



# **Red Hat**

## Ansible Automation Platform

# Ansible Linux Automation Workshop

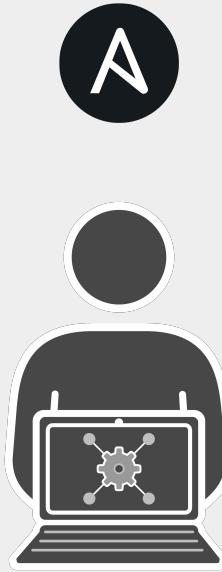
Introduction to, and hands on with, Ansible Automation Platform



# Intro to Ansible Automation Platform



**Red Hat**  
Ansible Automation  
Platform



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

# Teams are automating...



Lines Of Business



Network



Security



Operations

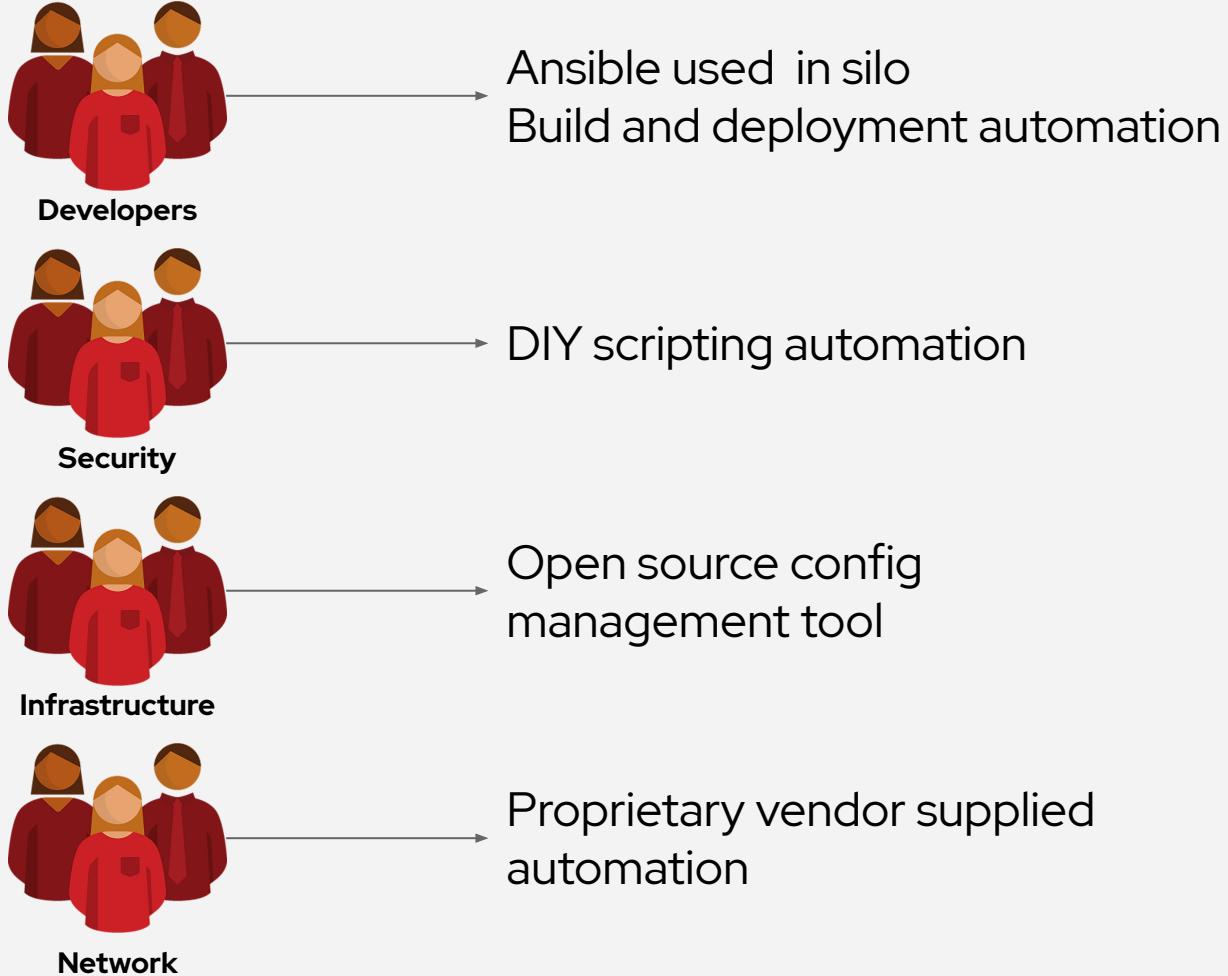


Developers



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**

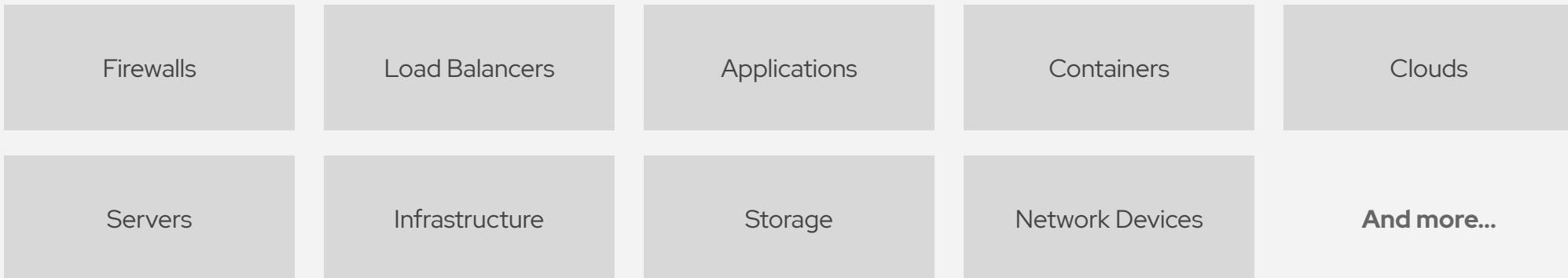
# What can I do using Ansible?

Automate the deployment and management of your entire IT footprint.

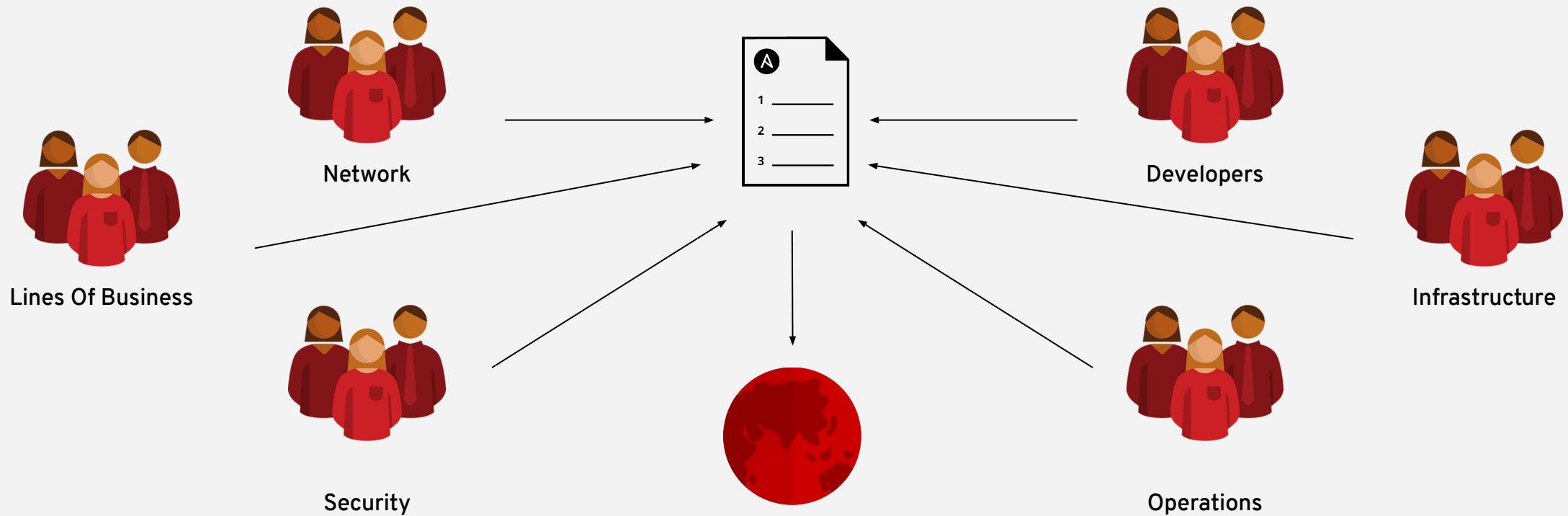
**Do this...**



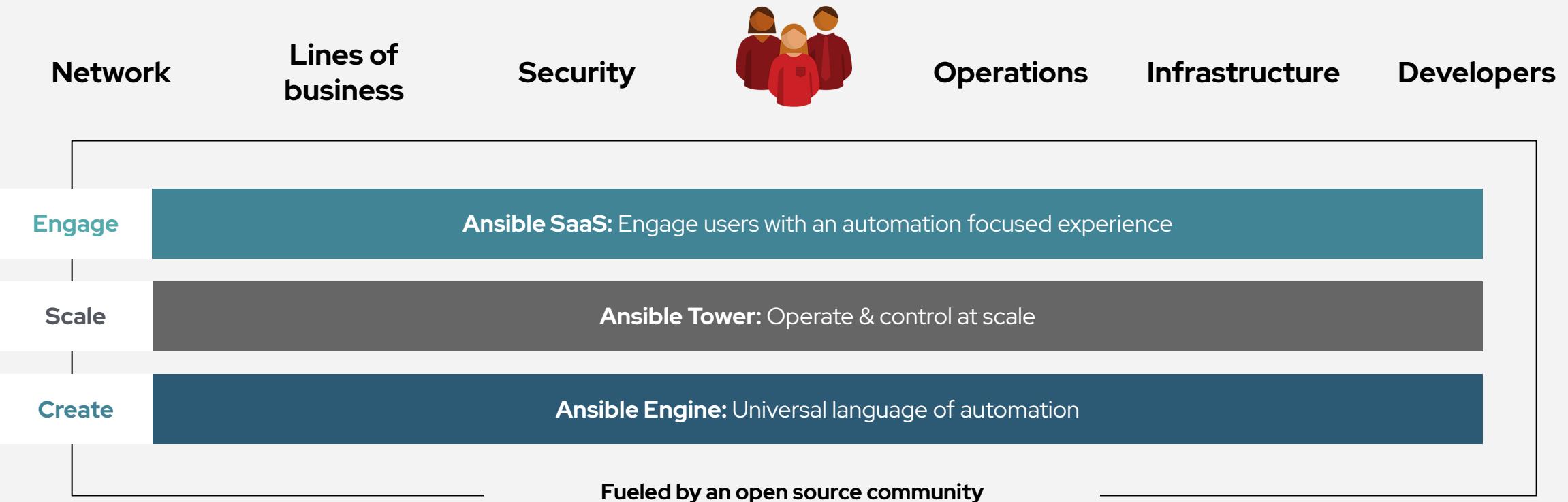
**On these...**



# When automation crosses teams, you need an automation platform



# 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](http://redhat.com/en/engage/total-economic-impact-ansible-tower-20180710)



# Breaking it down: Ansible Engine



**Red Hat**  
Ansible Automation  
Platform

# Red Hat Ansible Automation Platform

Network

Lines of  
business

Security

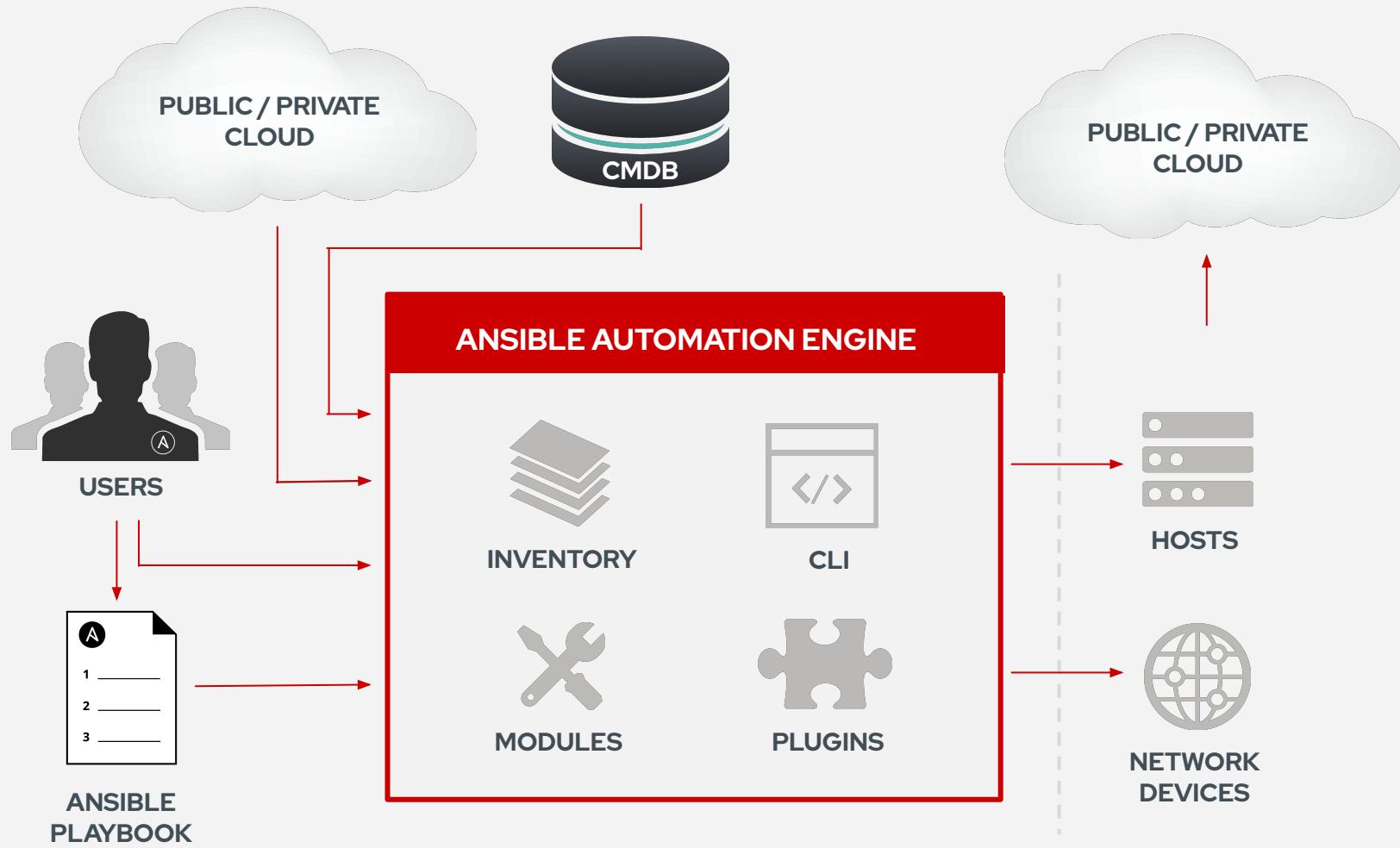


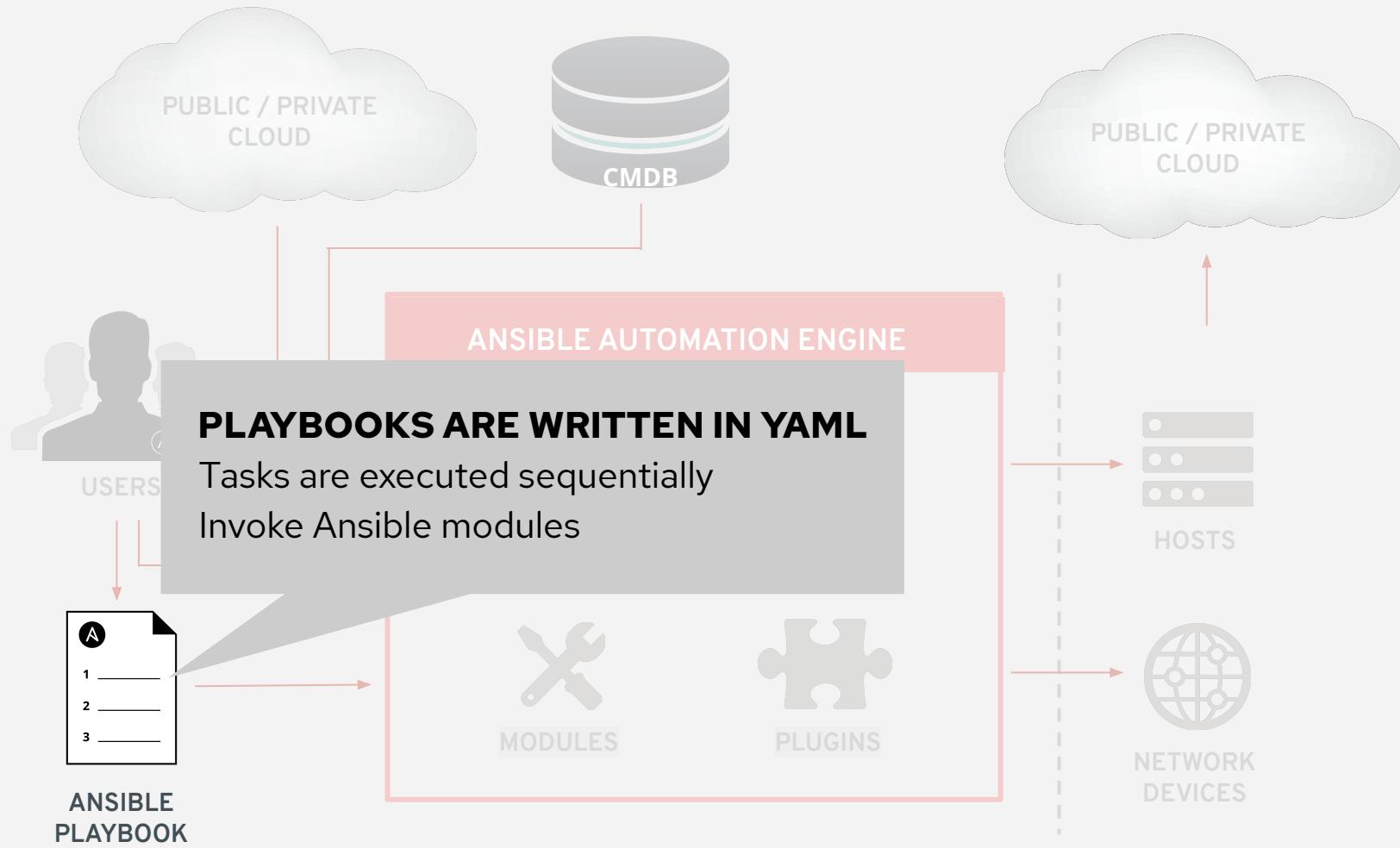
Operations

Infrastructure

Developers





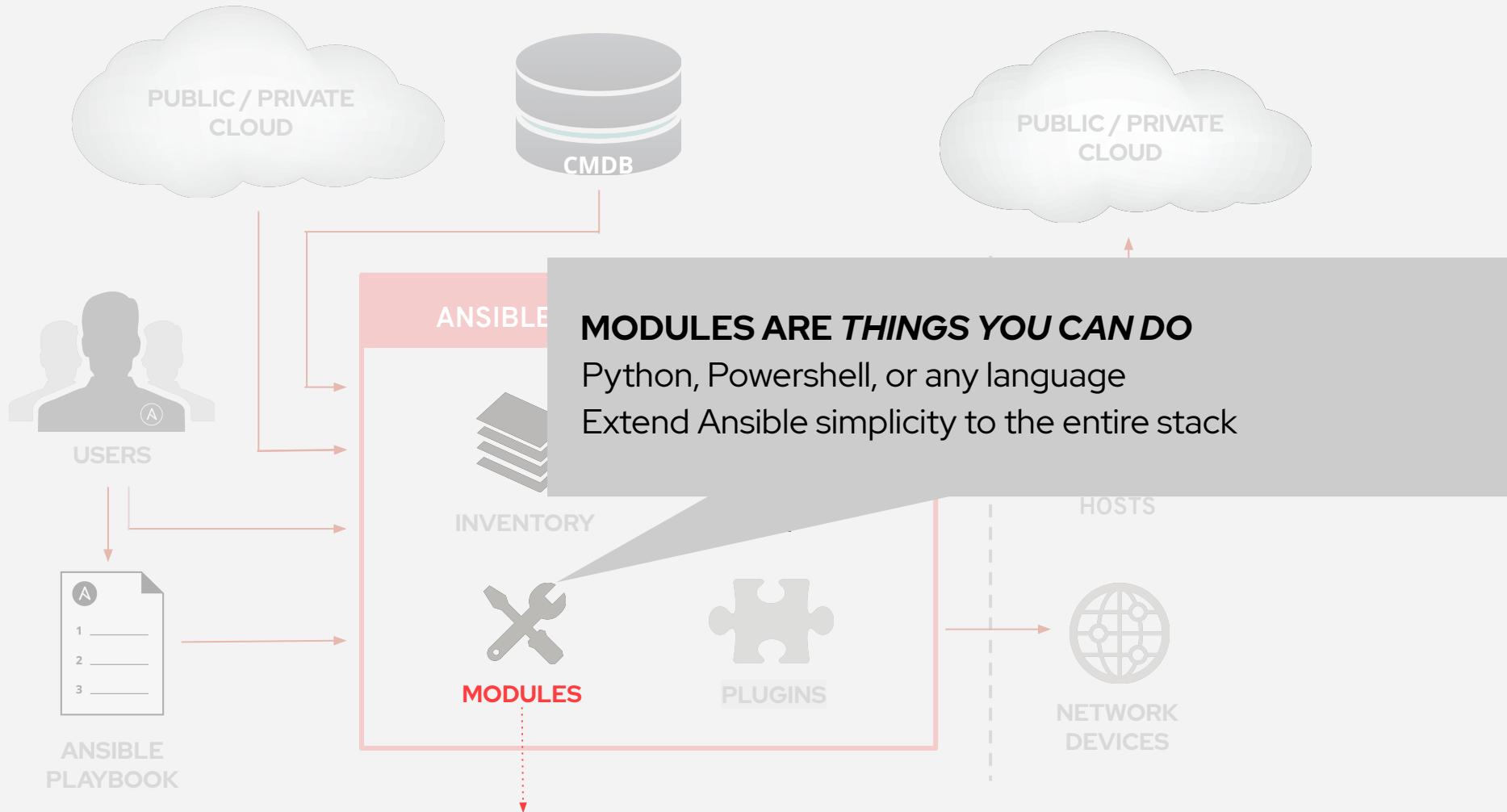


```
---
```

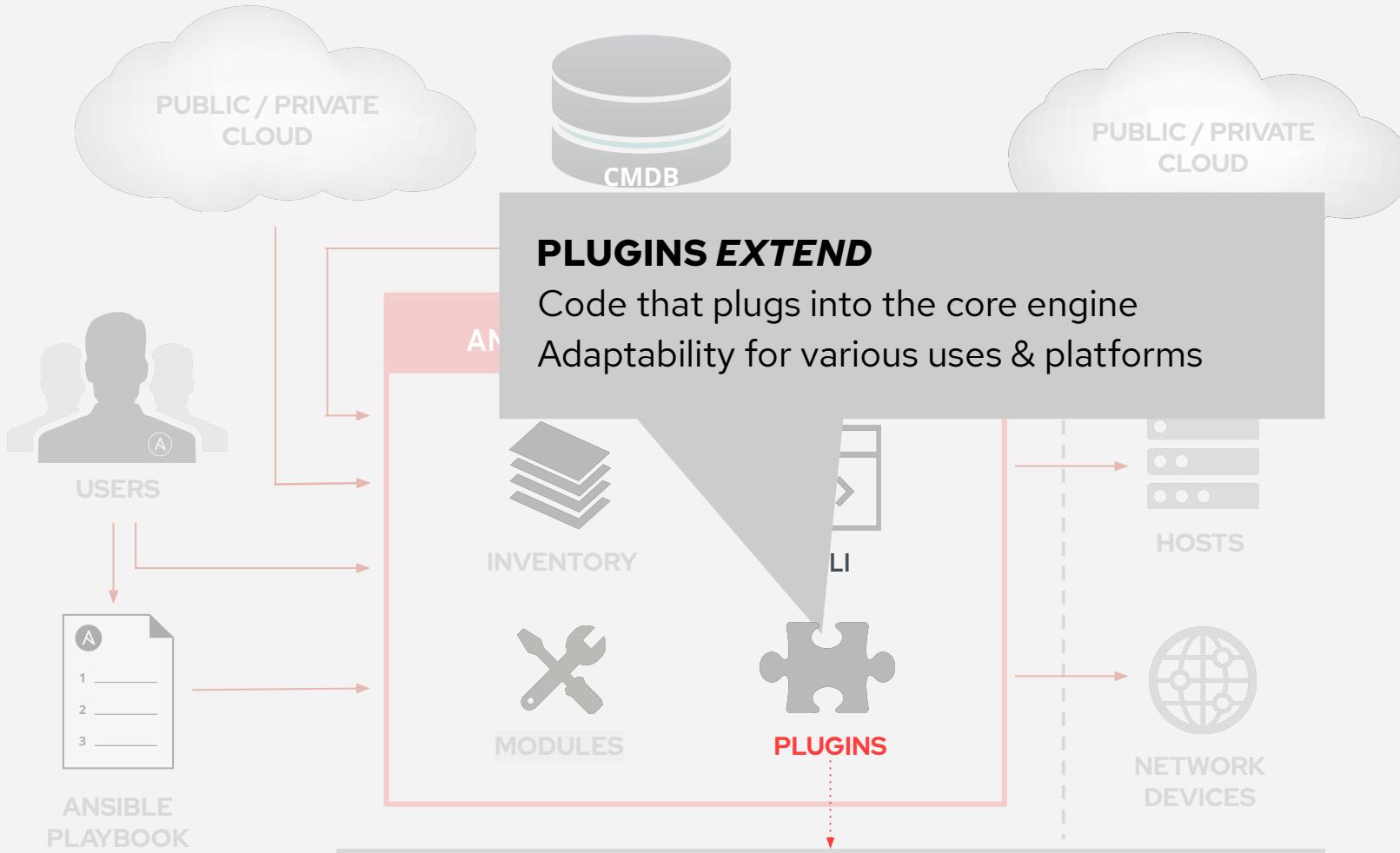
- **name: install and start apache**  
**hosts:** web  
**become:** yes

**tasks:**

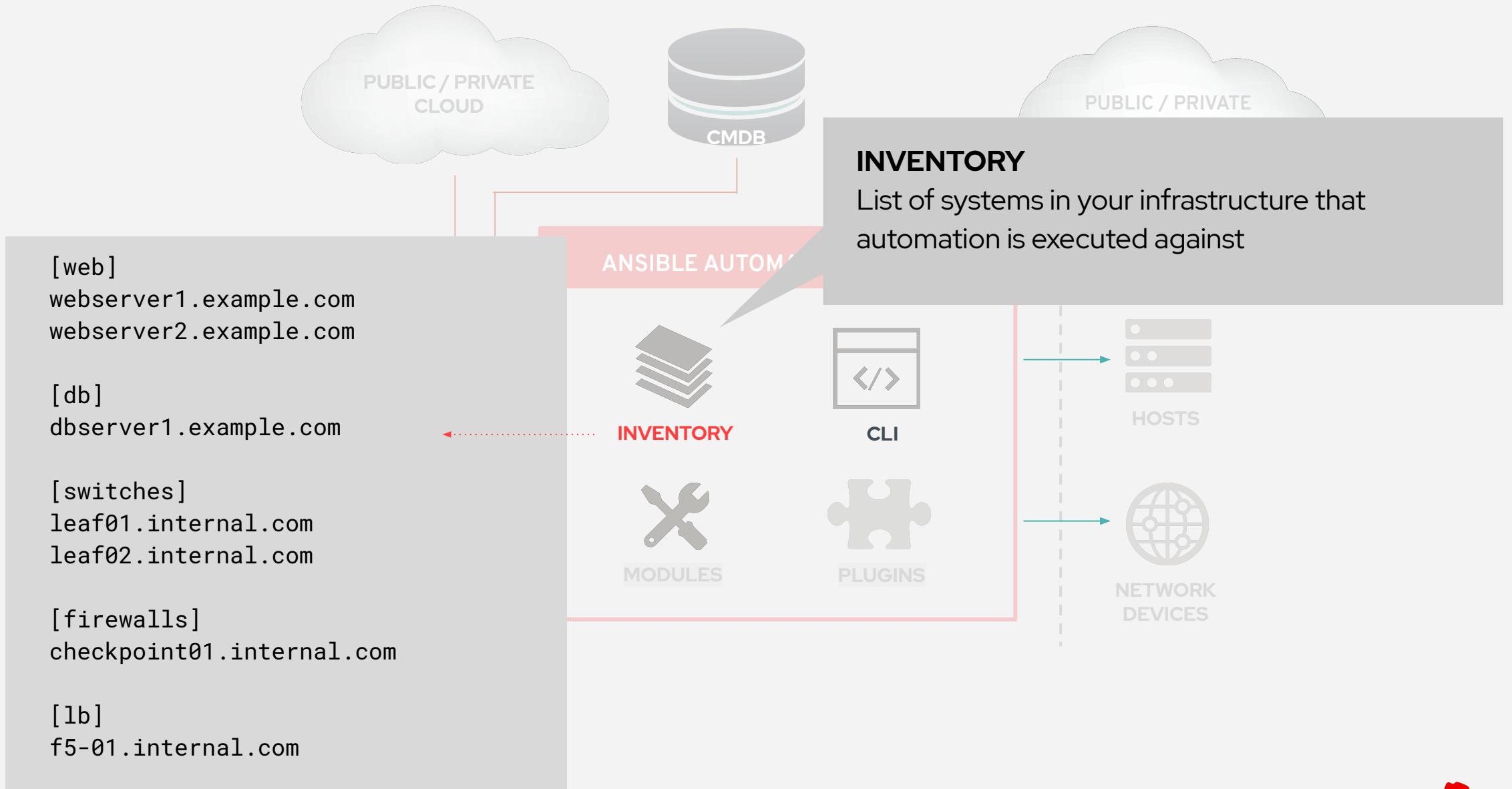
- **name: httpd package is present**  
**yum:**  
    **name:** httpd  
    **state:** latest
- **name: latest index.html file is present**  
**template:**  
    **src:** files/index.html  
    **dest:** /var/www/html/
- **name: httpd is started**  
**service:**  
    **name:** httpd  
    **state:** started



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



```
{ { some_variable | to_nice_yaml } }
```



# Inventory Groups

[**nashville**]

bnaapp01

bnaapp02

[**atlanta**]

atlapp03

Atlapp04

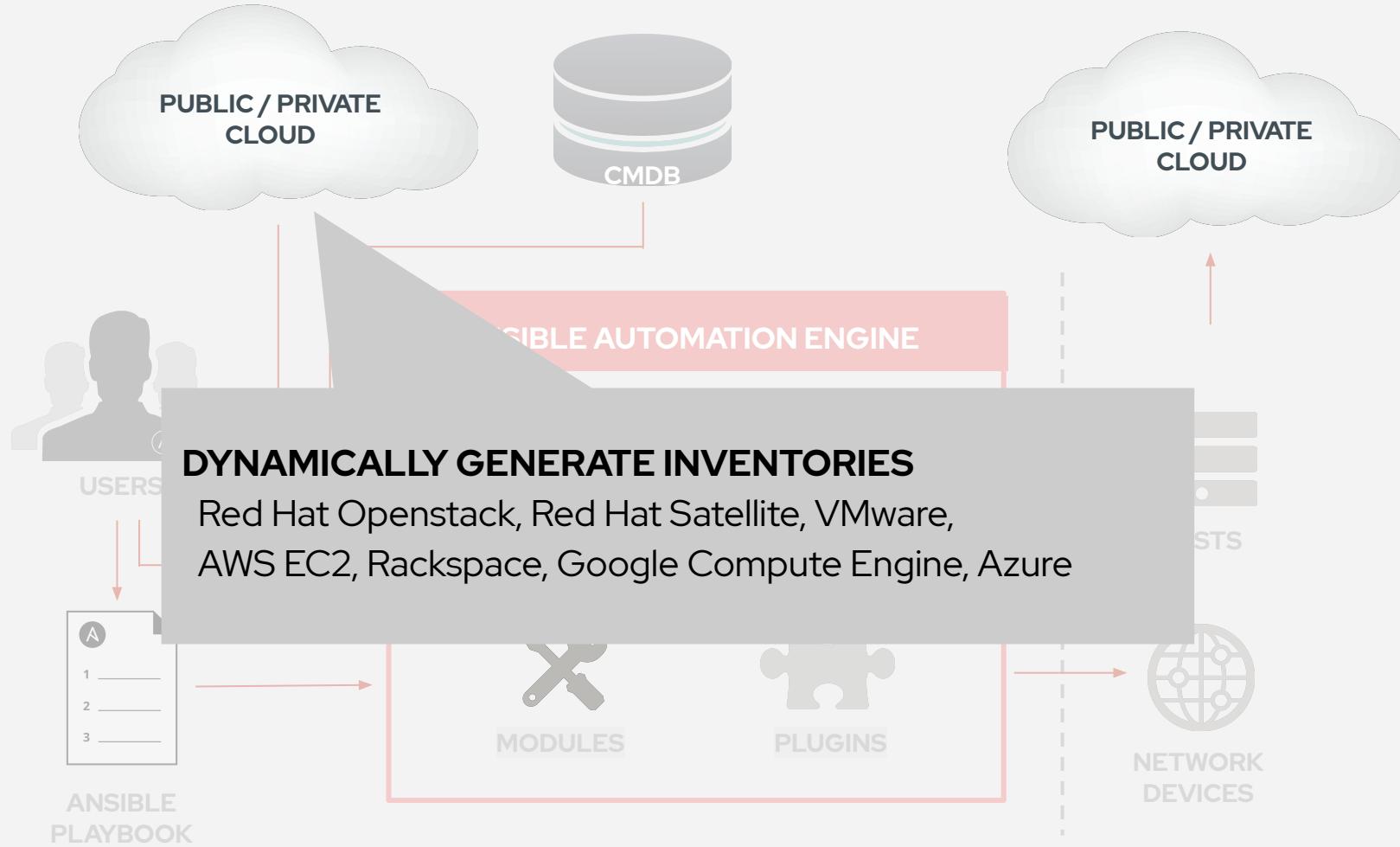
[**south**]

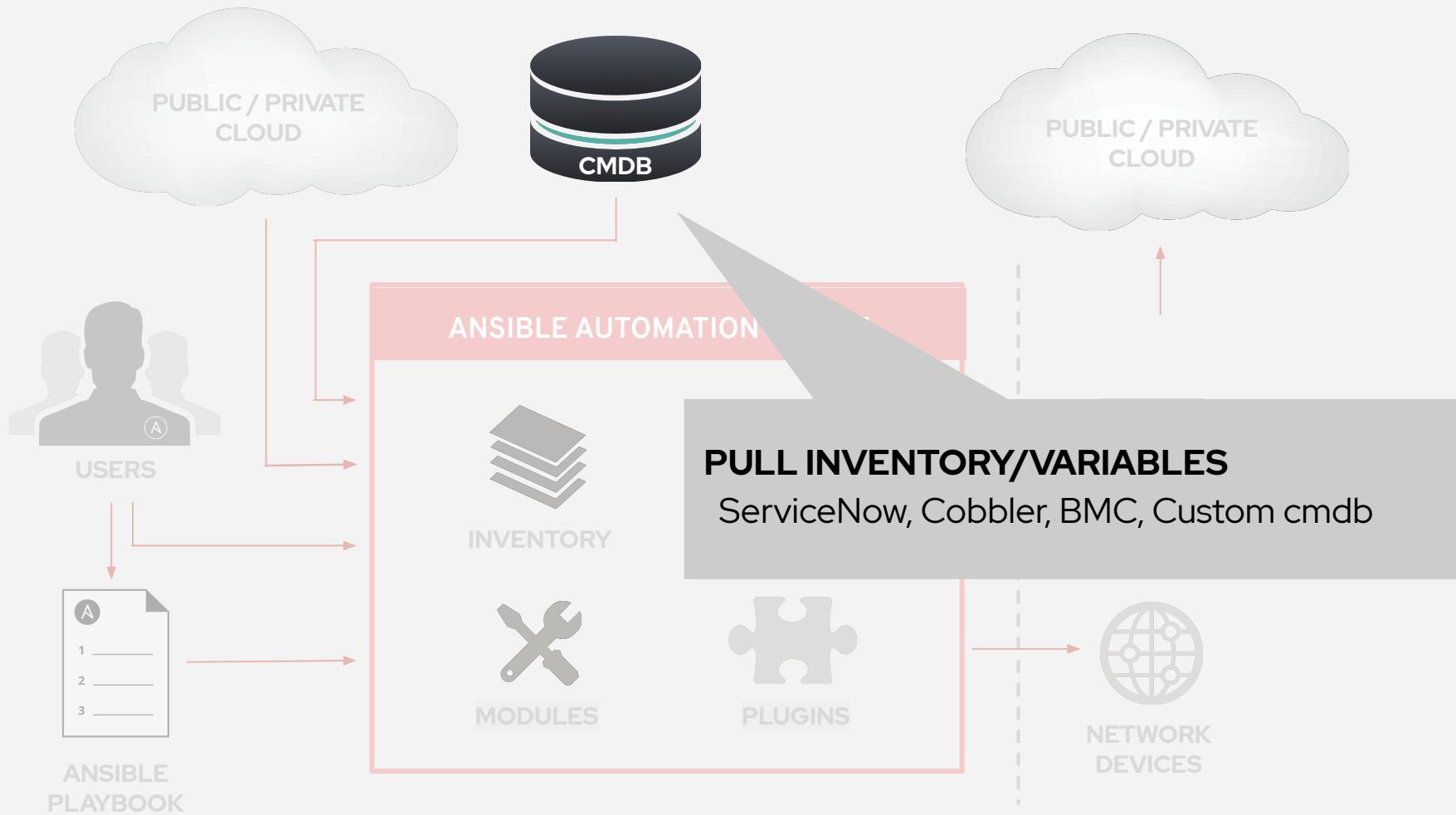
[**south:children**]

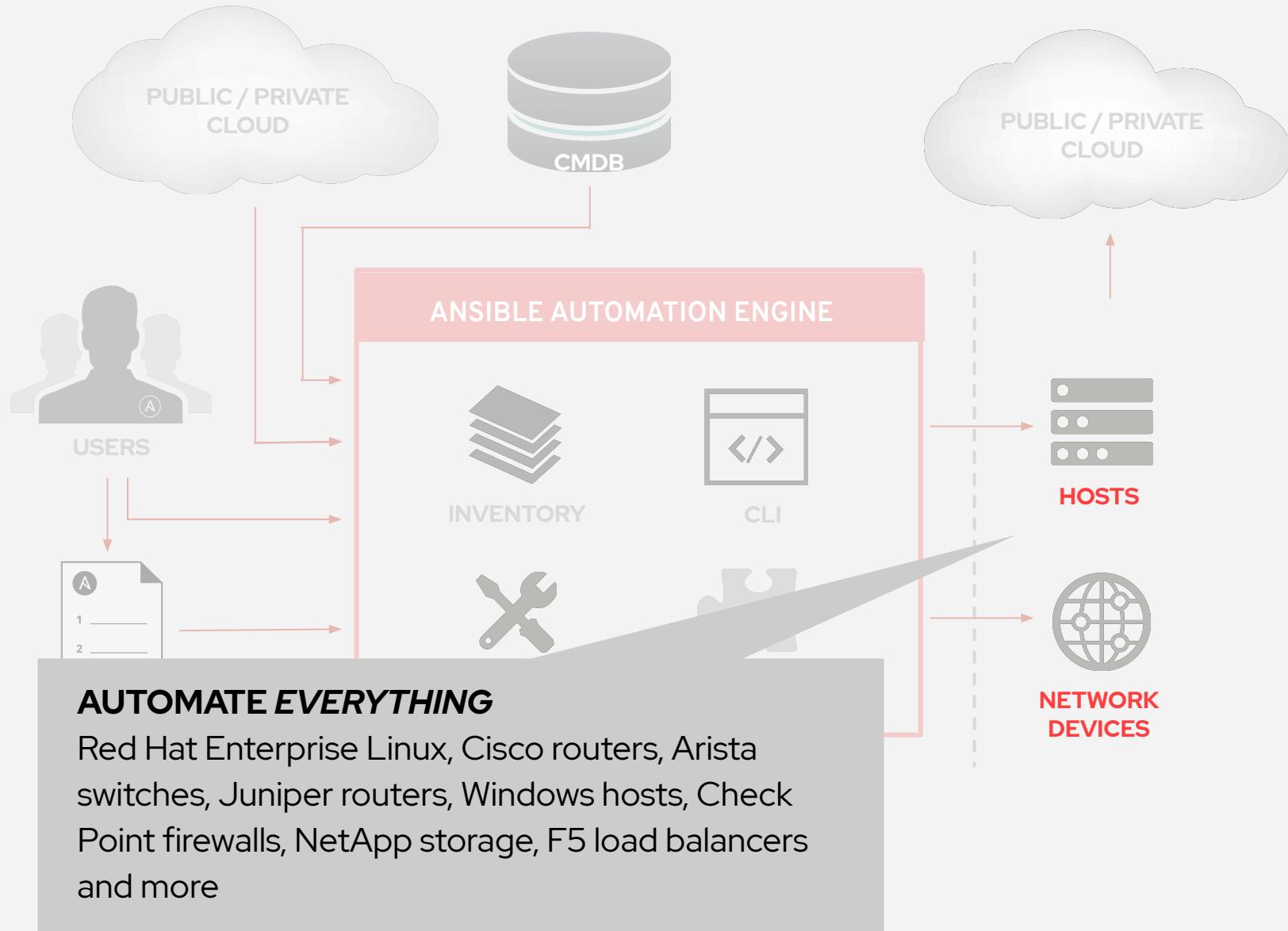
atlanta

nashville

hsvapp05







# The Ansible Command

Some basics to keep you from getting stuck

--help (Display some basic and extensive options)

```
[user@ansible ~]$ ansible --help
Usage: ansible <host-pattern> [options]
```

Define and run a single task 'playbook' against a set of hosts

Options:

```
-a MODULE_ARGS, --args=MODULE_ARGS
                      module arguments
--ask-vault-pass      ask for vault password
-B SECONDS, --background=SECONDS
<<<snippet, output removed for brevity>>>
```

# First Ad-Hoc Command: ping

- Single Ansible command to perform a task quickly directly on command line
- Most basic operation that can be performed
- Here: an example Ansible ping - not to be confused with ICMP

```
$ ansible all -m ping
```

# Ansible Modules

## Using **ansible-doc** to specify one module

```
[user@ansible ~]$ ansible-doc copy
> COPY      (/usr/lib/python2.7/site-packages/ansible/modules/files/copy.py)
```

The `copy' module copies a file from the local or remote machine to a location on the remote machine. Use the [fetch] module to copy files from remote locations to the local box. If you need variable interpolation in copied files, use the [template] module. For Windows targets, use the [win\_copy] module instead.

\* note: This module has a corresponding action plugin.

OPTIONS (= is mandatory):

- attributes

Attributes the file or directory should have. To get supported flags look at the man page for `chattr' on the target system. This string should contain the attributes in the same order as the one displayed by `lsattr'.  
`=' operator is assumed as default, otherwise `+' or `-' operators need to be included in the string.

(Aliases: attr) [Default: (null)]

version\_added: 2.3

# An Ansible Play in an Ansible Playbook

A play

```
- hosts: db
vars:
  software:
    - mariadb-server
roles:
  - install_wordpress_db

- hosts: web
vars:
  software:
    - httpd
    - php
roles:
  - install_wordpress_web
```

Another  
play

# An Ansible Play in an Ansible Playbook

A play

```
- hosts: db
  vars:
    software:
      - mariadb-server
  roles:
    - install_wordpress_db

- hosts: web
  vars:
    software:
      - httpd
      - php
  roles:
    - install_wordpress_web
```

Another play

# Running an Ansible Playbook:

The most important colors of Ansible

A task executed as expected, no change was made.

A task executed as expected, making a change

A task failed to execute successfully

# Running an Ansible Playbook

```
[user@ansible] $ ansible-playbook apache.yml

PLAY [webservers] ****
TASK [Gathering Facts] ****
ok: [web2]
ok: [web1]
ok: [web3]

TASK [Ensure httpd package is present] ****
changed: [web2]
changed: [web1]
changed: [web3]

TASK [Ensure latest index.html file is present] ****
changed: [web2]
changed: [web1]
changed: [web3]

TASK [Restart httpd] ****
changed: [web2]
changed: [web1]
changed: [web3]

PLAY RECAP ****
web2          : ok=1    changed=3  unreachable=0   failed=0
web1          : ok=1    changed=3  unreachable=0   failed=0
web3          : ok=1    changed=3  unreachable=0   failed=0
```

# An Ansible Playbook Variable Example

```
---
- name: variable playbook test
  hosts: localhost

  vars:
    var_one: awesome
    var_two: ansible is
    var_three: "{{ var_two }} {{ var_one }}"

  tasks:
    - name: print out var_three
      debug:
        msg: "{{var_three}}"
```

# Roles

- Roles: Think Ansible packages
- Roles provide Ansible with a way to load tasks, handlers, and variables from separate files.
- Roles group content, allowing easy sharing of code with others
- Roles make larger projects more manageable
- Roles can be developed in parallel by different administrators

# Role structure

- Defaults: default variables with lowest precedence (e.g. port)
- Handlers: contains all handlers
- Meta: role metadata including dependencies to other roles
- Tasks: plays or tasks
- Templates: templates to deploy
- Tests: place for playbook tests
- Vars: variables (e.g. override port)

```
user/
├── defaults
│   └── main.yml
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
└── tests
    ├── inventory
    └── test.yml
└── vars
    └── main.yml
```

# Q&A/Break + Lab time! (link to lab)



**Red Hat**  
Ansible Automation  
Platform

# Breaking it down again: Ansible Tower



**Red Hat**  
Ansible Automation  
Platform

# Red Hat Ansible Automation Platform



**Network**

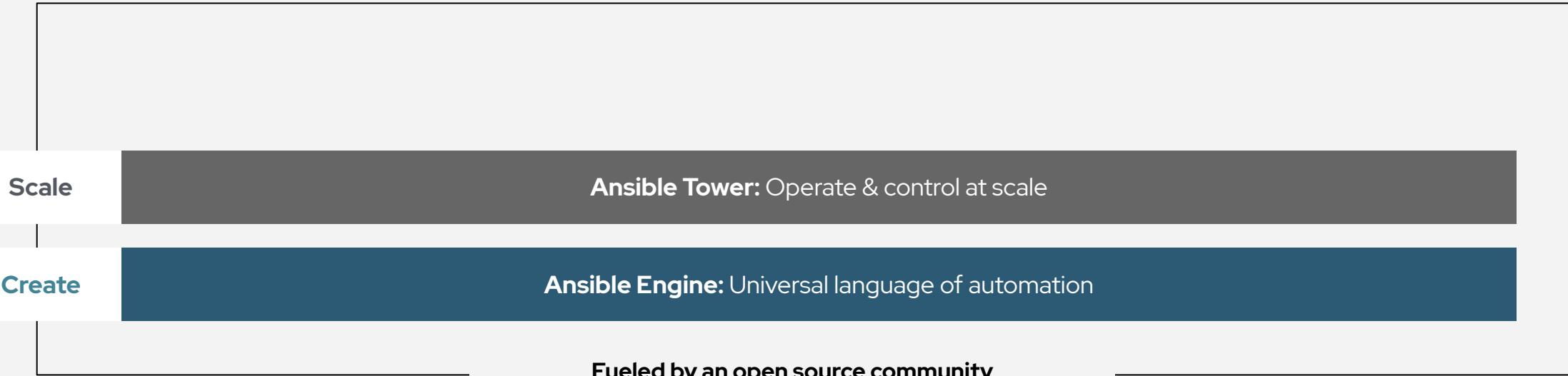
**Lines of business**

**Security**

**Operations**

**Infrastructure**

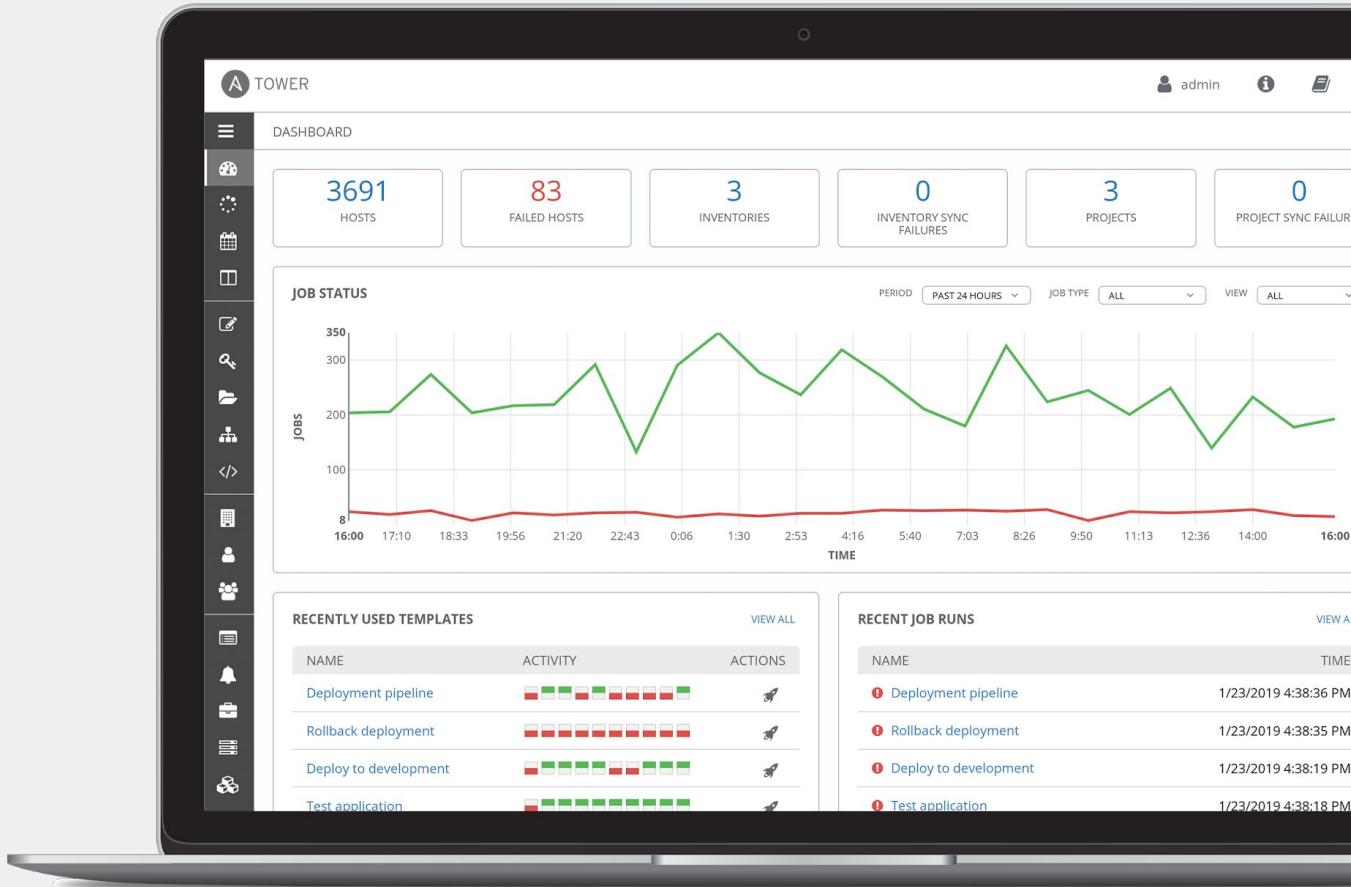
**Developers**



# 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



# 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

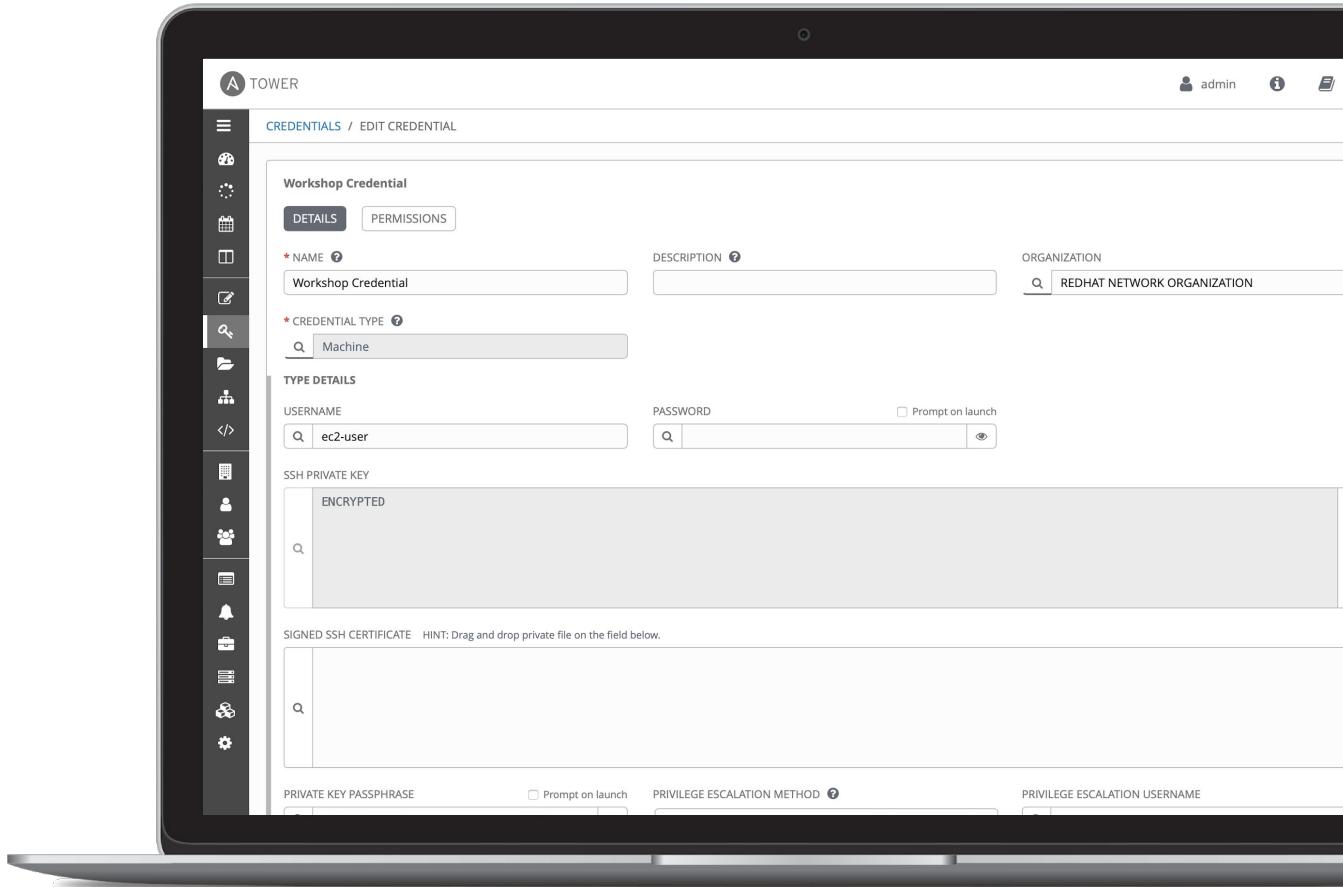
The screenshot shows the Ansible Tower web interface. The top navigation bar includes 'INVENTORIES / Workshop Inventory / HOSTS'. On the left is a sidebar with various icons for managing inventories, hosts, permissions, groups, sources, and completed jobs. The main content area displays a table titled 'Workshop Inventory' under the 'HOSTS' tab. The table lists five hosts: 'ON' (radio button), 'ansible' (radio button), 'rtr1' (radio button), 'rtr2' (radio button), 'rtr3' (radio button), and 'rtr4' (radio button). To the right of the host list are 'RELATED GROUPS' buttons for 'control', 'cisco', 'dc1', 'arista', 'dc2', 'dc1', 'juniper', 'arista', and 'dc2'. Below the host table is another search bar and a table with columns for 'INVENTORIES' and 'HOSTS', showing results 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

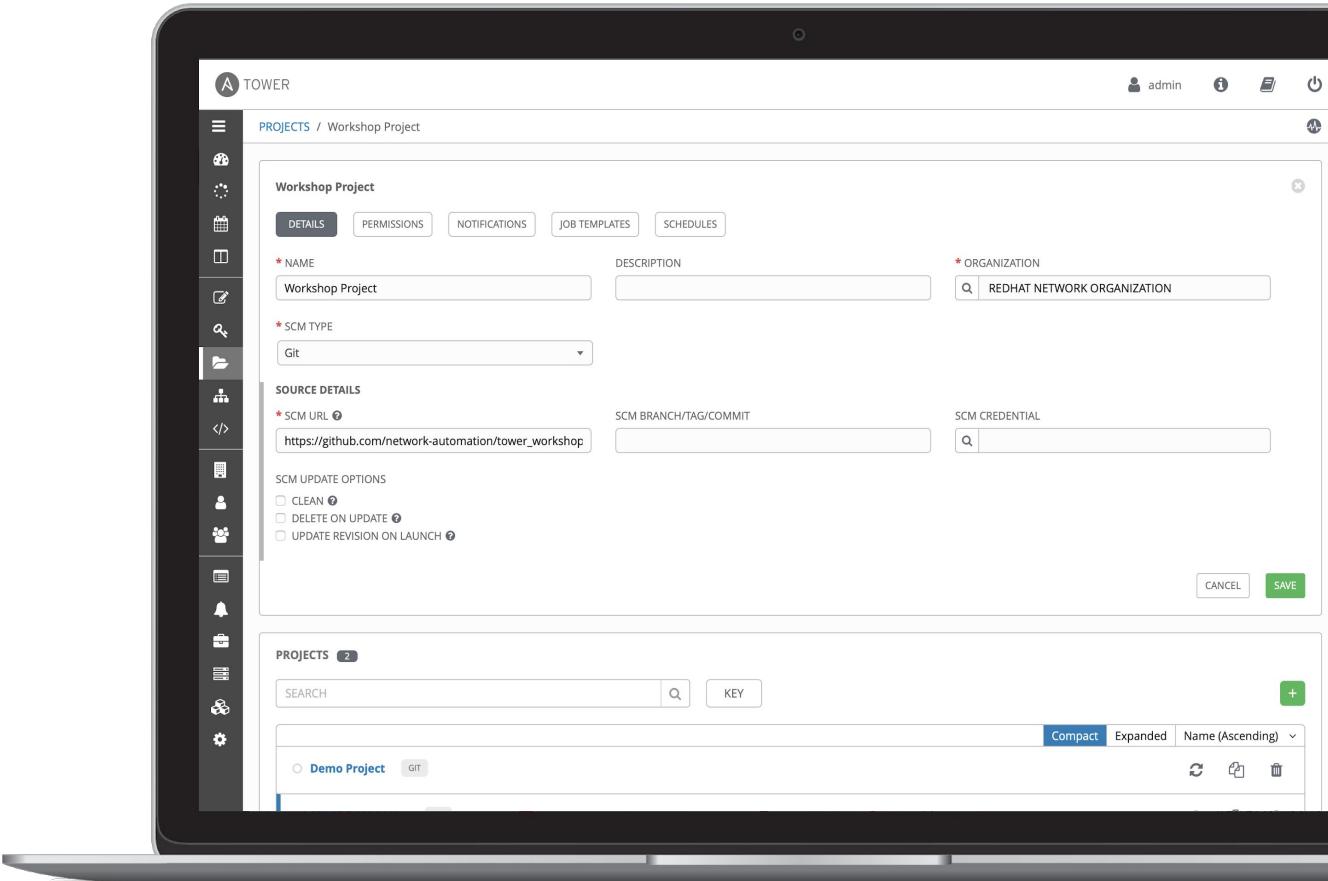
Centralized management of various credentials allows end users to leverage a secret without ever exposing that secret to them.



# 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.



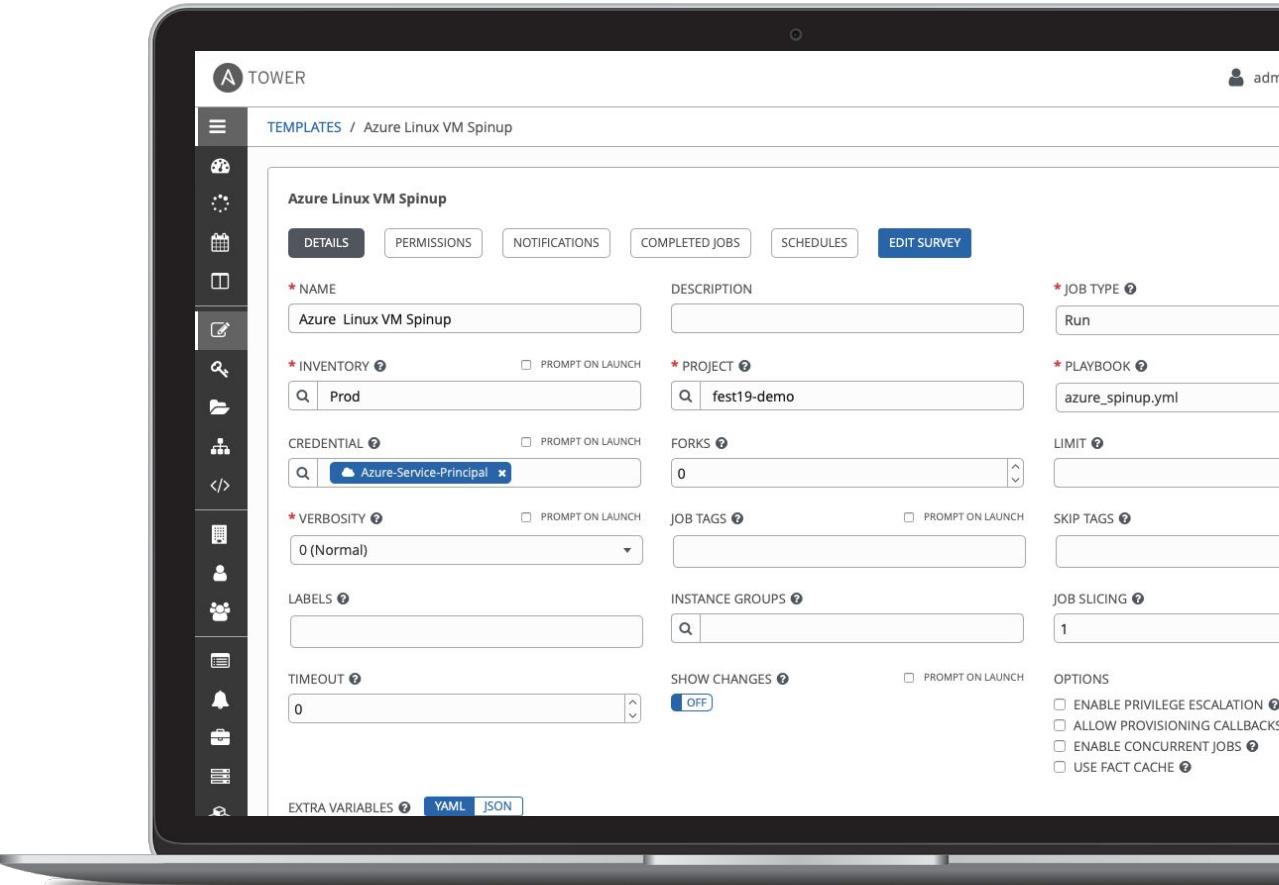
# Job Templates

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



# Surveys

Tower surveys allow you to configure how a job runs via a series of questions, making it simple to customize your jobs in a user-friendly way.

An Ansible Tower survey is a simple question-and-answer form that allows users to customize their job runs. Combine that with Tower's role-based access control, and you can build simple, easy self-service for your users.

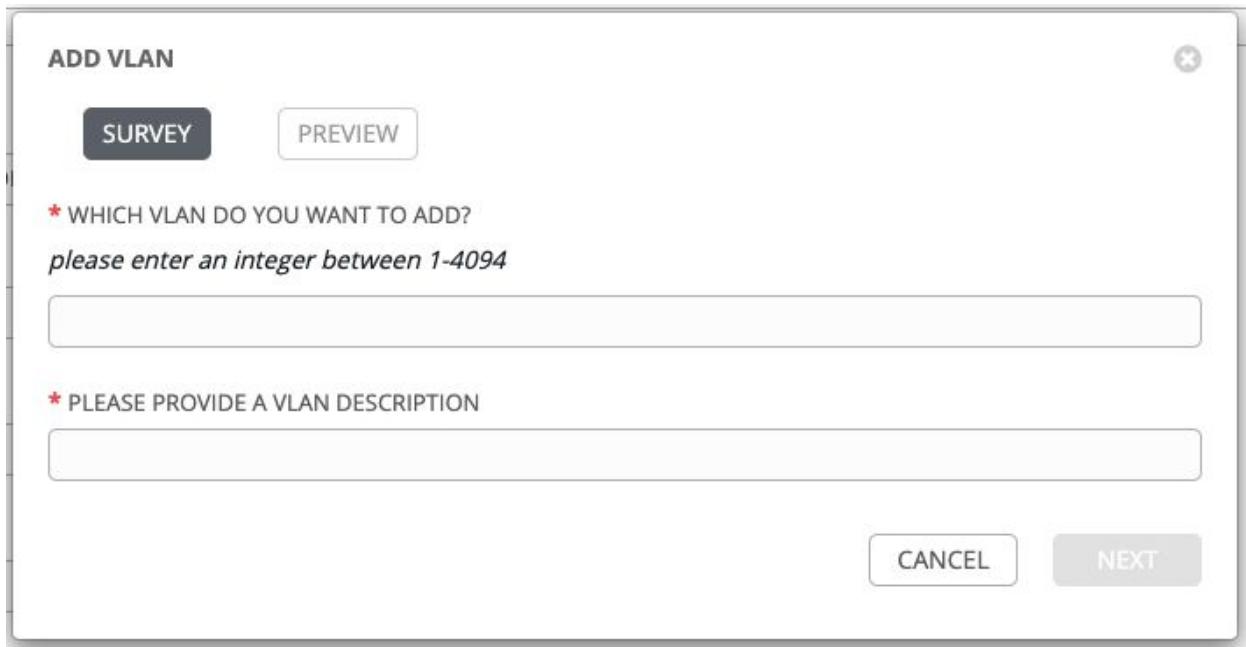
ADD VLAN

SURVEY PREVIEW

\* WHICH VLAN DO YOU WANT TO ADD?  
*please enter an integer between 1-4094*

\* PLEASE PROVIDE A VLAN DESCRIPTION

CANCEL NEXT



# Using a Survey

When launching a job, the user will now be prompted with the Survey. The user can be required to fill out the Survey before the Job Template will execute.

The screenshot shows the TOWER interface. On the left is a dark sidebar with various navigation options: Views, Dashboard, Jobs, Schedules, My View, Templates (which is selected), Credentials, Projects, Inventories, Inventory Scripts, Organizations, Users, and Teams. The main area is titled 'TEMPLATES' and lists four job templates: 'Network-Restore' (Job Template), 'Network-System' (Job Template), 'Network-Time' (Job Template), and 'Network-User' (Job Template). To the right of these templates is a 'CONFIGURE BANNER' dialog box. The dialog has tabs for 'SURVEY' (which is selected) and 'PREVIEW'. It contains a text field with the placeholder 'Please type into the text field the desired banner' and a note '\* PLEASE ENTER THE BANNER TEXT'. Below the text field are 'CANCEL' and 'NEXT' buttons. To the right of the dialog is a list of items, each with a green '+' button, a name, and three icons: a rocket, a clipboard, and a trash can. The names listed are 'Network-Restore', 'Network-System', 'Network-Time', and 'Network-User'. At the bottom of the page, it says 'ITEMS 1 - 7'.

# Role Based Access Control (RBAC)

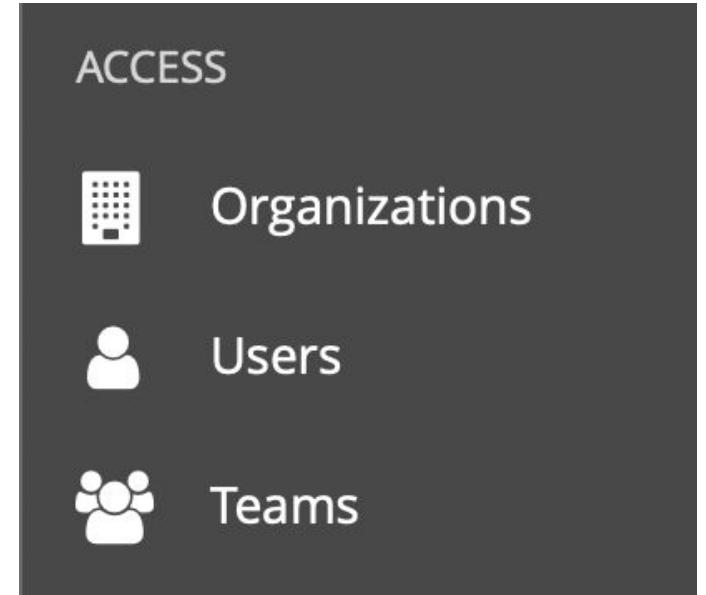
Role-Based Access Controls (RBAC) are built into Ansible Tower and allow administrators to delegate access to inventories, organizations, and more.

These controls allow Ansible Tower to help you increase security and streamline management of your Ansible automation.



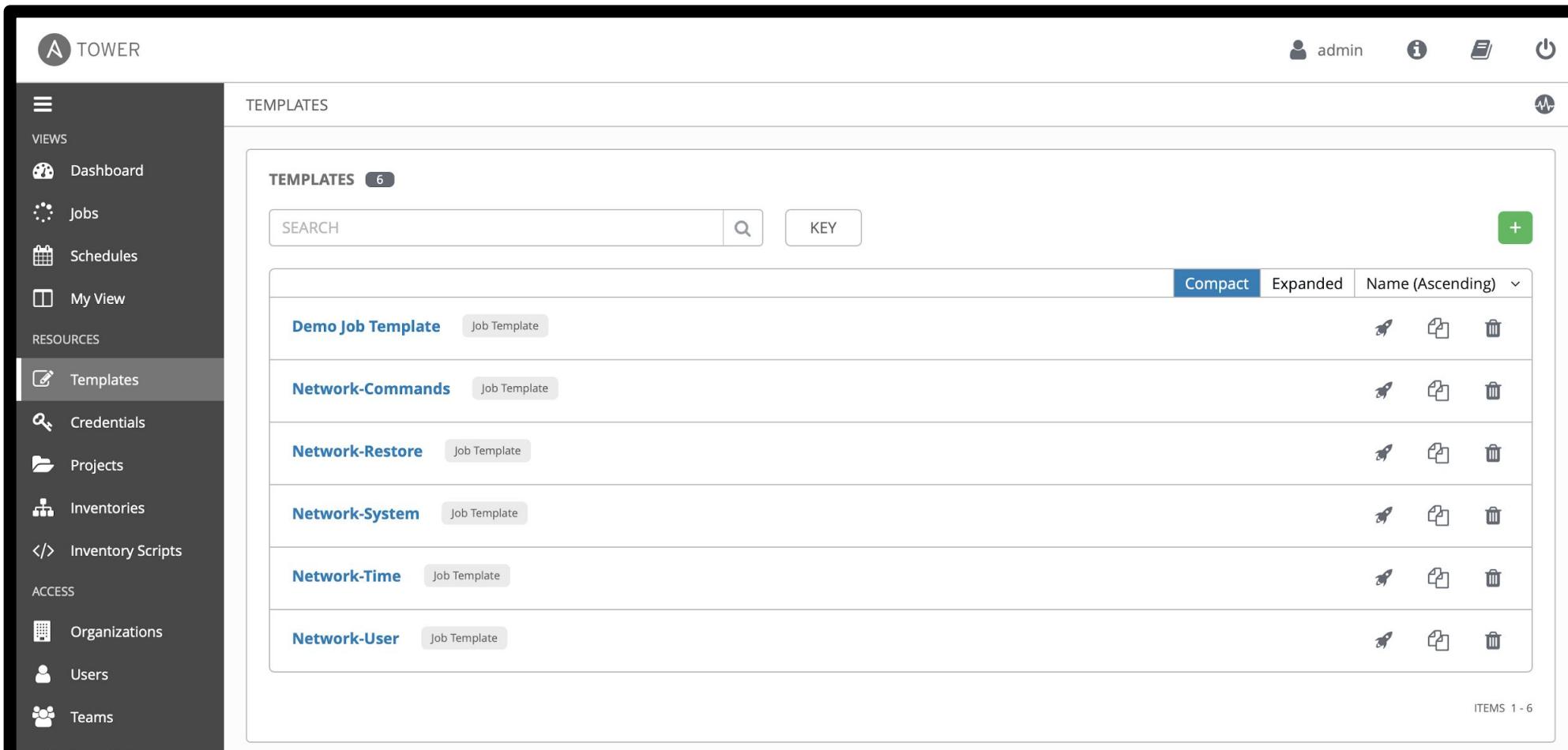
# User Management

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



# Workflows

Workflows can be found alongside Job Templates by clicking the **Templates**  button under the *RESOURCES* section on the left menu.



The screenshot shows the Ansible Tower web interface. The top navigation bar includes the TOWER logo, user info (admin), and various icons. On the far left is a dark sidebar with a navigation menu:

- VIEWS: Dashboard, Jobs, Schedules, My View
- RESOURCES: **Templates** (selected), Credentials, Projects, Inventories, Inventory Scripts
- ACCESS: Organizations, Users, Teams
- ADMINISTRATION

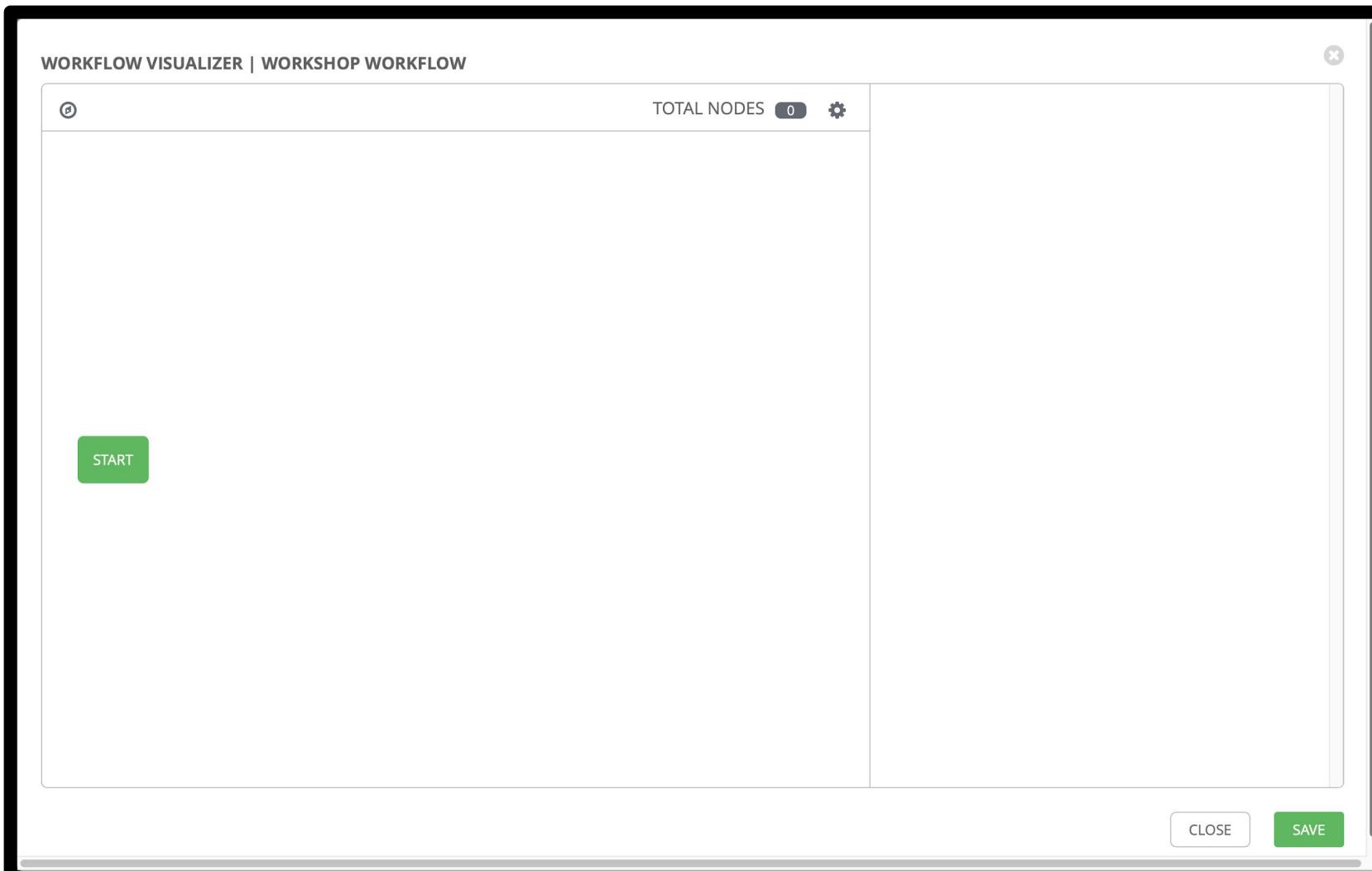
The main content area is titled "TEMPLATES" and shows a list of six items:

NAME	TYPED AS	OPTIONS
Demo Job Template	Job Template	
Network-Commands	Job Template	
Network-Restore	Job Template	
Network-System	Job Template	
Network-Time	Job Template	
Network-User	Job Template	

At the bottom right of the main content area, it says "ITEMS 1 - 6".

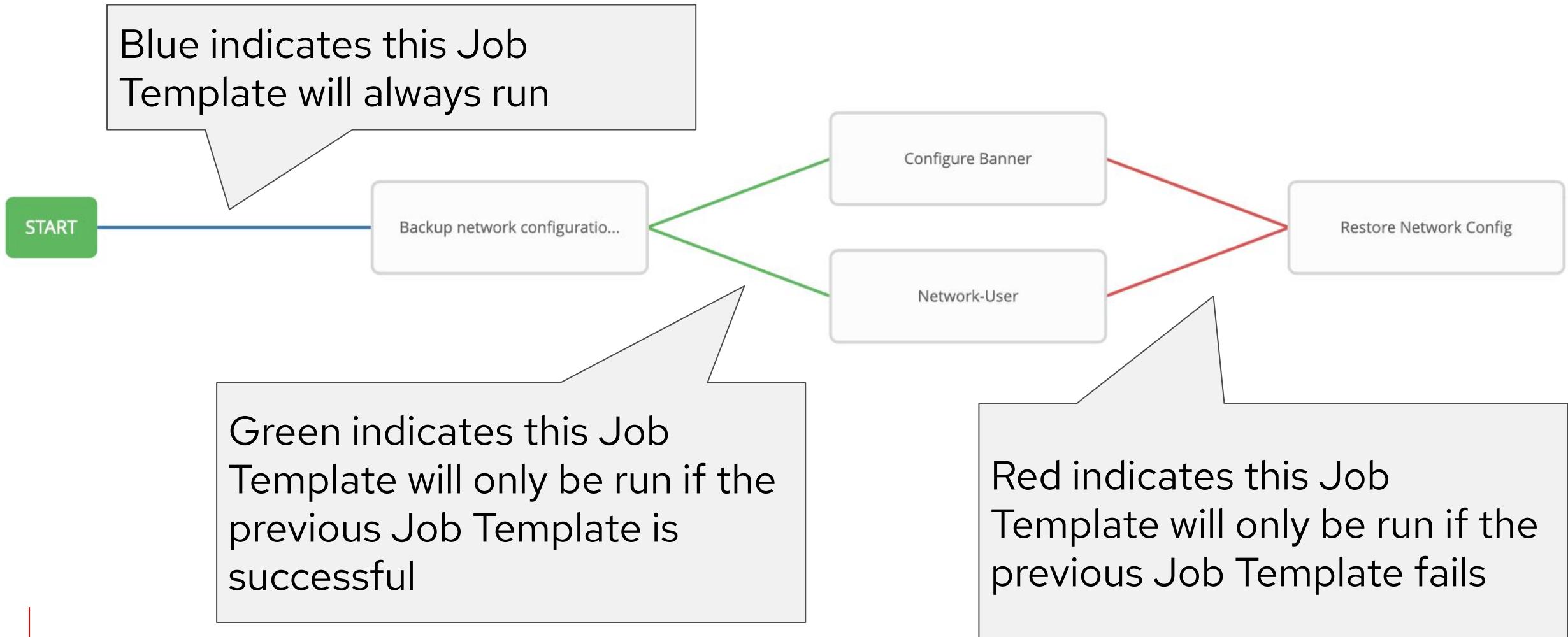
# Workflow Visualizer

The workflow visualizer will start as a blank canvas.



# Visualizing a Workflow

Workflows can branch out, or converge in.

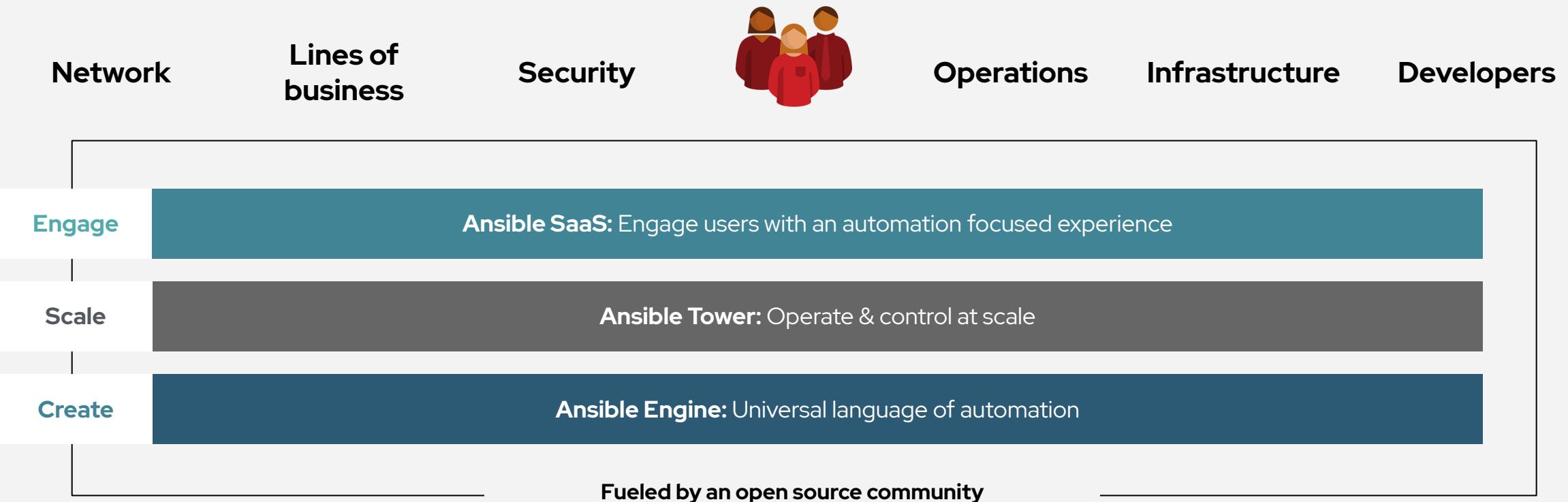


# Extras in Ansible Automation Platform



**Red Hat**  
Ansible Automation  
Platform

# Red Hat Ansible Automation Platform



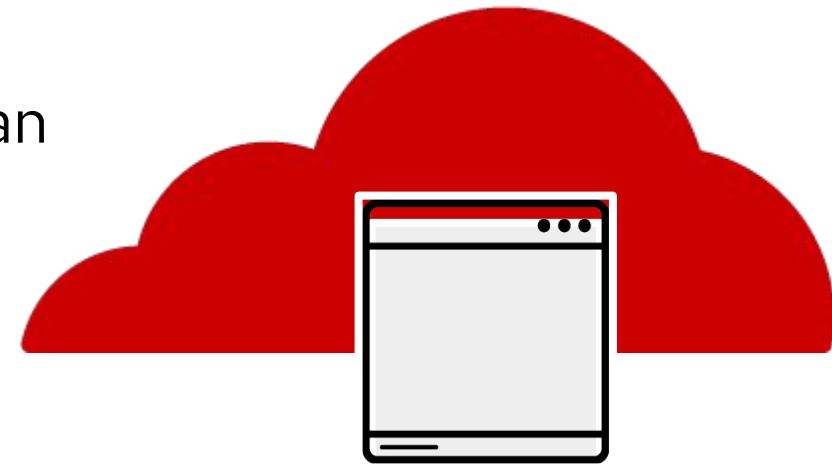
# Automation Analytics: What is it?

SaaS (Software as a Service) on  
[cloud.redhat.com](https://cloud.redhat.com)

Analytics for all Ansible Tower clusters for an organization

Includes:

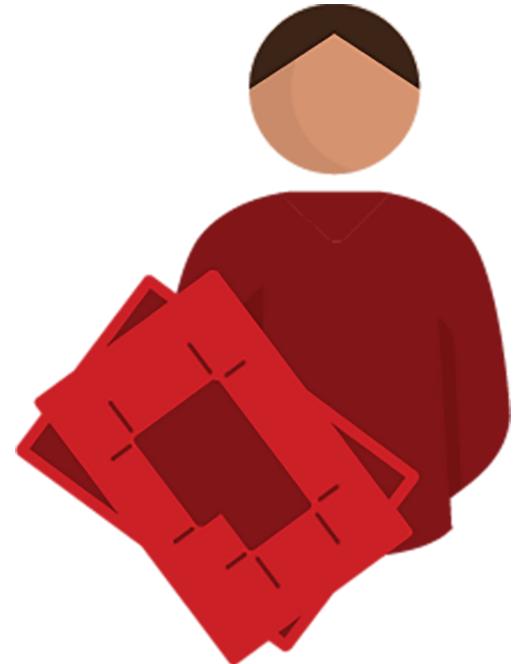
- visual dashboard
- health notifications
- organization statistics



# Automation Analytics: What does it provide?

Enables an Automation Center of Excellence

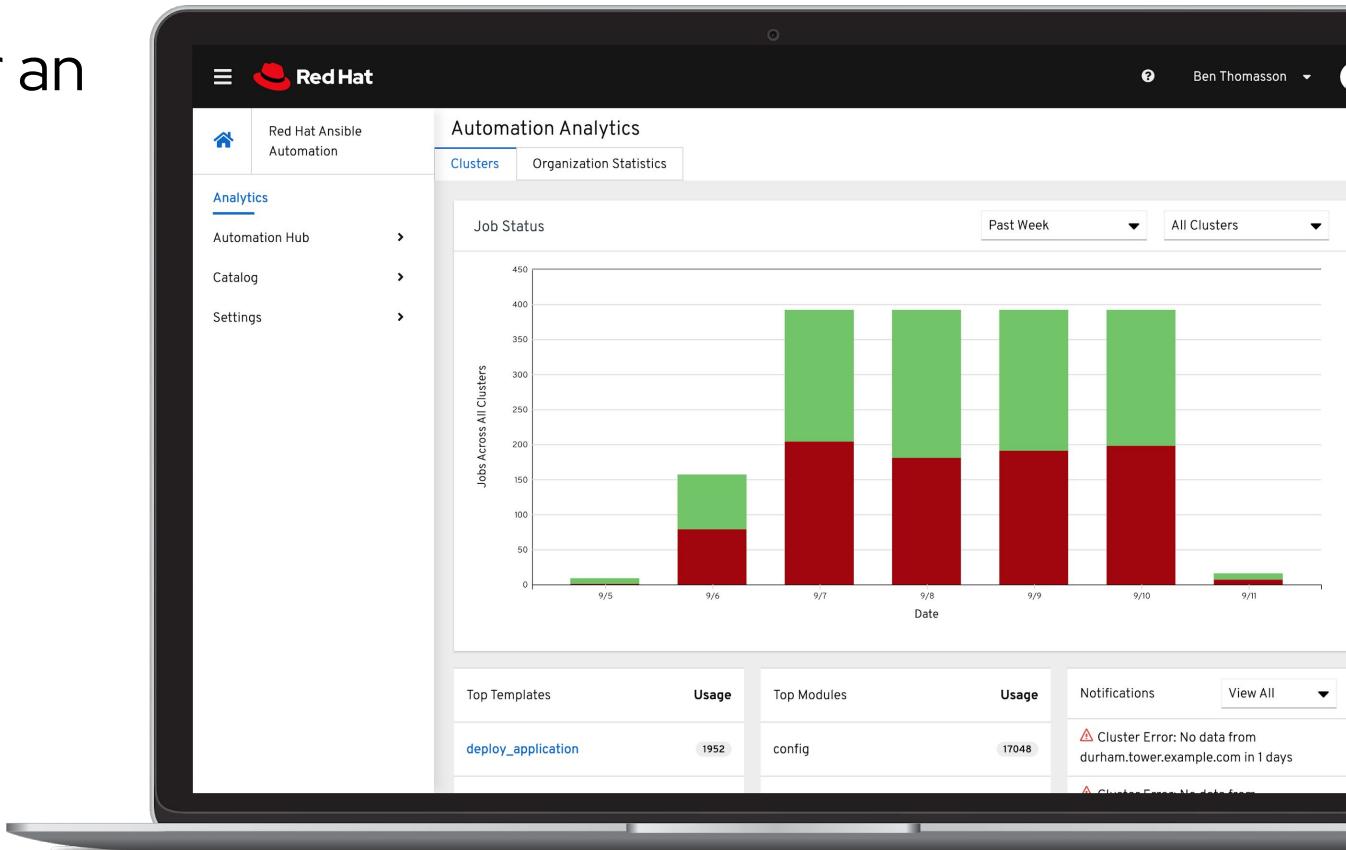
- View information about automation health, usage and performance across your enterprise.
- Gain information about automation in your enterprise:
  - Which organizations are using the most automation?
  - Utilization rates
  - Enterprise-wide success and failure rates for automation



# Analytics dashboard

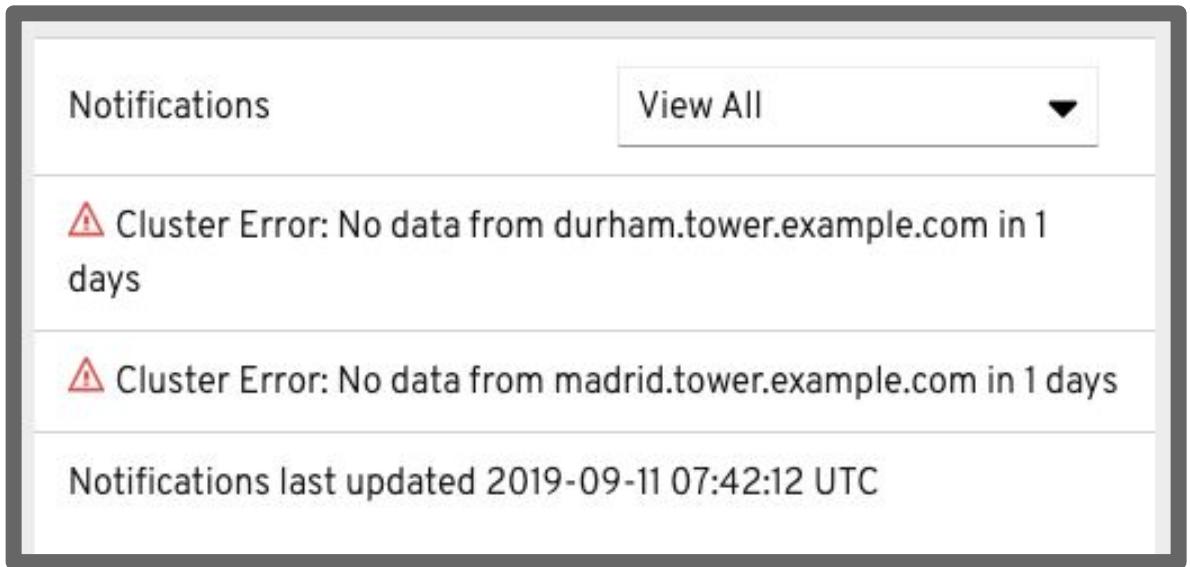
Information across all clusters for an enterprise:

- 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



The screenshot shows a 'Notifications' page with a header 'Notifications' and a dropdown 'View All'. There are two error messages listed:

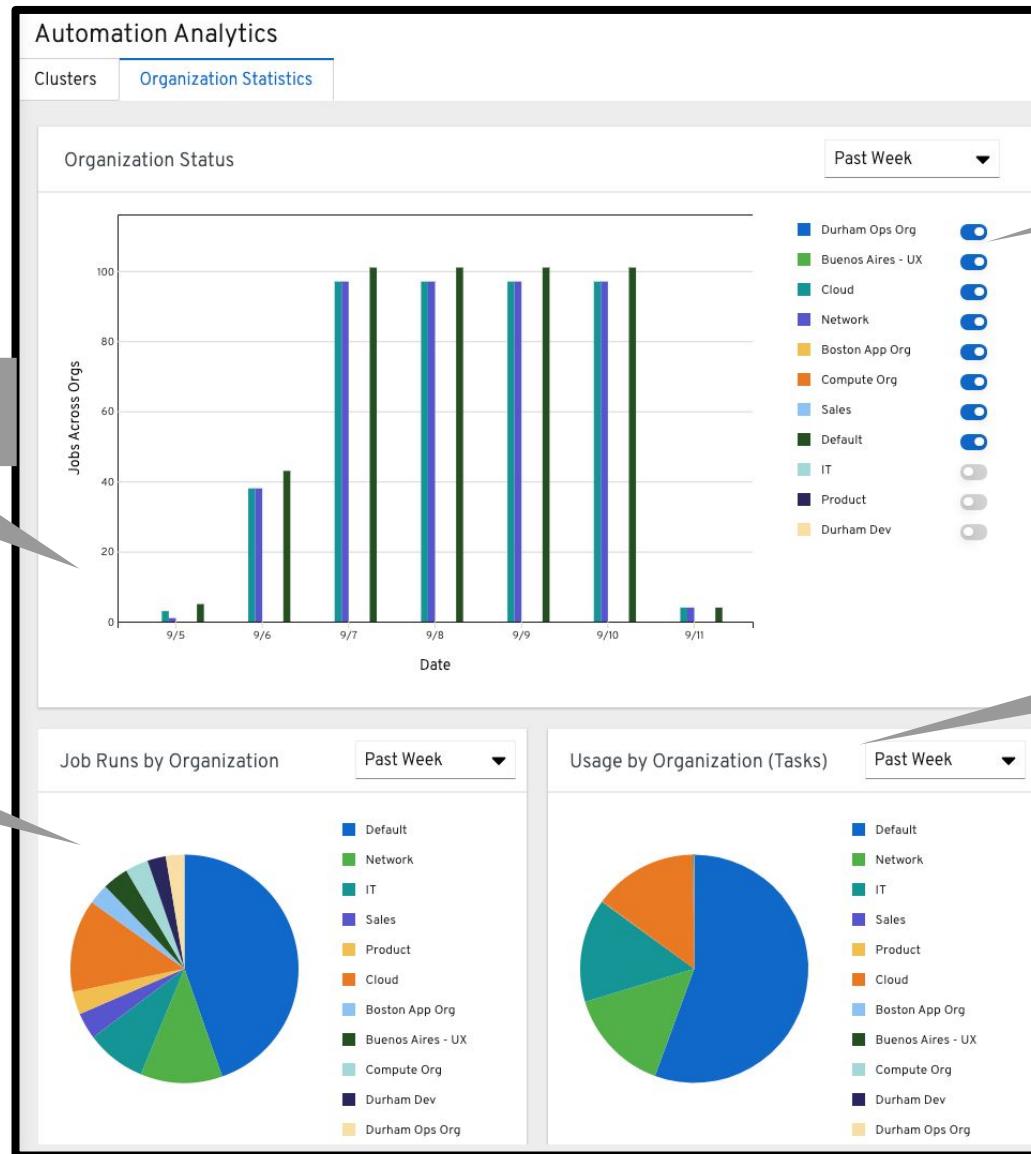
- ⚠ Cluster Error: No data from durham.tower.example.com in 1 days
- ⚠ Cluster Error: No data from madrid.tower.example.com in 1 days

At the bottom, it says 'Notifications last updated 2019-09-11 07:42:12 UTC'.

# Organizational statistics

Job Status by Organization

Job Runs by Organization



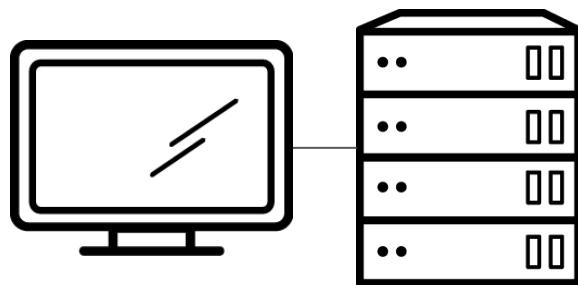
Filter by Organization

Usage by Organization

# Dashboard comparison

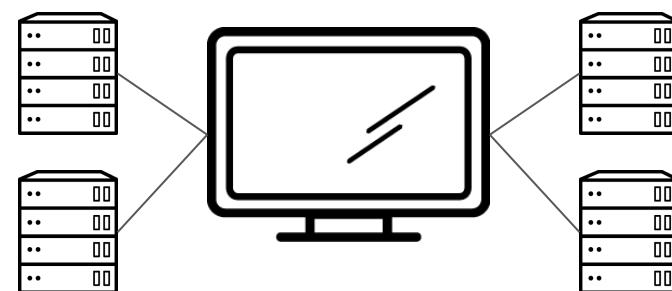
## Ansible Tower

- Recent job templates
- No module data
- One cluster



## Automation Analytics

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



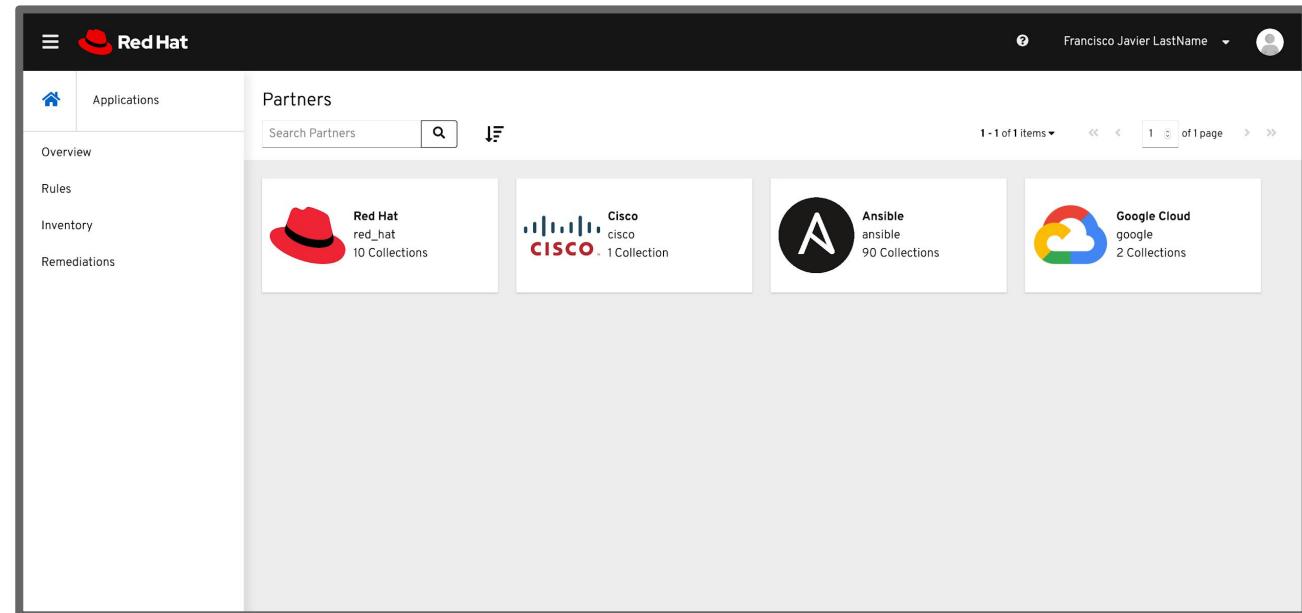
# Automation Hub

## 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.\*



# Demo: Ansible Tower within Azure



**Red Hat**  
Ansible Automation  
Platform

# Q&A/Break + Lab time! (link to lab)



**Red Hat**  
Ansible Automation  
Platform

# Next Steps

## GET STARTED

[ansible.com/get-started](https://ansible.com/get-started)

[ansible.com/tower-trial](https://ansible.com/tower-trial)

---

## WORKSHOPS & TRAINING

[ansible.com/workshops](https://ansible.com/workshops)

[Red Hat Training](#)

## JOIN THE COMMUNITY

[ansible.com/community](https://ansible.com/community)

---

## SHARE YOUR STORY

[Follow us @Ansible](#)

[Friend us on Facebook](#)

# Thank you!



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



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



[facebook.com/ansibleautomation](https://facebook.com/ansibleautomation)



[twitter.com/ansible](https://twitter.com/ansible)



[github.com/ansible](https://github.com/ansible)