
Red Hat
Ansible Automation
Platform
Ansible Linux Automation Workshop
Introduction to Ansible for Red Hat Enterprise Linux Automation
for System Administrators and Operators

What you will learn

- How Ansible Works
- Understanding Modules, Tasks, Playbooks
- Automation Controller Basics & Key Concepts
- Utilizing Projects & Job Template
- Role-based Access Control (RBAC)
- Self-service IT via surveys
- Overview of System Roles for Red Hat Enterprise Linux

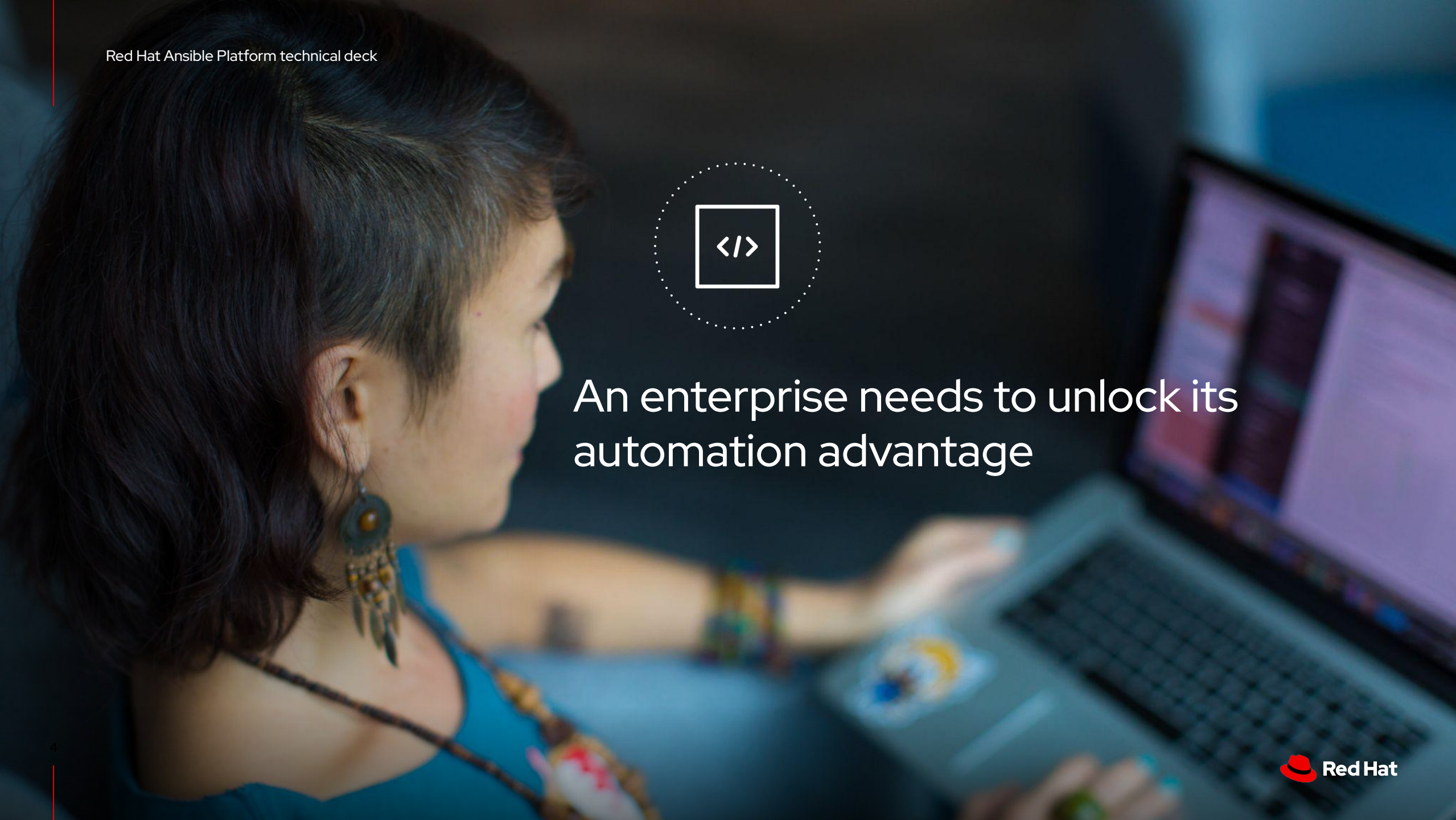
Red Hat
Ansible Automation
Platform

Introduction

Topics Covered:

- Why the Ansible Automation Platform?
- What can it do?

Red Hat



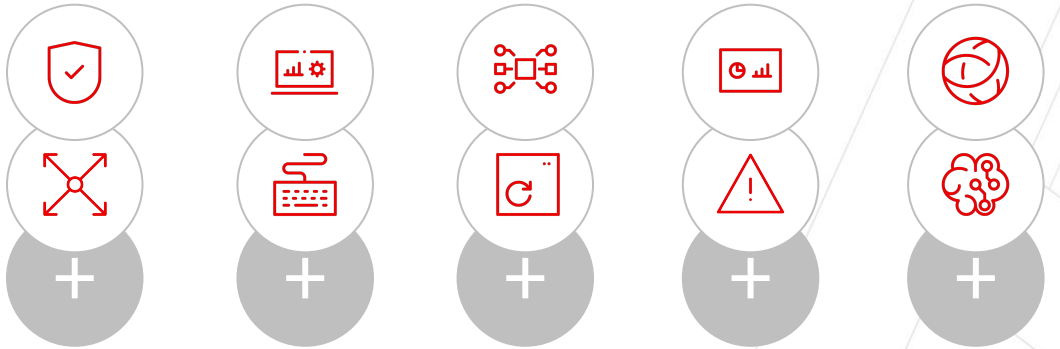
An enterprise needs to unlock its automation advantage

Red Hat

Many organizations share the same challenge.

Differing skills, roles, and responsibilities

Growing number of use cases

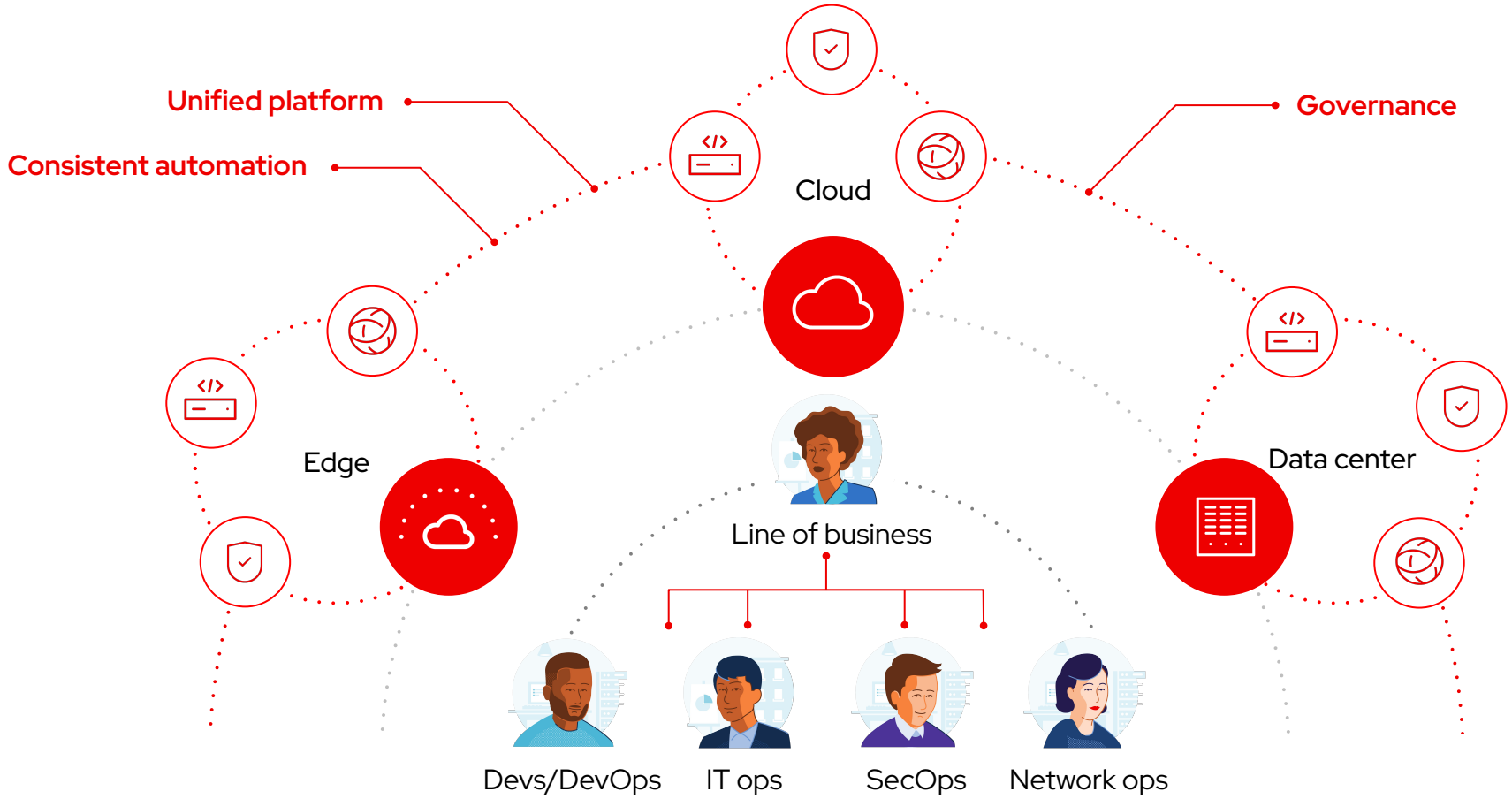


Multiple platforms and locations

Across multiple domains

Red Hat Ansible Automation Platform

The solution? Break down the silos.



Why Ansible Automation Platform?

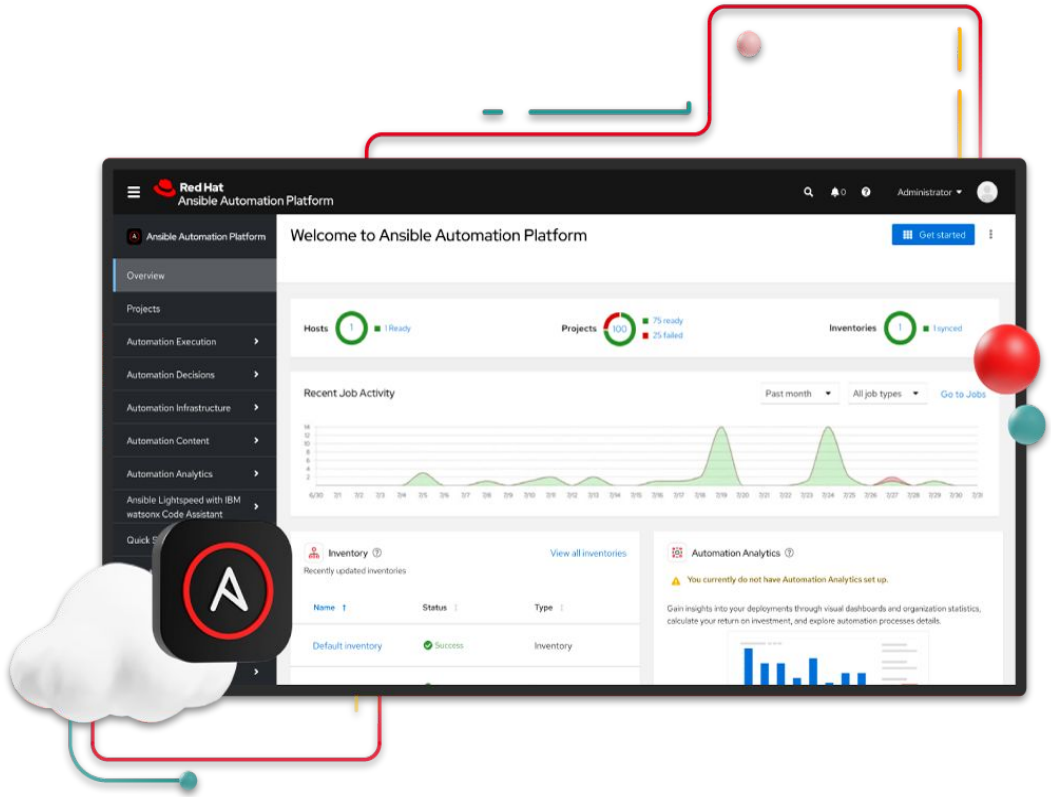
To unlock your automation advantage

Red Hat Ansible is becoming increasingly mission-critical for customers who rely on automation to bridge skills gaps, tame operational complexity across the enterprise, and mitigate costly tool sprawl.

Our goal with Red Hat Ansible Automation Platform 2.6 is to make it easier for every customer to fully unlock the potential of automation to transform IT - and deliver strategic advantages to the business.

This latest release is engineered to help our customers:

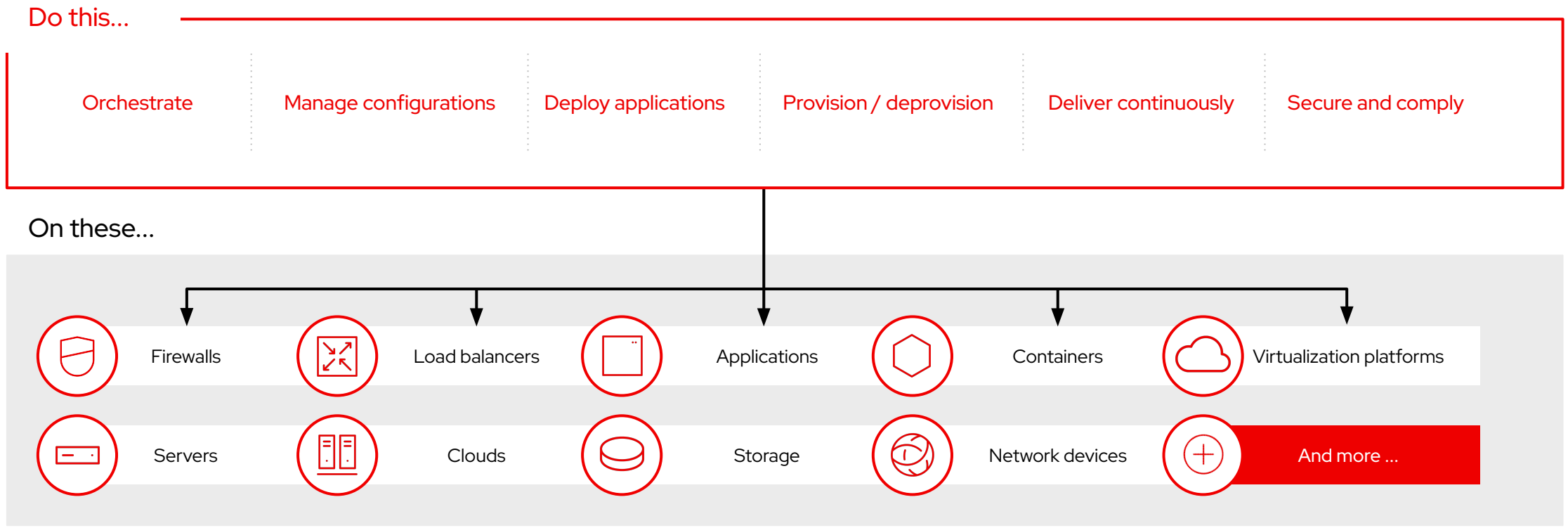
- **Accelerate automation adoption at scale,** with a reimagined automation platform experience, new features for enhanced usability, and tools for more effective collaboration and coordination.
- **Empower automation engineers,** with integrated developer tooling and generative AI capabilities designed to bolster ease and efficiency for a range of skill sets and experience across the entire automation creation lifecycle.
- **Orchestrate across the enterprise,** with event-driven automation capabilities that make intelligence from other tools more actionable, along with a robust ecosystem of integrations that put true end-to-end automation within reach.



Red Hat

Automate the deployment and management of automation

Your entire IT footprint



Red Hat

Supported and certified content you can trust.

170+

Certified and Validated Content Collections

70+

Certified technology partners

Infrastructure | Cloud | Network | Security | Edge

Red Hat Ansible Automation Platform

A platform for the entire automation team.

Architecture

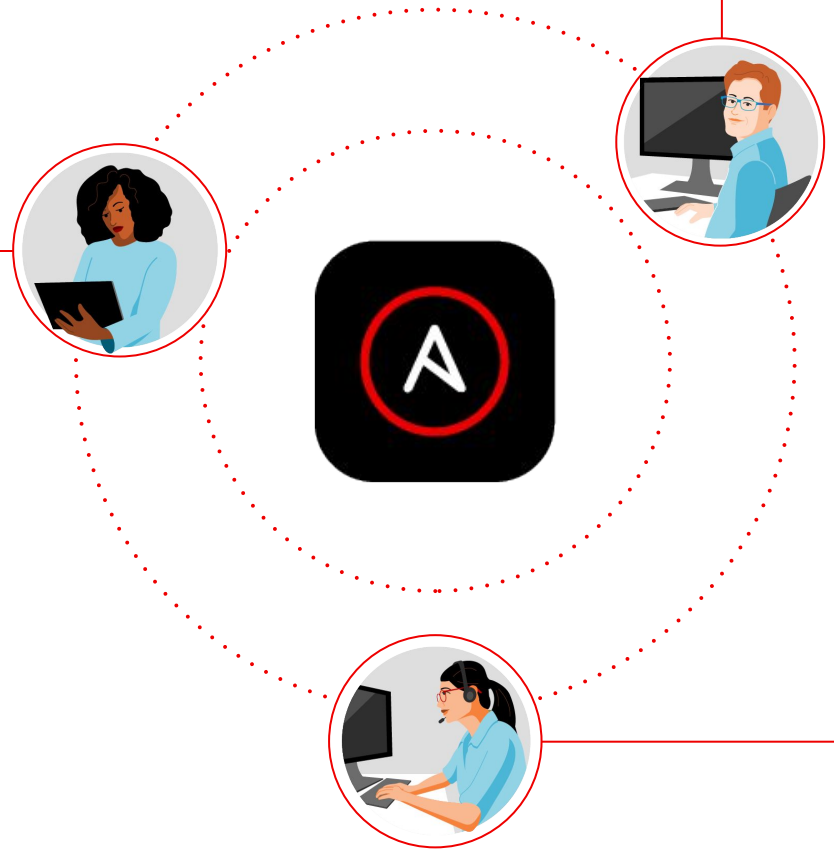
Flexible container-native architecture

Real-time analytics and reporting

Event-Driven Ansible

Containerized Installation

Available on all major cloud providers



Content creation

Ansible developer tools package

Integrated generative AI

Build automation communities

Large ecosystem of certified automation

Operations

Unified UI provides cohesive platform experience

Enterprise features: WebUI, API, role-based access control (RBAC), auditing and workflows for managing at scale

Ansible Automation Platform

by the numbers:

94% Reduction in recovery time following a security incident

84% Savings by deploying workloads to generic systems appliances using Ansible

67% Reduction in man hours required for customer deliveries

Financial summary:

146%

ROI on Ansible Automation Platform

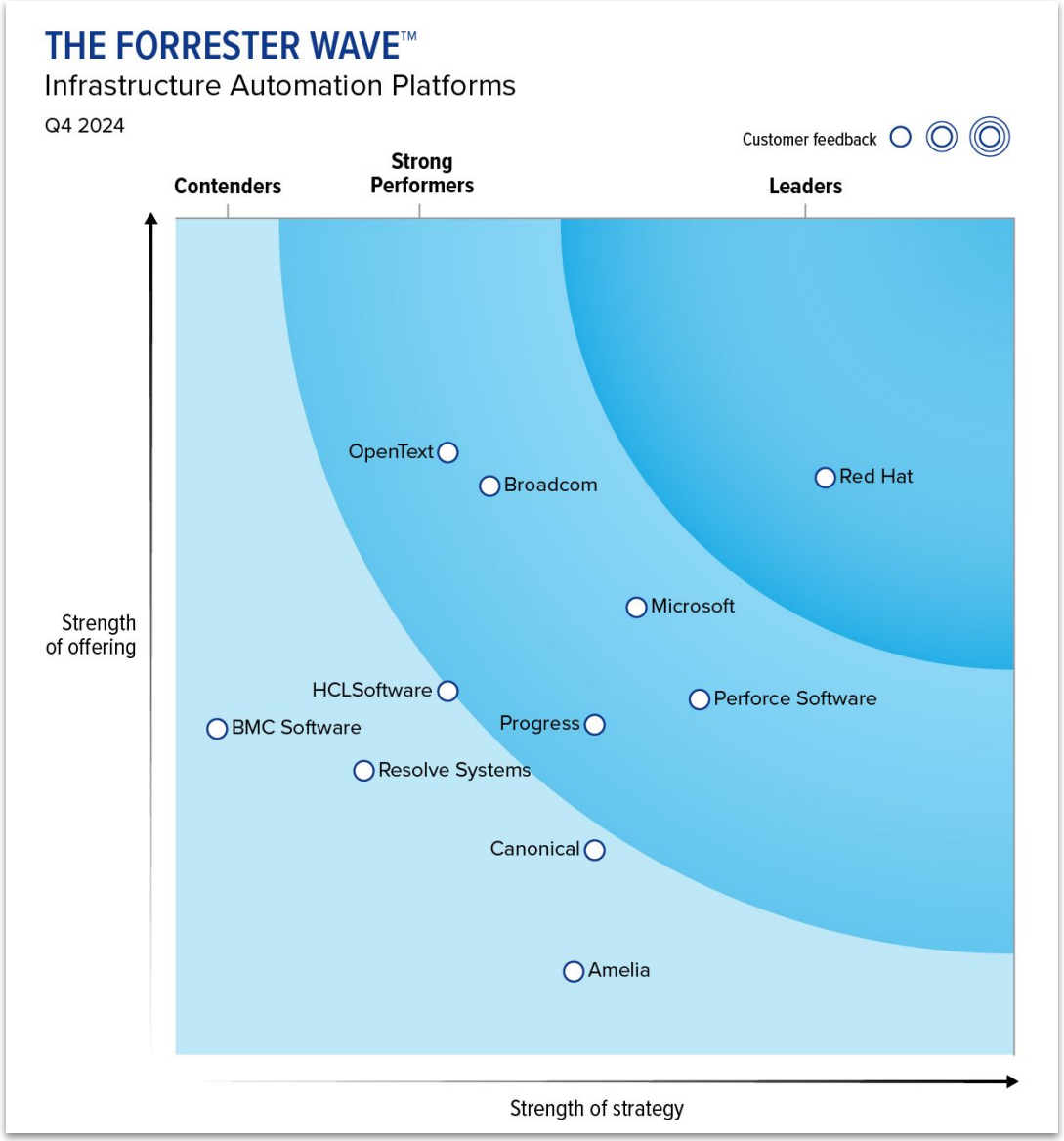
< 3 MONTHS

Payback on Ansible Automation Platform

SOURCE: *"The Total Economic Impact™ Of Red Hat Ansible Tower, a June 2018 commissioned study conducted by Forrester Consulting on behalf of Red Hat."*
redhat.com/en/engage/total-economic-impact-ansible-tower-20180710

Red Hat

Red Hat is *the leader* in the 2024 Forrester Wave™: Infrastructure Automation Platforms



Key takeaways from vendor profile

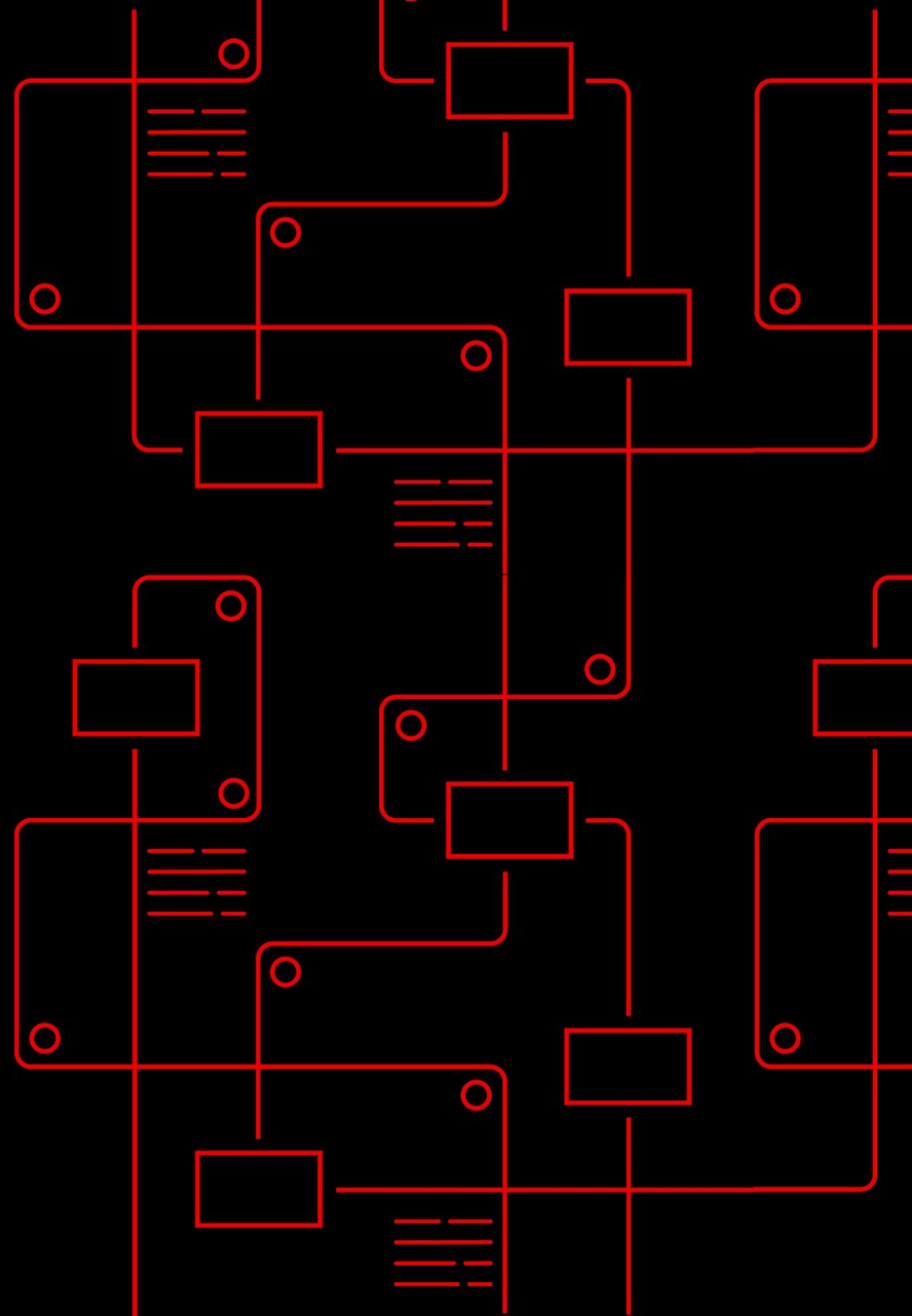
- "The Ansible Automation Platform is a comprehensive solution for various use cases, expanding into event-driven automation to reduce human intervention and reduce the burden of writing code for each scenario."
- "Red Hat's strategy for a collective and inclusive user community is spurring growth. Its vision and partner ecosystem are both strengths…supported by the community, Red Hat Ansible addresses numerous automation use cases."
- "Red Hat offers a good all-around automation tool for organizations that aim to enable users to develop mature automation and integrate technology silos."

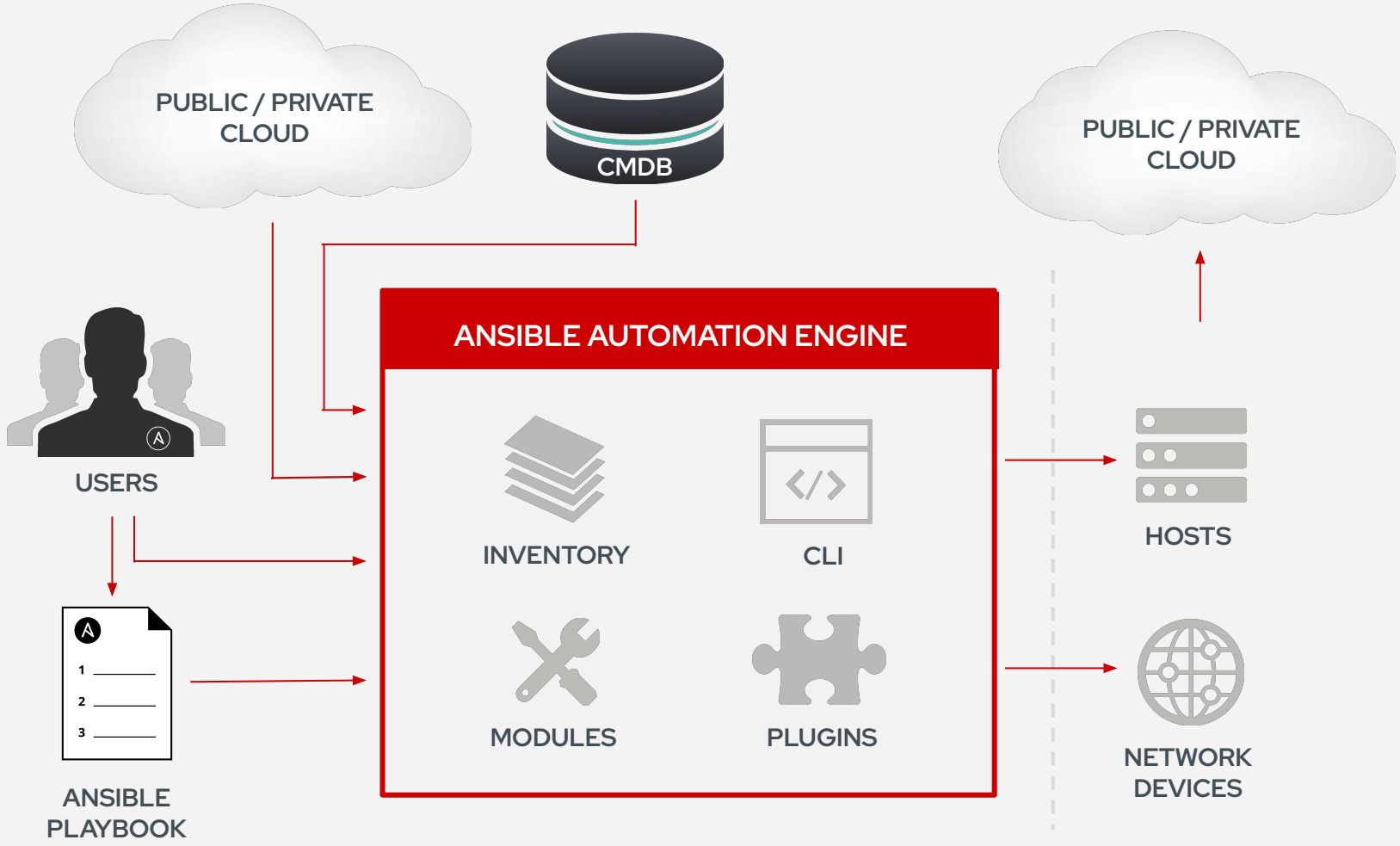
Source: Forrester Research, "The Forrester Wave™: Infrastructure Automation Platforms, Q4 2024," 2024.

Forrester does not endorse any company, product, brand, or service included in its research publications and does not advise any person to select the products or services of any company or brand based on the ratings included in such publications. Information is based on the best available resources. Opinions reflect judgment at the time and are subject to change. For more information, read about Forrester's objectivity here.

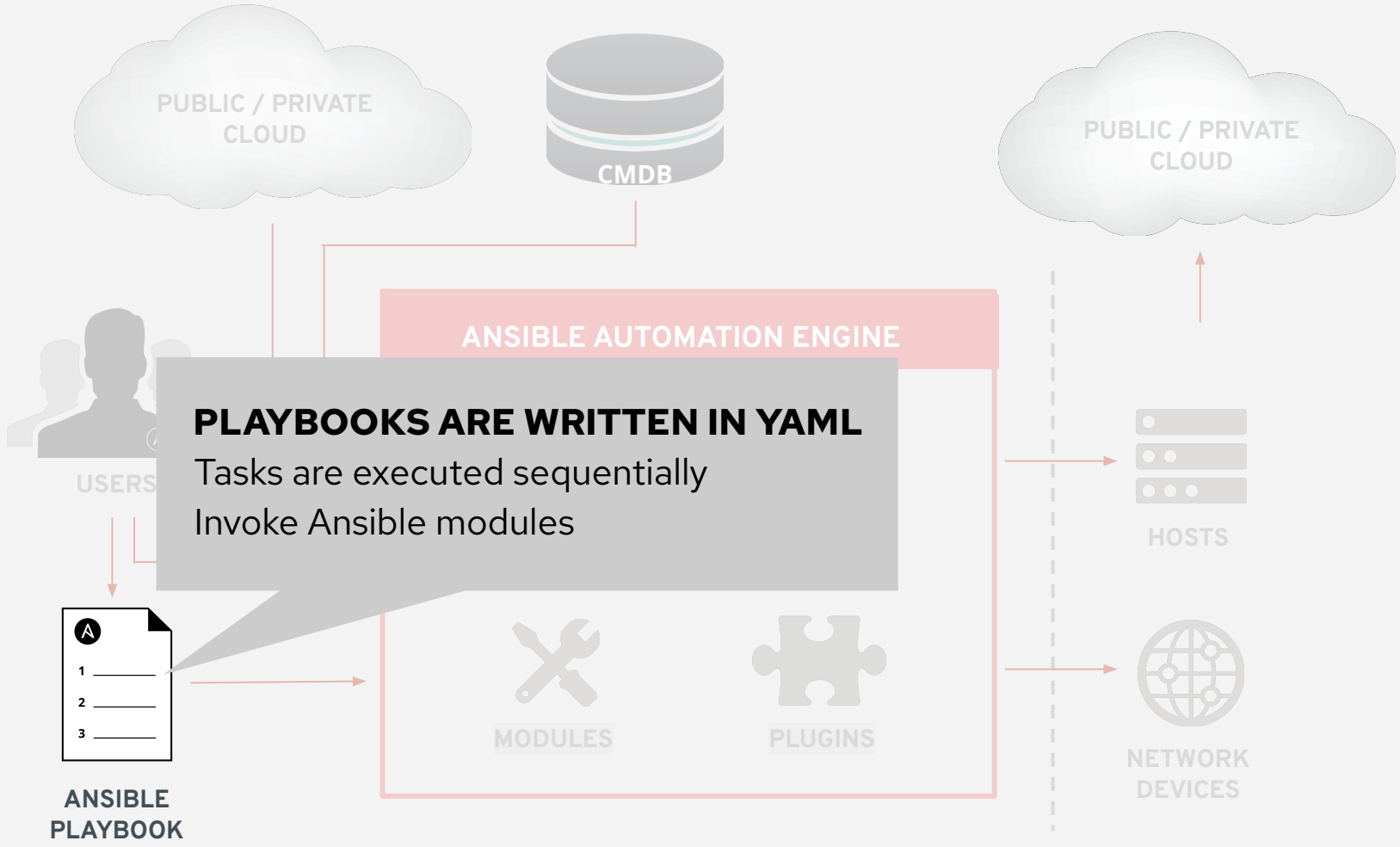
Red Hat

The Ansible Basics




PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE CLOUD
ANSIBLE AUTOMATION ENGINE
USERS
INVENTORY
CLI
HOSTS
MODULES
PLUGINS
NETWORK DEVICES
ANSIBLE PLAYBOOK

Red Hat


PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE CLOUD
ANSIBLE AUTOMATION ENGINE
USERS
PLAYBOOKS ARE WRITTEN IN YAML
Tasks are executed sequentially
Invoke Ansible modules
HOSTS
1
2
3
MODULES
PLUGINS
NETWORK DEVICES
ANSIBLE PLAYBOOK

Red Hat

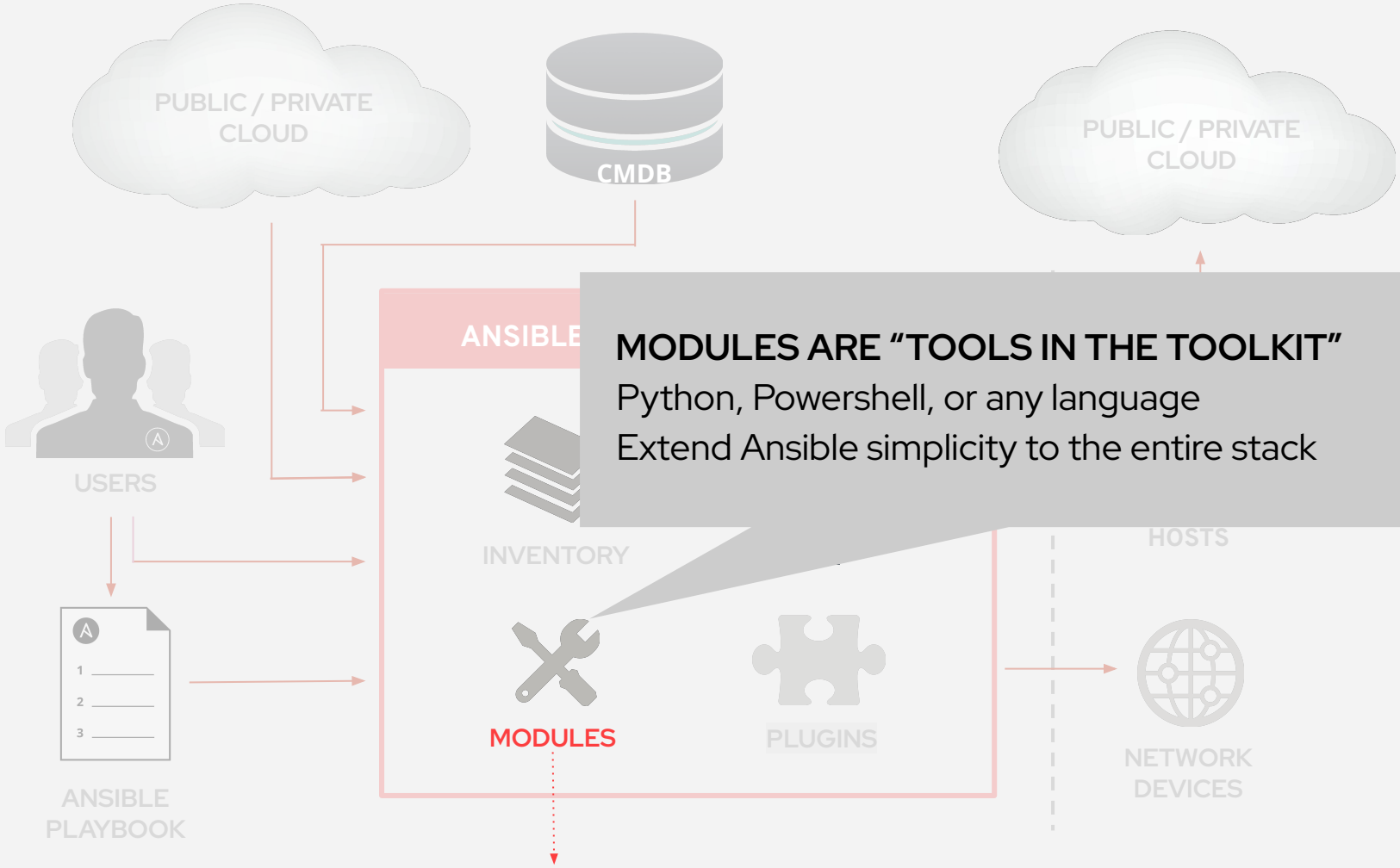
Ansible playbooks

```
---
- name: Install and start apache
  hosts: web
  become: true

  tasks:
    - name: Ensure the httpd package is installed
       ansible.builtin.package:
         name: httpd
         state: present

    - name: Create the index.html file
      ansible.builtin.template:
        src: files/index.html
        dest: /var/www/html/

    - name: Start the httpd service if needed
      ansible.builtin.service:
        name: httpd
        state: started
```



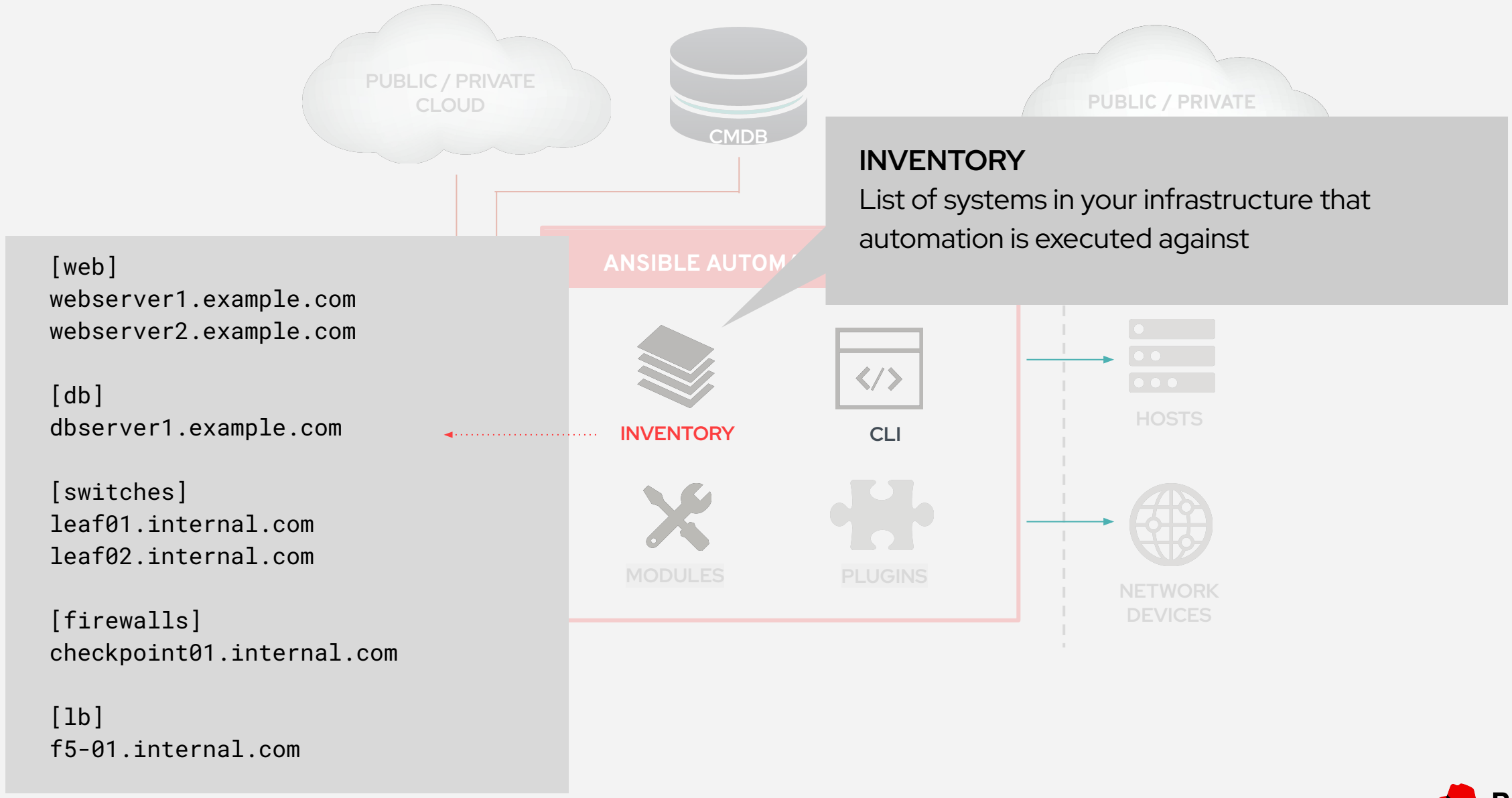
MODULES ARE "TOOLS IN THE TOOLKIT"

Python, Powershell, or any language

Extend Ansible simplicity to the entire stack

```
- name: Create the index.html file
ansible.buitlin.template:
    src: files/index.html
    dest: /var/www/html/
```

Red Hat


PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE
INVENTORY
List of systems in your infrastructure that automation is executed against
[web]
webserver1.example.com
webserver2.example.com
[db]
dbserver1.example.com
[switches]
leaf01.internal.com
leaf02.internal.com
[firewalls]
checkpoint01.internal.com
[lb]
f5-01.internal.com
ANSIBLE AUTOM
INVENTORY
CLI
MODULES
PLUGINS
HOSTS
NETWORK DEVICES

Red Hat

LINUX AUTOMATION

150+

Linux Modules

AUTOMATE EVERYTHING LINUX

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

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

ansible.com/get-started

Red Hat

Exercise 1

Topics Covered:

- Ansible Inventories
- Ansible Playbooks Basics
- Running an Ansible Playbook

Ansible Inventory. The systems that a playbook runs against.

What are they?

- List of systems in your infrastructure that automation is executed against

How do they work?

- Defines the systems that Ansible manages and targets for automation.
- Organizes hosts into groups (e.g., web servers, databases) for better management.
- Group variables apply settings across multiple systems efficiently.
- Host-specific variables allow detailed customization for individual systems.

```
[web]
webserver1.example.com
webserver2.example.com

[db]
dbserver1.example.com

[switches]
leaf01.internal.com
leaf02.internal.com
```

Ansible Inventory. The systems that a playbook runs against.

The Basics

- An example of a static Ansible inventory including systems with IP addresses as well as fully qualified domain name (FQDN)

```
[myservers]
10.42.0.2
10.42.0.6
10.42.0.7
10.42.0.8
10.42.0.100
host.example.com
```

Ansible Inventory

```
[app1srv]
appserver01 ansible_host=10.42.0.2
appserver02 ansible_host=10.42.0.3

[web]
node-[1:30]

[web:vars]
apache_listen_port=8080
apache_root_path=/var/www/mywebdocs/

[all:vars]
ansible_user=kev
ansible_ssh_private_key_file=/home/kev/.ssh/id_rsa
```

Ansible playbook

A play

```
---
- name: Install and start apache
  hosts: web
  become: true

  tasks:
    - name: Ensure the httpd package is installed
      ansible.builtin.package:
        name: httpd
        state: present

    - name: Create the index.html file
      ansible.builtin.template:
        src: files/index.html
        dest: /var/www/html/

    - name: Start the httpd service if needed
      ansible.builtin.service:
        name: httpd
        state: started
```

Ansible playbook

A task

```
---
- name: Install and start apache
  hosts: web
  become: true

  tasks:
    - name: Ensure the httpd package is installed
      ansible.builtin.package:
        name: httpd
        state: present

    - name: Create the index.html file
      ansible.builtin.template:
        src: files/index.html
        dest: /var/www/html/

    - name: Start the httpd service if needed
      ansible.builtin.service:
        name: httpd
        state: started
```

Ansible playbook

A module 

```
---
- name: Install and start apache
  hosts: web
  become: true

  tasks:
    - name: Ensure the httpd package is installed
      ansible.builtin.package:
        name: httpd
        state: present

    - name: Create the index.html file
      ansible.builtin.template:
        src: files/index.html
        dest: /var/www/html/

    - name: Start the httpd service if needed
      ansible.builtin.service:
        name: httpd
        state: started
```

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
```

Lab Time

Complete Exercise 1 - Writing Your First Playbook

Exercise 2

Topics Covered:

- Working with Ansible Variables
- Working with Ansible Facts

Red Hat

Ansible Variable Playbook

```
---
- 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
      ansible.builtin.debug:
        msg: "{{ var_three }}"
```

Ansible Variable Playbook

```
---
- 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
      ansible.builtin.debug:
        msg: "{{ var_three }}"
```

ansible is awesome

Ansible Facts

Facts

- Just like variables, really...
- ... but: coming from the host itself!
- Check them out with the setup module

```
tasks:
  - name: Collect all facts of host
    ansible.builtin.setup:
      gather_subset:
        - 'all'
```

Ansible Facts Playbook

```
---
- name: Ansible Facts playbook
  hosts: localhost
  gather_facts: true

  tasks:
    - name: Define custom fact
      ansible.builtin.set_fact:
        server_role: “frontend”

    - name: Use custom and built-in fact
      ansible.builtin.debug:
        msg: ‘The {{ ansible_hostname }} is assigned the role: {{ server_role }}’
```

```
$ ansible-playbook facts_playbook.yml
```

Ansible Inventory - Managing Variables In Files

```
$ tree ansible-files/

├── deploy_index_html.yml
├── files
│   ├── dev_web.html
│   └── prod_web.html
├── group_vars
│   └── web.yml
└── host_vars
    └── node2.yml
```

Ansible Inventory - Managing Variables In Files

```
├──
deploy_index_html.yml
├── files
│   ├── dev_web.html
│   └── prod_web.html
├── group_vars
│   └── web.yml
└── host_vars
    └── node2.yml
```

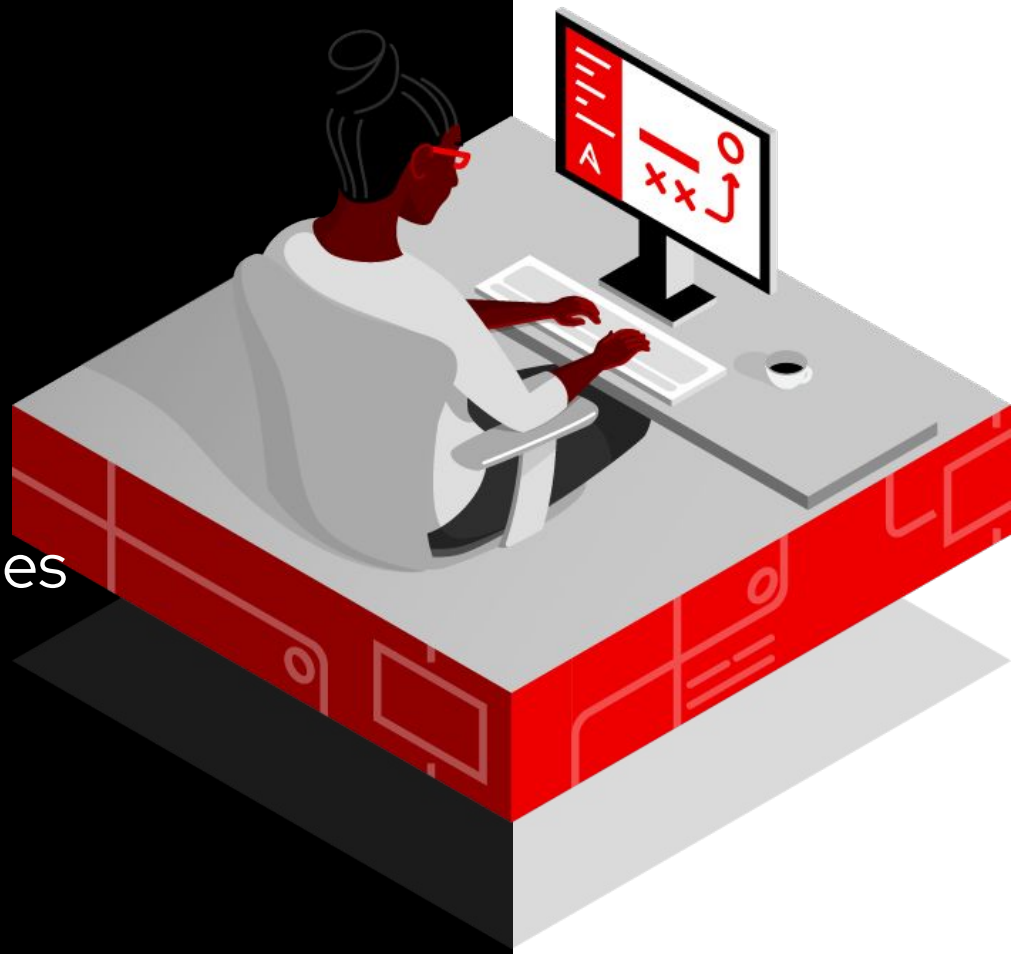
```
$ cat group_vars/web.yml
---
stage: dev
```

```
$ cat host_vars/node2.yml
---
stage: prod
```

```
- name: copy web.html
  copy:
    src: "{{ stage }}_web.html"
    dest: /var/www/html/index.html
```

Lab Time

Complete Exercise 2 - Using Variables



Exercise 3

Topics Covered:

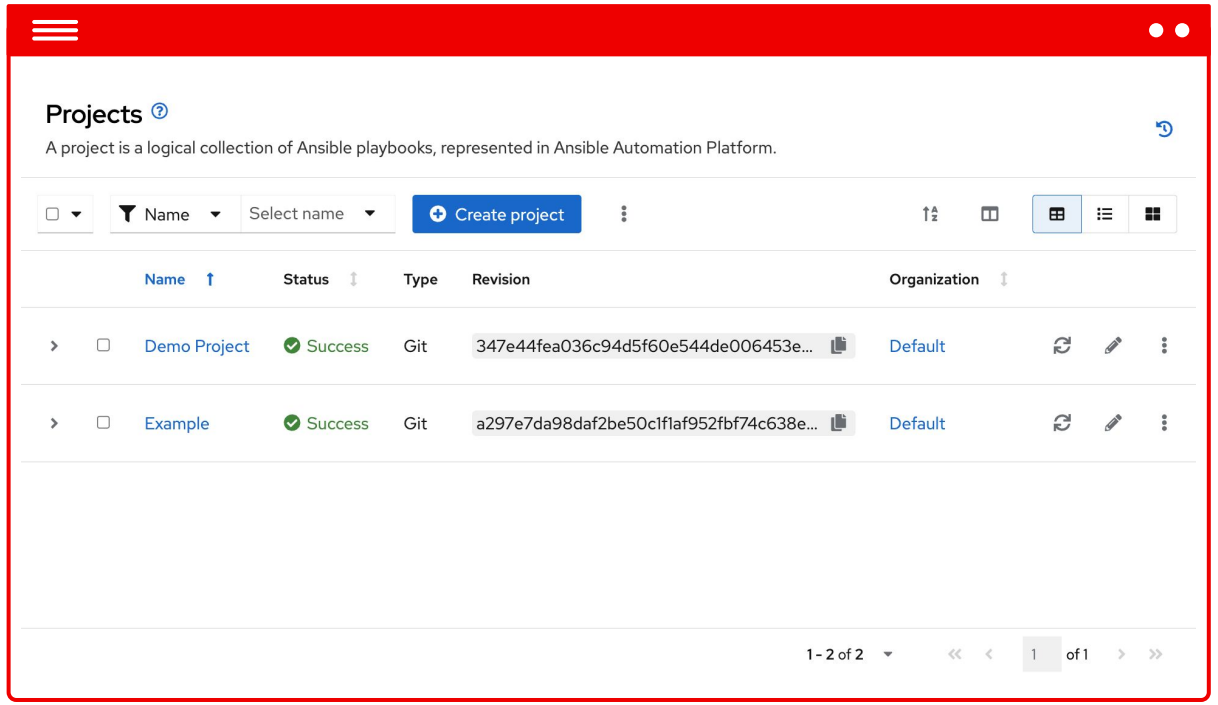
- Projects
- Job Templates

Projects. Adding your automation content to controller.

What is it?

Logical collection of your playbooks:

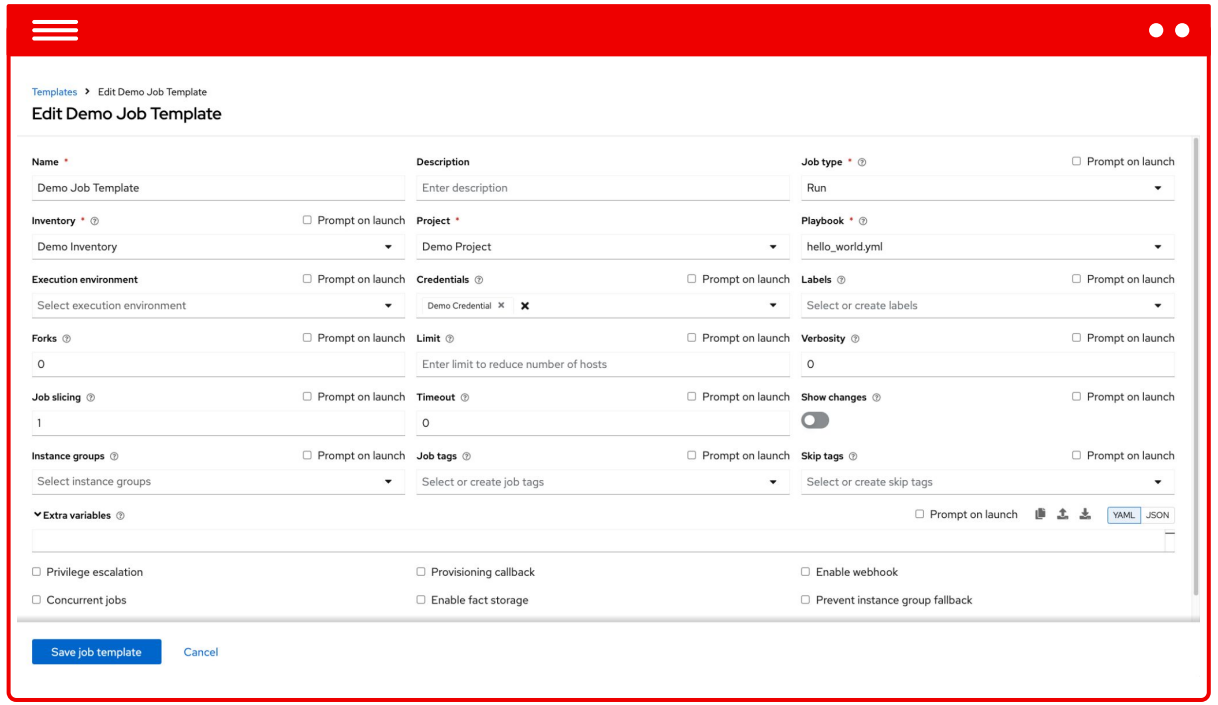
- Multiple source types supported
- Source Control Management (SCM) integration and update strategies
- Role-based access control (RBAC) and schedules



Job Templates. Bringing it all together.

What is it?

- Define and standardize running automation
- Reusable and shareable
- Leverage agile practices, such as GitOps and event-driven automation

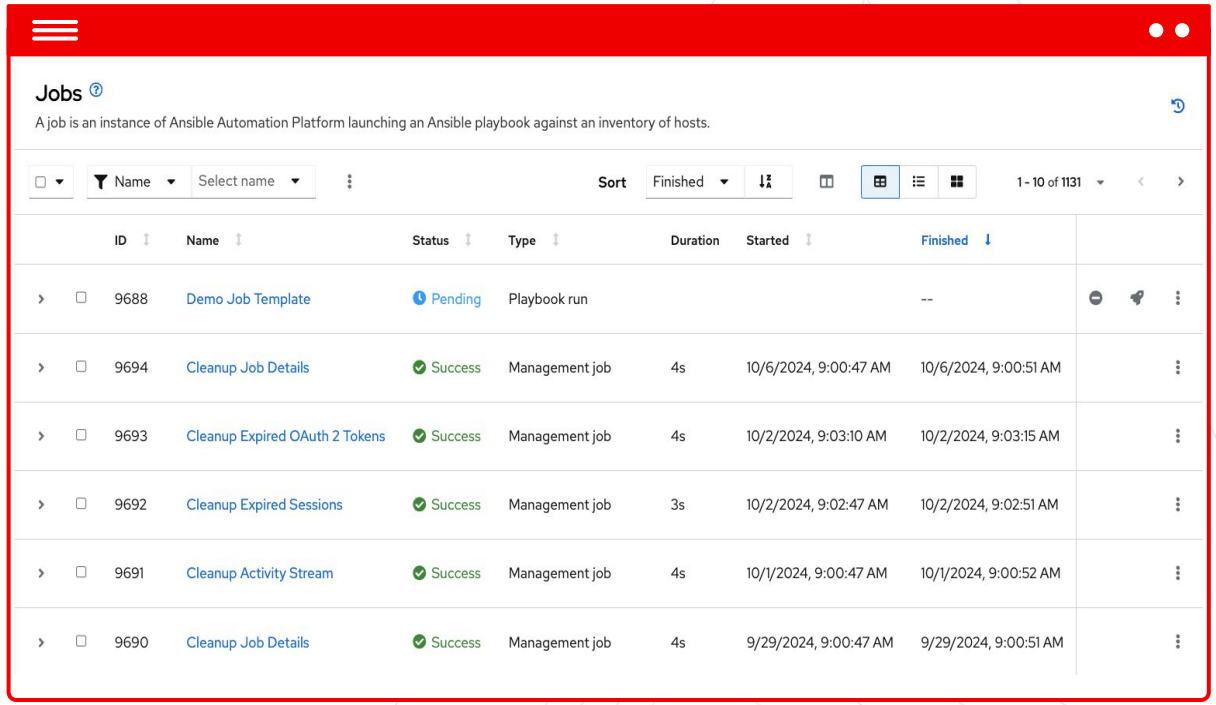


Red Hat
Ansible Automation
Platform

Automation jobs. Executing your defined automation.

What is it?

- Controller launching an instance of defined automation
- Relaunch automation jobs
- Use Job Details to view job outputs
- Troubleshoot automation execution using filtered views



Red Hat
Ansible Automation
Platform

Lab Time

Complete Exercise 3 - Projects & Job Templates

Exercise 4

Topics Covered:

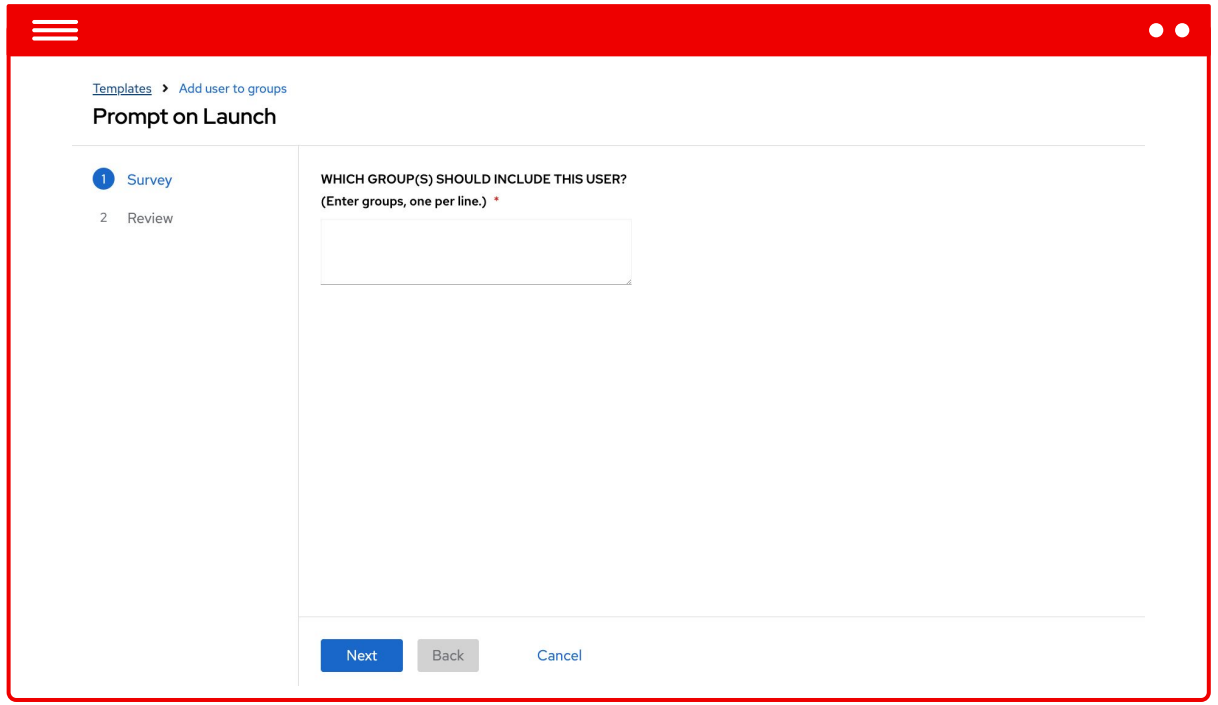
- Ansible Surveys

Red Hat

Automation controller surveys. Adopt and grow.

What is it?

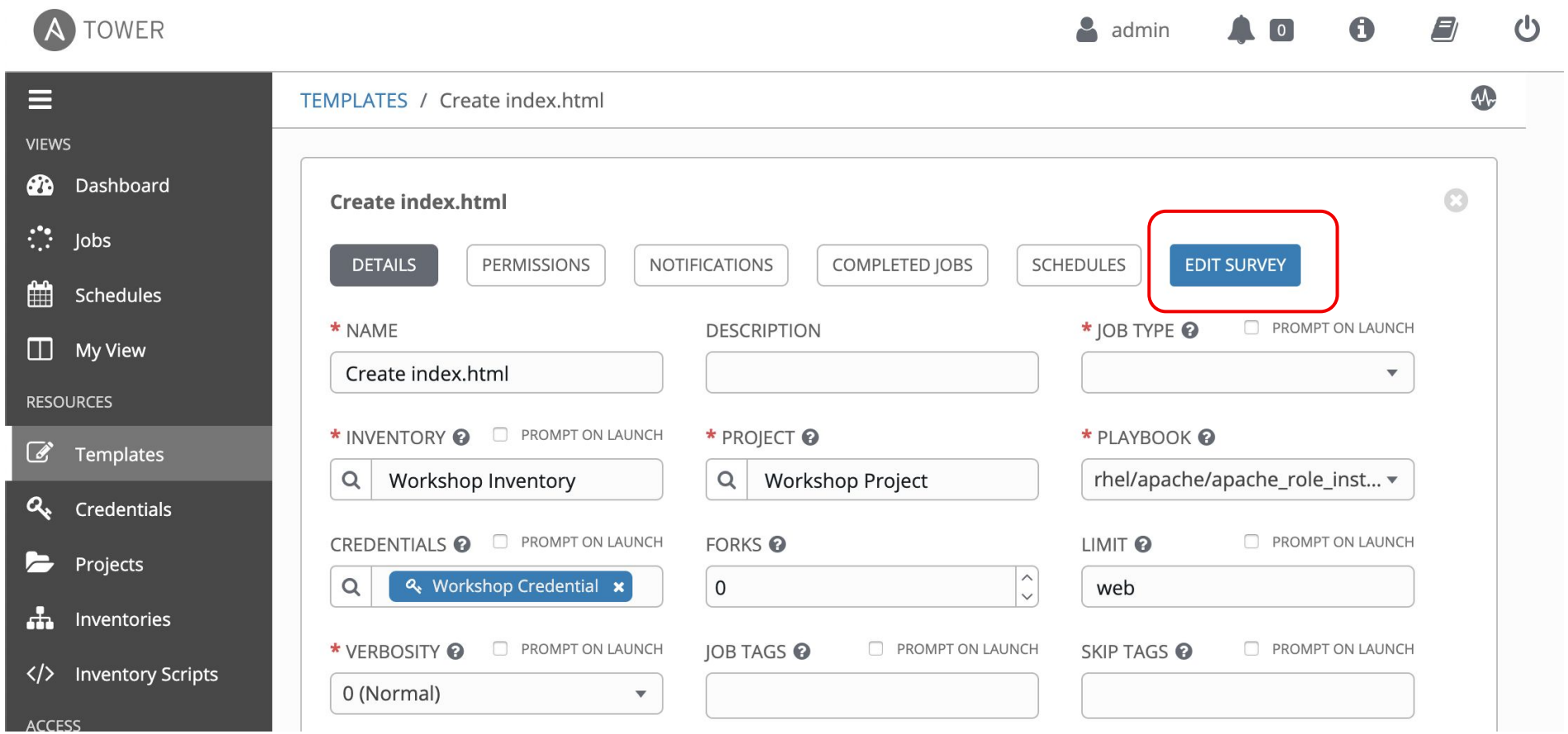
- User-friendly, self-service interface in automation controller
- Abstracts complexity using question and answer format
- Best suited for teams directly accessing automation and close to the automation practice
- Access and execution governed using controller features



Creating a Survey (1/2)

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

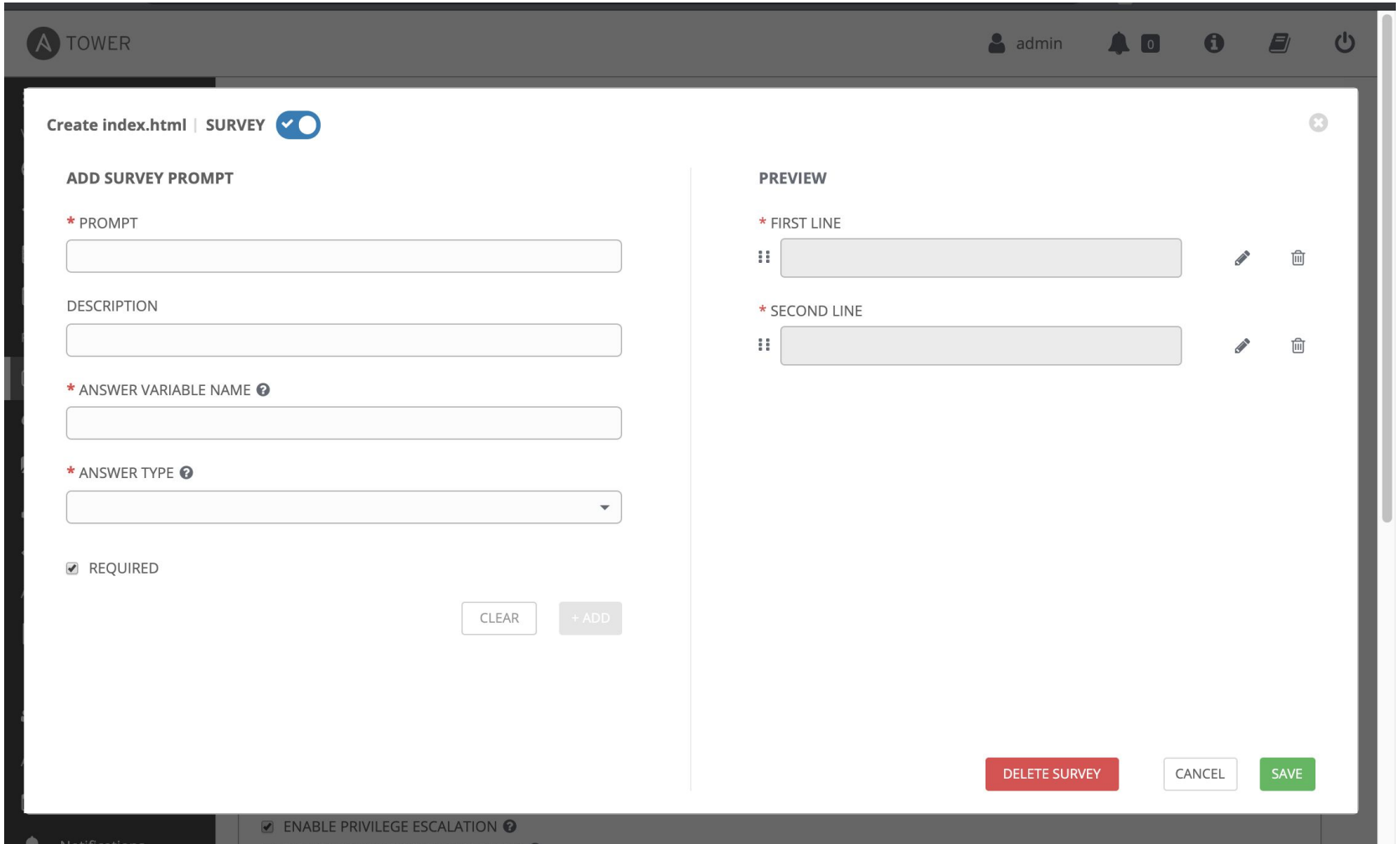
Click the button to open the Add Survey window.



Red Hat

Creating a Survey (2/2)

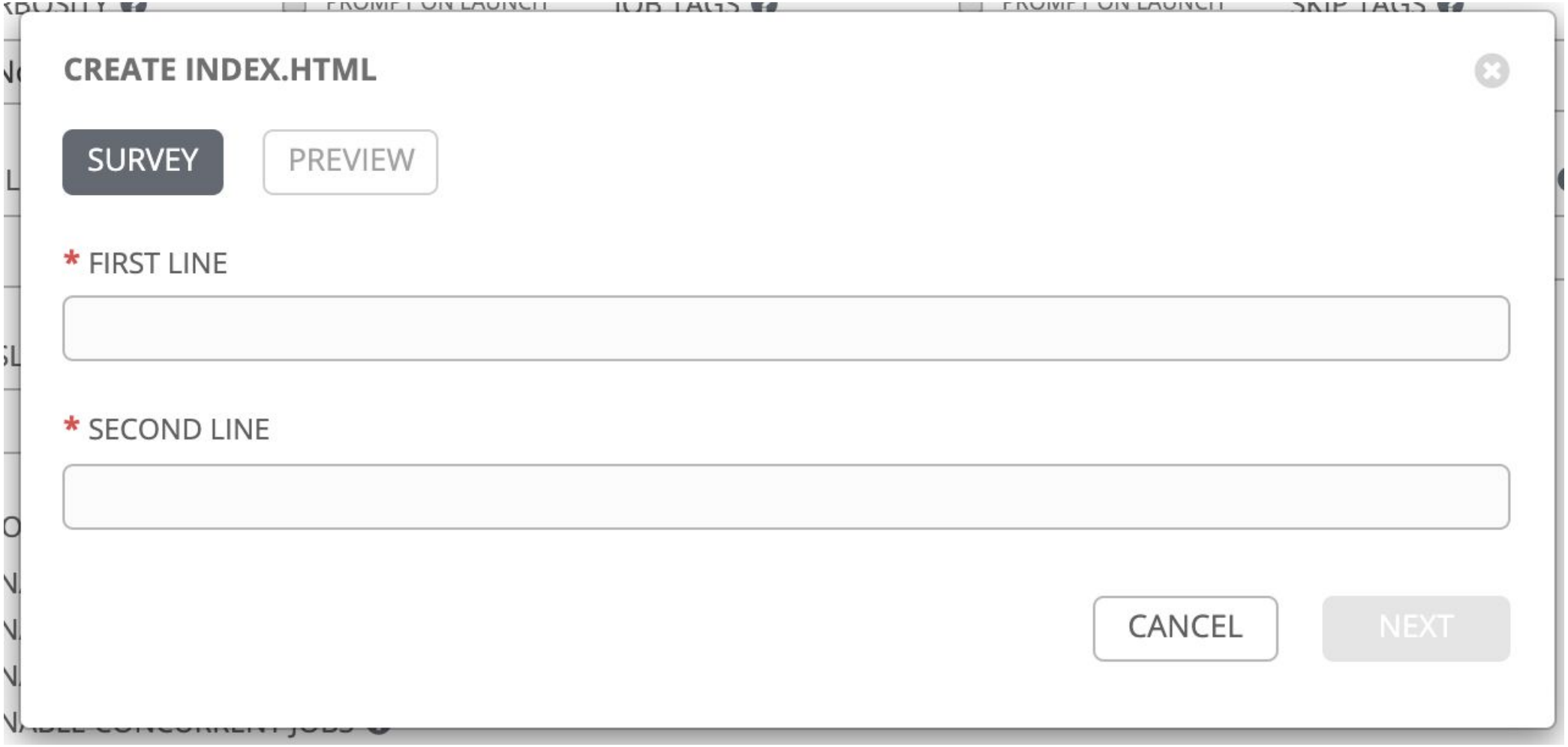
The Add Survey window allows the Job Template to prompt users for one or more questions. The answers provided become variables for use in the Ansible Playbook.



Red Hat

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.



Red Hat

Lab Time

Complete Exercise 4 - Surveys

Exercise 5

Topics Covered:

- Role Based Access Control (RBAC)

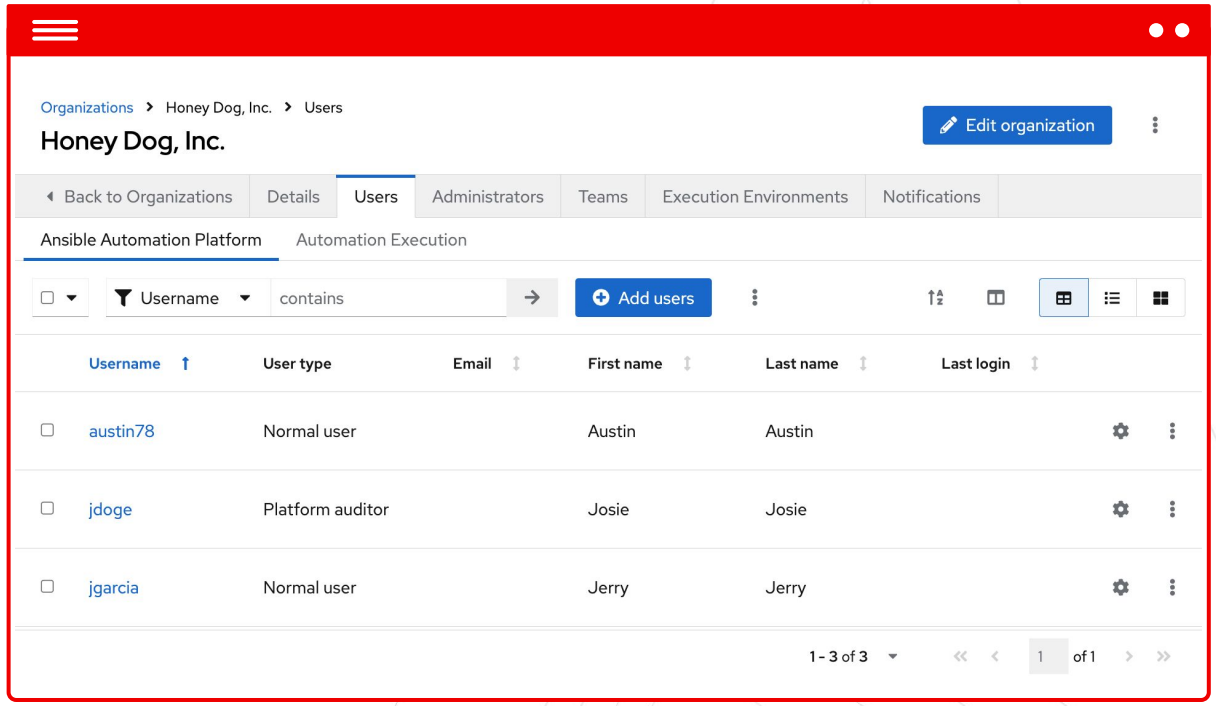
Red Hat

Role-Based Access Control. Who can use my automation?

What is it?

Securely govern access to your automation

- Logically group controller objects and grant users and teams read, execute, edit permissions
- Use predefined roles to grant access
- Integrates with your existing enterprise authentication systems



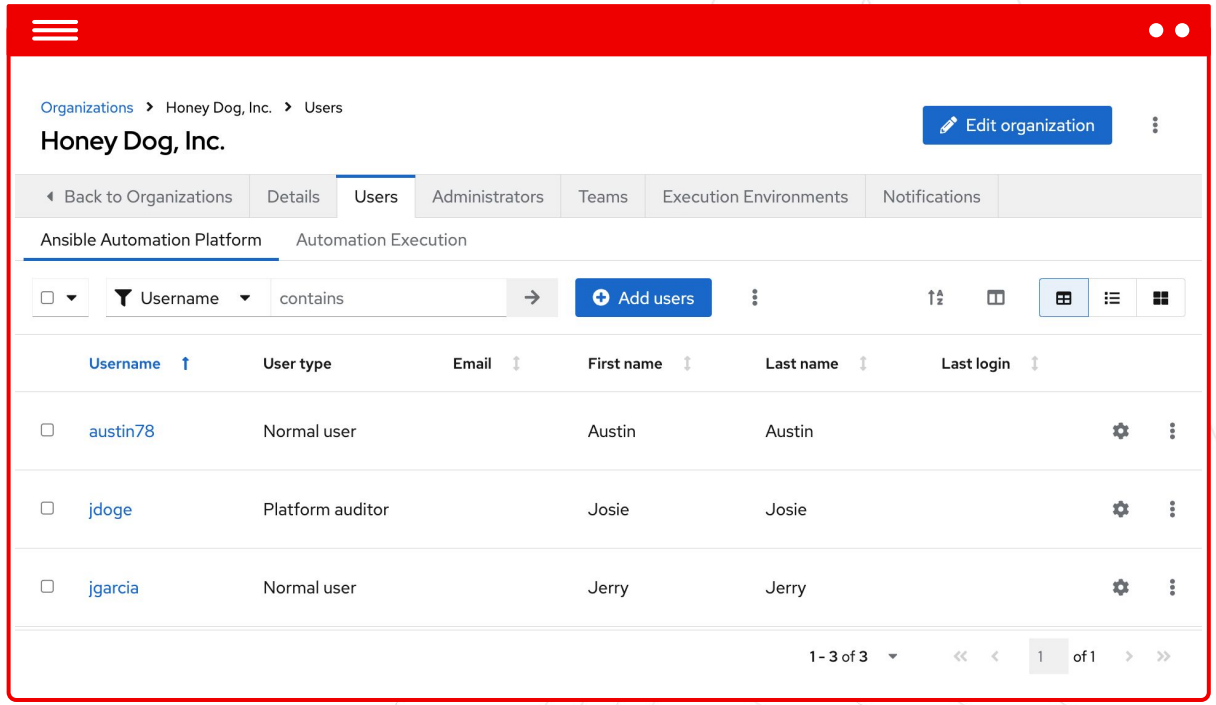
Red Hat
Ansible Automation
Platform

User Management

What is it?

Securely govern access to your automation

- Logically group controller objects and grant users and teams read, execute, edit permissions
- Use predefined roles to grant access
- Integrates with your existing enterprise authentication systems

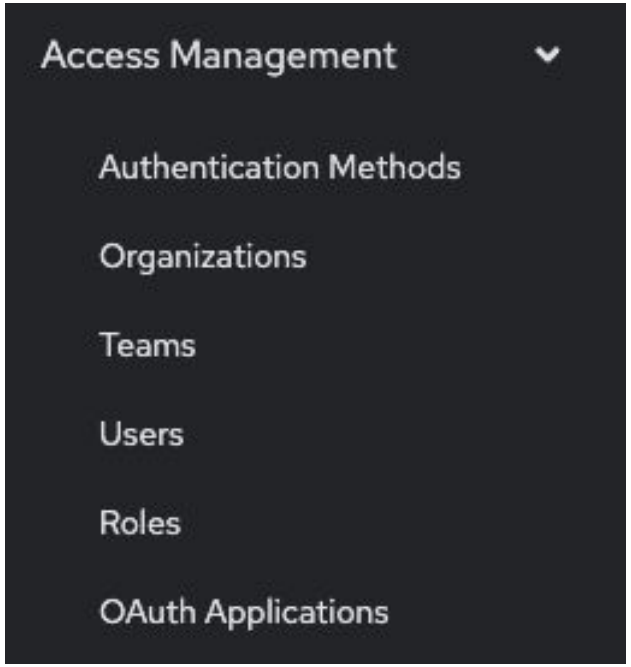


Red Hat
Ansible Automation
Platform

User Management

Govern access to your automation

- An **organization** is a logical collection of users, teams, projects, inventories and more. All entities belong to an organization.
- A **user** is an account to access Ansible Automation Controller 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.



Red Hat

Lab Time

Complete Exercise 5 - Role Based Access Control

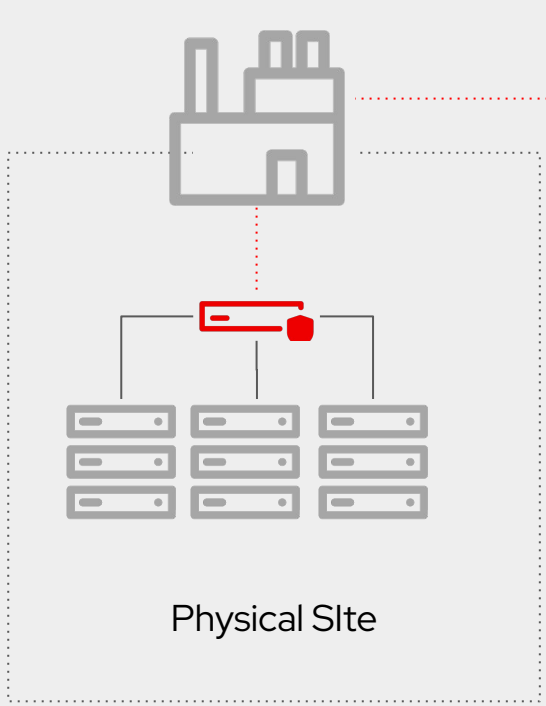
Exercise 6

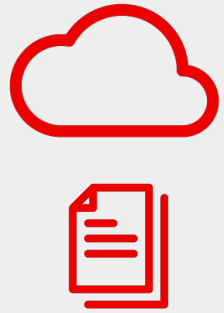
Topics Covered:

- Red Hat Enterprise Linux System Roles

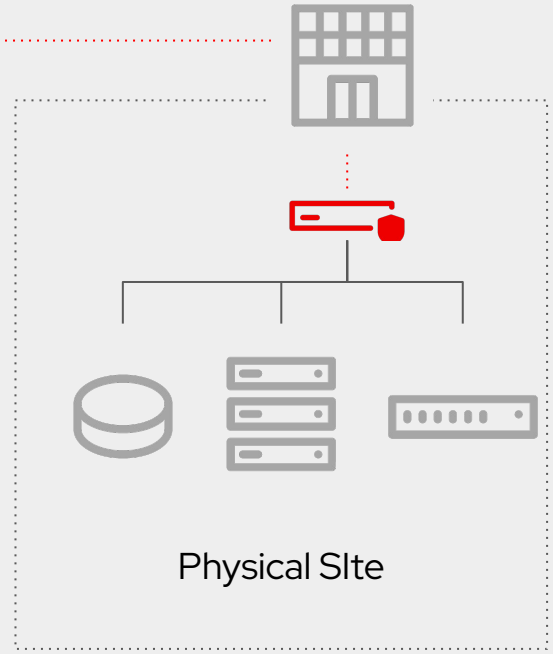
Red Hat

Automation Hub and Ansible Galaxy





Ansible Content
Roles & Collections

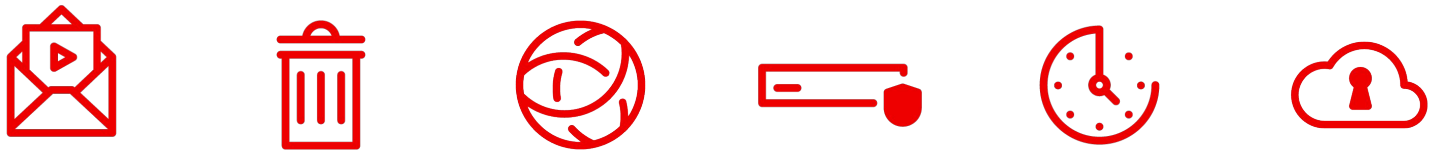


Red Hat

Linux System Roles

- Consistent user interface to provide settings to a given subsystem that is abstract from any particular implementation

Examples



Red Hat

An Ansible Playbook Variable Example

```
---
- name: example system roles playbook
  hosts: web

  tasks:

    - name: Configure Firewall
      include_role:
        name: linux-system-roles.firewall

    - name: Configure Timesync
      include_role:
        name: linux-system-roles.timesync
```

Lab Time

Complete Exercise 6 - RHEL System Roles

Exercise 7

Topics Covered:

- Wrap-Up

Red Hat

Lab Time

Complete Exercise 7 - Wrap-Up

Where to go next

Learn more

- Workshops
- Documents
- Youtube
- X

Get started

- Self-paced labs
- Ansible Automation Platform trial
- console.redhat.com
- Ansible Lightspeed trial

Get serious

- Red Hat Automation Adoption Journey
- Red Hat Training
- Red Hat Consulting

Thank you


linkedin.com/company/red-hat


youtube.com/AnsibleAutomation


facebook.com/ansibleautomation


twitter.com/ansible


github.com/ansible

Red Hat