



Red Hat Ansible Automation Platform

Ansible Linux Automation Workshop

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



Red Hat

Agenda

13:30 - 14:00	30 min	Welcome & Ansible Automation overview
14:00 - 14:10	10 mins	Lab Setup & introduction the lab controls
14:10 - 14:55	45 mins	Hands-on 1+2 Overview of public cloud provisioning + The Ansible Basics
15:00 - 15:45	45 mins	Hands-on 3+4 Deploying applications to linux hosts Retrieving information from hosts
15:45 - 16:25	40 mins	Hands-on 5+6 Self-service IT via surveys Overview of system roles for RHEL
16:25 - 16:30	5 mins	Wrap-up: hands-on workshop review

Ansible Workshop Link

- Ansible Workshop Link: <https://red.ht/ansiblelab>
- Ansible Lab Link :
 - <https://rh5dcd.example.opentlc.com/>
 - <https://rh8628.example.opentlc.com/>

Workshop Name: rh5dcd

This is the workshop launch page. If you require more information please refer to this [YouTube video](#).

[Ansible RHEL Exercises](#)

[Ansible Technical Deck](#)

Workbench Information

Please enter your Name and Email address.

Name:	<input type="text"/>
Email:	<input type="text"/>
<input type="button" value="Submit"/>	

Red Hat Customer Success



Technical Account Manager

- Yearly Paid Subscription
- Named technical contact
- Personalized, proactive support and guidance
- Champions product adoption at technical level
- Helps with escalations



Global Professional Services

- Plan for scope creep
- Buy time to define
- Prepare for the future



Customer Success Manager

- Assigned to strategic accounts
- Proactive focus on customers business outcomes
- Drives product adoption at business level
- Non-technical



Global Learning Services

- in-person or remote training
- Hands-on immersive instruction
- Cloud-based lab

How can the TAM help?

Primary point of contact

key contact for Red Hat technical topics as well as **multi-vendor support issues**

Advise and advocate

learn **best practices** and have **your interests** promoted within Red Hat

Proactive planning and technical reviews

identify areas of concern early and **prevent issues** before they happen

Facilitate issue resolution

manage **strategic cases** and engage Subject Matter Experts as needed

Visibility into Red Hat roadmaps

regular information on **upcoming relevant releases**

Taiwan TAM Team

Platform TAM	Expertise is based on the platform specialty group, with knowledge of the kernel, Red Hat Satellite, filesystems, logical volume management (LVM), and other Red Hat Enterprise Linux®-based platform tool sets.	4
OpenShift TAM	Expertise is based in the cloud specialty group, with knowledge of Red Hat OpenShift® and Red Hat Platform-as-a-Service (PaaS)-based tool sets	4
Middleware TAM	Expertise is based in the middleware specialty group, with knowledge of Red Hat JBoss® tools, OpenShift development, and mobile-based tool sets	2
Storage TAM	Expertise is based in the storage specialty group, with knowledge of Red Hat Ceph® Storage, Red Hat Gluster® Storage, and Red Hat Storage-based tool sets	PF
OpenStack TAM	Expertise is based in the cloud specialty group, with knowledge of Red Hat OpenStack® Platform.	OCP

What you will learn

- Overview of public cloud provisioning
- Converting shell commands into Ansible commands
- Retrieving information from hosts
- Deploying applications at scale
- Self-service IT via surveys
- Overview of System Roles for Red Hat Enterprise Linux
- Overview of Red Hat Insights integration



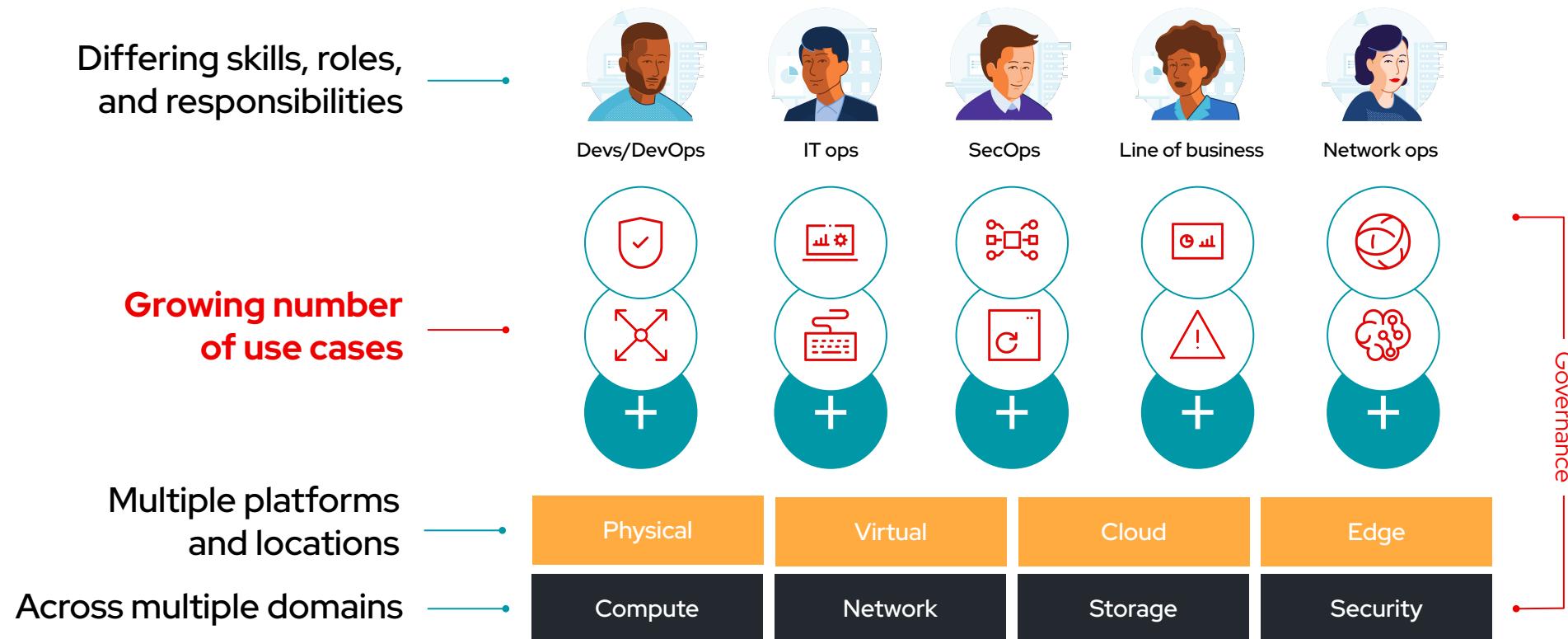
Red Hat
Ansible Automation
Platform

Introduction

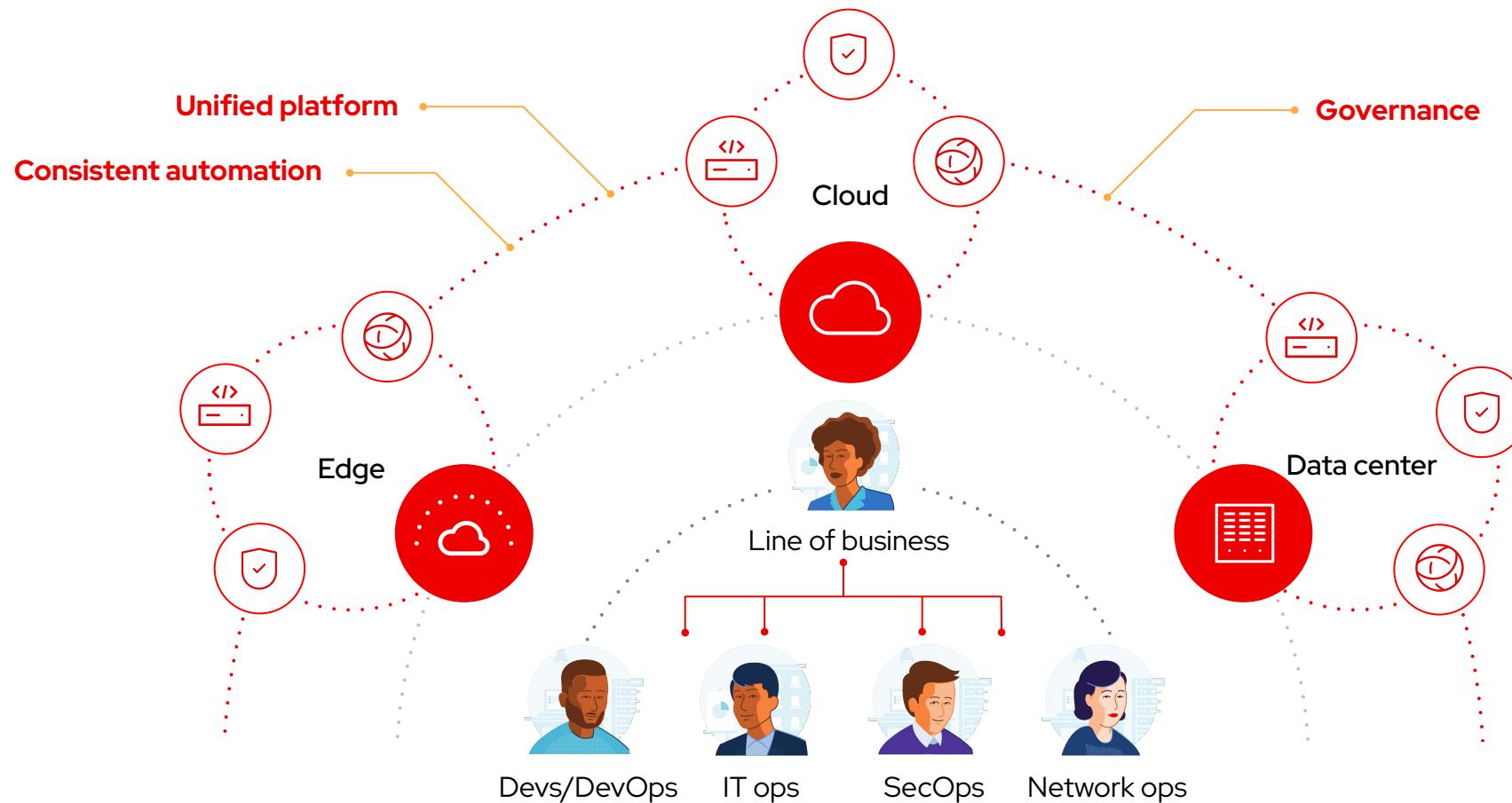
Topics Covered:

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

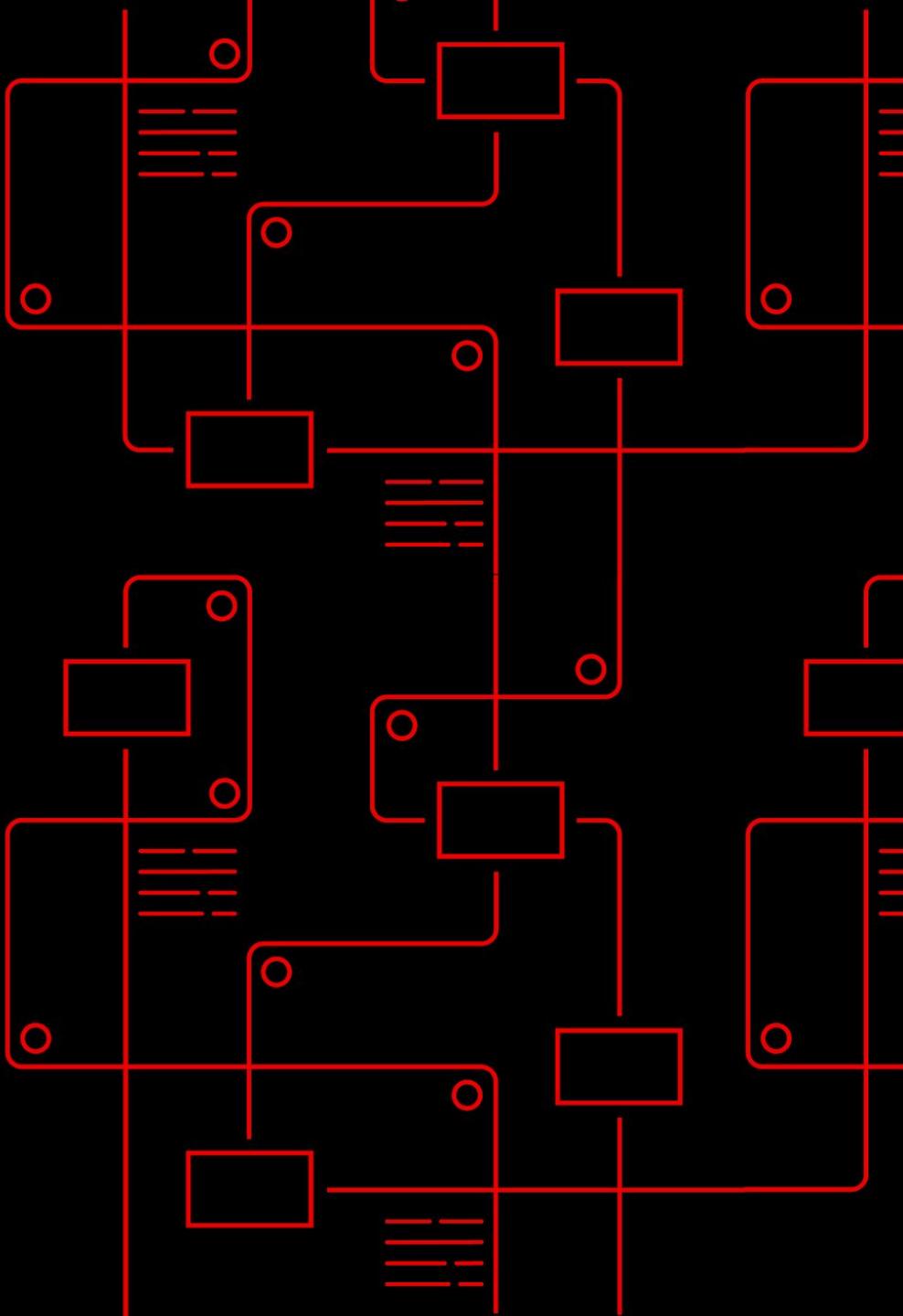
Many organizations share the same challenge



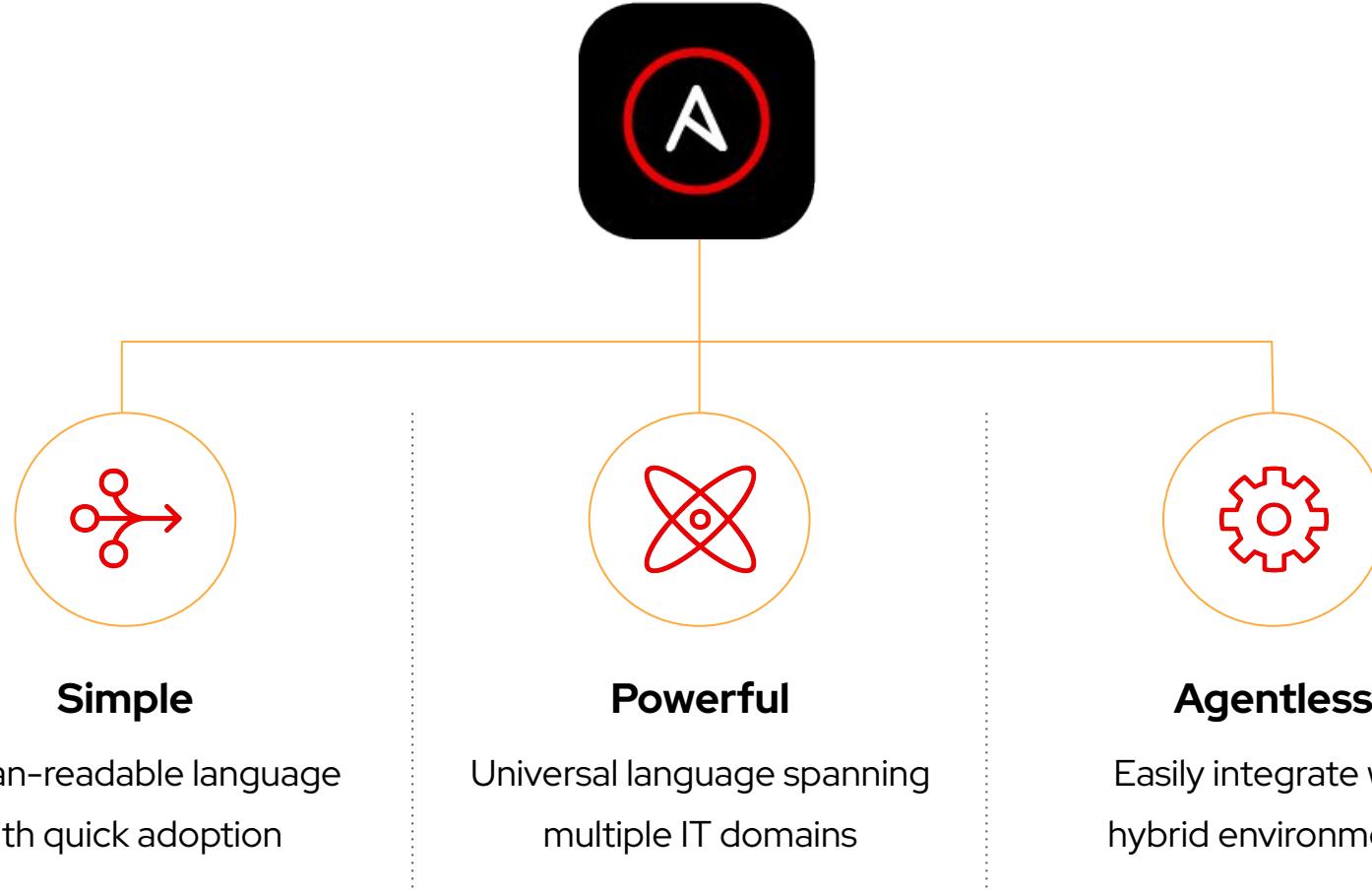
The solution? Break down the silos.



Why Red Hat® Ansible® Automation Platform?



Ansible is the **de facto** automation language.



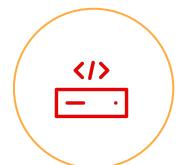
Supported and certified **content** you can trust.

130+

Certified Content
Collections

55+

Certified technology
partners



Infrastructure



Cloud



Network



Security



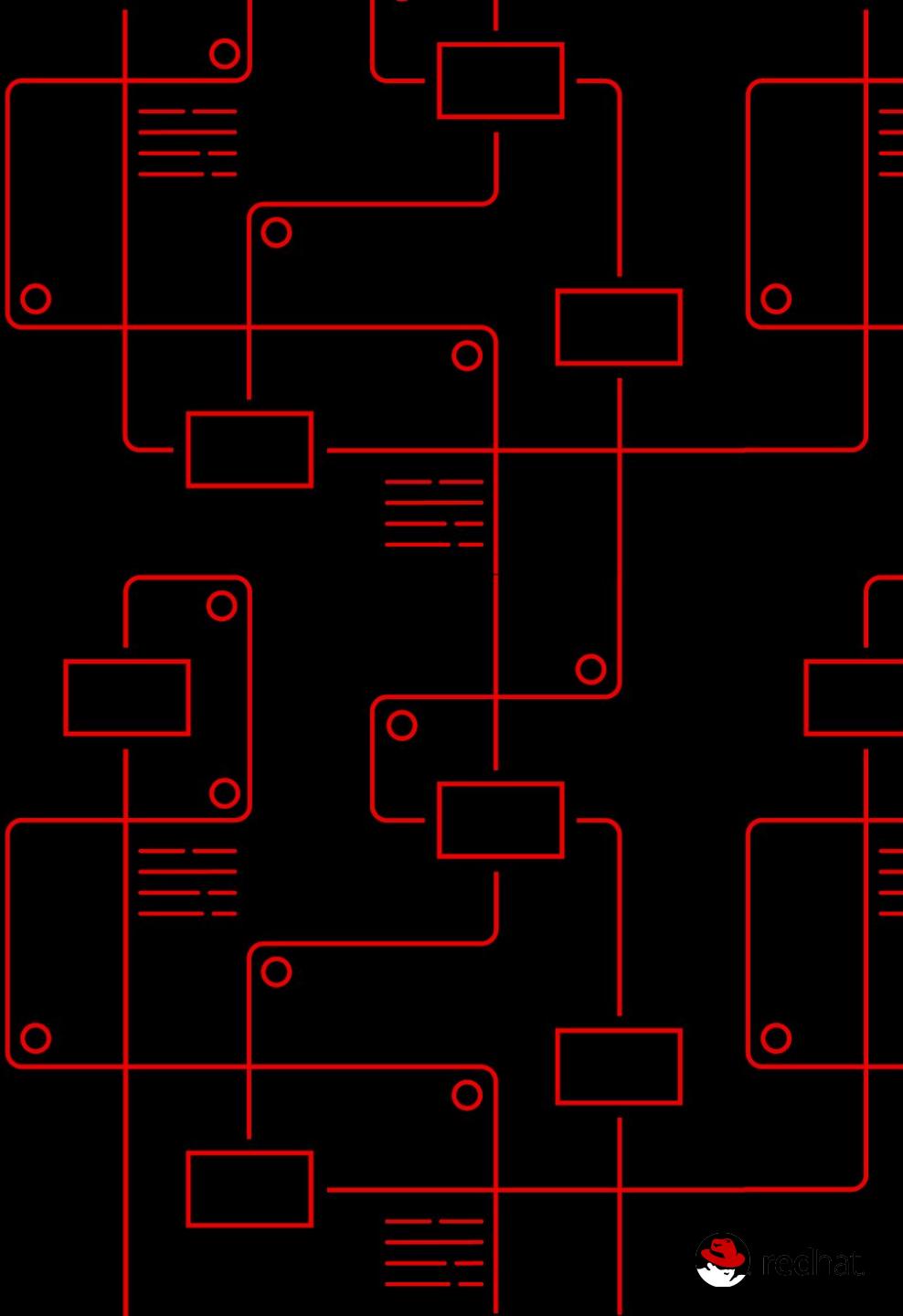
Edge



The flexibility to scale, wherever that may be.



What is Red Hat® Ansible® Automation Platform?



A platform for the entire automation team.

Architecture

Flexible container-native architecture

Real-time analytics and reporting

Scale globally with distributed execution across regions



Content creation

Content creation tooling

Portable distribution and reliable execution

Large ecosystem of certified automation



Operations

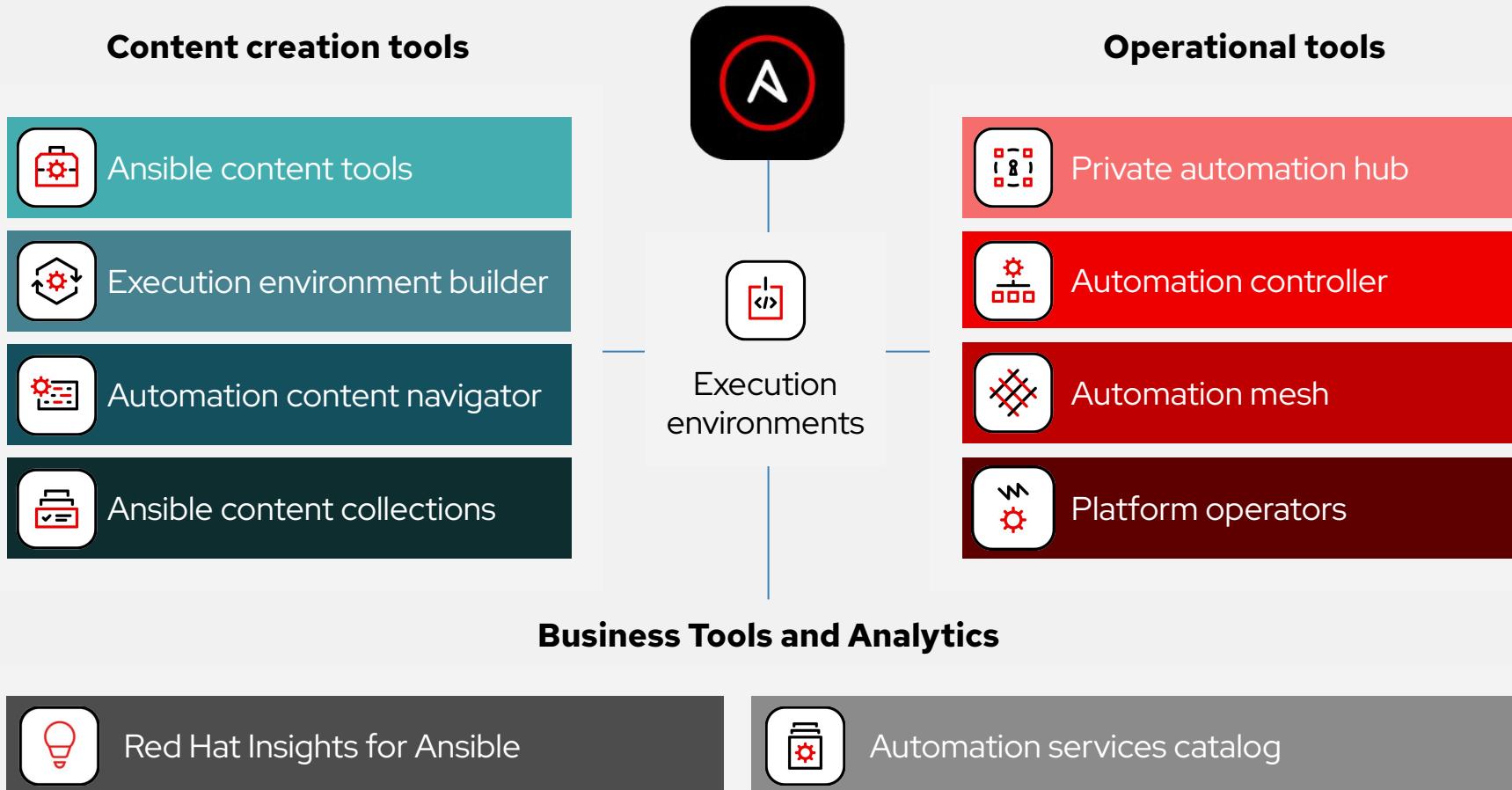
Enterprise features:
WebUI, API, role-based access control (RBAC), auditing and workflows for managing at scale

Hosted and private content management solutions

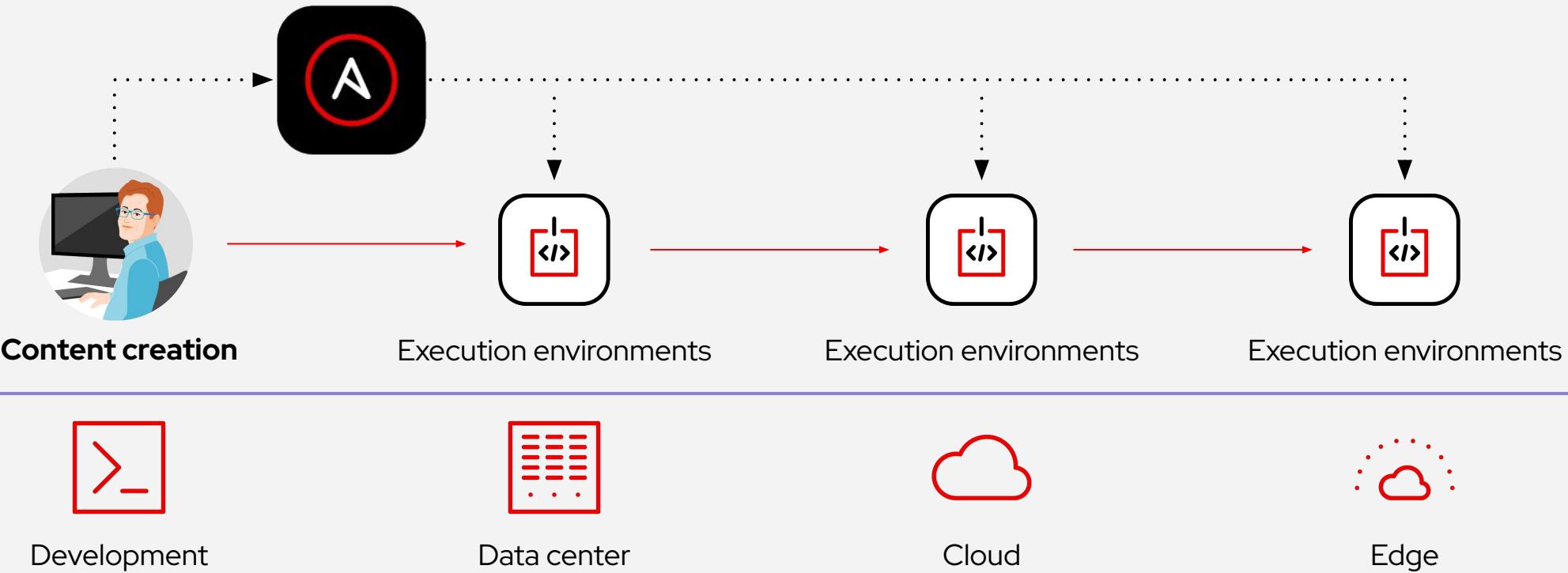
Integrates with your environment



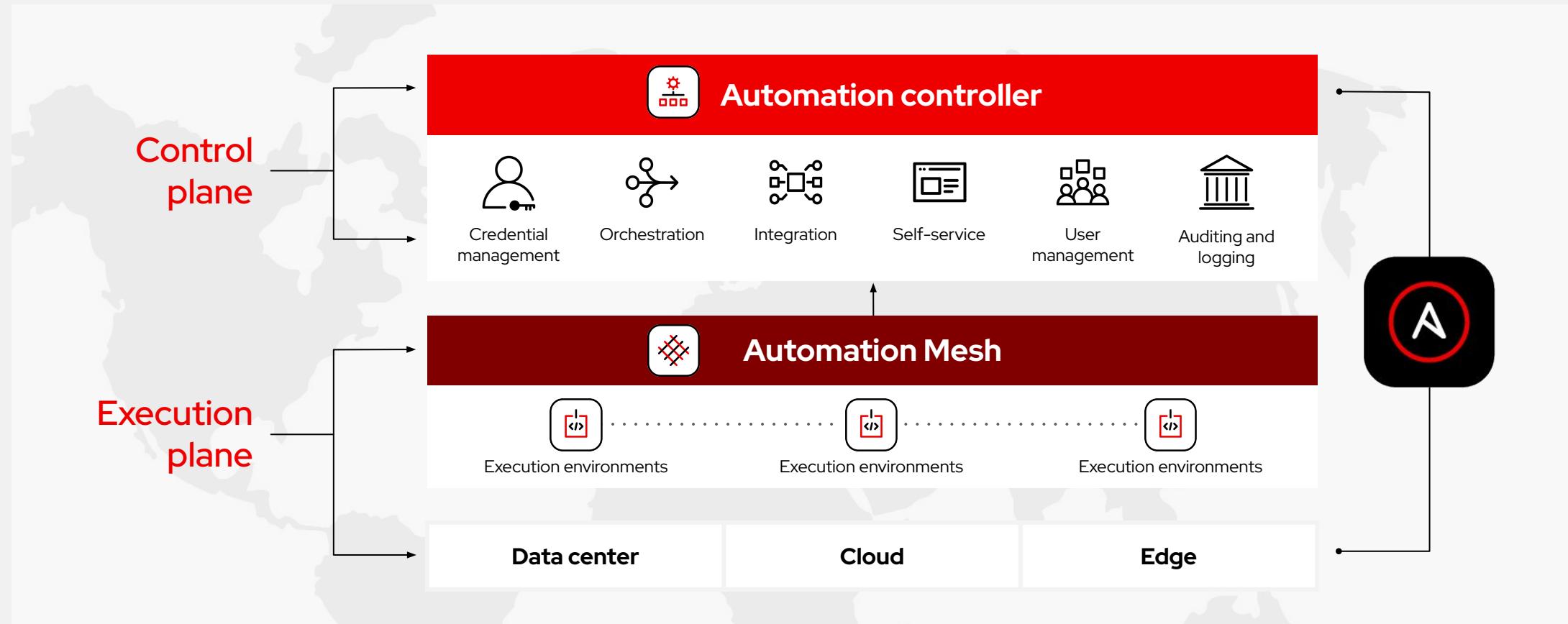
An integrated solution for the enterprise.



Built for consistency. Portability is reliability.



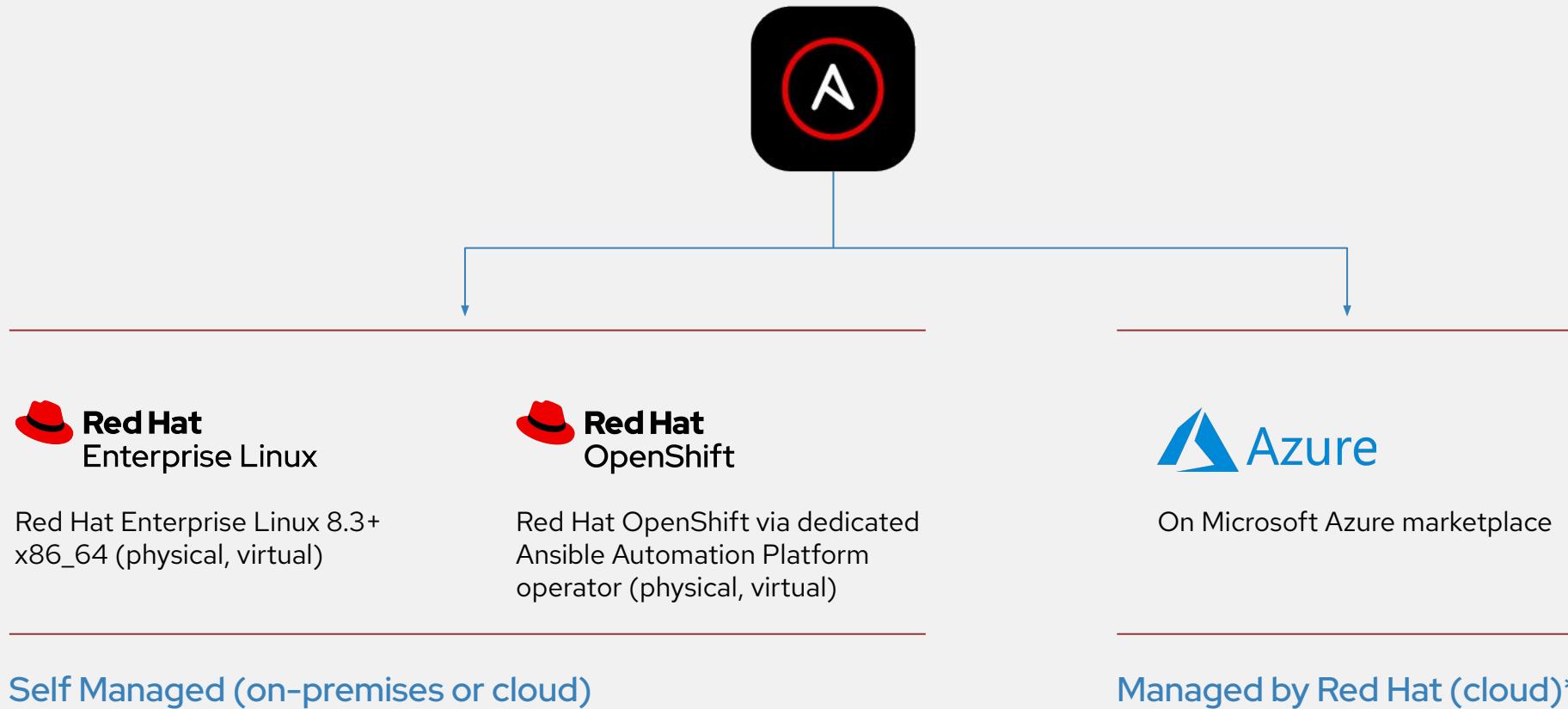
A distributed architecture built for scale.



The capabilities you need across your IT footprint.



Ansible Automation Platform hosting options



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

Cloud

Topics Covered:

- Understanding the Ansible Infrastructure
- Check the prerequisites



Red Hat
Ansible Automation
Platform

The lab environment today

- **Drink our own champagne.**

Provisioned by, configured by, and managed by Red Hat Ansible Automation Platform.

<https://github.com/ansible/workshops>

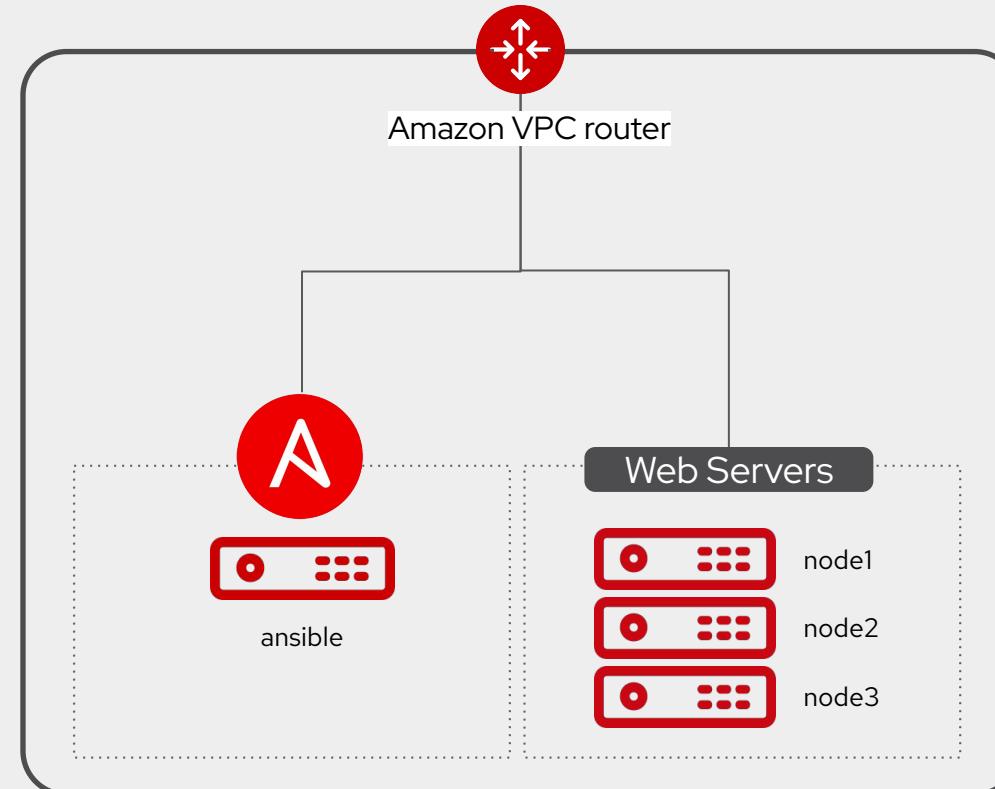
- **Learn with the real thing**

Every student will have their own fully licensed Red Hat Ansible Tower control node. No emulators or simulators here.

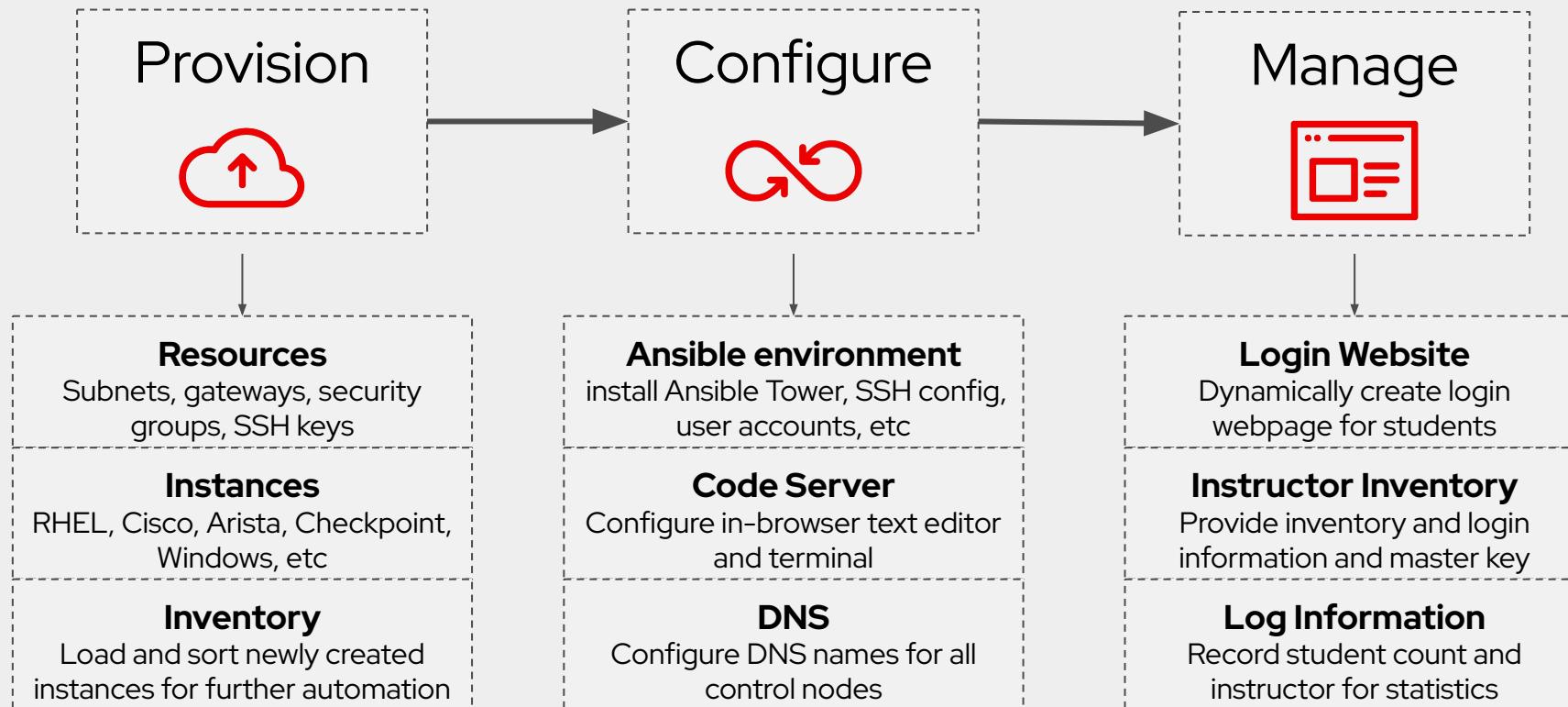
- **Red Hat Enterprise Linux**

All four nodes are enterprise Linux, showcasing real life use-cases to help spark ideas for what you can automate today.

Workbench Topology



How does it work?





Red Hat
Ansible Automation
Platform

Exercise 1

Topics Covered:

- Understanding the Ansible Infrastructure
- Check the prerequisites



Ansible Core (`ansible-core`)

What is it?

- ▶ The main building block for Ansible
- ▶ Simple YAML syntax to develop Ansible Playbooks
- ▶ Provides CLI tools to develop, test and run playbooks
- ▶ Pluggable architecture that allows extensions through Content Collections

```
---  
- name: Shutdown VM guest  
hosts: localhost  
gather_facts: false  
tasks:  
  - name: Turn off specified VM guest  
    vmware.vmware_rest.vcenter_vm_guest_power:  
      state: shutdown  
      vm: 1021343  
      vcenter_hostname: vcenter.demoredhat.com  
      vcenter_username: admin  
      vcenter_password: tedlasso
```

Ansible playbooks

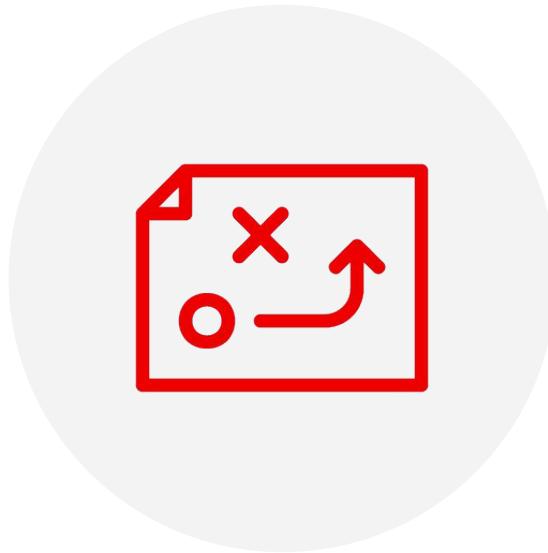
```
---
```

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

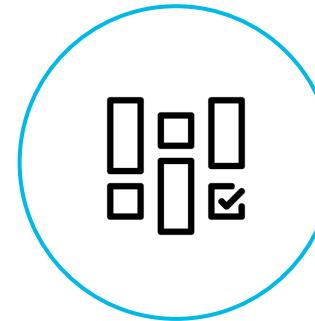
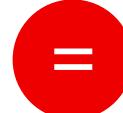
tasks:

- name: Ensure the httpd package is installed
ansible.builtin.yum:
 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

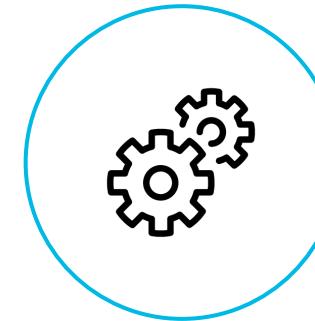
What makes up an Ansible playbook?



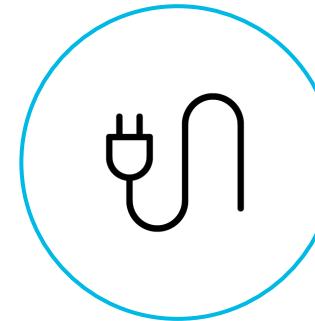
Playbook



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

```
...  
- name: Ensure the httpd package is installed  
  hosts: web  
  become: true
```



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 created in Python, or Powershell for Windows setups, but can be developed in any language

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



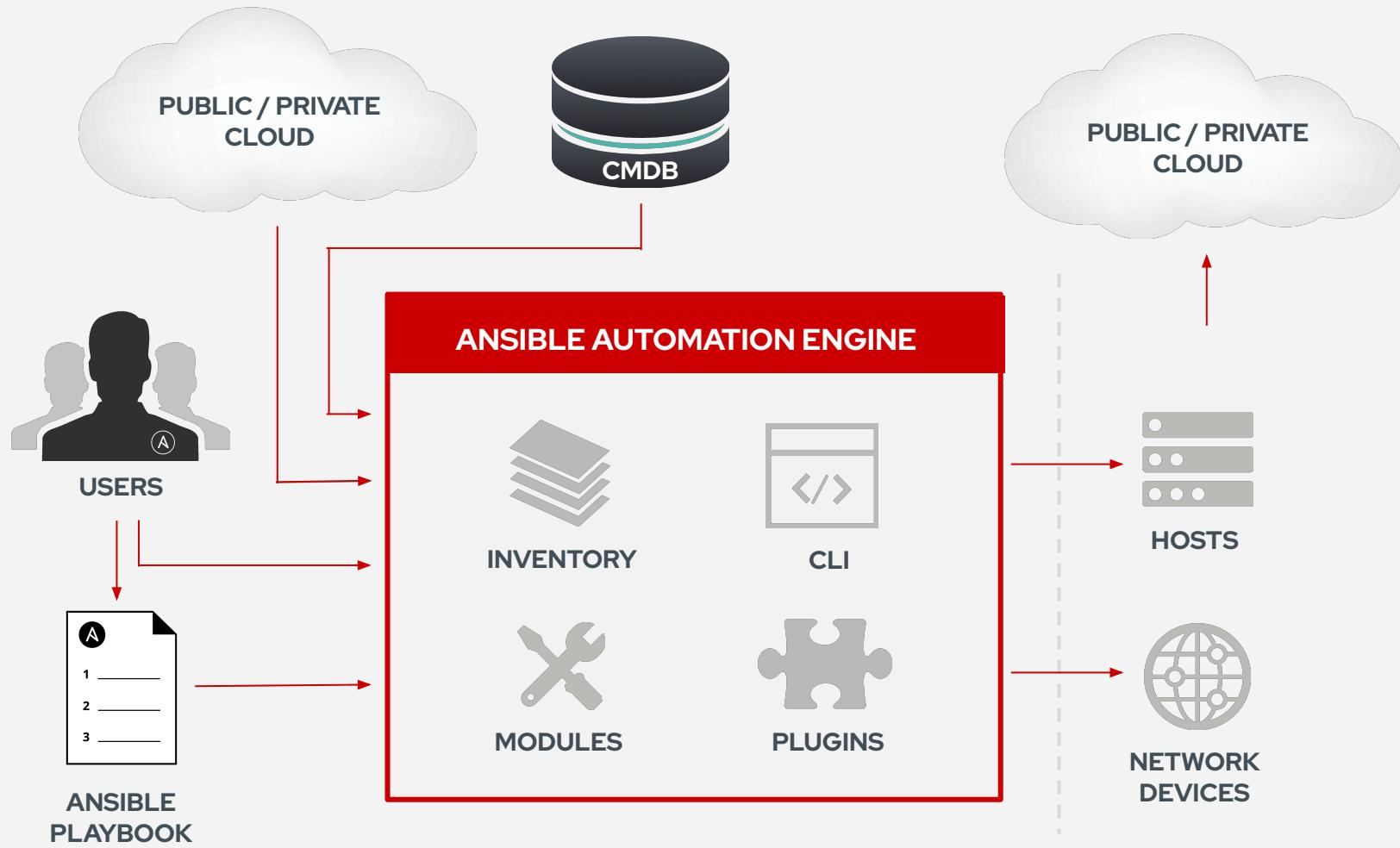
Ansible plugins. The “extra bits”.

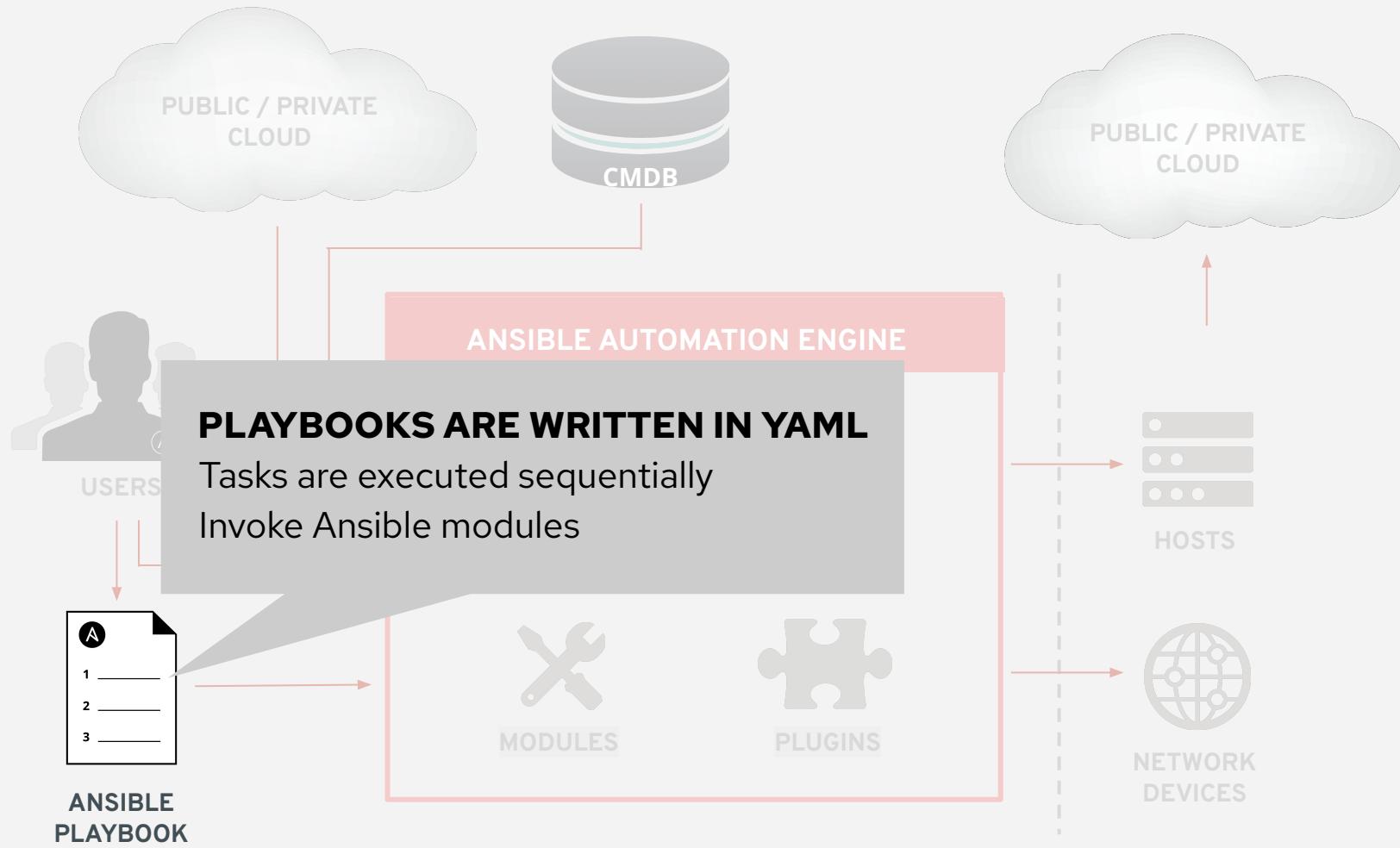
What are they?

- ▶ 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

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

Example filter plugins:
{{ some_variable | to_nice_json }}
{{ some_variable | to_nice_yaml }}
```



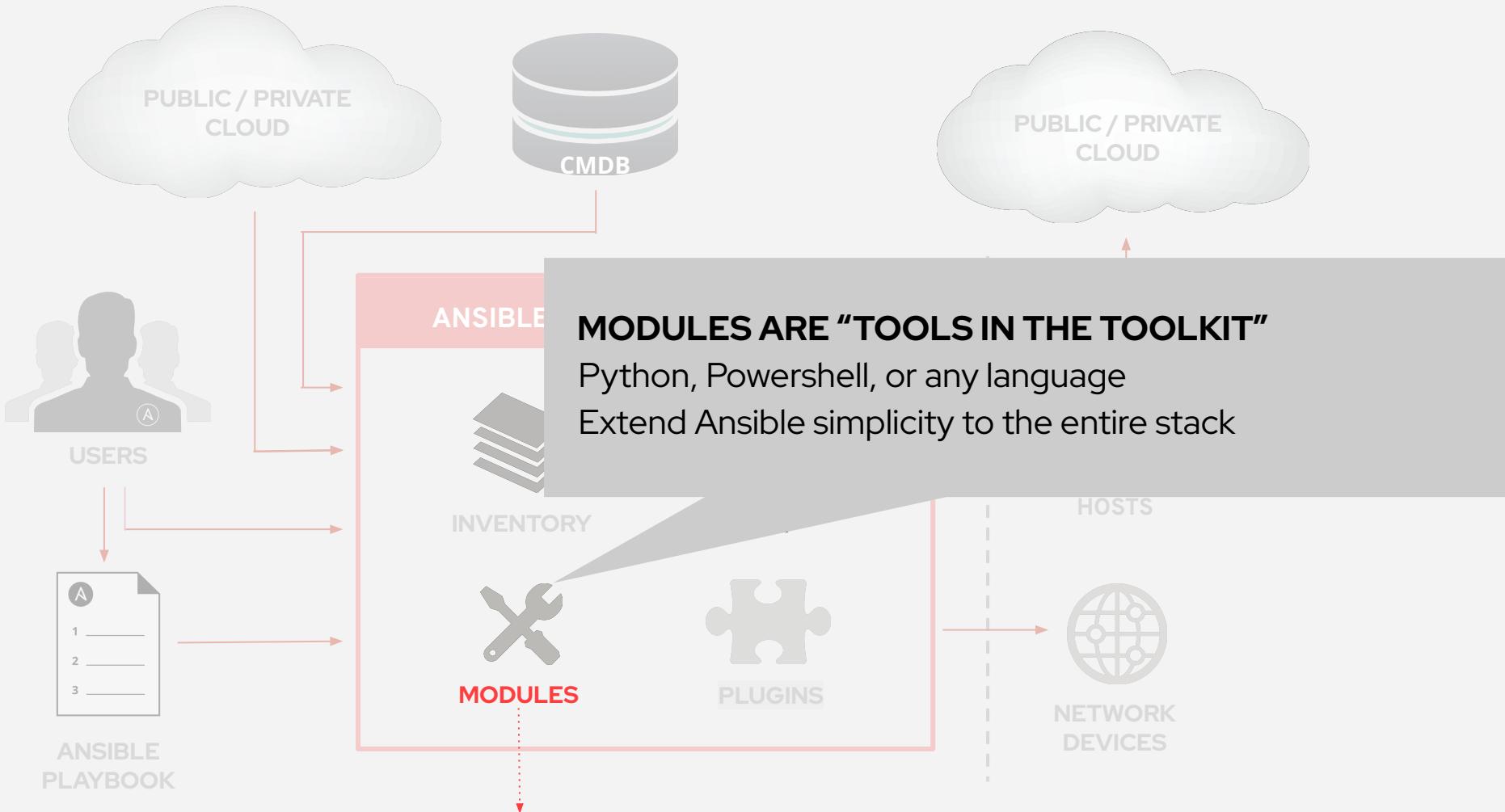


```
---
```

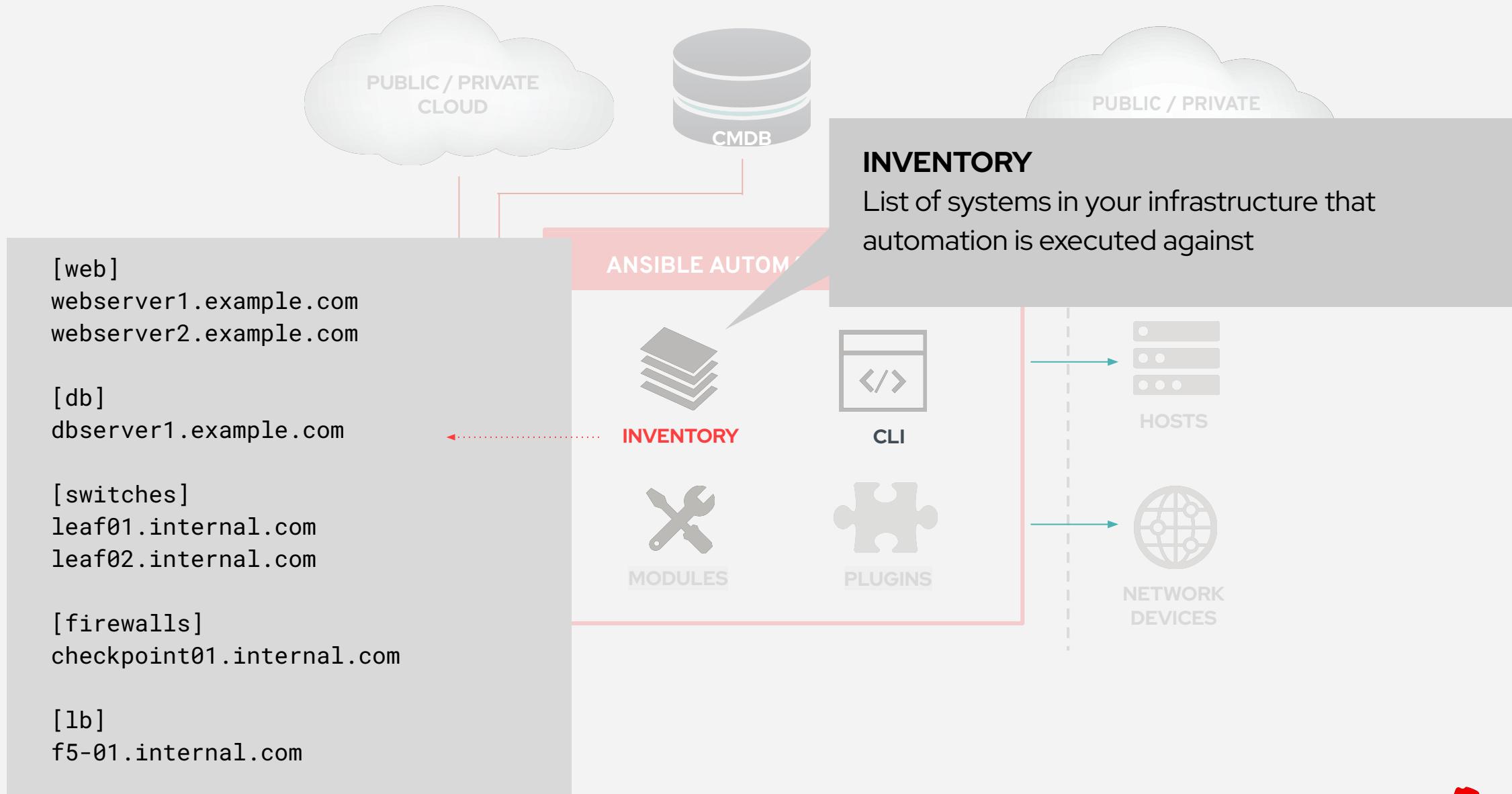
- **name: install and start apache**
hosts: web
become: yes

tasks:

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



```
- name: latest index.html file is present
  template:
    src: files/index.j2
    dest: /var/www/html/
```



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 Ansible Automation Platform

Lab Time

Complete exercise **1-setup** now in your lab environment



Red Hat



Red Hat
Ansible Automation
Platform

Exercise 2

Topics Covered:

- Ansible inventories
- Main Ansible config file
- Modules and ad-hoc commands
- Example: Bash vs. Ansible

Inventory

- Ansible works against multiple systems in an **inventory**
- Inventory is usually file based
- Can have multiple groups
- Can have variables for each group or even host

Understanding Inventory - Basic

```
node1  
node2  
node3  
ansible  
10.20.30.40
```

Understanding Inventory - Basic

[web]

```
node1 ansible_host=3.22.77.141
node2 ansible_host=3.15.193.71
node3 ansible_host=3.15.1.72
```

[control]

```
ansible ansible_host=18.217.162.148
```

Understanding Inventory - Variables

[all:vars]

```
ansible user=student1  
ansible ssh pass=ansible1234  
ansible_port=22
```

[web]

```
node1 ansible host=3.22.77.141  
node2 ansible host=3.15.193.71  
node3 ansible_host=3.15.1.72
```

[control]

```
ansible ansible_host=18.217.162.148
```

First Ad-Hoc Command: ping

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

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

Ad-Hoc Commands ping

```
# Check connections (submarine ping, not ICMP)
[user@ansible] $ ansible all -m ping
```

```
node1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python":
"/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

Bash vs. Ansible

```
echo Running mssql-conf setup...
sudo
MSSQL_SA_PASSWORD=$MSSQL_SA_PASSWORD \
MSSQL_PID=$MSSQL_PID \
/opt/mssql/bin/mssql-conf -n setup accept-eula

echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >>
~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >>
~/.bashrc
source ~/.bashrc
```

```
- name: Run mssql-conf setup
  command: /opt/mssql/bin/mssql-conf -n setup
  accept-eula
  environment:
    - MSSQL_SA_PASSWORD: "{{ MSSQL_SA_PASSWORD }}"
    - MSSQL_PID: "{{ MSSQL_PID }}"
  when: install is changed

- name: Add mssql-tools to $PATH
  lineinfile:
    path: "{{ item }}"
    line: export PATH="$PATH:/opt/mssql-tools/bin"
  loop:
    - ~/.bash_profile
    - ~/.bashrc
```



Red Hat

Ansible Automation Platform

Lab Time

Complete exercise **2-adhoc** now in your lab environment

Extra lab:

https://dchan-redhat.github.io/exercises/ansible_rhel_90/2-adhoc/



Red Hat



Red Hat
Ansible Automation
Platform

Exercise 3

Topics Covered:

- Playbooks basics
- Running a playbook

An Ansible Playbook

A play

```
---
- 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
```

An Ansible Playbook

A task

```
---
- 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
```

An Ansible Playbook

module



```
---
- 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
```

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
```



Red Hat

Ansible Automation Platform

Lab Time

Complete exercise **3-playbooks** now in your lab environment



Red Hat



Red Hat
Ansible Automation
Platform

Exercise 4

Topics Covered:

- Working with variables
- What are facts?

An Ansible Playbook Variable Example

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

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

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

An Ansible Playbook Variable Example

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

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

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

ansible is awesome



Facts

- Structured data in the form of Ansible variables
- Information is capture from the host
- Ad-hoc command **setup** will show facts

```
"ansible_facts": {  
    "ansible_default_ipv4": {  
        "address": "10.41.17.37",  
        "macaddress": "00:69:08:3b:a9:16",  
        "interface": "eth0",  
    ...  
}
```

Ansible Variables and Facts

```
---
```

- name: Output facts within a playbook
hosts: all

tasks:

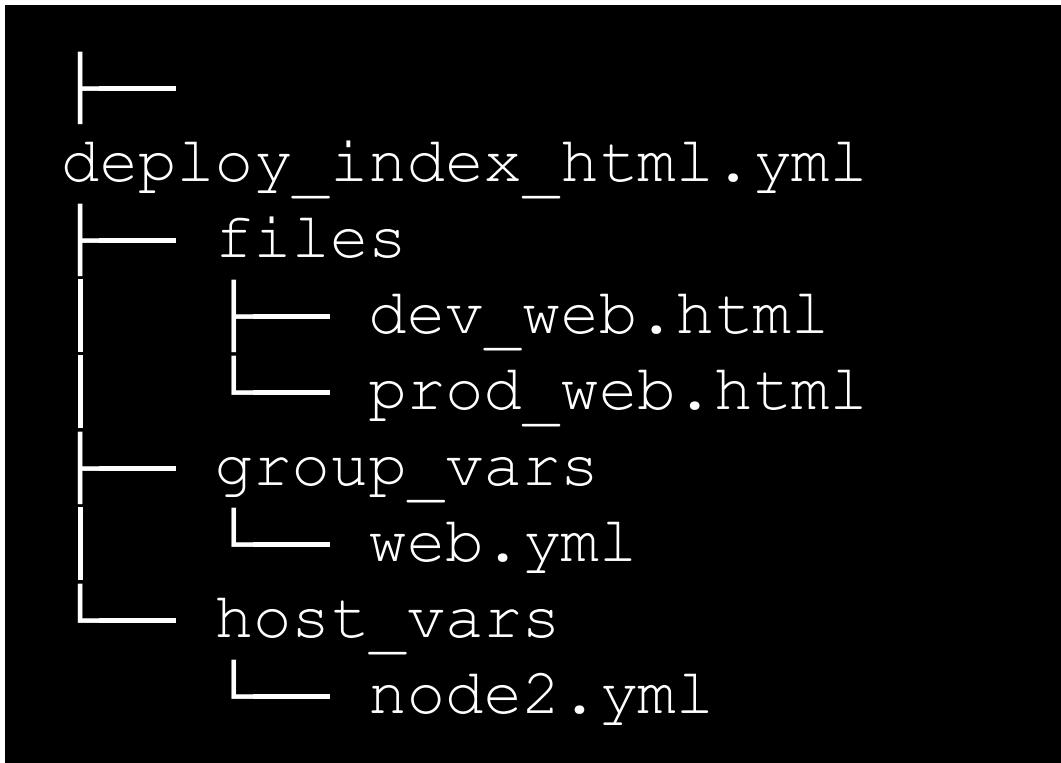
- name: Prints Ansible facts
debug:
msg: "The default IPv4 address of {{ ansible_fqdn }}
is {{ ansible_default_ipv4.address }}"

```
TASK [Prints Ansible facts] ****
ok: [node3] =>
    msg: The default IPv4 address of node3 is 172.16.63.104
ok: [node1] =>
    msg: The default IPv4 address of node1 is 172.16.178.80
ok: [node2] =>
    msg: The default IPv4 address of node2 is 172.16.166.120
ok: [ansible] =>
    msg: The default IPv4 address of student1.sean-may4.rhdemo.io is 172.16.86.242
```

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



```
$ 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
```



Red Hat Ansible Automation Platform

Lab Time

Complete exercise **4-variables** now in your lab environment



Red Hat



Red Hat
Ansible Automation
Platform

Exercise 5

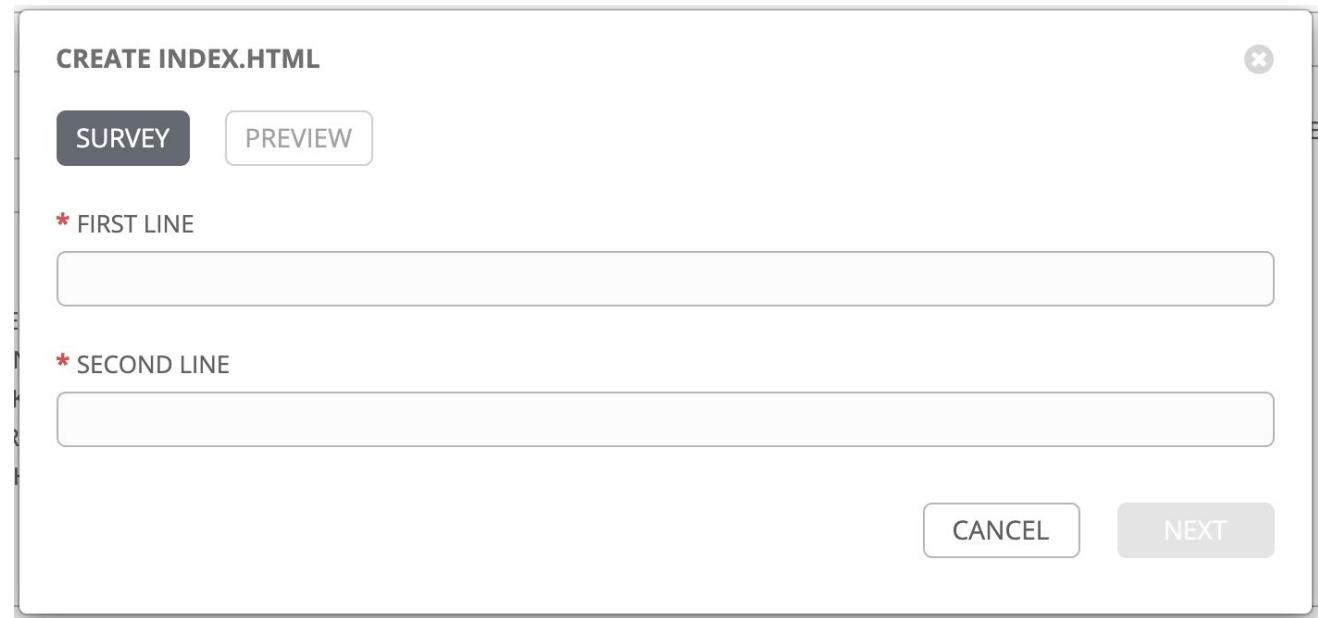
Topics Covered:

- Surveys

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.



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 Red Hat Ansible Tower web interface. On the left is a dark sidebar with navigation links: VIEWS (Dashboard, Jobs, Schedules, My View), RESOURCES (Templates, Credentials, Projects, Inventories, Inventory Scripts), and ACCESS. The 'Templates' link is currently selected. The main content area has a title 'TEMPLATES / Create index.html'. Below the title is a modal dialog titled 'Create index.html' with several tabs: DETAILS (selected), PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and EDIT SURVEY. The 'EDIT SURVEY' tab is highlighted with a red box. The modal contains fields for NAME ('Create index.html'), DESCRIPTION, and JOB TYPE. It also includes sections for INVENTORY ('Workshop Inventory'), PROJECT ('Workshop Project'), PLAYBOOK ('rhel/apache/apache_role_inst...'), CREDENTIALS ('Workshop Credential'), FORKS ('0'), LIMIT ('web'), VERBOSITY ('0 (Normal)'), JOB TAGS, and SKIP TAGS. Each section has a 'PROMPT ON LAUNCH' checkbox.

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 'Create index.html | SURVEY' configuration window in the Tower interface. On the left, the 'ADD SURVEY PROMPT' section contains fields for 'PROMPT' (a text input field), 'DESCRIPTION' (a text input field), 'ANSWER VARIABLE NAME' (a text input field), 'ANSWER TYPE' (a dropdown menu), and a checked 'REQUIRED' checkbox. Below these are 'CLEAR' and '+ ADD' buttons. On the right, the 'PREVIEW' section displays two lines of survey prompts: 'FIRST LINE' and 'SECOND LINE', each with edit and delete icons. At the bottom are 'DELETE SURVEY', 'CANCEL', and 'SAVE' buttons. The top navigation bar includes 'admin', a notification bell (0), and other UI elements.

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 a survey dialog box titled "CREATE INDEX.HTML". At the top, there are two buttons: "SURVEY" (which is highlighted in dark grey) and "PREVIEW" (which is white with a grey border). In the center, there are two input fields labeled with asterisks: "* FIRST LINE" and "* SECOND LINE". Both fields have a large, empty rectangular input area. At the bottom right, there are two buttons: "CANCEL" (white with a grey border) and "NEXT" (light grey).



Red Hat Ansible Automation Platform

Lab Time

Complete exercise **5-surveys** now in your lab environment



Red Hat



Red Hat
Ansible Automation
Platform

Exercise 6

Topics Covered:

- Red Hat Enterprise Linux System Roles

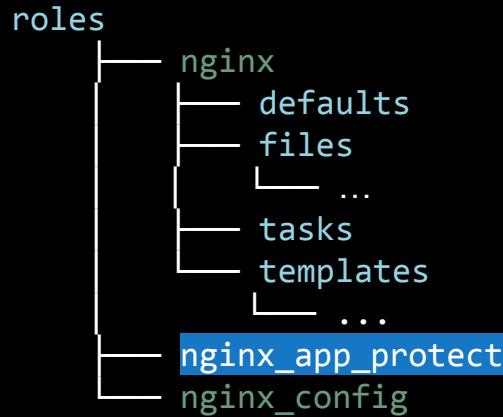


Ansible Roles. Reusable automation actions.

What are they?

- ▶ Group 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

```
...  
- name: Install and start apache  
hosts: web  
ansible.builtin.roles:  
  - common  
  - webservers
```



deploy-nginx.yml

```
---
```

```
- name: Install NGINX Plus
  hosts: all
  tasks:
```

```
  - name: Install NGINX App Protect
    include_role:
      name: nginx_app_protect
  vars:
    nginx_app_protect_setup_license: false
    nginx_app_protect_remove_license: false
    nginx_app_protect_install_signatures: false
```

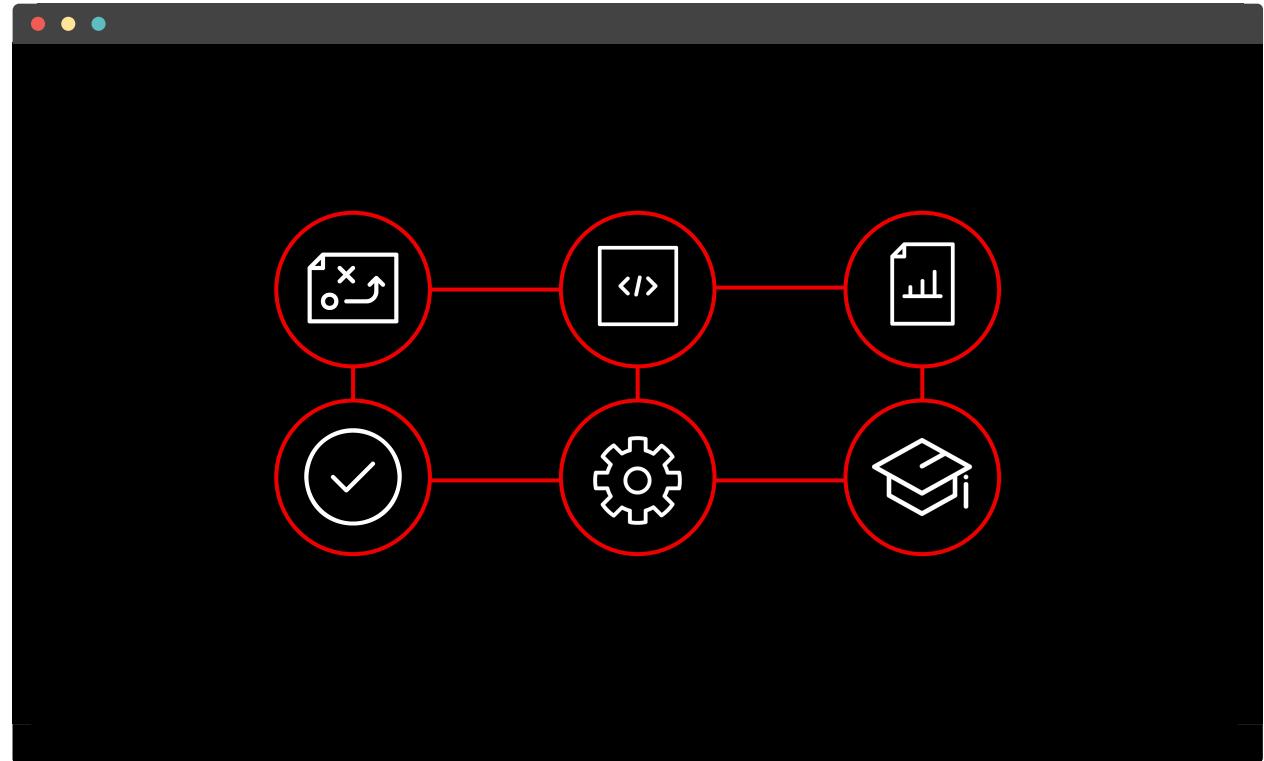


Content Collections.

Simplified, consistent content delivery.

What are they?

- ▶ Group 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





```
nginx_core
├── galaxy.yml
├── meta
└── playbooks
    └── deploy-nginx.yml
    ...
plugins
README.md
roles
└── nginx
    ├── defaults
    ├── files
    │   └── ...
    ├── tasks
    └── templates
        └── ...
    └── nginx_app_protect
    └── nginx_config
```

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 automation content.

console.redhat.com

What is it?

- ▶ Hosted source of trusted Red Hat and Certified Partner Content Collections
- ▶ Integrated documentation and examples
- ▶ Configurable as primary content collection source for your automation environment
- ▶ Access to hosted automation hub and content included in subscription

The screenshot shows the 'Approval dashboard' with a red border. At the top, there's a search bar with dropdowns for 'Repository' and 'published', and a magnifying glass icon. Below the search bar, there are filter buttons for 'repository' and 'published', and a 'Clear all filters' link. The main area is a table with columns: 'Namespace', 'Collection', 'Version', 'Date created', and 'Status'. A single row is shown: 'kyleabenson' under Namespace, 'fest2020collection' under Collection, '1.1.1' under Version, '3 hours ago' under Date created, and 'Approved' with a checkmark under Status. At the bottom of the table, there are pagination controls showing '1-1 of 1' and '1 of 1'.

Namespace	Collection	Version	Date created	Status
kyleabenson	fest2020collection	1.1.1	3 hours ago	Approved

Linux System Roles

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

Examples



Email



kdump



network



selinux



timesync



firewall

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
```



Red Hat

Ansible Automation Platform

Lab Time

Complete exercise **6-system-roles** now in your lab environment



Red Hat



Red Hat
Ansible Automation
Platform

1 more Exercise

Topics Covered:

- Workflows

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 left sidebar has a dark theme with white icons and text. It includes sections for Views (Dashboard, Jobs, Schedules, My View), Resources (Templates, Credentials, Projects, Inventories, Inventory Scripts), Access (Organizations, Users, Teams), and Administration. The 'Templates' icon is highlighted with a light blue background. The main content area is titled 'TEMPLATES' and shows a list of six job templates. Each template entry includes the name, type ('Job Template'), and three action icons (pencil, copy, delete). The list is ordered by name (ascending). At the top of the main content area, there is a search bar, a key icon, and a green '+' button. The top right corner shows the user 'admin' and some system status icons.

Template Name	Type	Action Icons
Demo Job Template	Job Template	Pencil, Copy, Delete
Network-Commands	Job Template	Pencil, Copy, Delete
Network-Restore	Job Template	Pencil, Copy, Delete
Network-System	Job Template	Pencil, Copy, Delete
Network-Time	Job Template	Pencil, Copy, Delete
Network-User	Job Template	Pencil, Copy, Delete

Adding a new Workflow Template

To add a new **Workflow** click on the green + button



This time select the **Workflow Template**

The screenshot shows the Ansible Tower interface with the 'TEMPLATES' view selected. On the left, there's a sidebar with various navigation options like Dashboard, Jobs, Schedules, My View, Templates (which is currently selected), Credentials, Projects, Inventories, Inventory Scripts, Organizations, and Users. The main area displays a list of templates with their names, types (Job Template or Workflow Template), and actions (Edit, Copy, Delete). A red box highlights the 'Workflow Template' option in the dropdown menu for the first template, 'Backup network configurations'. The interface includes a search bar, a key icon, and filter buttons for 'Compact' and 'Expanded' view.

Template Name	Type	Action
Backup network configurations	Job Template	Edit Copy Delete
Configure Banner	Job Template	Edit Copy Delete
Demo Job Template	Job Template	Edit Copy Delete
Network-Commands	Job Template	Edit Copy Delete
Network-Restore	Job Template	Edit Copy Delete
Network-System	Job Template	Edit Copy Delete

Creating the workflow

Fill out the required parameters and click **SAVE**. As soon as the Workflow Template is saved the WORKFLOW VISUALIZER will open.

The screenshot shows the Tower interface for creating a new workflow template. The left sidebar has sections for Views, Resources (Templates selected), Credentials, Projects, Inventories, Inventory Scripts, Access, and Organizations. The main area is titled 'TEMPLATES / WORKSHOP WORKFLOW' and contains a modal for 'WORKSHOP WORKFLOW'. The modal has tabs for DETAILS (selected), PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and ADD SURVEY. The 'WORKFLOW VISUALIZER' button is highlighted with a red box. The form fields include:

- NAME:** WORKSHOP WORKFLOW
- DESCRIPTION:** (empty)
- ORGANIZATION:** Default
- INVENTORY:** Workshop Inventory
- LABELS:** (empty)
- OPTIONS:** ENABLE CONCURRENT JOBS
- EXTRA VARIABLES:** (empty)
- PROMPT ON LAUNCH:** (checkbox)

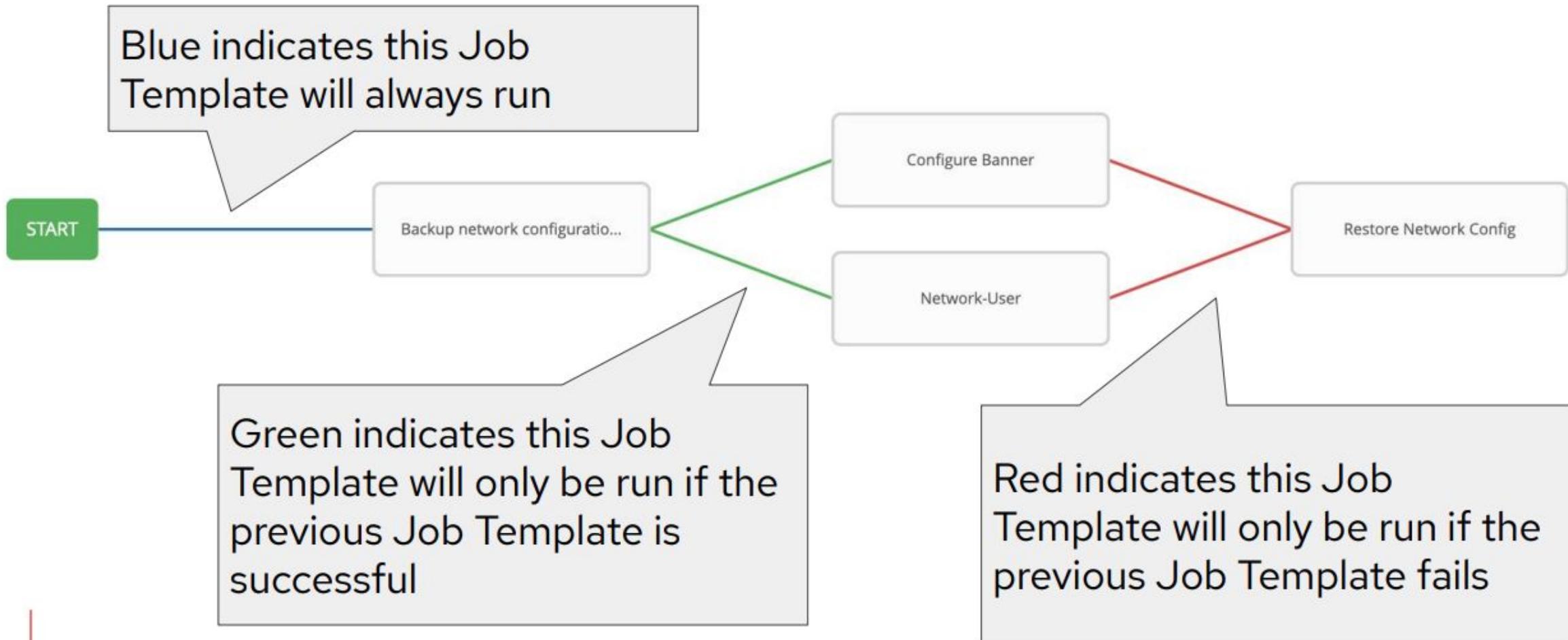
Workflow Visualizer

The workflow visualizer will start as a blank canvas.



Visualizing a Workflow

Workflows can branch out, or converge in.





Red Hat
Ansible Automation
Platform

2 more Exercise

Topics Covered:

- Red Hat Insights intro
- Insights integration

Red Hat Insights

Included with your Red Hat Enterprise Linux subscription

Assesses

customer's Red Hat environments

Remediates

findings with prescriptive remediation steps or an Ansible playbook

Insights

rule contributions directly from Red Hat subject matter experts

Identifying risks for Availability, performance, stability and security



Red Hat Insights

Remediations > May2019_Critical_Fixes

Overview

Rules

Inventory

Remediations

Documentation

May2019_Critical_Fixes

Download Playbook

Delete

Systems reboot

6

No reboot

0

Reboot required



Auto reboot

Playbook details

Created by: John Spinks

Created: a minute ago

Last modified by: John Spinks

Insights plans with Ansible playbooks

Solve common issues through Ansible Automation

Pages > >>

Actions ↑

Resolution

Reboot required

Systems

Type

▼	Dnsmasq with listening processes vulnerable to remote code execution via crafted DNS requests (CVE-2017-14491)	Update dnsmasq package and restart related service(s)	6	Insights
---	--	---	---	----------

Systems

ic3.example.com

ic4.example.com

ic6.example.com

ic7.example.com

ANSIBLE & INSIGHTS

While Insights includes Ansible playbooks for risks, Insights alone can't perform remediation of the risks.

Insights

- Insights provides Ansible Playbooks for resolving many common risks.
- Dynamically generates Ansible Playbooks for risk remediation
- Playbooks can be downloaded and run via `ansible-playbook` or Satellite

Insights connected to Ansible Tower

- View identified risks in the Tower inventory
- Execute generated Ansible Playbook as a Tower job
- Use Tower for enterprise risk remediation

Next Steps

GET STARTED

ansible.com/get-started

ansible.com/tower-trial

WORKSHOPS & TRAINING

ansible.com/workshops

[Red Hat Training](#)

JOIN THE COMMUNITY

ansible.com/community

SHARE YOUR STORY

[Follow us @Ansible](#)

[Friend us on Facebook](#)



AnsibleFest

October 13-14, 2020 | Virtual Experience



Thank you



linkedin.com/company/red-hat



youtube.com/AnsibleAutomation



facebook.com/ansibleautomation



twitter.com/ansible



github.com/ansible