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	**Leb 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

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**

Red Hat

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**

https://www.redhat.com/en/services/support/technical-account-management

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

Red Hat

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

Red Hat
Ansible Automation
Platform

Introduction

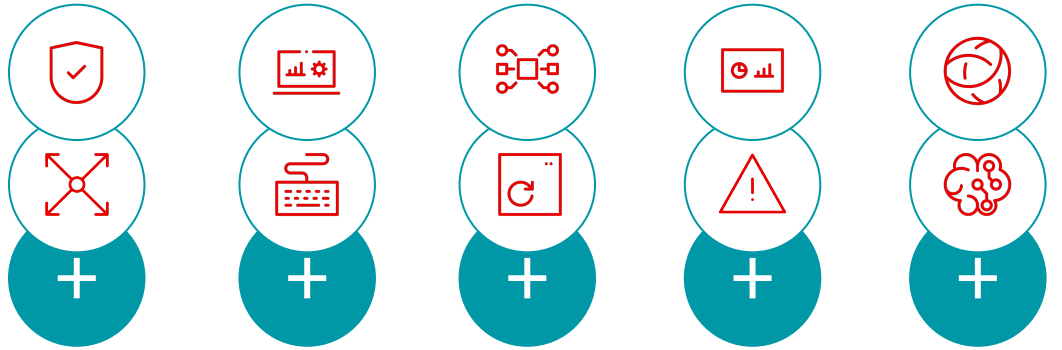
Topics Covered:

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

Many organizations share the same challenge

Differing skills, roles, and responsibilities

Growing number of use cases

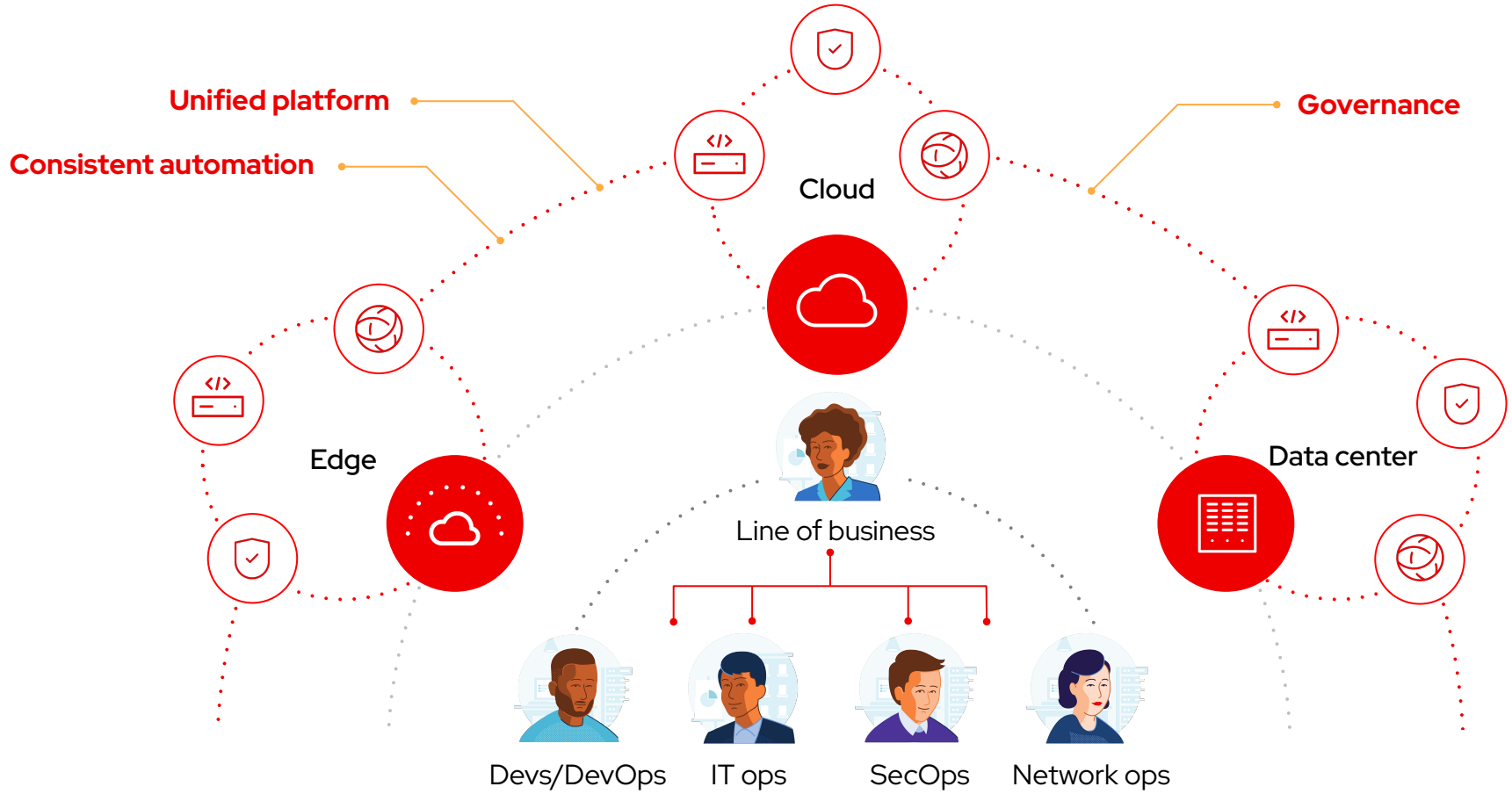


Multiple platforms and locations

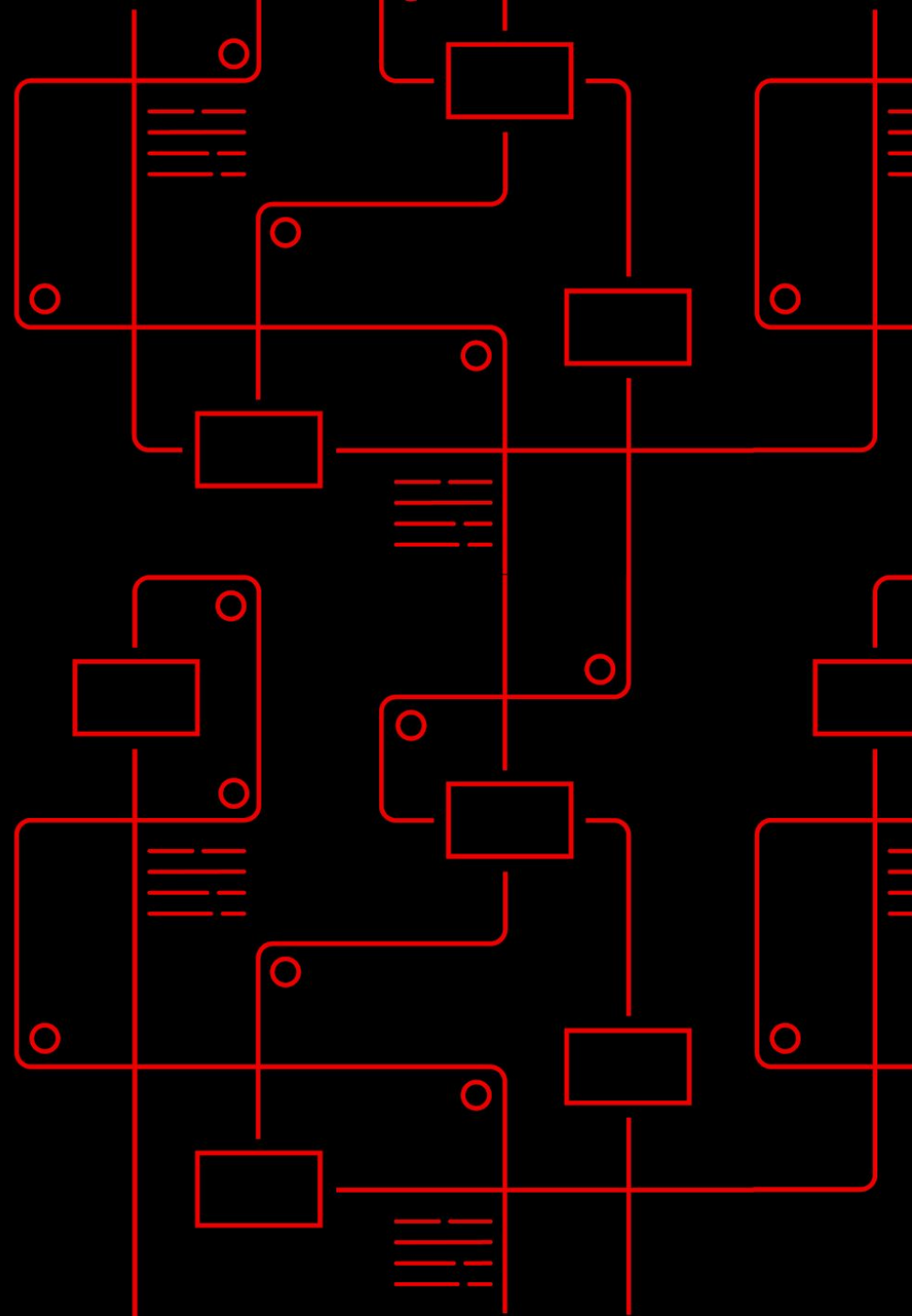
Across multiple domains

Governance

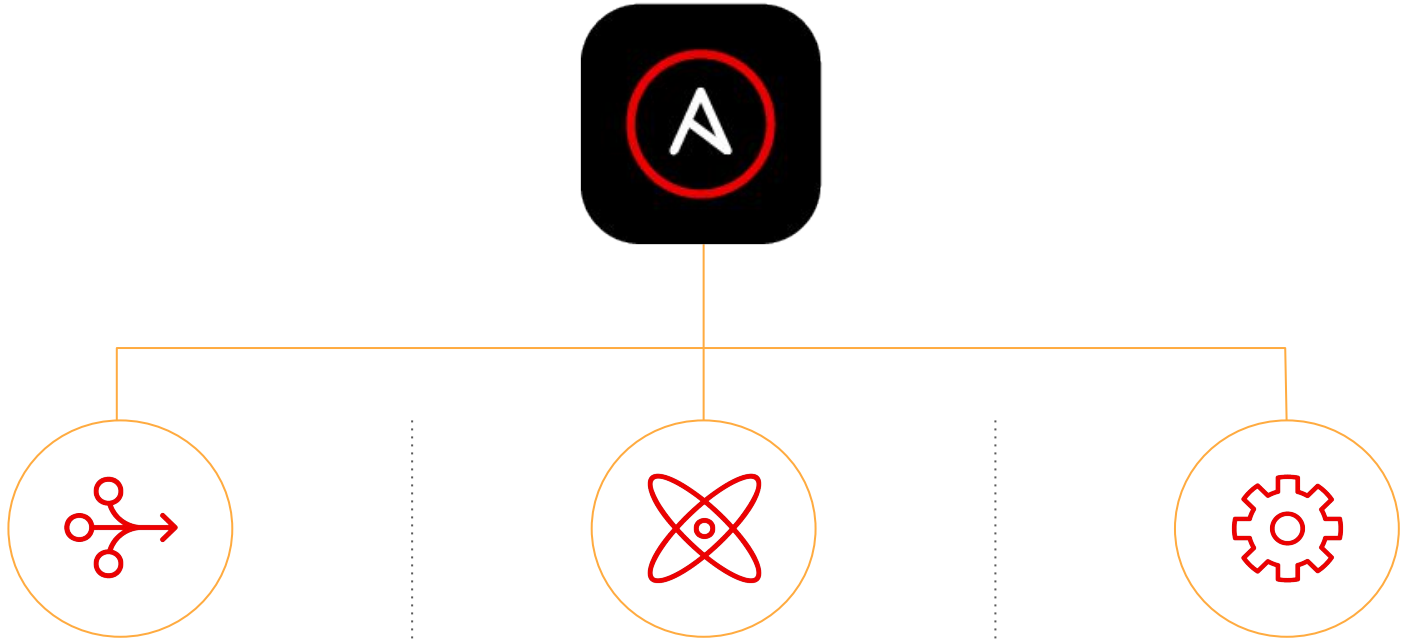
The solution? Break down the silos.



Why Red Hat® Ansible® Automation Platform?



Ansible is the de facto automation language.



Simple

Human-readable language with quick adoption

Powerful

Universal language spanning multiple IT domains

Agentless

Easily integrate with hybrid environments

Supported and certified content you can trust.

130+

Certified Content Collections

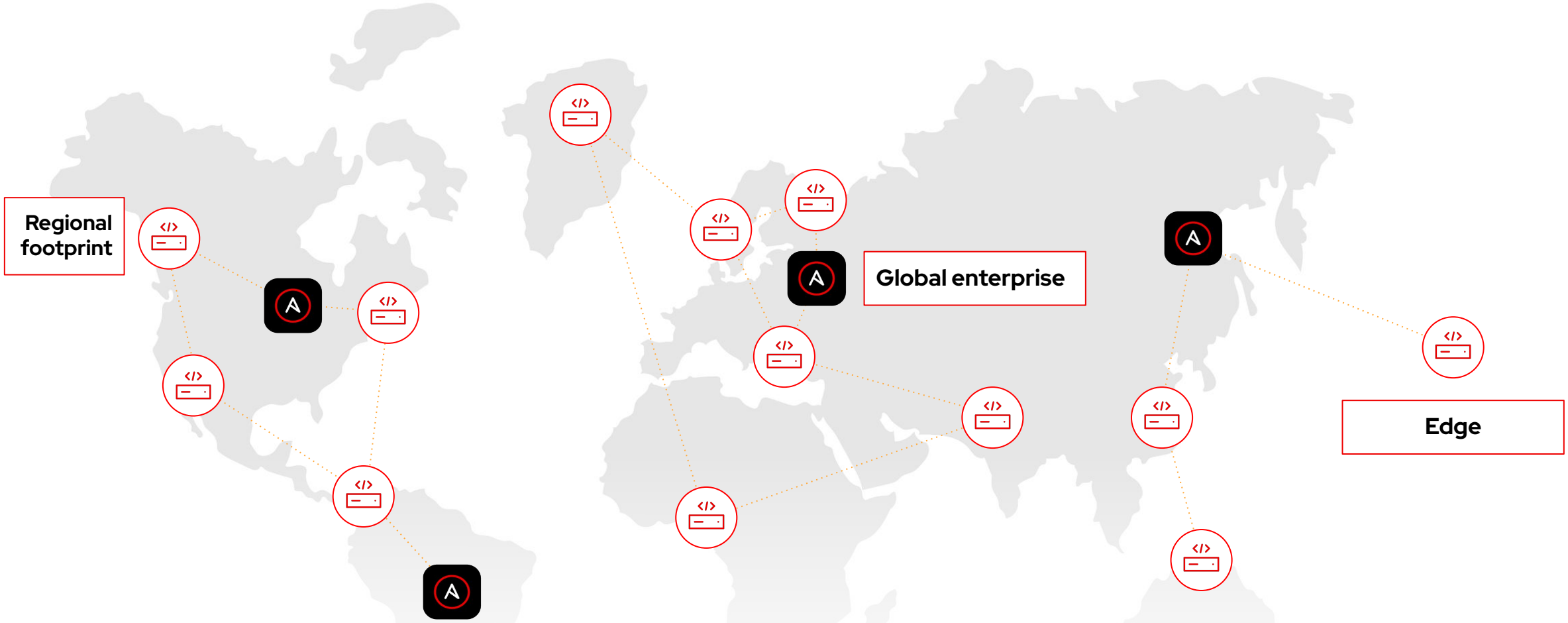
55+

Certified technology partners

Infrastructure | Cloud | Network | Security | Edge

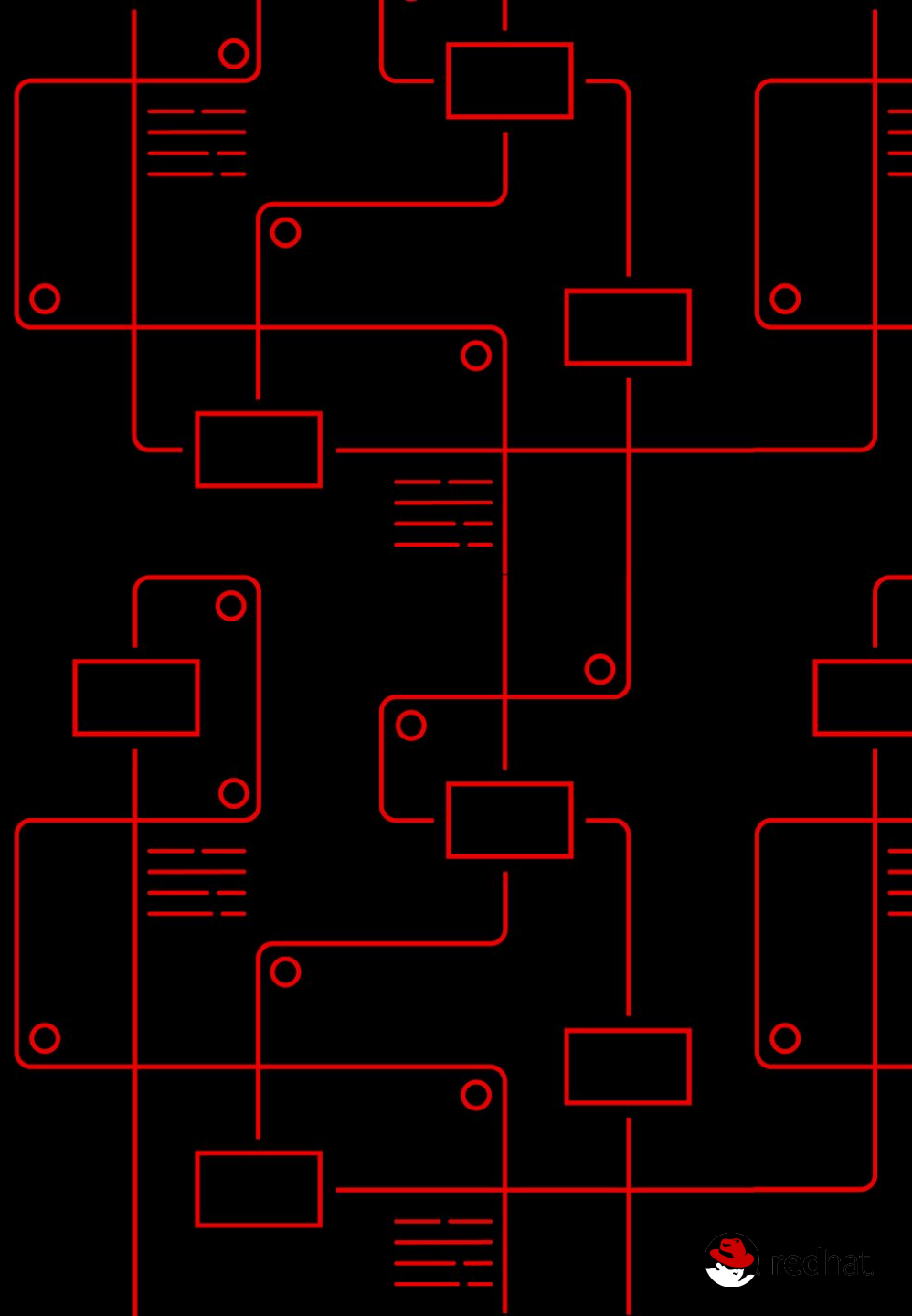
Red Hat Ansible Automation Platform

The flexibility to scale, wherever that may be.



Red Hat
Ansible Automation
Platform

What is Red Hat® Ansible® Automation Platform?



Red Hat
Ansible Automation
Platform

redhat

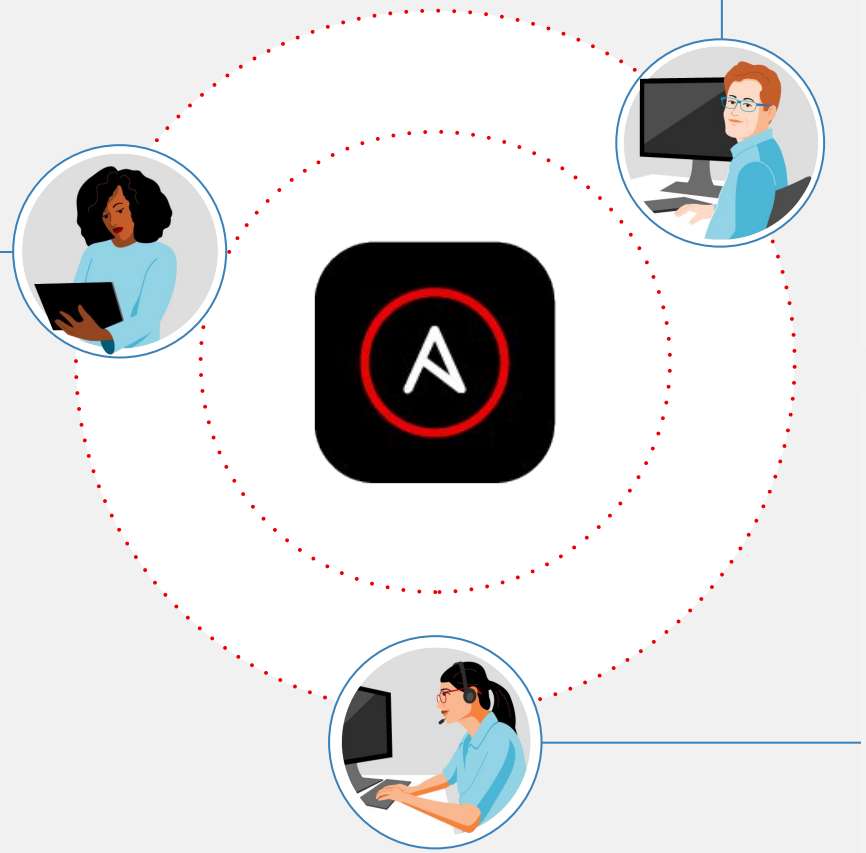
A platform for the entire automation team.

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

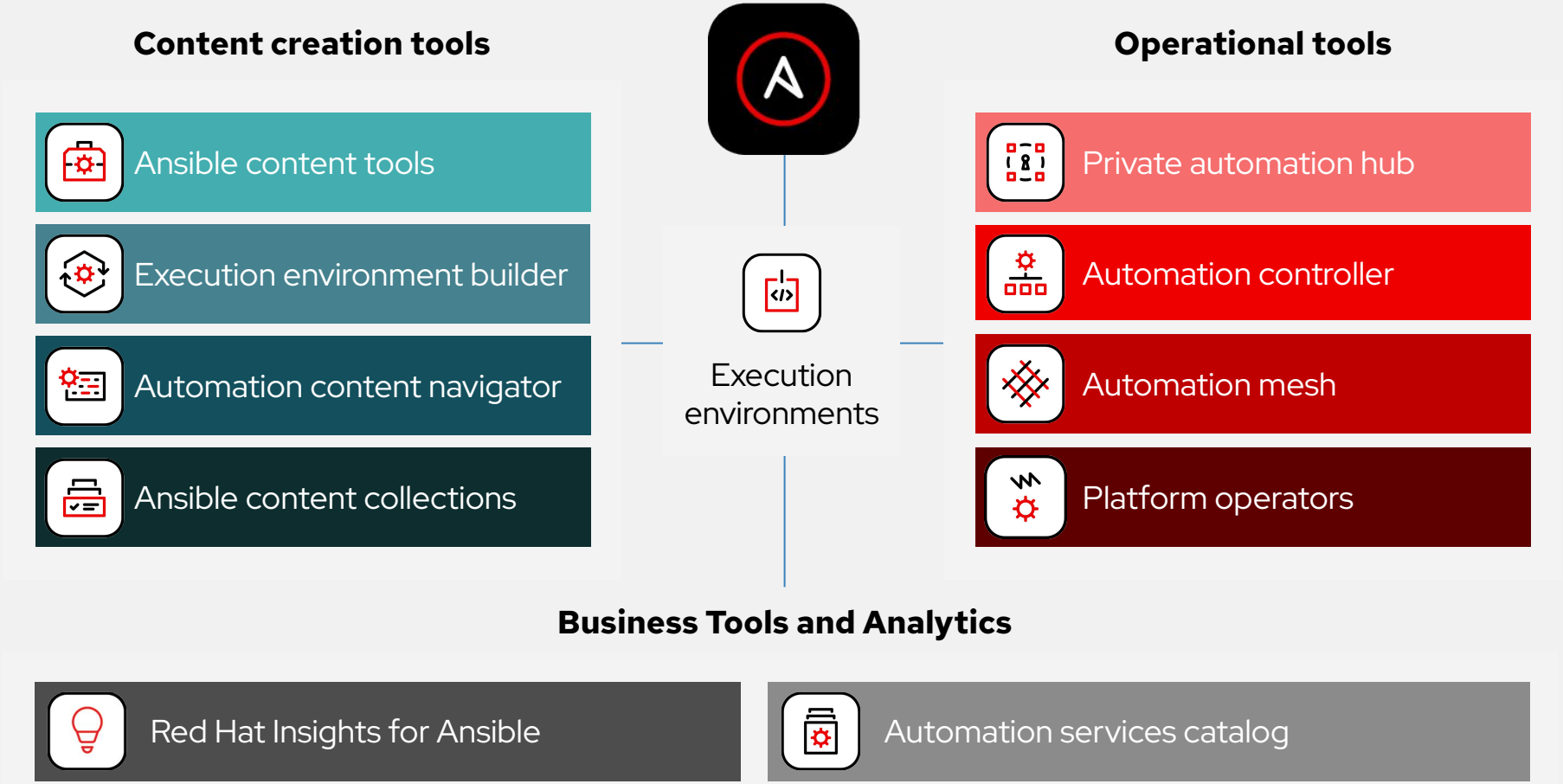
Operations

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

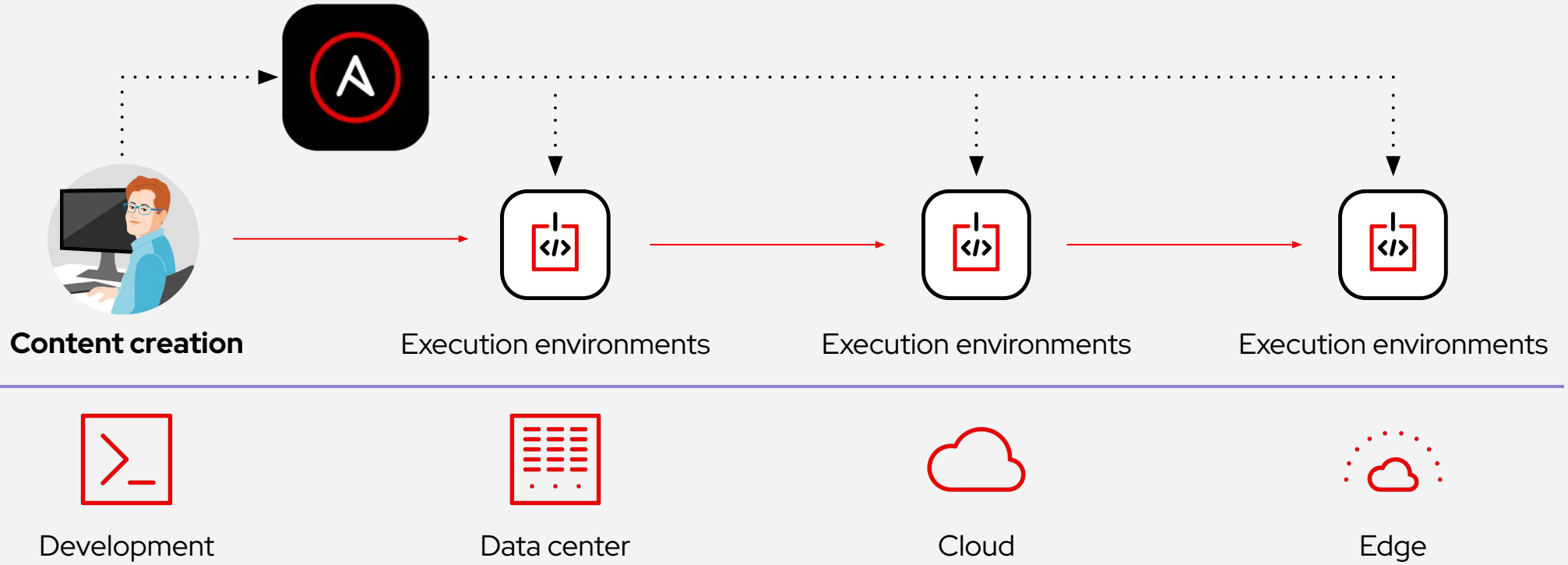
Hosted and private content management solutions

Integrates with your environment

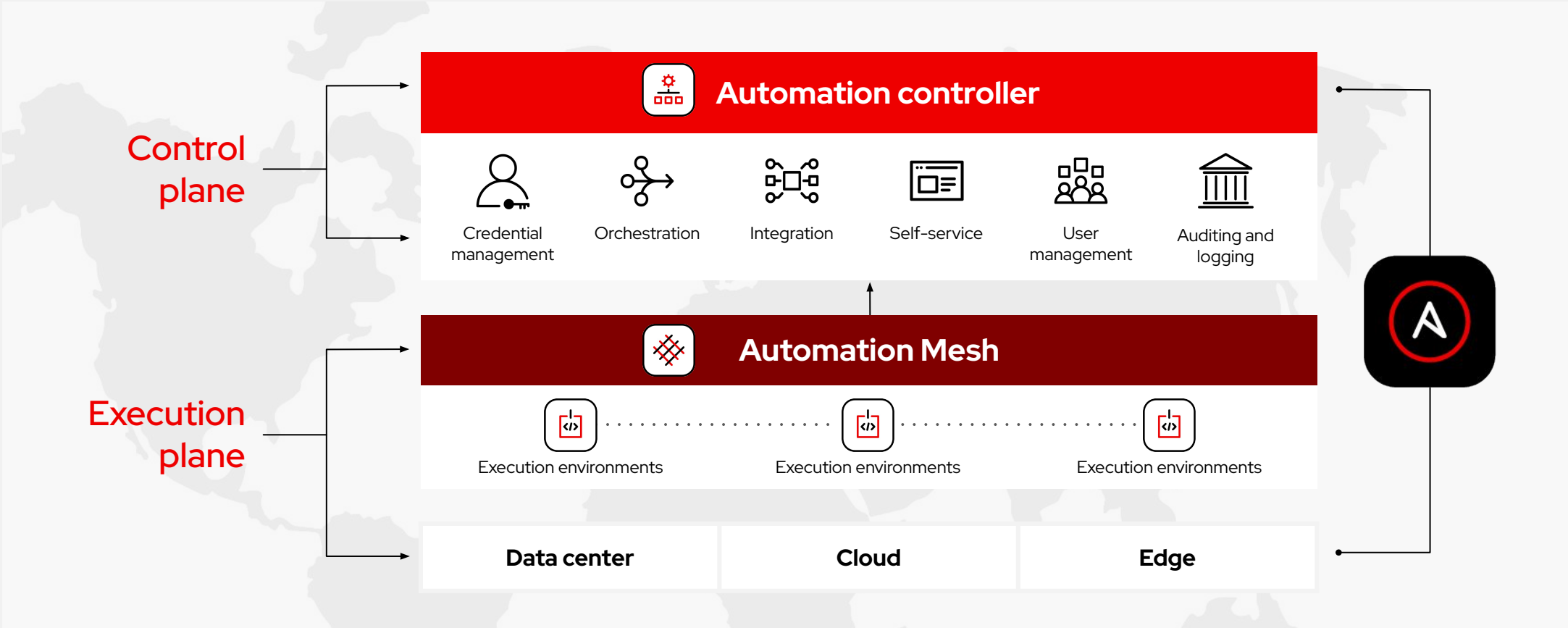
An integrated solution for the enterprise.



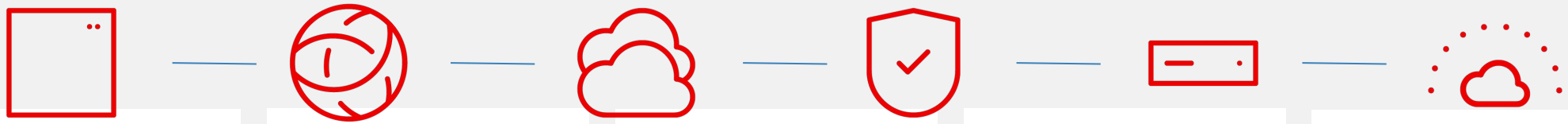
Built for consistency. Portability is reliability.



A distributed architecture built for scale.



The capabilities you need across your IT footprint.



Applications

- DevOps
- CI/CD
- GitOps

Network

- Configuration management
- Infrastructure awareness
- Network validation

Cloud

- Orchestration
- Operationalisation
- Governance

Security

- Investigation enrichment
- Threat hunting
- Incident response

Infrastructure

- Deployment
- Provisioning
- Management

Edge

- Extend security
- Scalability
- Interoperability

Ansible Automation Platform hosting options

Red Hat Enterprise Linux

Red Hat Enterprise Linux 8.3+ x86_64 (physical, virtual)

Red Hat OpenShift

Red Hat OpenShift via dedicated Ansible Automation Platform operator (physical, virtual)

Azure

On Microsoft Azure marketplace

Self Managed (on-premises or cloud)

Managed by Red Hat (cloud)*

** Red Hat Ansible Automation Platform on Azure is now available to all customers in North America and EMEA, with global availability coming soon.*

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
Regedits
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

Red Hat
Ansible Automation
Platform

Cloud

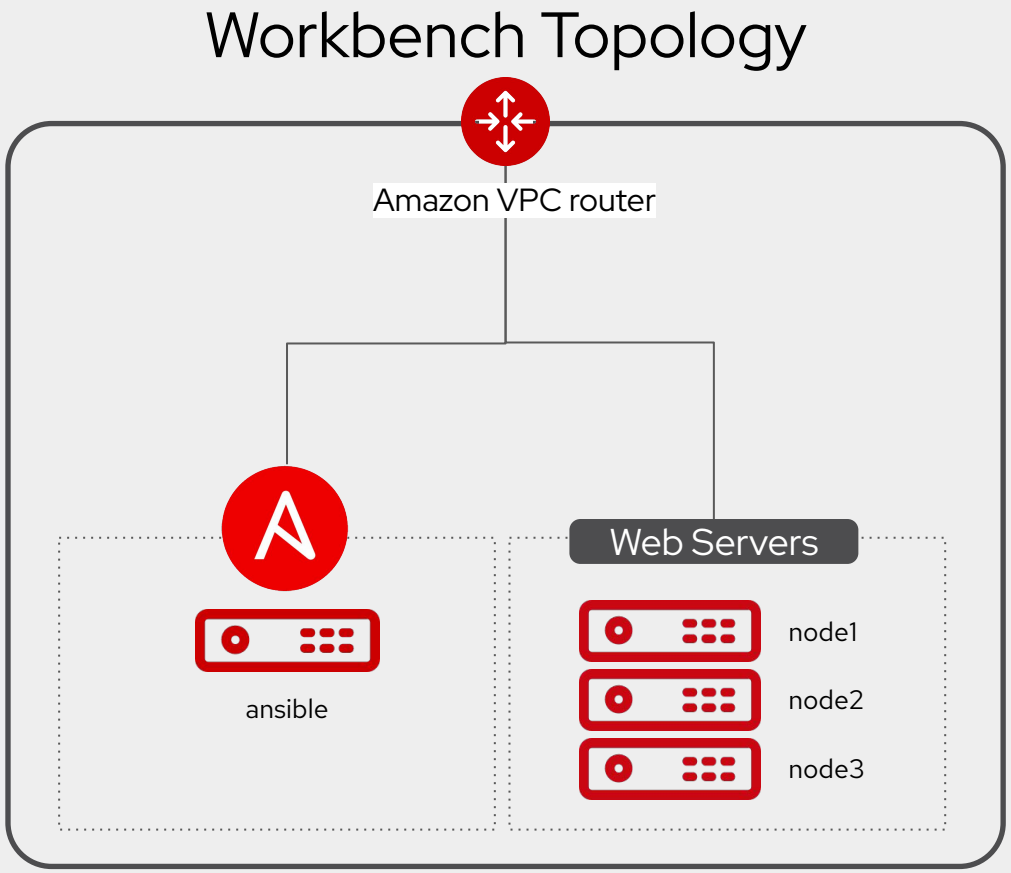
Topics Covered:

- Understanding the Ansible Infrastructure
- Check the prerequisites

Red Hat

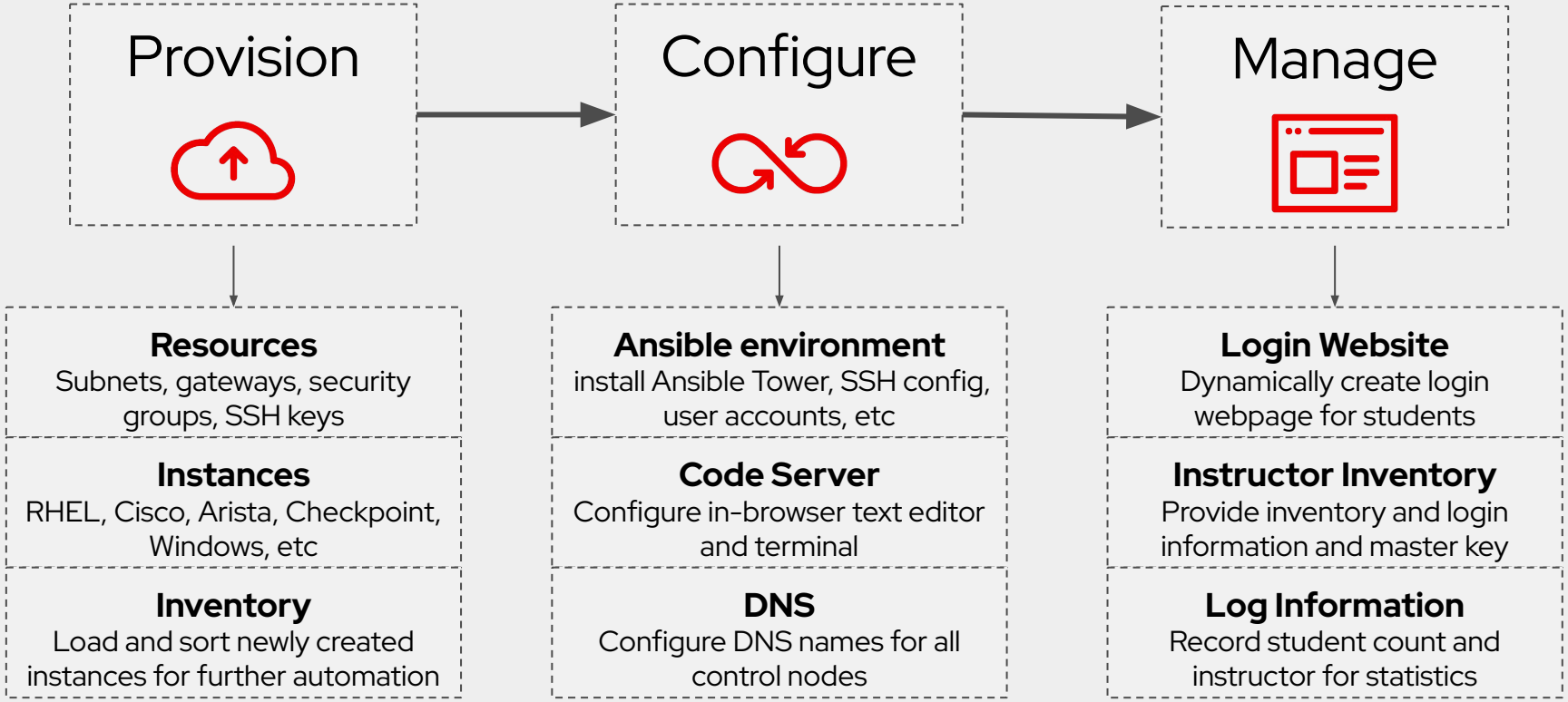
The lab environment today

- **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.



Red Hat

How does it work?



Red Hat

Red Hat
Ansible Automation
Platform

Exercise 1

Topics Covered:

- Understanding the Ansible Infrastructure
- Check the prerequisites

Red Hat

Ansible Core (ansible-core)

What is it?

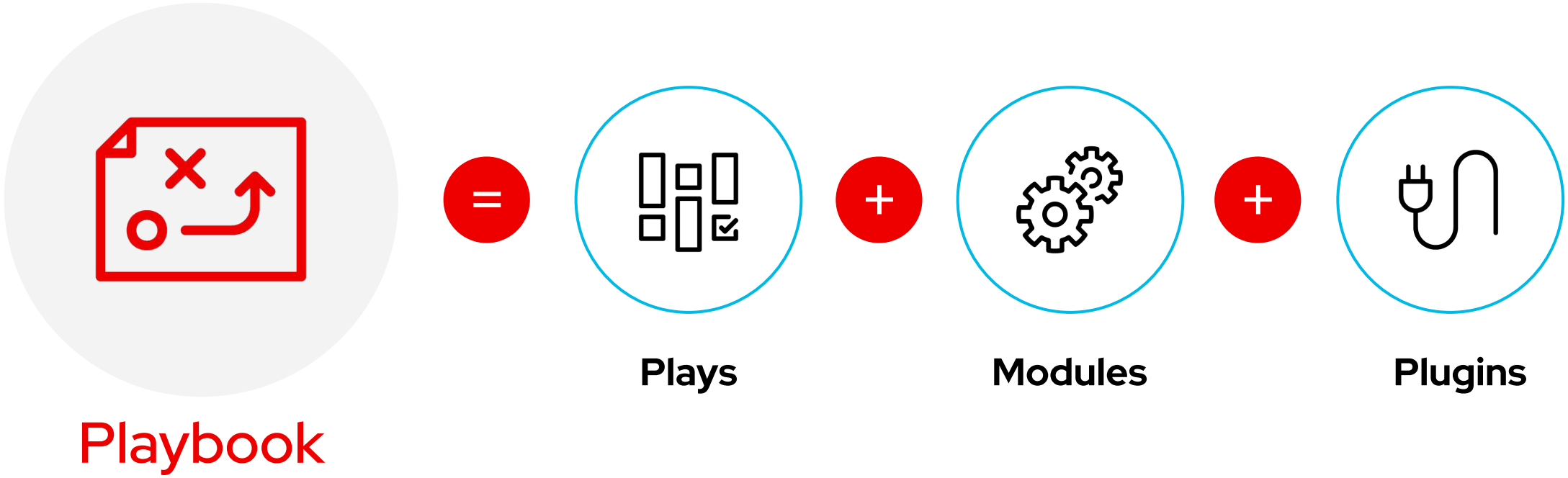
- The main building block for Ansible
- Simple YAML syntax to develop Ansible Playbooks
- Provides CLI tools to develop, test and run playbooks
- Pluggable architecture that allows extensions through Content Collections

```
---
- 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?



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?

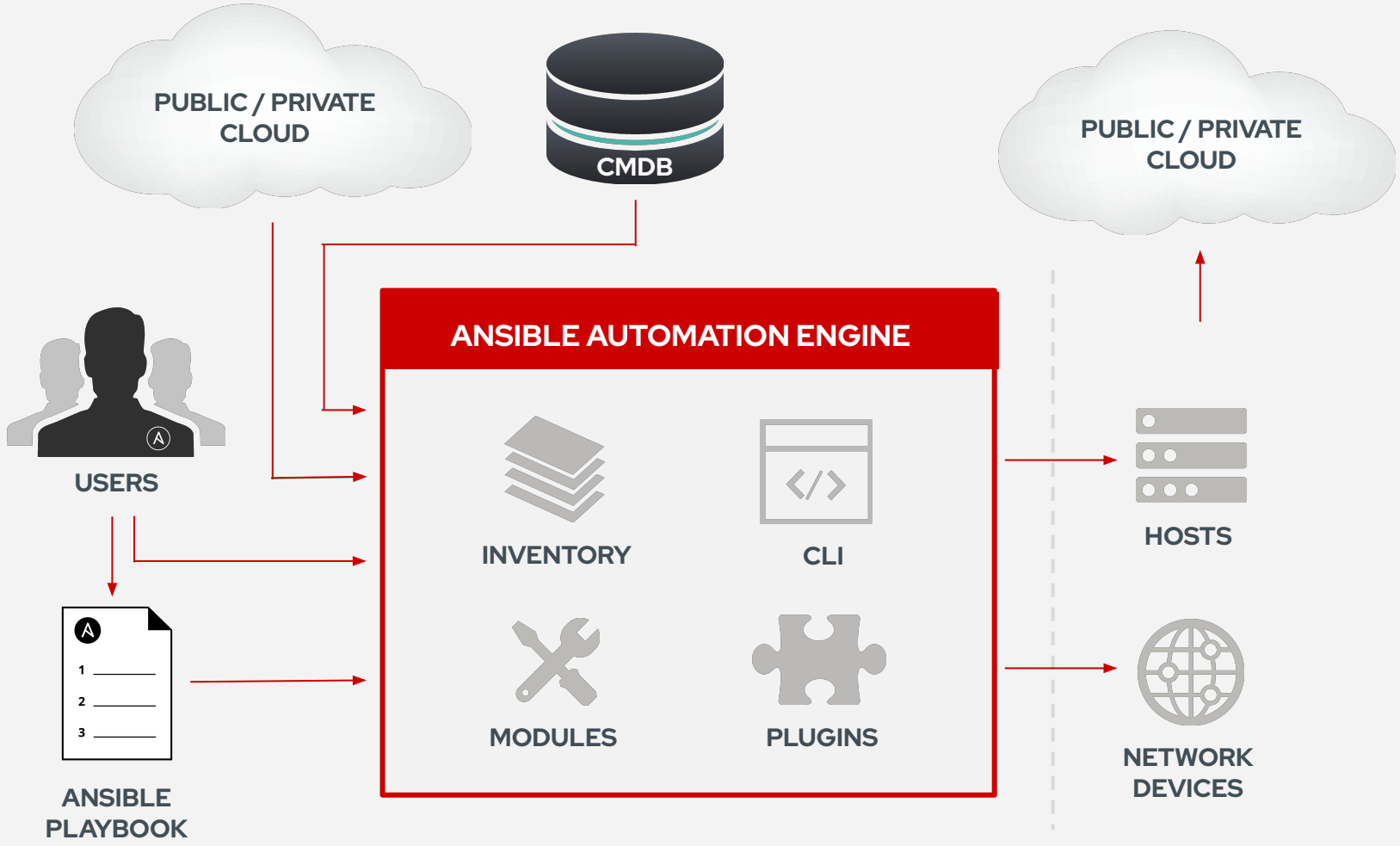
- Plugins are pieces of code that augment Ansible’s core functionality
- Ansible uses a plugin architecture to enable a rich, flexible, and expandable feature set

Example become plugin:

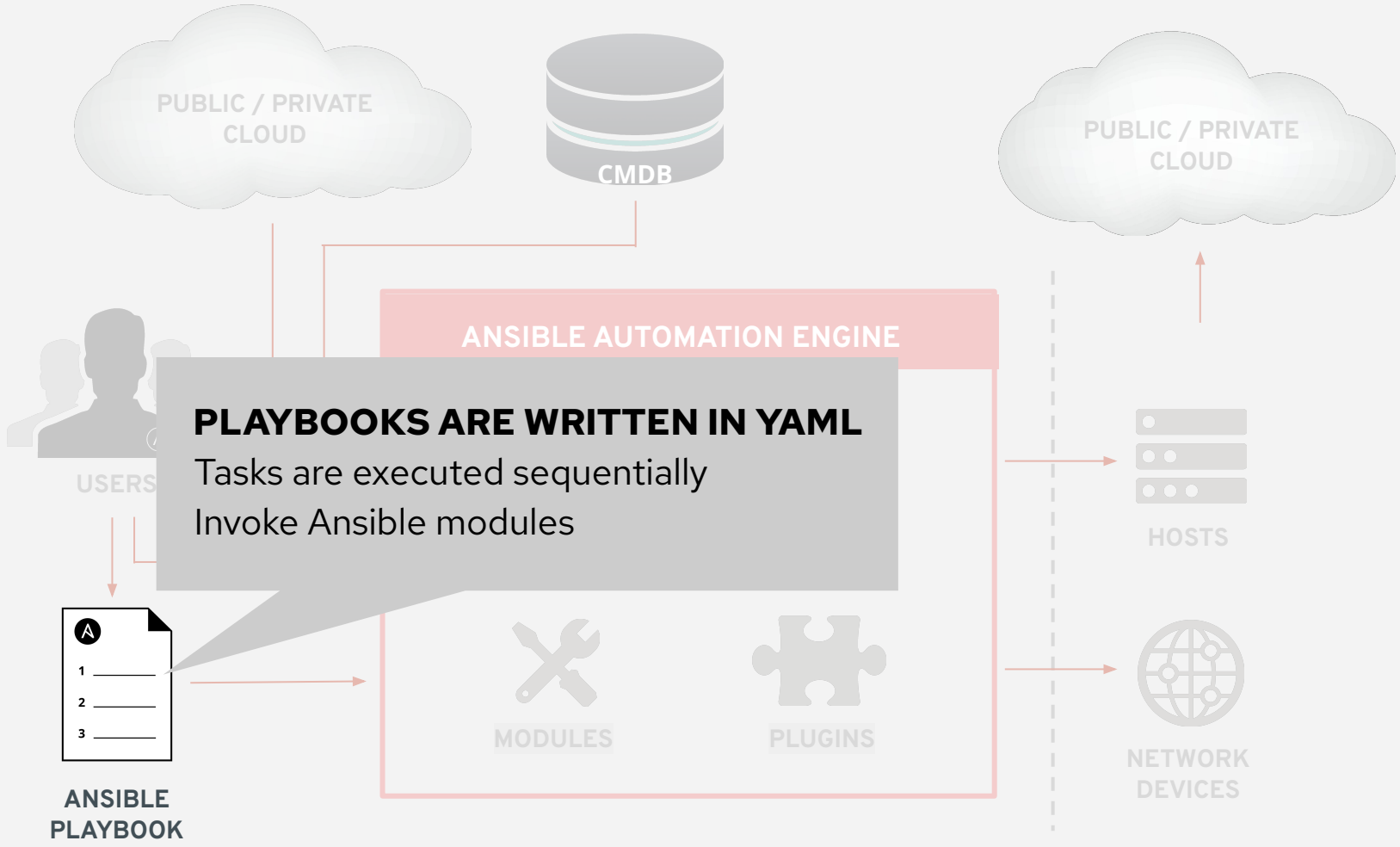
```
---
- name: Install and start apache
  hosts: web
  become: true
```

Example filter plugins:

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


PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE CLOUD
ANSIBLE AUTOMATION ENGINE
USERS
INVENTORY
CLI
HOSTS
MODULES
PLUGINS
NETWORK DEVICES
1
2
3
ANSIBLE PLAYBOOK

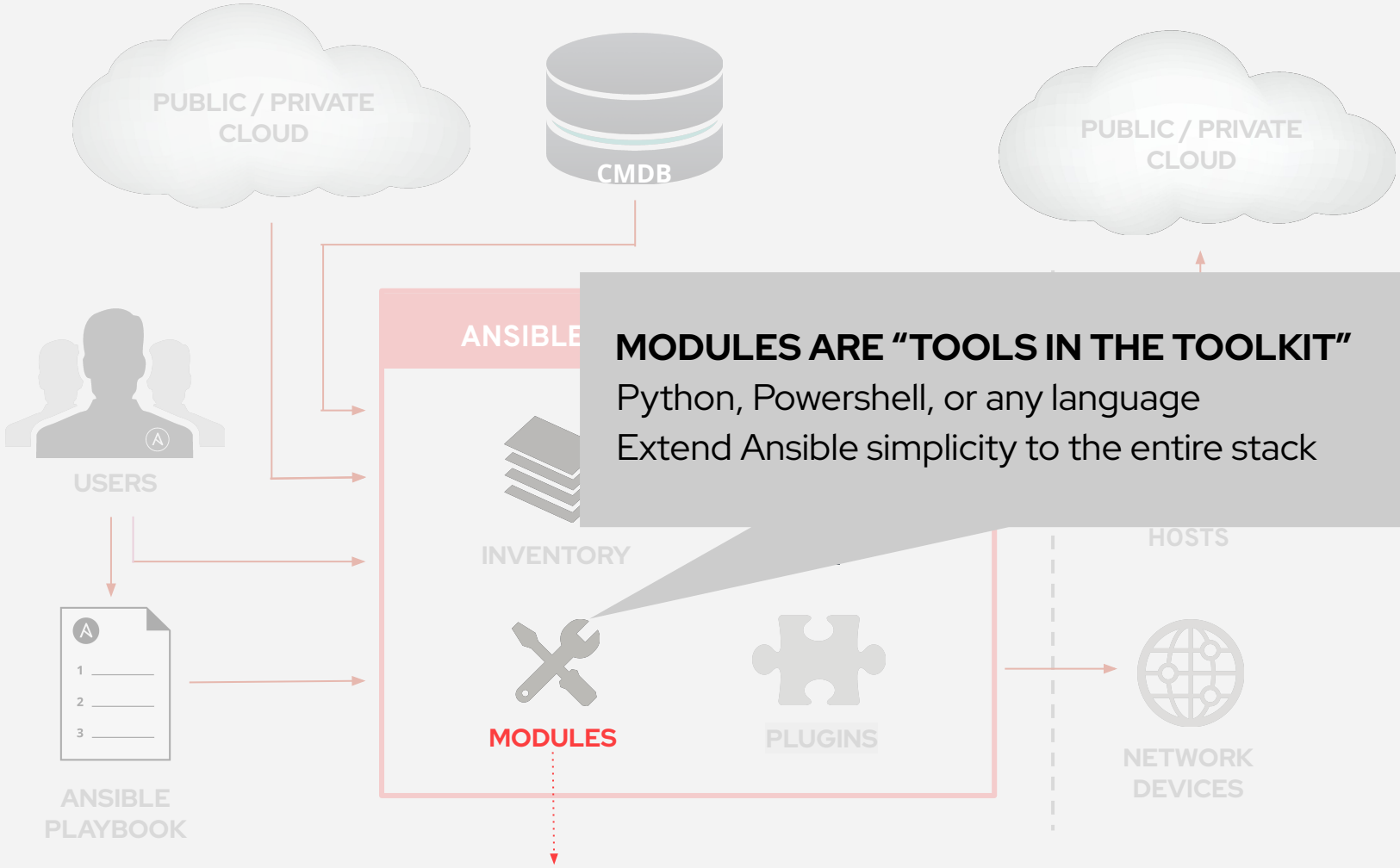
Red Hat


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

```
---
- 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
```

Red Hat



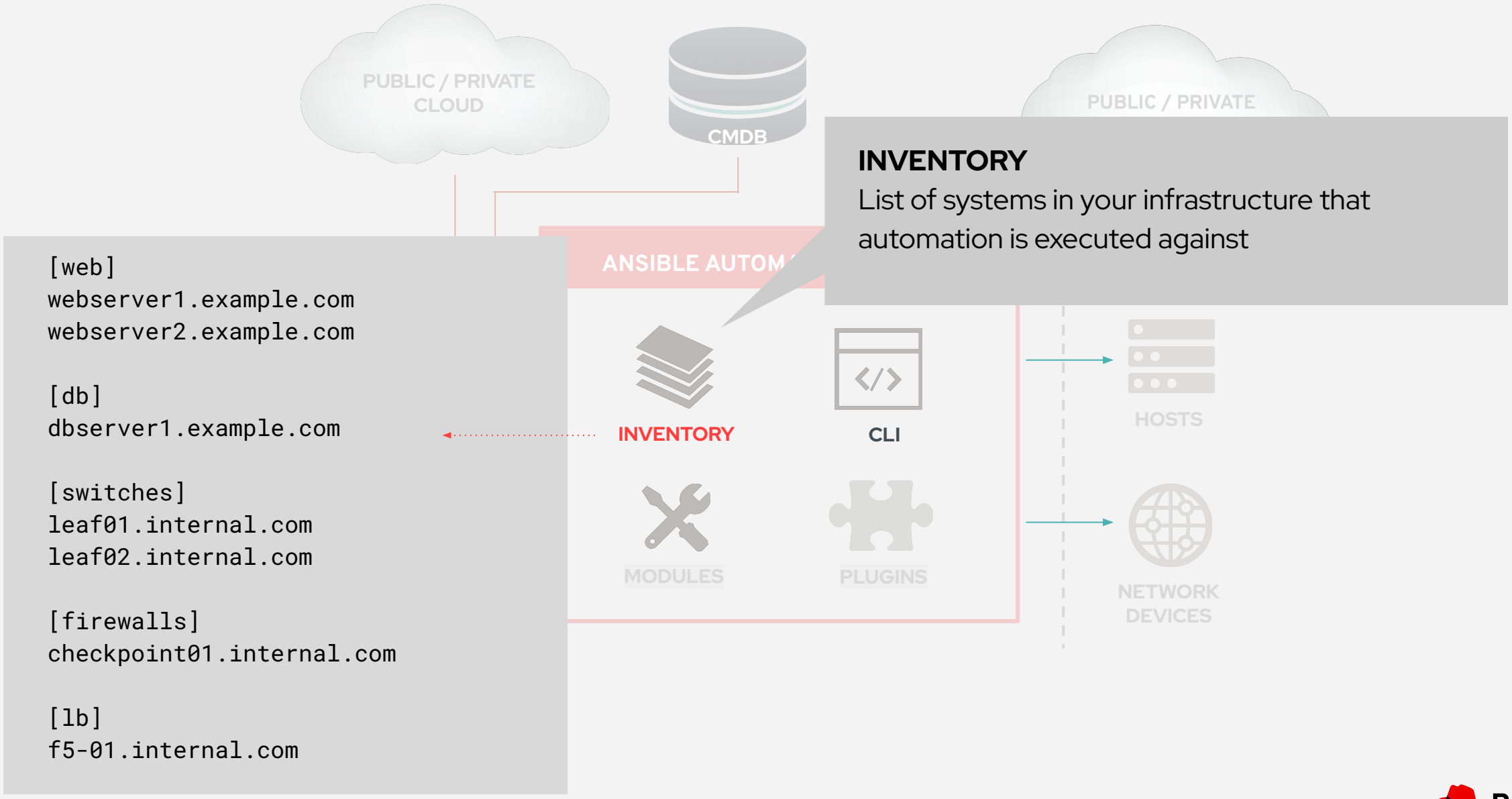
MODULES ARE "TOOLS IN THE TOOLKIT"

Python, Powershell, or any language

Extend Ansible simplicity to the entire stack

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

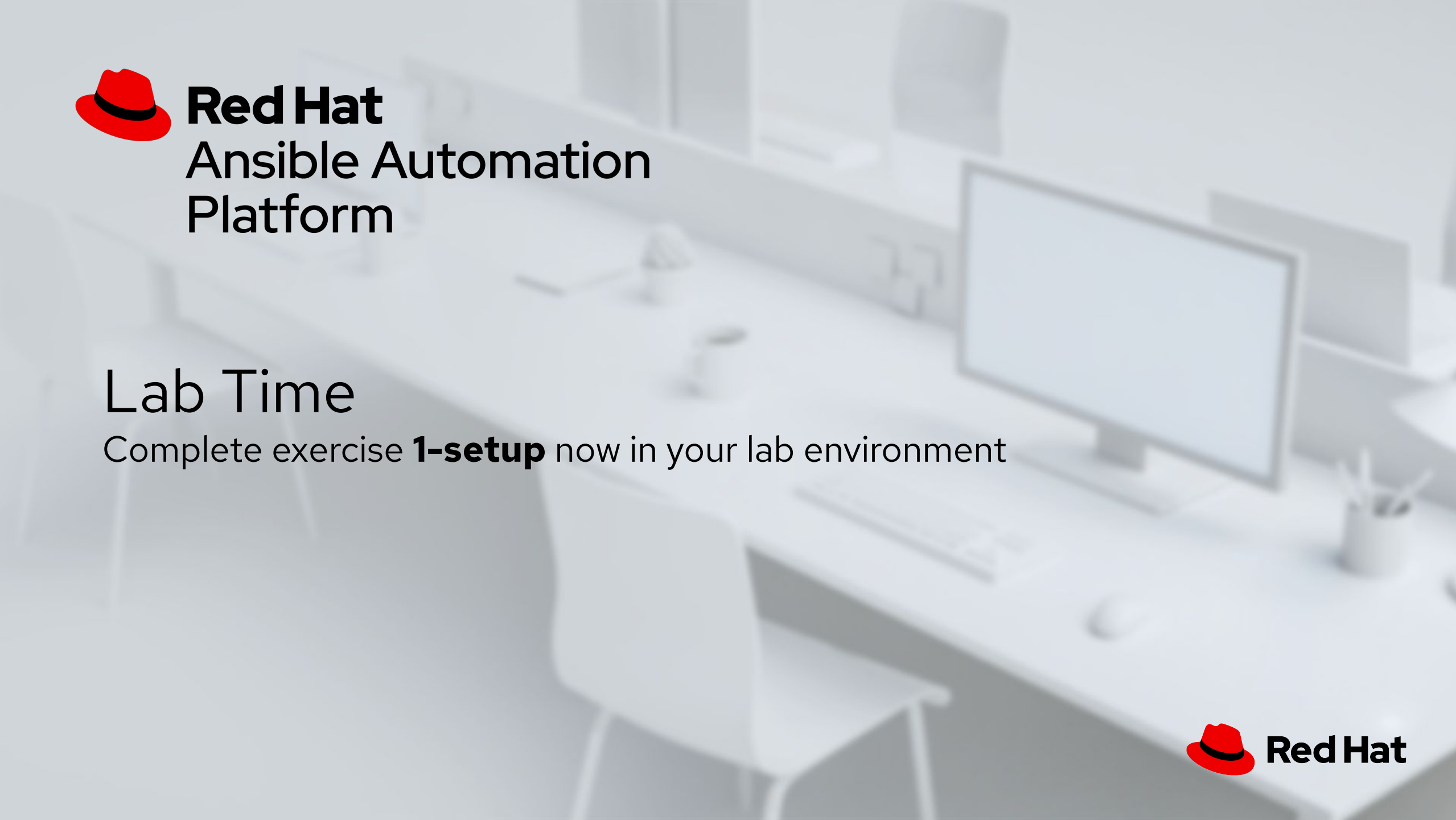
AUTOMATE EVERYTHING LINUX

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


Red Hat
Ansible Automation
Platform
Lab Time
Complete exercise **1-setup** now in your lab environment
Red Hat

Exercise 2

Topics Covered:

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

Red Hat
Ansible Automation Platform

Red Hat

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

Red Hat

Understanding Inventory - Basic

```
node1
node2
node3
ansible
10.20.30.40
```

Red Hat

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
```

Red Hat

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
```

Red Hat

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
```


Red Hat

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"
}
```


Red Hat

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

more 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

Red Hat

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
```


Red Hat

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


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?

Red Hat

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}}"
```

Red Hat

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

Red Hat

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",
...
```


Red Hat

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
```

Red Hat

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
```


Red Hat

Ansible Inventory - Managing Variables In Files

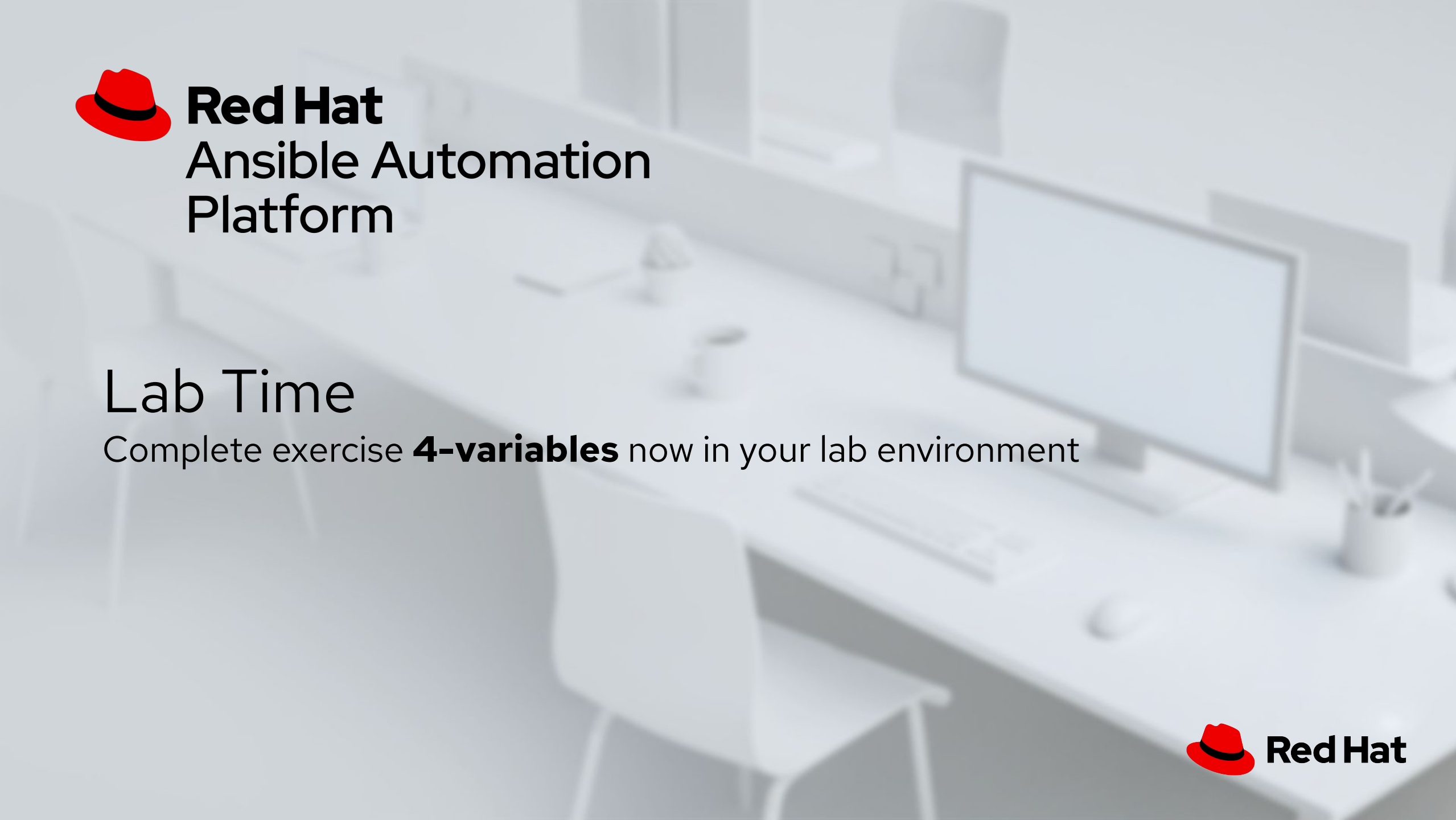
```
├──
deploy_index_html.yml
├── files
│   ├── dev_web.html
│   └── prod_web.html
├── group_vars
│   └── web.yml
└── host_vars
    └── node2.yml
```

```
$ 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
```


Red Hat


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

Red Hat

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.

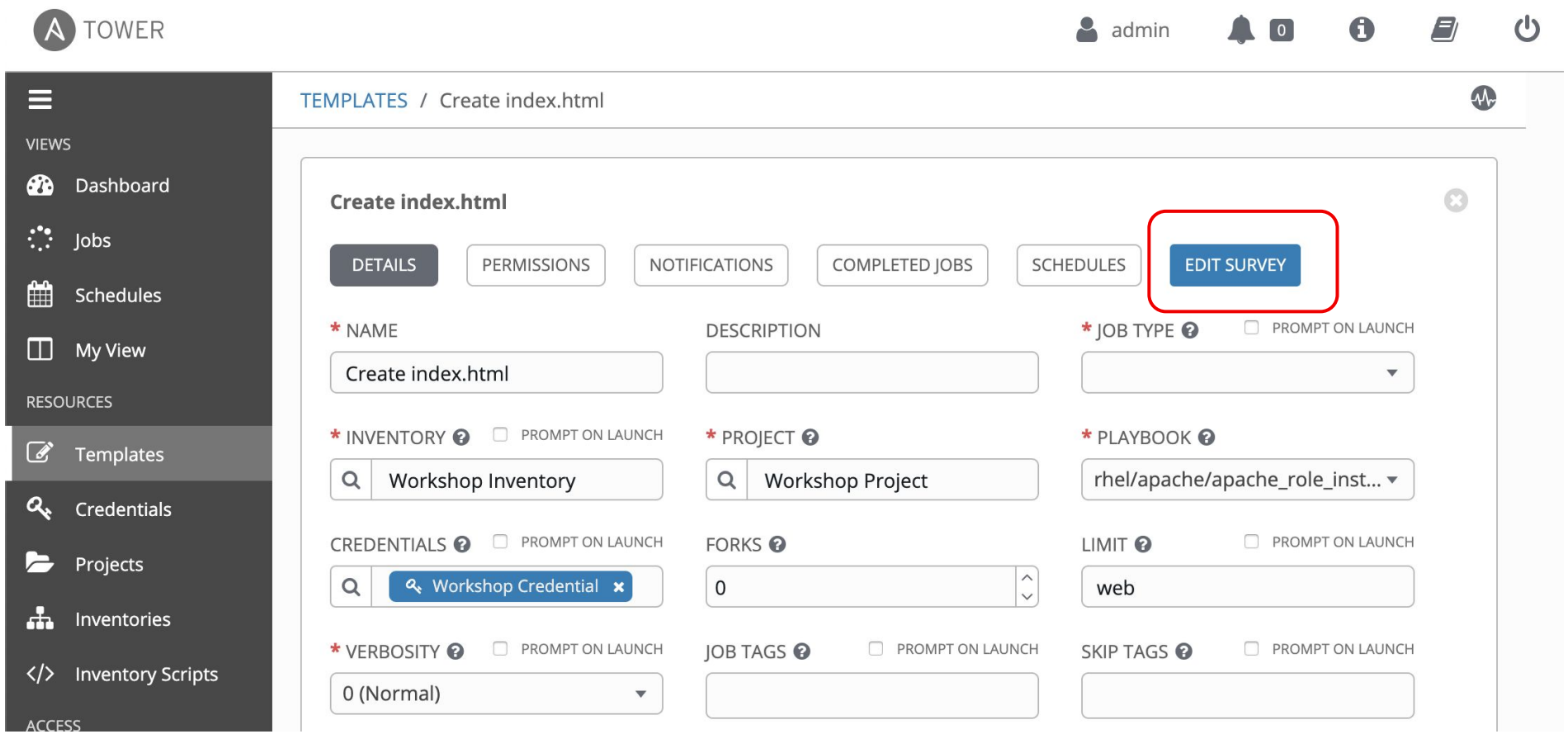
An Ansible Tower survey is a simple question-and-answer form that allows users to customize their job runs. Combine that with Tower's role-based access control, and you can build simple, easy self-service for your users.

Red Hat

Creating a Survey (1/2)

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

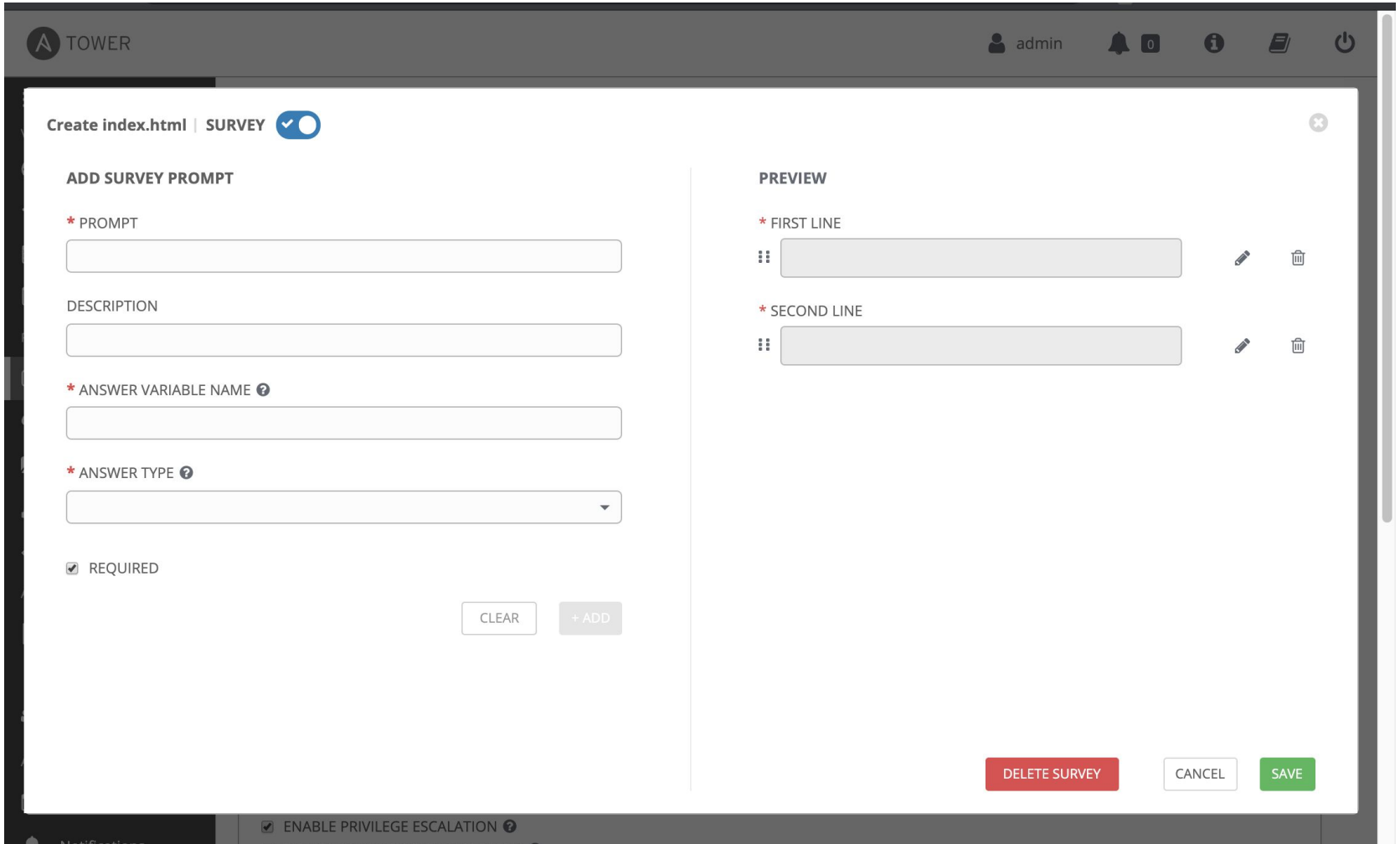
Click the button to open the Add Survey window.



Red Hat

Creating a Survey (2/2)

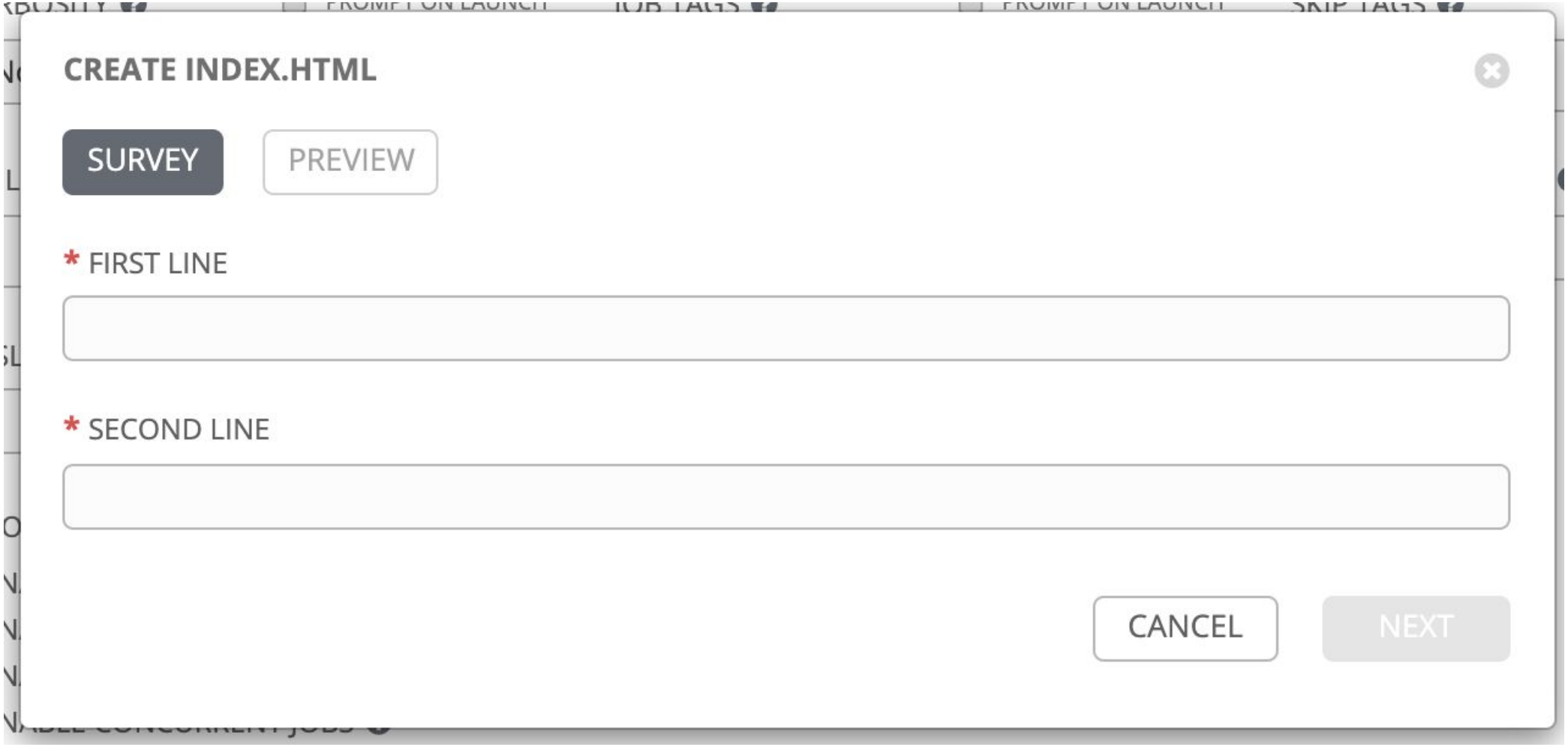
The Add Survey window allows the Job Template to prompt users for one or more questions. The answers provided become variables for use in the Ansible Playbook.



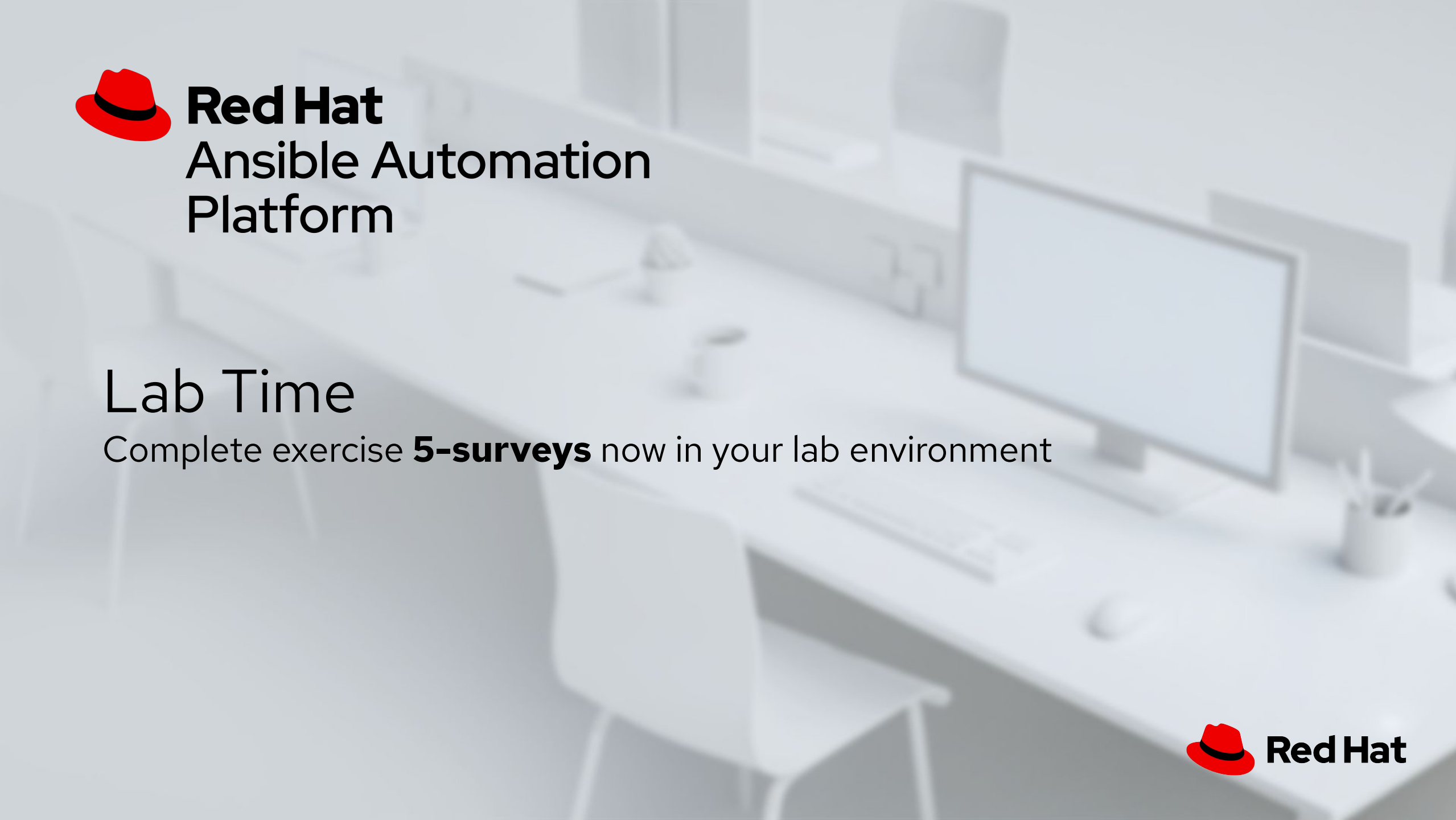
Red Hat

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


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

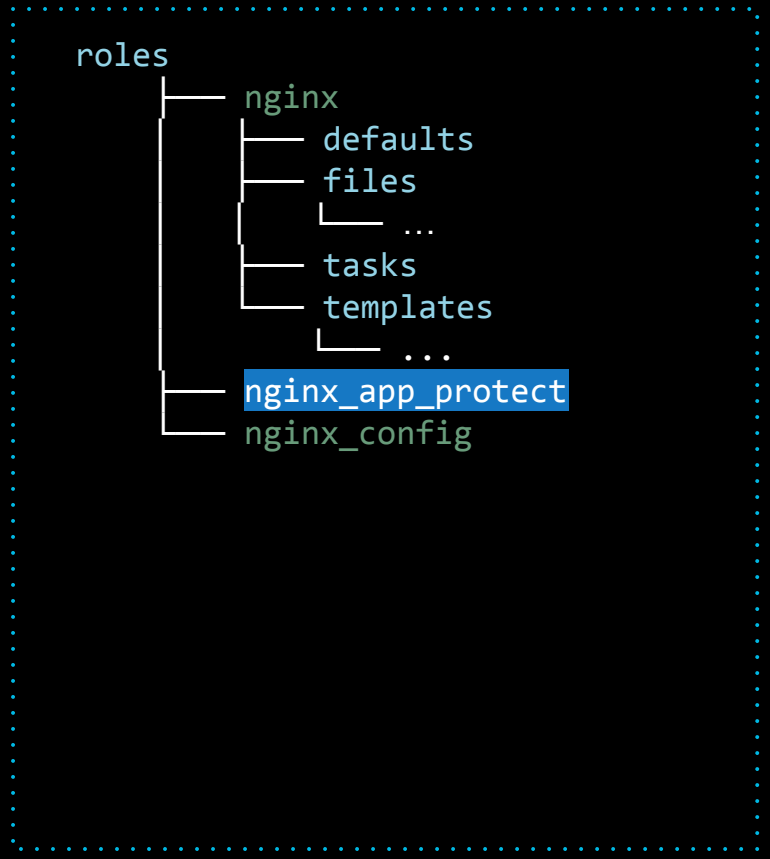
Red Hat

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
```



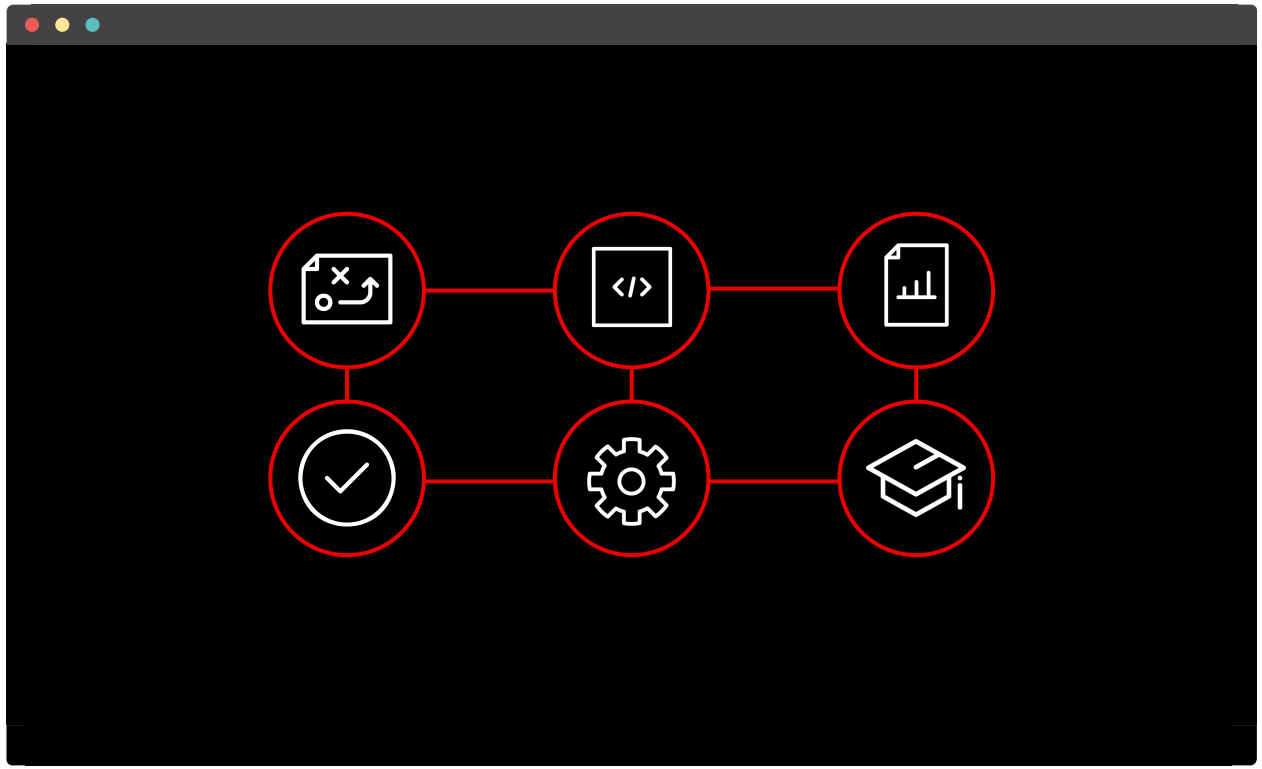
deploy-nginx.yml

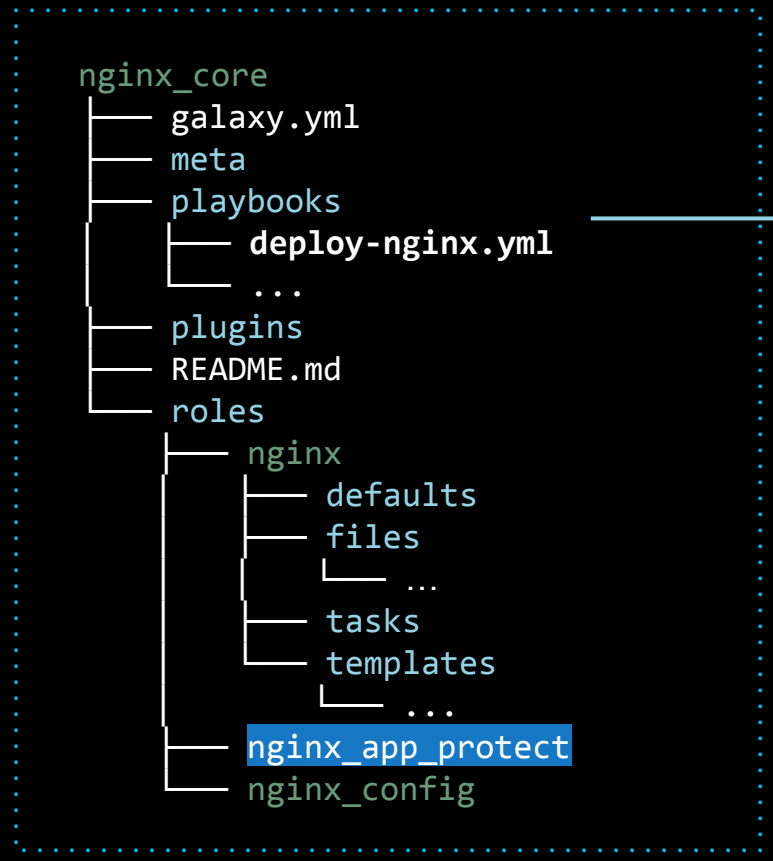
```
---
- name: Install NGINX Plus
  hosts: all
  tasks:

    - 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





deploy-nginx.yml

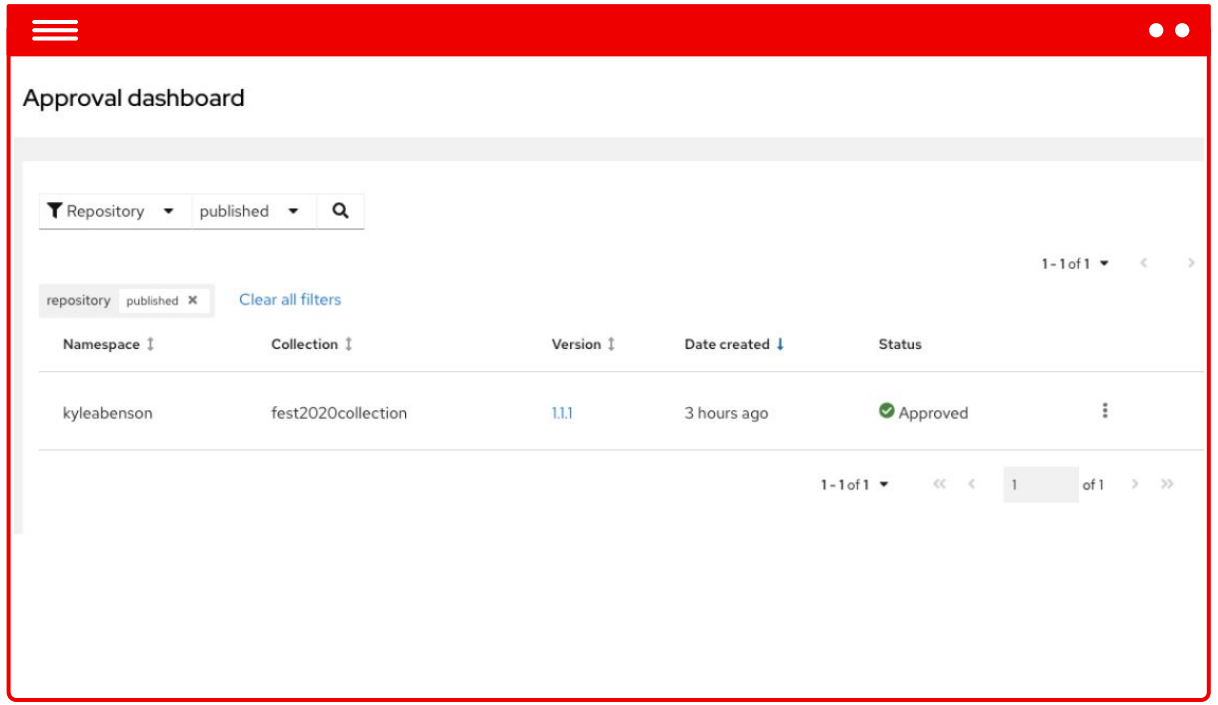
```yaml
---
- name: Install NGINX Plus
  hosts: all
  tasks:
    - name: Install NGINX
      include_role:
        name: nginxinc.nginx
      vars:
        nginx_type: plus

    - 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

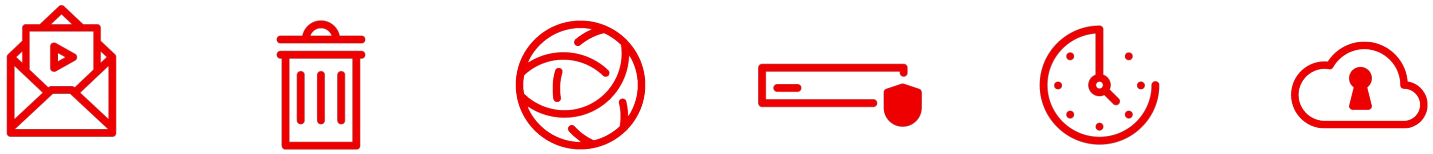


Red Hat
Ansible Automation
Platform

Linux System Roles

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

Examples

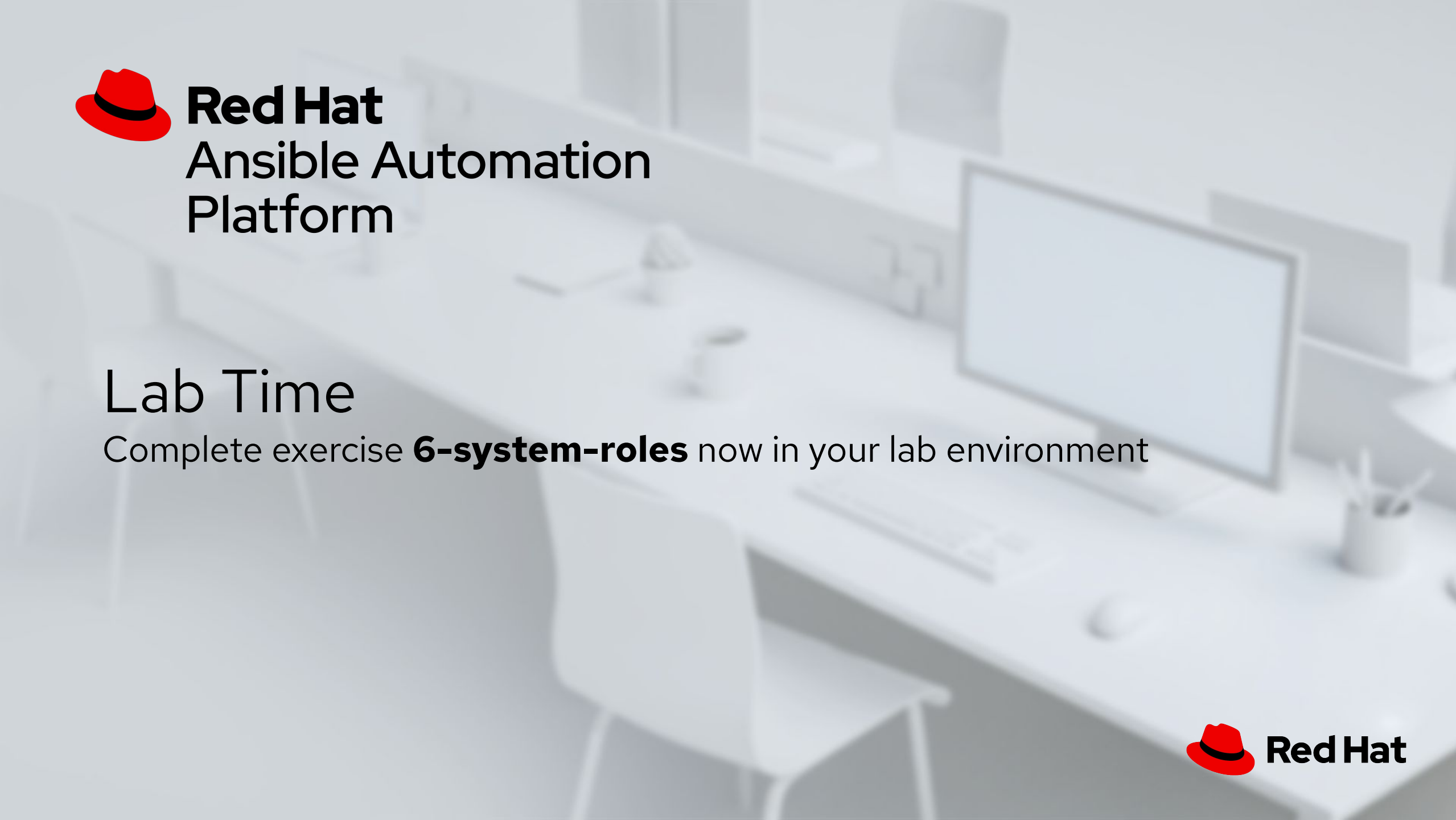


Red Hat

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
```

Red Hat


Red Hat
Ansible Automation
Platform
Lab Time
Complete exercise **6-system-roles** now in your lab environment
Red Hat

1 more Exercise

Topics Covered:

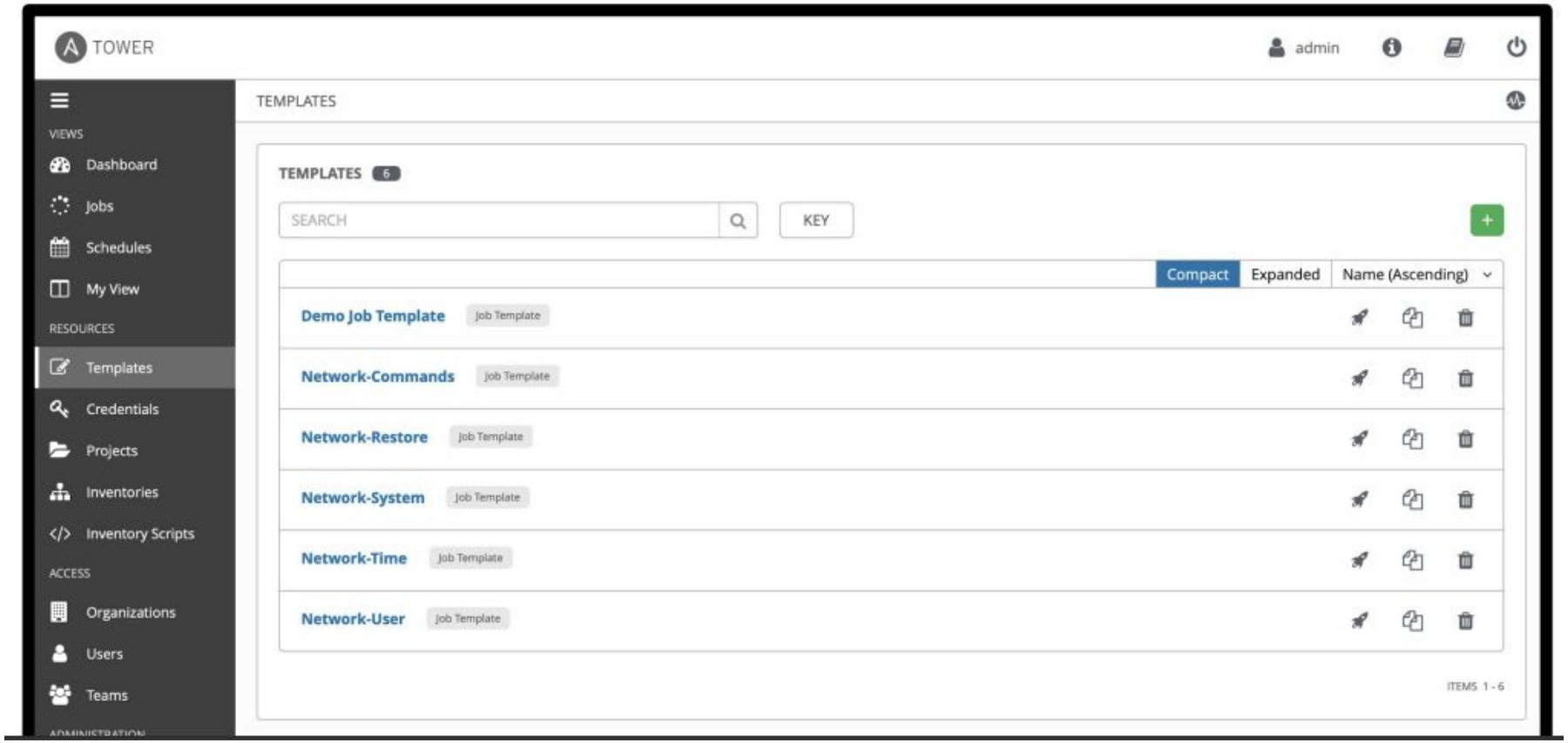
- Workflows

Red Hat
Ansible Automation Platform

Red Hat

Workflows

Workflows can be found alongside Job Templates by clicking the Templates button under the RESOURCES section on the left menu

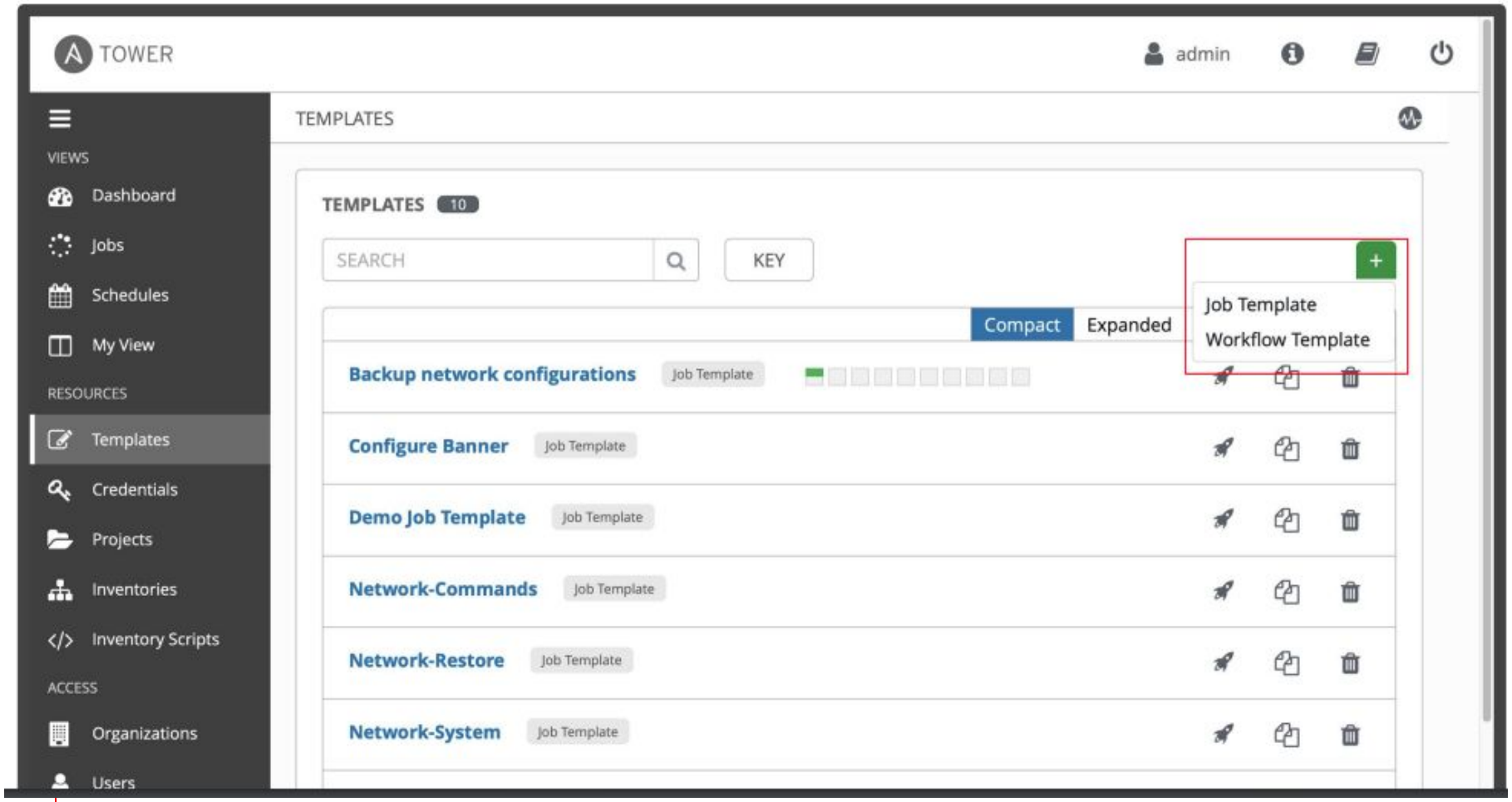


Red Hat

Adding a new Workflow Template

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

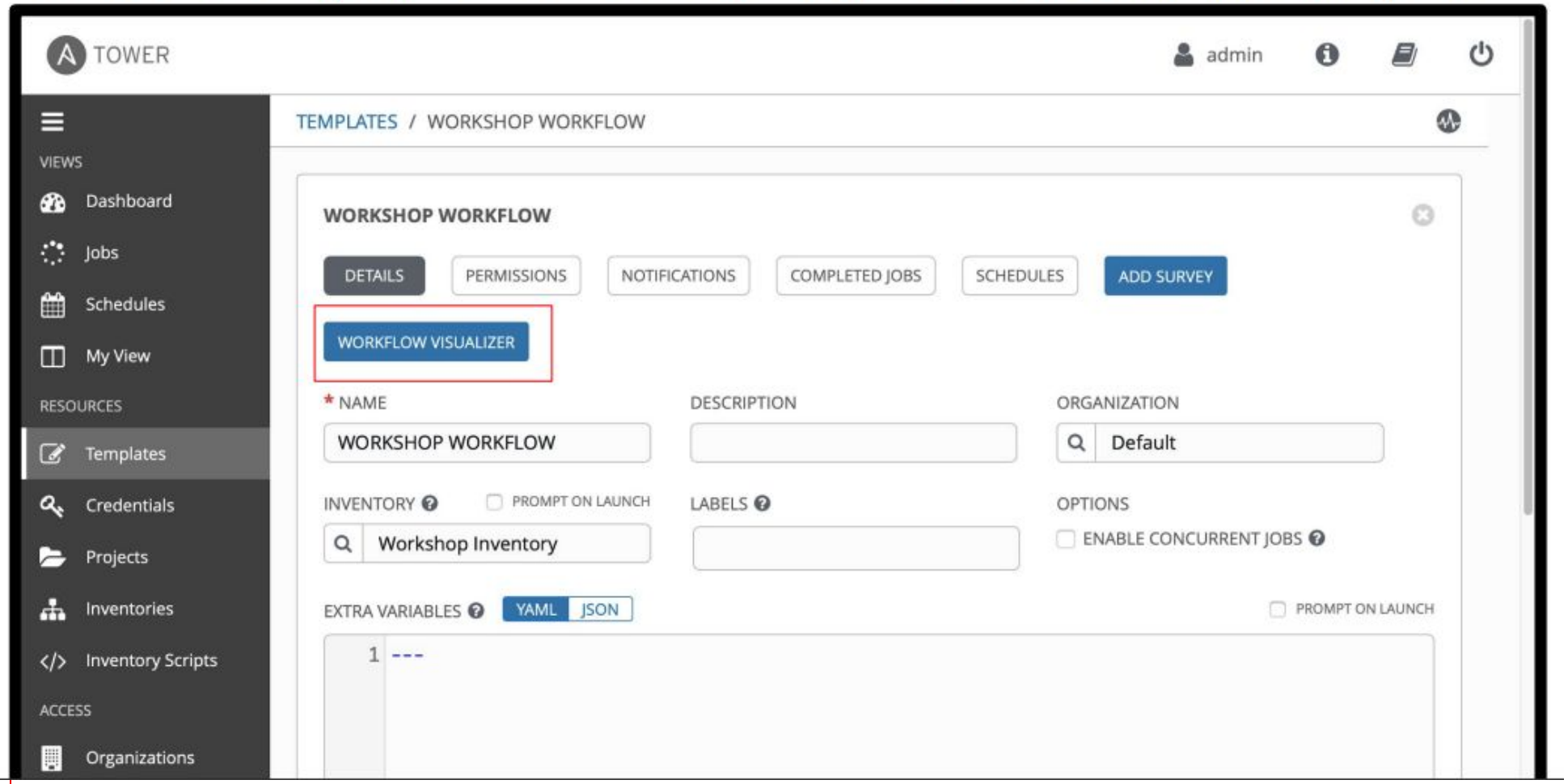
This time select the **Workflow Template**



Red Hat

Creating the workflow

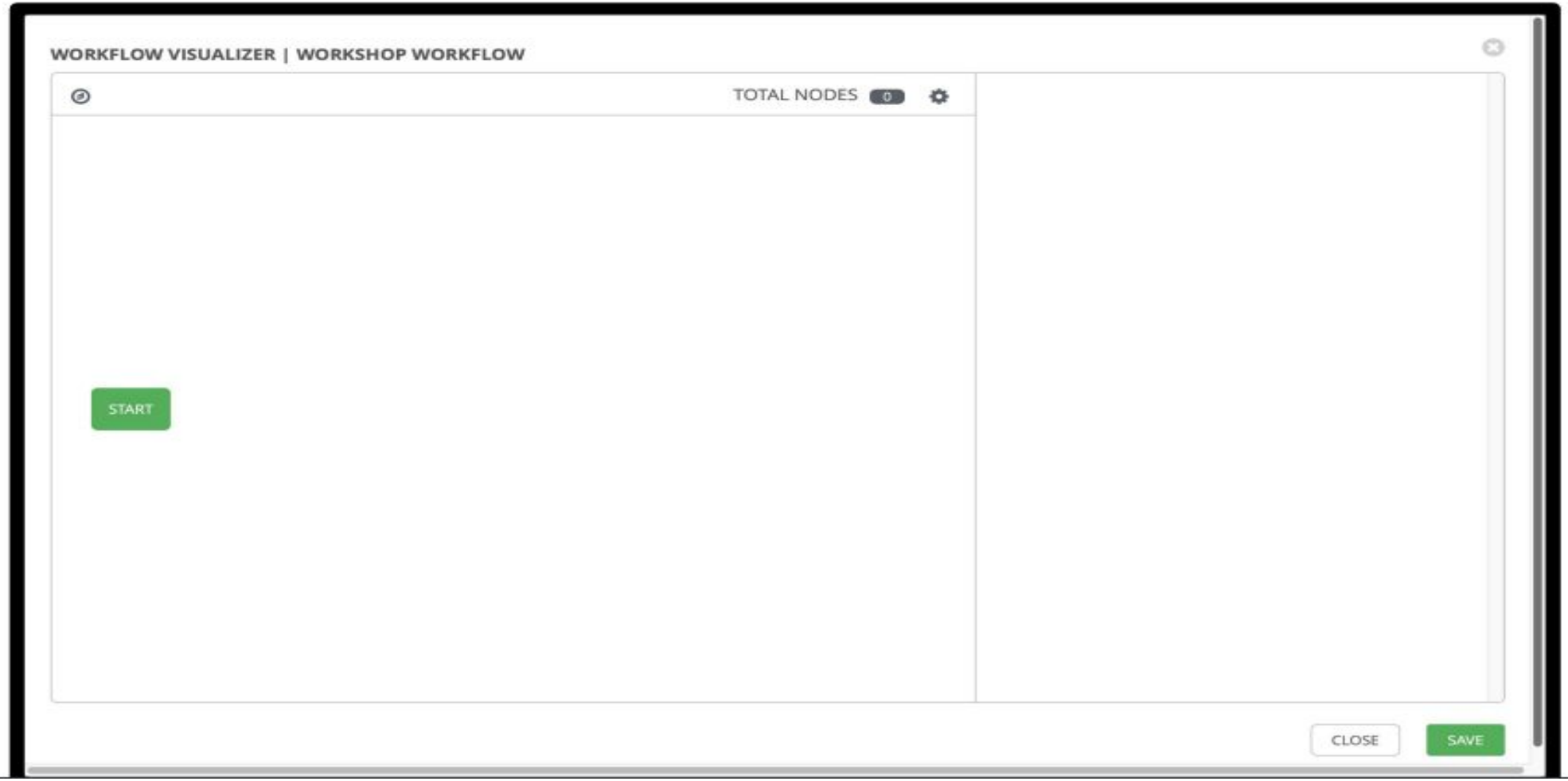
Fill out the required parameters and click **SAVE.** As soon as the Workflow Template is saved the WORKFLOW VISUALIZER will open.



Red Hat

Workflow Visualizer

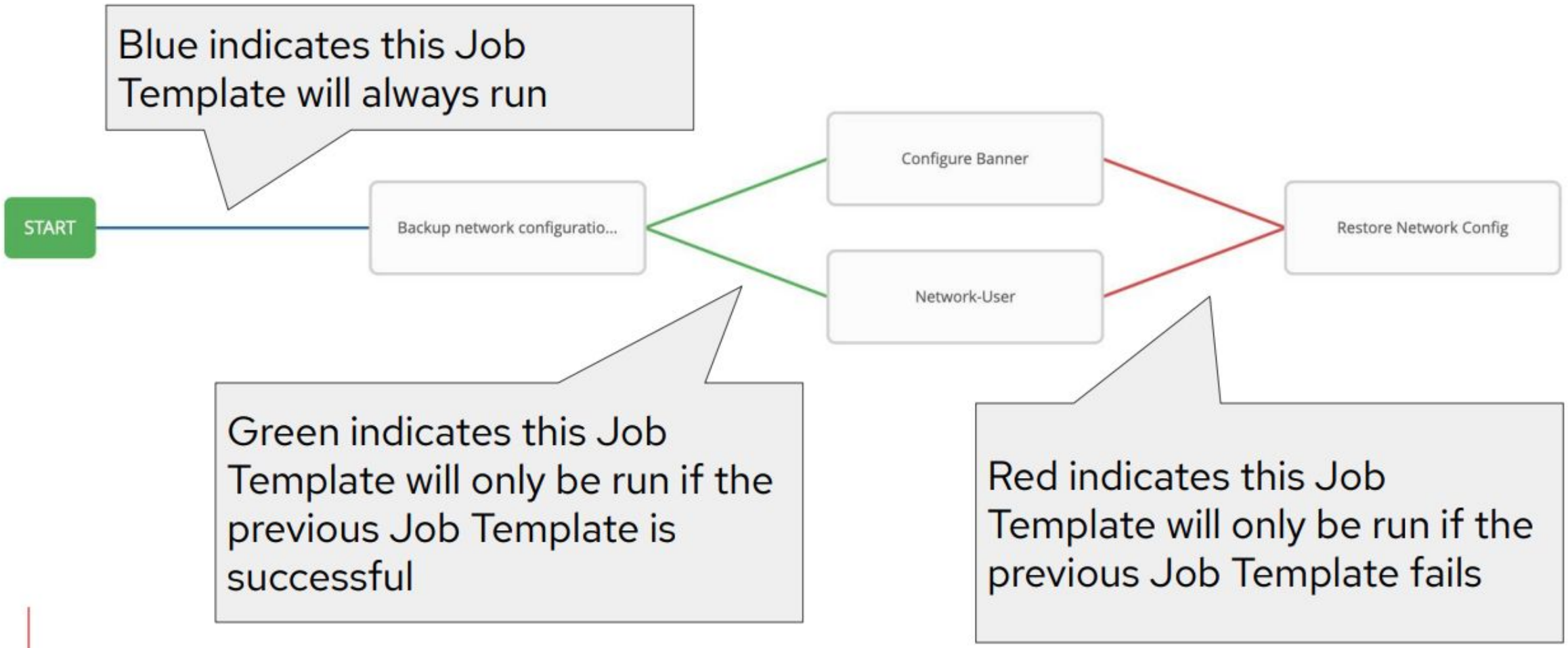
The workflow visualizer will start as a blank canvas.



Red Hat

Visualizing a Workflow

Workflows can branch out, or converge in.



2 more Exercise

Topics Covered:

- Red Hat Insights intro
- Insights integration

Red Hat
Ansible Automation
Platform

Red Hat

Red Hat Insights

Included with your Red Hat Enterprise Linux subscription

Assesses

customer's Red Hat environments

Remediates

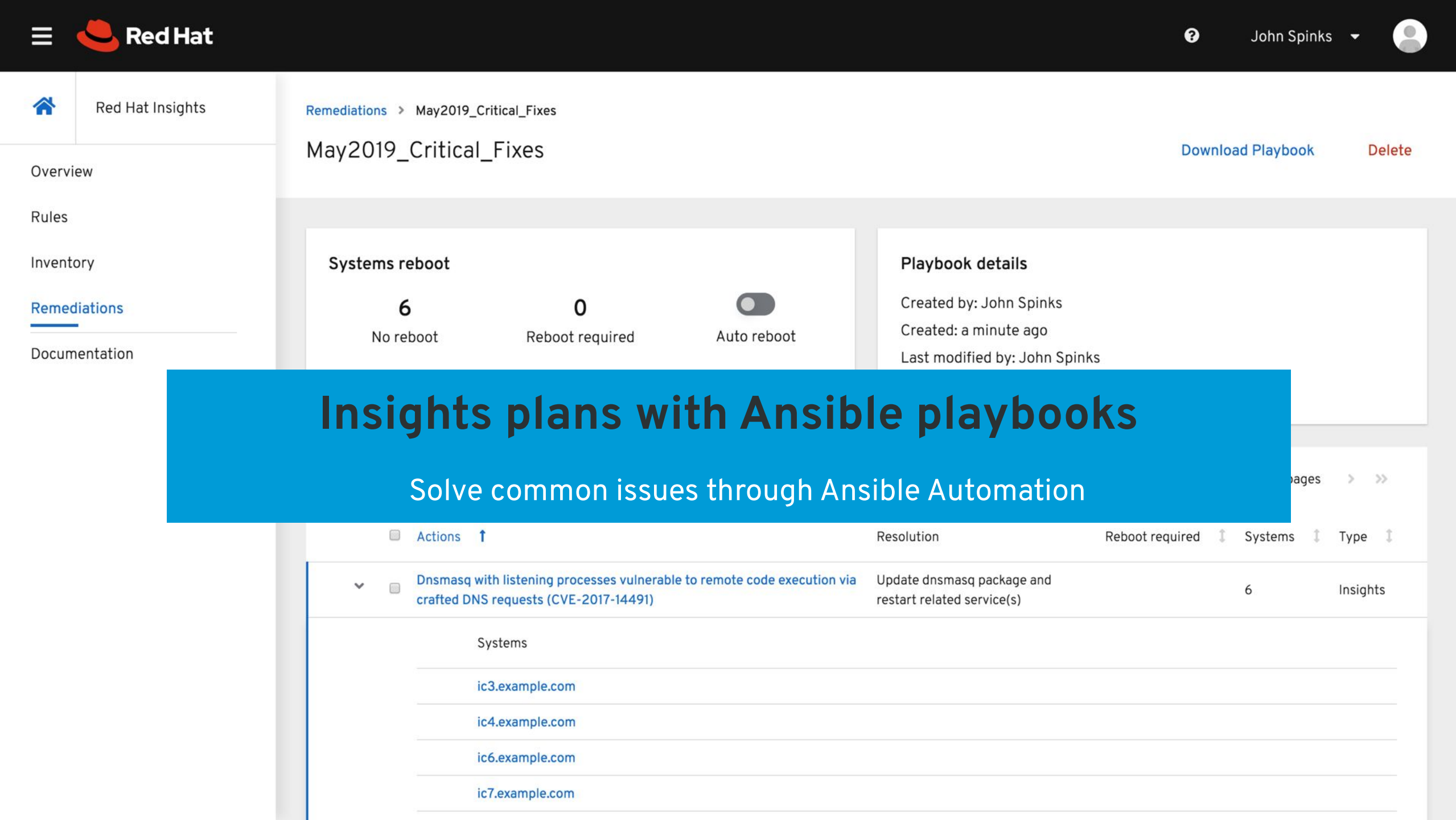
findings with prescriptive remediation steps or an Ansible playbook

Insights

rule contributions directly from Red Hat subject matter experts

Identifying risks for Availability, performance, stability and security

Red Hat


Red Hat
John Spinks
Red Hat Insights
Overview
Rules
Inventory
Remediations
Documentation
Remediations > May2019_Critical_Fixes
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
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

JOIN THE COMMUNITY

ansible.com/community

WORKSHOPS & TRAINING

ansible.com/workshops

Red Hat Training

SHARE YOUR STORY

Follow us @Ansible

Friend us on Facebook

Red Hat

AnsibleFest

October 13-14, 2020 | Virtual Experience

Red Hat

Thank you


linkedin.com/company/red-hat


youtube.com/AnsibleAutomation


facebook.com/ansibleautomation


twitter.com/ansible


github.com/ansible

Red Hat