



**Red Hat**

Ansible Automation  
Platform

# Red Hat Ansible Platform technical deck

Creating, operating, and consuming automation

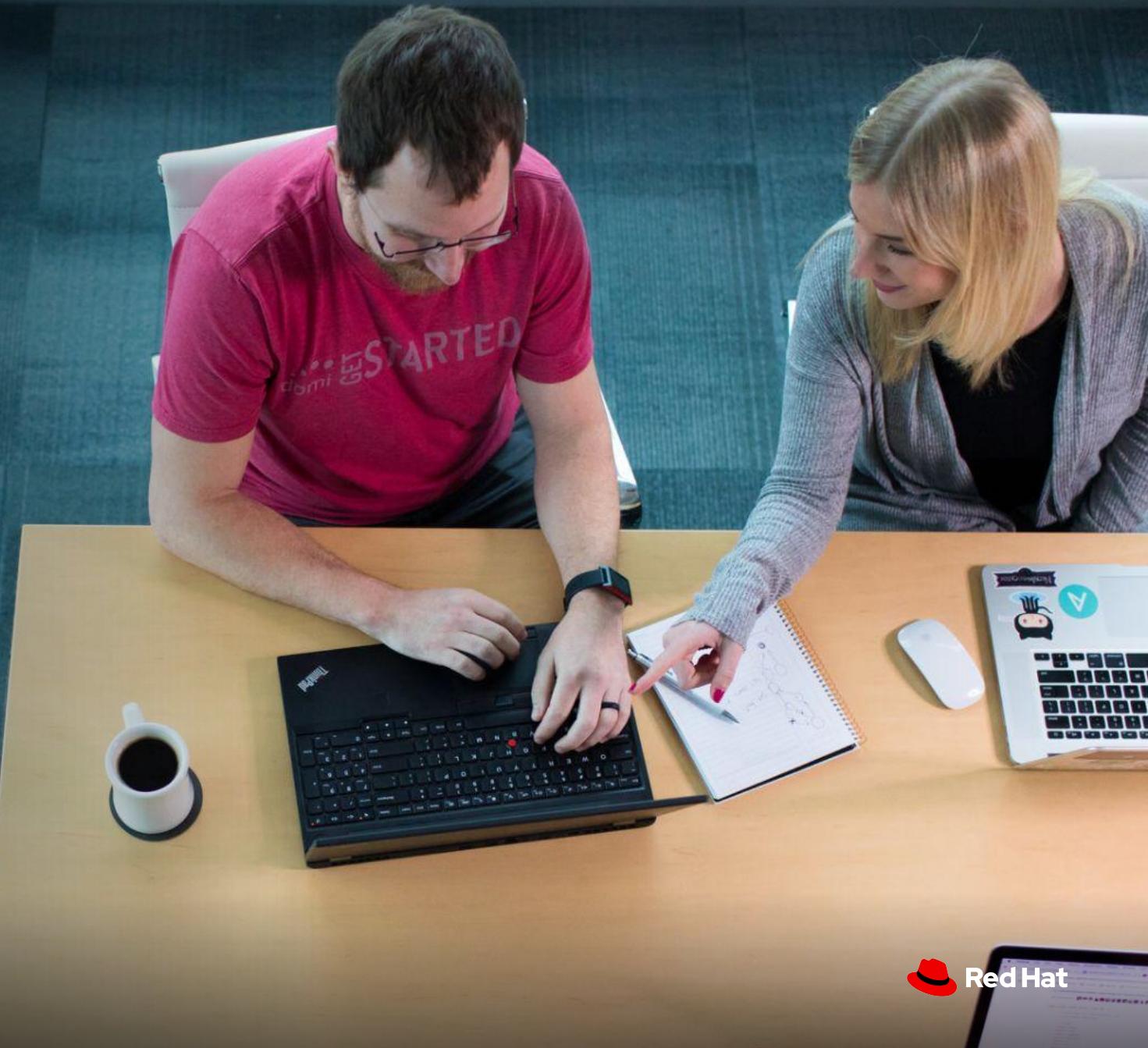
Andreas Stolzenberger

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

Principal Solution Architect

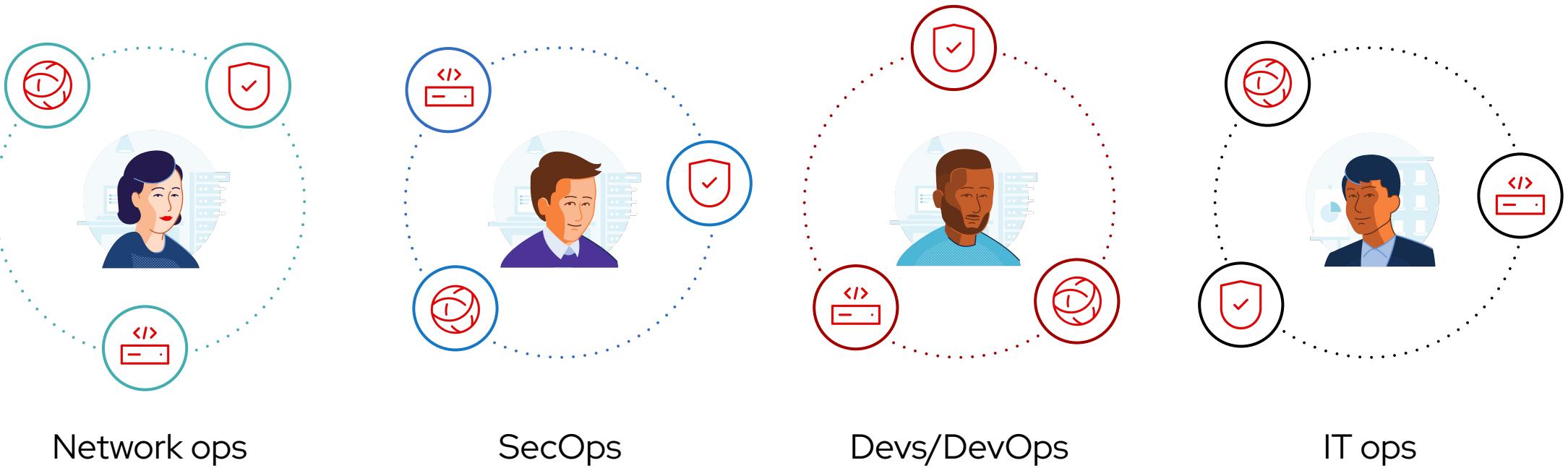


Anyone can automate...  
but an enterprise needs  
to coordinate and scale



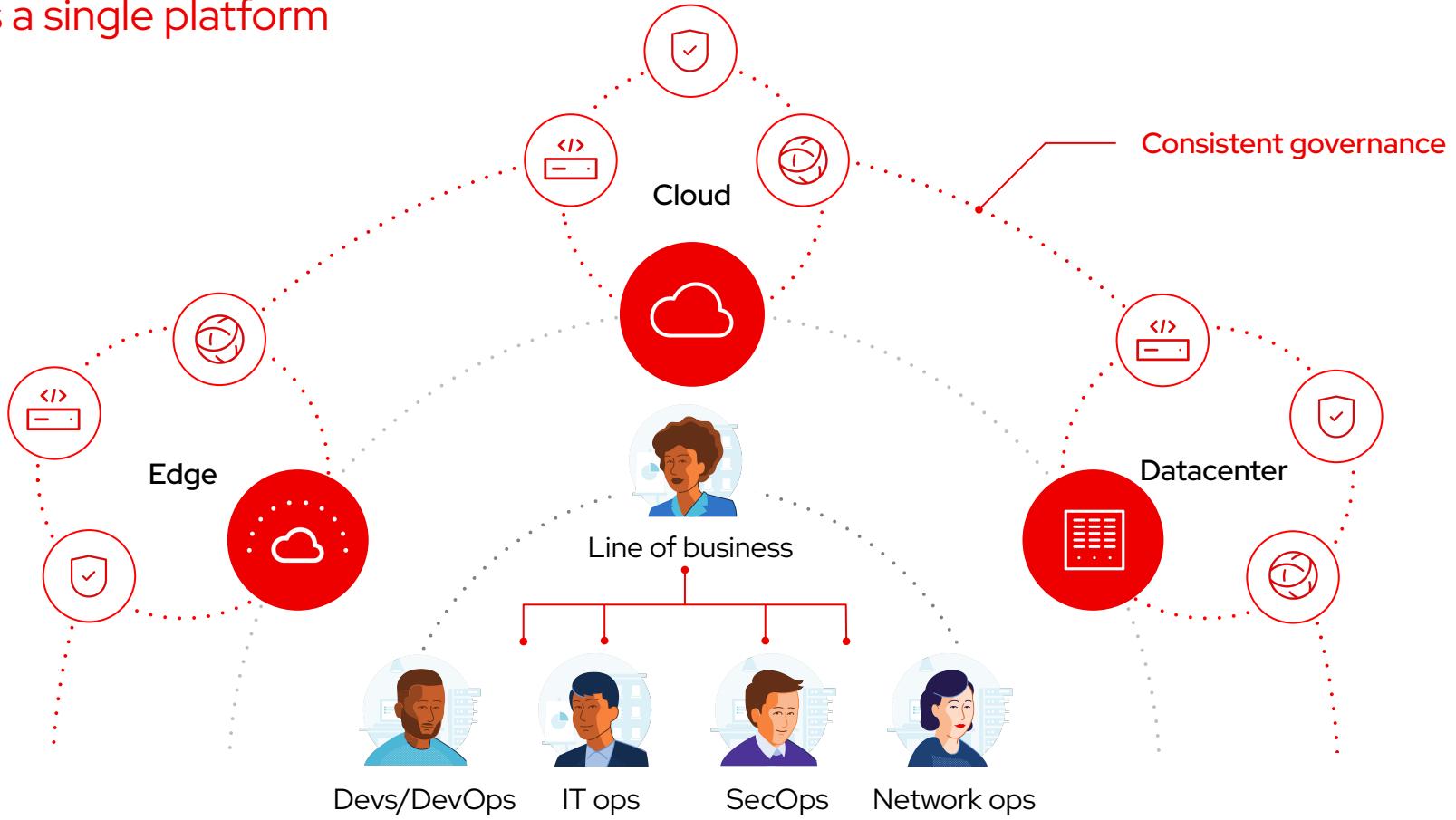
# Many organizations share the same challenge

Too many unintegrated, domain-specific tools



# Break down silos

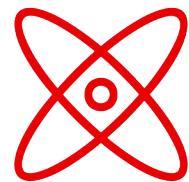
Different teams a single platform



---

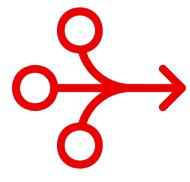
# Why the Red Hat<sup>®</sup> Ansible<sup>®</sup> Automation Platform?

# Why the Ansible Automation Platform?



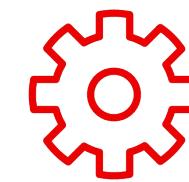
## Powerful

Orchestrate complex  
processes at enterprise scale.



## Simple

Simplify automation creation  
and management across  
multiple domains.



## Agentless

Easily integrate with  
hybrid environments.

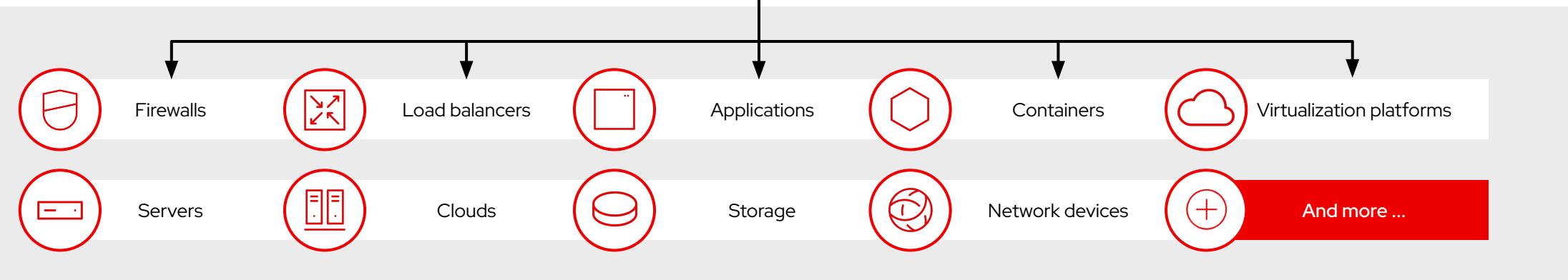
# Automate the deployment and management of automation

Your entire IT footprint

Do this...

Orchestrate      Manage configurations      Deploy applications      Provision / deprovision      Deliver continuously      Secure and comply

On these...



90+  
certified platforms



Infrastructure



Cloud



Network



Security



ARISTA



Check Point®  
SOFTWARE TECHNOLOGIES LTD



FORTINET®

---

# What makes a platform?



## Red Hat Ansible Automation Platform

Combining the universal automation language with cloud services and certified content for automating, deploying, and operating applications, infrastructure and services securely at enterprise scale.



### Ansible automation

Providing scalable, secure implementation for describing, building, and managing the deployment of enterprise IT applications across diverse enterprise architectures.

### Cloud services

Cloud services that facilitate team collaboration and provide operational analytics for automating heterogeneous, hybrid environments.

### Certified content

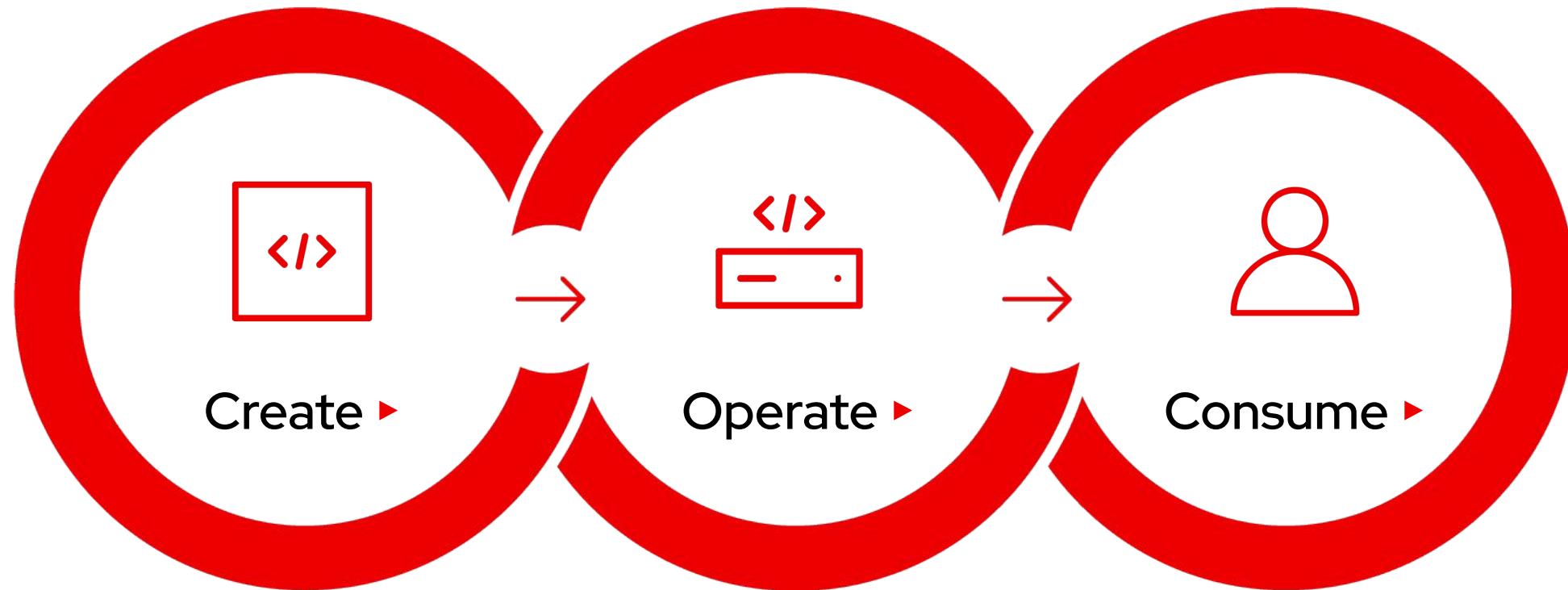
Extends native platform capabilities with certified, supported content designed to expand the automation domain and accelerate adoption for enterprise customers.



**Red Hat**

Ansible Automation  
Platform

Holistic automation for your enterprise .....





## Red Hat Ansible Automation Platform



Content creators



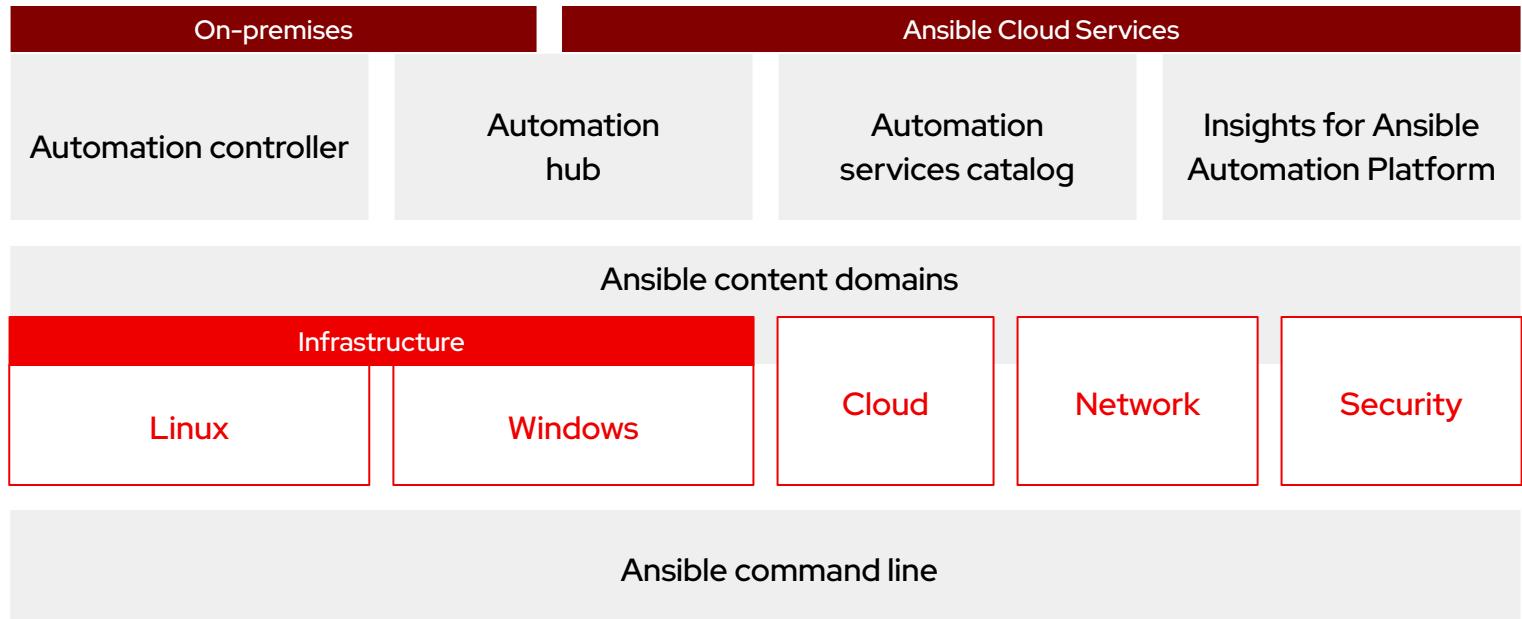
Operators



Domain experts



Users



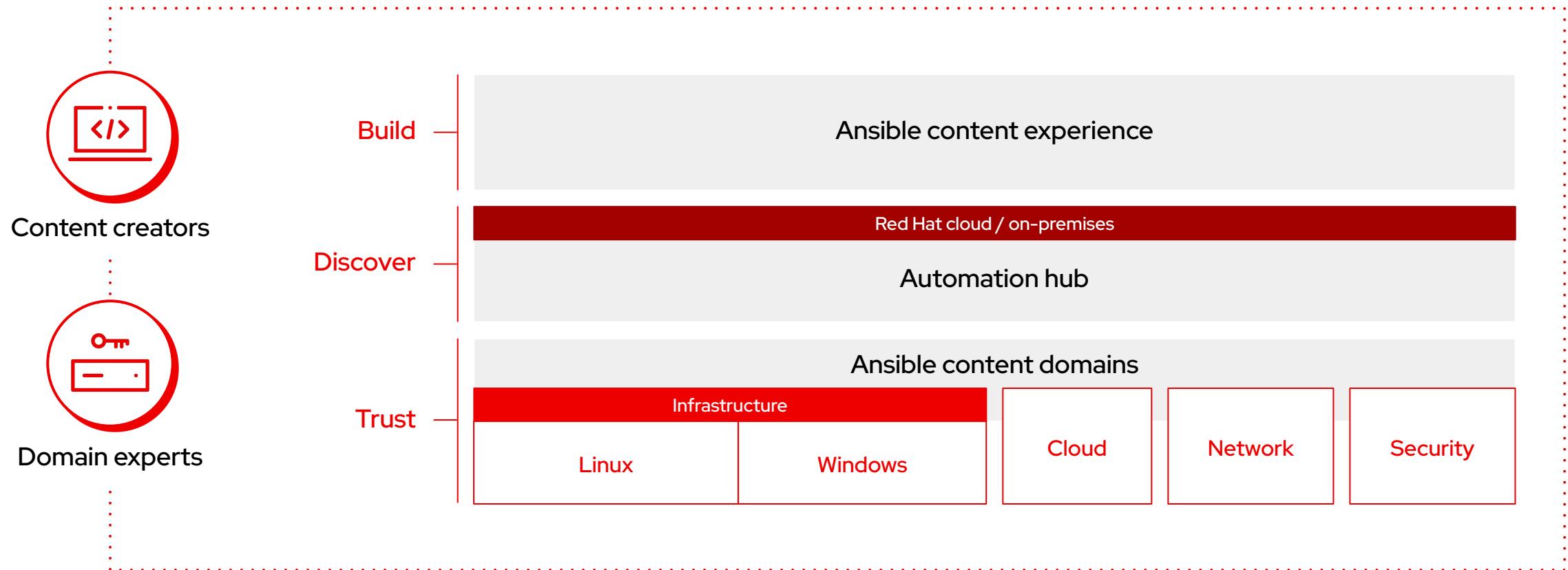
Fueled by an  
open source community



# Create

# Create

## The automation lifecycle





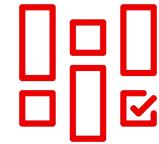
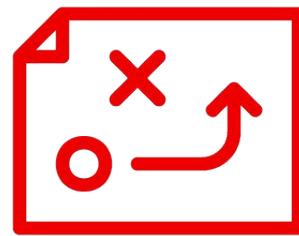
```
---
- name: install and start apache
  hosts: web
  become: yes

  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
```

# What makes up an Ansible playbook?



Plays

Modules

Plugins

# Ansible plays

What am I automating?



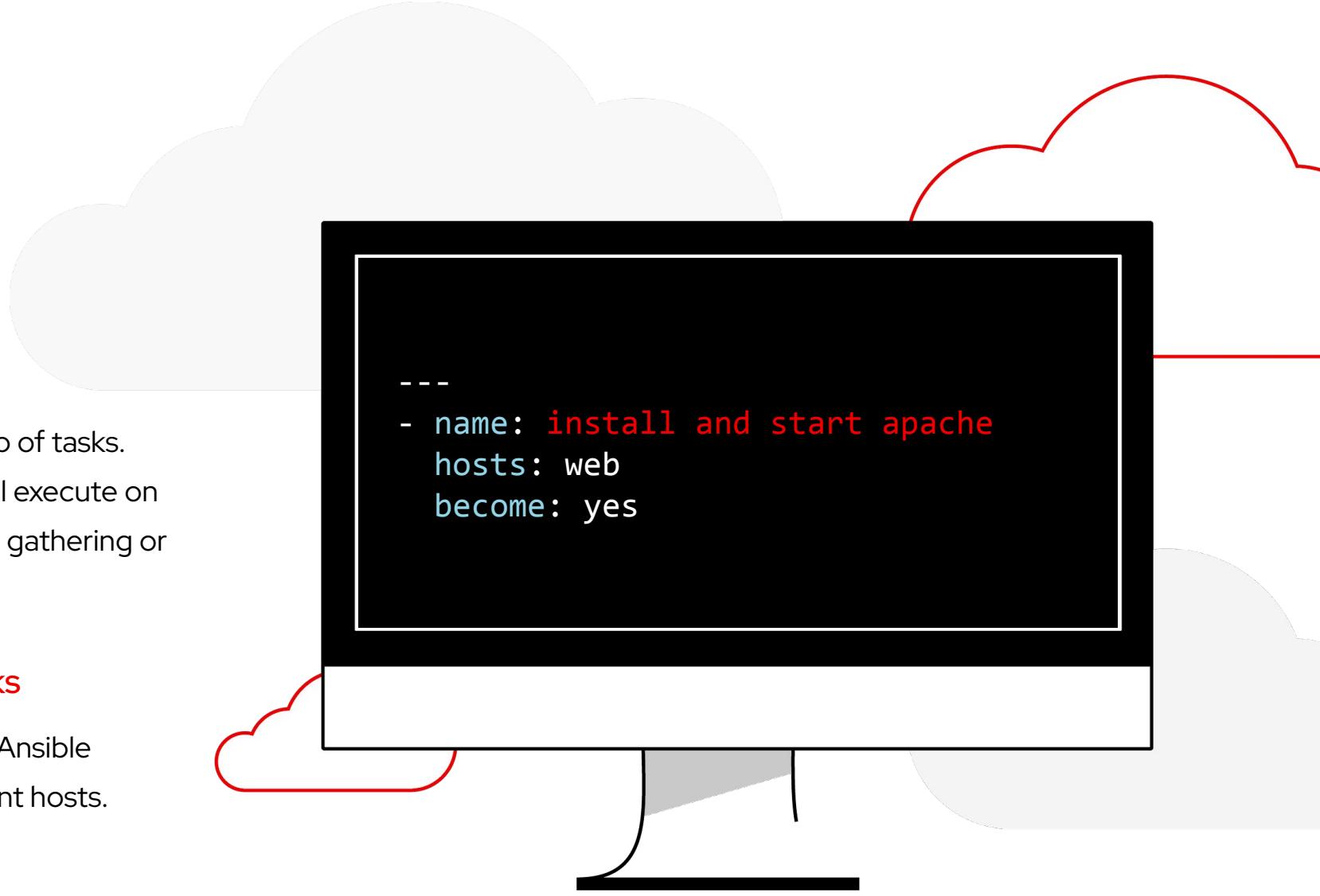
## What are they?

Top level specification for a group of tasks.  
Will tell that play which hosts it will execute on  
and control behavior such as fact gathering or  
privilege level.



## Building blocks for playbooks

Multiple plays can exist within an Ansible  
playbook that execute on different hosts.



# Ansible modules

The “tools in the toolkit”



## What are they?

Parametrized components with internal logic,  
representing a single step to be done.  
The modules “do” things in Ansible.



## Language

Usually Python, or Powershell for Windows  
setups. But can be of any language.



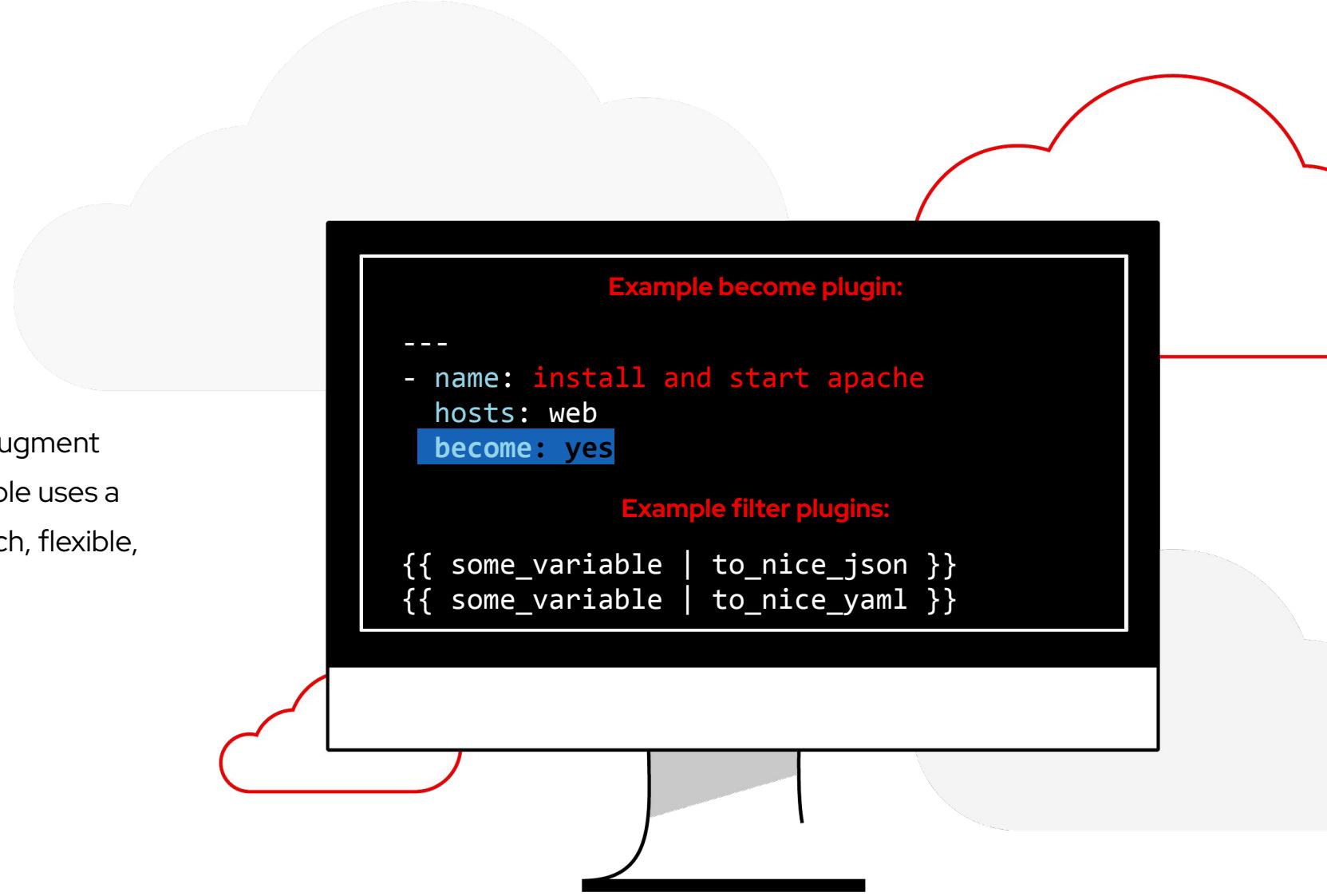
# Ansible plugins

The “extra bits”



## What are they?

Plugins are pieces of code that augment Ansible's core functionality. Ansible uses a plugin architecture to enable a rich, flexible, and expandable feature set.



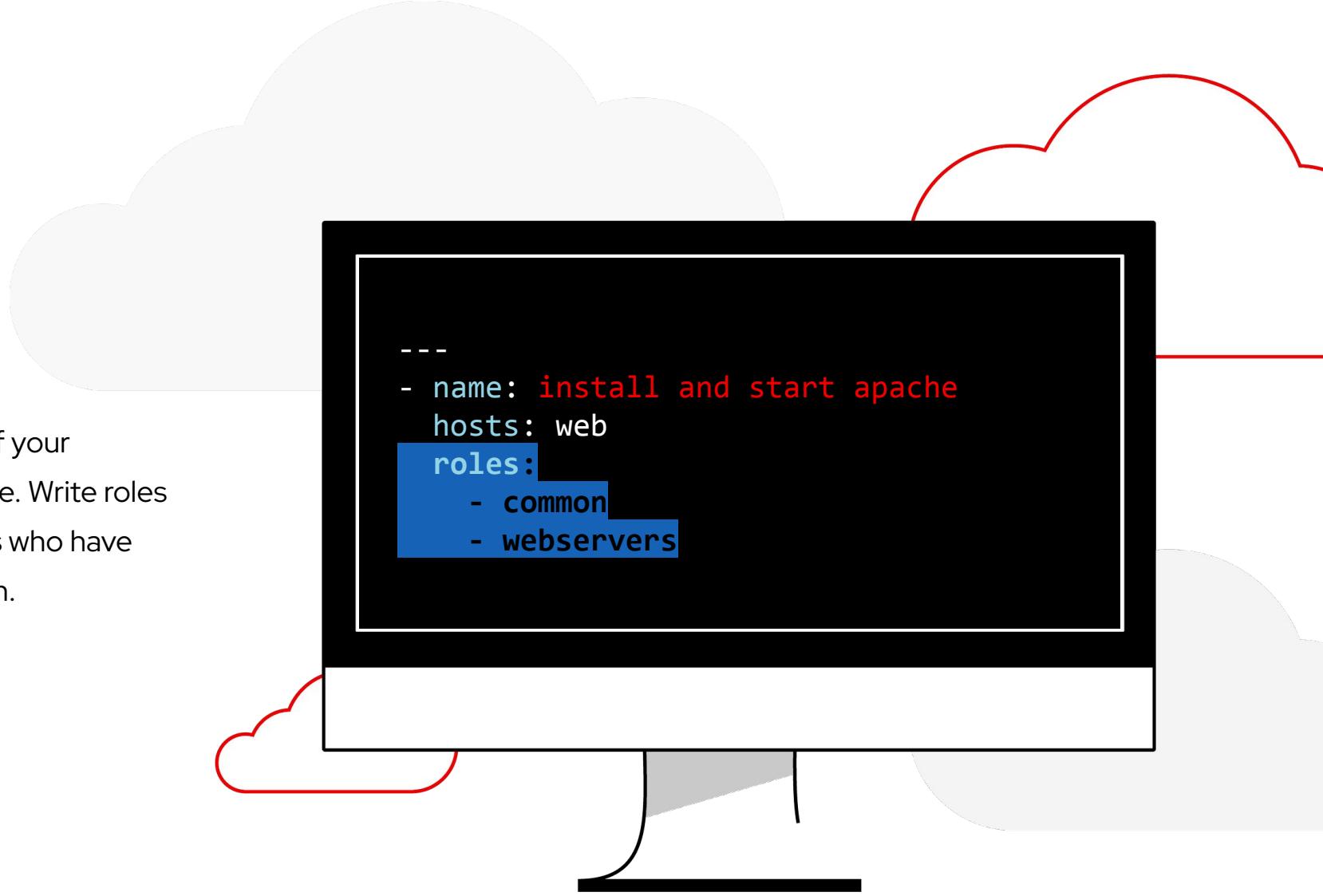
# Ansible roles

Reusable automation actions



## What are they?

Group your tasks and variables of your automation in a reusable structure. Write roles once, and share them with others who have similar challenges in front of them.



# Collections

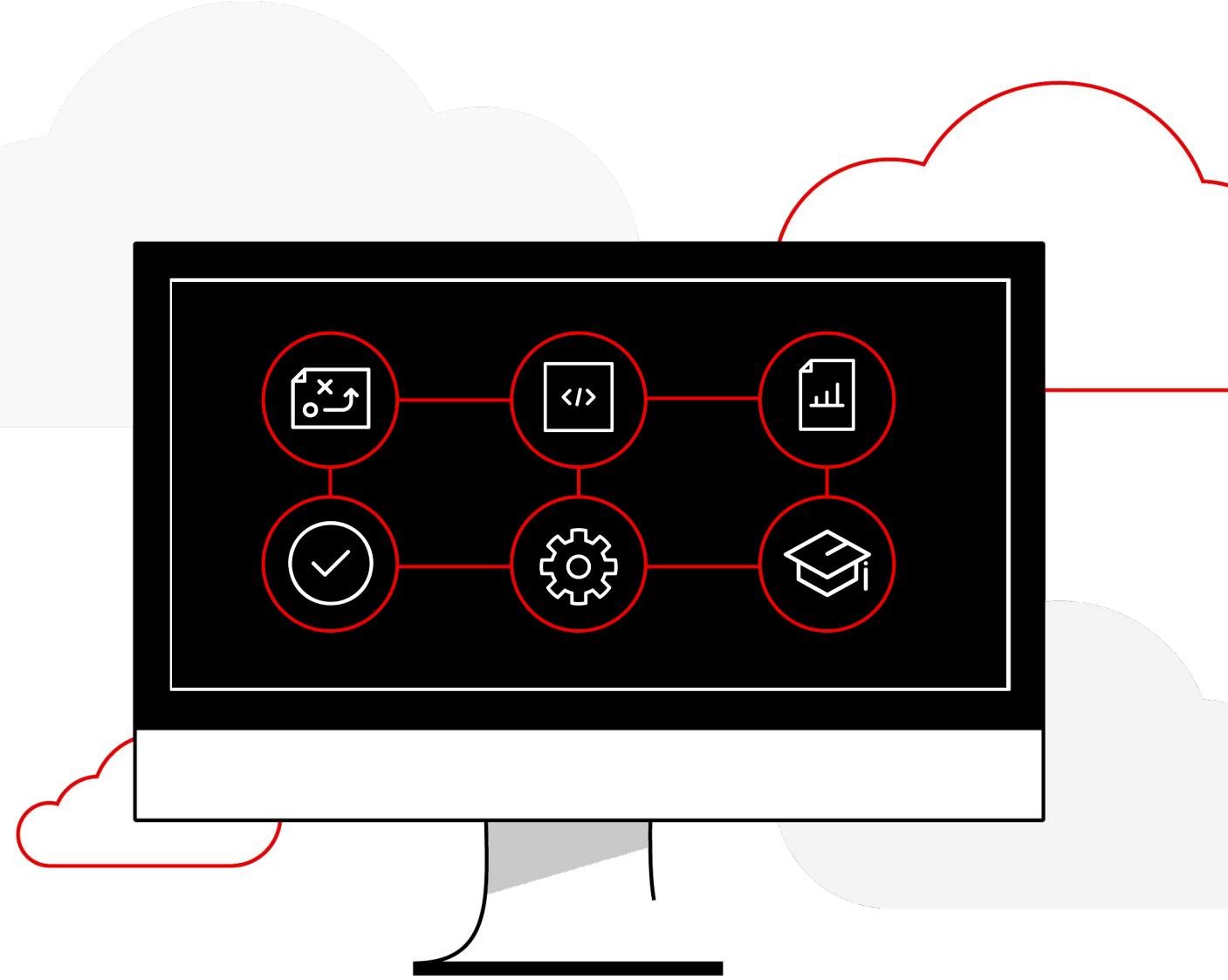
Simplified and consistent content delivery

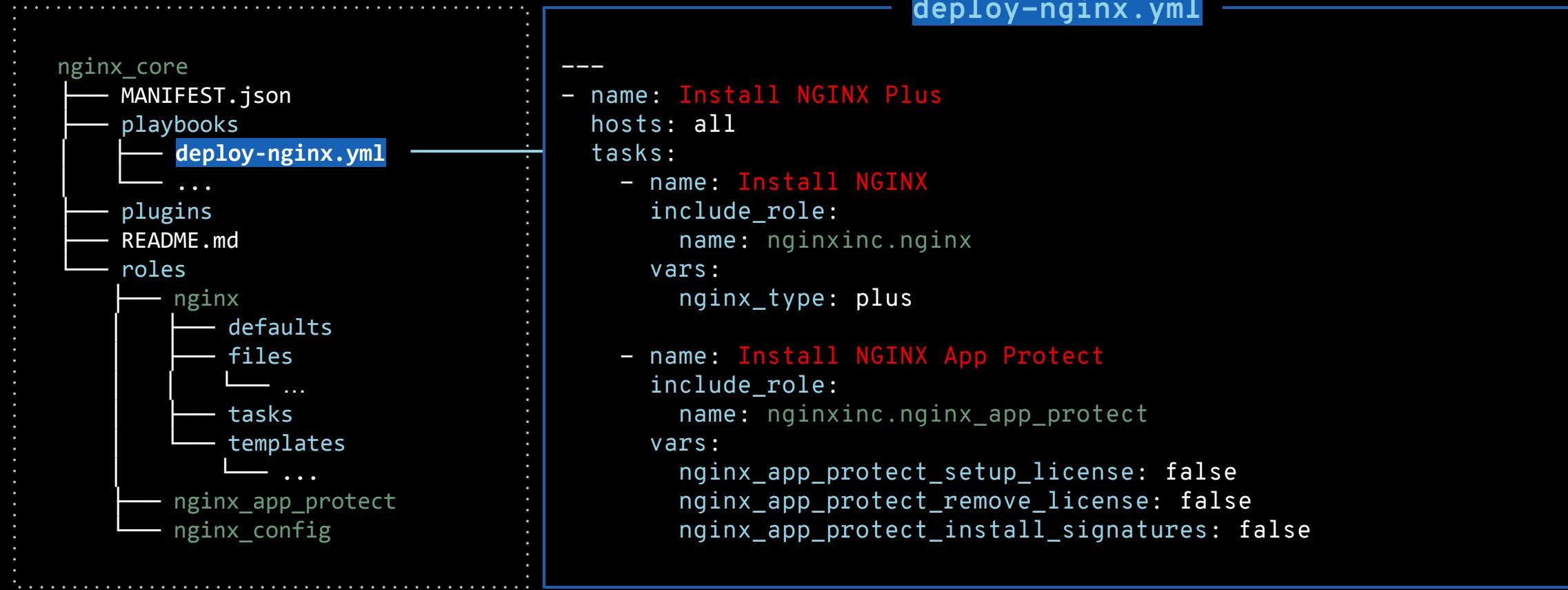


## What are they?

Collections are a data structure containing automation content:

- ▶ Modules
- ▶ Playbooks
- ▶ Roles
- ▶ Plugins
- ▶ Docs
- ▶ Tests





### deploy-nginx.yml

```
---
```

- name: Install NGINX Plus  
hosts: all  
tasks:
  - name: Install NGINX  
include\_role:
    - name: nginxinc.nginx
  - vars:
    - nginx\_type: plus
- name: Install NGINX App Protect  
include\_role:
  - name: nginxinc.nginx\_app\_protect
- vars:
  - nginx\_app\_protect\_setup\_license: false
  - nginx\_app\_protect\_remove\_license: false
  - nginx\_app\_protect\_install\_signatures: false

# Automation Hub

## Trusted source

### Customer controlled

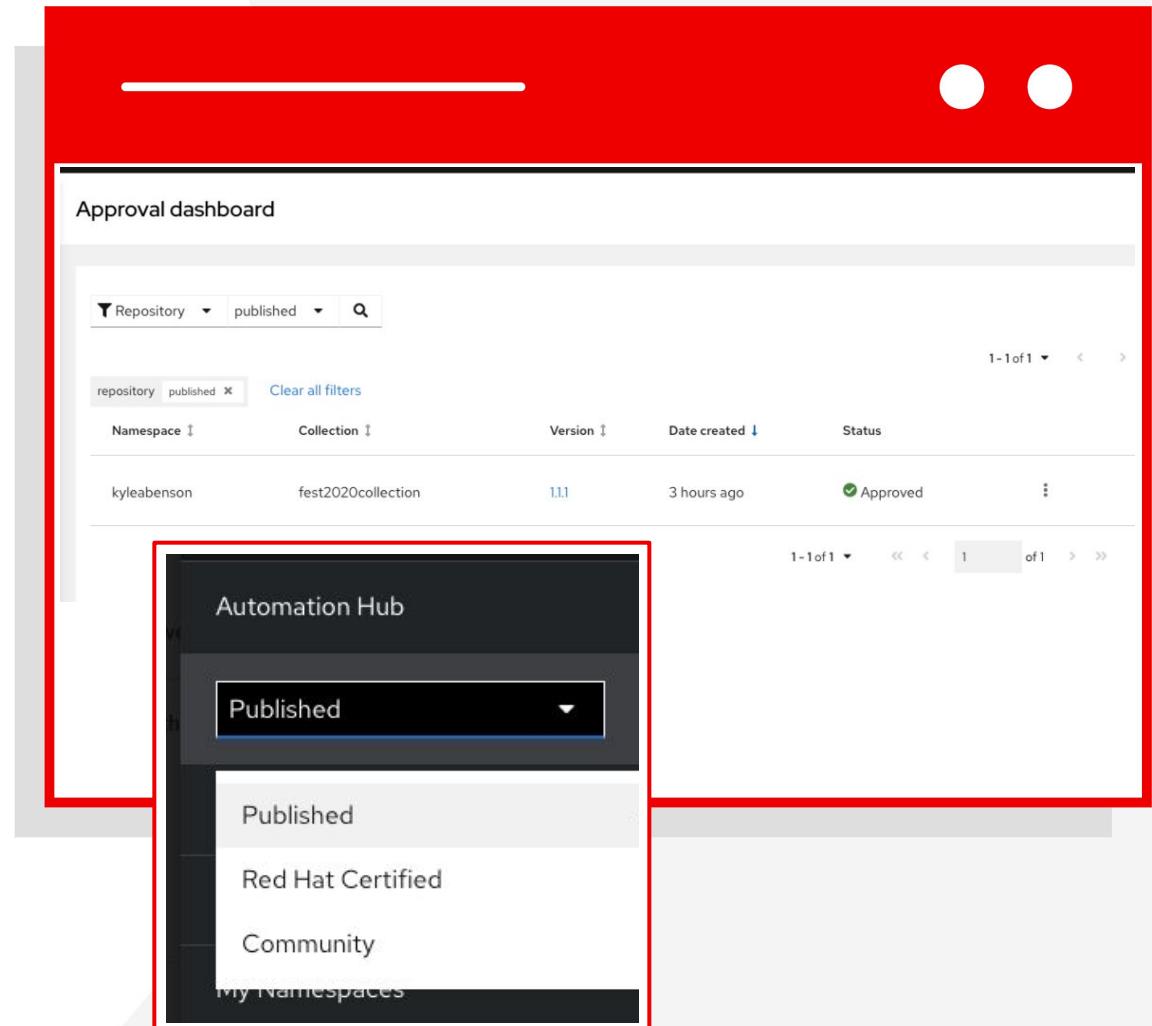
Deploying either on-prem or to a cloud, customers can run their own private instances of Automation Hub integrated into Red Hat Ansible Platform.

### Private content

Manage the lifecycle and internal distribution of in-house Ansible content within Private Automation hub.

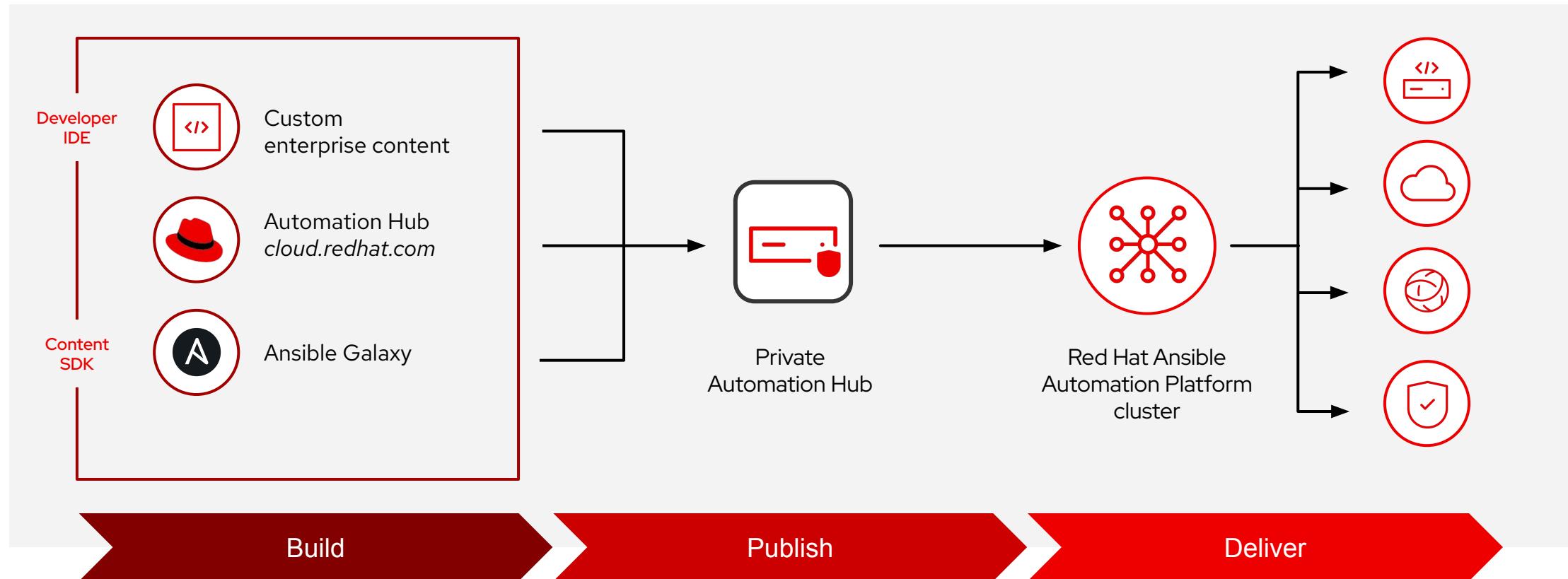
### Customizable Content Catalog

Via synch from community (Galaxy) and supported (Automation Hub) sources, customers can supply internal users with approved content in one controlled location in Private Automation hub.



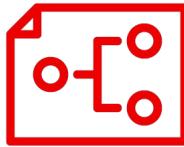
# Private Automation Hub architecture

Value of private Automation Hub



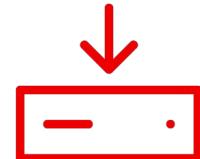
# Accessing collections

How to get them



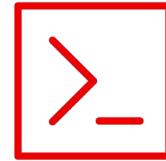
## Requirements file

Requirements file defines the required collections for a playbook



## Pull via controller

Automation controller pulls the collections from Automation Hub automatically

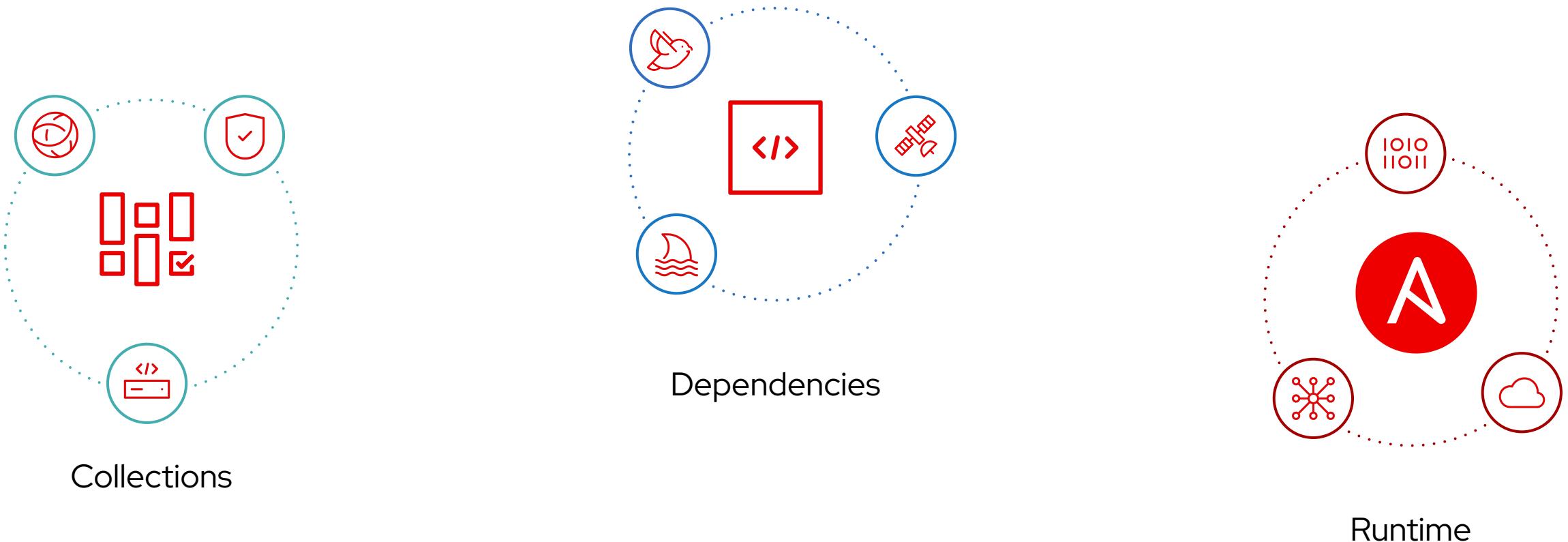


## Command line

CLI access is also possible via ansible-galaxy command

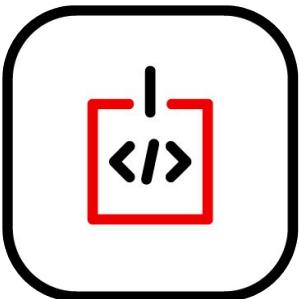
# Many technologies, different life cycles

How to keep runtime environment, collections and dependencies aligned?



# Automation Execution Environments

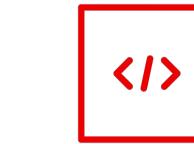
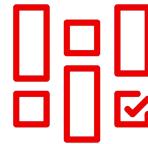
Components needed for automation, packaged in a cloud-native way



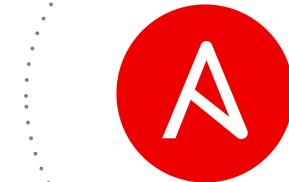
Execution  
Environments



Collections



Libraries

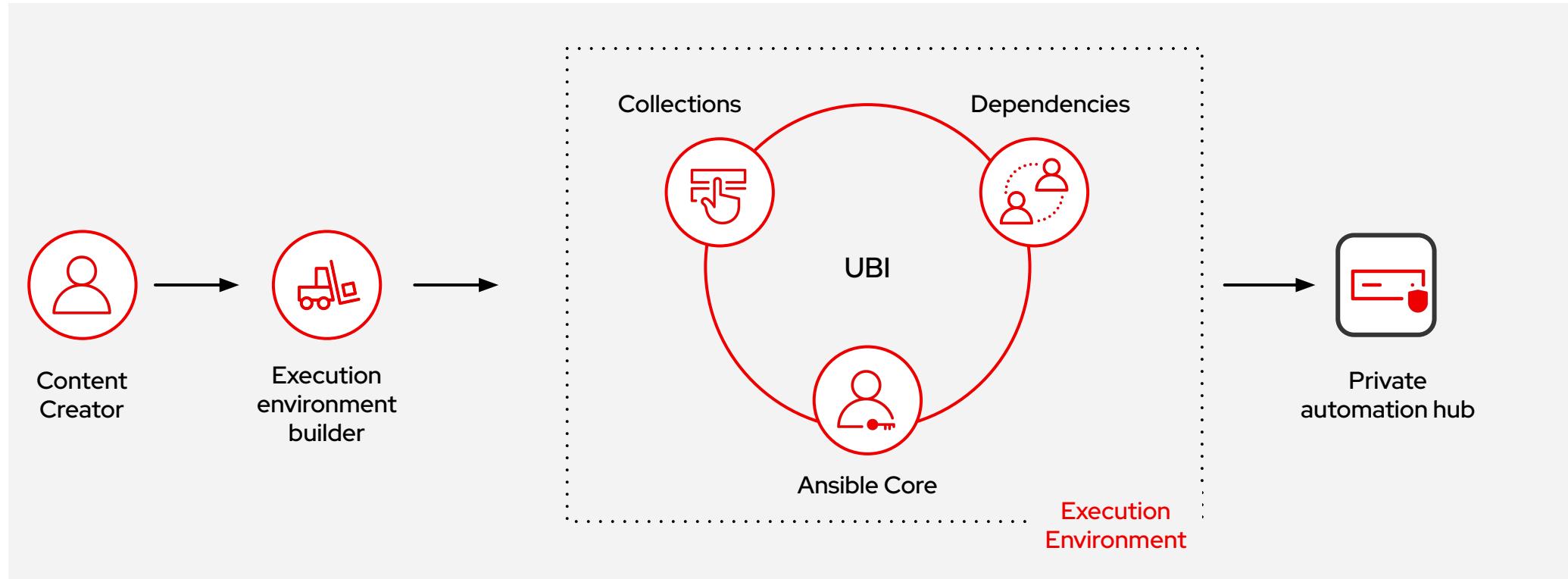


Ansible Core

Universal Base Image

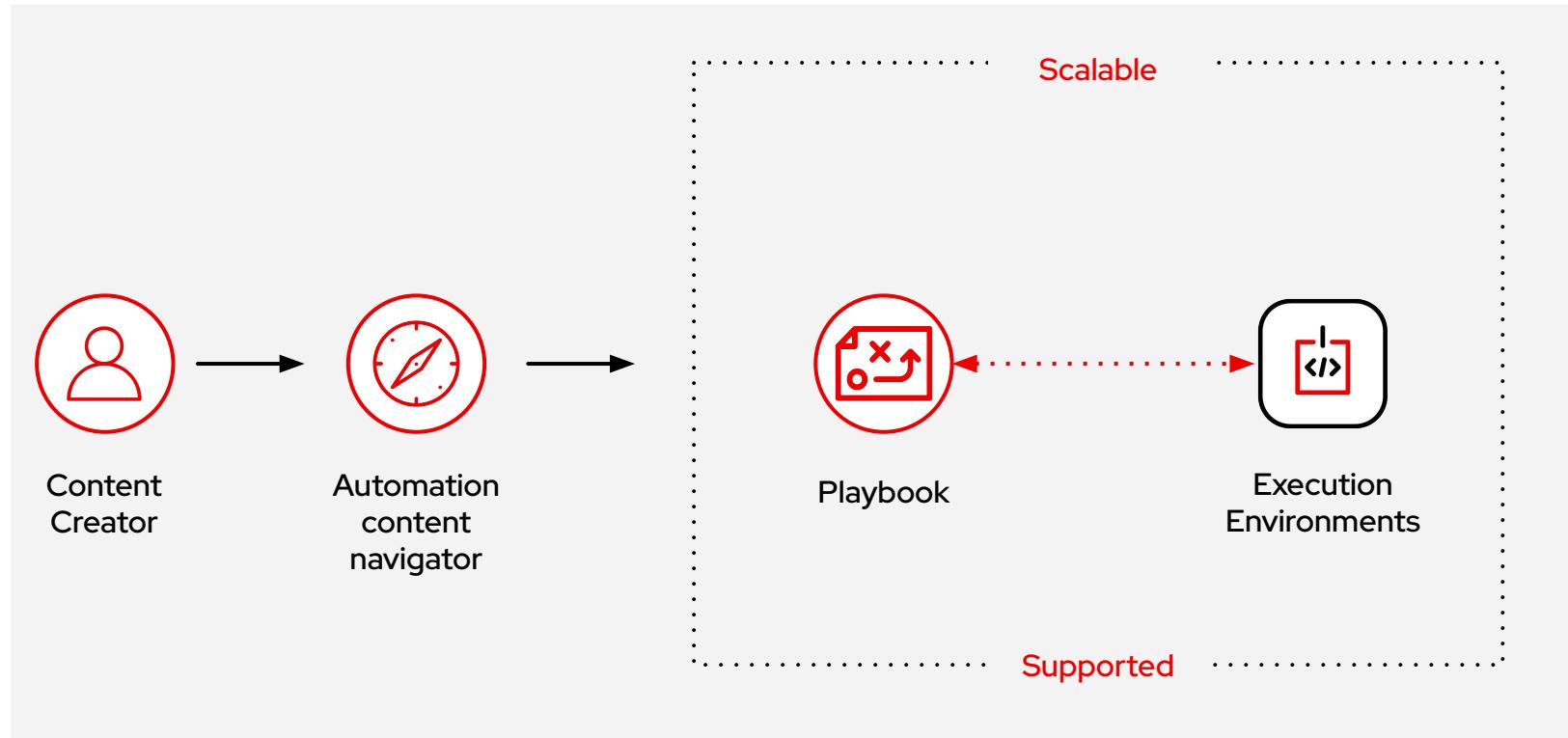
# Build, create, publish

Development cycle of an automation execution environment

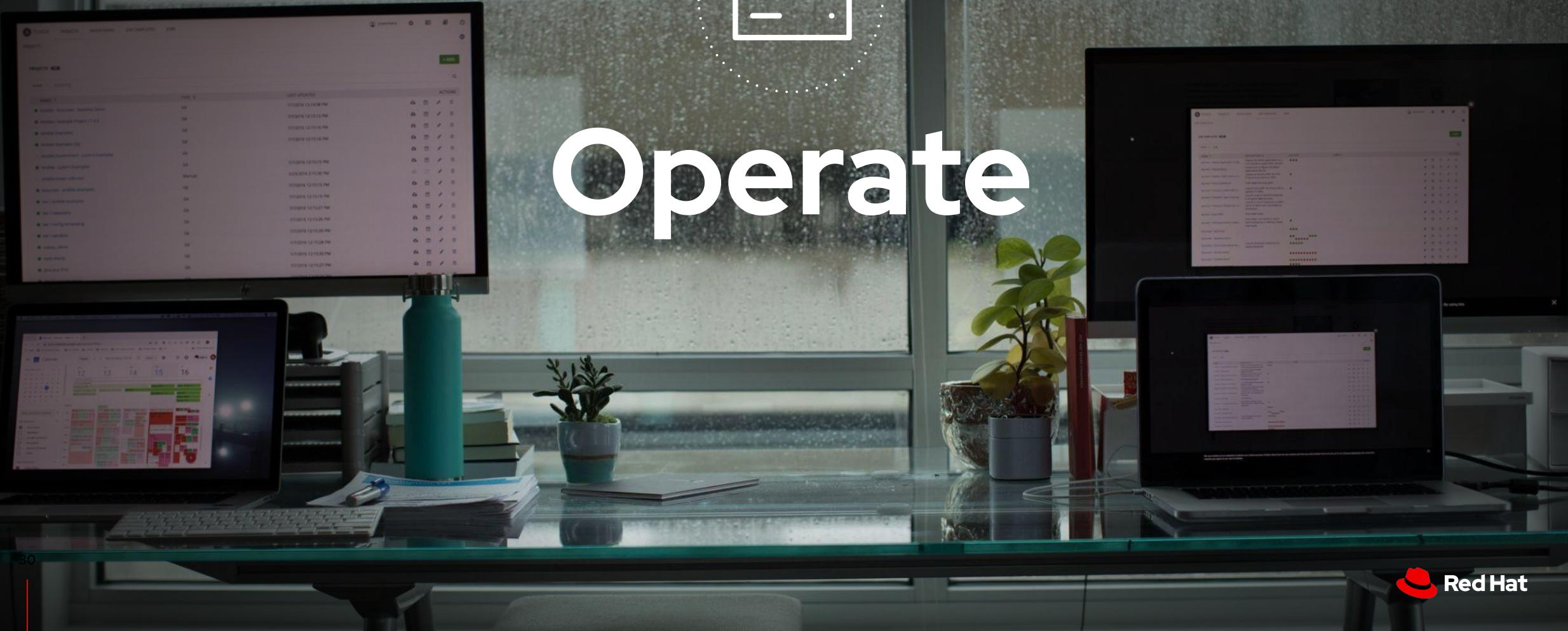


# Develop, test, run

How to develop, test and run containerized Ansible content

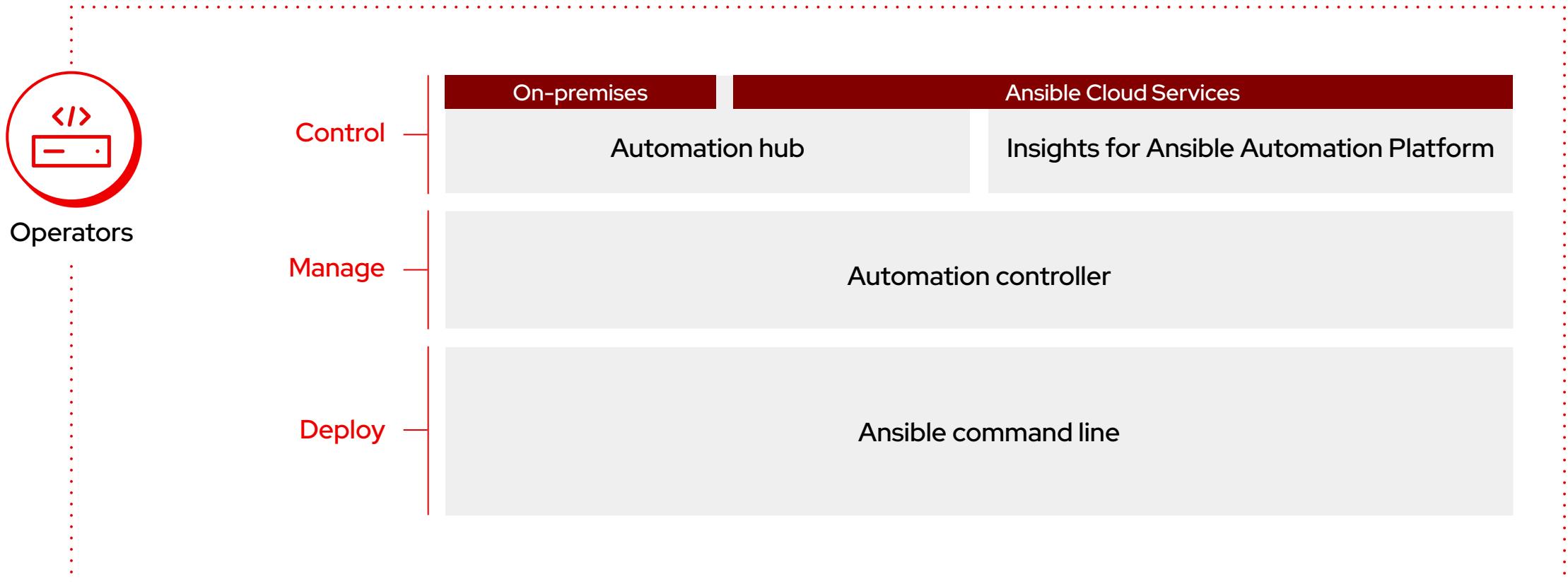


# Operate



# Operate

## The automation lifecycle



# A playbook run

## Where it all starts

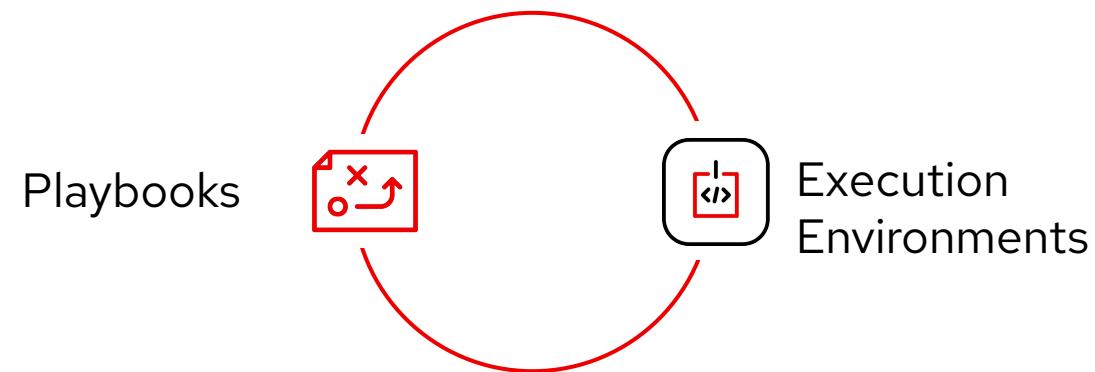
- ▶ A playbook is interpreted and run against one or multiple hosts - task by task. The order of the tasks defines the execution.
- ▶ In each task, the module does the actual work.



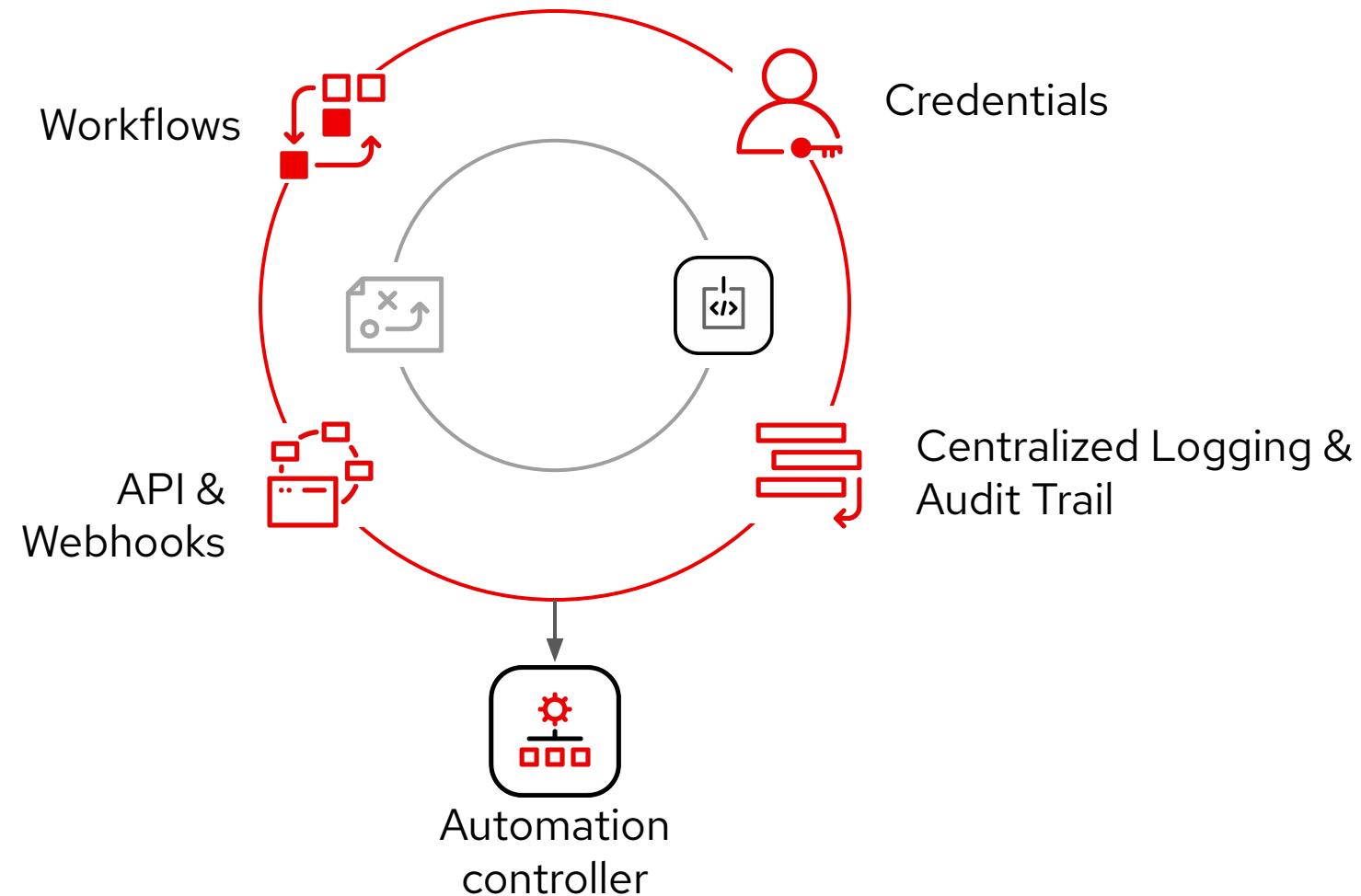
The screenshot shows a terminal window with a red header containing two white circles and a search bar with 'SEARCH' and 'KEY' buttons. The main area displays a log of a playbook run. The log includes:

```
1 Identity added: /tmp/awx_2896_5sdng5le/artifacts/2896/ssh_key_data (/tmp/awx_2896_5sdng5le/artifacts/2896/ssh_key_data)
2
3 PLAY [install and start apache] *****
4
5 TASK [Gathering Facts] *****
6 ok: [node1]
7 ok: [node3]
8 ok: [node2]
9
10 TASK [httpd package is present] *****
11 changed: [node1]
12 changed: [node2]
13 changed: [node3]
14
15 TASK [latest index.html file is present] *****
16 changed: [node1]
17 changed: [node2]
18 changed: [node3]
19
20 TASK [httpd is started] *****
21 changed: [node1]
22 changed: [node2]
23 changed: [node3]
24
25 PLAY RECAP *****
26 node1 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
27 node2 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
28 node3 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
29
```

# Components of Automation



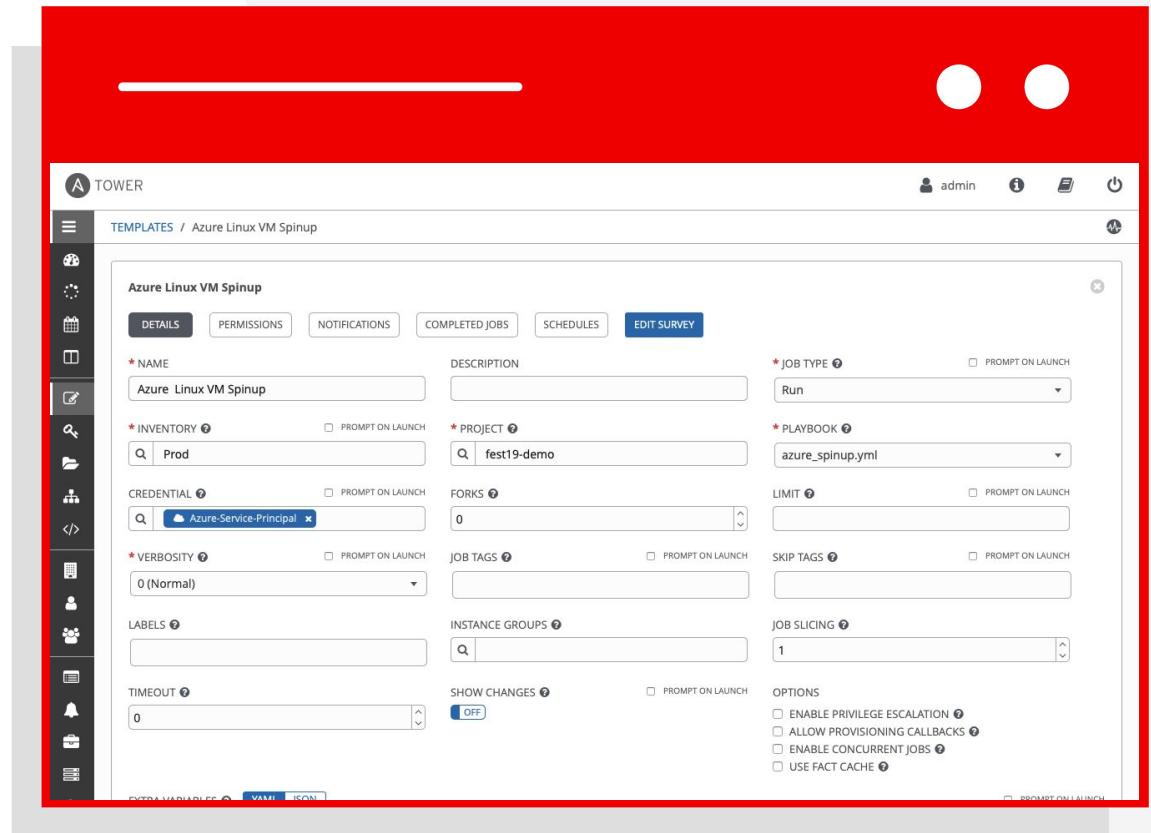
# Anatomy of Automation Operation



# Execution of content

## Running at the core

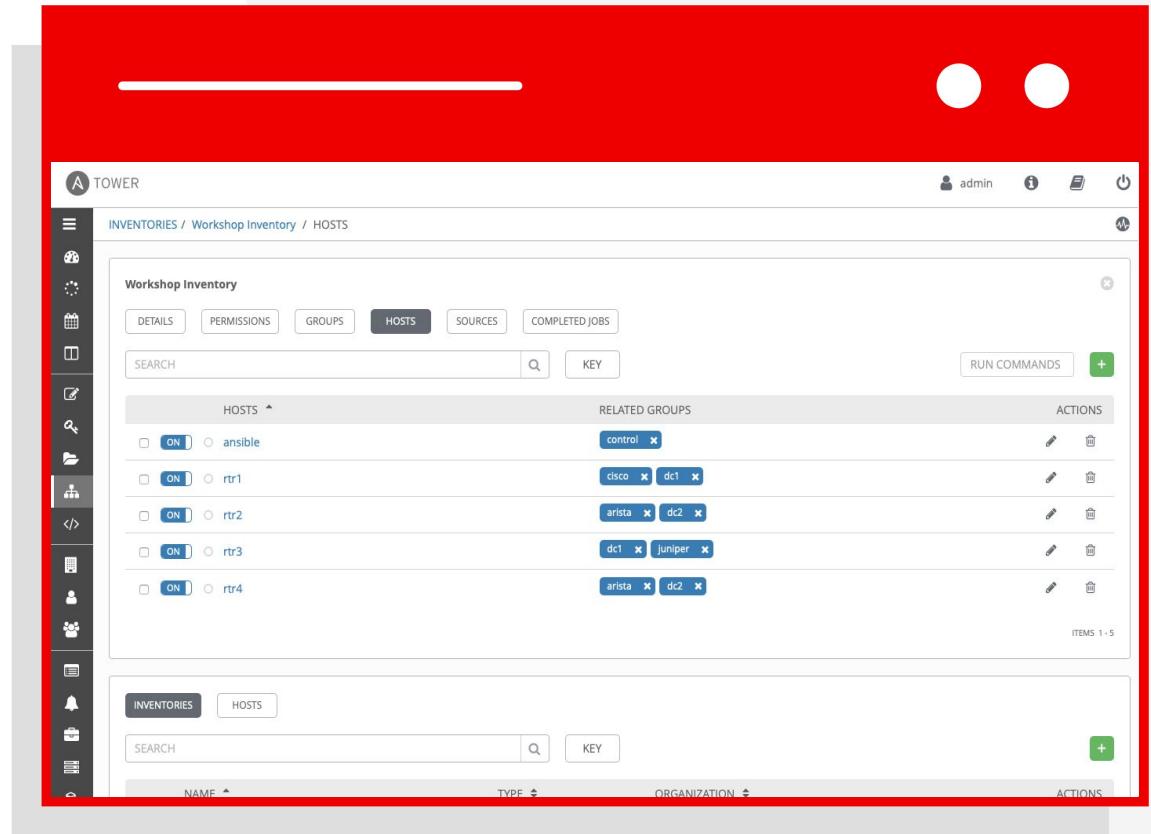
- ▶ The central execution of automation content is managed and done either via central cluster..
- ▶ Can also sync git repositories, takes care of execution environments, collections, credentials, inventory and logging.
- ▶ Full audit trail of the execution, including what version of content was executed, what variable values were provided, etc.



# Inventories and credentials

## How to talk to others

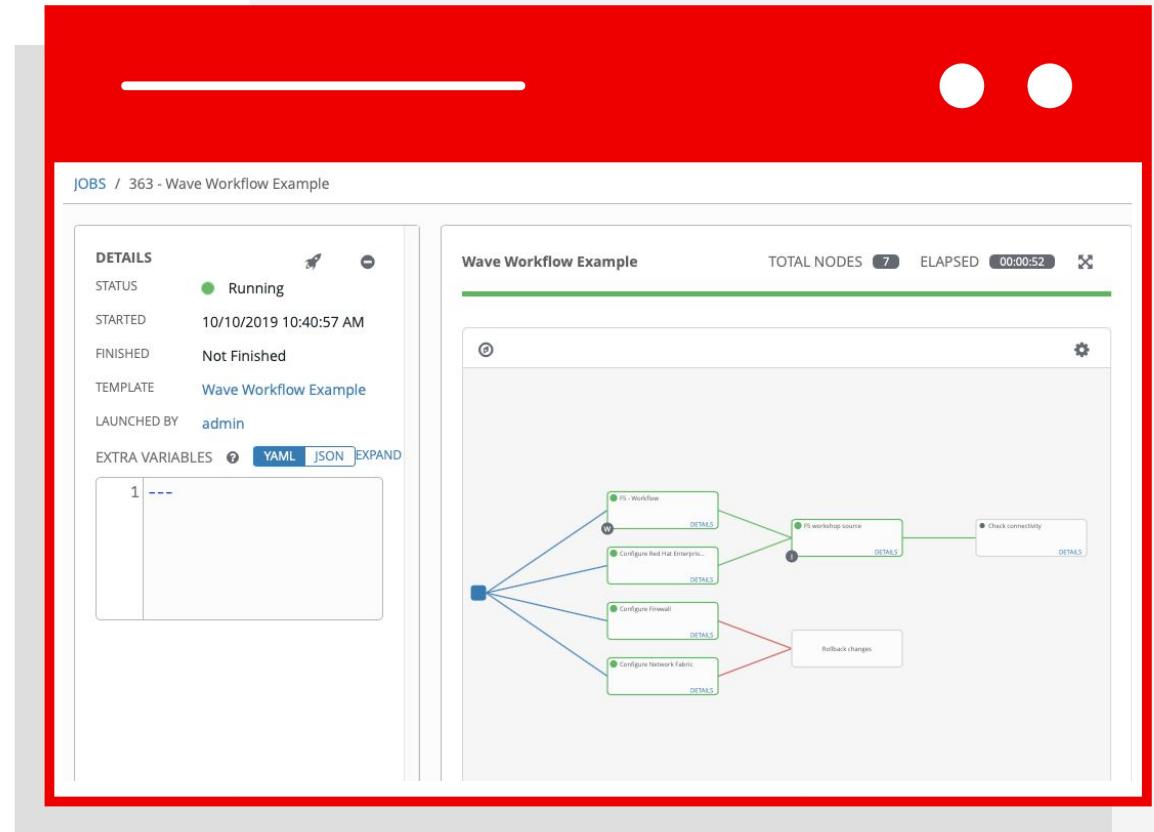
- ▶ An inventory is a collection of hosts (nodes) with associated data and groupings that the automation platform can connect to and manage:
  - Nodes
  - Groups
  - Can be static or dynamic
  - Smart inventories possible
- ▶ And what usernames and passwords do you use during connection? That is kept in the credentials.



# Workflows

Combine automation to create something bigger

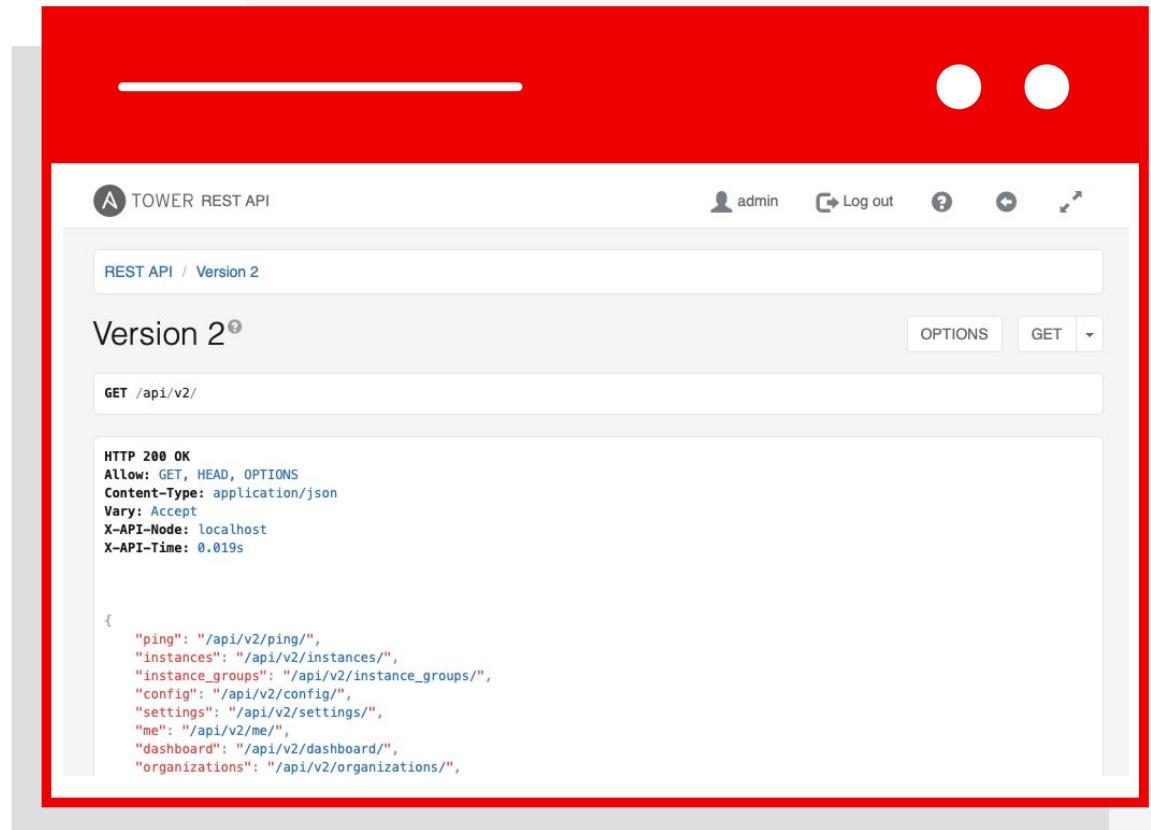
- ▶ Workflows enable the creation of powerful holistic automation, chaining together multiple pieces of automation and events.
- ▶ Simple logic inside these workflows can trigger automation depending on the success or failure of previous steps.



# API

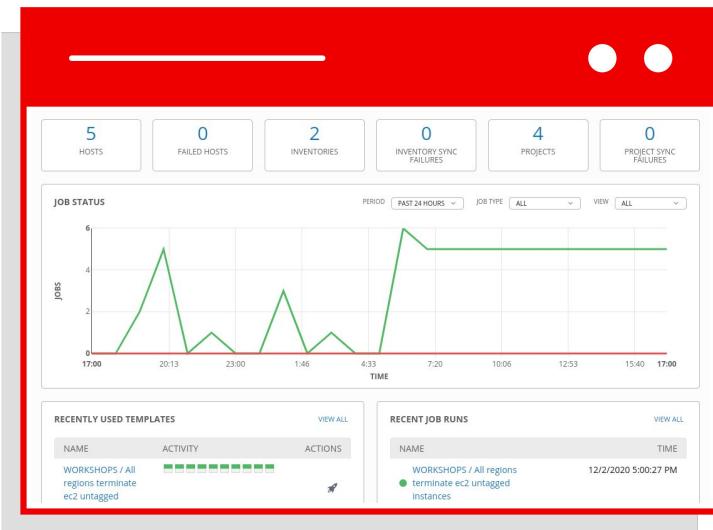
## Integration of automation into larger workflows

- ▶ The API provides programmatic access to the automation via a defined interface.
- ▶ Underneath it is still powered by the same bits and pieces which are at the core: workflows, inventories, etc.
- ▶ It offers simple integration into other tools like ITSM, SOAR, etc.

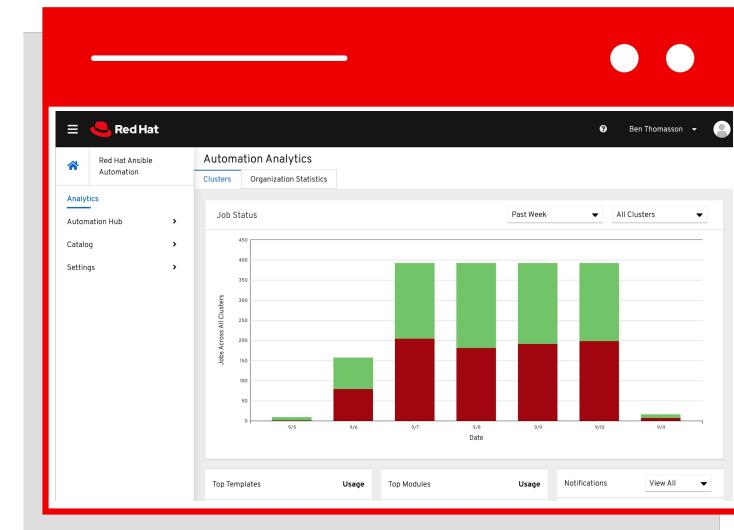


# Dashboards

Understand what is going on



Overview of the actual cluster, the jobs happening, the nodes connected, what works and fails right at this moment.



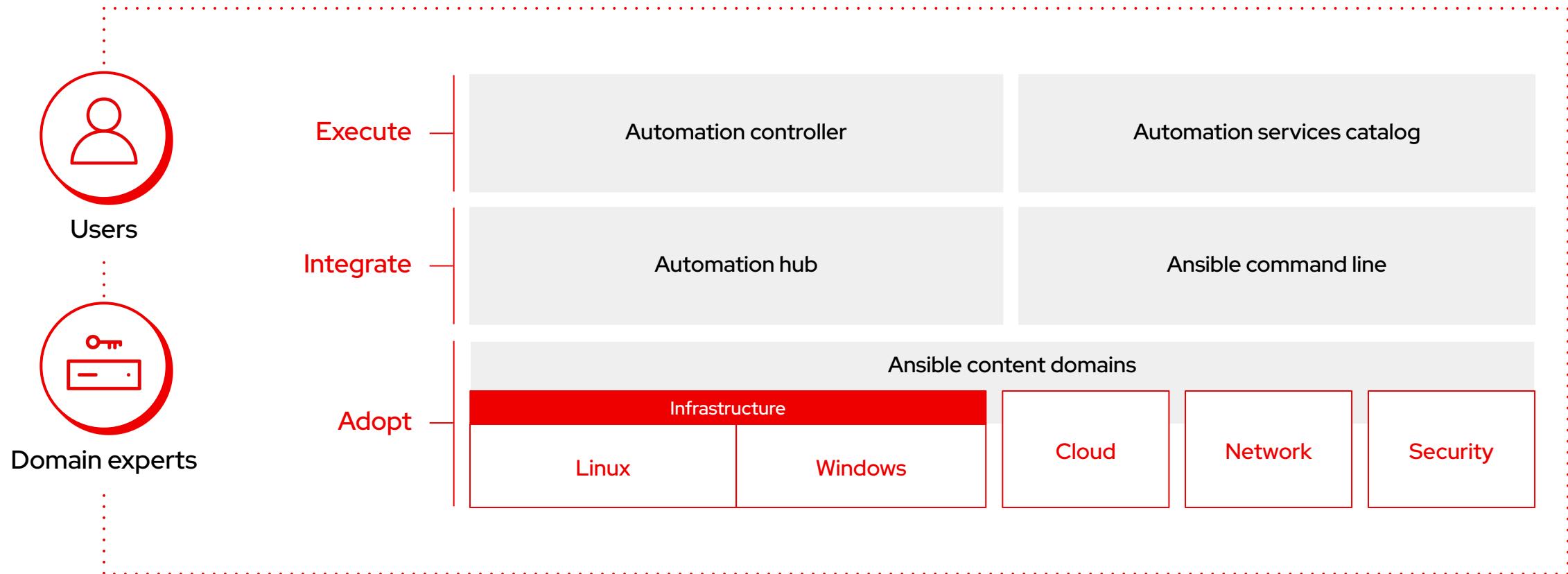
Overview across clusters, better insight into use cases of automation; insight into adoption of automation per organization.

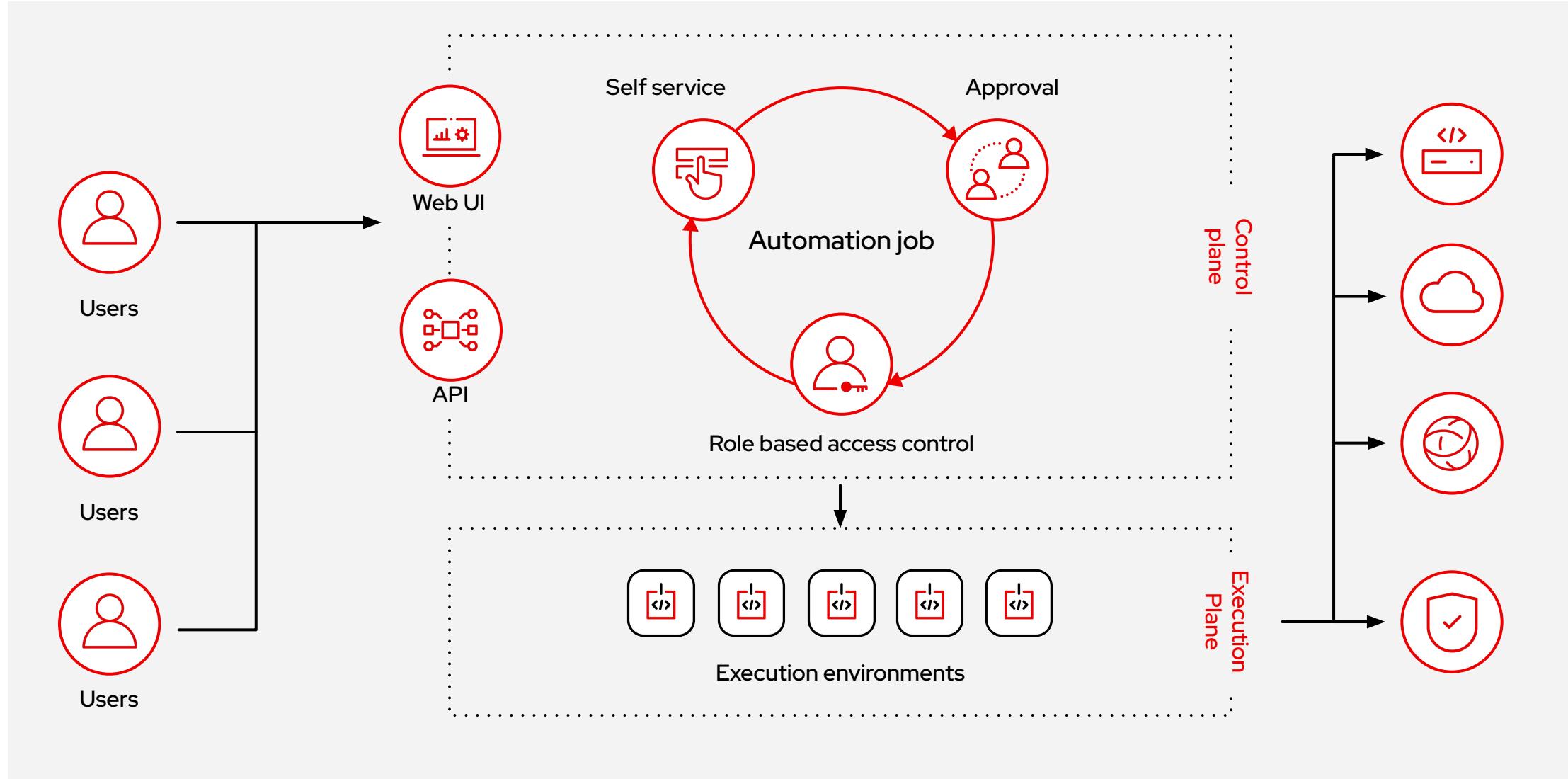
# Consume



# Consume

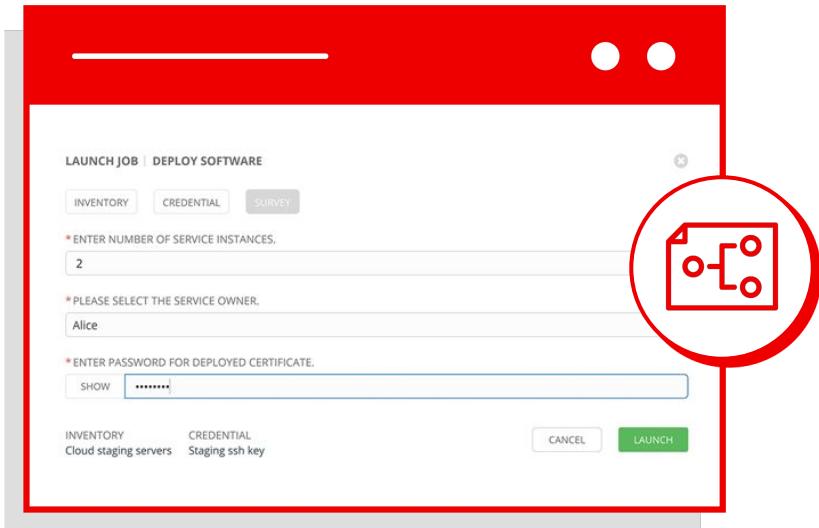
## The automation lifecycle



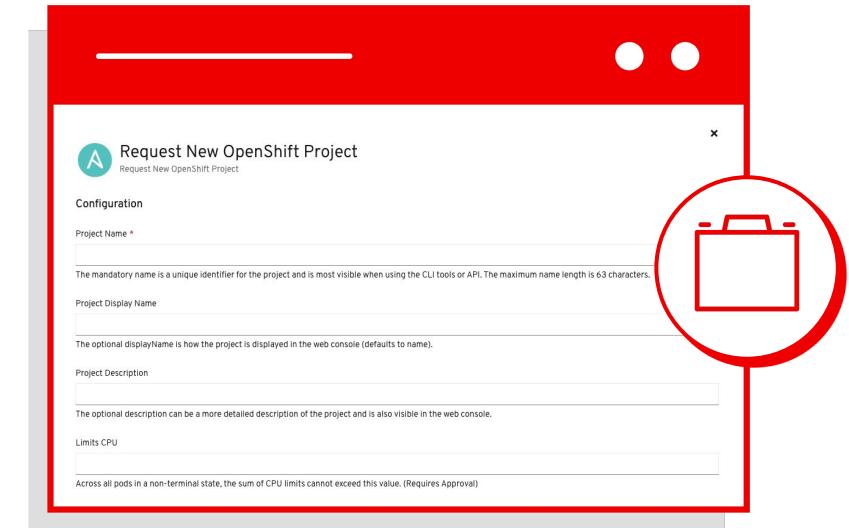


# Self service

Provide others the mean to consume



For teams **accessing directly** the automation environment and are close to the automation platform processes.



For **high level access** distant or not even aware of the underlying automation environment; lines of business for example.

# Role-based access control

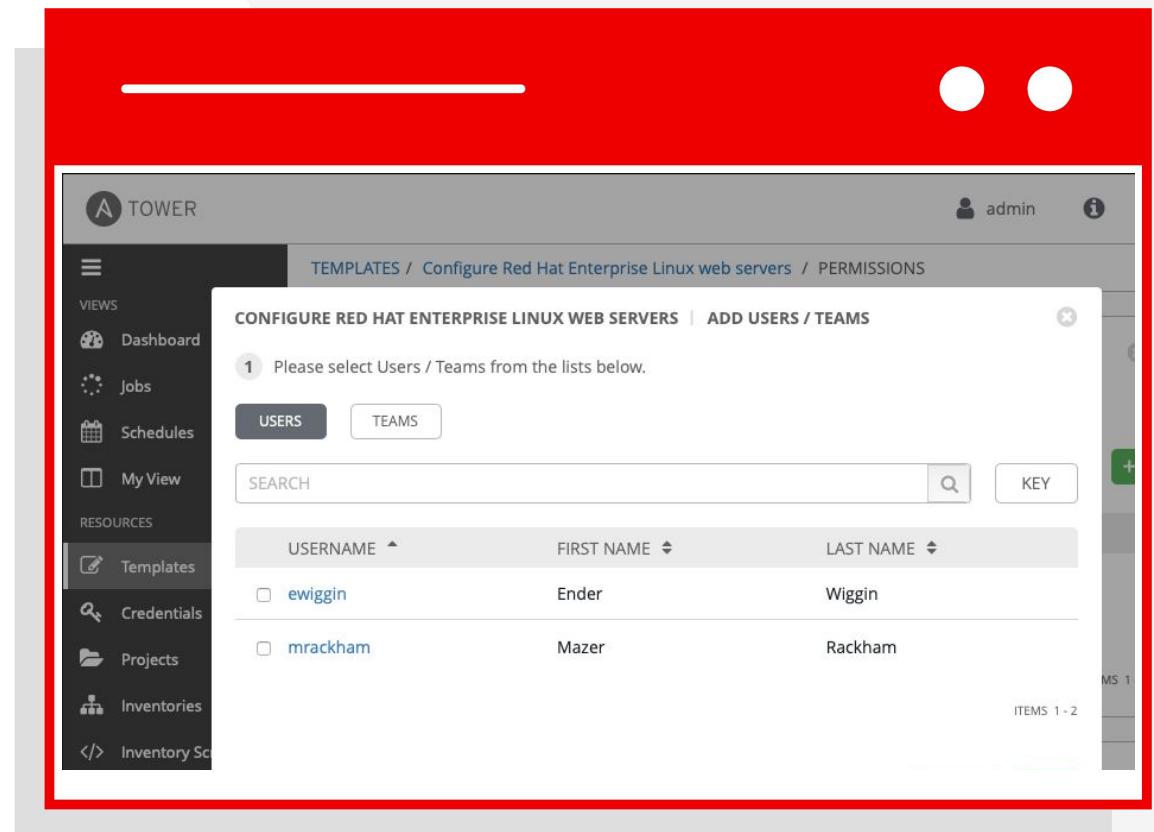
## How to manage access

- ▶ Role-based access control system:

Users can be grouped in teams, and roles can be assigned to the teams.

- ▶ Rights to edit or use can be assigned across all objects.

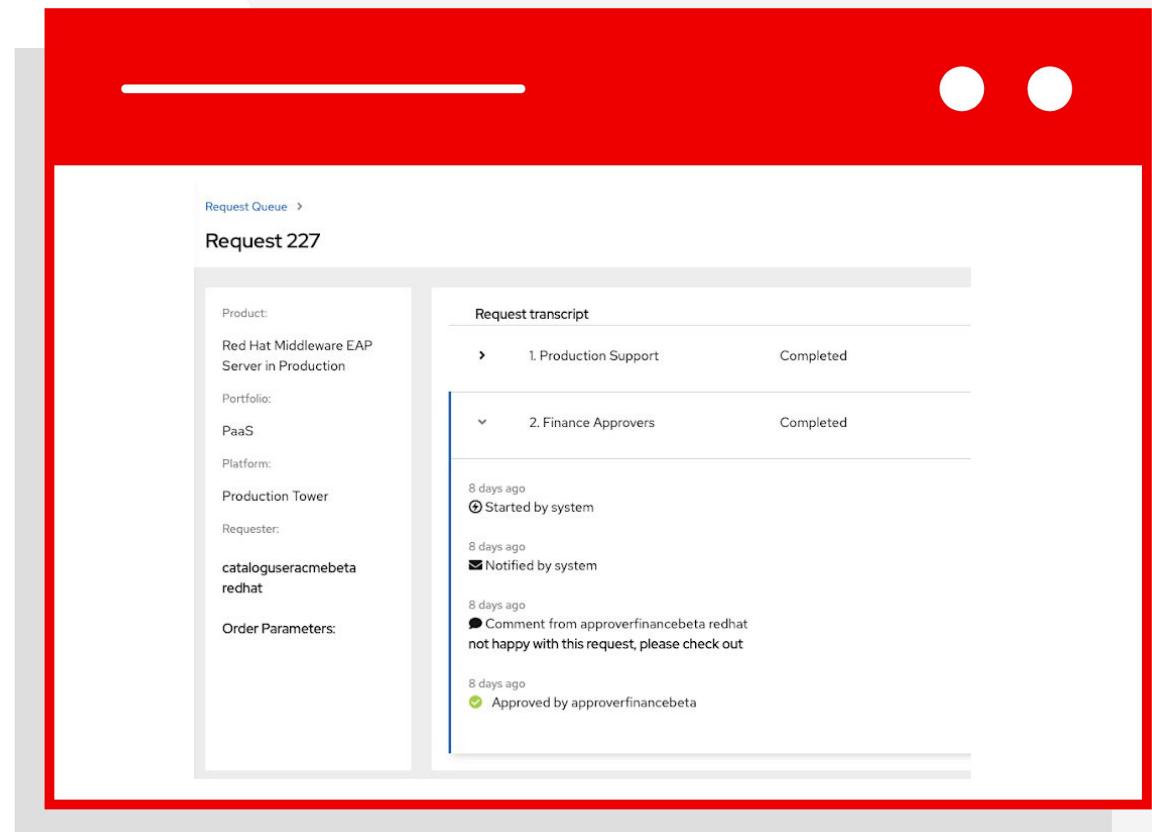
- ▶ All backed by enterprise authentication if needed.



# Approval

## Govern access

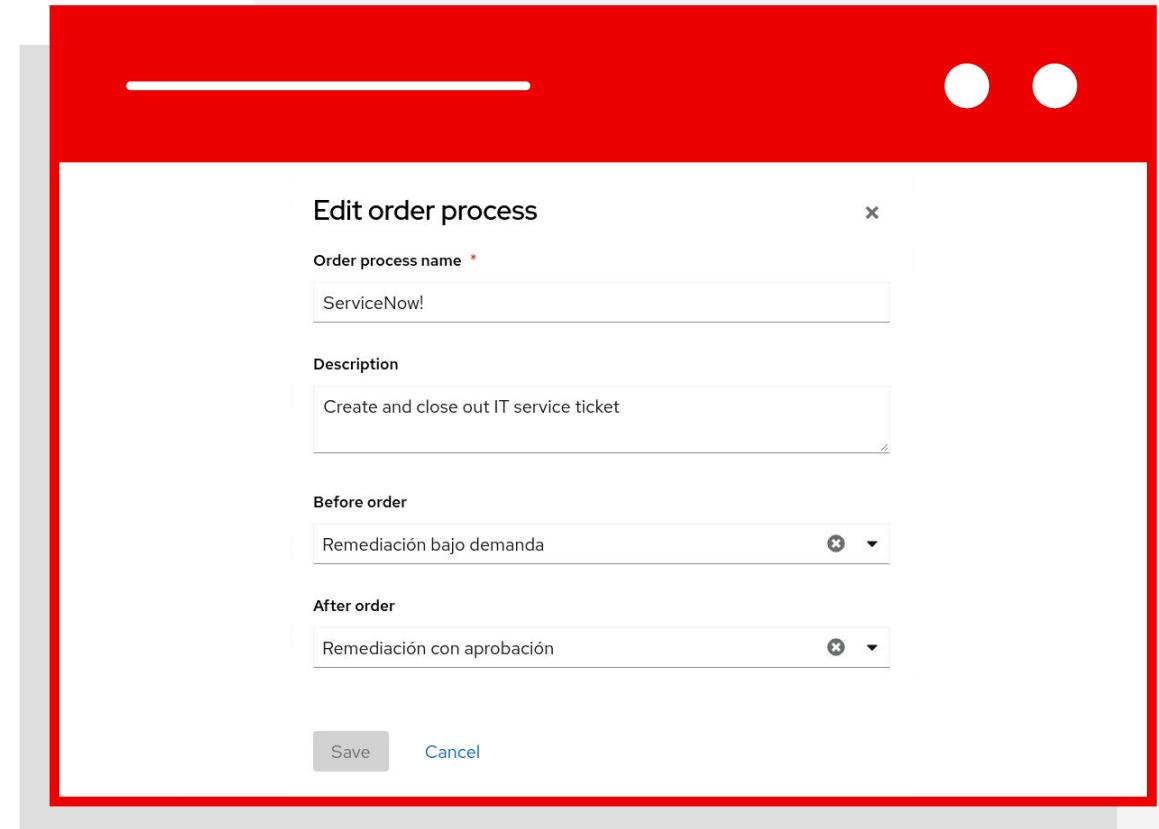
- ▶ Approval processes allow a human interaction to the automation, to add governance like human level administration.
- ▶ Both available at the operational level on the Automation controller UI as well as in the service catalog level.

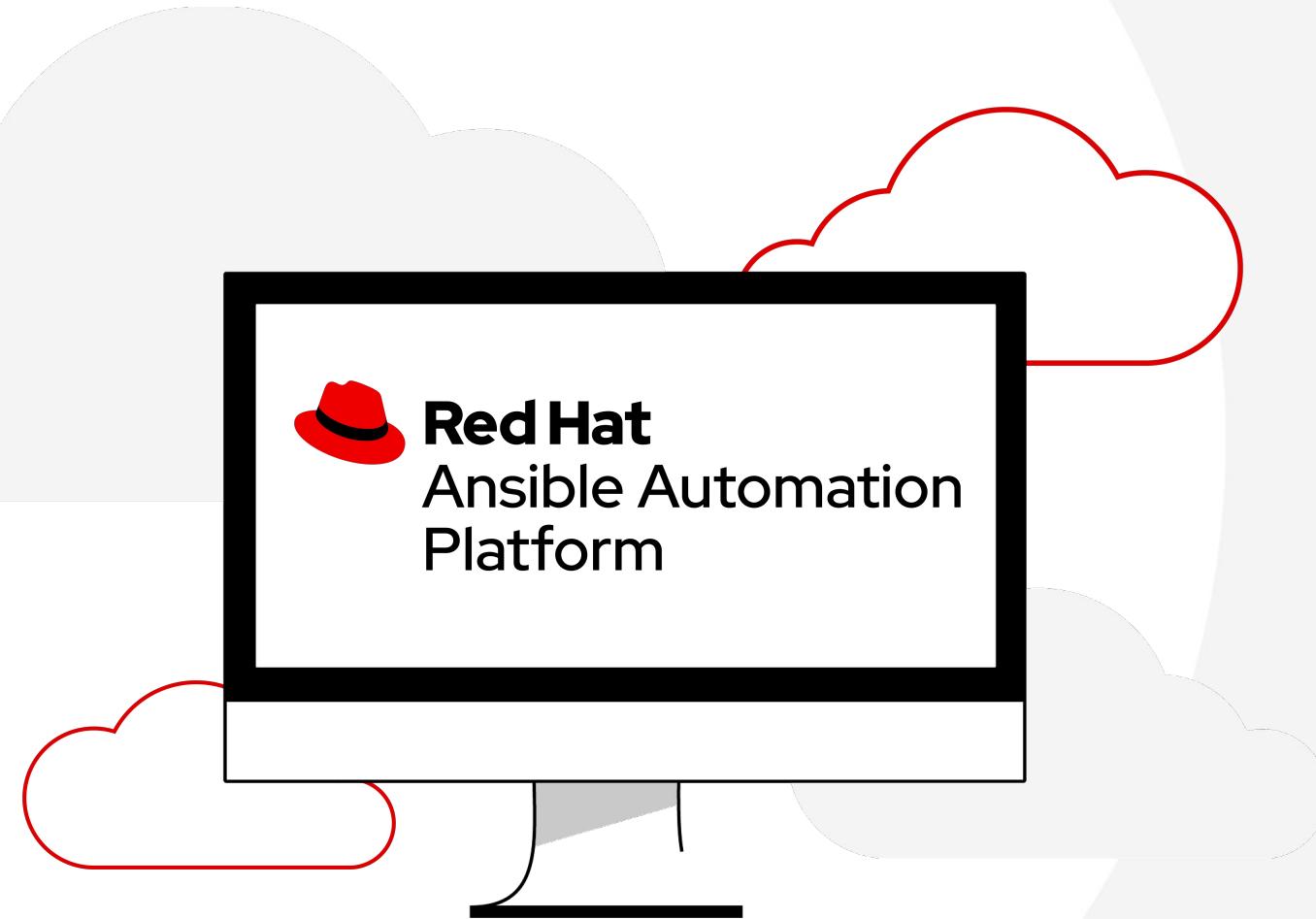


# IT service management integration (ITSM)

## Incorporate automation into your ITSM

- ▶ Integrate high level workflows in existing ITSM toolsets with the automation platform.
- ▶ Have the automation platform reach out to the ITSM system whenever things are changing, including data transmission between the tools.





# Where to go next

## Learn more

- ▶ [Workshops](#)
- ▶ [Documents](#)
- ▶ [Youtube](#)
- ▶ [Twitter](#)

## Get started

- ▶ [Evals](#)
- ▶ [cloud.redhat.com](#)

## Get serious

- ▶ [Red Hat Automation Adoption Journey](#)
- ▶ [Red Hat Training](#)
- ▶ [Red Hat Consulting](#)

# 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/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

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

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