



# Mainline Corporate Overview

Troy Webb  
Red Hat Brand Executive

The Technology Partner for Business Results

## Mainline At A Glance

Founded in **1989**

Headquarters:  
**Tallahassee, FL**

# Mainline

- Over **400** national employees who hold nearly **1,000** product and technical certifications

- Over **40+** software resources
- Over **200+** service professionals

**\$1B** in revenue

**30** years experience in Infrastructure, Data Center, Software, and Consulting Services

- **Top** partnership levels with industry leading technology manufacturers & providers including IBM, HPE, Dell EMC, Red Hat, CA Broadcom, NetApp and many others

## Why Work with Mainline?

Expertise on  
multi-platform,  
multi-vendor  
environments

Trusted partner,  
advisor &  
advocate

Deep technical  
knowledge and  
committed to  
ongoing training

Provides  
comprehensive,  
end-to-end  
solutions

# Mainline

# Solution Confusion?



## Solutions



# Mainline Partnership Eco System!

## Partnerships

### Strategic



COHESITY



Hewlett Packard Enterprise



JUNIPER

NETSCAPE



redhat.

### Eco System

actifio



LUMINEX



ANW

Crayon



ARISTA

DalCorr



asg

ESTUATE



riverbed



A V I R E

FORTINET



techir2

BlueAlly

GALILEO



SanDisk



bmc

GTSoftware



veeam

BROCADE

HITACHI



Sas



Centrify

intel



simplivity

vmware

cITRIX

IntelliMagic



Zerto





# **Red Hat**

## Ansible Automation Platform

# Network Automation Workshop

Introduction to Ansible for network engineers and operators



**Red Hat**



## Hi! I'm Laine Vyvyan.

- I'm a Channel Solutions Architect at Red Hat. I cover all of the Great Lakes states.
- I live in Lansing, MI.
- My favorite color is *glitter*.

[lvyyyan@redhat.com](mailto:lvyyyan@redhat.com)



@lainie\_ftw

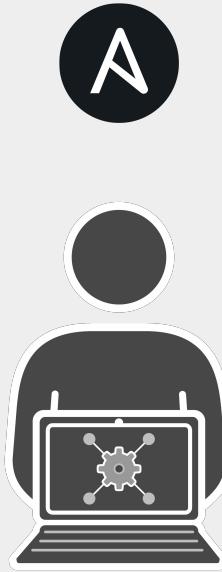


# Agenda

- Why Automate?
- How Ansible Works
- Commands, Plays, and Playbooks
- Roles
- Demo!



**Red Hat**  
Ansible  
Automation



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

# 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...**

Orchestration

Configuration Management

Application Deployment

Provisioning

Continuous Delivery

Security and Compliance

**On these...**

Firewalls

Load Balancers

Applications

Containers

Clouds

Servers

Infrastructure

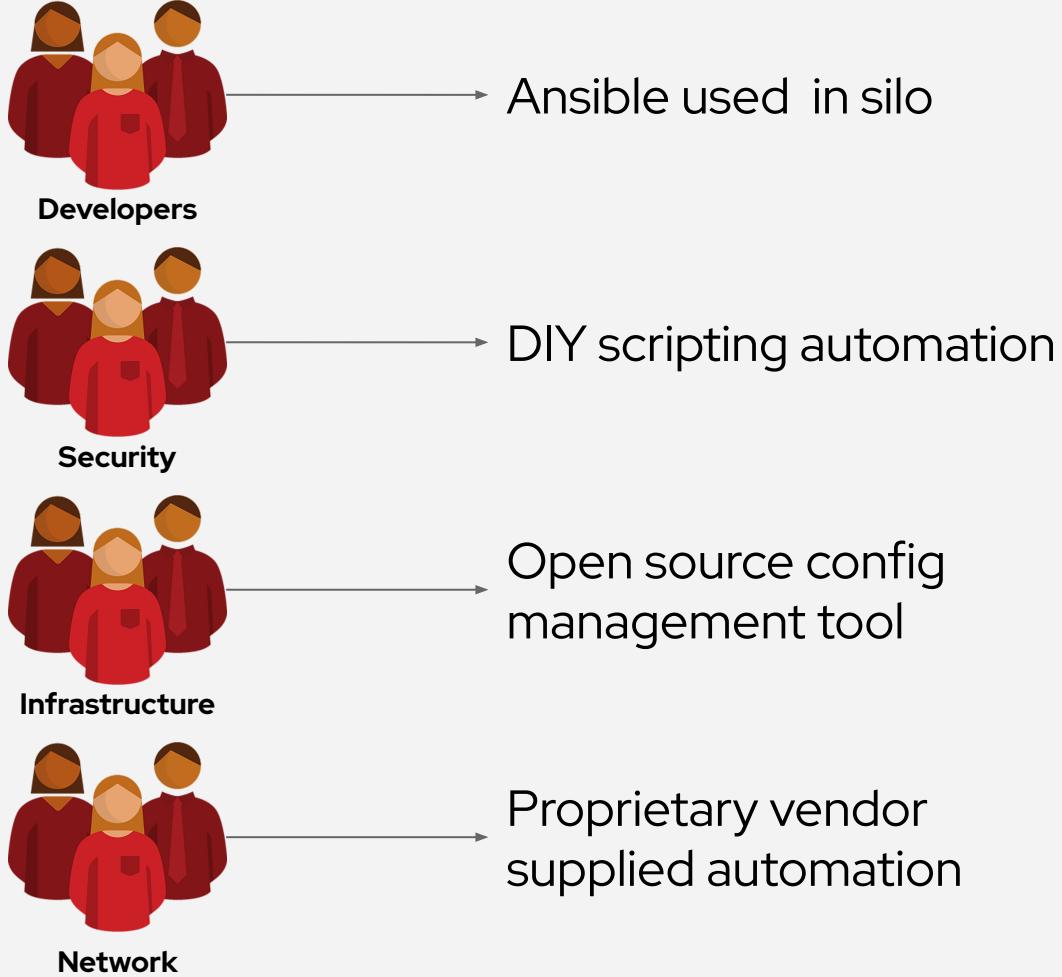
Storage

Network Devices

**And more...**

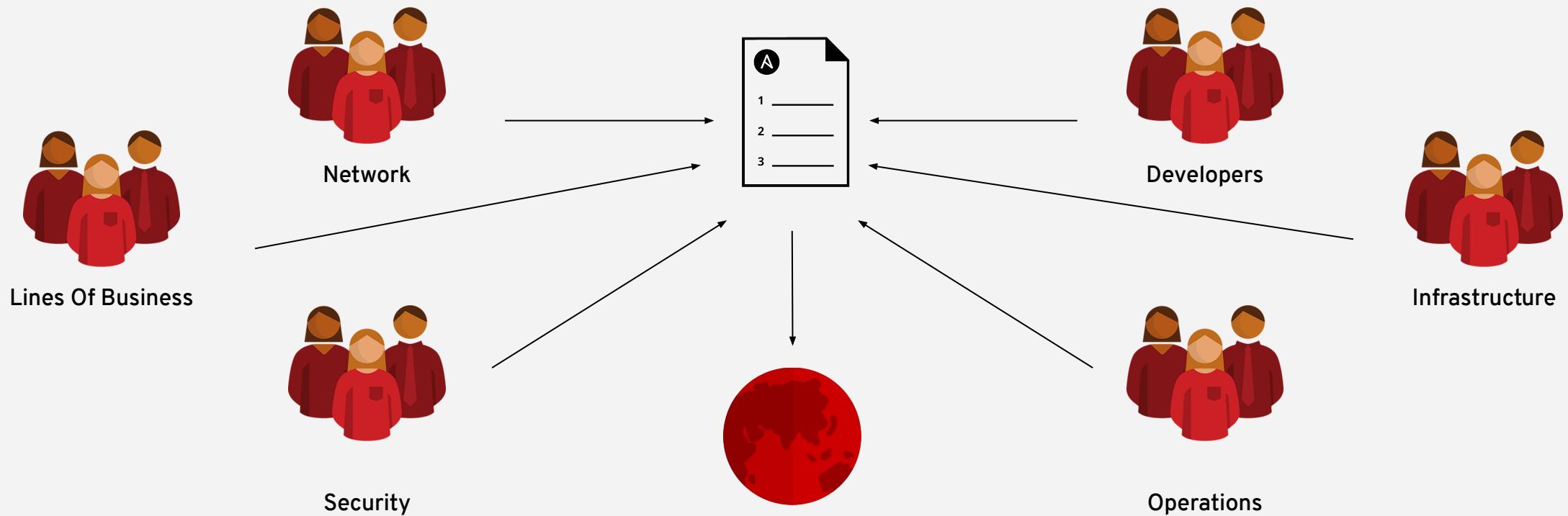


# Ad-hoc Automation is happening in silos

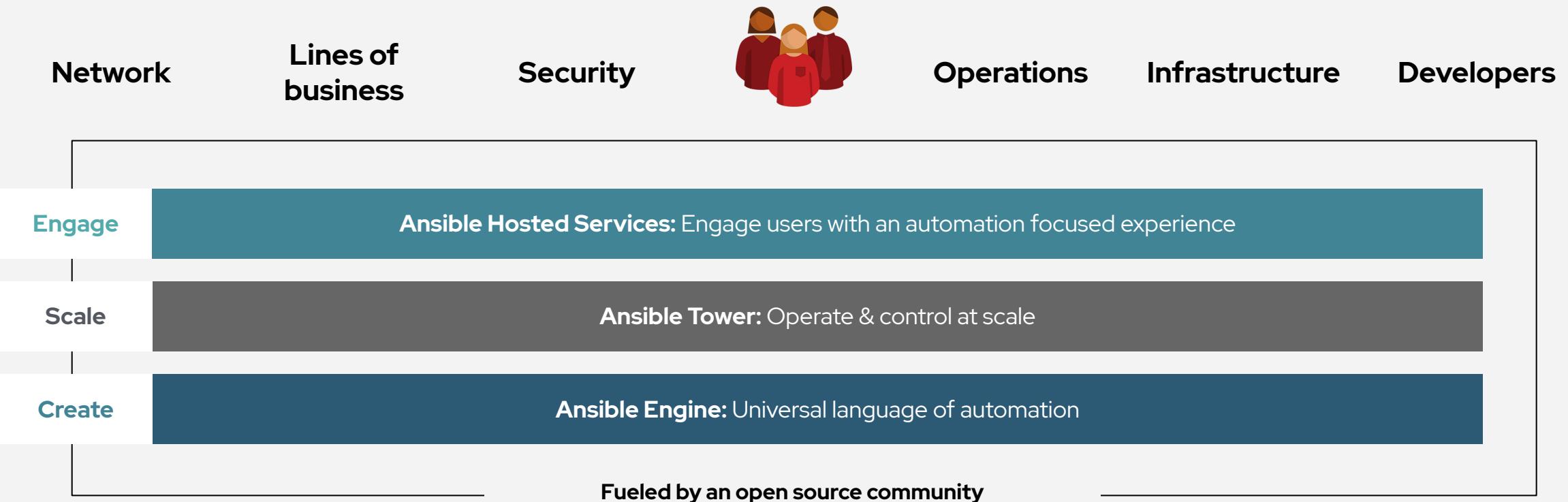


Is ad-hoc  
automation enough?

# 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)





**Red Hat**  
Ansible Automation  
Platform

Ansible Automation Platform for  
**NETWORK AUTOMATION**

A photograph of a man with short brown hair and a beard, seen from the back and side. He is wearing a dark t-shirt and is seated at a desk, working on a laptop computer. He is positioned in front of a large server rack filled with various network hardware. The background shows more server racks in a dimly lit room.

**71%**

of networks are still  
driven manually via CLI

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

# Why Ansible for Networking?

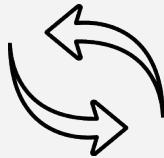


## Simple

For operators, not developers

Download and go

Reuse of existing knowledge



## Powerful

Connect via plugins

Easy platform enablement

Leverage Linux tools



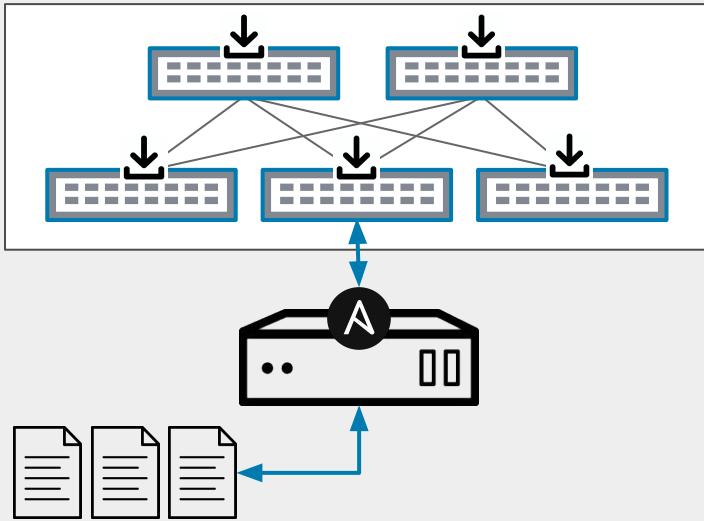
## Agentless

Ideal for network gear

No agents to exploit or update

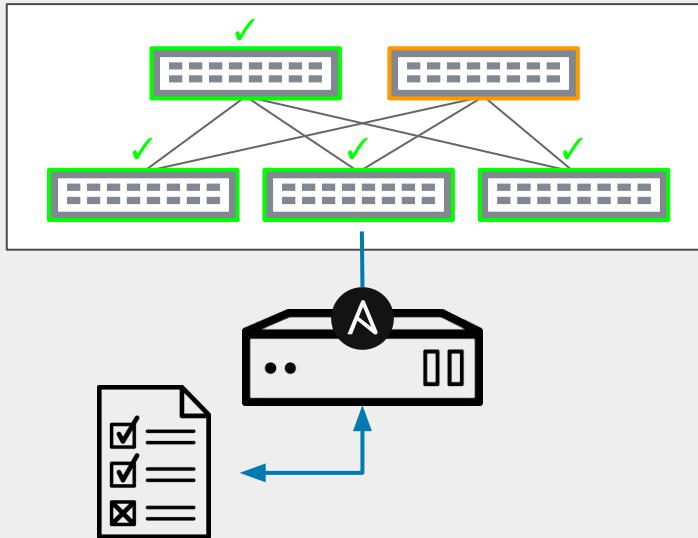
Standards-based SSH

# Common use cases



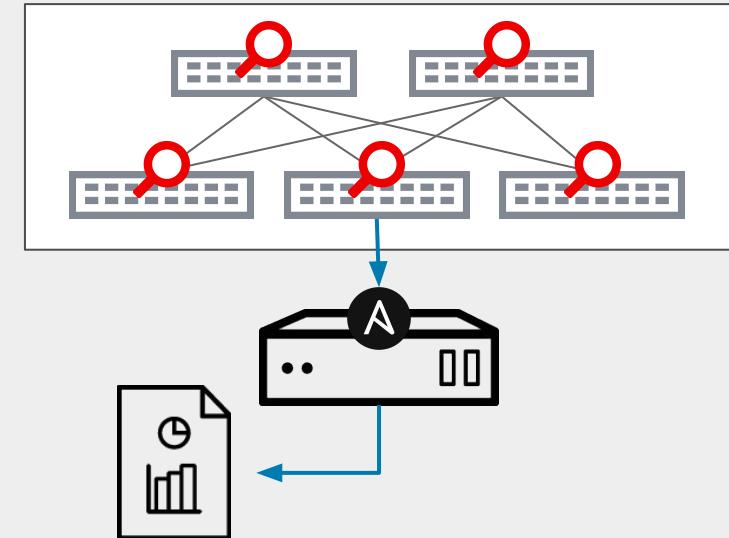
## Backup and Restore

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



## Configuration Compliance

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



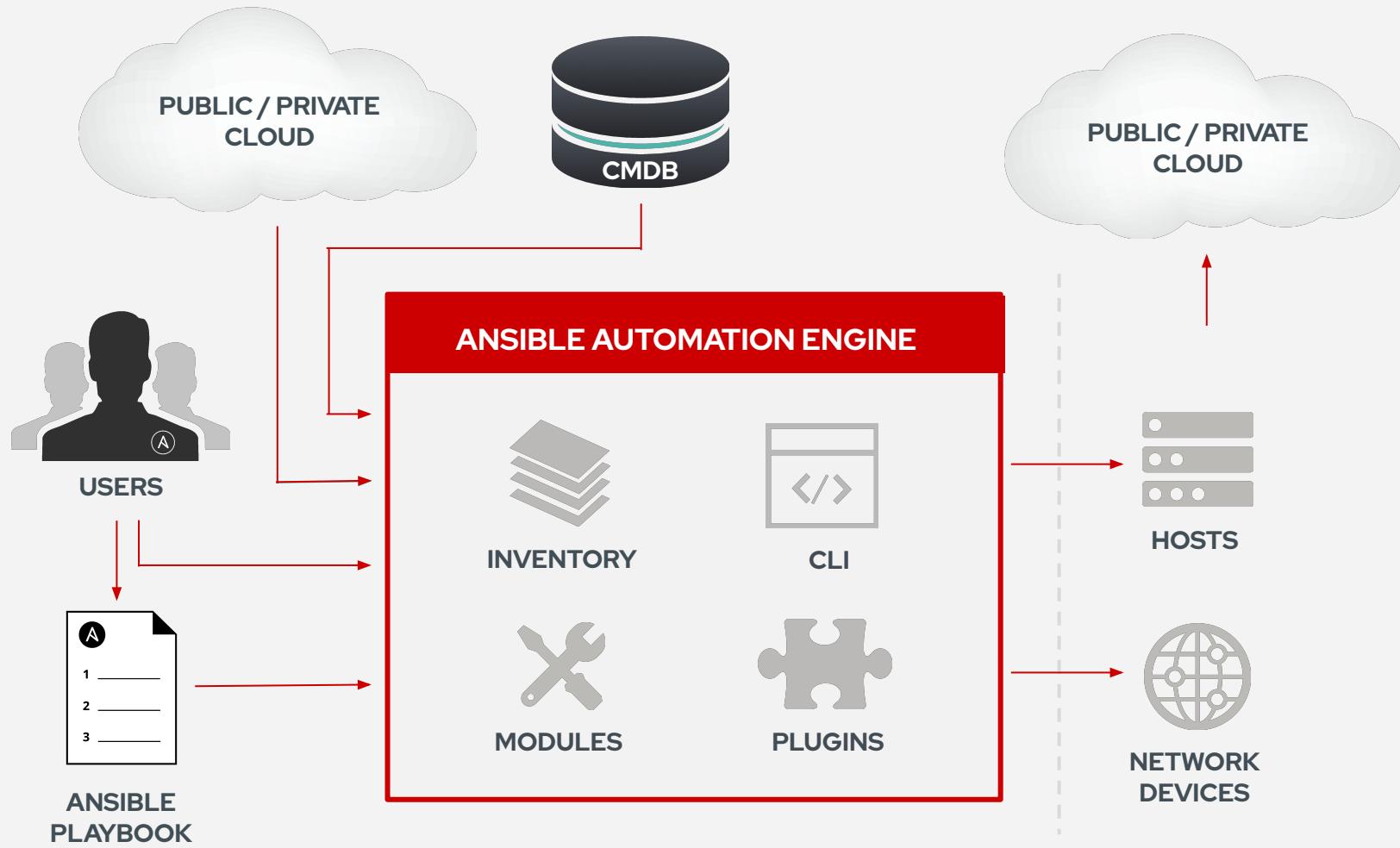
## Dynamic Documentation

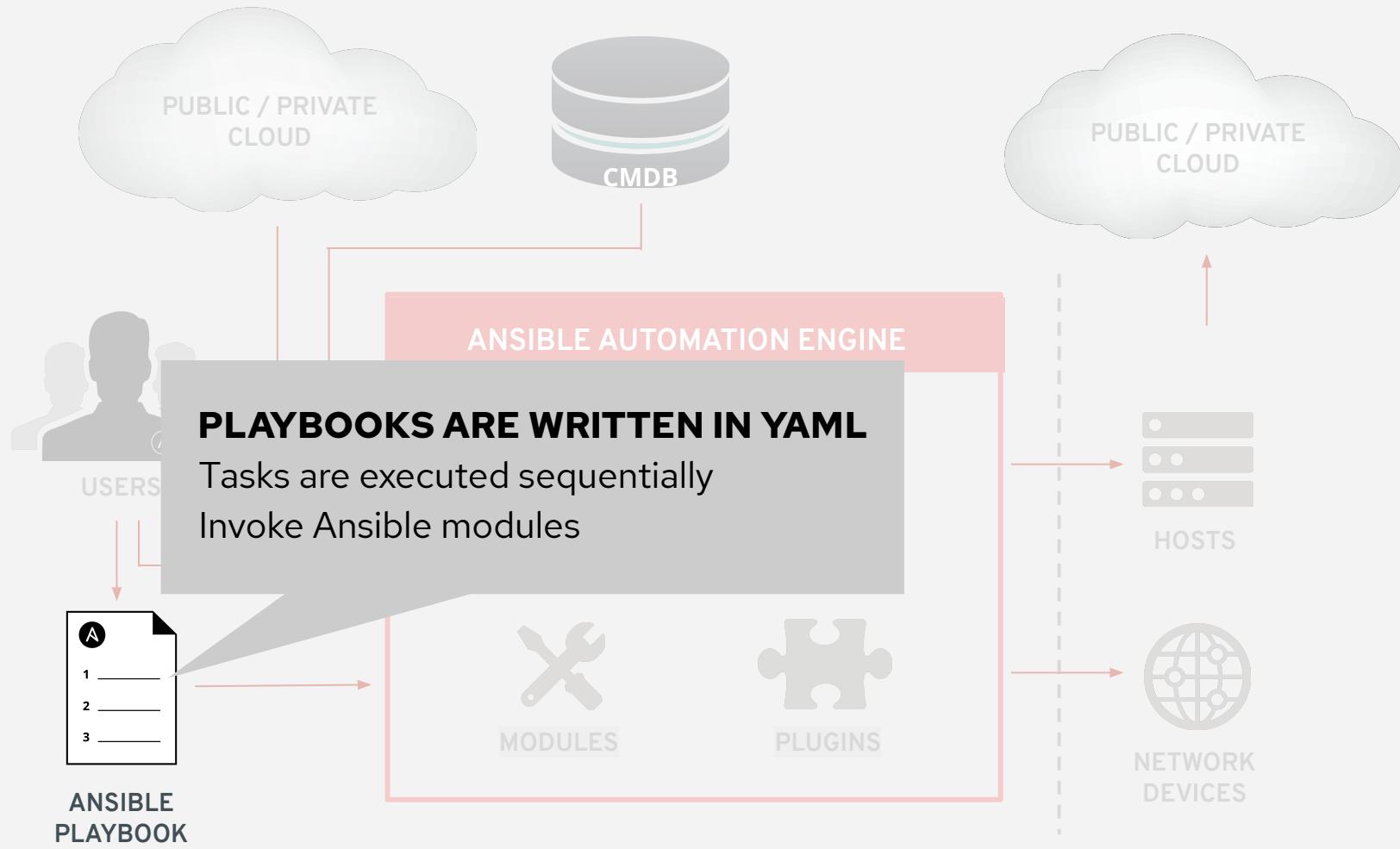
- Build reports
- Grab software versions, MTU, interfaces status
- Audit system services and other common config

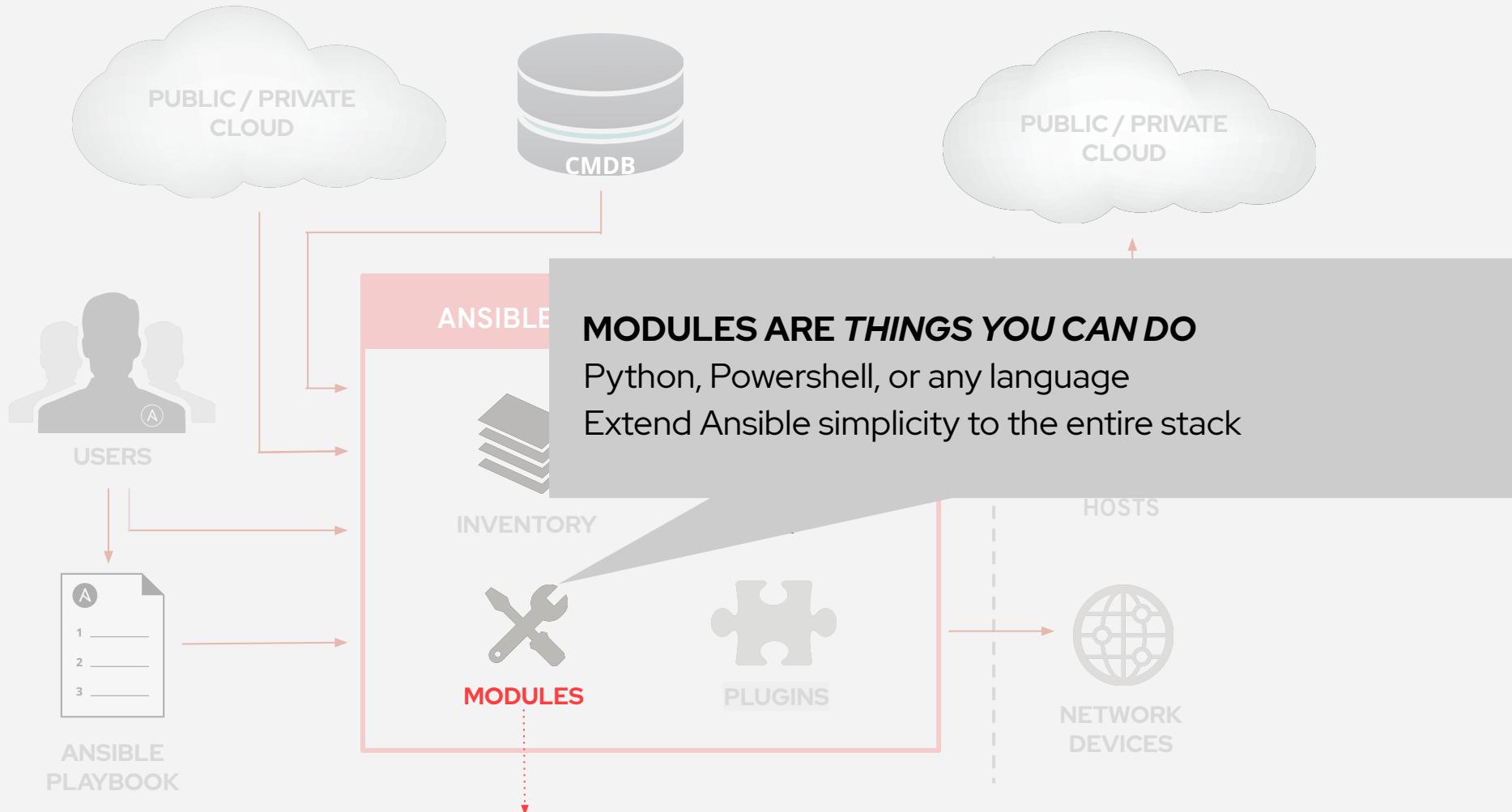
# How Ansible Works



**Red Hat**  
Ansible  
Automation







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

# ANSIBLE NETWORK AUTOMATION

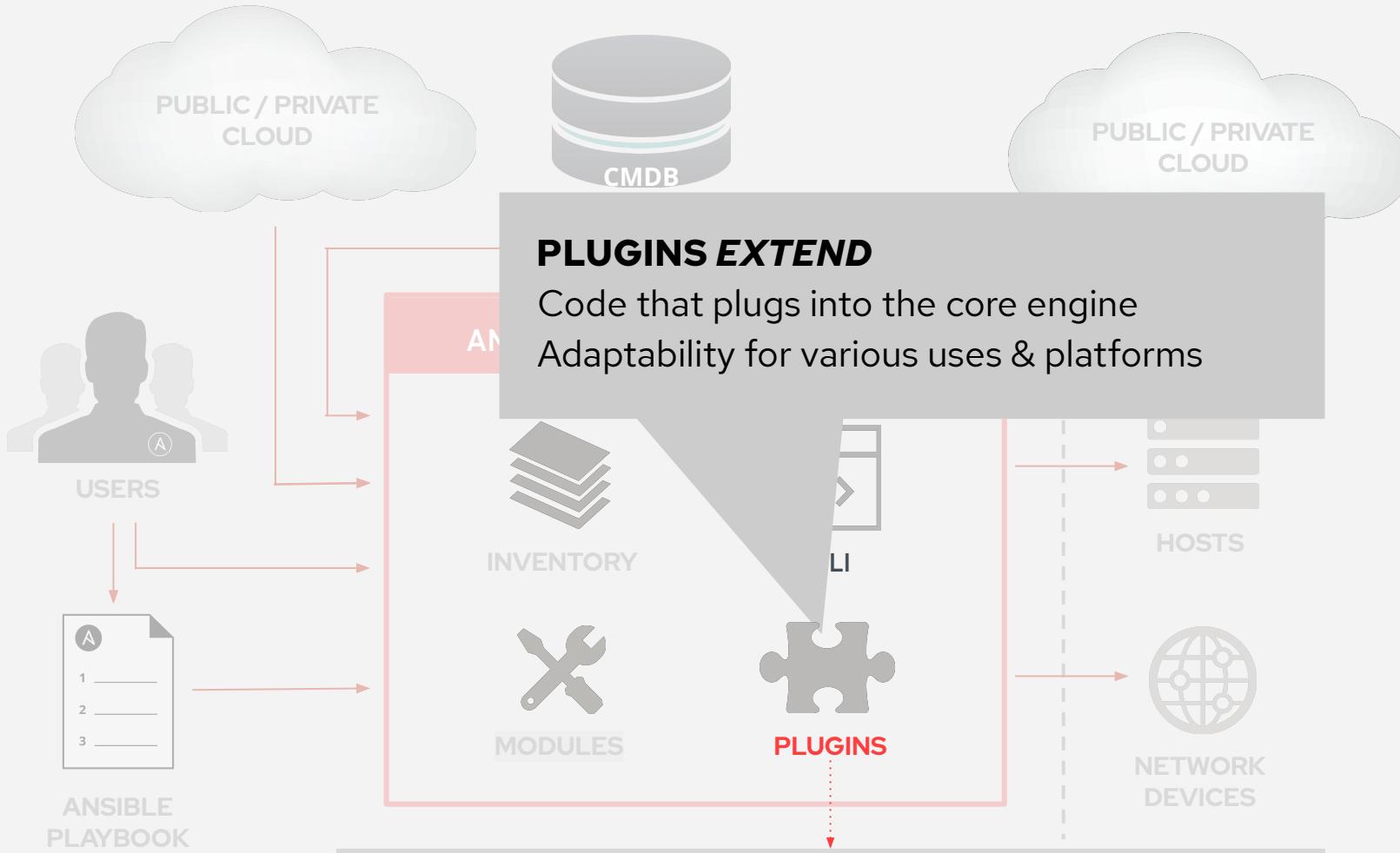
**65+**

Network  
Platforms

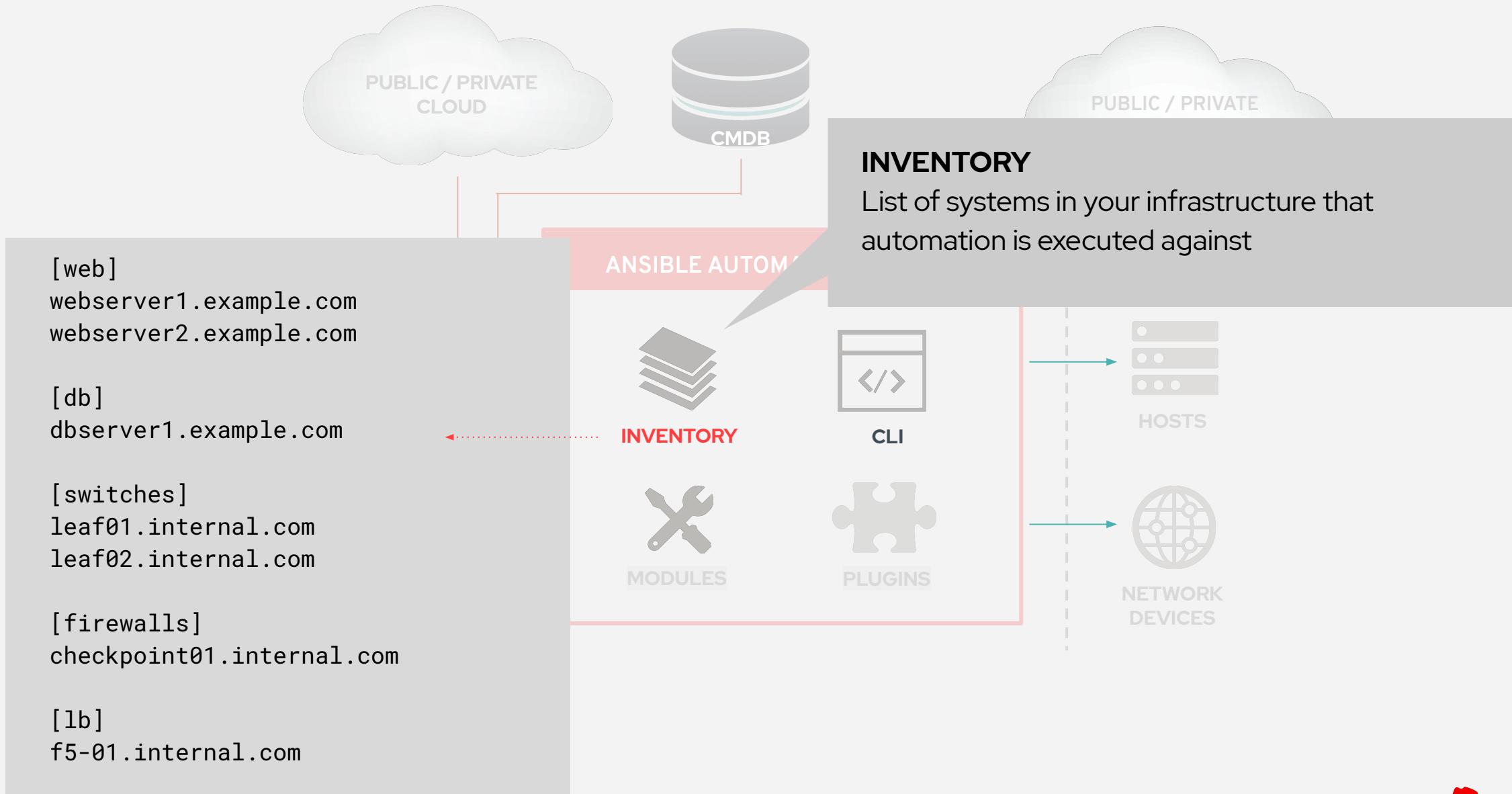
**1000+**

Network  
Modules

[ansible.com/for/networks](http://ansible.com/for/networks)



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



# Understanding Inventory

```
rtr1 ansible_host=18.220.156.59
rtr2 ansible_host=18.221.53.11
rtr3 ansible_host=13.59.242.237
rtr4 ansible_host=3.16.82.231
rtr5
rtr6
```

# Understanding Inventory - Groups

There is always a group called "**all**" by default

```
[cisco]
rtr1 ansible_host=18.220.156.59 private_ip=172.16.184.164
[arista]
rtr2 ansible_host=18.221.53.11 private_ip=172.17.229.213
rtr4 ansible_host=3.16.82.231 private_ip=172.17.209.186
[juniper]
rtr3 ansible_host=13.59.242.237 private_ip=172.16.39.75
```

Groups can be nested

```
[routers:children]
cisco
juniper
arista
```

# Understanding Inventory - Variables

Host variables apply to the host and override group vars

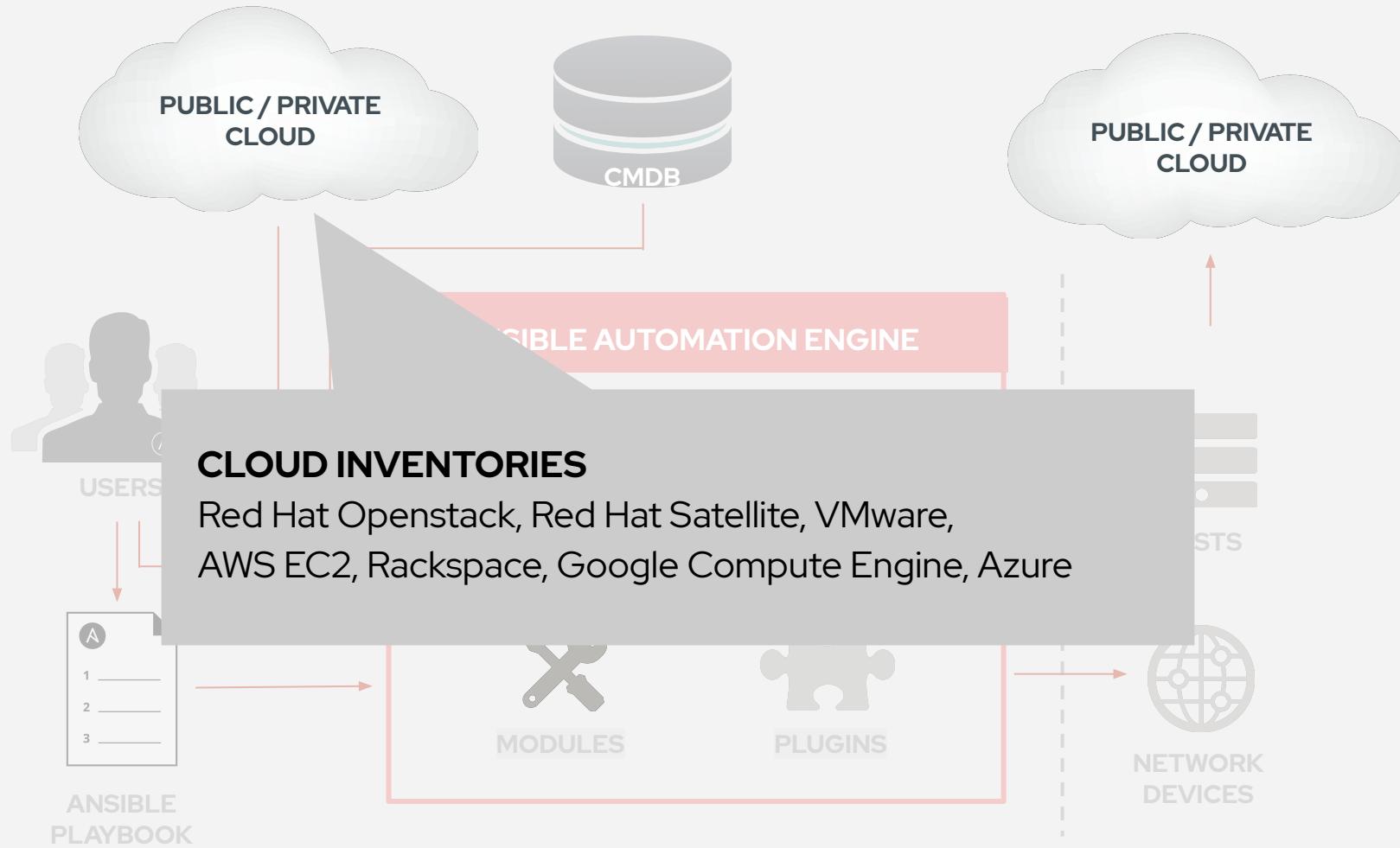
```
[cisco]
rtr1 ansible_host=52.14.208.176 private_ip=172.16.59.243
```

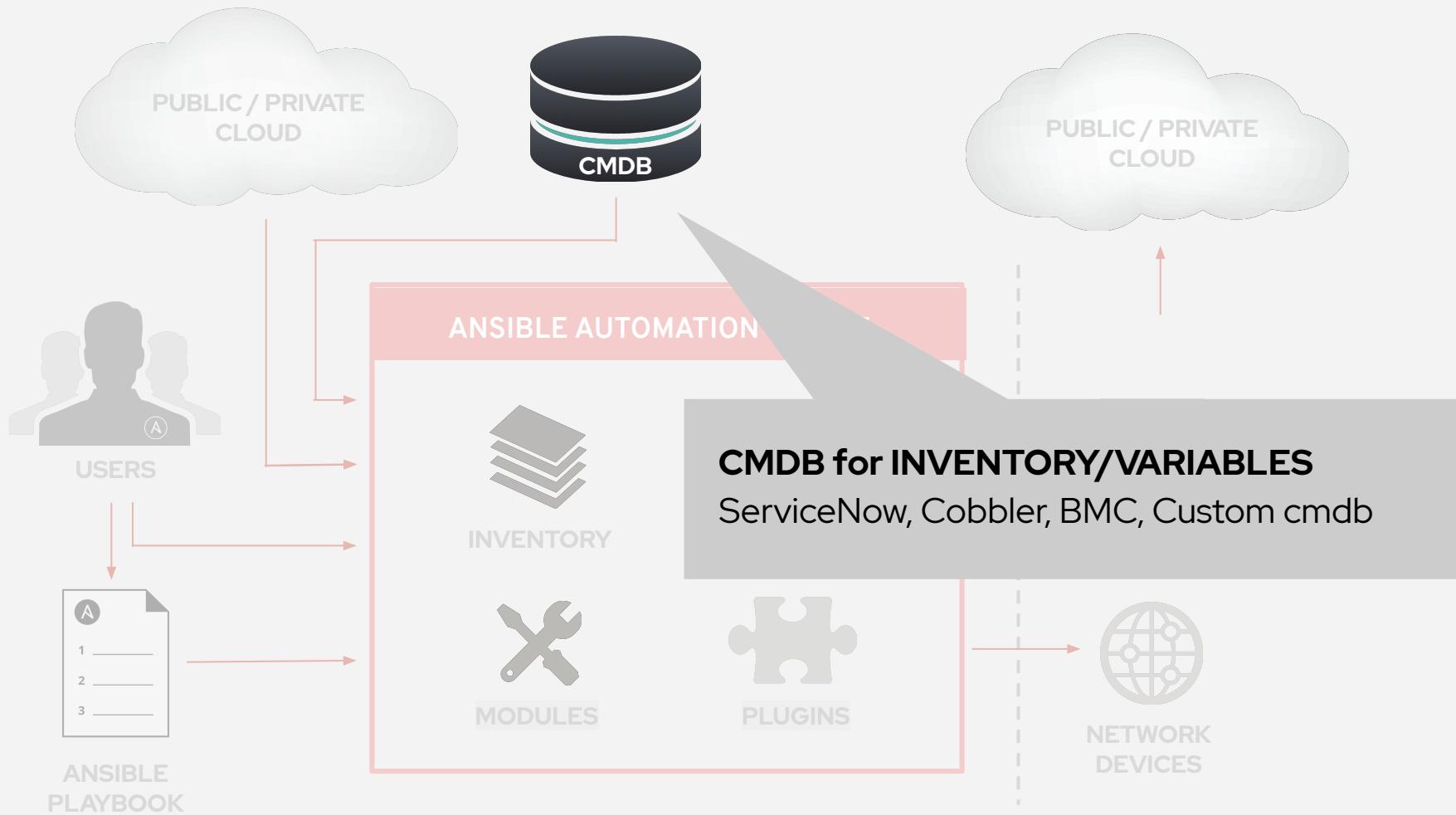
```
[arista]
rtr2 ansible_host=18.221.195.152 private_ip=172.17.235.51
rtr4 ansible_host=18.188.124.127 private_ip=172.17.43.134
```

```
[juniper]
rtr3 ansible_host=3.15.11.56 private_ip=172.16.94.233
```

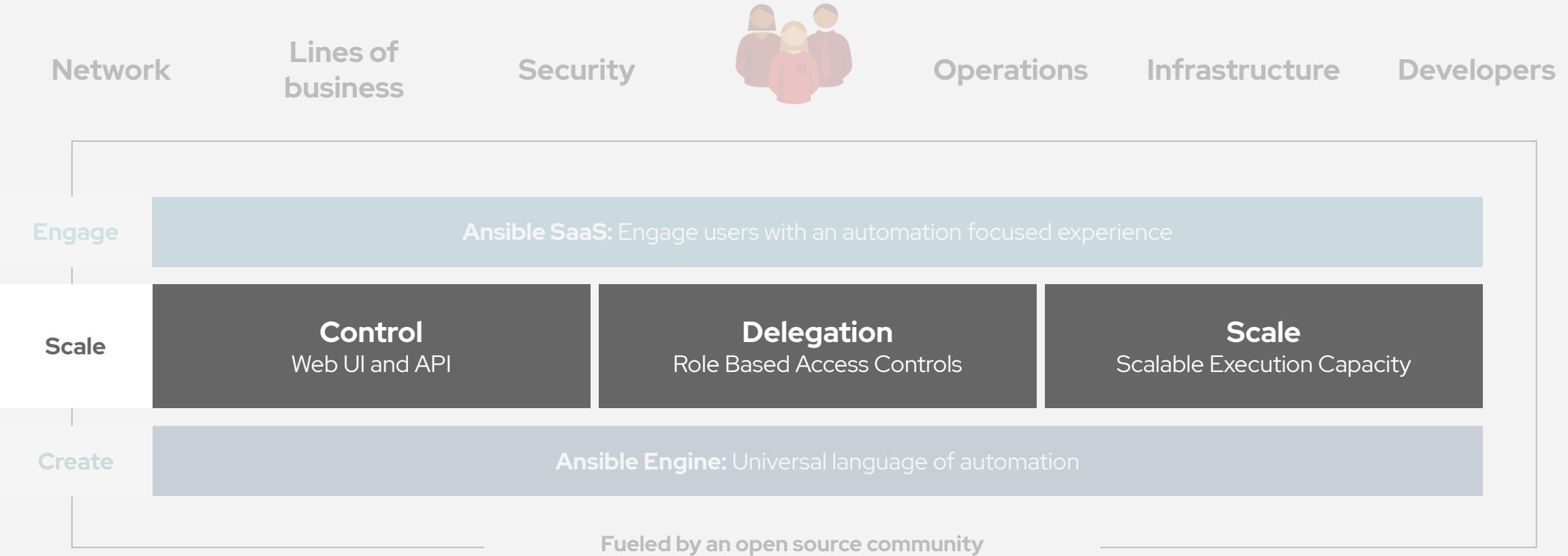
```
[cisco:vars]
ansible_user=ec2-user
ansible_network_os=ios
ansible_connection=network_cli
```

Group variables apply for all devices in that group





# 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



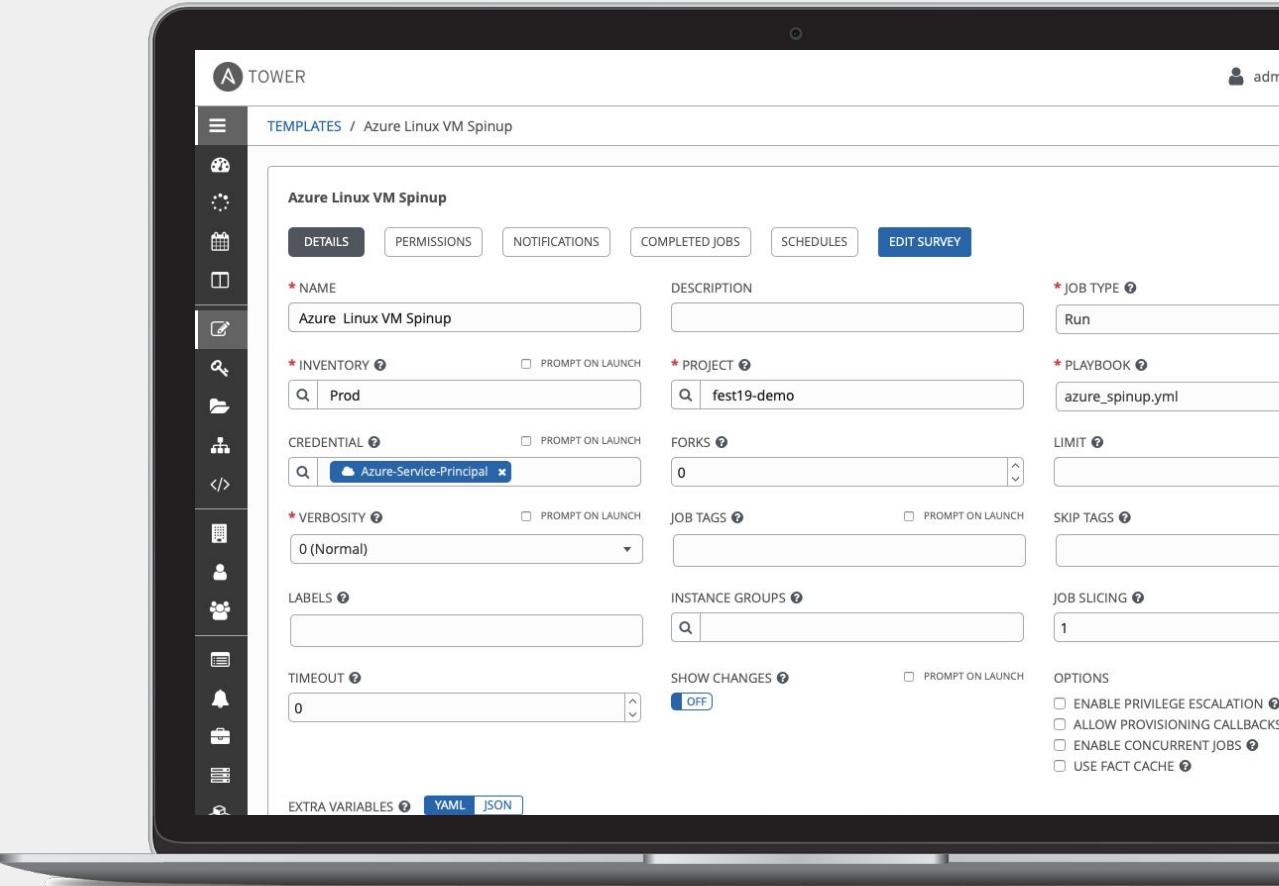
# Job Templates

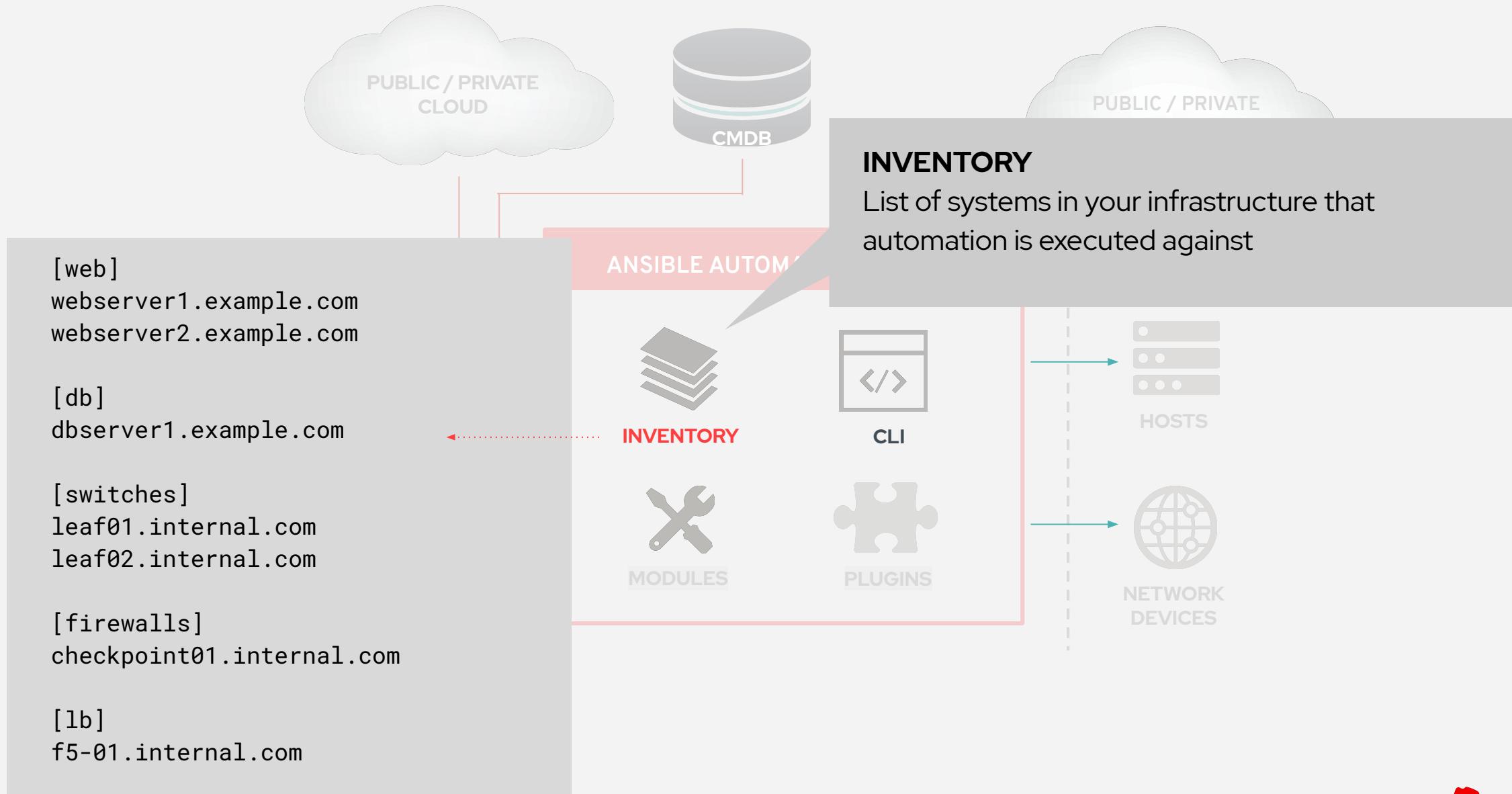
Everything in Ansible Tower revolves around the Job Template. Job Templates allow Ansible Playbooks to be controlled, delegated and scaled for an organization.

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

An 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

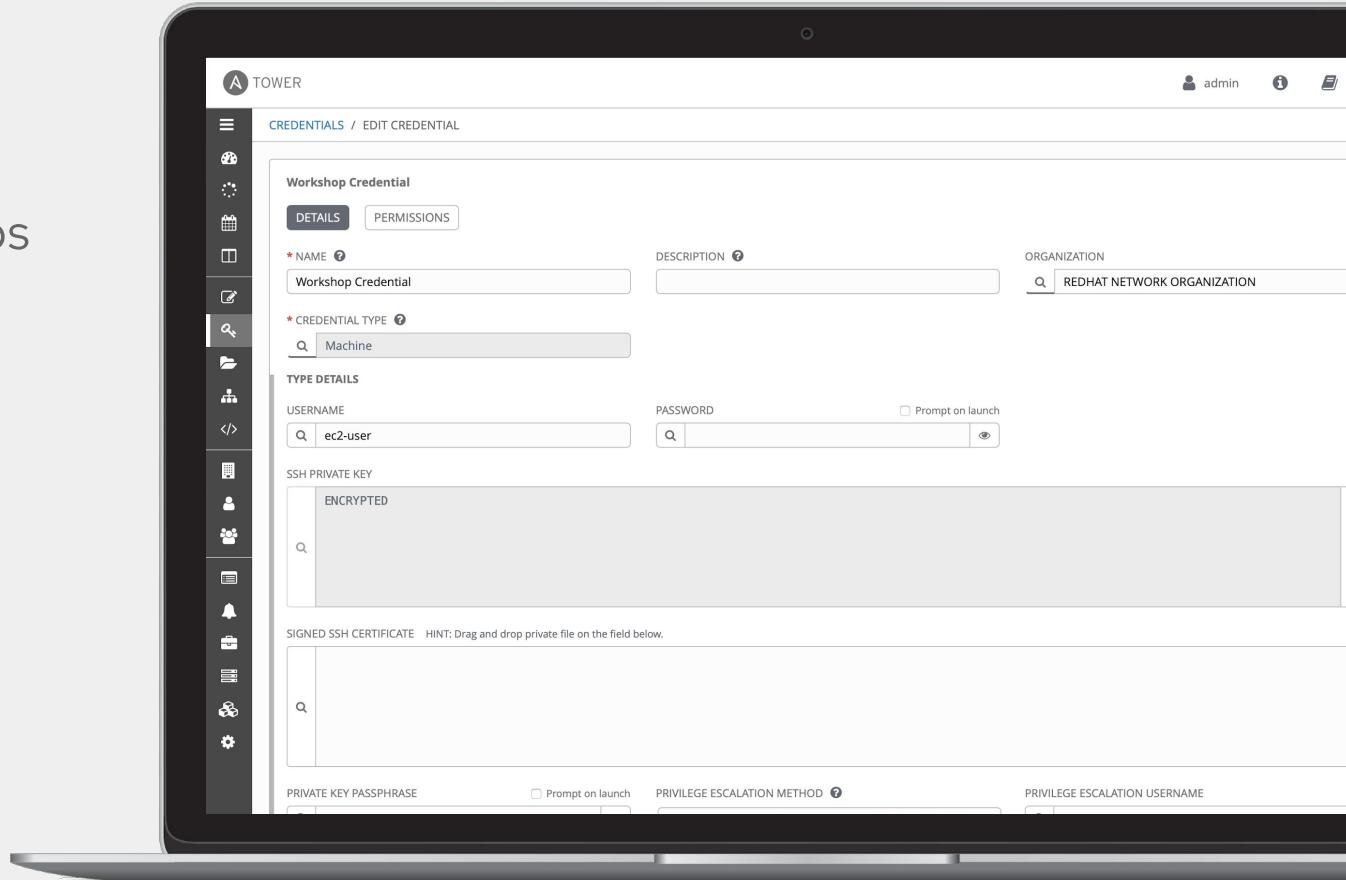
The screenshot shows the Ansible Tower web interface. The top navigation bar includes 'INVENTORIES / Workshop Inventory / HOSTS'. On the left is a sidebar with various icons for managing inventories, hosts, permissions, groups, sources, and completed jobs. The main content area displays the 'Workshop Inventory' hosts list. The 'HOSTS' tab is selected. The list shows five hosts: 'ansible', 'rtr1', 'rtr2', 'rtr3', and 'rtr4', each with an 'ON' status indicator and a 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 list is another search bar and a table header with columns 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

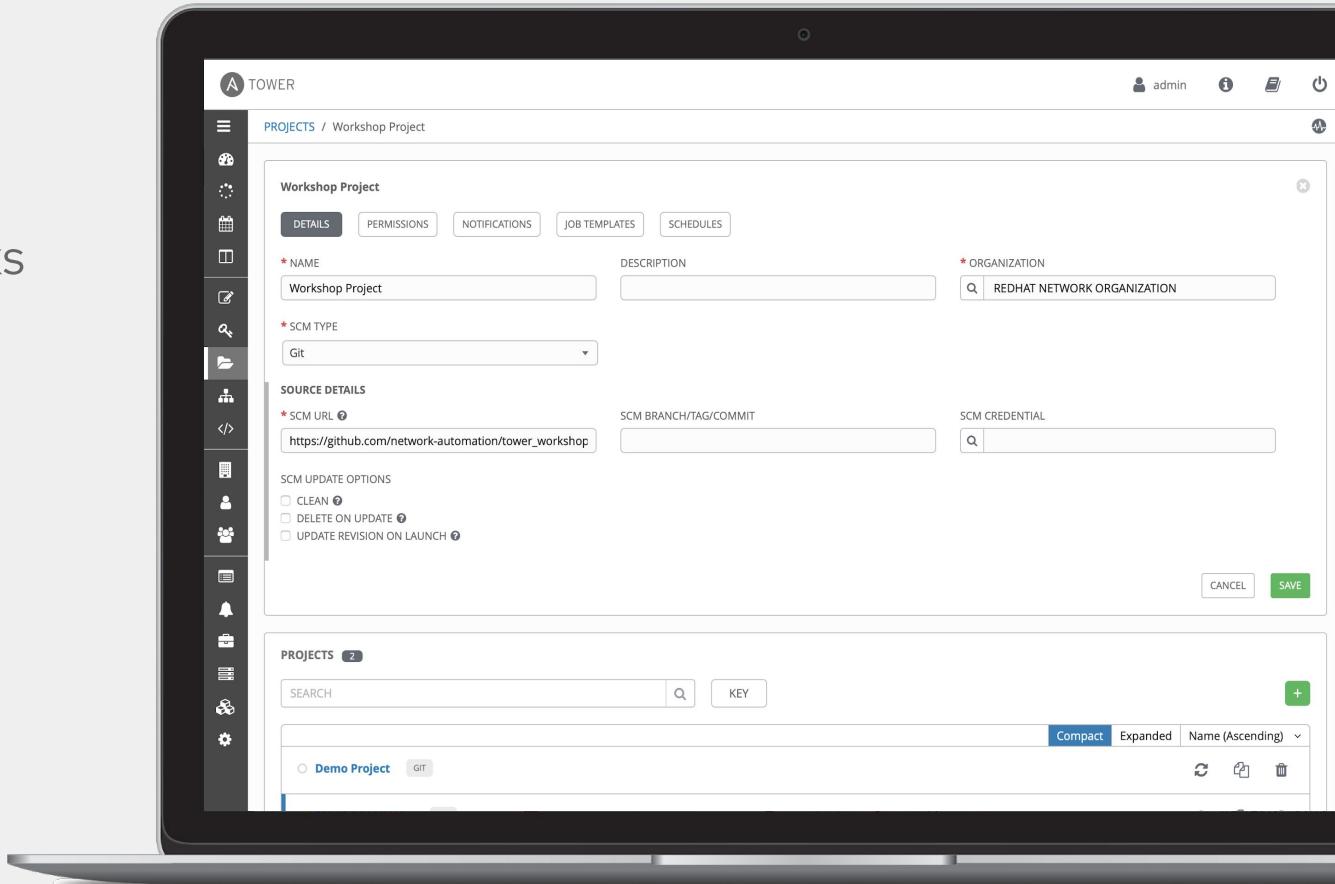
Centralized management of credentials allows users to leverage them without ever *seeing* them.



# Project

A project is a logical **collection** of Ansible Playbooks.

You can (and should!) manage Ansible Playbooks and Playbook directories by placing them in a source code management tool supported by Ansible Tower - including Git, Subversion, and Mercurial.



# Role Based Access Control (RBAC)

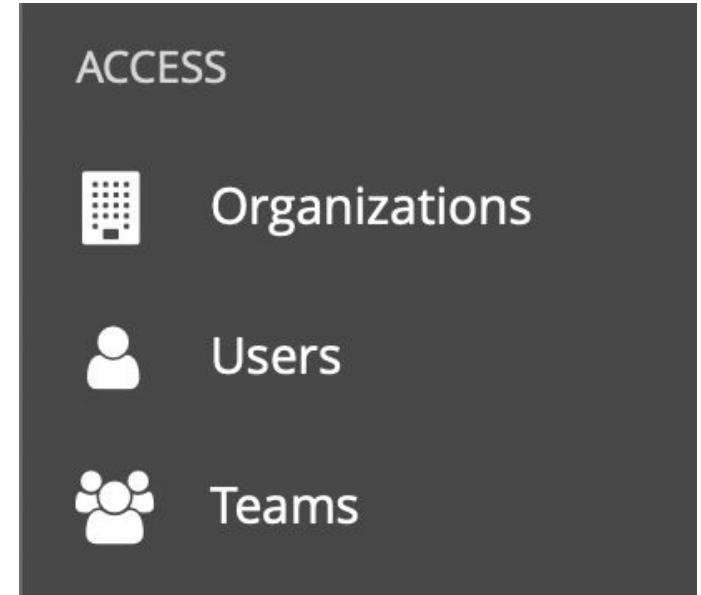
Role-Based Access Controls (RBAC) are built into Ansible Tower and allow administrators to delegate access to inventories, organizations, and more.



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.



# 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.

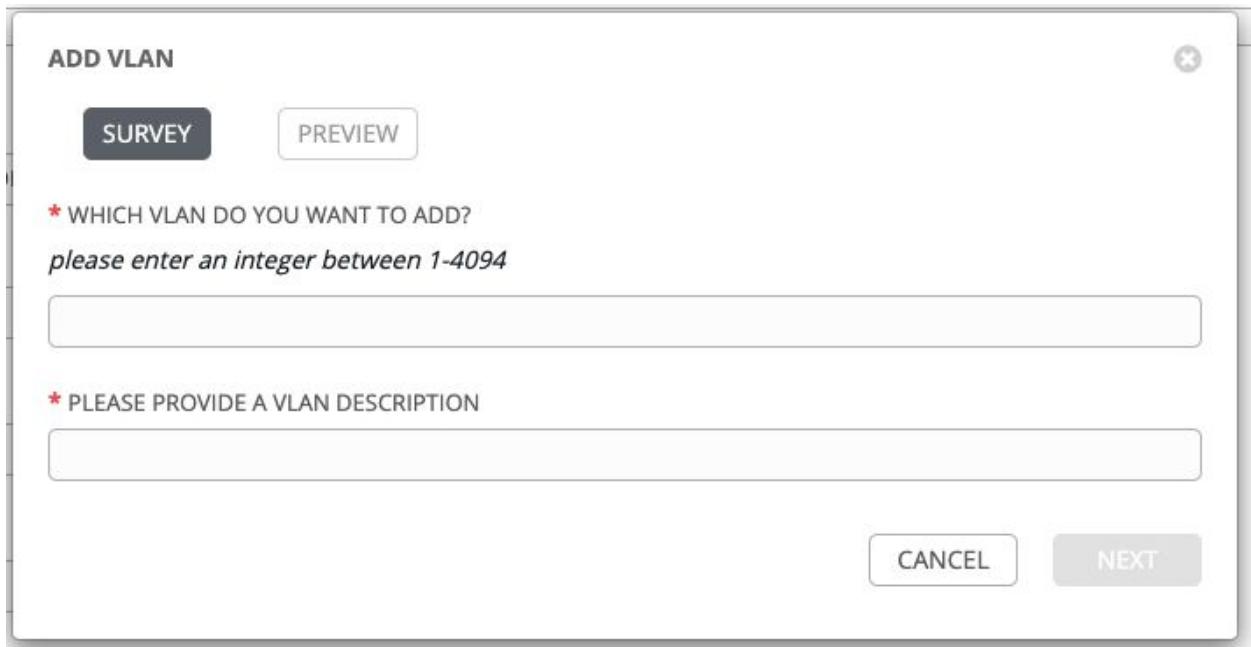
ADD VLAN

SURVEY PREVIEW

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

\* PLEASE PROVIDE A VLAN DESCRIPTION

CANCEL NEXT



# Creating a Survey (1/2)

Once a Job Template is saved, the **Add Survey Button** will appear

**ADD SURVEY**

Click the button to open the Add Survey window.

The screenshot shows the Ansible Tower web interface. On the left is a dark sidebar with navigation links: Views, Dashboard, Jobs, Schedules, My View, Resources, Templates (which is selected), Credentials, Projects, Inventories, Inventory Scripts, Access, and Organizations. The main content area has a title 'TEMPLATES / Configure Banner'. A modal window titled 'Configure Banner' is open. Inside the modal, there are several configuration sections: 'DETAILS' (selected), 'PERMISSIONS', 'NOTIFICATIONS', 'COMPLETED JOBS', 'SCHEDULES', and 'EDIT SURVEY' (which is highlighted with a red rectangle). Below these are fields for 'NAME' ('Configure Banner'), 'DESCRIPTION', 'JOB TYPE' ('Run'), 'INVENTORY' ('Workshop Inventory'), 'PROJECT' ('Workshop Project'), 'PLAYBOOK' ('network\_banner.yml'), 'CREDENTIAL' ('Workshop Credential'), 'FORKS' ('0'), 'LIMIT' (empty), 'VERBOSITY' ('0 (Normal)'), 'JOB TAGS' (empty), 'SKIP TAGS' (empty), and 'LABELS' (empty). There are also 'PROMPT ON LAUNCH' checkboxes for various fields.

# Creating a Survey (2/2)

The Add Survey window allows the Job Template to prompt users for one or more questions. The answers provided become variables for use in the Ansible Playbook.

The screenshot shows the 'Edit Survey Prompt' configuration window. It includes fields for 'PROMPT' (containing 'Please enter the banner text'), 'DESCRIPTION' (containing 'Please type into the text field the desired banner'), 'ANSWER VARIABLE NAME' (set to 'net\_banner'), 'ANSWER TYPE' (set to 'Textarea'), 'MINIMUM LENGTH' (set to 0), 'MAXIMUM LENGTH' (set to 4096), and a 'DEFAULT ANSWER' field. A 'REQUIRED' checkbox is checked. On the right, a 'PREVIEW' panel shows the survey prompt with the same text and instructions, along with edit and delete icons.

CONFIGURE BANNER | SURVEY ON

EDIT SURVEY PROMPT

\* PROMPT  
Please enter the banner text

DESCRIPTION  
Please type into the text field the desired banner

\* ANSWER VARIABLE NAME ⓘ  
net\_banner

\* ANSWER TYPE ⓘ  
Textarea

MINIMUM LENGTH  
0

MAXIMUM LENGTH  
4096

DEFAULT ANSWER

REQUIRED

PREVIEW

\* PLEASE ENTER THE BANNER TEXT  
Please type into the text field the desired banner

edit delete

CLEAR UPDATE CANCEL SAVE

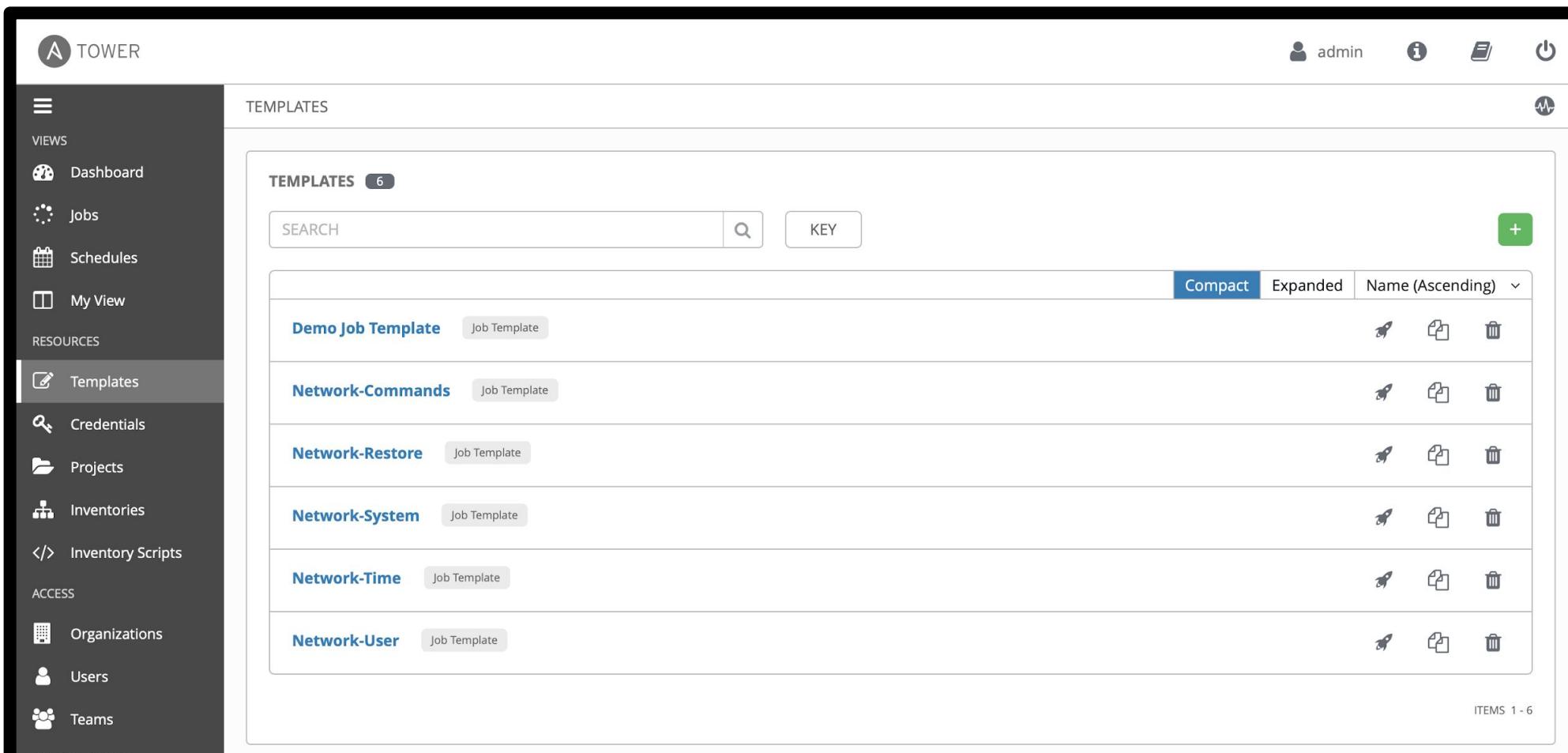
# Using a Survey

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'.

# 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

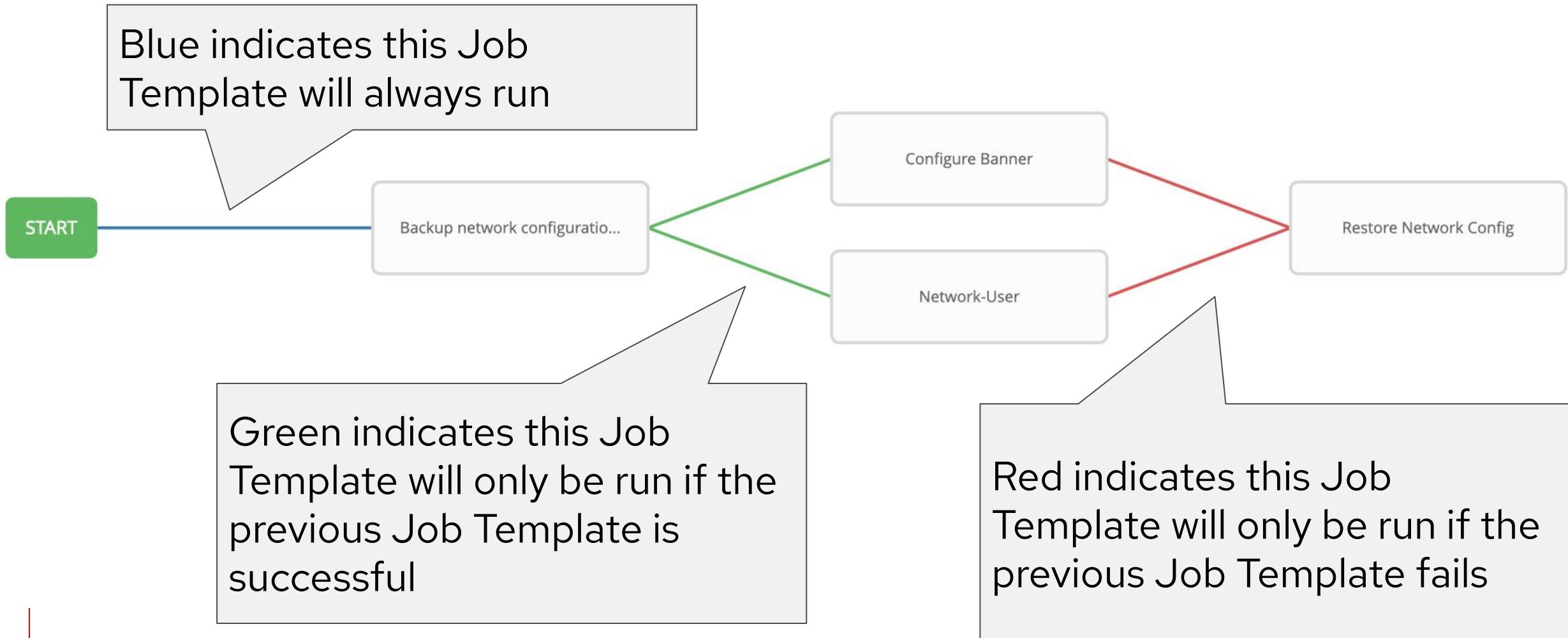
The main content area is titled "TEMPLATES" and shows a list of six items:

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".

# Visualizing a Workflow

Workflows can branch out, or converge in.



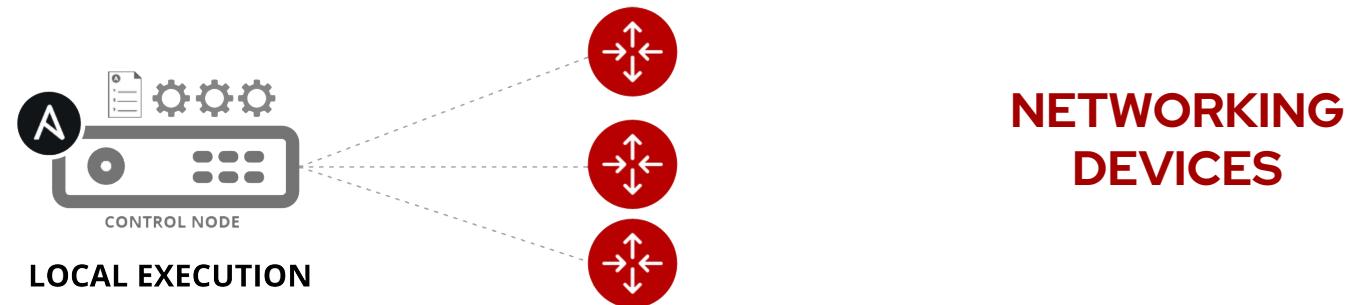
# Plays and Playbooks



**Red Hat**  
Ansible  
Automation

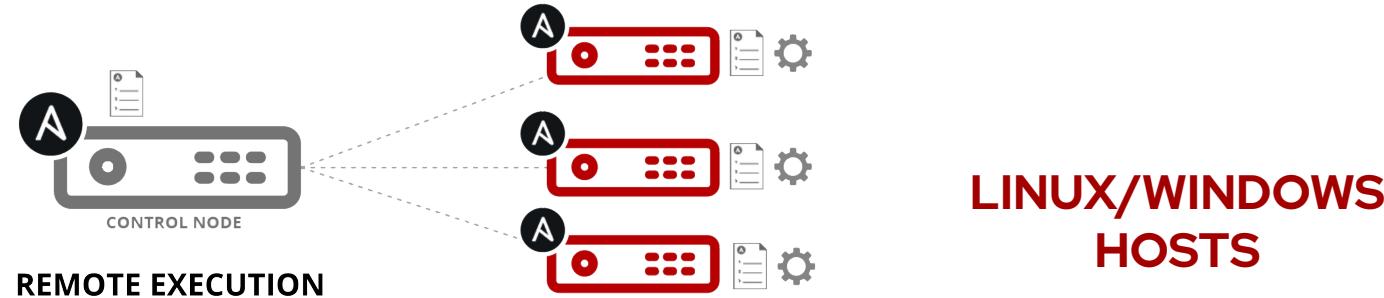
# How Ansible Network Automation works

*Module code is executed locally on the control node*



**NETWORKING  
DEVICES**

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



**LINUX/WINDOWS  
HOSTS**

# A Sample Ansible Networking Playbook

```
---
```

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

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

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

# An Ansible Playbook Example

```
---
- name: snmp ro/rw string configuration
  hosts: cisco
  gather_facts: no

  tasks:
    - name: ensure snmp strings are present
      ios_config:
        lines:
          - snmp-server community ansible-public RO
          - snmp-server community ansible-private RW
```

# Ansible Playbook - Play definition

- The **name** parameter describes the Ansible Play
- Target devices using the **hosts** parameter
- Optionally disable **gather\_facts**

```
---
- name: snmp ro/rw string configuration
  hosts: cisco
  gather_facts: no
```

# Modules

Modules do the actual work in Ansible, they are what gets executed in each playbook task.

- Typically written in Python (but not limited to it)
- Modules can be idempotent
- Modules take user input in the form of parameters

```
tasks:  
  - name: ensure snmp strings are present  
    ios_config:  
      commands:  
        - snmp-server community ansible-public RO  
        - snmp-server community ansible-private RW
```

# Network modules

Ansible modules for network automation typically references the vendor OS followed by the module name.

- \*\_facts
- \*\_command
- \*\_config

More modules depending on platform

Arista EOS = eos\_\*

Cisco IOS/IOS-XE = ios\_\*

Cisco NX-OS = nxos\_\*

Cisco IOS-XR = iosxr\_\*

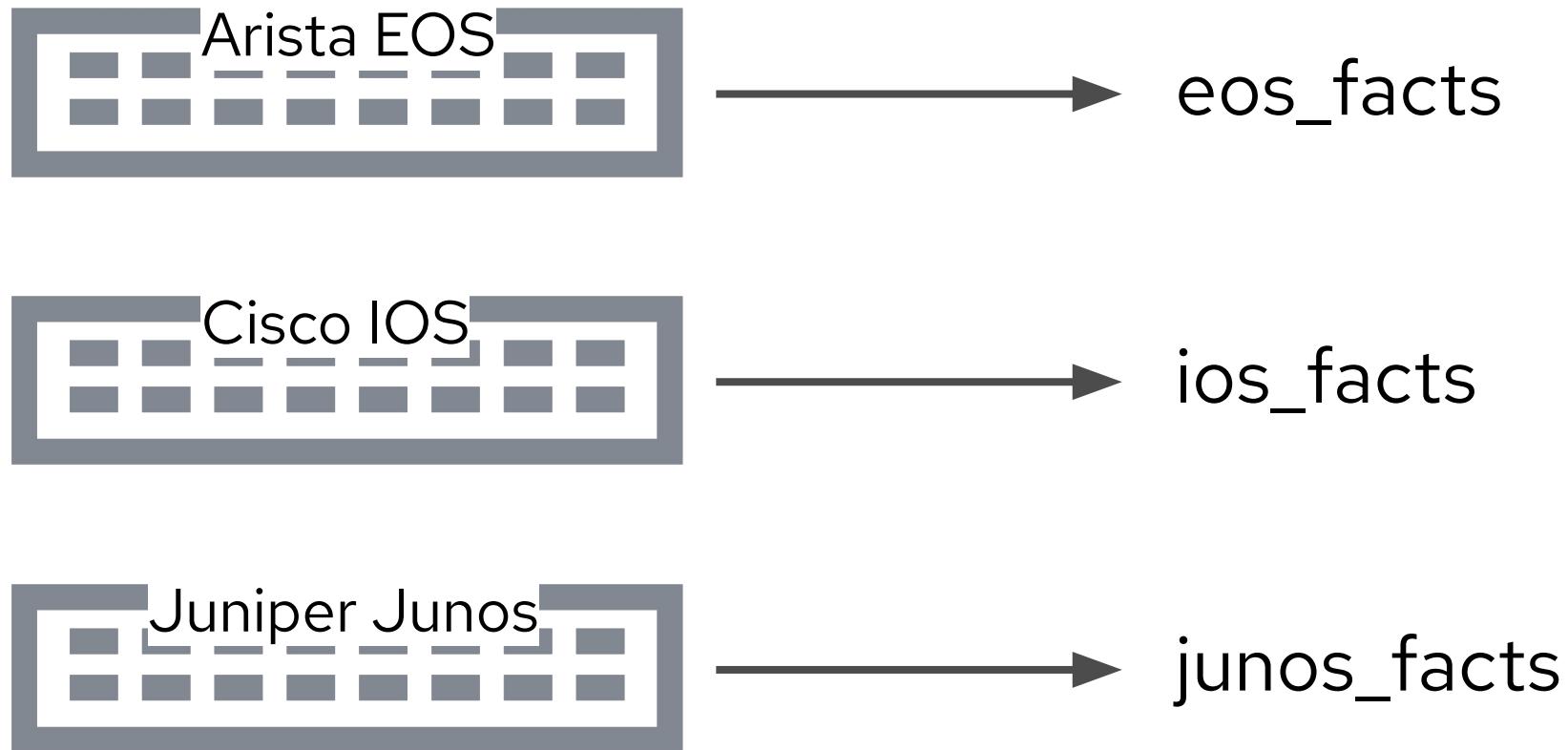
F5 BIG-IP = bigip\_\*

F5 BIG-IQ = bigiq\_\*

Juniper Junos = junos\_\*

VyOS = vyos\_\*

# Fact modules



# Fact modules return structured data

```
rtr1#show version
Cisco IOS XE Software, Version 16.09.02
Cisco IOS Software [Fuji], Virtual XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.9.2, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 05-Nov-18 19:26 by mcpred
..
.
<rest of output removed for brevity>
```

```
[student1@ansible ~]$ ansible -m ios_facts rtr1
.<<abbreviated output>>
.
{
    "ansible_net_iostype": "IOS-XE",
    "ansible_net_memfree_mb": 1853921,
    "ansible_net_memtotal_mb": 2180495,
    "ansible_net_model": "CSR1000V",
    "ansible_net_neighbors": {},
    "ansible_net_python_version": "2.7.5",
    "ansible_net_serialnum": "964A1H0D1RM",
    "ansible_net_system": "ios",
    "ansible_net_version": "16.09.02",
}
.
```

# Ansible Fact Playbook Example

```
---
- name: gather information from routers
  hosts: cisco
  gather_facts: no

  tasks:
    - name: gather router facts
      ios_facts:
```

# Build reports with Ansible Facts

Hostname	Model Type	Mgmt0 IP Address	Code Version
n9k	Nexus9000 9000v Chassis	192.168.2.3	7.0(3)I7(1)
n9k2	Nexus9000 9000v Chassis	192.168.2.4	7.0(3)I7(1)
n9k3	Nexus9000 9000v Chassis	192.168.2.5	7.0(3)I7(1)
n9k4	Nexus9000 9000v Chassis	192.168.2.6	7.0(2)I7(1)
n9k5	Nexus9000 9000v Chassis	192.168.2.7	7.0(3)I7(1)
n9k6	Nexus9000 9000v Chassis	192.168.2.8	7.0(3)I7(1)

# Running a playbook

```
---
- name: snmp ro/rw string configuration
  hosts: cisco
  gather_facts: no

  tasks:
    - name: ensure snmp strings are present
      ios_config:
        commands:
          - snmp-server community ansible-public RO
          - snmp-server community ansible-private RW
```

```
[student1@ansible networking-workshop]$ ansible-playbook playbook.yml

PLAY [snmp ro/rw string configuration] ****
TASK [ensure that the desired snmp strings are present] ****
changed: [rtr1]

PLAY RECAP ****
rtr1 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```



# Displaying output

```
[student1@ansible networking-workshop]$ ansible-playbook playbook.yml -v
Using /home/student1/.ansible.cfg as config file

PLAY [snmp ro/rw string configuration] ****
TASK [ensure that the desired snmp strings are present] ****
changed: [rtr1] => changed=true
  ansible_facts:
    discovered_interpreter_python: /usr/bin/python
  banners: {}
  commands:
    - snmp-server community ansible-public RO
    - snmp-server community ansible-private RW
  updates:
    - snmp-server community ansible-public RO
    - snmp-server community ansible-private RW

PLAY RECAP ****
rtr1      : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

**Increase the level of verbosity by adding more "v's" -vvvv**



# "Ansible for Network Automation" Documentation

The screenshot shows the Ansible documentation website for version 2.8. The top navigation bar includes links for ANSIBLEFEST, PRODUCTS, COMMUNITY, WEBINARS & TRAINING, and BLOG. The main content area is titled "Ansible for Network Automation". It explains that Ansible Network modules extend automation to network administrators and teams, mentioning configuration, testing, validation, discovery, and drift correction. It provides links to "Getting Started with Ansible for Network Automation" and "Advanced Topics with Ansible for Network Automation". Below this, it lists network modules maintained by the Ansible community. A sidebar on the left contains links for Installation, Upgrade & Configuration, Using Ansible, Contributing to Ansible, Extending Ansible, Common Ansible Scenarios, and Ansible for Network Automation. The "Ansible for Network Automation" section in the sidebar is expanded, showing sub-links for Getting Started, Advanced Topics, Developer Guide, and Reference & Appendices. A search bar at the bottom right allows users to search the site.

<http://bit.ly/AnsibleNetwork>



# 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](#)

# Chat with us

- **Slack**

<https://ansiblenetwork.slack.com>

Join by clicking here <http://bit.ly/ansibleslack>

- **IRC**

#ansible-network on freenode

<http://webchat.freenode.net/?channels=ansible-network>



# #ANSIBLEFEST

Join us October 13-15 in San Diego

Stay up to date at [AnsibleFest.com](http://AnsibleFest.com)

# Demo time!

All of the playbooks we'll run in the demo today are on Github at:

[https://github.com/network-automation/tower\\_workshop](https://github.com/network-automation/tower_workshop)

# 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)