

# Mainline Corporate Overview

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



**Red Hat**

Ansible  
Automation

# Automating Windows with Ansible



**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**

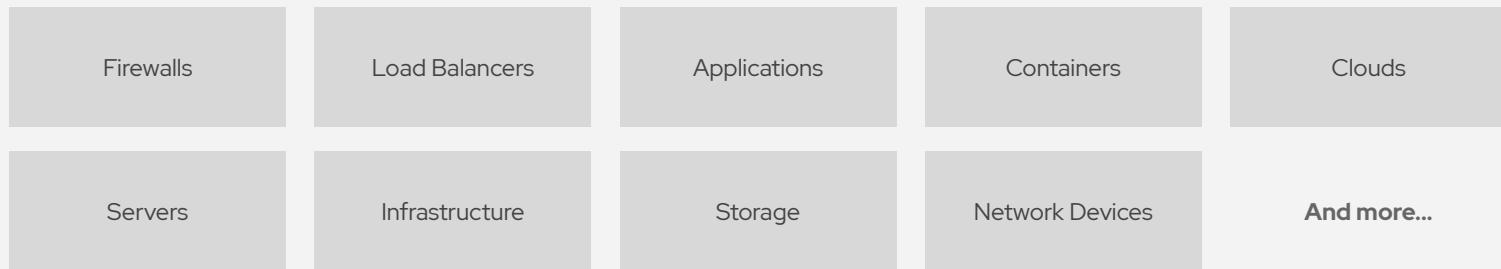
# What can I do using Ansible?

Automate the deployment and management of your *entire* IT footprint.

**Do this...**



**On these...**



# Ad-hoc Automation is happening in silos



Developers

→ Ansible used in silo



Security

→ DIY scripting automation



Infrastructure

→ Open source config management tool

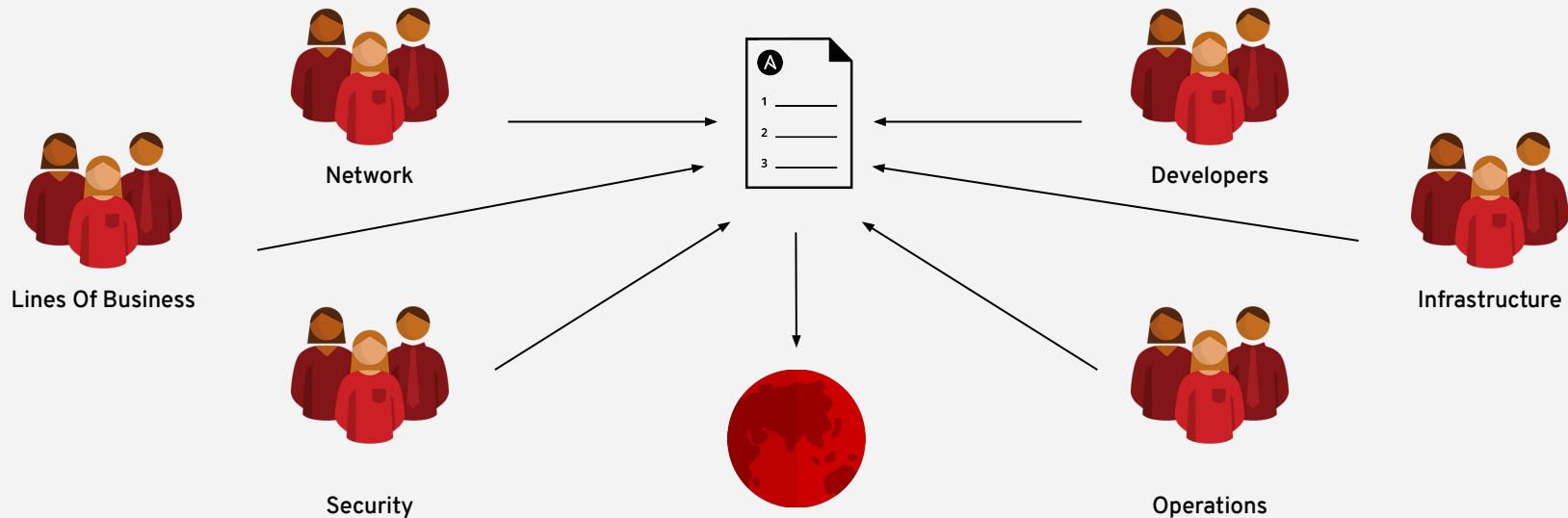


Network

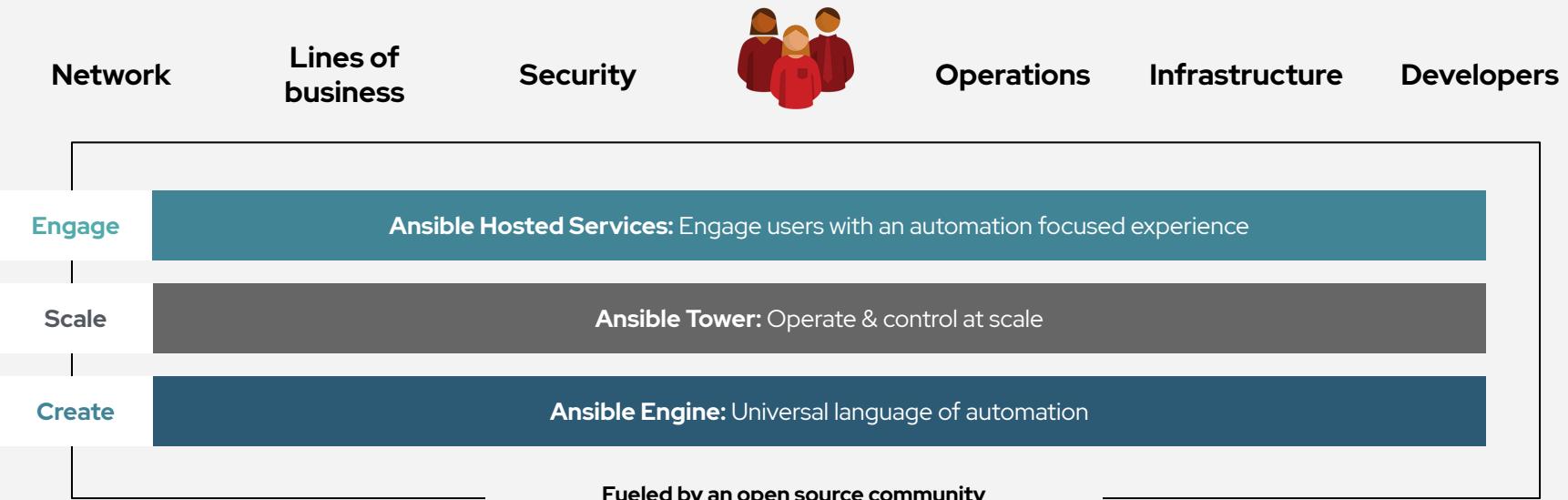
→ Proprietary vendor supplied automation

## 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

AWS  
Azure  
Digital Ocean  
Google  
OpenStack  
Rackspace  
**+more**

## Operating Systems

RHEL  
Linux  
Windows  
**+more**

## Virt & Container

Docker  
VMware  
RHV  
OpenStack  
OpenShift  
**+more**

## Storage

Netapp  
Red Hat Storage  
Infinidat  
**+more**

## Windows

ACLs  
Files  
Packages  
IIS  
Regedit  
Shares  
Services  
Configs  
Users  
Domains  
**+more**

## Network

A10  
Arista  
Aruba  
Cumulus  
Bigswitch  
Cisco  
Dell  
Extreme  
F5  
Lenovo  
MikroTik  
Juniper  
OpenSwitch  
**+more**

## Security

Checkpoint  
Cisco  
CyberArk  
F5  
Fortinet  
Juniper  
IBM  
Palo Alto  
Snort  
**+more**

## Monitoring

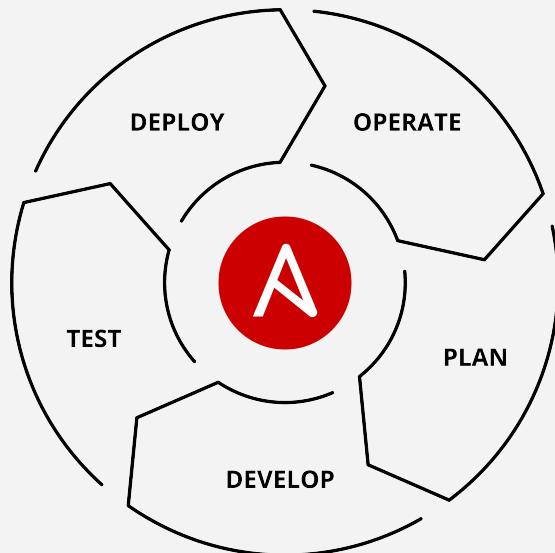
Dynatrace  
Datadog  
LogicMonitor  
New Relic  
Sensu  
**+more**

## Devops

Jira  
GitHub  
Vagrant  
Jenkins  
Slack  
**+more**

# WHAT CAN I DO USING ANSIBLE FOR WINDOWS?

Native Windows support uses PowerShell remoting to manage Windows in the same Ansible agentless way



- Install and uninstall MSIs
- Gather facts on Windows hosts
- Enable and disable Windows features
- Start, stop, and manage Windows Services
- Create and Manage local users and groups
- Manage Windows packages via [Chocolatey package manager](#)
- Manage and install Windows updates
- Fetch files from remote sites
- Push and execute any Powershell scripts

# 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**

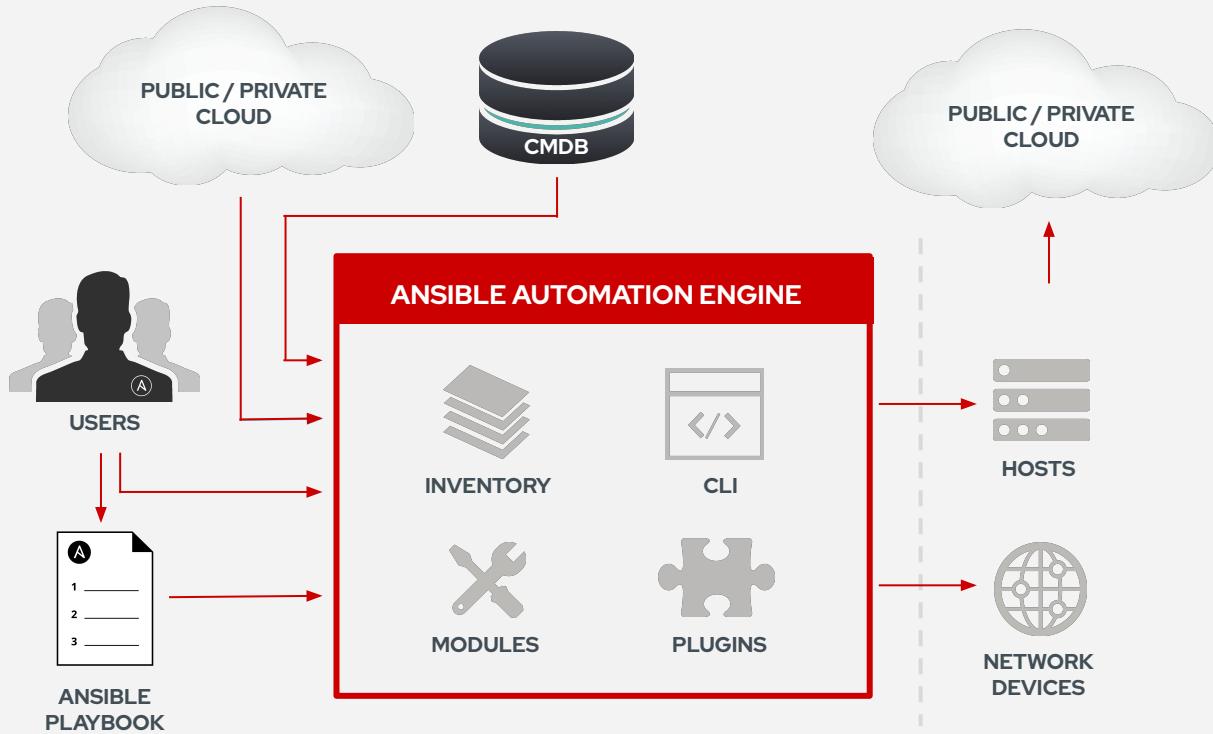
<3 MONTHS

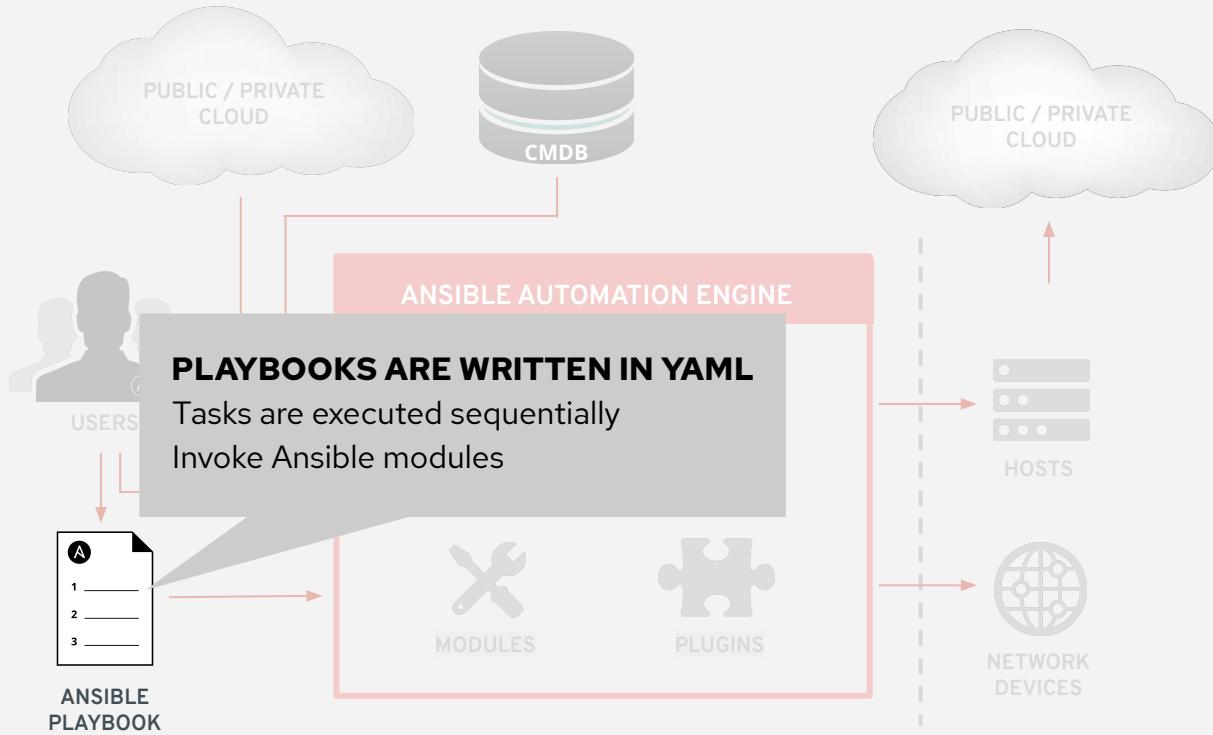
**Payback on Ansible Tower**

# How Ansible Works



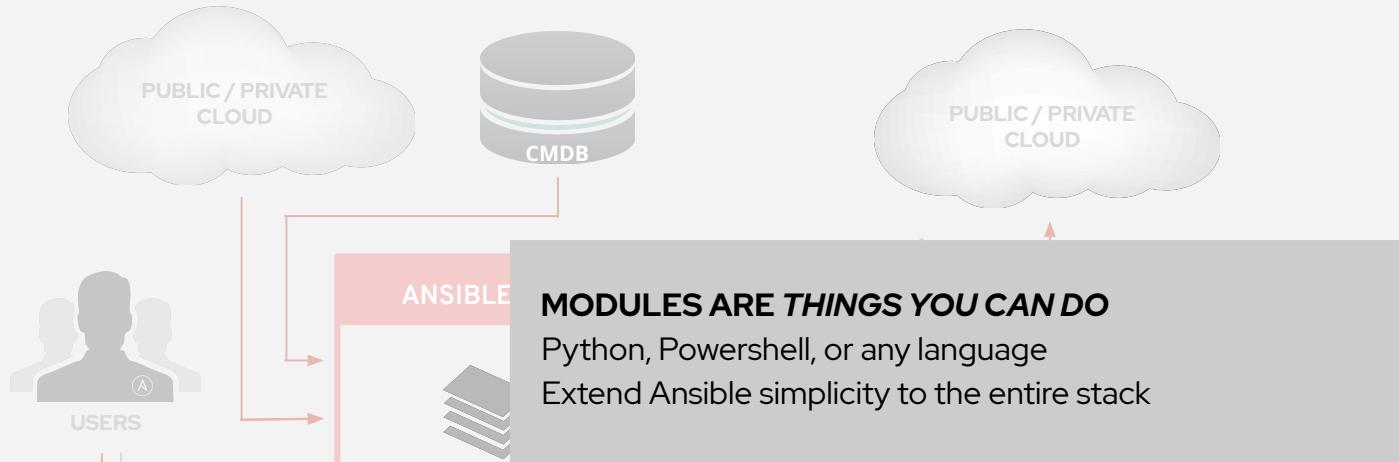
**Red Hat**  
Ansible  
Automation





```
---
```

- **name: start IIS/stop firewall**  
  **hosts:** windows-web  
  **become:** yes  
  **tasks:**
  - **name: IIS is running**  
    **win\_service:**
    - name:** W3Svc
    - state:** running
  - **name: firewall service is stopped/disabled**  
    **win\_service:**
    - name:** MpsSvc
    - state:** stopped
    - start\_mode:** disabled



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

# Modules

Modules do the work in Ansible, they are what each Playbook task executes:

- Written in Powershell
- Modules are idempotent
- Modules take user input in the form of parameters

```
tasks:  
  - name: start IIS  
    win_service:  
      name: W3Svc  
      state: running
```

# Windows Modules

Ansible modules for Windows automation typically begin with `win_*`

**win\_copy** - Copies files to remote locations on windows hosts

**win\_service** - Manage and query Windows services

**win\_domain** - Ensures the existence of a Windows domain

**win\_reboot** - Reboot a windows machine

**win\_regedit** - `win_regedit` – Add, change, or remove registry keys and values

**win\_ping** - A windows version of the classic ping module

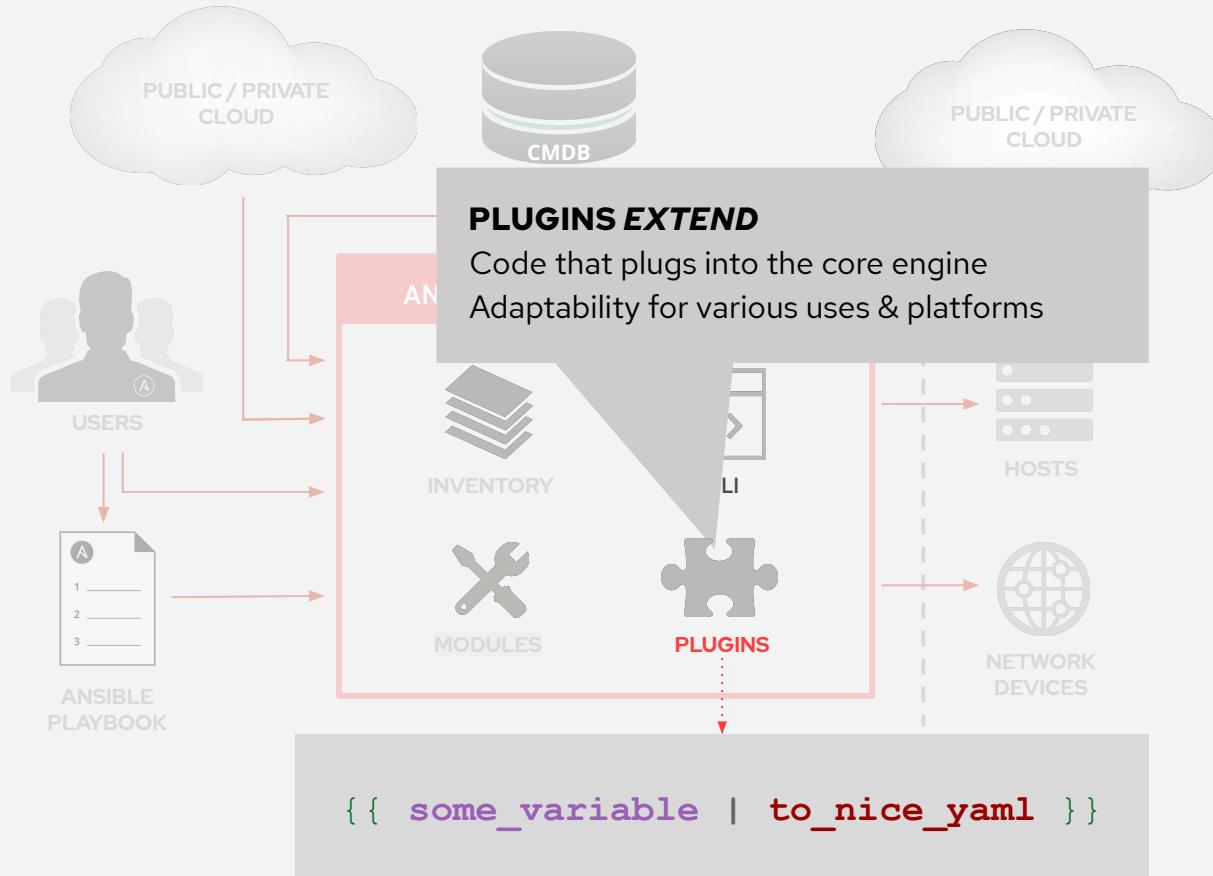
**win\_dsc** - Invokes a PowerShell DSC configuration

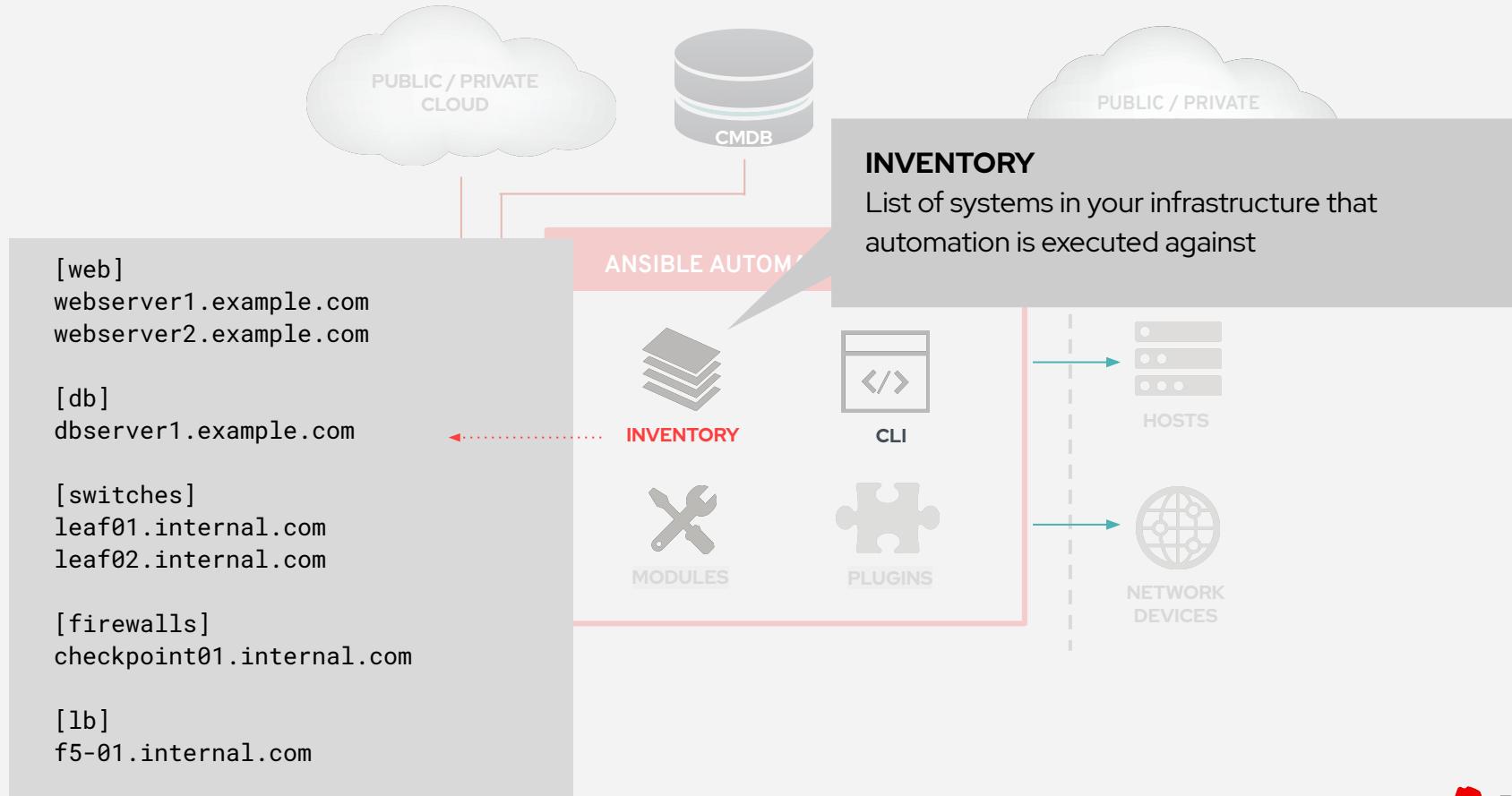
**win\_acl** - Set file/directory/registry permissions for a system user or group

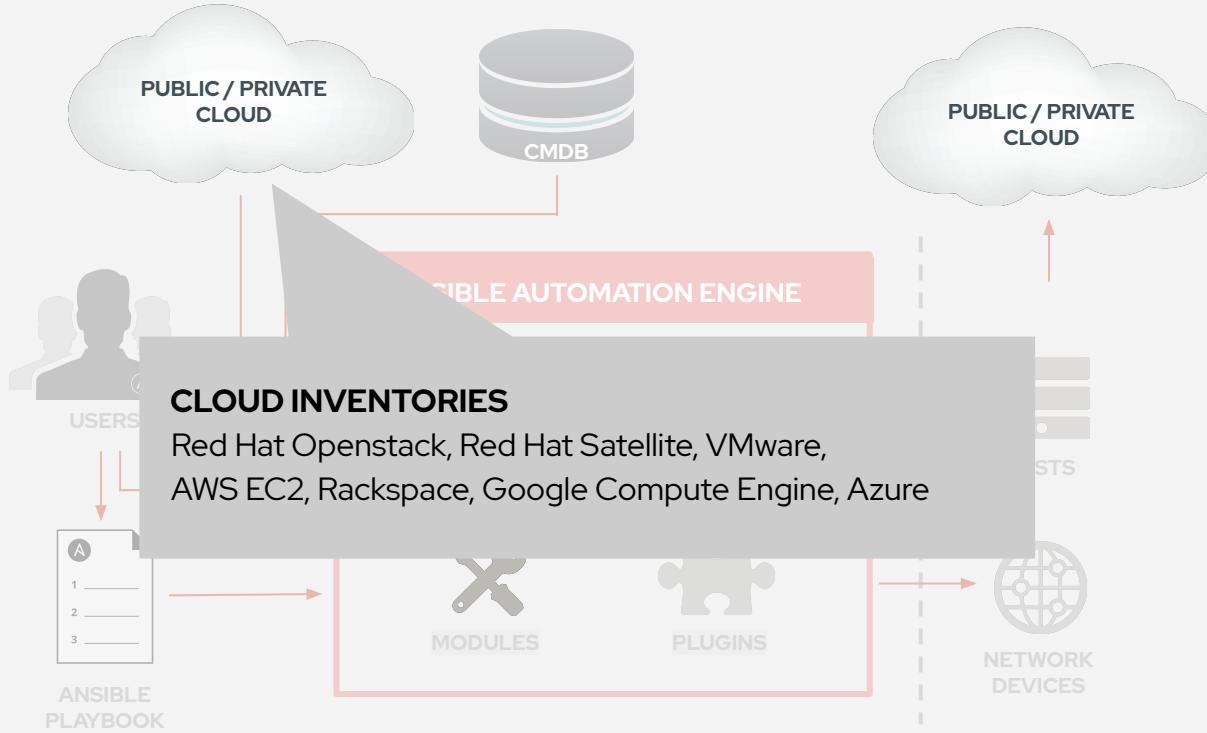
The full list is here:

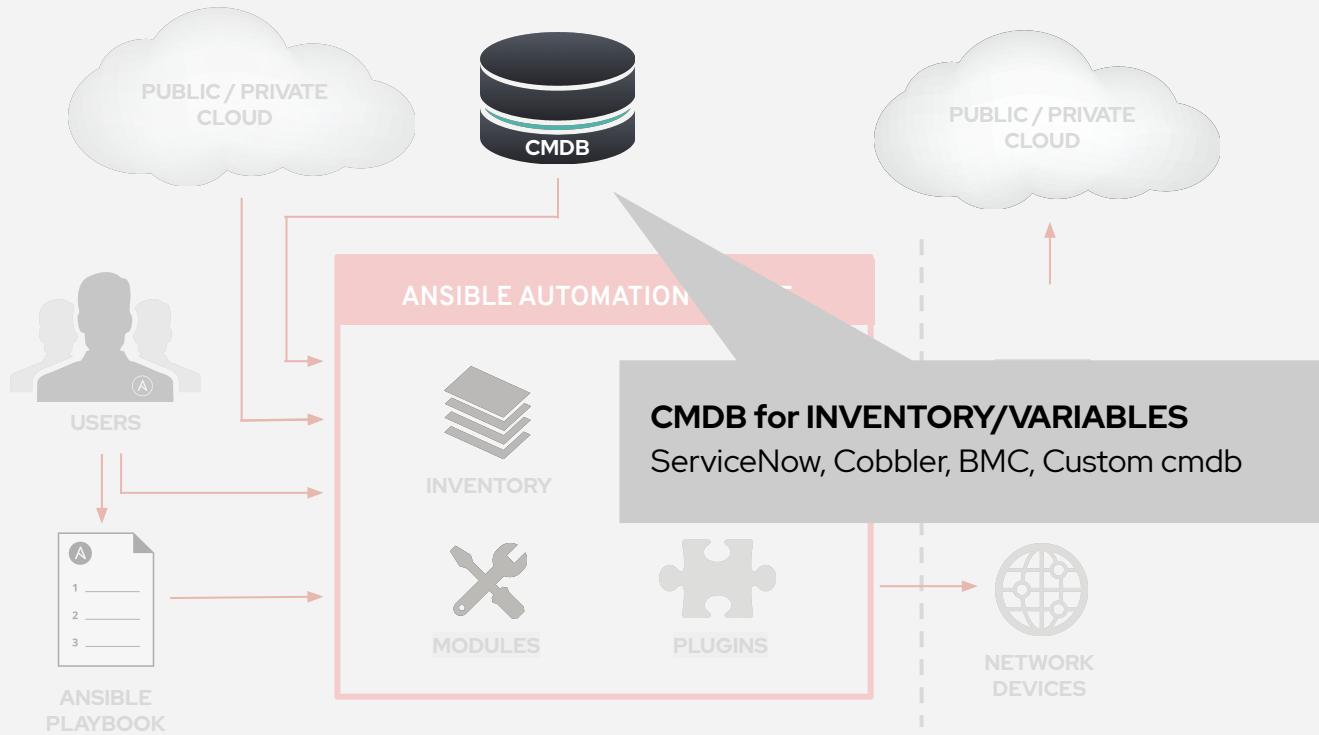
[https://docs.ansible.com/ansible/latest/modules/list\\_of\\_windows\\_modules.html#windows-modules](https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html#windows-modules)







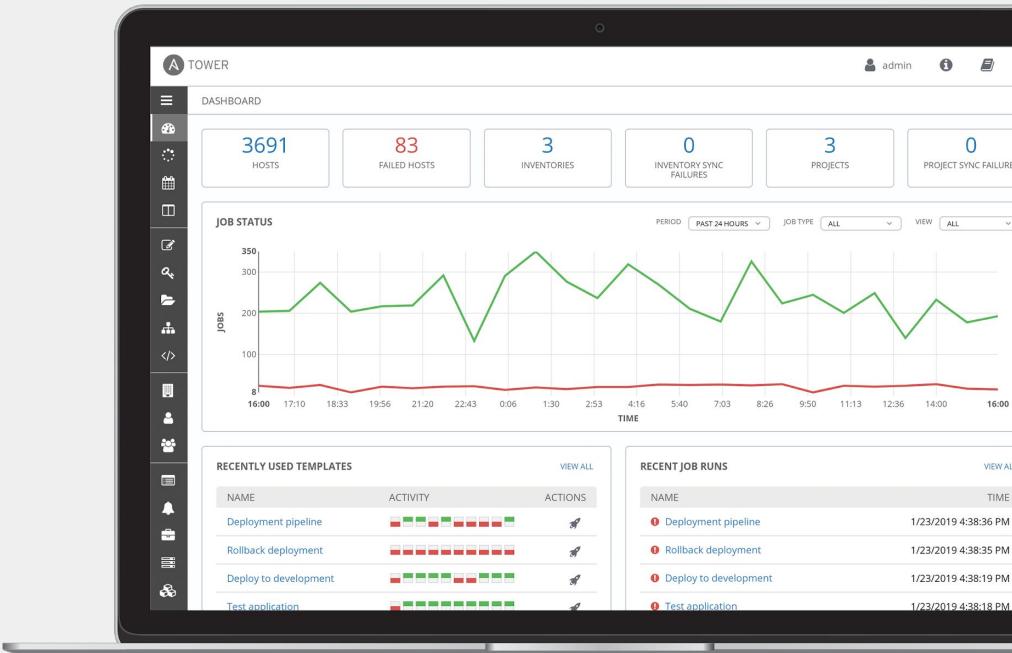




# What is Ansible Tower?

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.

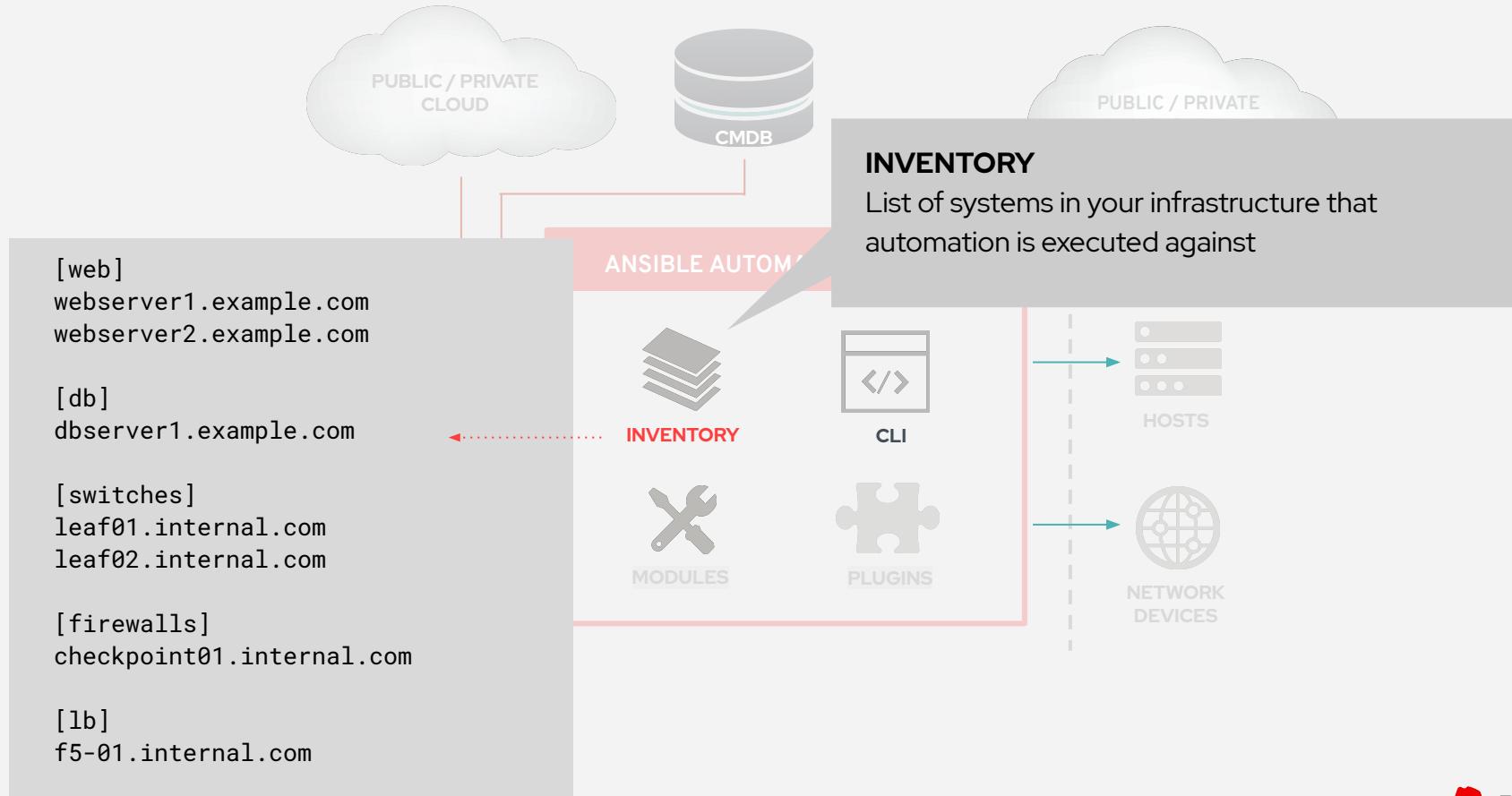
Job templates also encourage the reuse of Ansible Playbook content and collaboration between teams.

A Job Template requires:

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

The screenshot shows the Ansible Tower web interface with the 'TEMPLATES / Azure Linux VM Spinup' page open. The page has several tabs at the top: DETAILS, PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and EDIT SURVEY. The DETAILS tab is selected. The configuration fields include:

- NAME:** Azure Linux VM Spinup
- DESCRIPTION:** (empty)
- JOB TYPE:** Run
- INVENTORY:** Prod
- CREDENTIAL:** Azure-Service-Principal
- PROJECT:** fest19-demo
- PLAYBOOK:** azure\_spinup.yml
- LIMIT:** (empty)
- Skip Tags:** (empty)
- Verbosity:** 0 (Normal)
- Job Tags:** (empty)
- Labels:** (empty)
- Instance Groups:** (empty)
- Timeout:** 0
- Show Changes:** OFF
- Prompt on Launch:** (checkbox)
- OPTIONS:**
  - ENABLE PRIVILEGE ESCALATION (checkbox)
  - ALLOW PROVISIONING CALLBACKS (checkbox)
  - ENABLE CONCURRENT JOBS (checkbox)
  - USE FACT CACHE (checkbox)
- EXTRA VARIABLES:** (YAML tab is selected)



# 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

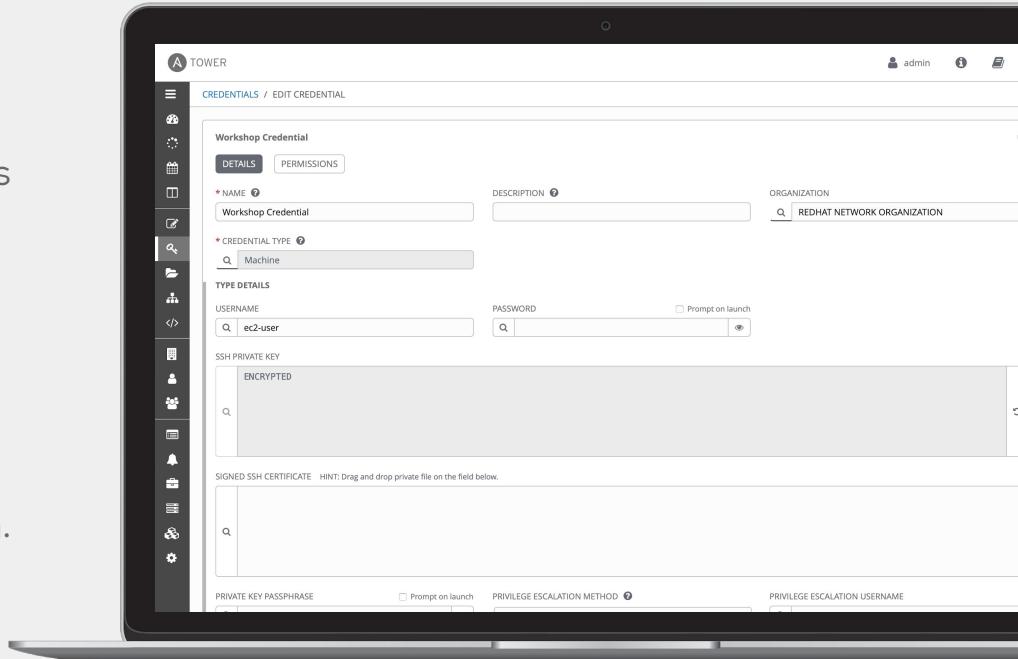
The screenshot shows the Ansible Tower web interface on a laptop screen. The main window displays the 'Workshop Inventory' hosts list. The left sidebar contains navigation icons for inventories, hosts, groups, and other management functions. The top navigation bar includes tabs for 'INVENTORIES / Workshop Inventory / HOSTS', and buttons 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 shows five entries: 'ansible', 'rtr1', 'rtr2', 'rtr3', and 'rtr4', each with an 'ON' status indicator and a radio button. To the right of the host list, a 'RELATED GROUPS' section lists several groups with their respective hosts: 'control' (ansible), 'cisco' (rtr1, rtr2), 'arista' (rtr3, rtr4), 'dc1' (rtr1, rtr2), 'dc2' (rtr3, rtr4), 'dc1' (rtr3), 'juniper' (rtr4), and 'arista' (rtr4). Below the host list, there are additional tabs for 'INVENTORIES' and 'HOSTS', along with a search bar and filter options 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

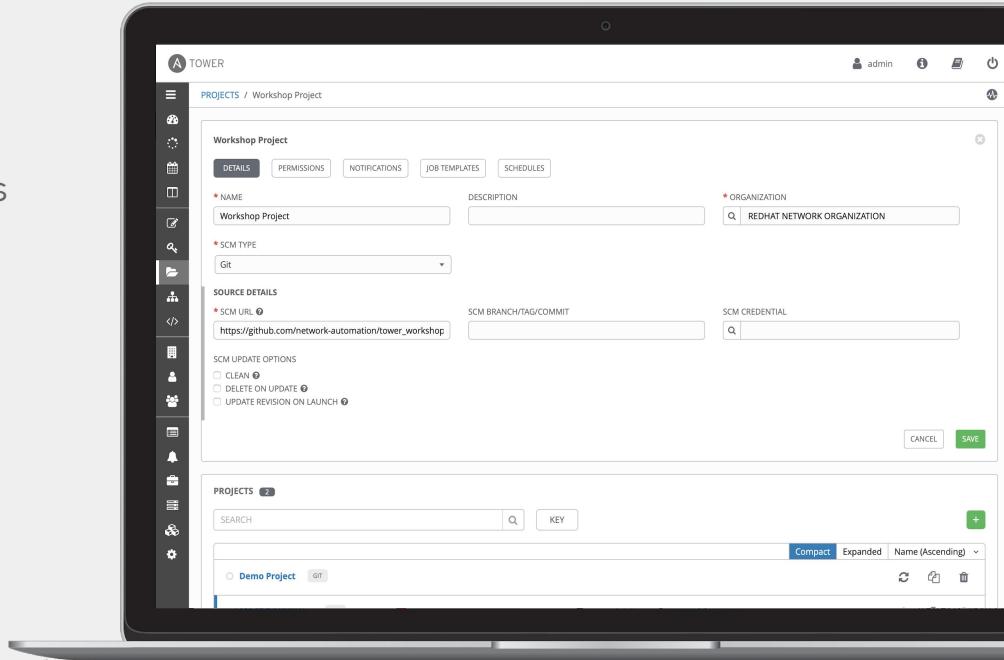
Centralized management of credentials allows users to leverage them without ever seeing them.



# 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.



# Commands, Plays, and Playbooks



# Ad-hoc Commands

An ad-hoc command is a single Ansible task to perform quickly, but don't want to save for later.

# Ad-hoc Commands: Common Arguments In

- **-m MODULE\_NAME, --module-name=MODULE\_NAME**  
Module name to execute the ad-hoc command
- **-a MODULE\_ARGS, --args=MODULE\_ARGS**  
Module arguments for the ad-hoc command
- **-b, --become**  
Run ad-hoc command with elevated rights such as sudo, the default method
- **-e EXTRA\_VARS, --extra-vars=EXTRA\_VARS**  
Set additional variables as key=value or YAML/JSON
- **--version**  
Display the version of Ansible
- **--help**  
Display the MAN page for the Ansible tool

# Ad-hoc Commands

```
# check all my inventory hosts are ready to be
# managed by Ansible
$ ansible all -m ping

# collect and display the discovered facts
# for the localhost
$ ansible localhost -m setup

# run the uptime command on all hosts in the
# web group
$ ansible web -m command -a "uptime"
```

# Variables

Ansible can work with metadata from various sources and manage their context in the form of variables.

- Command line parameters
- Plays and tasks
- Files
- Inventory
- Discovered facts
- Roles

```
---
```

- **name: start IIS/stop firewall**  
  **hosts:** windows-web  
  **become:** yes  
  **tasks:**
  - **name: IIS is running**  
    **win\_service:**
    - name:** W3Svc
    - state:** running
  - **name: firewall service is stopped/disabled**  
    **win\_service:**
    - name:** MpsSvc
    - state:** stopped
    - start\_mode:** disabled

# Plays and playbooks

Plays are **ordered sets of tasks** to execute against host selections from your inventory.

A playbook is a file containing one or more plays.

# Let's walk through a playbook!

```
---
```

- name: Ensure IIS is installed and started
  - hosts: web
  - become: yes
  - vars:
    - service\_name: IIS Admin Service
- tasks:
  - name: Ensure IIS Server is present
    - win\_feature:
      - name: Web-Server
      - state: present
  - name: Ensure latest index.html file is present
    - win\_copy:
      - src: files/index.html
      - dest: c:\www\
  - name: Ensure IIS is started
    - win\_service:
      - name: "{{ server\_name }}"
      - state: started

# Meaningful names

```
---
```

- `name: Ensure IIS is installed and started`
  - `hosts: web`
  - `become: yes`
  - `vars:`
    - `service_name: IIS Admin Service`
- `tasks:`
  - `name: Ensure IIS Server is present`
    - `win_feature:`
      - `name: Web-Server`
      - `state: present`
  - `name: Ensure latest index.html file is present`
    - `win_copy:`
      - `src: files/index.html`
      - `dest: c:\www\`
  - `name: Ensure IIS is started`
    - `win_service:`
      - `name: "{{ service_name }}"`
      - `state: started`

# Host selector

```
---
```

- name: Ensure IIS is installed and started
  - hosts: web
  - become: yes
  - vars:
    - service\_name: IIS Admin Service
- tasks:
  - name: Ensure IIS Server is present
    - win\_feature:
      - name: Web-Server
      - state: present
  - name: Ensure latest index.html file is present
    - win\_copy:
      - src: files/index.html
      - dest: c:\www\
  - name: Ensure IIS is started
    - win\_service:
      - name: "{{ service\_name }}"
      - state: started

# Privilege escalation

```
---
```

```
- name: Ensure IIS is installed and started
  hosts: web
  become: yes
  vars:
    service_name: IIS Admin Service

  tasks:
    - name: Ensure IIS Server is present
      win_feature:
        name: Web-Server
        state: present

    - name: Ensure latest index.html file is present
      win_copy:
        src: files/index.html
        dest: c:\www\

    - name: Ensure IIS is started
      win_service:
        name: "{{ service_name }}"
        state: started
```

# Plays variables

```
---
```

- name: Ensure IIS is installed and started
  - hosts: web
  - become: yes

```
vars:
```

- service\_name: IIS Admin Service

```
tasks:
```

- name: Ensure IIS Server is present
  - win\_feature:
    - name: Web-Server
    - state: present
- name: Ensure latest index.html file is present
  - win\_copy:
    - src: files/index.html
    - dest: c:\www\
- name: Ensure IIS is started
  - win\_service:
    - name: "{{ service\_name }}"
    - state: started

# Tasks

```
---
```

- name: Ensure IIS is installed and started
  - hosts: web
  - become: yes
  - vars:
    - service\_name: IIS Admin Service
- tasks:
  - name: Ensure IIS Server is present
    - win\_feature:
      - name: Web-Server
      - state: present
  - name: Ensure latest index.html file is present
    - win\_copy:
      - src: files/index.html
      - dest: c:\www\
  - name: Ensure IIS is started
    - win\_service:
      - name: "{{ service\_name }}"
      - state: started

# Roles



**Red Hat**  
Ansible  
Automation

# Roles

Roles are packages of closely related Ansible content that can be shared more easily than plays/playbooks alone.

- Improves readability and maintainability of complex plays
- Eases sharing, reuse and standardization of automation processes
- Enables Ansible content to exist independently of playbooks, projects -- even organizations
- Provides functional conveniences such as file path resolution and default values

# Roles

## Project with Embedded Roles Example

```
# site.yml
---
- name: Execute common and iis role
  hosts: web
  roles:
    - common
    - iis
```

# Roles

**<http://galaxy.ansible.com>**

Ansible Galaxy is a hub for finding, reusing and sharing Ansible content.

Jump-start your automation project with content contributed and reviewed by the Ansible community.

# 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](#)

## **April 15 - What's New in RHEL 8**

Want to learn the latest and greatest from the newest major release of Red Hat® Enterprise Linux®? Join Red Hat Solutions Architect, Brad Krumme, as he walks through the new, cool features of RHEL 8, including an overview of infrastructure predictive analytics tool Red Hat Insights.

[REGISTER NOW >](#)

## **April 22 - Ansible is for Humans**

As IT departments continue to see increasing responsibility within their organization, the inclusion of more automation is crucial to make the commitments required of them. Distilling the lessons of automation from other industries like manufacturing, we can find out what an ideal method of applying universal automation principles within IT organizations. These lessons are that an automation tool needs to be simple, powerful, and universal. We will show how Ansible's agentless architecture is a perfect fit with an IT framework. This allows for IT departments to have quality, agility, and speed in delivering their workloads and services to the rest of the organization.

[REGISTER NOW >](#)

## **April 29 - Automating Networking with Ansible**

As of 2018, approximately 71% of networks are still driven manually - that's a huge amount of time and effort that could instead be done by robots. Red Hat Solutions Architect Laine Vyvyan will give an overview and demo of the Red Hat Ansible Automation Platform and explain how it can help recover the time, resources, and focus spent manually automating networks.

[REGISTER NOW >](#)



# #ANSIBLEFEST

Join us October 13-15 in San Diego

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

# Demo!



# 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)