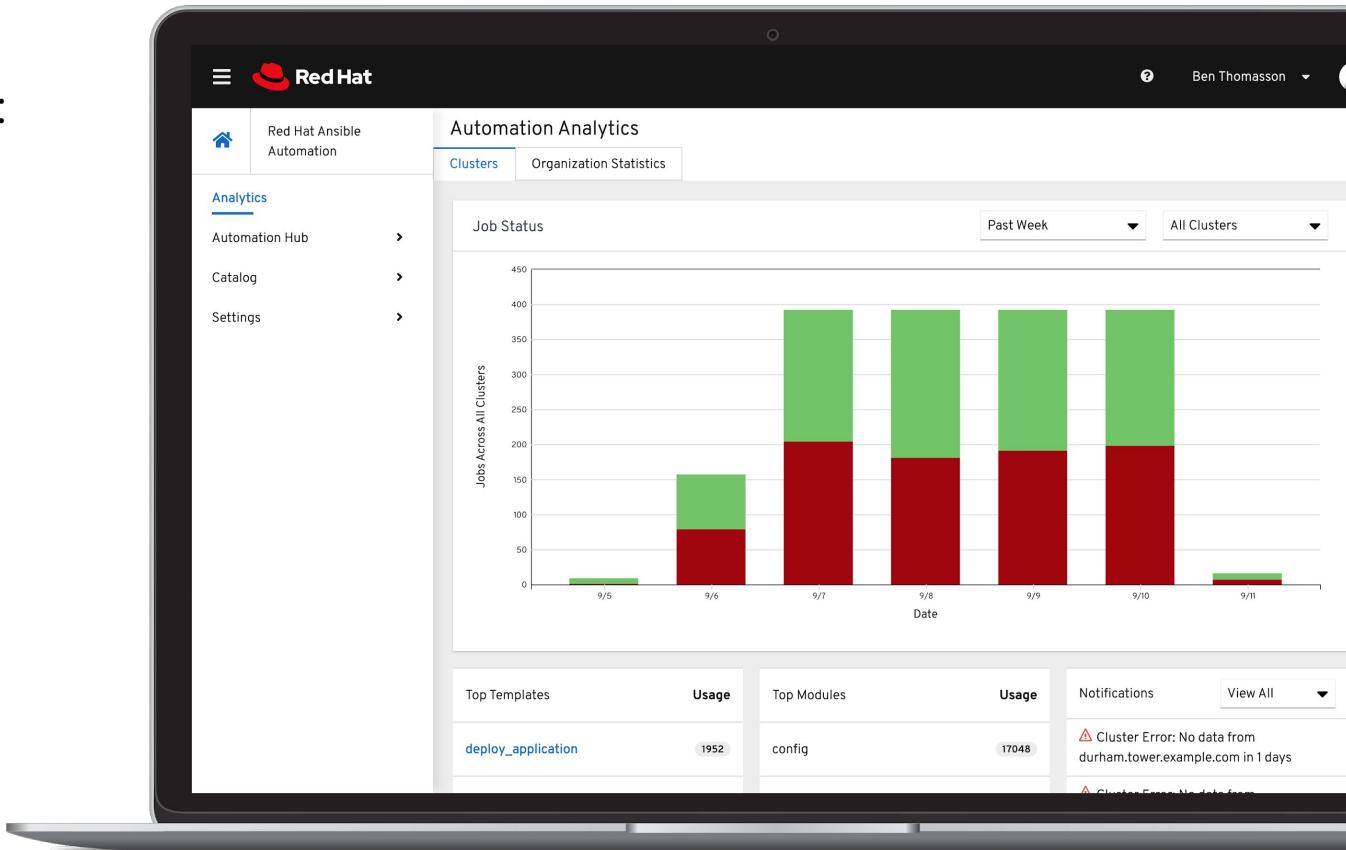
- View information about automation health, usage and performance across your enterprise.
- Gain information about automation in your enterprise:
 - Which organizations are using the most automation?
 - Utilization rates
 - Enterprise-wide success and failure rates for automation



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

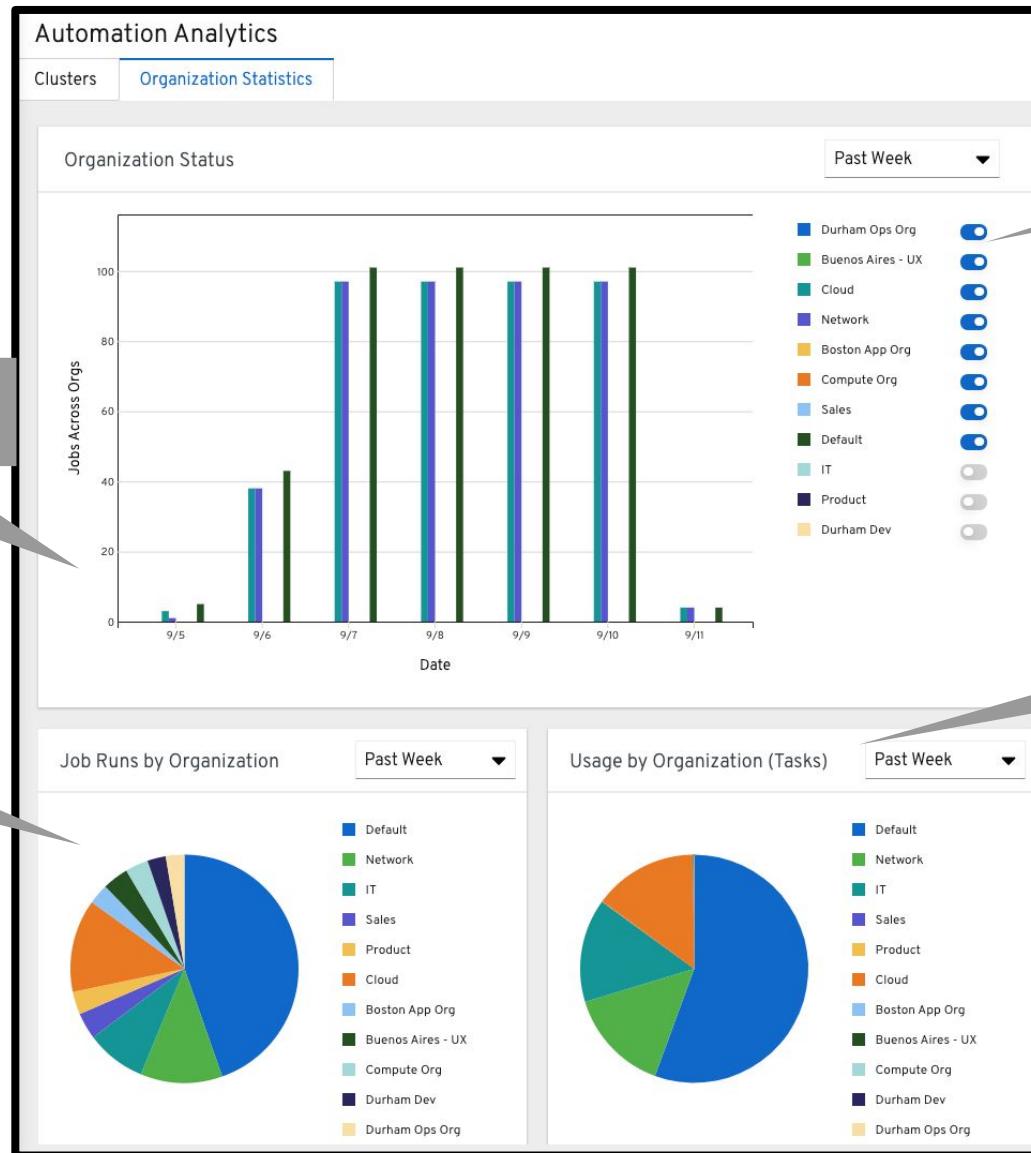
⚠ Cluster Error: No data from madrid.tower.example.com in 1 days

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

Organizational statistics

Job Status by Organization

Job Runs by Organization



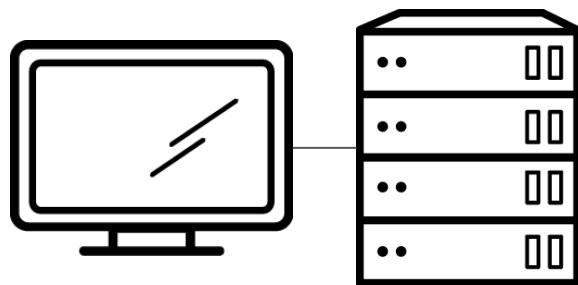
Filter by Organization

Usage by Organization

Dashboard comparison

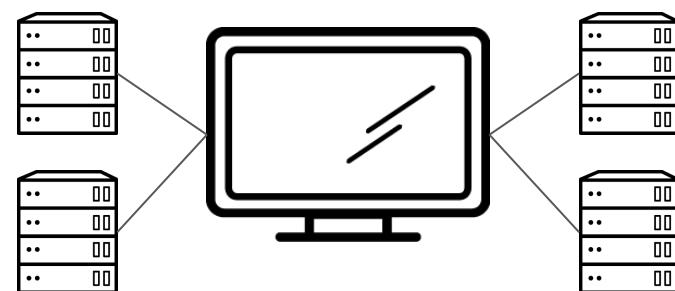
Ansible Tower

- 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

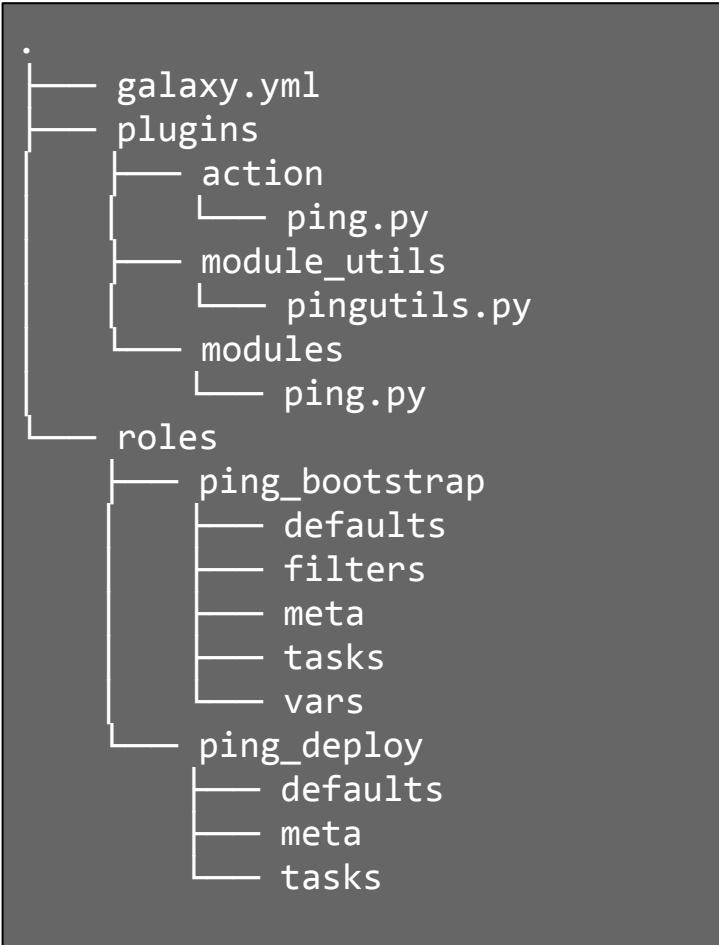
Simplifies internal collaboration, distribution, versioning

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:
```

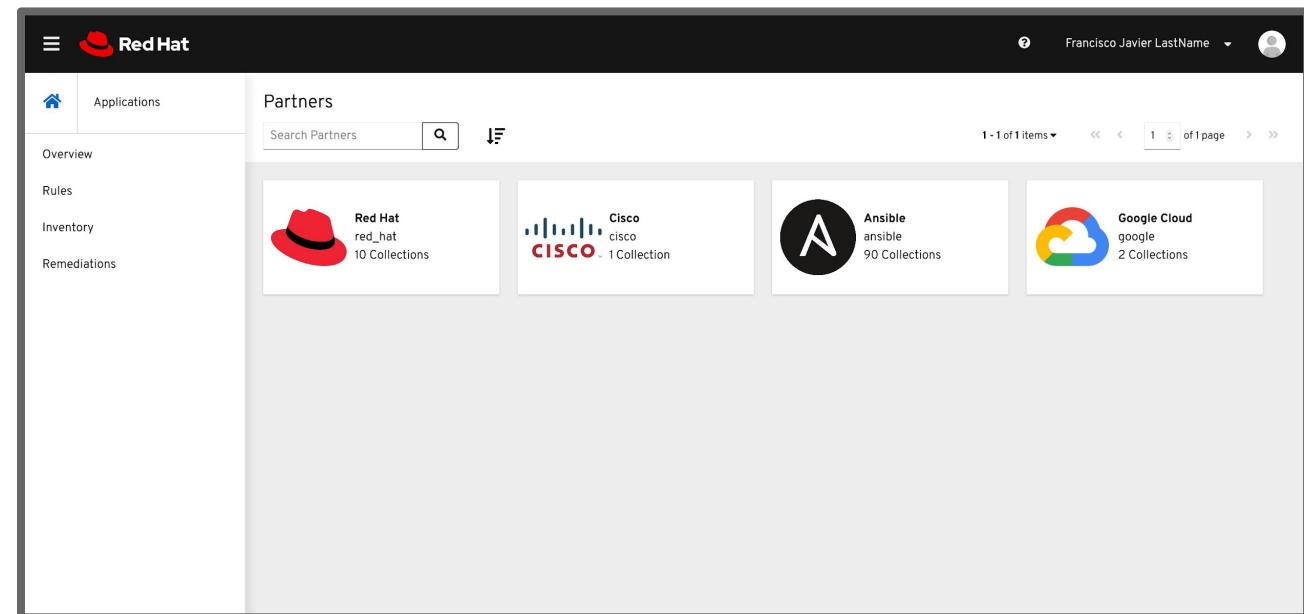
Automation Hub

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.*

Manage your locally available automation via on-premise.*



Next steps:

Get started

ansible.com/get-started

ansible.com/tower-trial

Join the community

ansible.com/community

Workshops and training

ansible.com/workshops

[Red Hat Training](#)

Share your story

[Follow us @Ansible](#)

[Friend us on Facebook](#)

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](https://www.linkedin.com/company/red-hat)



[youtube.com/user/AnsibleAutomation](https://www.youtube.com/user/AnsibleAutomation)



[facebook.com/AnsibleAutomation](https://www.facebook.com/AnsibleAutomation)



twitter.com/Ansible



Red Hat

Ansible Automation Platform

Strategic Use Cases



Red Hat
Ansible Automation
Platform

USE CASE:
LINUX AUTOMATION

LINUX AUTOMATION

150+
Linux Modules

AUTOMATE EVERYTHING LINUX

Red Hat Enterprise Linux, BSD,
Debian, Ubuntu and many more!

ONLY REQUIREMENTS:
Python 2 (2.6 or later)
or Python 3 (3.5 or later)

ansible.com/get-started



AUTOMATION FOR EVERYONE: SYSTEM ADMINISTRATORS

```
---  
- name: upgrade rhel packages  
  hosts: rhel  
  
  tasks:  
    - name: upgrade all packages  
      yum:  
        name: '*'  
        state: latest
```

AUTOMATION FOR EVERYONE: SYSTEM ADMINISTRATORS

```
---  
- name: reboot rhel hosts  
  hosts: rhel  
  
  tasks:  
    - name: reboot the machine  
      reboot:
```

AUTOMATION FOR EVERYONE: SYSTEM ADMINISTRATORS

```
---
```

- **name: check services on rhel hosts**
hosts: rhel
become: yes

tasks:

- **name: ensure nginx is started**
service:
name: nginx
state: started



Red Hat
Ansible Automation
Platform

USE CASE:
NETWORK AUTOMATION

ANSIBLE NETWORK AUTOMATION

65+

Network
Platforms

1000+

Network
Modules

15*

Galaxy
Network Roles

ansible.com/for/networks

galaxy.ansible.com/ansible-network

*Roles developed and maintained by Ansible Network Engineering



WHY AUTOMATE YOUR NETWORK?

PLAN AND PROTOTYPE VIRTUALLY

Use tasks as reusable building blocks

USE YOUR CURRENT DEVELOPMENT PRACTICES

Agile, DevOps, Waterfall

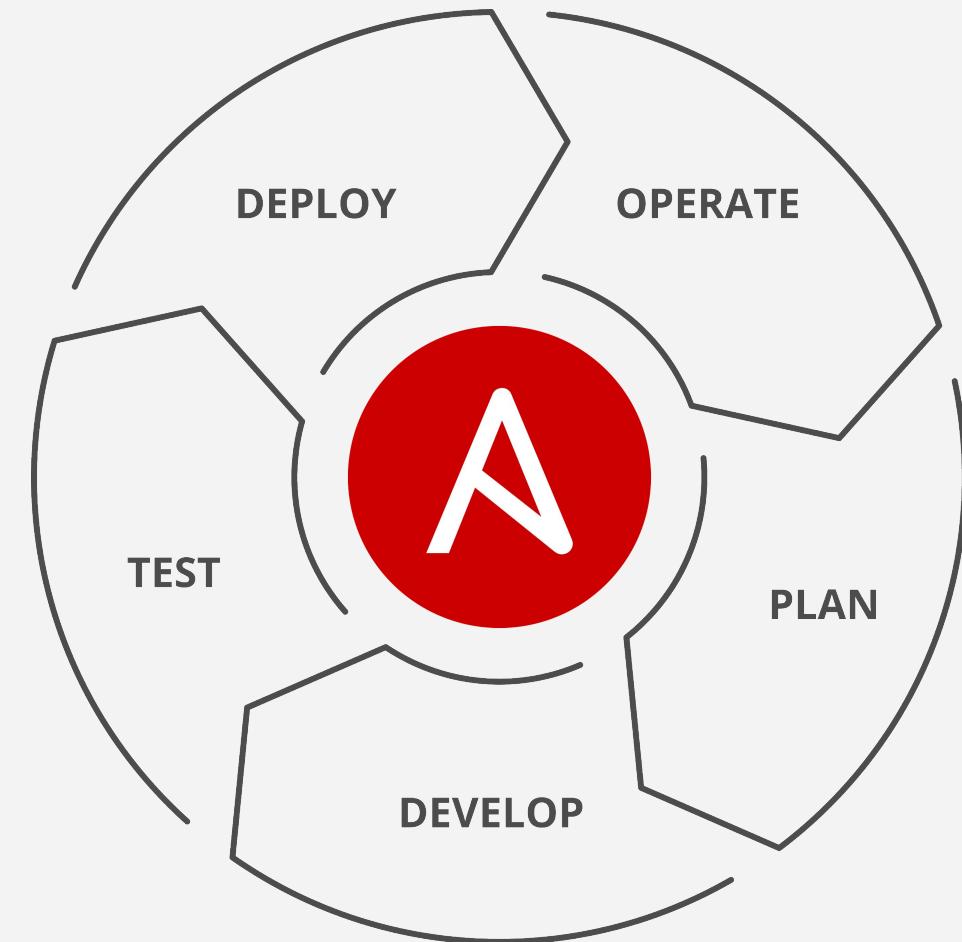
GO BEYOND THE “PING” TEST

Integrate with formal testing platforms

BE CONFIDENT DURING DEPLOYMENT

Validate changes were successful

ENSURE AN ON-GOING STEADY-STATE



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]$
```



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

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",           "10.0.0.4/8      ",
        "em1.0              up   up   inet    128.0.0.1/2     ,
        "                  "               128.0.0.4/2     ,
        "                  "               fe80::5254:ff:fe12:bdfe/64",
        "                  "               fec0::a:0:0:4/64",
        "                  "               0x4"
    ]
}
ok: [rtr4] => {
    "result.stdout_lines": [
        "Interface          Admin Link Proto  Local                  Remote",
        "em1                up   up",           "10.0.0.4/8      ",
        "em1.0              up   up   inet    128.0.0.1/2     ,
        "                  "               128.0.0.4/2     ,
        "                  "               fe80::5254:ff:fe12:bdfe/64",
        "                  "               fec0::a:0:0:4/64",
        "                  "               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]$
```





Red Hat
Ansible Automation
Platform

USE CASE:
WINDOWS AUTOMATION

WINDOWS AUTOMATION

90+

Windows
Modules

1,300+

Powershell DSC
resources

ansible.com/windows

AUTOMATION FOR EVERYONE: WINDOWS ADMINS

```
---
```

```
- name: windows playbook

  hosts: new_servers


  tasks:
    - name: ensure local admin account exists
      win_user:
        name: localadmin
        password: '{{ local_admin_password }}'
      groups: Administrators
```

AUTOMATION FOR EVERYONE: WINDOWS ADMINS

```
---
```

```
- name: windows playbook
  hosts: windows_machines

  tasks:
    - name: ensure common tools are installed
      win_chocolatey:
        name: '{{ item }}'
      loop: ['sysinternals', 'googlechrome']
```

AUTOMATION FOR EVERYONE: WINDOWS ADMINS

```
---
```

- **name: update and reboot**
hosts: windows_servers
tasks:
 - **name: ensure common OS updates are current**
win_updates:
register: update_result
 - **name: reboot and wait for host if updates change require it**
win_reboot:
when: update_result.reboot_required

AUTOMATION FOR EVERYONE: WINDOWS ADMINS

```
---
```

- **name:** update domain and reboot
 - hosts:** windows_servers
 - tasks:**
 - **name:** ensure domain membership
 - win_domain_membership:**
 - dns_domain_name:** contoso.corp
 - domain_admin_user:** '{{ domain_admin_username }}'
 - domain_admin_password:** '{{ domain_admin_password }}'
 - state:** domain
 - register:** domain_result
 - **name:** reboot and wait for host if domain change require it
 - win_reboot:**
 - when:** domain_result.reboot_required



Red Hat
Ansible Automation
Platform

USE CASE:
Cloud automation

CLOUD AUTOMATION

800+

Cloud
Modules

30+

Cloud Platforms

ansible.com/cloud

PLAYBOOK EXAMPLE: AWS

```
---
```

- **name:** aws playbook
 - hosts:** localhost
 - connection:** local

- tasks:**
 - **name:** create AWS VPC ansible-vpc
 - ec2_vpc_net:**
 - name:** "ansible-vpc"
 - cidr_block:** "192.168.0.0/24"
 - tags:**
 - demo:** the demo vpc
 - register:** create_vpc

PLAYBOOK EXAMPLE: AZURE

```
---
```

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

  tasks:
    - name: create virtual network
      azure_rm_virtualnetwork:
        resource_group: myResourceGroup
        name: myVnet
        address_prefixes: "10.0.0.0/16"
```

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



Red Hat
Ansible Automation
Platform

USE CASE:

Security Automation

What Is It?

Ansible Security Automation is our expansion deeper into the security use case. The goal is to provide a more efficient, streamlined way for security teams to automate their various processes for the identification, search, and response to security events. This is more complex and higher-value than the application of a security baseline (PCI, STIG, CIS) to a server.

Ansible Security Automation is a supported set of Ansible modules, roles and playbooks designed to unify the security response to cyberattacks.

What Does It Do?



Triage Of Suspicious Activities

Enabling programmatic access to log configurations such as destination, verbosity, etc.



Threat Hunting

Automating alerts, correlation searches and signature manipulation



Incident Response

Creating new security policies to whitelist, blacklist or quarantine a machine

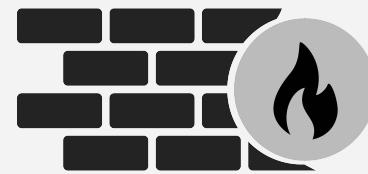
Partners - Security ISVs



Security Information &
Events Management

splunk®

IBM®



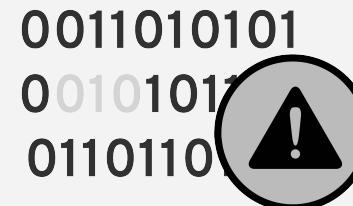
Enterprise
Firewalls

Check Point®
SOFTWARE TECHNOLOGIES LTD

CISCO™

f5®

FORTINET®



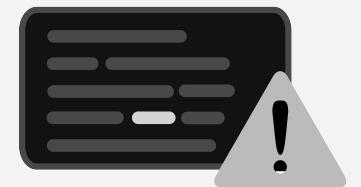
Intrusion Detection &
Prevention Systems

SNORT®



Check Point®
SOFTWARE TECHNOLOGIES LTD

FORTINET®



Privileged Access
Management

CYBERARK®

yncope™

Red Hat

AUTOMATION FOR EVERYONE: SECURITY OPERATIONS

```
- name: Create access rule in Checkpoint  
  
hosts: checkpoint  
  
connection: httpapi  
  
  
tasks:  
  - name: create access rule  
    checkpoint_access_rule:  
      layer: Network  
      name: "Drop attacker"  
      position: top  
      source: attacker  
      destination: Any  
      action: Drop
```

AUTOMATION FOR EVERYONE: SECURITY OPERATIONS

```
---
```

- name: Change QRadar rule state
 - hosts: qradar
 - tasks:
 - name: get info about qradar rule
 - qradar_rule_info:
 - name: "Potential DDoS Against Single Host (TCP)"
 - register: rule_info
 - name: disable rule by id
 - qradar_rule:
 - state: disabled
 - id: "{{ rule_info.rules[0]['id'] }}"

AUTOMATION FOR EVERYONE: SECURITY OPERATIONS

```
---
```

- **name:** Add Snort rule
 - hosts:** snort
 - tasks:**
 - **name:** Add snort password attack rule
 - include_role:**
 - name:** "ansible_security.ids_rule"
 - vars:**
 - rule:** "alert tcp any 443 -> 192.168.12.0/24 any"
 - state:** present
 - rules_file:** /etc/snort/rules/grab_everything_http.rules