



Red Hat Ansible Automation Platform

Ansible Automation Workshop

Introduction to Ansible Automation

Agenda

1400-1430	30 mins	Welcome & Ansible Automation overview presentation
1430-1440	10 mins	Lab setup & introducing the lab controls
1440-1525	45 mins	Hands-on 1+2 Overview of public cloud provisioning Converting shell commands into Ansible commands
1525-1610	45 mins	Hands-on 3+4 Retrieving information from hosts Deploying applications at scale
1610-1655	45 mins	Hands-on 5+6 Self-service IT via surveys Automation Workflows
1655-1700	5 mins	Wrap-up: hands-on workshop review

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
- Automation Workflows

Introduction

Topics Covered:

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



Red Hat
Ansible Automation
Platform

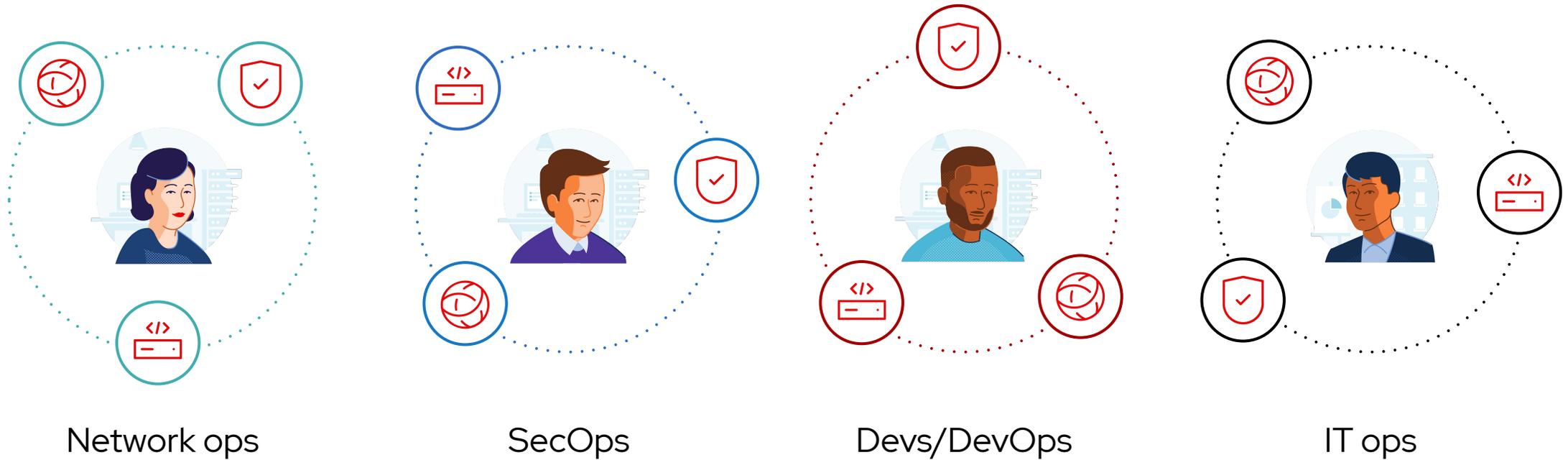


Anyone can automate...
but an enterprise needs
to coordinate and scale



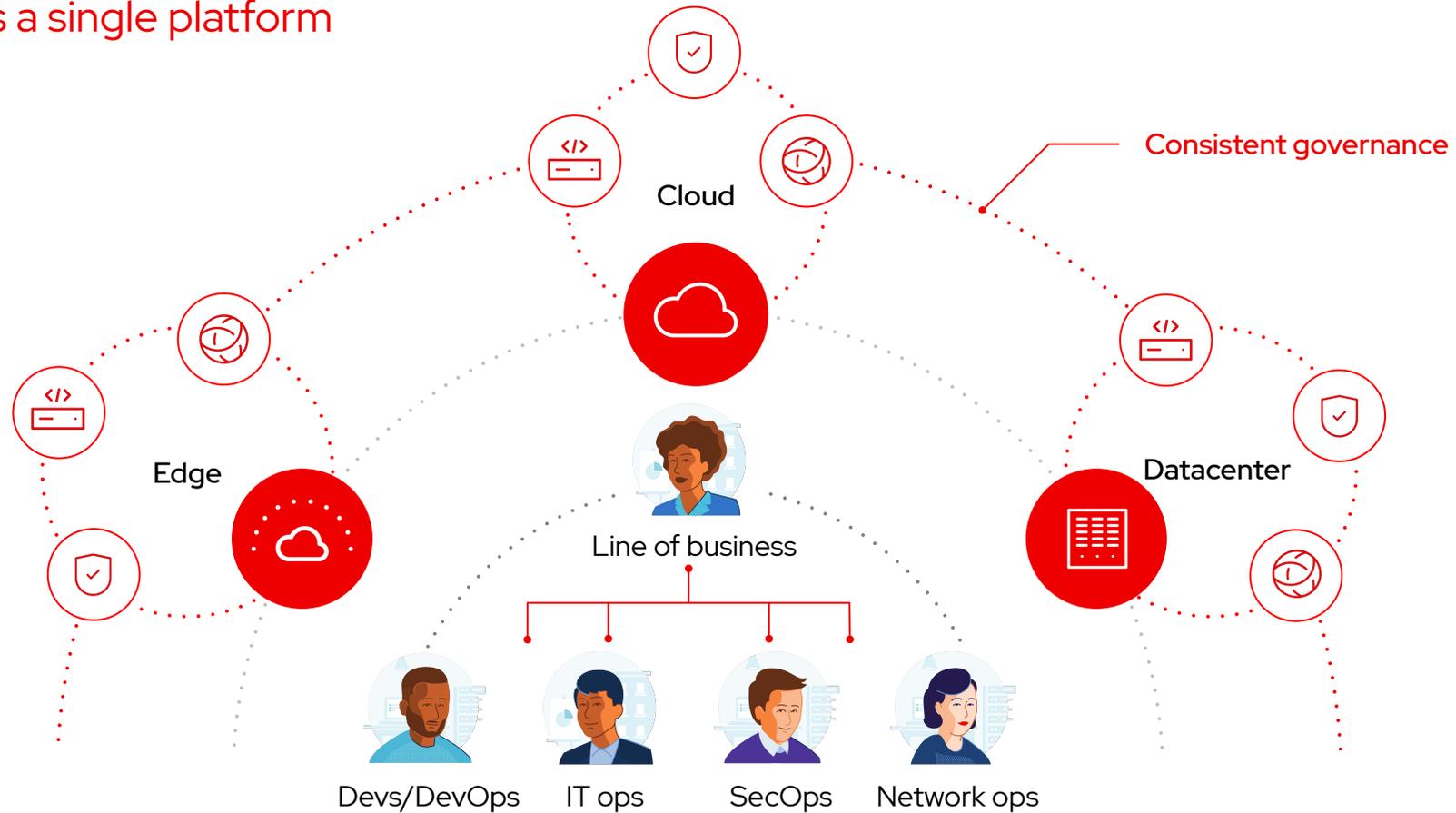
Many organizations share the same challenge

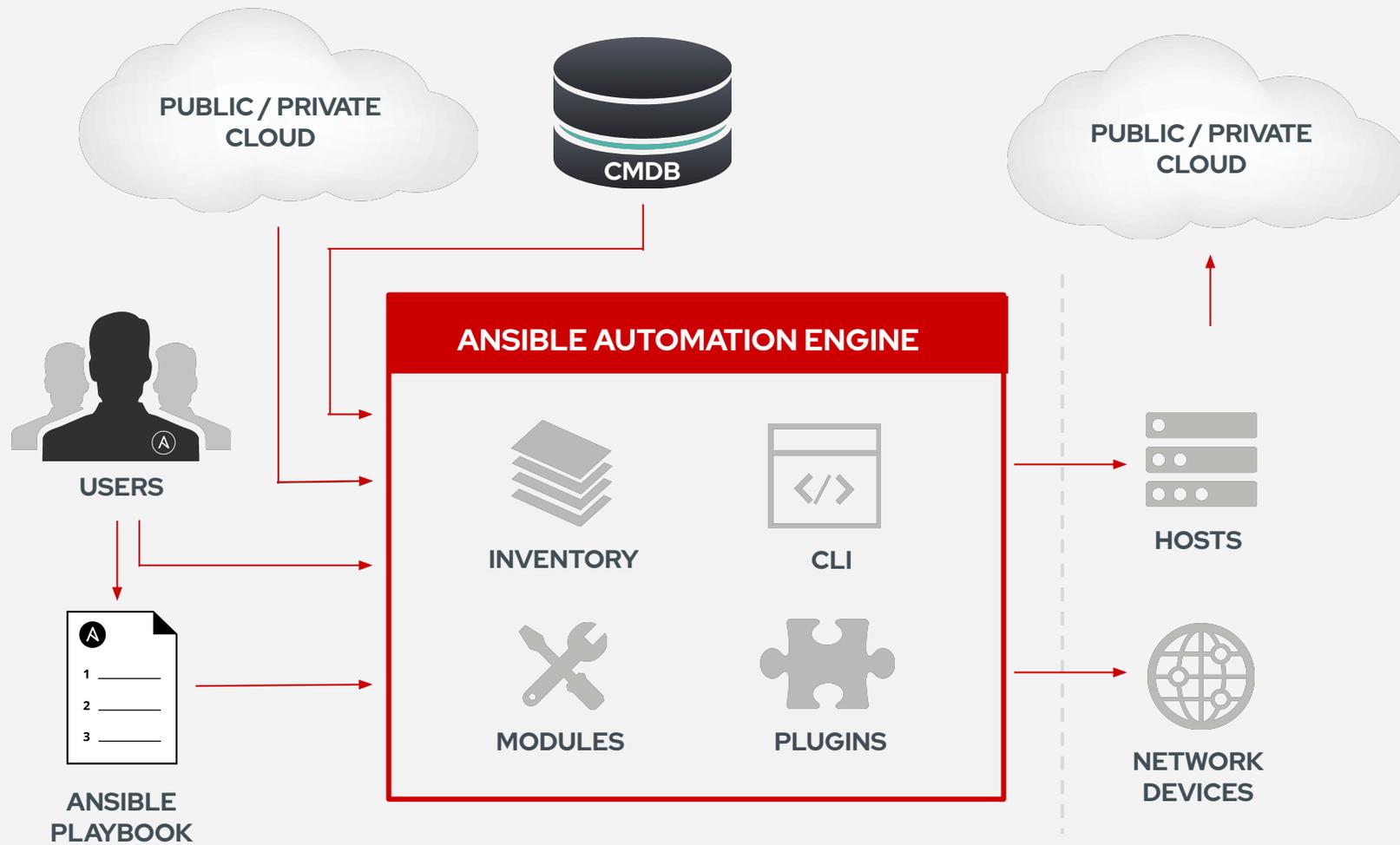
Too many unintegrated, domain-specific tools



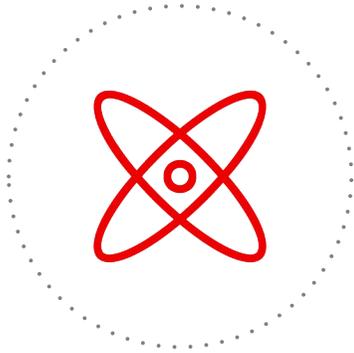
Break down silos

Different teams a single platform



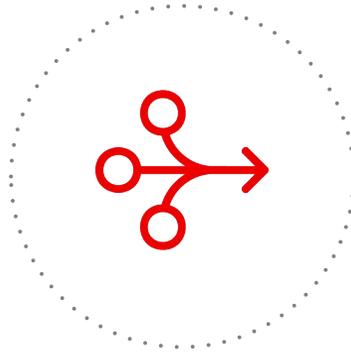


Why the Ansible Automation Platform?



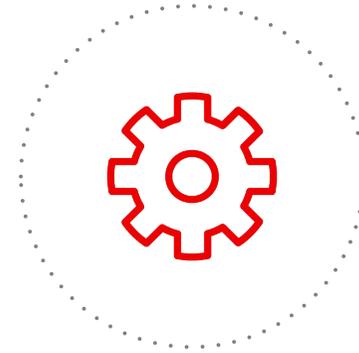
Powerful

Orchestrate complex processes at enterprise scale.



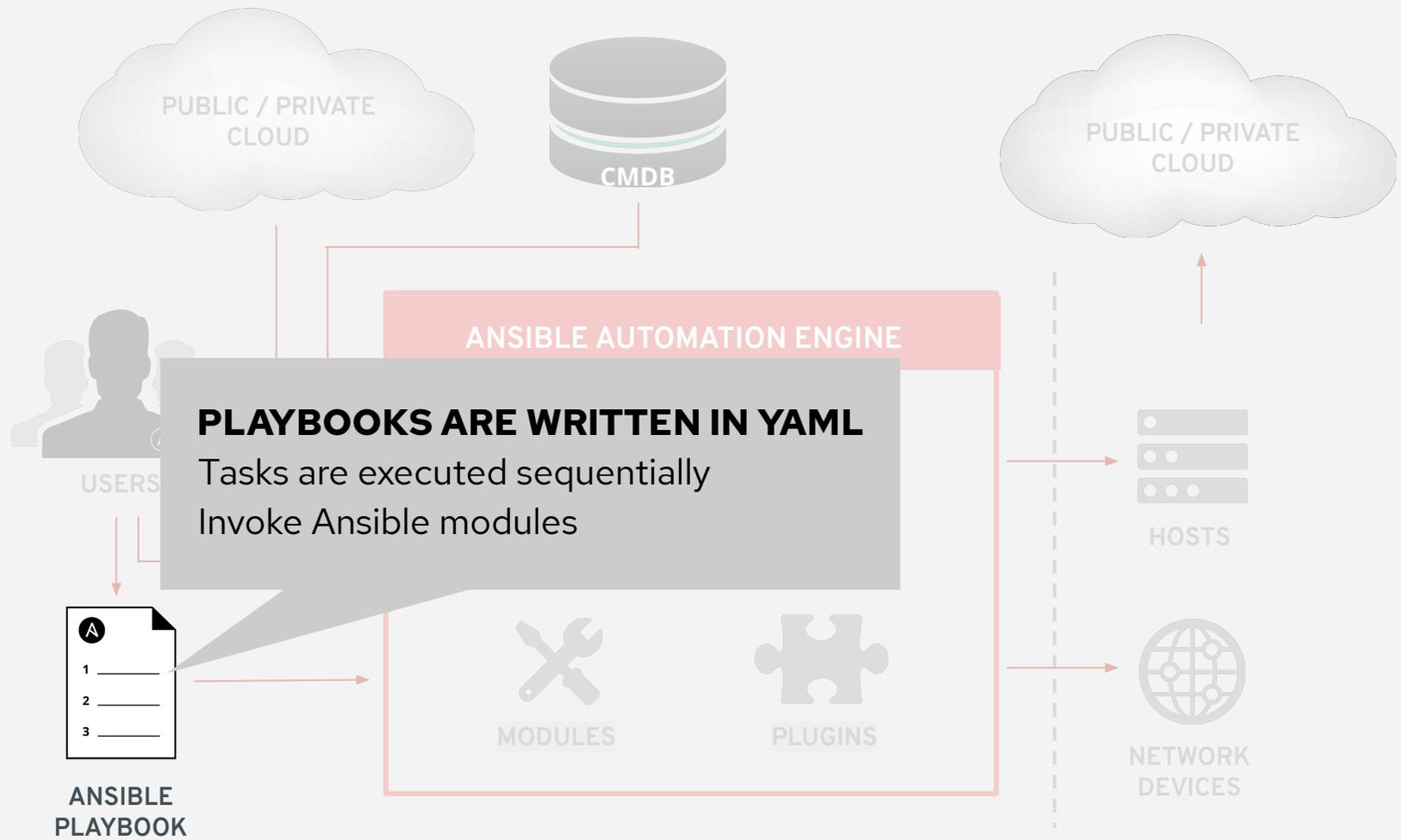
Simple

Simplify automation creation and management across multiple domains.

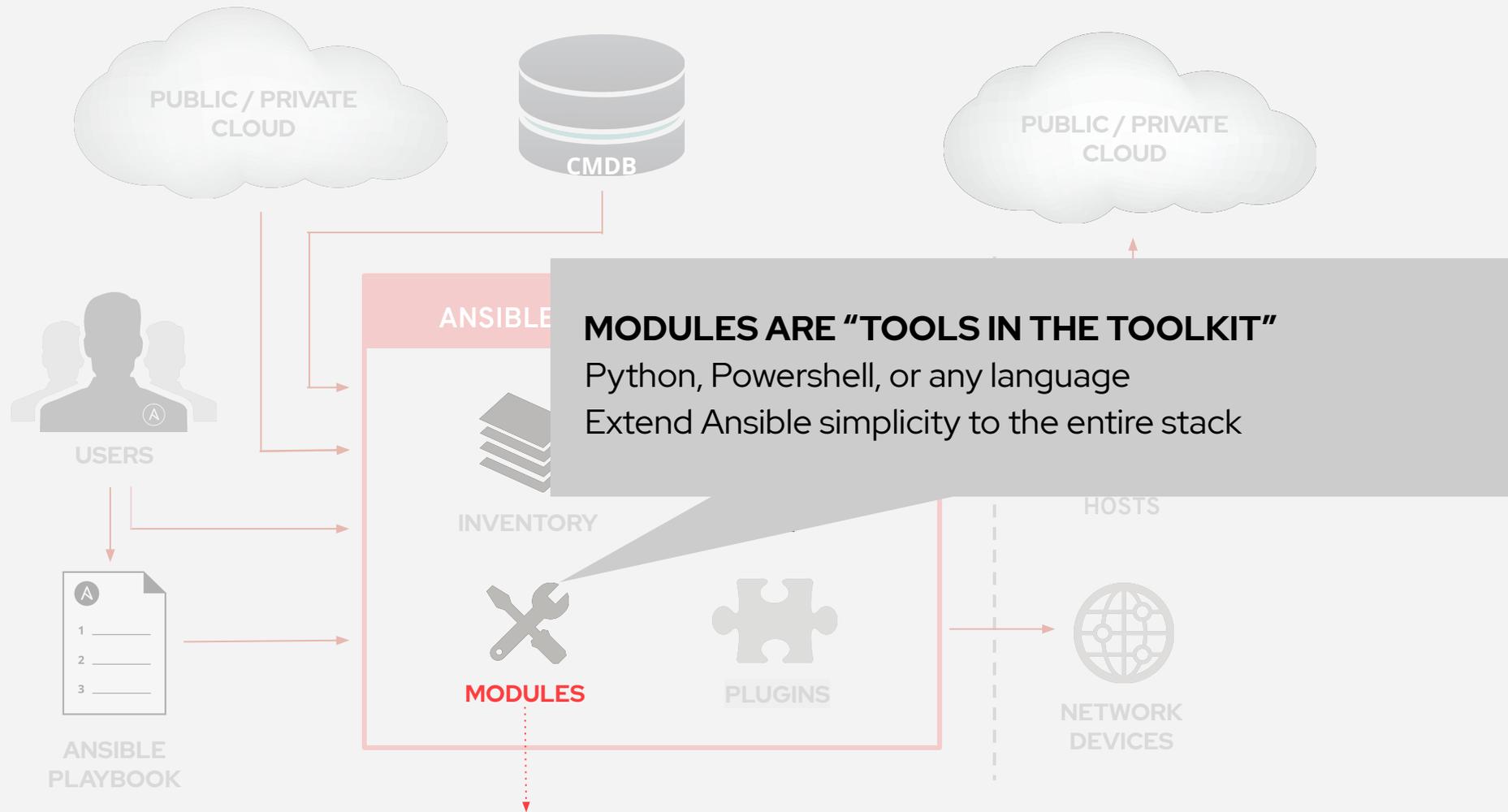


Agentless

Easily integrate with hybrid environments.



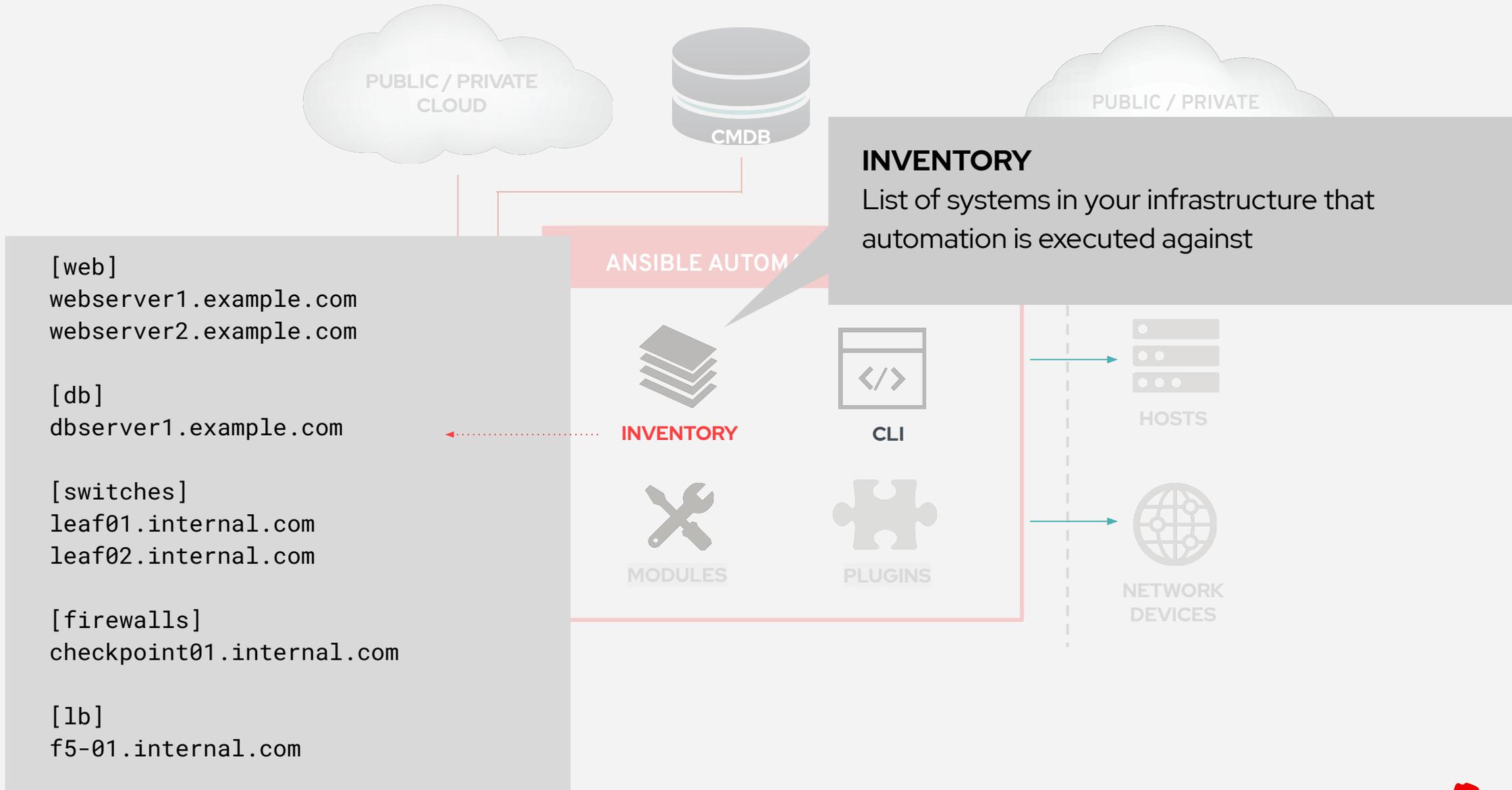
```
---  
- name: install and start apache  
  hosts: web  
  become: yes  
  
  tasks:  
    - name: httpd package is present  
      yum:  
        name: httpd  
        state: latest  
  
    - name: latest index.html file is present  
      file:  
        src: files/index.html  
        dest: /var/www/html/  
  
    - 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/

```



Ansible Automates Technologies You Use

Time to Automate is Measured in Minutes

CLOUD

AWS
Azure
Digital Ocean
Google
OpenStack
Rackspace
+more

OPERATING SYSTEMS

RHEL and Linux
UNIX
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

Arista
Aruba
A10
Cumulus
Bigswitch
Cisco
Dell
Ericsson
F5
Juniper
OpenSwitch
Ruckus
VyOS
+more

DEVOPS

Jira
GitHub
Vagrant
Jenkins
Bamboo
Atlassian
Subversion
Slack
Hipchat
+more

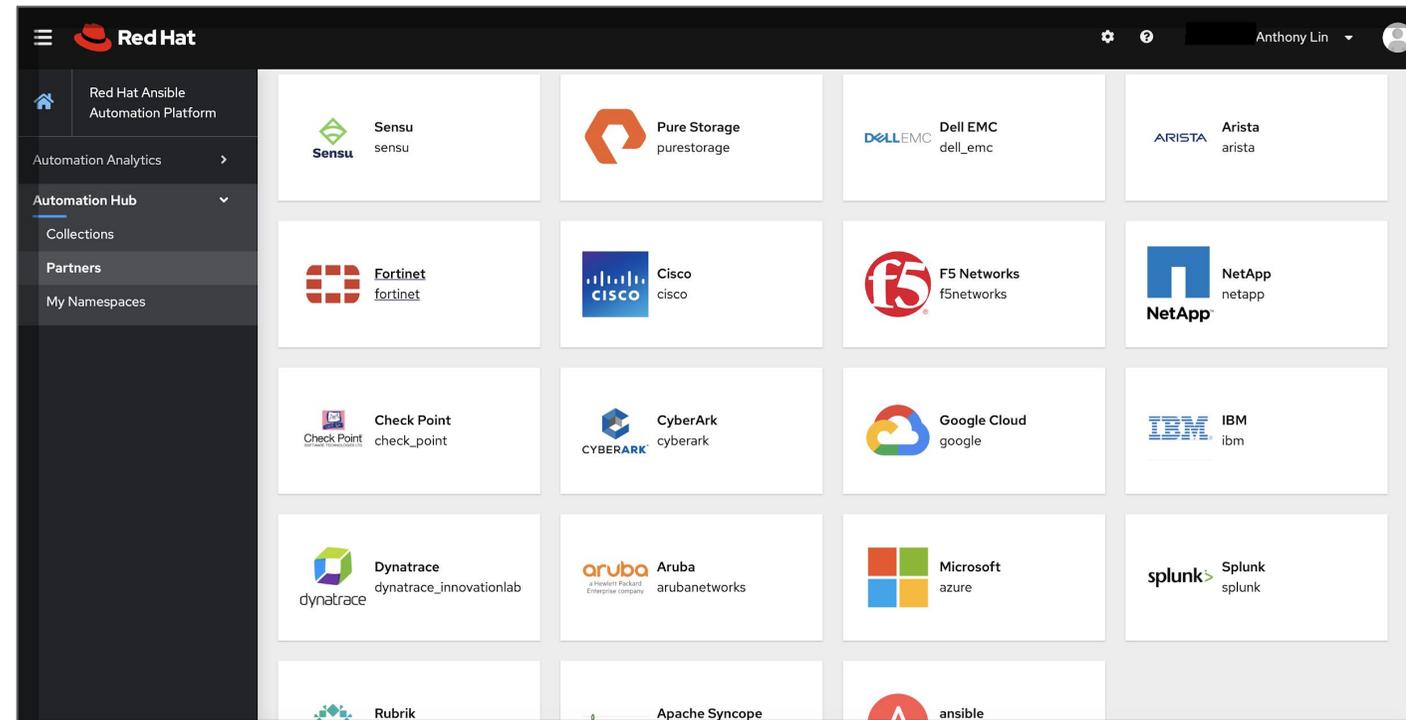
MONITORING

Dynatrace
Airbrake
BigPanda
Datadog
LogicMonitor
Nagios
New Relic
PagerDuty
Sensu
StackDriver
Zabbix
+more

Automation Hub

Discover, publish, and manage Collections

- Quickly discover available Red Hat and certified content through Collections.
- Manage and test your organization's view of available content.*
- Manage your locally available automation via on-premise.*



Automate the deployment and management of automation

Your entire IT footprint

Do this...

Orchestrate

Manage configurations

Deploy applications

Provision / deprovision

Deliver continuously

Secure and comply

On these...



Firewalls



Load balancers



Applications



Containers



Virtualization platforms



Servers



Clouds



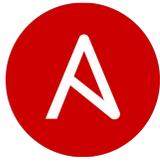
Storage



Network devices



And more ...



RED HAT®
ANSIBLE®
Tower

ACCESS CONTROL

Role-based access control
& LDAP integration

DELEGATION OF CREDENTIALS

Delegate credentials without
giving away secrets

INVENTORY MANAGEMENT

Graphically manage your internal &
cloud resources

PUSH-BUTTON LAUNCH

Launch automation jobs with a button

API & CLI

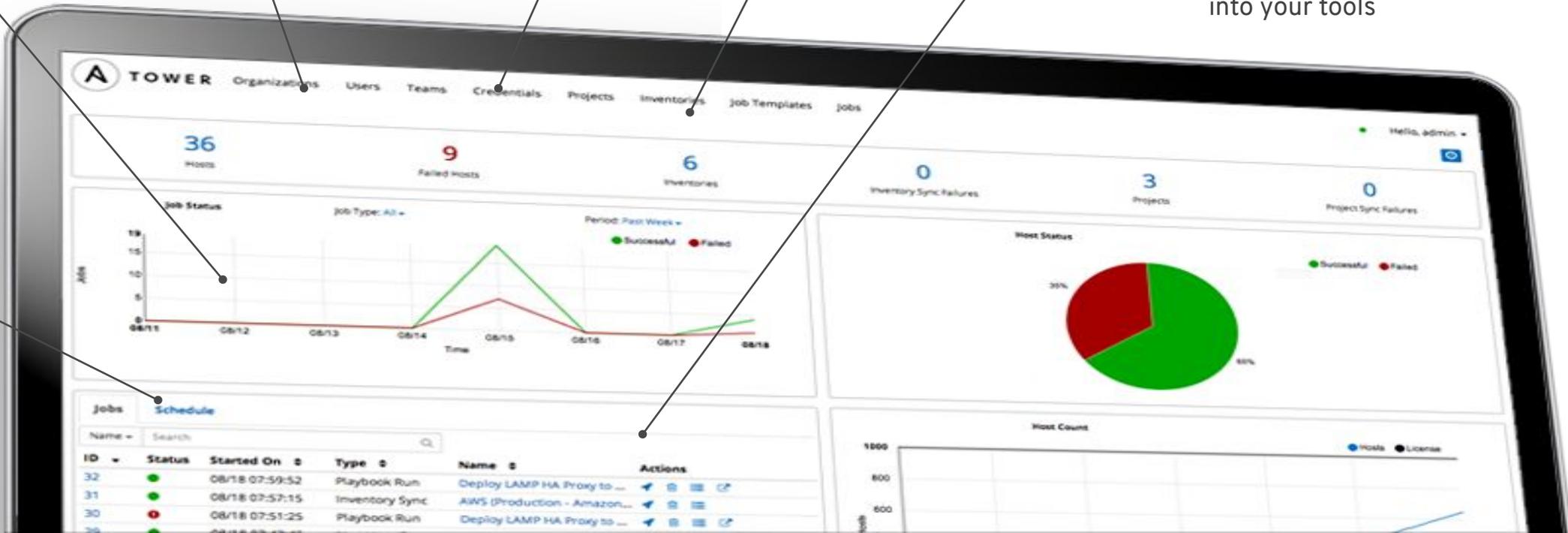
Documented RESTful API and
Tower CLI to integrate Tower
into your tools

AUDITING

See a full Ansible job
history with drill-in details

SCHEDULING

Schedule
automation jobs
(great for
periodic
remediation)



What makes a platform?

Red Hat Ansible Automation Platform



Content creators



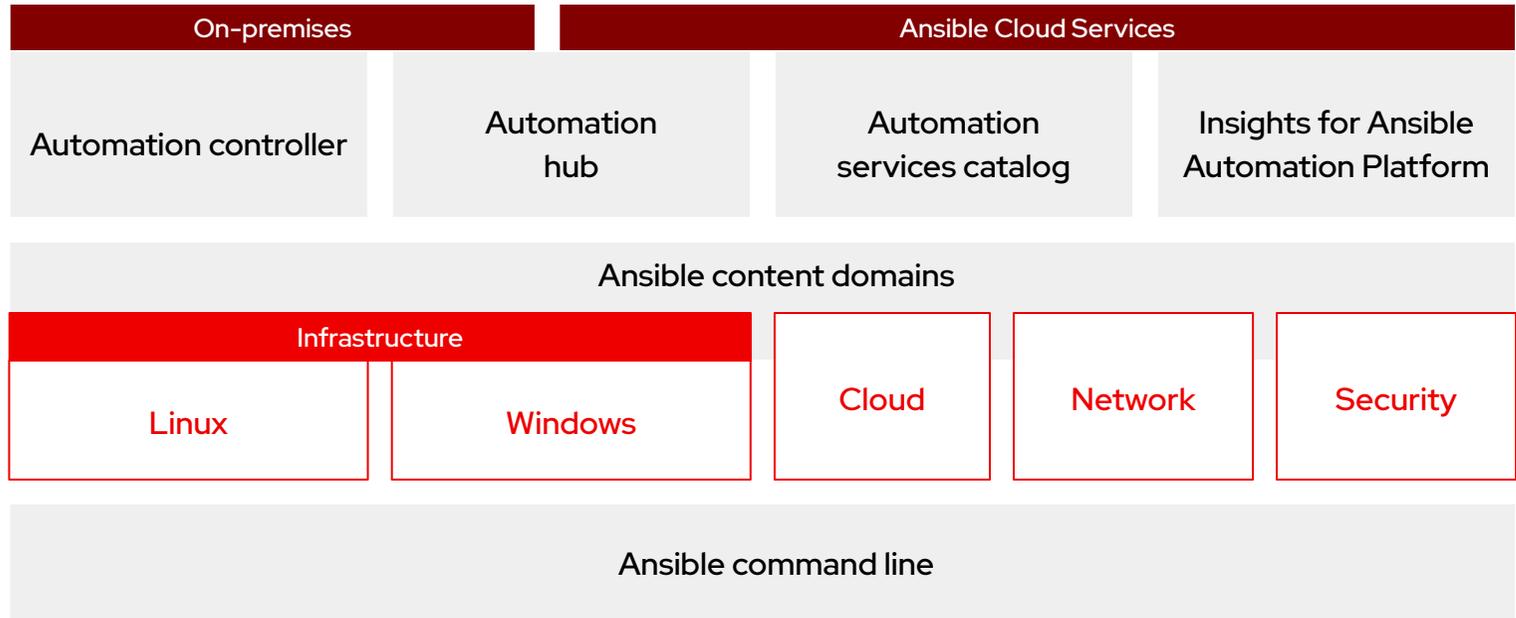
Operators



Domain experts

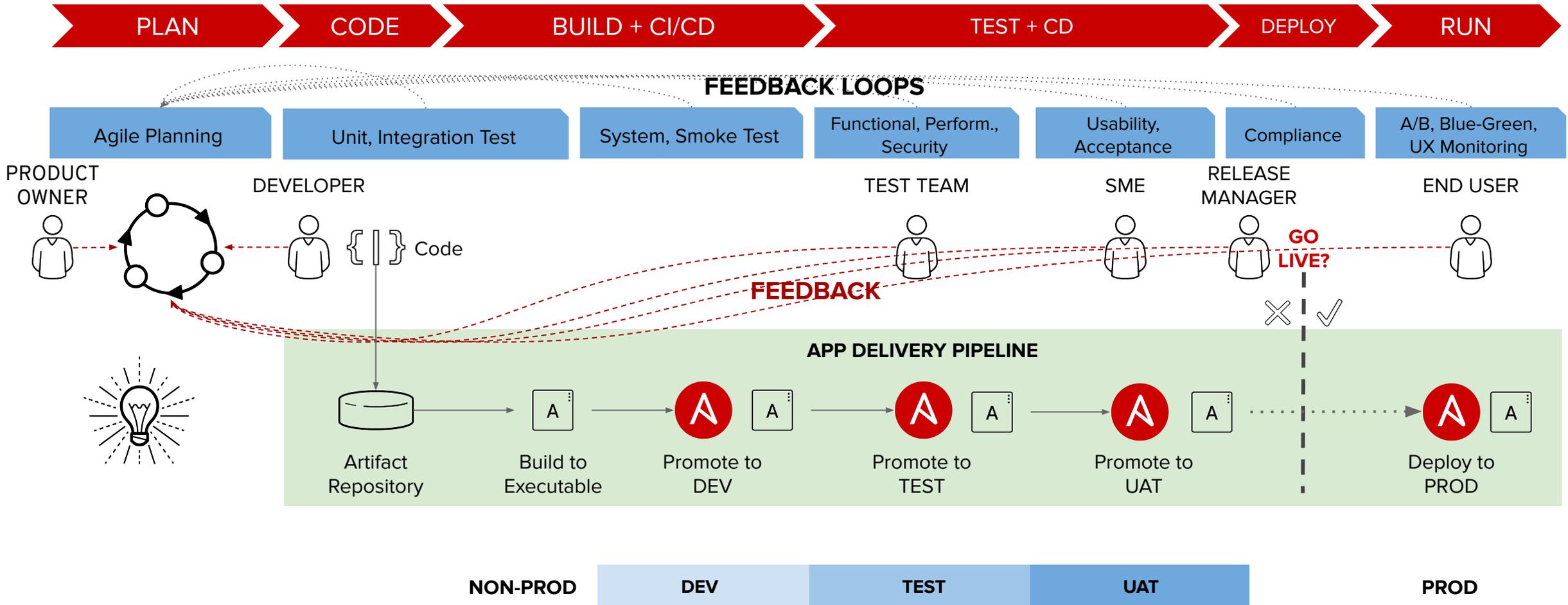


Users



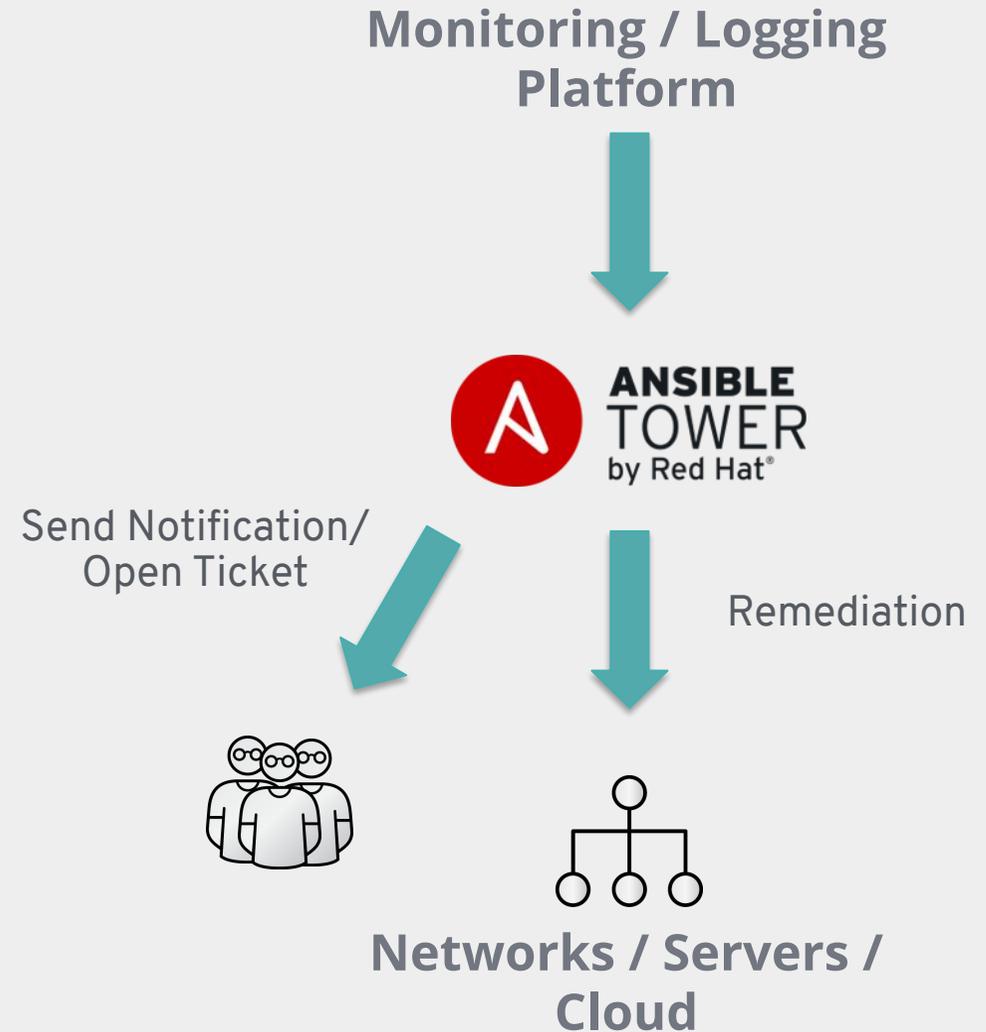
Fueled by an
open source community

DevOps Pipeline with Ansible

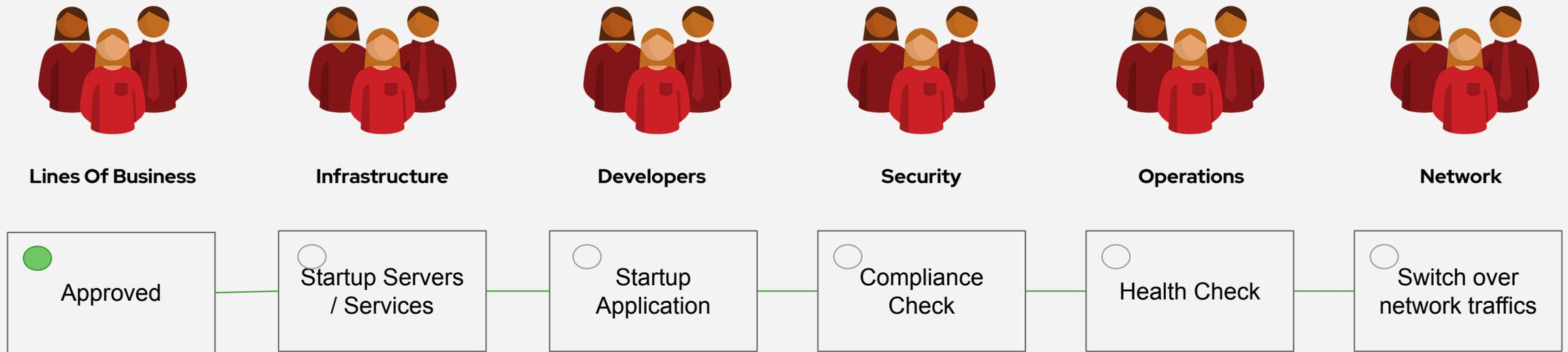


Remediate Automation

1. Monitoring/Logging Platform detects security issues and calls the Ansible Tower API
2. Ansible Tower runs a playbook to automate remediation in servers / equipments
3. Ansible Tower runs a playbook to open a support ticket and/or notify security managers / system administrators

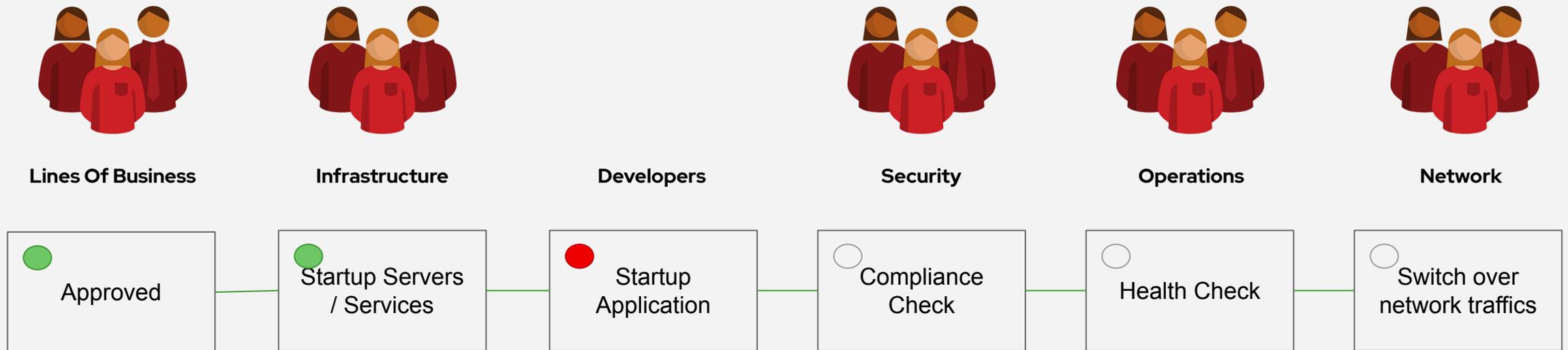


Teams are implementing Disaster Recovery...



Some planning tasks (e.g. Disaster Recovery Drill) usually required different teams to work together.

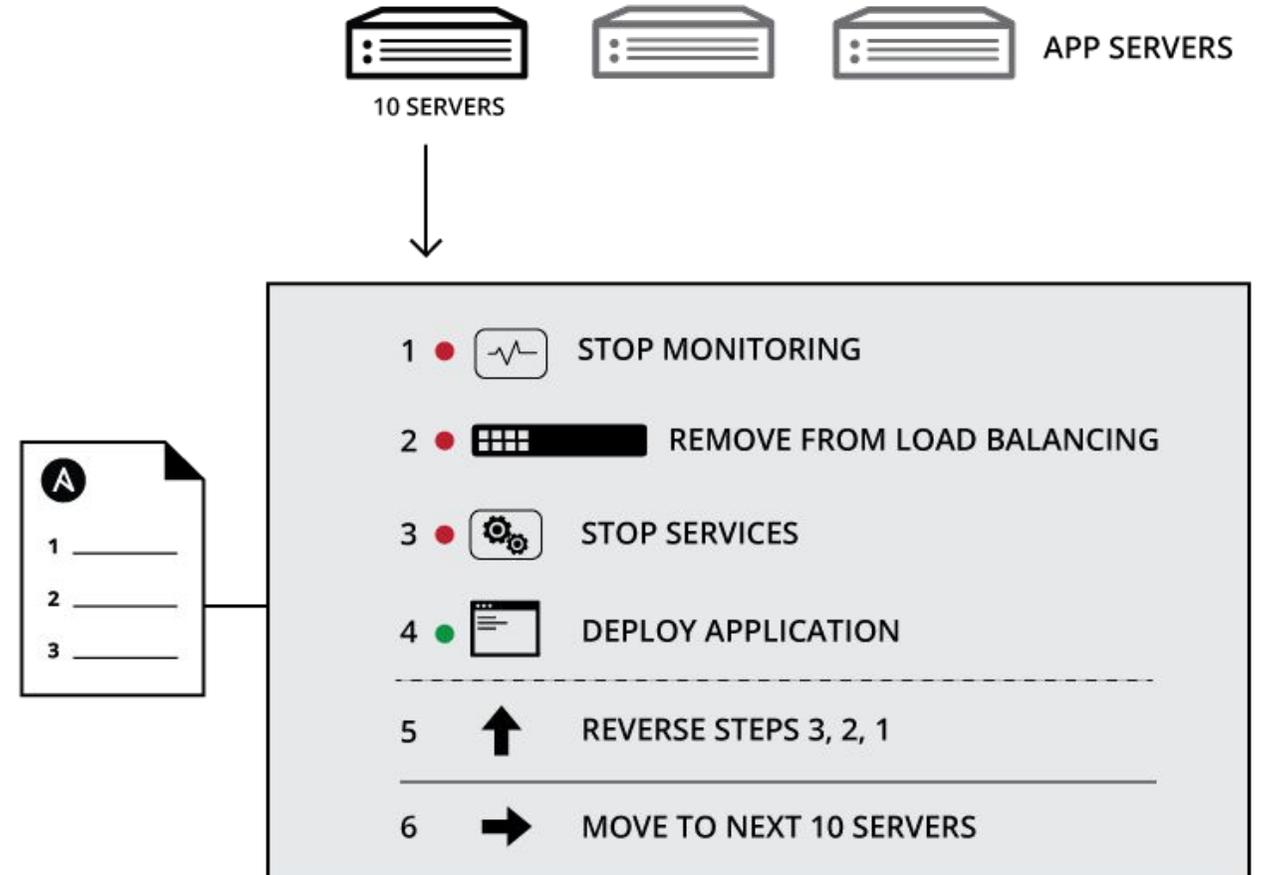
Teams are implementing Disaster Recovery...



Some planning tasks (e.g. Disaster Recovery Drill) can't be performed as it usually required different teams to work together.

Rolling Upgrade / Patching

Your applications and systems are more than just collections of configurations. They're a finely tuned and ordered list of tasks and processes that result in your working application.



Security Compliance – C2S



Ansible remediation role for profile C2S

Profile Title: C2S for Red Hat Enterprise Linux 7

```
- hosts: all
  roles:
    - role: RedHatOfficial.rhel7_c2s
      when:
        - ansible_os_family == 'RedHat'
        - ansible_distribution_major_version | version_compare('7', '=')
```

- Huge manual works for checking / remediation of security compliance settings

After Ansible Automation:

- Shorten the time for manual works



[RHEL 7 C2S](#)

<https://rhelblog.redhat.com/2018/06/19/automating-security-compliance-with-ease/>



Exercise 1

Topics Covered:

- Understanding the Ansible Infrastructure
- Check the prerequisites

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>

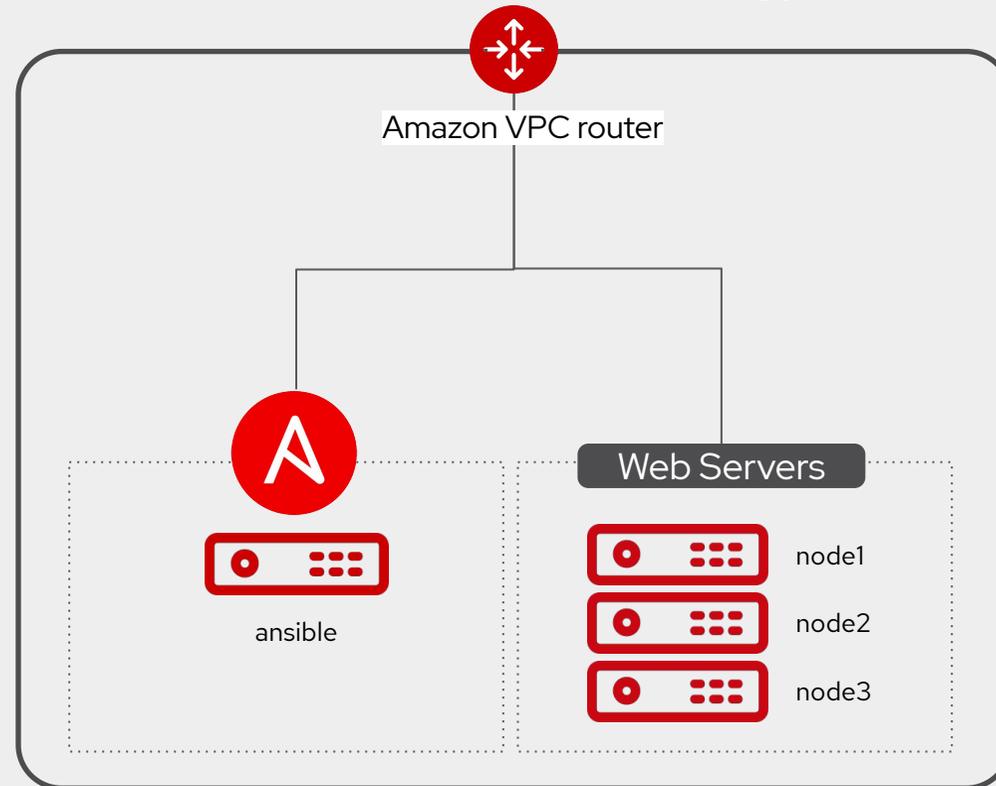
- **Learn with the real thing**

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

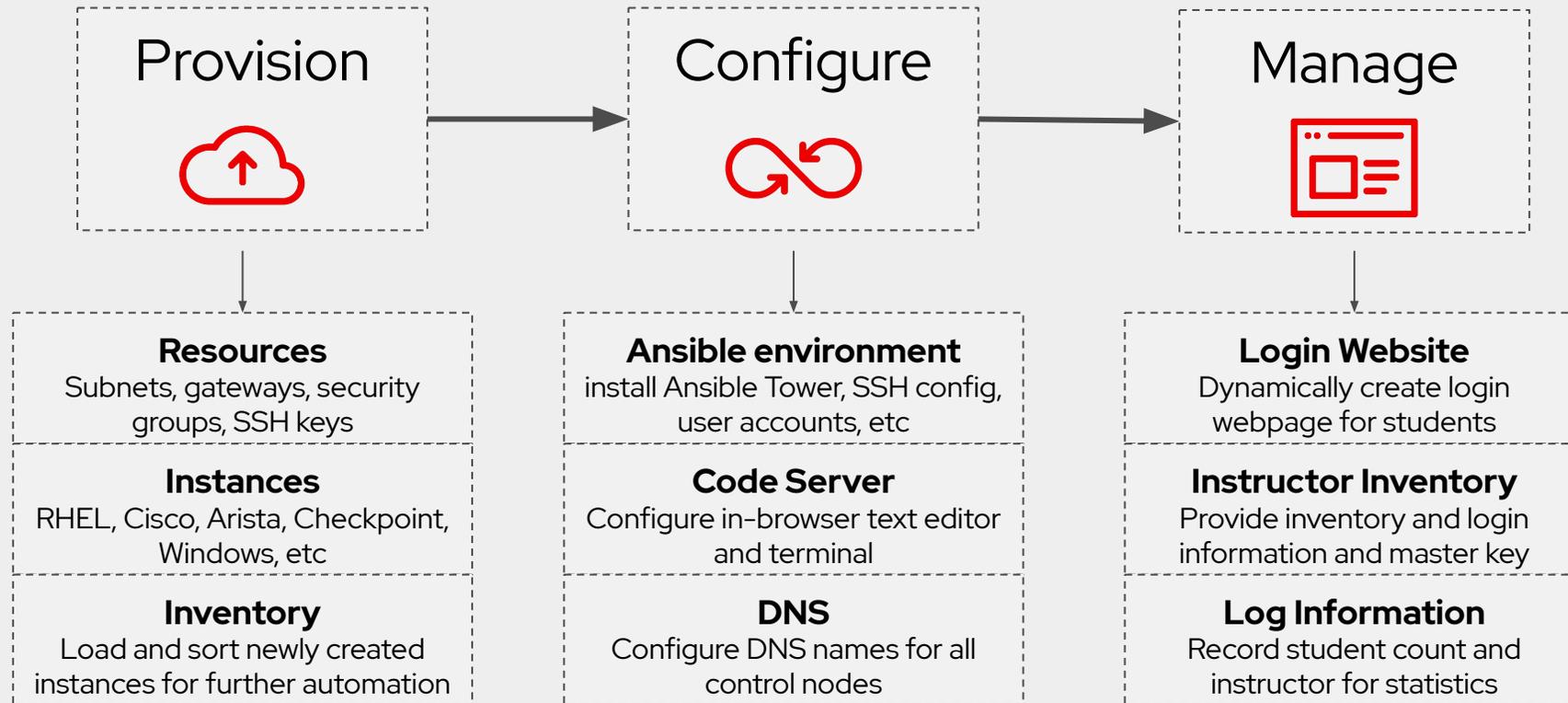
- **Red Hat Enterprise Linux**

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

Lab Time

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

<https://red.ht/ansibleworkshop>



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

<https://red.ht/ansibleworkshop>

Exercise 3

Topics Covered:

- Playbooks basics
- Running a playbook



Red Hat
Ansible Automation
Platform

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

<https://red.ht/ansibleworkshop>

Exercise 4

Topics Covered:

- Working with variables
- What are facts?



Red Hat
Ansible Automation
Platform

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

```
├──
├── 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 Ansible Automation Platform

Lab Time

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

<https://red.ht/ansibleworkshop>

Exercise 5

Topics Covered:

- Surveys



Red Hat
Ansible Automation
Platform

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.

CREATE INDEX.HTML

SURVEY PREVIEW

* FIRST LINE

* SECOND LINE

CANCEL NEXT

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.

ADD SURVEY

The screenshot shows the Tower web interface. At the top left is the 'TOWER' logo. The top right shows the user 'admin' and several notification icons. The main content area is titled 'TEMPLATES / Create index.html'. On the left is a dark sidebar with navigation options: VIEWS (Dashboard, Jobs, Schedules, My View) and RESOURCES (Templates, Credentials, Projects, Inventories, Inventory Scripts). The main configuration area is titled 'Create index.html' and has several tabs: DETAILS, PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and EDIT SURVEY. The 'EDIT SURVEY' tab is highlighted with a red box. Below the tabs are several configuration fields:

- * NAME**: Create index.html
- DESCRIPTION**: (empty)
- * JOB TYPE**: (dropdown menu)
- PROMPT ON LAUNCH
- * INVENTORY**: Workshop Inventory
- PROMPT ON LAUNCH
- * PROJECT**: Workshop Project
- * PLAYBOOK**: rhel/apache/apache_role_inst...
- PROMPT ON LAUNCH
- CREDENTIALS**: Workshop Credential
- PROMPT ON LAUNCH
- FORKS**: 0
- LIMIT**: web
- PROMPT ON LAUNCH
- * VERBOSITY**: 0 (Normal)
- PROMPT ON LAUNCH
- JOB TAGS**: (empty)
- PROMPT ON LAUNCH
- SKIP TAGS**: (empty)
- PROMPT ON LAUNCH

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 'Add Survey' window in Ansible Tower. The window title is 'Create index.html | SURVEY'. The left pane, titled 'ADD SURVEY PROMPT', contains the following fields and controls:

- * PROMPT**: A text input field.
- DESCRIPTION**: A text input field.
- * ANSWER VARIABLE NAME**: A text input field with a help icon.
- * ANSWER TYPE**: A dropdown menu.
- REQUIRED**
- CLEAR** and **+ ADD** buttons.

The right pane, titled 'PREVIEW', shows two lines of text:

- * FIRST LINE**: A text input field with a list icon, an edit icon, and a delete icon.
- * SECOND LINE**: A text input field with a list icon, an edit icon, and a delete icon.

At the bottom of the window, there are three buttons: **DELETE SURVEY** (red), **CANCEL** (white), and **SAVE** (green).

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 image shows a screenshot of a survey dialog box titled "CREATE INDEX.HTML". The dialog has a close button (X) in the top right corner. Below the title, there are two buttons: "SURVEY" (highlighted in dark grey) and "PREVIEW" (light grey). Below these buttons, there are two required text input fields, each preceded by a red asterisk and the label "FIRST LINE" and "SECOND LINE" respectively. At the bottom of the dialog, there are two buttons: "CANCEL" (light grey) and "NEXT" (dark grey).



Red Hat Ansible Automation Platform

Lab Time

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

<https://red.ht/ansibleworkshop>

Exercise 6

Topics Covered:

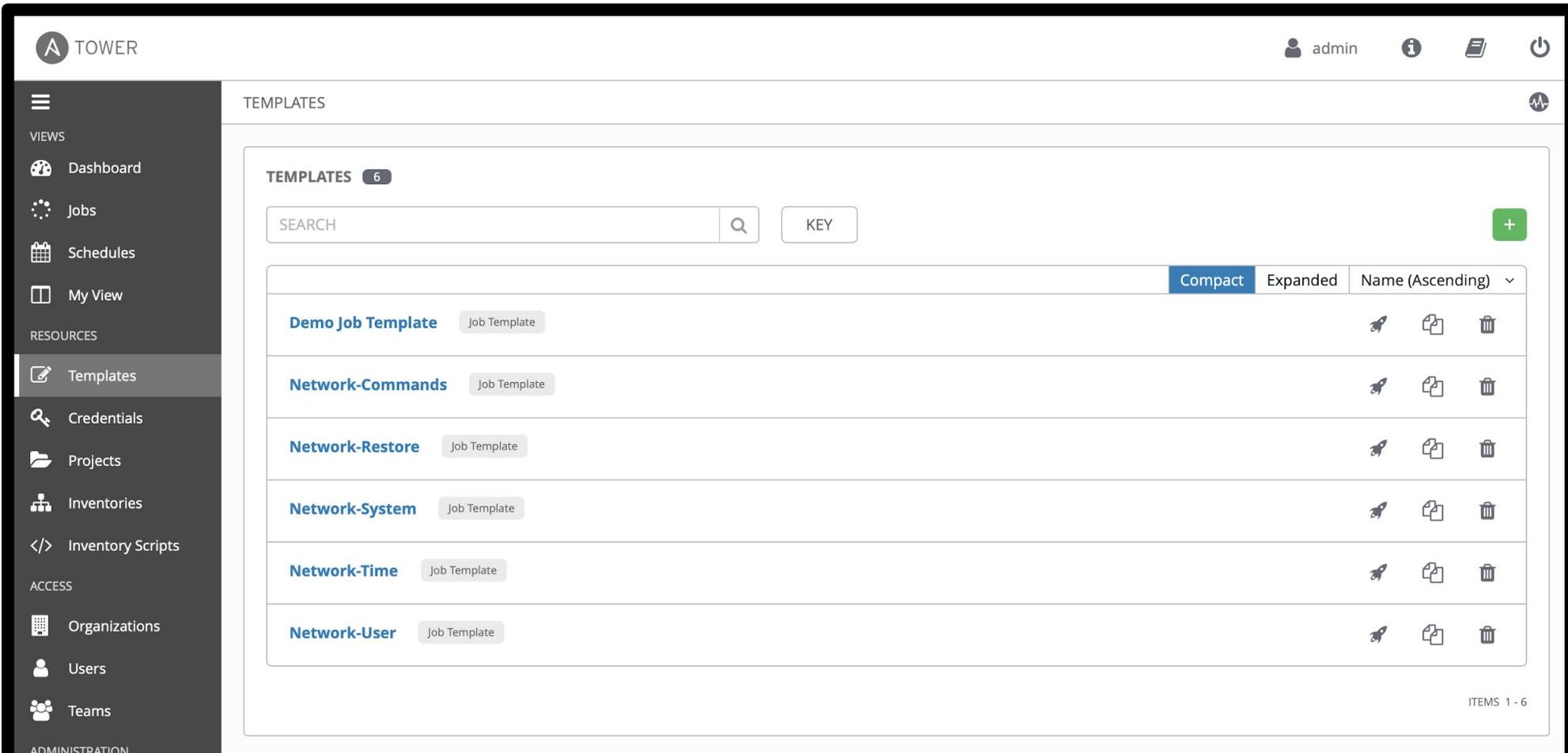
- Workflows



Red Hat
Ansible Automation
Platform

Workflows

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



The screenshot displays the Tower web interface. The left sidebar shows the navigation menu with the 'Templates' button highlighted under the 'RESOURCES' section. The main content area is titled 'TEMPLATES' and shows a list of job templates. The list includes a search bar, a 'KEY' button, and a '+ ' button. The table below shows the following templates:

TEMPLATES 6		SEARCH	Q	KEY	+
		Compact	Expanded	Name (Ascending)	▼
Demo Job Template	Job Template				
Network-Commands	Job Template				
Network-Restore	Job Template				
Network-System	Job Template				
Network-Time	Job Template				
Network-User	Job Template				

ITEMS 1 - 6

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 Tower web interface. The top navigation bar includes the Tower logo, the user name 'admin', and several utility icons. The left sidebar contains a menu with categories: VIEWS (Dashboard, Jobs, Schedules, My View), RESOURCES (Templates, Credentials, Projects, Inventories, Inventory Scripts), and ACCESS (Organizations, Users). The main content area is titled 'TEMPLATES' and shows a list of templates. A search bar and a 'KEY' button are at the top of the list. A dropdown menu is open, showing 'Job Template' and 'Workflow Template' options. The 'Workflow Template' option is highlighted. The list of templates includes 'Backup network configurations', 'Configure Banner', 'Demo Job Template', 'Network-Commands', 'Network-Restore', and 'Network-System'. Each template entry has a 'Job Template' label and a set of icons for actions like run, copy, and delete.

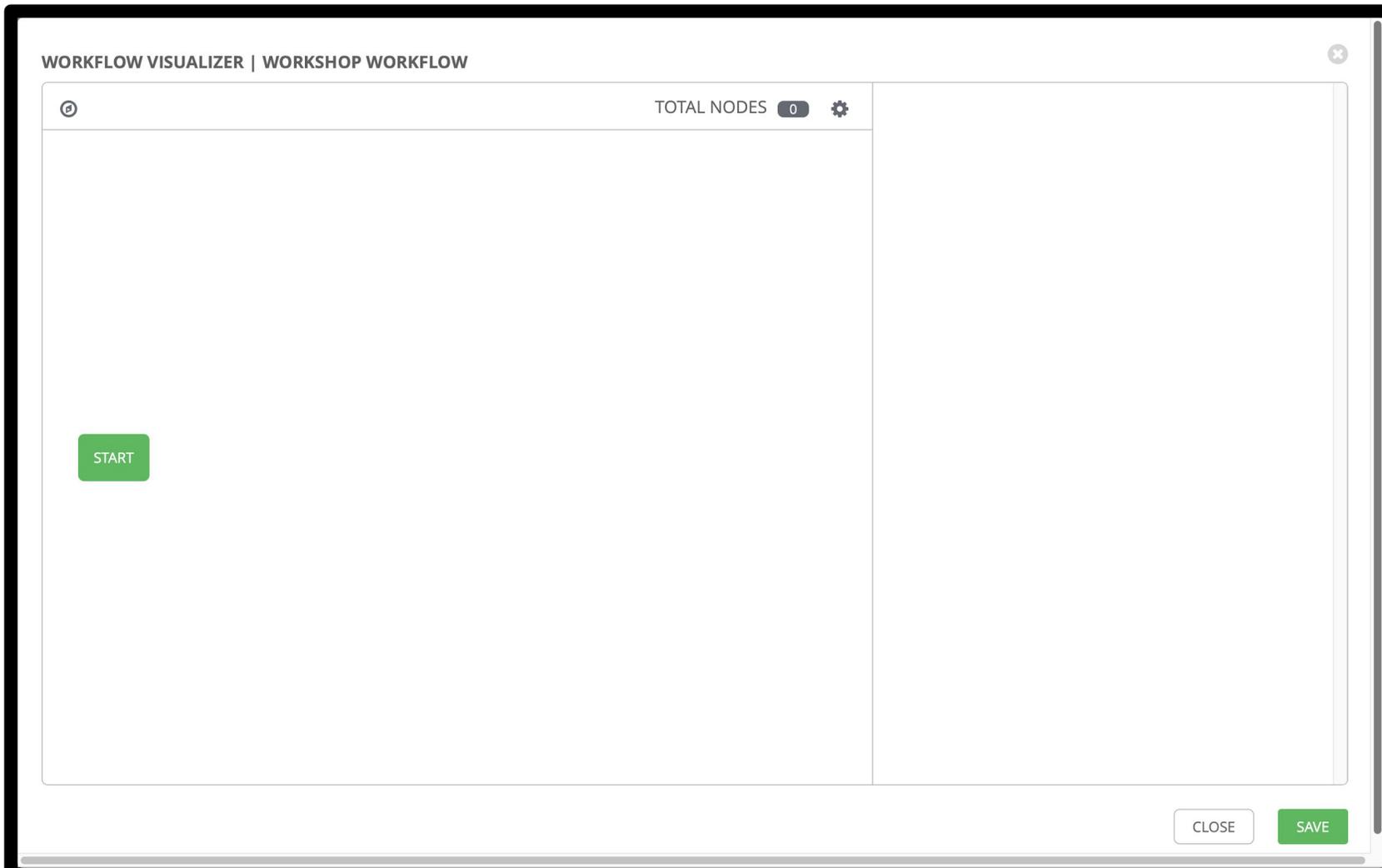
Creating the Workflow

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

The screenshot displays the Tower web interface for configuring a 'WORKSHOP WORKFLOW'. The top navigation bar shows the 'TOWER' logo, the user 'admin', and various utility icons. The left sidebar contains navigation options under 'VIEWS', 'RESOURCES', and 'ACCESS'. The main content area is titled 'TEMPLATES / WORKSHOP WORKFLOW' and features a tabbed interface with buttons for 'DETAILS', 'PERMISSIONS', 'NOTIFICATIONS', 'COMPLETED JOBS', 'SCHEDULES', and 'ADD SURVEY'. The 'WORKFLOW VISUALIZER' button is highlighted with a red box. Below the tabs, there are input fields for 'NAME' (set to 'WORKSHOP WORKFLOW'), 'DESCRIPTION', 'ORGANIZATION' (set to 'Default'), 'INVENTORY' (set to 'Workshop Inventory'), 'LABELS', and 'OPTIONS' (with 'ENABLE CONCURRENT JOBS' checked). At the bottom, there is a section for 'EXTRA VARIABLES' with tabs for 'YAML' and 'JSON', and a 'PROMPT ON LAUNCH' checkbox.

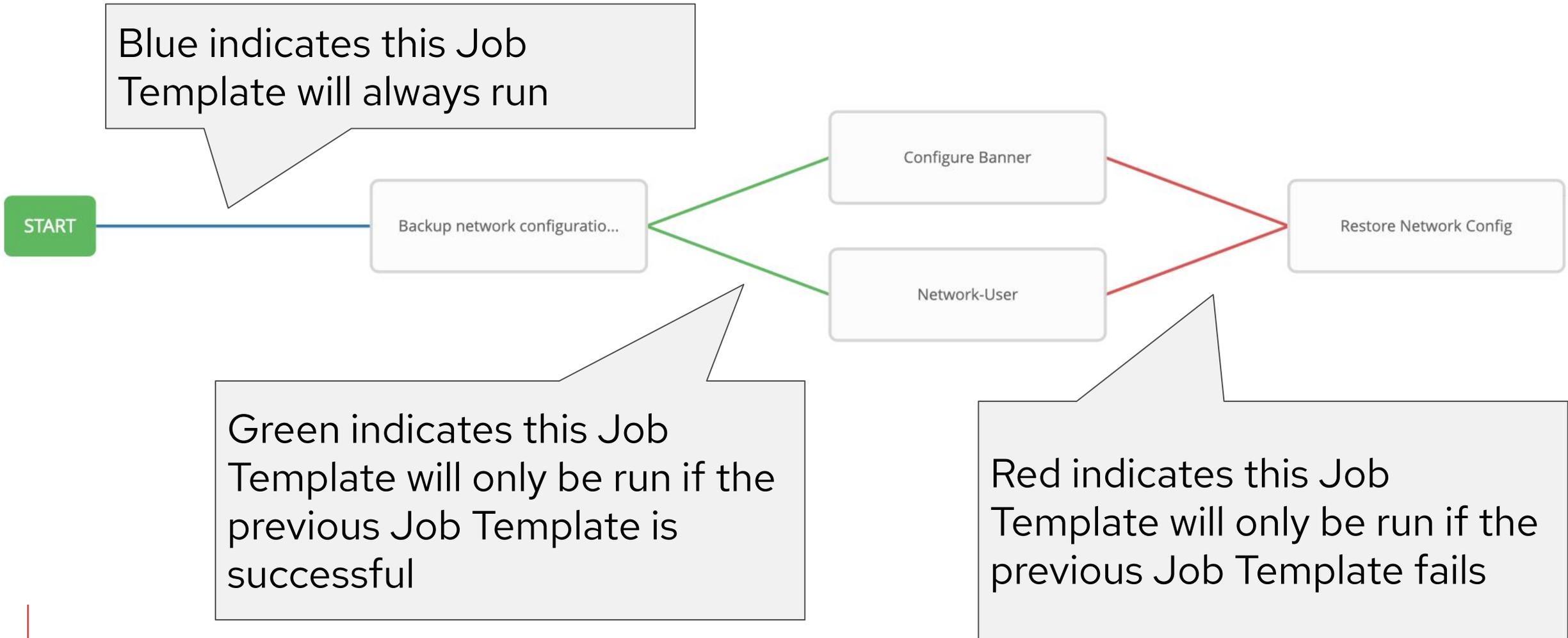
Workflow Visualizer

The workflow visualizer will start as a blank canvas.



Visualizing a Workflow

Workflows can branch out, or converge in.





Red Hat Ansible Automation Platform

Lab Time

Complete exercise **6-workflow** now in your lab environment

<https://red.ht/ansibleworkshop>

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

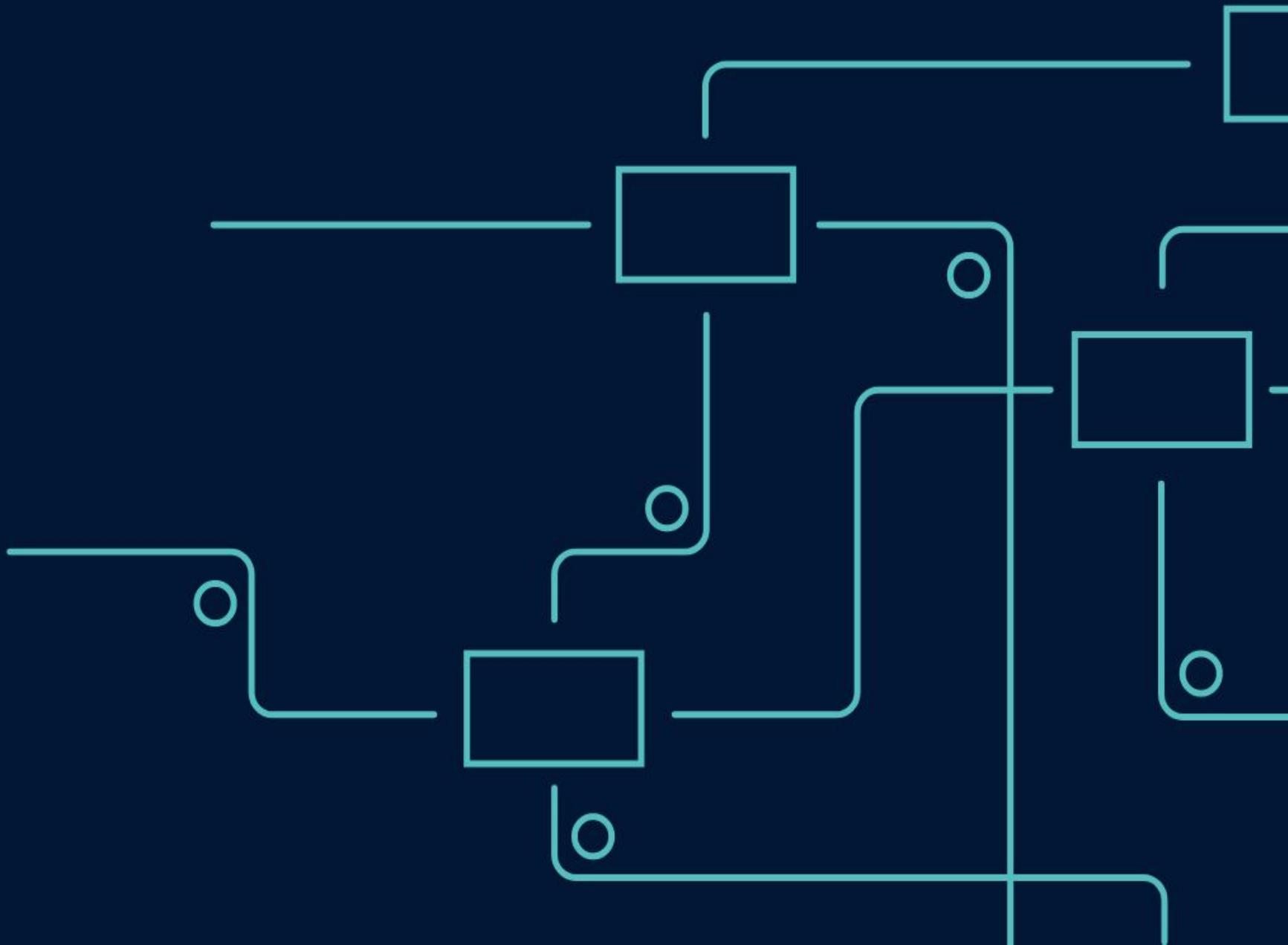
[Follow us @Ansible](#)

[Friend us on Facebook](#)

AnsibleFest 2021

Automate. Innovate. Accelerate.
September 29-30

[Register now](#)



Thank you



<http://bit.ly/3jZSbJH>

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/AnsibleAutomation](https://www.youtube.com/AnsibleAutomation)

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

 twitter.com/ansible

 github.com/ansible