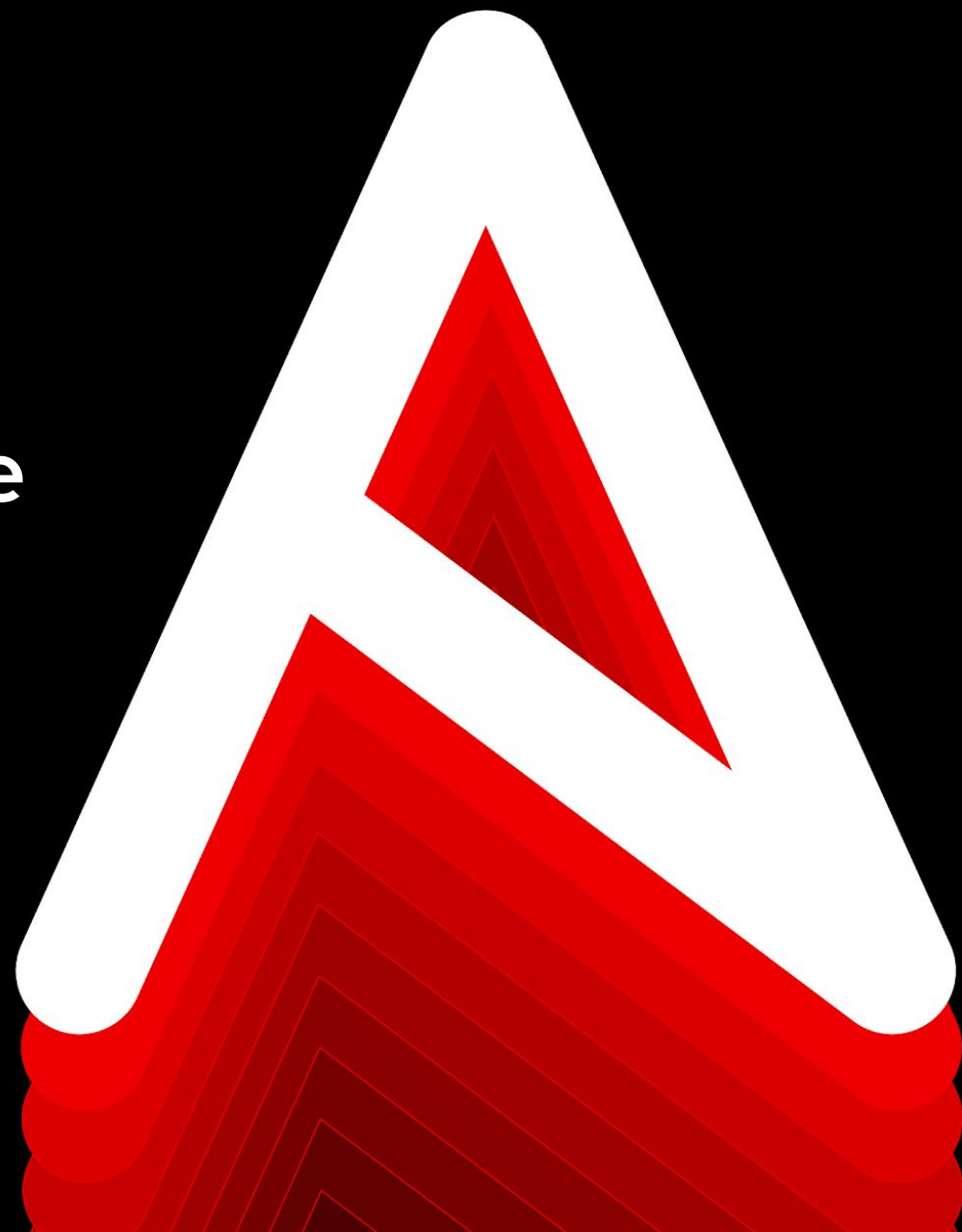


WS:D3

Network resiliency with Event Driven automation using Ansible

Learn how to supercharge your networks with AAP!

Slack Channel: #ac2-ws-d3
<https://bit.ly/ansibleautocon>



Lab Instructors



Ganesh Nalawade

Senior Principal Software Engineer

Red Hat Ansible

@Ganesh634



Nilashish Chakraborty

Principal Software Engineer

Red Hat Ansible

@NilashishC



Sean Cavanaugh

Technical Marketing Engineer

Red Hat Ansible

@IPvSean

NAF PRESENTS

AUTOCON 2

WORKSHOP SURVEY



NETWORKAUTOMATION.FORUM

Workshop Agenda

□ Workshop Introduction

□ Introduction to Ansible Network Automation

- Configuring Ansible Lightspeed in Visual Studio Code
- basics & creating inventory
- verify banner configuration
- creating the playbook
- execute the playbook
- adding variables
- using Ansible documentation
- network facts

□ Using Resource Modules

- verify SNMP configuration
- configuring SNMP
- state gathered

□ Introduction to Ansible Automation Platform

- Automation Controller
- creating an inventory
- creating your first project
- get to know credentials
- creating a job template
- configuring BGP (Part 1)
- configuring BGP (Part 2)
- creating a workflow job template

□ Introduction to Event-Driven Ansible

- ansible-rulebook
- ansible-rulebook with Kafka
- Introduction to Event-Driven Ansible Controller
- setting up telemetry
- setup rulebook and remediation
- automated remediations
- automated notifications
- chatops

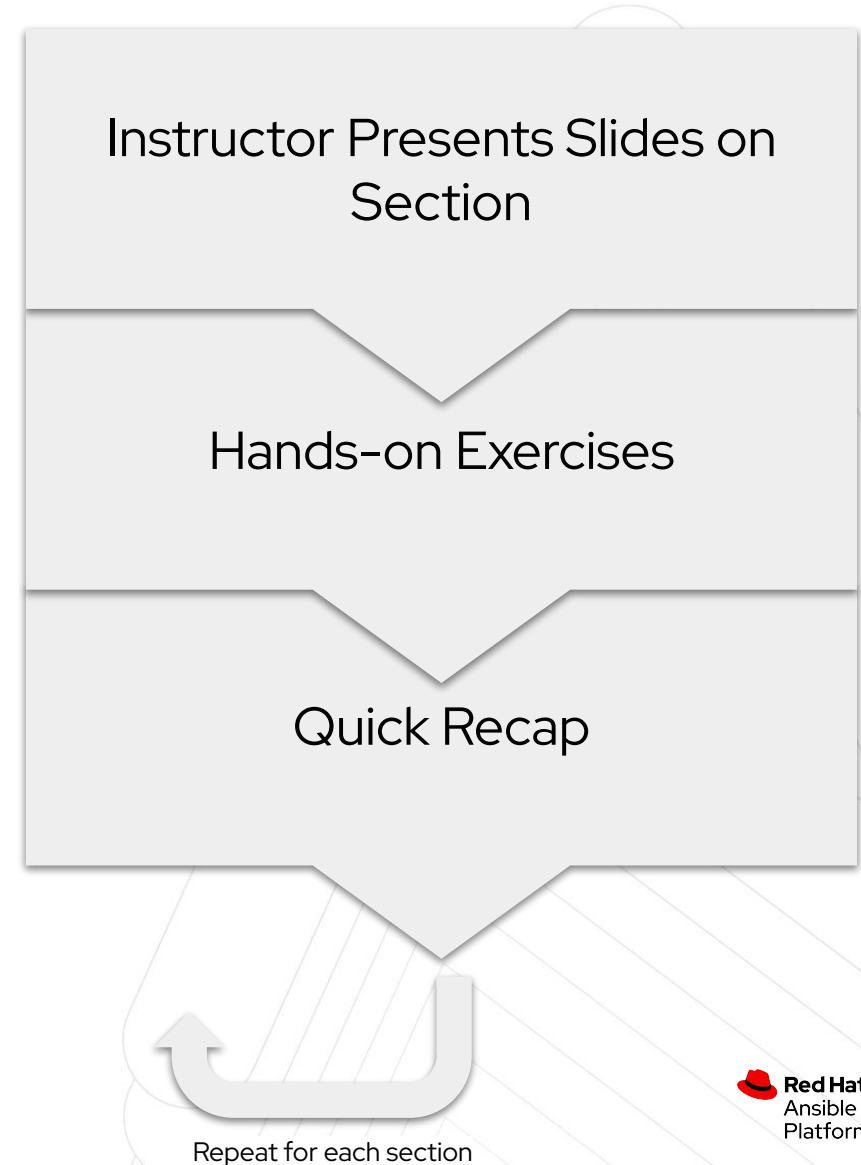
Housekeeping

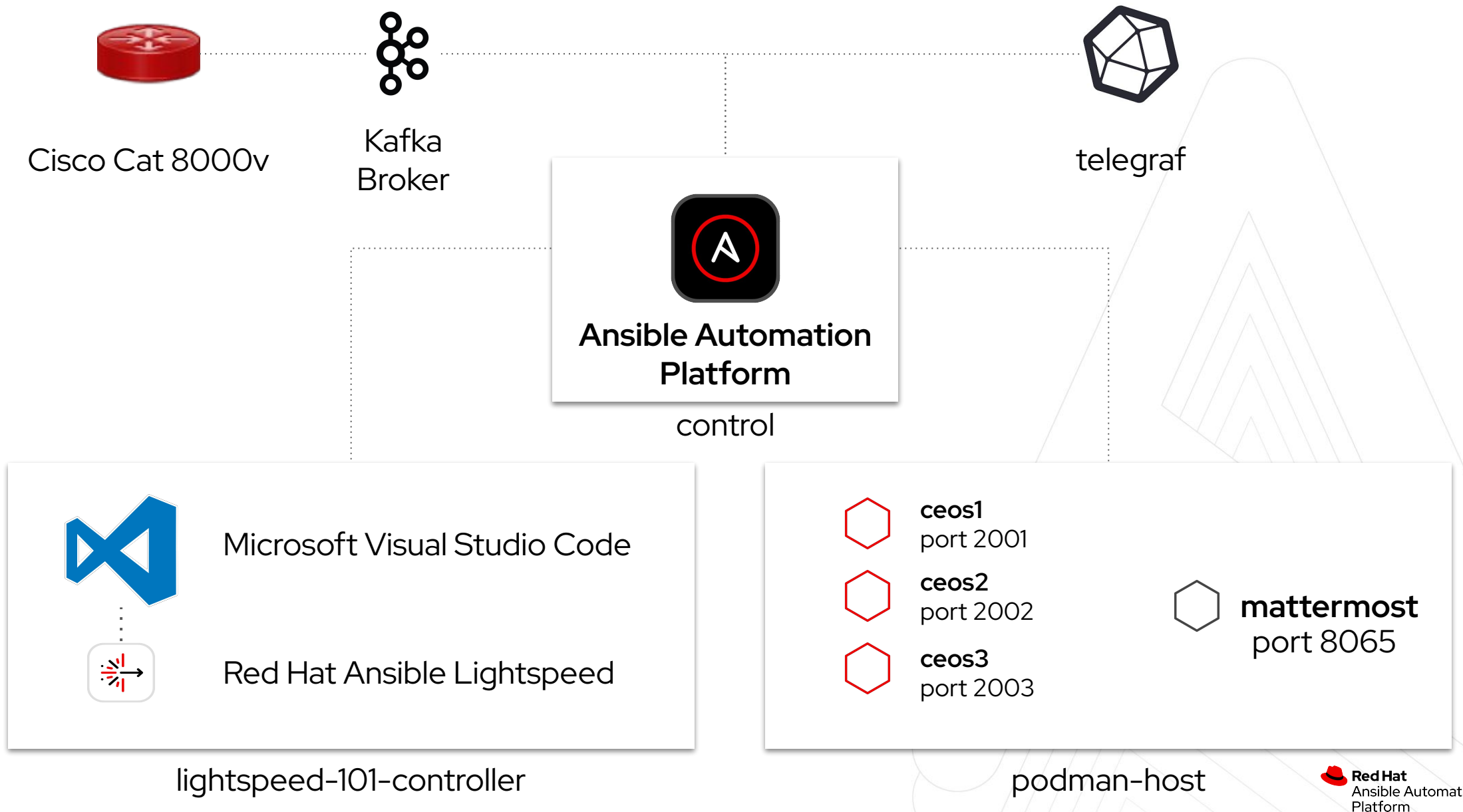
Understanding the format of this workshop

- Timing
- Breaks
- Takeaways



Section Flow





Section 1

Introduction to Ansible Network Automation

Topics Covered:

- ▶ Ansible Lightspeed
- ▶ Understanding an Ansible Playbook and Inventory

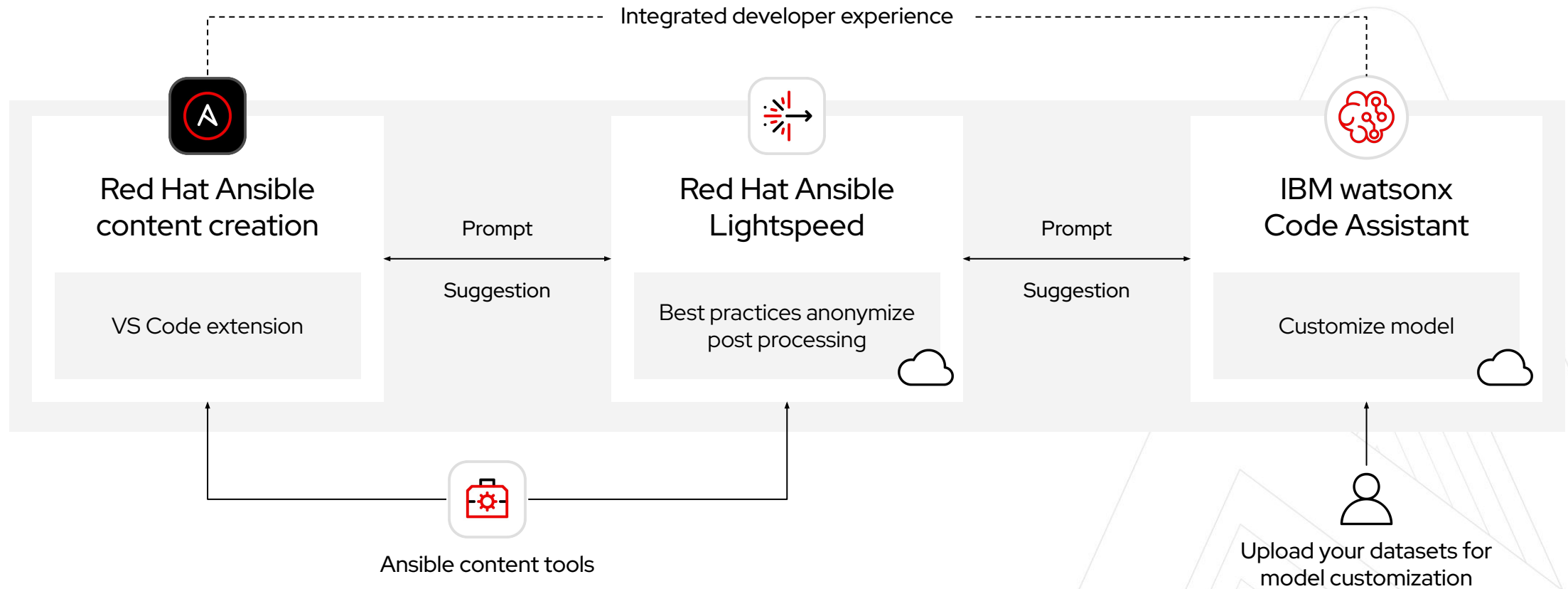


Ansible Lightspeed



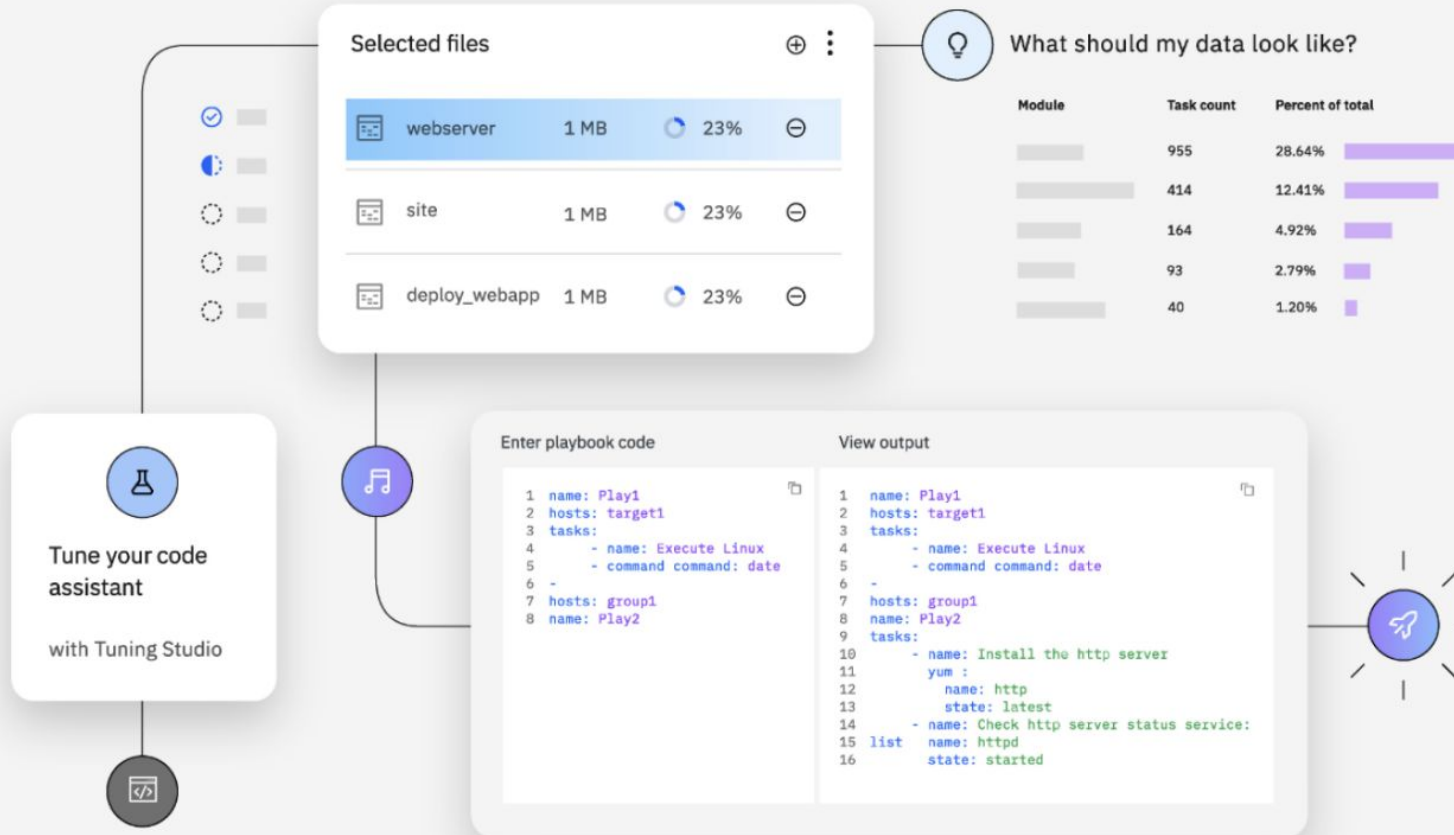
Topics Covered:

- ▶ Ansible Lightspeed components
- ▶ Ansible VS Code extension setup
- ▶ Ansible Playbook generation



Data security

- + Prompt/Suggestion data is encrypted in transit and at rest and is ephemeral
- + Customer metadata and customized models are stored in customer-owned IBM cloud object storage and are not shared with IBM, Red Hat, or any other customers



Create

Foundational model for Ansible

- + 3B parameter IBM watsonx Granite foundation model, trained on Ansible datasets, and infused with IBM and Red Hat automation expertise
- + Customer data is isolated to their IBM Cloud instance

The image displays three overlapping screenshots of the Red Hat Ansible Lightspeed web interface, illustrating its features for creating and refining Ansible playbooks.

Top Left Screenshot (Playbook generation): Shows a generated Ansible playbook in a dark-themed editor. The playbook includes tasks for installing Docker, adding the Docker repository, and running the Docker service. A tooltip labeled "Playbook generation" is visible.

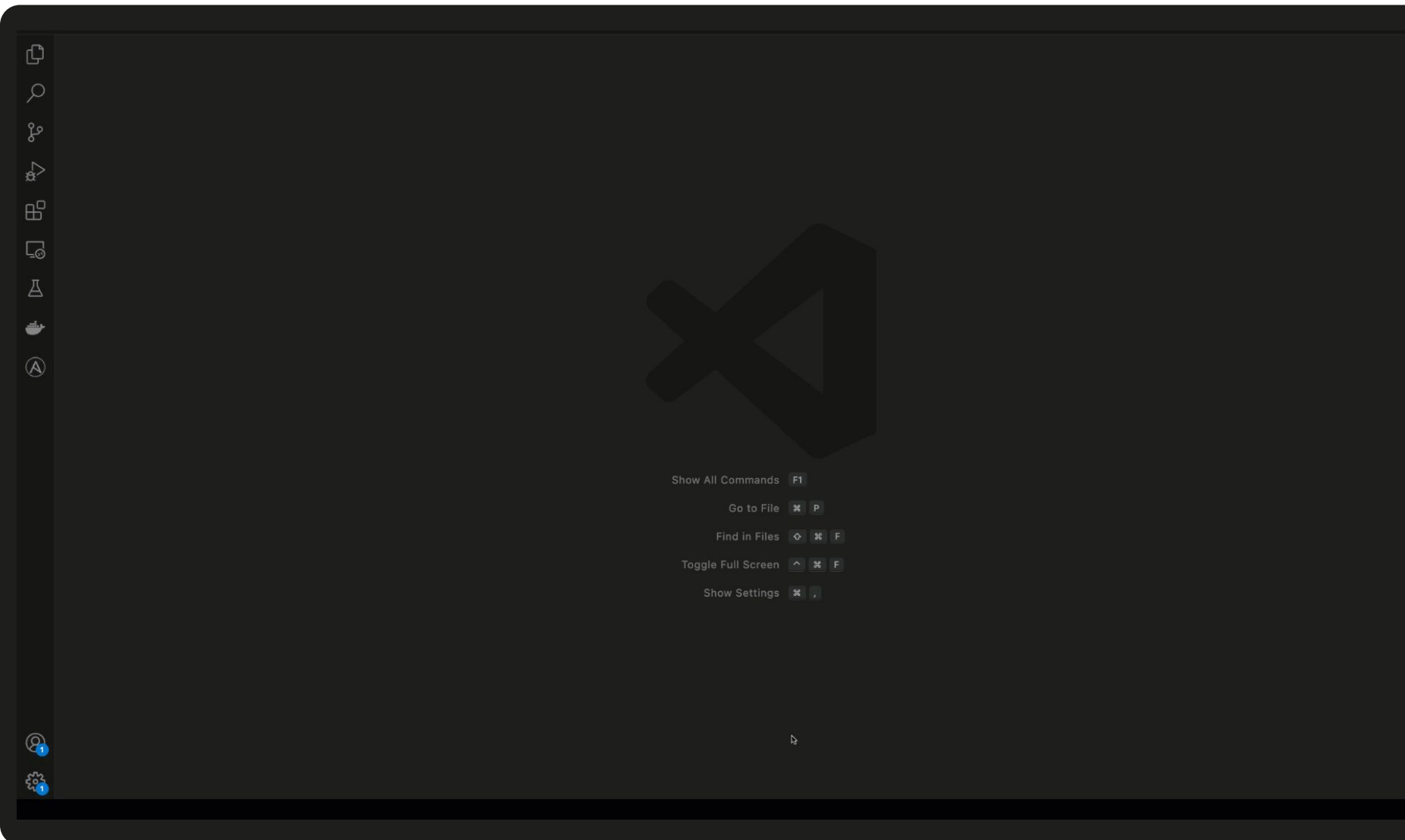
Top Right Screenshot (Guided chat): Shows the "Create a playbook" screen with a "Guided chat" button. The main area asks "What would you want the playbook to accomplish?" and provides a text input field with a blue arrow button. Below the input field, there are "Examples" of playbooks, such as "Create IIS websites on port 8080 and 8081 and open firewall."

Bottom Screenshot (Prompt refinement): Shows the "Create a playbook" screen with a "Prompt refinement" button. The main area asks "Does this look like the right playbook for you?" and provides a text input field with a blue arrow button. Below the input field, there are "Examples" of playbooks, such as "The steps this playbook will perform is as follows: 1. Install Docker on all hosts using the apt module. 2. Pull the nginx image from Docker Hub using the docker_image module. 3. Start a container from the nginx image using the docker_container module."

Create

Ansible Playbook generation

- + Guided chat to boost developer productivity
- + Playbook explanation to accelerate learning of syntax and structure
- + WCA 20B Granite model to generate high-quality recommendations



Create

Ansible Playbook generation

- + Guided chat to boost developer productivity
- + Playbook explanation to accelerate learning of syntax and structure
- + WCA 20B Granite model to generate high-quality recommendations

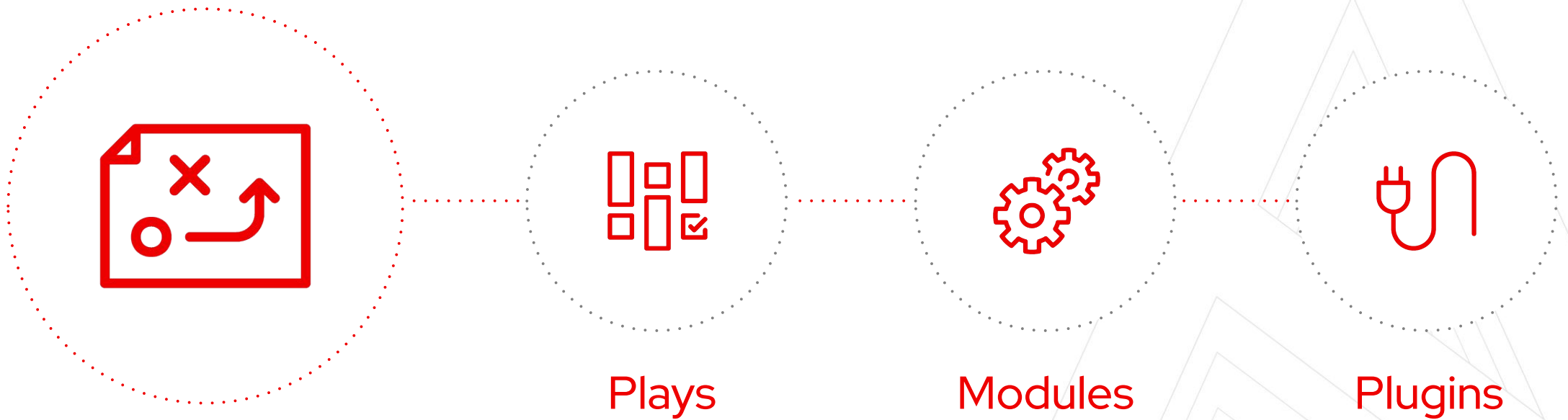
Ansible Basics



Topics Covered:

- ▶ Ansible Playbooks
- ▶ Ansible Inventory
- ▶ Ansible Collections

What makes up an Ansible playbook?



A Sample Ansible Playbook

```
---
- name: Configure VLANs
  hosts: cisco
  gather_facts: false
  tasks:
    - name: VLANs task
      cisco.nxos.vlans:
        config:
          - vlan_id: 5
            name: WEB
          - vlan_id: 10
```

- A playbook is a list of plays.
- Each play is a list of tasks.
- Tasks invoke modules.
- A playbook can contain more than one play.

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: configure vlans  
  hosts: switches  
  gather_facts: 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 Python, or Powershell for Windows setups. But can be of any language.

```
- name: VLANs task
  cisco.nxos.vlans:
    config:
      - vlan_id: 5
      - vlan_id: 10
```


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 filter plugins:

```
my_subnet: 192.168.1.0/24
{{ my_subnet | ipaddr('network') }}
renders 192.168.1.0

{{ my_subnet | ipaddr('prefix') }}
renders 24
```

Ansible Inventory

The systems that a playbook runs against



What are they?

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
```

Understanding Inventory - Groups

There is always a group called "**all**" by default

```
[cisco]
rtr1 ansible_host=18.220.156.59 private_ip=172.16.184.164
[arista]
rtr2 ansible_host=18.221.53.11 private_ip=172.17.229.213
rtr4 ansible_host=3.16.82.231 private_ip=172.17.209.186
[juniper]
rtr3 ansible_host=13.59.242.237 private_ip=172.16.39.75
```

Groups can be nested

```
[routers:children]
cisco
juniper
arista
```

Understanding Inventory - Variables

Host variables apply to the host and override group vars

```
[cisco]
rtr1 ansible_host=18.220.156.59 private_ip=172.16.184.164
[arista]
rtr2 ansible_host=18.221.53.11 private_ip=172.17.229.213
rtr4 ansible_host=3.16.82.231 private_ip=172.17.209.186
[juniper]
rtr3 ansible_host=13.59.242.237 private_ip=172.16.39.75

[cisco:vars]
ansible_user=ec2-user
ansible_network_os=ios
ansible_connection=network_cli
```

Group variables apply for all devices in that group

Collections

Simplified and consistent content delivery



What are they?

Collections are a directory structure containing automation content:

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



170+
certified content
collections



Infrastructure



Cloud



Network



Security



ARISTA



Check Point
SOFTWARE TECHNOLOGIES LTD



CYBERARK



FORTINET

Network Automation compared to servers

Module code is
executed locally on the
control node



Network Devices /
API Endpoints

Module code is copied
to the managed node,
executed, then
removed



Linux / Windows
Hosts

Network Connection Plugins

Use your vendor connection of choice

`ansible_connection`

- **`netconf`** - XML over netconf

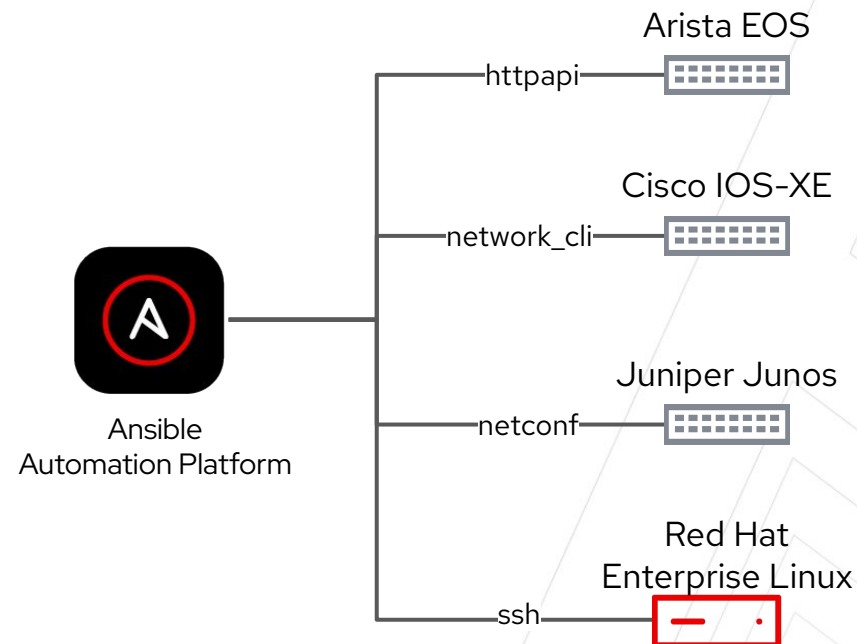
example: Juniper Junos

- **`network_cli`** - command line over SSH

example: Cisco IOS-XE, Arista EOS

- **`httpapi`** - vendor API

example: Arista eAPI, Cisco NX-API



<https://docs.ansible.com/ansible/latest/plugins/connection.html>

Running an Ansible Playbook

Using the latest `ansible-navigator` command



What is `ansible-navigator`?

`ansible-navigator` command line utility and text-based user interface (TUI) for running and developing Ansible automation content.

It replaces the previous command used to run playbooks “`ansible-playbook`”.

```
$ ansible-navigator run playbook.yml
```

ansible-navigator

Bye ansible-playbook, Hello ansible-navigator



How do I use ansible-navigator?

As previously mentioned, it replaces the ansible-playbook command.

As such it brings two methods of running playbooks:

- ▶ Direct command-line interface
- ▶ Text-based User Interface (TUI)

```
# Direct command-line interface method
$ ansible-navigator run playbook.yml -m stdout

# Text-based User Interface method
$ ansible-navigator run playbook.yml
```

Lab Time

Section 1 - Configuring Ansible Lightspeed & Introduction to Ansible Network Automation

 In this lab you will explore the lab environment, configure Ansible Lightspeed and go through basics of Network Automation using Ansible

 Approximate time: 35 mins

Section 2

Using Resource Modules

Topics Covered:

- ▶ What are Resource Modules?
- ▶ Why should you use them?



Network Automation Modules

How do we interact with network devices?

command



run arbitrary commands

facts



retrieve information

config



generic catch-all configuration
and templating

resource



read and configure specific
network resources

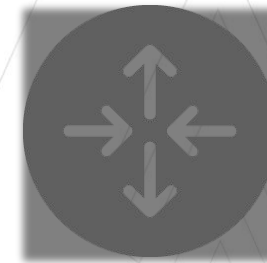
What's so special about Resource Modules?

Focus on re-usable structured data

```
interfaces:
- name: Vlan10
  description: SVI-1
  ip_forward: true
- name: Ethernet1/1
  description: INTF-1
  mode: layer3
  mtu: '1800'
  enabled: true
- name: Ethernet1/2
  description: INTF-2
  speed: '10000'
- name: Ethernet1/3
  mtu: '1500'
- name: Ethernet1/4
  mtu: '1500'
```



Resource
Module
(using **action states**)



Native
Configuration
(**on-box**)

Rendering native configuration commands *from structured data* using
Resource Modules

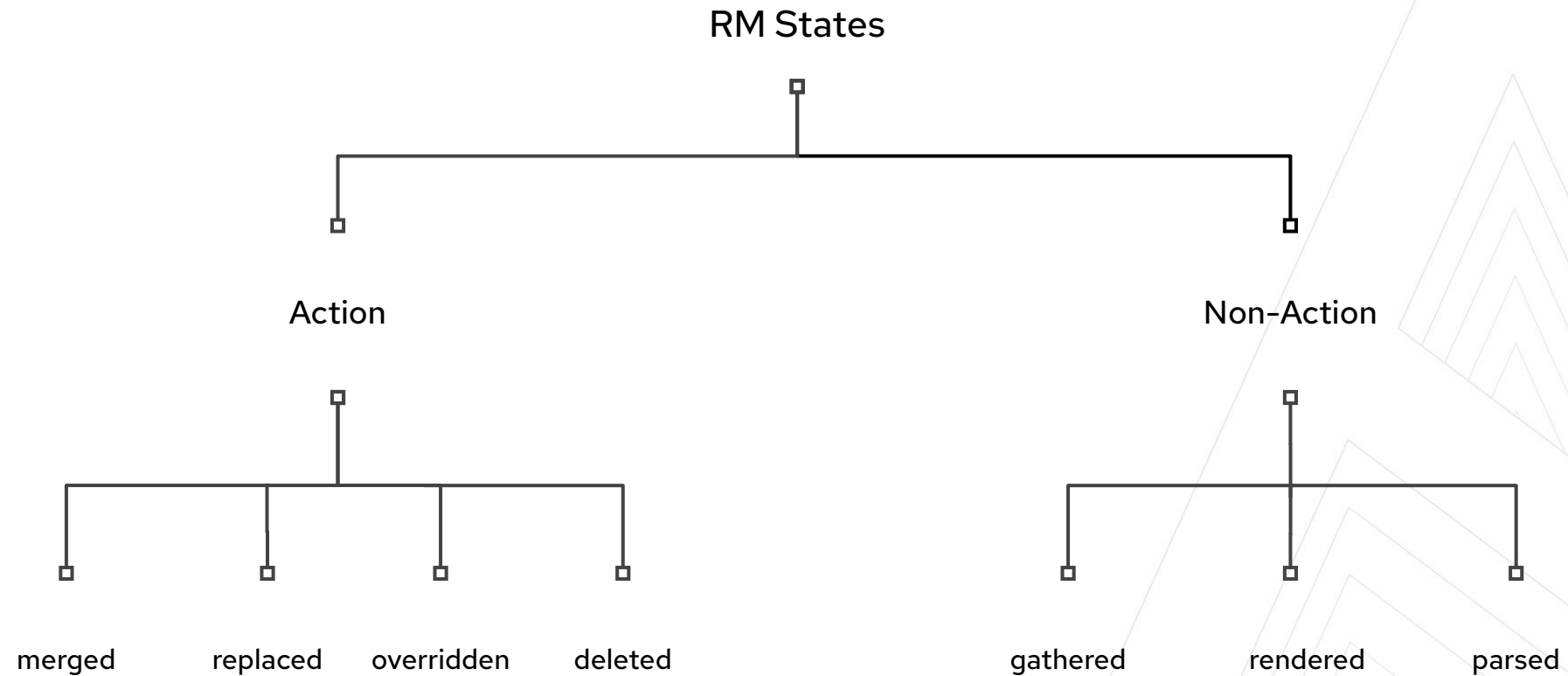
Focus on re-usable structured data (cont.)

```
interfaces:
- name: Vlan10
  description: SVI-1
  ip_forward: true
- name: Ethernet1/1
  description: INTF-1
  mode: layer3
  mtu: '1800'
  enabled: true
- name: Ethernet1/2
  description: INTF-2
  speed: '10000'
- name: Ethernet1/3
  mtu: '1500'
- name: Ethernet1/4
  mtu: '1500'
```



Converting native (on-box) configuration *to structured data* using
Resource Modules

Supports multiple states of operation



Enhanced Return Values

```
interface Ethernet1/1
  no switchport
  ip address 192.168.1.1/24
```

(existing on-box configuration)

```
name: Update IP address
cisco.nxos.l3_interfaces:
  config:
    - name: Ethernet1/1
      ipv4:
        - address: 192.168.1.1/24
      ipv6:
        - address: fc00::100/64
```

(task to add an IPv6 address)

```
before:
- name: Ethernet1/1
  ipv4:
    - address: 192.168.1.1/24
```

```
commands:
- interface Ethernet1/1
- ipv6 address fc00::100/64
```

```
after:
- name: Ethernet1/1
  ipv4:
    - address: 192.168.1.1/24
  ipv6:
    - address: fc00::100/64
```

Bidirectional Capable

- **name:** **grab nxos facts**
cisco.nxos.nxos_facts:
 gather_subset: ['!all', '!min']
 gather_network_resources: l3_interfaces
- **name:** **ensure that the IP address information is accurate**
cisco.nxos.nxos_l3_interfaces:
 config: "{ { ansible_network_resources['l3_interfaces'] } }"
 register: result
- **name:** **ensure config did not change**
assert:
 that: not result.changed



Round trip gathered facts to the corresponding Resource Module

Lab Time

Section 2 - Using Resource Modules

 In this lab you will learn how to use Network Resource Modules and push the playbooks you have written to a Github repository.

 Approximate time: 35 mins

Section 3

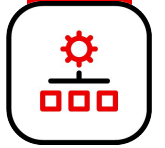
Introduction to Ansible Automation Platform

Topics Covered:

- ▶ What is Automation controller?
- ▶ Operationalizing Ansible Automation



Automation controller. Define, operate, and delegate.

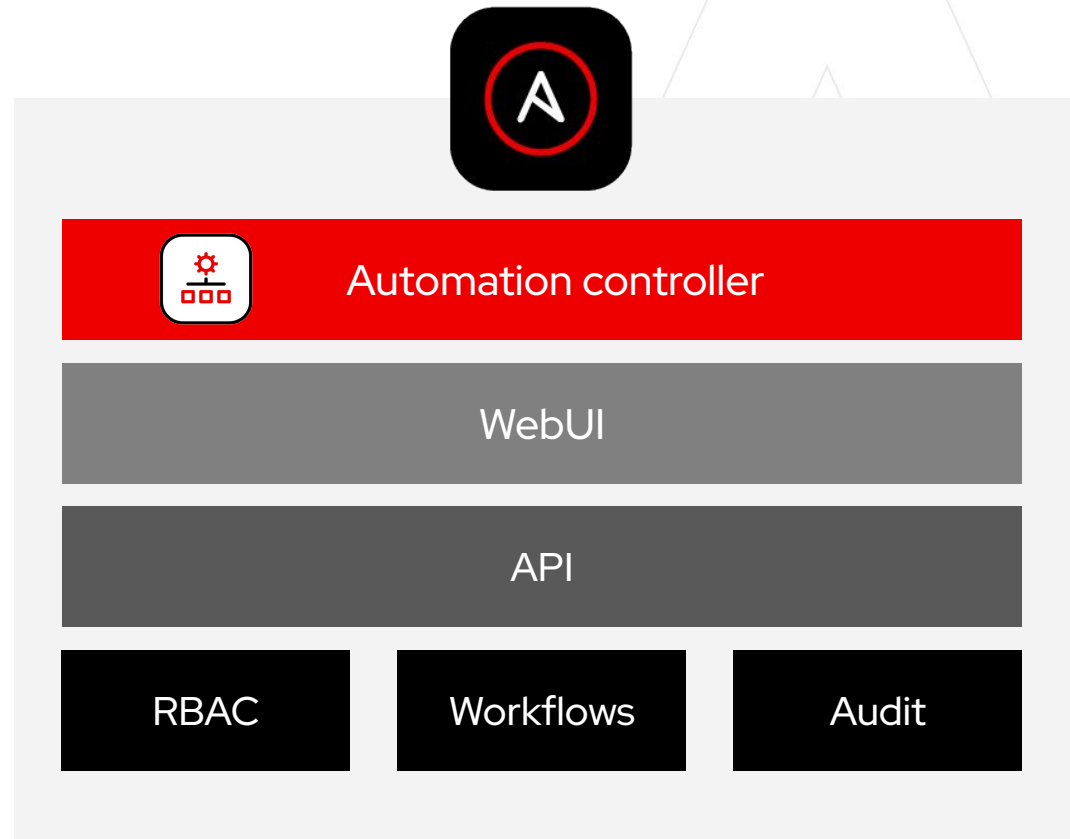


What is it?

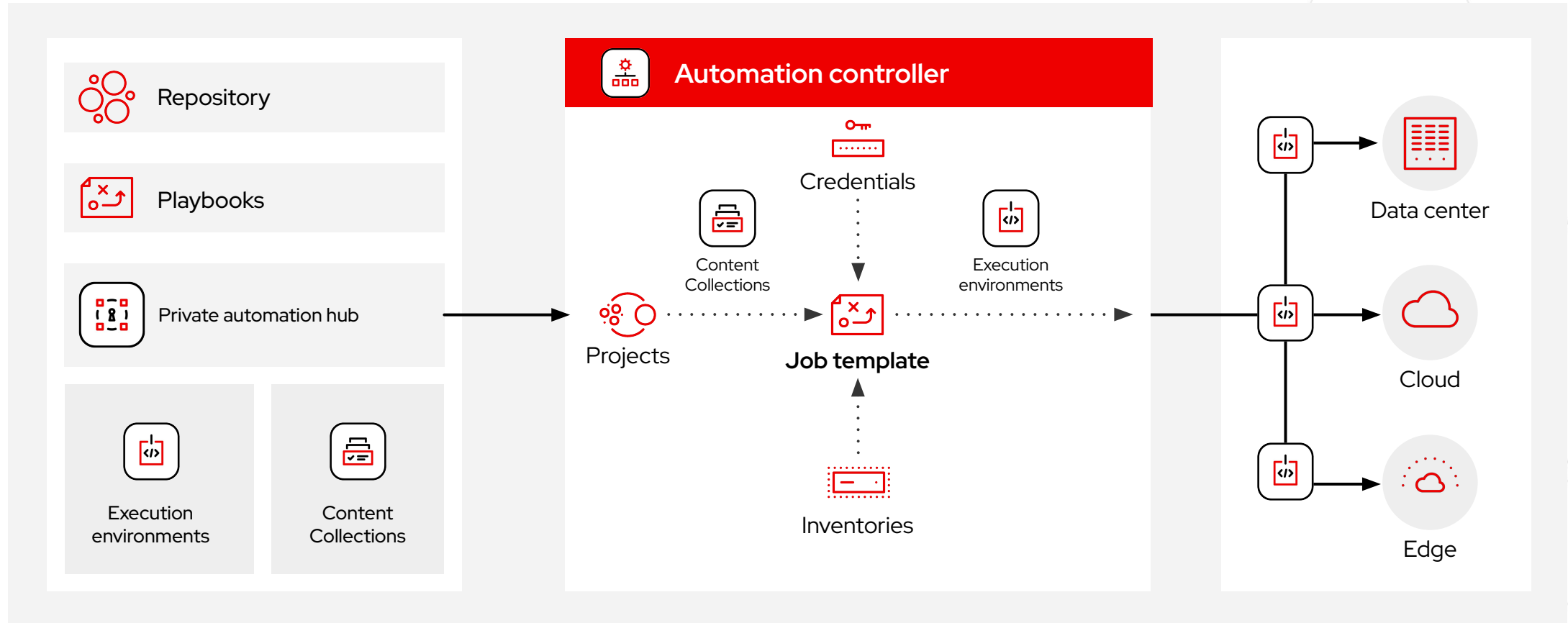
Automation controller is the Ansible Automation Platform control plane which enables users to define, operate, and delegate automation across their enterprise

Automation controller provides:

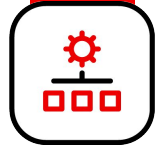
- ▶ WebUI and API
- ▶ Role-based access control
- ▶ Powerful workflows
- ▶ Centralized logging
- ▶ Credential management
- ▶ Push-button automation



Defining how your content runs in **automation controller**.



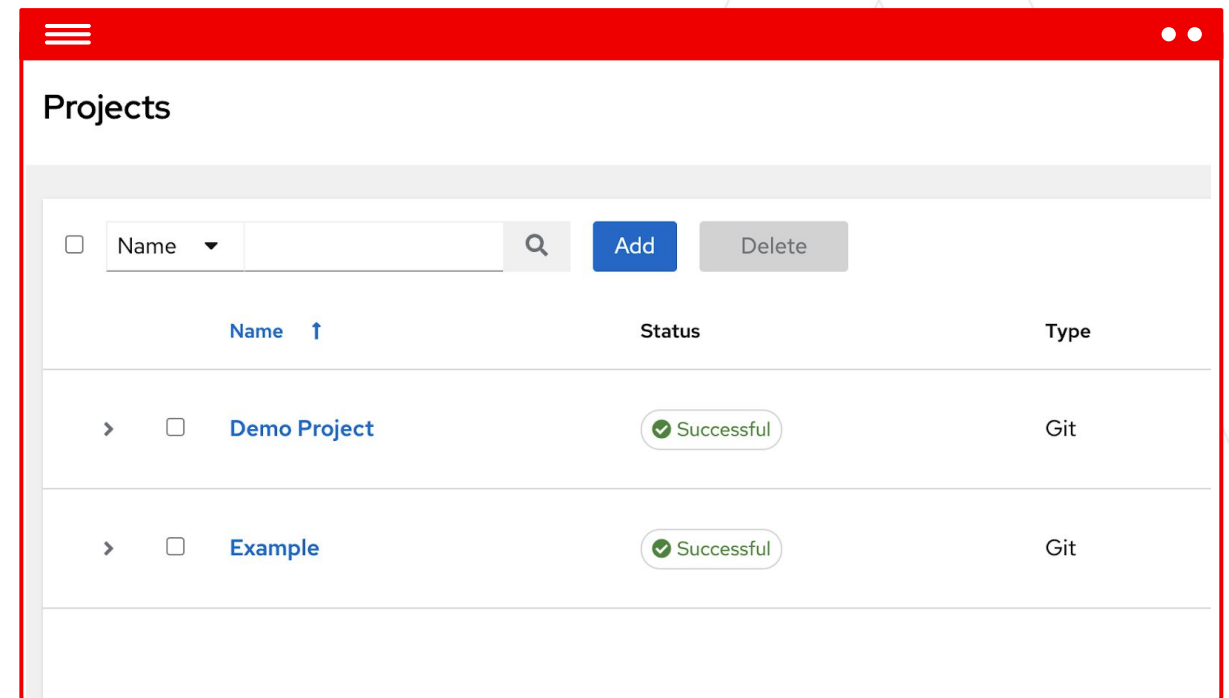
Projects. Adding your automation content to controller.



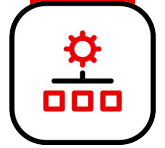
What is it?

Logical collection of your playbooks:

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



Inventories. What do I want to run my automation on?



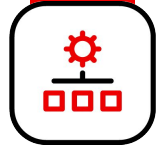
What is it?

Collection of endpoints against which jobs may be launched

- ▶ Multiple inventory sources supported
- ▶ Dynamic endpoint discovery
- ▶ Logically group endpoints by metadata or user-defined filters
- ▶ Granular RBAC permissions

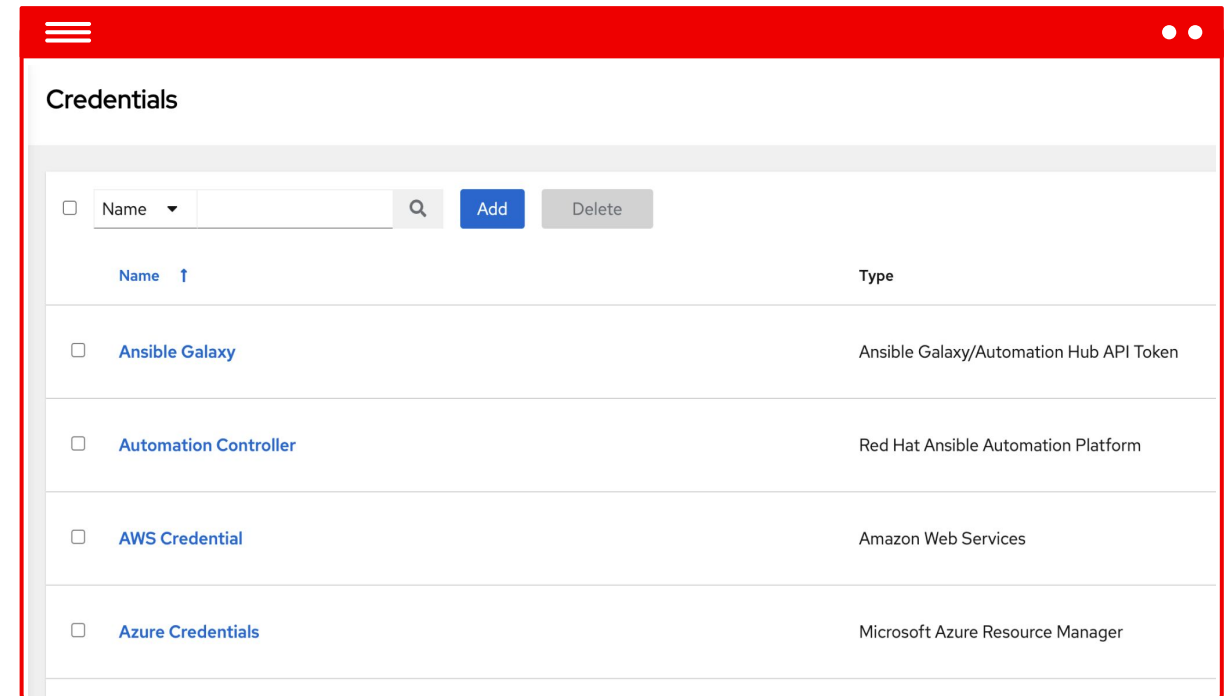
Name	Status	Type	Organization	Actions
☐ AWS Inventory	Disabled	Inventory	ACME Corp	
☐ Azure Inventory	Disabled	Inventory	ACME Corp	
☐ Data center	Disabled	Inventory	ACME Corp	
☐ Remote Office A	Disabled	Inventory	ACME Corp	

Credentials. **Securing resource and endpoint access.**

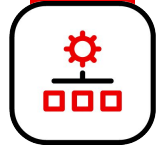


What is it?

- ▶ Securely manage credentials needed for automation resources
- ▶ Multiple credential types supported
- ▶ Integrate external secret management systems
- ▶ Create custom credential types and plugins
- ▶ Use RBAC to govern access
- ▶ Actual credential never exposed



Job Templates. Bringing it all together.



What is it?

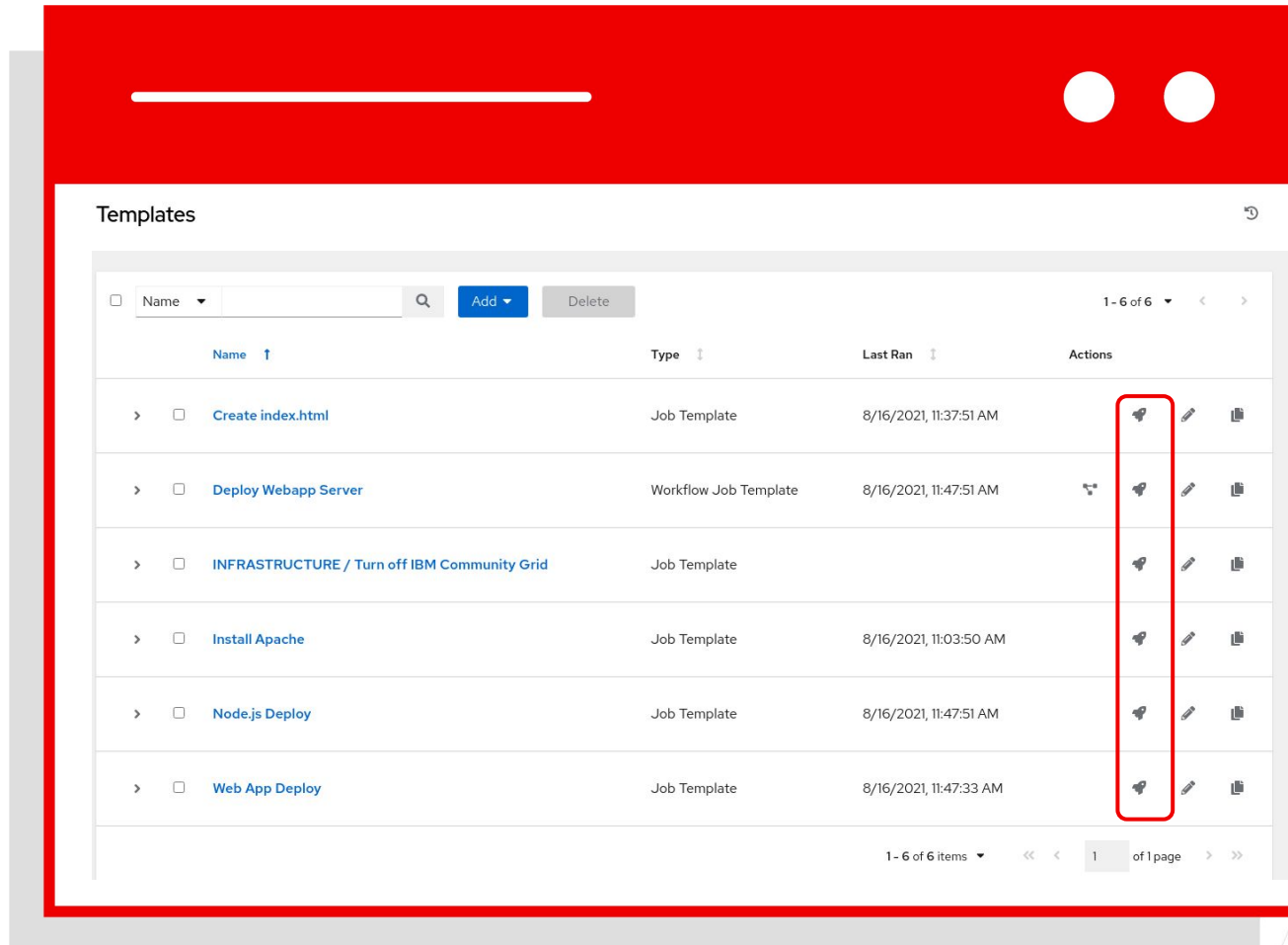
- ▶ Define and standardize running automation
- ▶ Reusable and shareable
- ▶ Leverage agile practices, such as GitOps and event-driven automation

The screenshot displays the 'Job template with slicing' configuration page in the Red Hat Ansible Automation Platform. The interface features a red header bar with a menu icon and window controls. Below the header, there are tabs for 'DETAILS', 'PERMISSIONS', 'NOTIFICATIONS', 'COMPLETED JOBS', 'SCHEDULES', and 'ADD SURVEY'. The 'DETAILS' tab is active, showing various configuration fields:

- * NAME:** 'Job template with slicing'
- DESCRIPTION:** (Empty text field)
- * JOB TYPE:** 'Run' (with a 'PR' checkbox)
- * INVENTORY:** 'King PLC' (with a search icon and 'PROMPT ON LAUNCH' checkbox)
- * PROJECT:** 'Project from Git' (with a search icon)
- * PLAYBOOK:** 'gen_host_status.yml'
- CREDENTIALS:** (Empty text field with search icon and 'PROMPT ON LAUNCH' checkbox)
- FORKS:** '0' (with up/down arrows)
- LIMIT:** (Empty text field with 'PR' checkbox)
- * VERBOSITY:** '0 (Normal)' (dropdown menu with 'PROMPT ON LAUNCH' checkbox)
- JOB TAGS:** (Empty text field with search icon and 'PROMPT ON LAUNCH' checkbox)
- SKIP TAGS:** (Empty text field with 'PR' checkbox)
- LABELS:** (Empty text field)
- INSTANCE GROUPS:** (Empty text field with search icon)
- JOB SLICING:** '1'
- TIMEOUT:** '0' (with up/down arrows)
- SHOW CHANGES:** (Toggle switch, currently off, with 'PROMPT ON LAUNCH' checkbox)
- OPTIONS:**
 - ☐ ENABLE PRIVILEGE ESCALATION
 - ☐ ENABLE PROVISIONING CALLBACKS
 - ☐ ENABLE WEBHOOK
 - ☒ ENABLE CONCURRENT JOBS
 - ☐ ENABLE FACT CACHE

Executing an existing **Job Template**

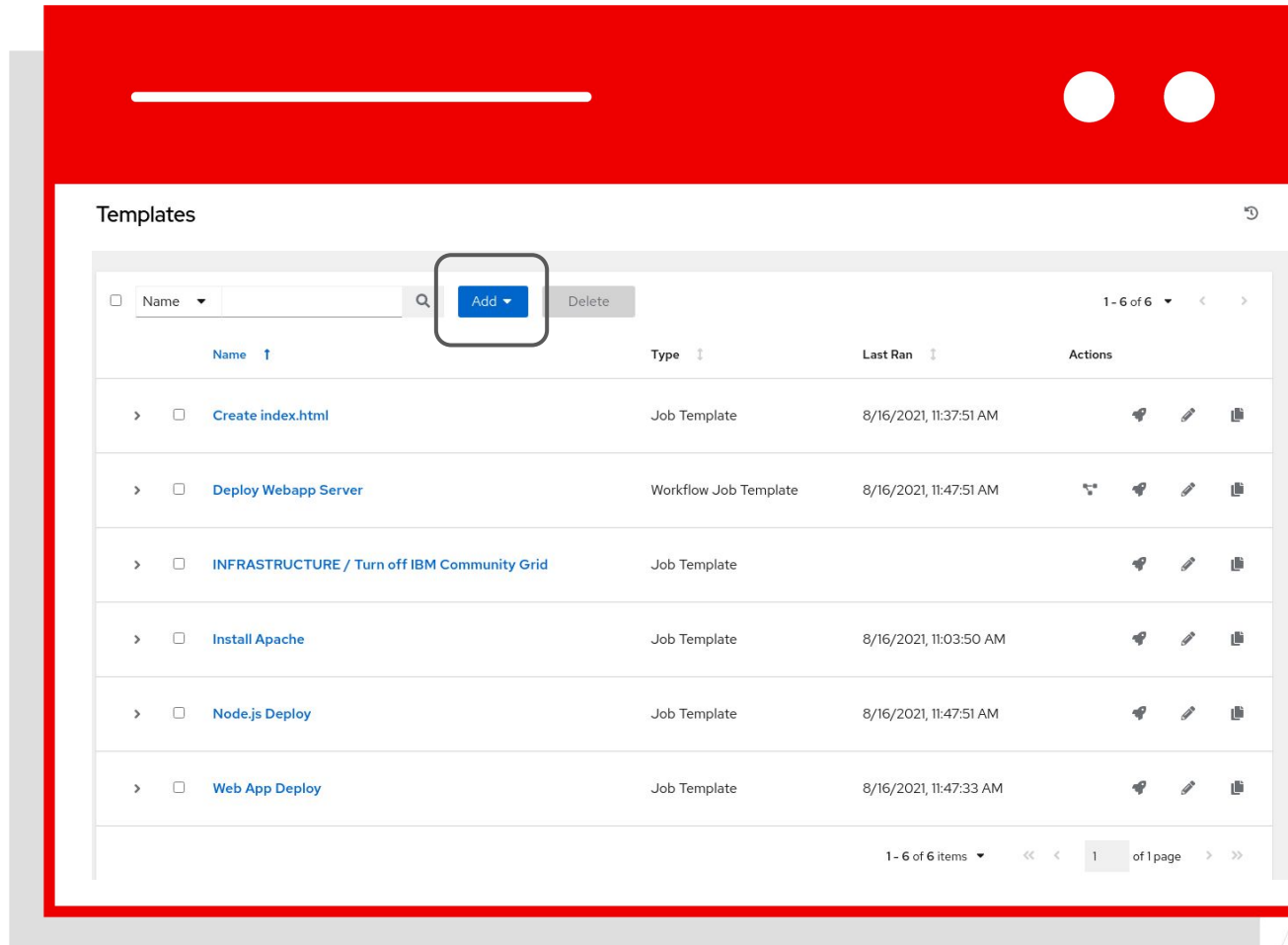
Job Templates can be found and created by clicking the **Templates** button under the *Resources* section on the left menu.



Name	Type	Last Ran	Actions
> ☐ Create index.html	Job Template	8/16/2021, 11:37:51 AM	  
> ☐ Deploy Webapp Server	Workflow Job Template	8/16/2021, 11:47:51 AM	  
> ☐ INFRASTRUCTURE / Turn off IBM Community Grid	Job Template		  
> ☐ Install Apache	Job Template	8/16/2021, 11:03:50 AM	  
> ☐ Node.js Deploy	Job Template	8/16/2021, 11:47:51 AM	  
> ☐ Web App Deploy	Job Template	8/16/2021, 11:47:33 AM	  

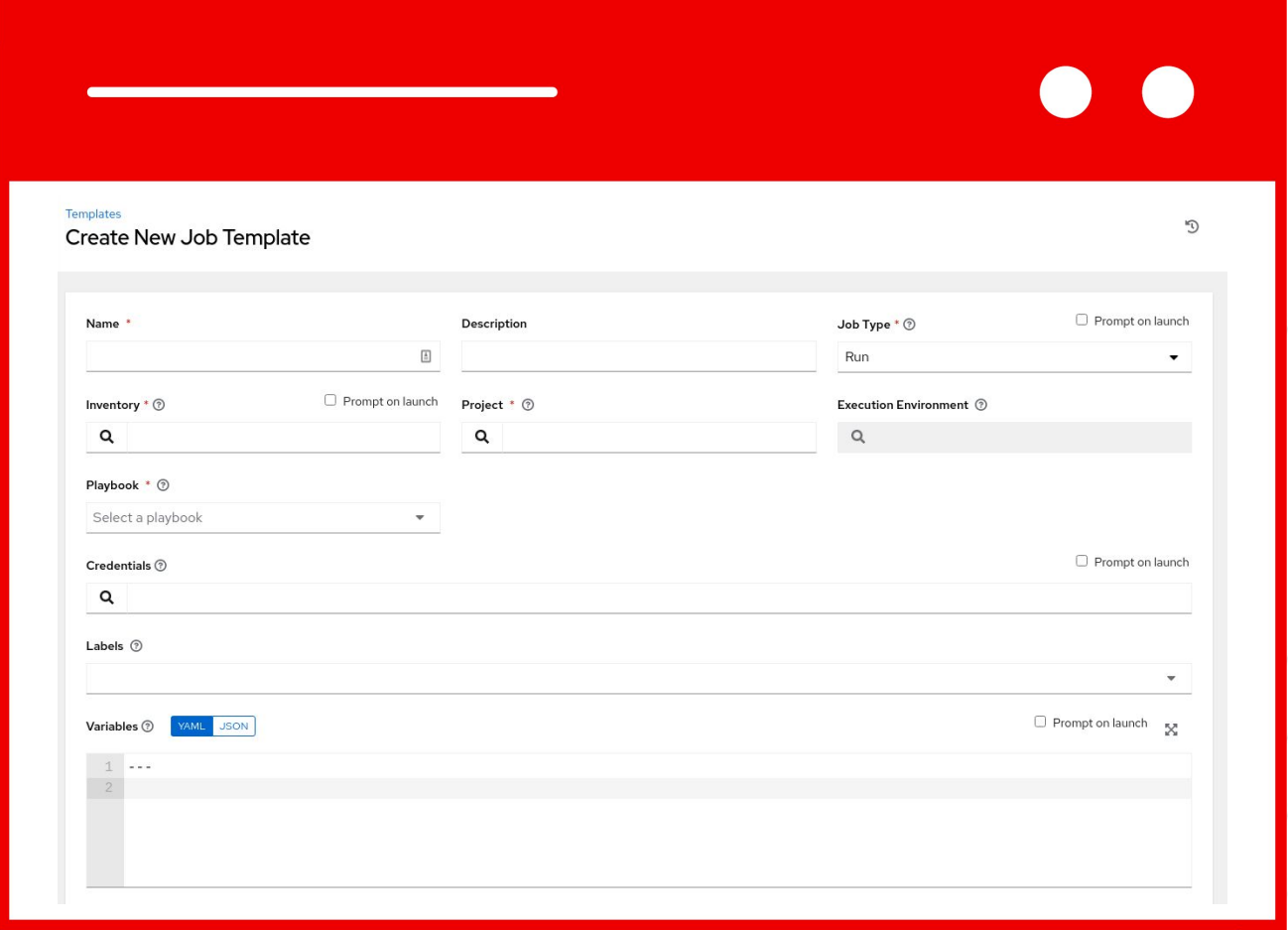
Creating a new Job Template

New Job Templates can be created by clicking the **Add button**



Creating a new Job Template

This **New Job Template** window is where the inventory, project and credential are assigned. The red asterisk * means the field is required .



Templates

Create New Job Template

Name *

Description

Job Type * ☐ Prompt on launch

Inventory * ☐ Prompt on launch

Project * ☐ Prompt on launch

Execution Environment

Playbook *

Credentials ☐ Prompt on launch

Labels

Variables ☐ Prompt on launch

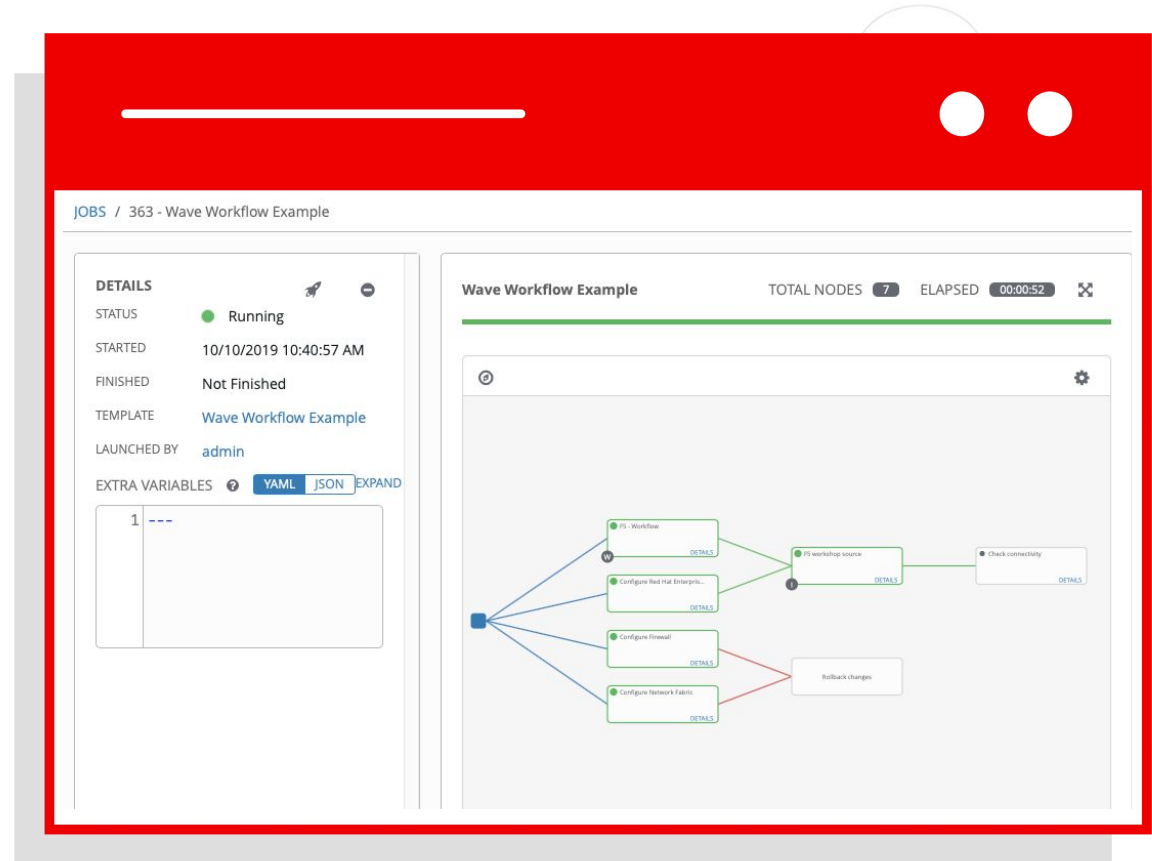
YAML JSON

```
1 ---
2
```

Workflows

Combine automation to create something bigger

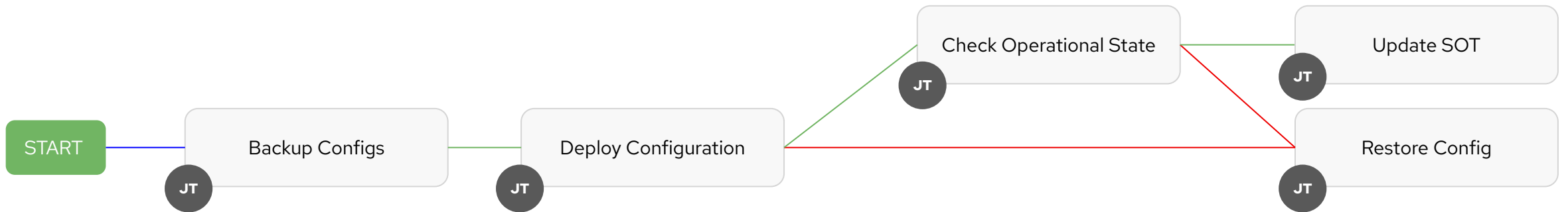
- ▶ Workflows enable the creation of powerful holistic automation, chaining together multiple pieces of automation and events.
- ▶ Simple logic inside these workflows can trigger automation depending on the success or failure of previous steps.



Ansible Automation Platform

Using workflows to enhance your automation

WORKFLOW VISUALIZER | Operational State Workflow



CLOSE

SAVE

Lab Time

Section 3 - Using Automation Controller

 In this lab you will explore the Ansible Automation Platform 2.5 unified UI and learn how to launch the playbooks you have written in the previous exercise in Automation Controller.

 Approximate time: 40 mins

Section 4

Event-Driven Ansible

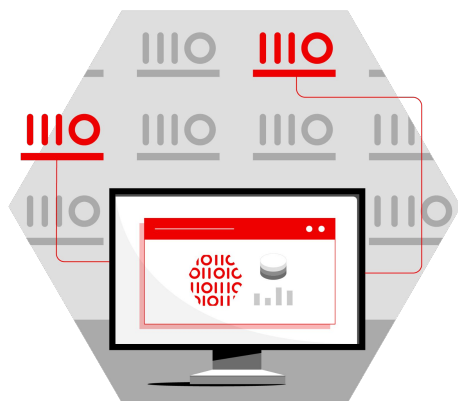
Topics Covered:

- ▶ Introduction to Event-Driven Ansible (EDA)
- ▶ Writing your first Ansible rulebook
- ▶ Learn how to use EDA Controller



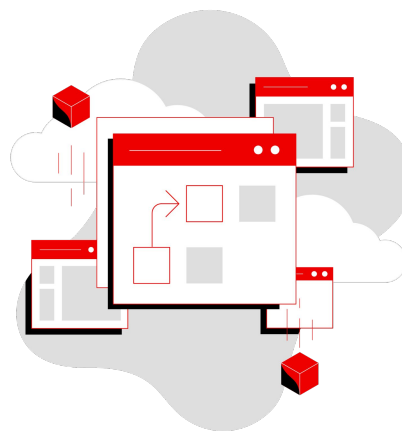
Key building blocks in Event-Driven Ansible

Simple, powerful, agentless



Sources

All the sources of event data you want to use



Rules

What you will create using Event-Driven Ansible®

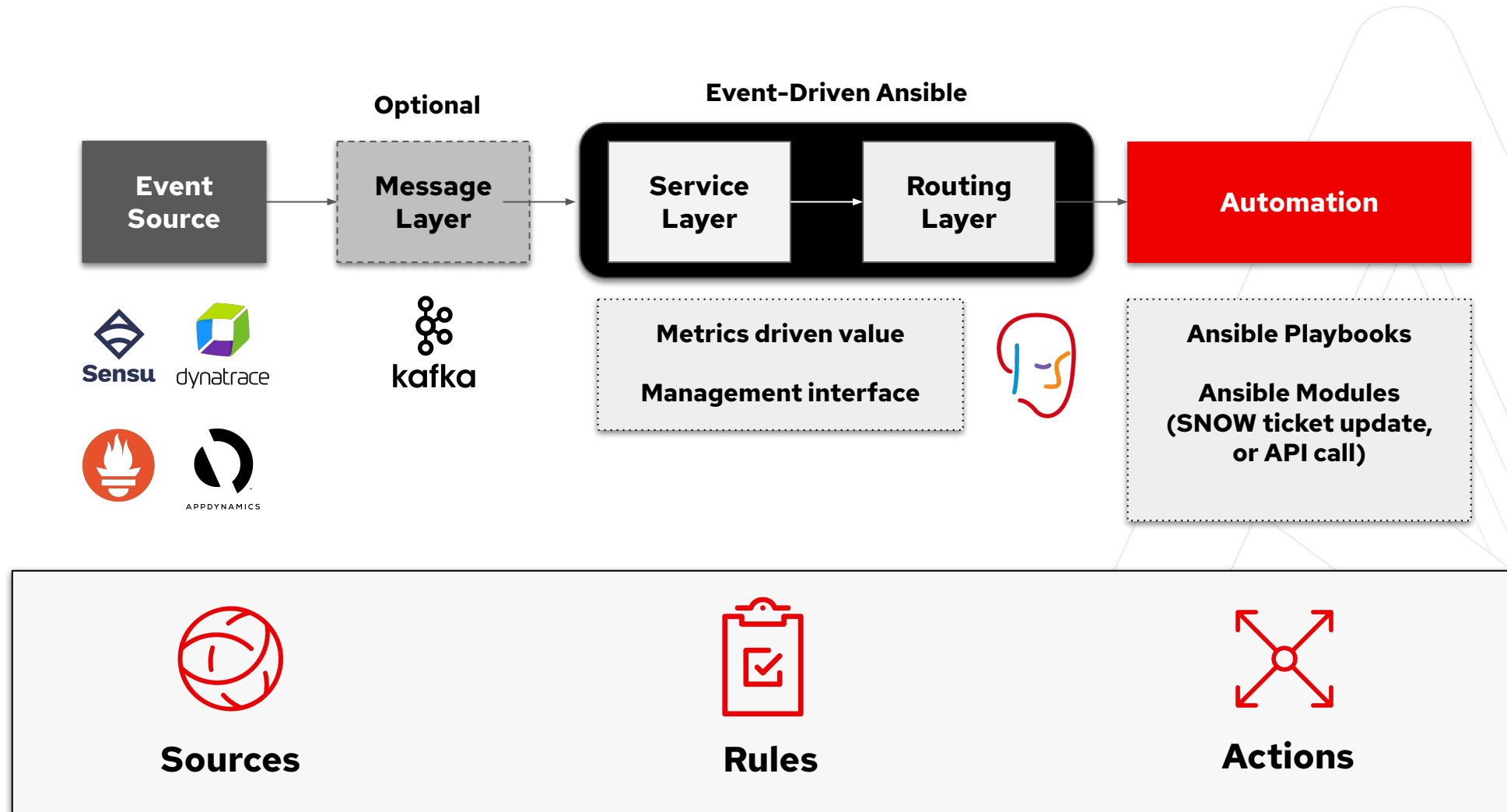


Actions

When a condition or event is met, the Ansible Rulebook executes

Ansible Rulebooks contain the source of the event, as well as the instructions on what steps to perform when a certain condition is met—and it is all very flexible.

Execution layers of Event Driven Ansible



Rules

Executing actions from matched conditions

▶ Rules consist of conditional statements

- ▶ EDA uses rules to trigger action
- ▶ Rules can have single or multiple actions
- ▶ Rules are organized into Rulesets

▶ Conditional mapping of actions to events

- ▶ Simple YAML structure for logical conditions
- ▶ Throttle events to mitigate event storms
- ▶ Events can trigger different types of actions:
 - ▶ *run_playbook, run_template, run_module*
 - ▶ *set_fact, post_event, print_event*
 - ▶ *retract_fact, shutdown, debug*

```
rules:
  - name: A rule with 1 action
    condition: event.fields.outage == true
    throttle:
      once_within: 30 minutes
      group_by_attributes:
        - event.tags.source
    action:
      run_playbook:
        name: remediate_outage.yaml

  - name: A rule with multiple actions
    condition: event.fields.outage == true
    actions:
      - run_playbook:
          name: remediate_outage.yaml
      - run_job_template:
          name: report_devops_team.yaml
          organization: ProdOrg
```

Rulesets

Rulesets: from source to action

- ▶ **Rulesets require:**
 - ▶ A unique name
 - ▶ A defined event source(s)
 - ▶ Hosts similar to Ansible Playbooks
 - ▶ A list of defined rules
- ▶ **Set execution_strategy**
 - ▶ Run actions in parallel or sequentially
- ▶ **Gather facts from hosts and upload to rules engine**
 - ▶ Use this data in match conditions

```
- name: My unique ruleset
hosts: all
execution_strategy: sequential
sources:
  - name: Listen for events from Kafka
    ansible.eda.kafka:
      host: 192.168.1.105
      port: 9092
      topic: eda
  - name: Listen for events from webhook
    ansible.eda.webhook:
      host: 0.0.0.0
      port: 5000
rules:
  - name: A rule with 1 action
    condition: event.fields.outage == true
    throttle:
      once_within: 30 minutes
      group_by_attributes:
        - event.tags.source
    action:
      run_playbook:
        name: remediate_outage.yaml
```

Rulebooks

Ansible Playbook has many plays, Ansible Rulebook has many rulesets

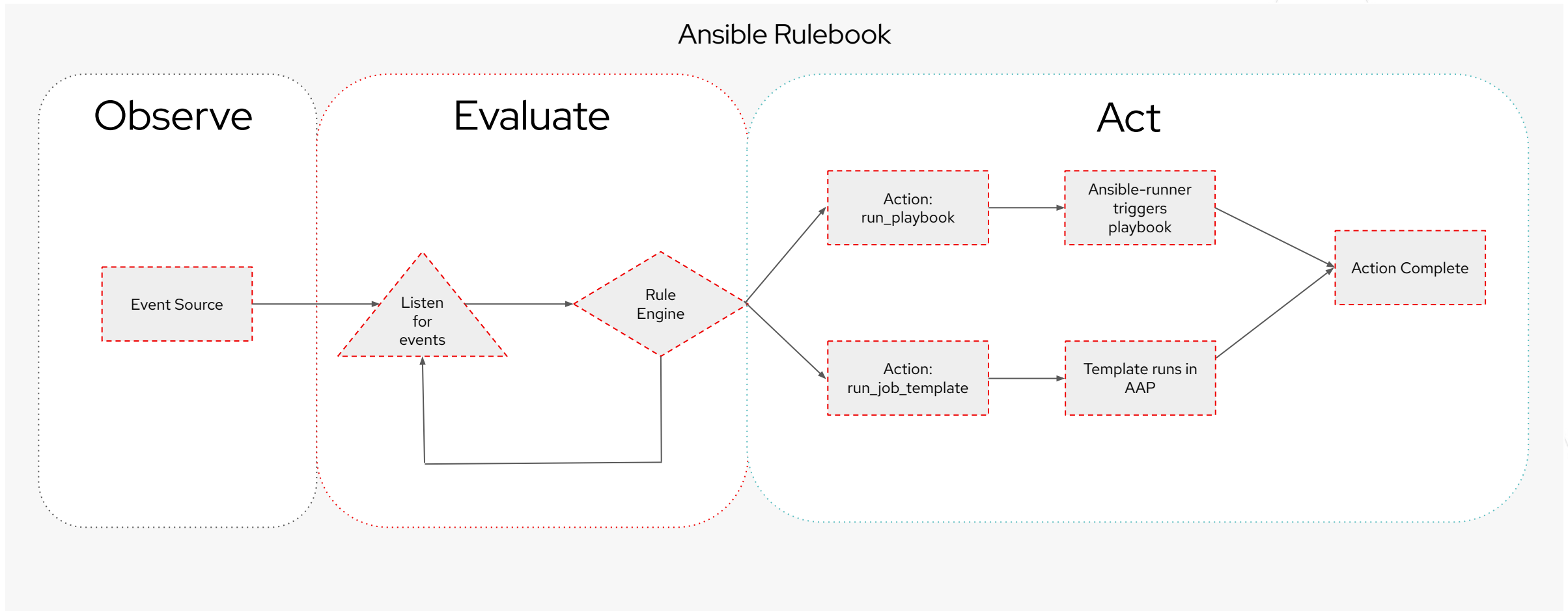
▶ Rulebooks comprises of Rulesets

- ▶ Rulebooks can contain multiple rulesets
- ▶ Rulebooks are analogous to Playbooks

```
- name: My unique ruleset - 1
hosts: switches
sources:
  - name: listen for events
    ansible.eda.kafka:
      host: 192.168.1.105
      port: 9092
rules:
  - name: An remediation rule with 1 action
    condition: event.outage == true
    action:
      run_playbook:
        name: remediate_outage.yml

- name: My unique ruleset - 2
hosts: servers
sources:
  - name: listen for events
    ansible.eda.alertmanager:
      host: 0.0.0.0
      port: 8000
rules:
  - name: A remediation rule for ruleset - 2
    condition: event.failure == true
    actions:
      - run_playbook:
          name: report_failure.yml
```

Event-Driven Ansible Workflow



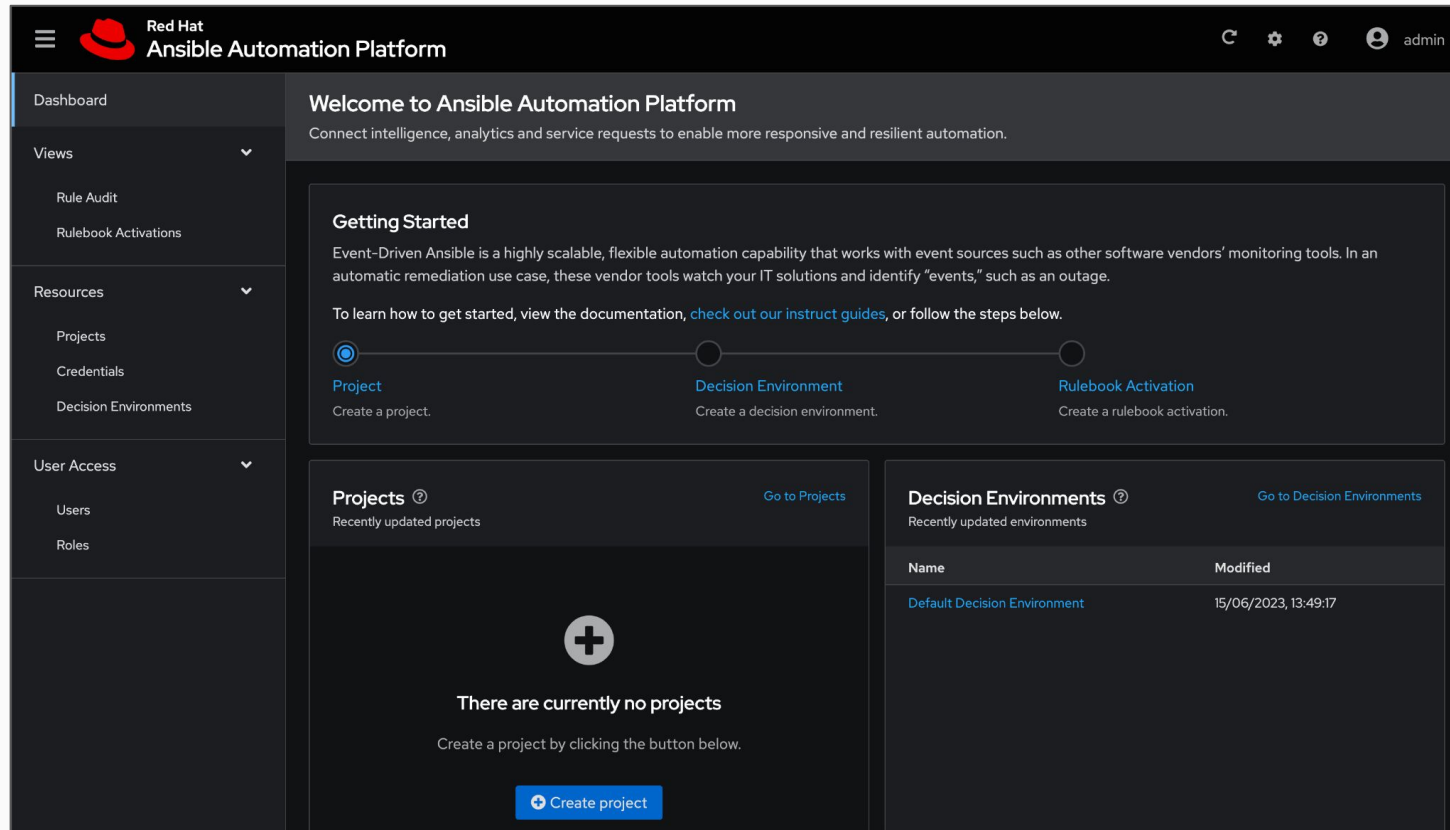
Getting Started with EDA Controller

- Projects
- Decision Environments
- Rulebook Activation



EDA Controller

Event-Driven Ansible Management



Red Hat
Ansible Automation Platform

admin

Dashboard

Views

- Rule Audit
- Rulebook Activations

Resources

- Projects
- Credentials
- Decision Environments

User Access

- Users
- Roles

Welcome to Ansible Automation Platform

Connect intelligence, analytics and service requests to enable more responsive and resilient automation.

Getting Started

Event-Driven Ansible is a highly scalable, flexible automation capability that works with event sources such as other software vendors' monitoring tools. In an automatic remediation use case, these vendor tools watch your IT solutions and identify "events," such as an outage.

To learn how to get started, view the documentation, [check out our instruct guides](#), or follow the steps below.

- Project**
Create a project.
- Decision Environment**
Create a decision environment.
- Rulebook Activation**
Create a rulebook activation.

Projects

Recently updated projects

[Go to Projects](#)

There are currently no projects

Create a project by clicking the button below.

[+ Create project](#)

Decision Environments

Recently updated environments

[Go to Decision Environments](#)

Name	Modified
Default Decision Environment	15/06/2023, 13:49:17

- ▶ **EDA Controller**
 - ▶ Manage EDA projects, Decision Environments and rulebook activation
 - ▶ Audit running rulebooks
 - ▶ View event history
 - ▶ Securely connected to **Automation Controller** via **Token**

Getting Started

Event-Driven Ansible Management

Getting Started

Event-Driven Ansible is a highly scalable, flexible automation capability that works with event sources such as other software vendors' monitoring tools. In an automatic remediation use case, these vendor tools watch your IT solutions and identify "events," such as an outage.

To learn how to get started, view the documentation, [check out our instruct guides](#), or follow the steps below.



Project

Create a project.



Decision Environment

Create a decision environment.



Rulebook Activation

Create a rulebook activation.

- ▶ **Project**
 - ▶ Grab your Rulebooks and synchronize from source control
- ▶ **Decision Environment**
 - ▶ Select the Container environment that contains ansible-rulebook and any additional Ansible content that is needed.
- ▶ **Rulebook Activation**
 - ▶ Select your Ansible Rulebook from the project and configure how it runs.

Projects

Single source of truth for your Rulebooks

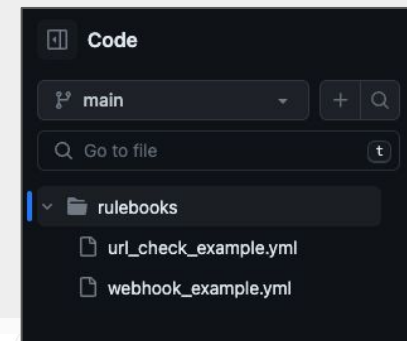
The screenshot shows the 'Create Project' form in the Red Hat Ansible Automation Platform. The form has a dark theme and includes the following fields:

- Name:** CloudVision Monitoring
- Description:** Enter description
- SCM type:** Git
- SCM URL:** https://github.com/nmartins0611/cloud_vision.git
- Credential:** Github

At the bottom of the form, there are two buttons: 'Create project' and 'Cancel'.

► Projects

- Similar to Automation controller, we create projects on EDA Controller and synchronize from a source of truth.
- Projects will contain all the resources you need such as Ansible Rulebooks.
- **Note:**
 - Rulebooks need to be kept in a rulebook subfolder in your project repository.



Decision Environments

Container based execution of Event-Driven Ansible

Decision Environments Create Decision Environment

Create Decision Environment

Name *

Description

Image *

Credential

Decision Environments

Decision environments contain an ansible-rulebook image that dictates where the rulebooks will run.

☐ starts with

Default Decision Enviro...

Decision Environment

Image

registry.redhat.io/ansible-automation-platfor...

▶ Decision Environments

- ▶ Provides a container environment similar to an Execution Environment.
- ▶ Contains an ansible-rulebook to run rulebooks
- ▶ Contains any additional Ansible content needed

Configure Rulebook Activation

Event-Driven Ansible Management

Rulebook Activations Create Rulebook Activation

Create Rulebook Activation

Name *	Description	Project ?
URL_Check	Enter description	eda_samples
Rulebook *	Decision environment *	Restart policy ?
Select project rulebook	Default Decision Environment	Always
Variables ?		
1		
Rulebook activation enabled? ?		
☒ Enabled		

Create rulebook activation Cancel

▶ Rulebook Configuration

- ▶ Rulebooks from our Projects can be configured via Rulebook Activation. Restart Policies, Decision Environments, Variables and enablement are configured here.
- ▶ Additional Variables can be supplied to the rulebooks
- ▶ Restart Policy allows us to restart rulebooks if needed.
 - Always
 - Never
 - On Failure

Rulebook Activation

Activated Automation

The screenshot shows the 'Rulebook Activations' page for a 'Dev_test' environment. It displays a table with columns for Name, Decision environment, Rulebook, Restart policy, Project, Activation status, Project git hash, Restarted count, and Created. Below the table, two callout boxes illustrate the activation process:

- Rulebook Activated**: A box with a red border and a downward-pointing arrow, indicating the transition from activation to waiting for events.
- Waiting for Events**: A box with a red border, indicating the state where the rulebook is listening for events.

The table data is as follows:

Name	Decision environment	Rulebook	Restart policy	Project	Activation status	Project git hash	Restarted count	Created
Dev_test	ansible-rulebook	hello_controller.yml	Always	eda-project2	Running	a6bd882ded2b413b9346edaa9e6faf7b30ebfb13	0	22/06/2023, 19:39:04

▶ Rulebook Activation

- ▶ Once Rulebooks are activated and running they are listening for events.
- ▶ Event data and Event History can be viewed from the Rulebook Activation window.
- ▶ Further inspection of the Rulebook events and actions can be done via **Rule Audit**

Rulebook Output

Viewing ansible-rulebook output and history

```
Rulebook Activations  Dev_test  History  25 - Dev_test

25 - Dev_test

Details

Name                Activation status    Start date
25 - Dev_test        running              22/06/2023, 19:39:04

Output

2023-06-22 17:39:08,718 - ansible_rulebook.app - INFO - Starting worker mode
2023-06-22 17:39:08,718 - ansible_rulebook.websocket - INFO - websocket ws://eda-daphne:8001/api/eda/ws/ansible-rulebook connecting
2023-06-22 17:39:08,752 - ansible_rulebook.websocket - INFO - websocket ws://eda-daphne:8001/api/eda/ws/ansible-rulebook connected
2023-06-22 17:39:08,818 - ansible_rulebook.job_template_runner - INFO - Attempting to connect to Controller
https://controller.apps.aap-dt.ocp4.testing.ansible.com
2023-06-22 17:39:09,013 - ansible_rulebook.app - INFO - AAP Version 4.4.0
2023-06-22 17:39:09,013 - ansible_rulebook.app - INFO - Starting sources
2023-06-22 17:39:09,013 - ansible_rulebook.app - INFO - Starting rules
2023-06-22 17:39:09,013 - ansible_rulebook.engine - INFO - run_ruleset
2023-06-22 17:39:09,015 - drools.ruleset - INFO - Using jar: /opt/app-root/lib/python3.9/site-packages/drools/jars/drools-ansible-
rulebook-integration-runtime-1.0.2-SNAPSHOT.jar
2023-06-22 17:39:10,053 [main] INFO org.drools.ansible.rulebook.integration.api.rulesengine.AbstractRulesEvaluator - Start automatic
pseudo clock with a tick every 100 milliseconds
2023-06-22 17:39:10,060 - ansible_rulebook.engine - INFO - ruleset define: {"name": "Hello Controller", "hosts": ["all"], "sources":
[{"EventSource": {"name": "ansible.eda.range", "source_name": "ansible.eda.range", "source_args": {"limit": 5}, "source_filters":
[]}], "rules": [{"Rule": {"name": "Run job template", "condition": {"AllCondition": [{"EqualsExpression": {"lhs": {"Event": "i"},
"rhs": {"Integer": 1}}]}, "actions": [{"Action": {"action": "run_job_template", "action_args": {"name": "hello", "organization":
"Default", "job_args": {"extra_vars": {"hello": "Fred"}}, "retries": 1, "delay": 10}}], "enabled": true}}]}]}
2023-06-22 17:39:10,078 - ansible_rulebook.engine - INFO - load source
2023-06-22 17:39:10,847 - ansible_rulebook.engine - INFO - load source filters
2023-06-22 17:39:10,847 - ansible_rulebook.engine - INFO - loading eda.builtins.insert_meta_info
2023-06-22 17:39:11,568 - ansible_rulebook.engine - INFO - Calling main in ansible.eda.range
2023-06-22 17:39:11,569 - ansible_rulebook.websocket - INFO - websocket ws://eda-daphne:8001/api/eda/ws/ansible-rulebook connecting
2023-06-22 17:39:11,572 - ansible_rulebook.engine - INFO - Waiting for all ruleset tasks to end
2023-06-22 17:39:11,573 [drools-async-evaluator-thread] INFO org.drools.ansible.rulebook.integration.api.io.RuleExecutorChannel -
Async channel connected
2023-06-22 17:39:11,608 - ansible_rulebook.rule_set_runner - INFO - Waiting for actions on events from Hello Controller
2023-06-22 17:39:11,608 - ansible_rulebook.rule_set_runner - INFO - Waiting for events, ruleset: Hello Controller
2023-06-22 17:39:11,668 [main] INFO org.drools.ansible.rulebook.integration.api.rulesengine.RegisterOnlyAgendaFilter - Activation of
effective rule "Run job template" with facts: {m={meta={received_at=2023-06-22T17:39:11.570384Z, source={name=ansible.eda.range,
type=ansible.eda.range}, uuid=666eee67-c9ab-4218-a9e0-5e7abb2ec86b}, i=1}}
2023-06-22 17:39:11,678 - ansible_rulebook.rule_generator - INFO - calling Run job template
2023-06-22 17:39:11,681 - ansible_rulebook.rule_set_runner - INFO - call_action run_job_template
2023-06-22 17:39:11,682 - ansible_rulebook.rule_set_runner - INFO - substitute_variables [{"name": "hello", "organization":
'Default', 'job_args': {'extra_vars': {'hello': 'Fred'}}, 'retries': 1, 'delay': 10}] [{"event": {'meta': {'received_at': '2023-06-
22T17:39:11.570384Z', 'source': {'name': 'ansible.eda.range', 'type': 'ansible.eda.range'}, 'uuid': '666eee67-c9ab-4218-a9e0-
5e7abb2ec86b'}, 'i': 1}}]}
2023-06-22 17:39:11,682 - ansible_rulebook.rule_set_runner - INFO - action args: {'name': 'hello', 'organization': 'Default',
'job_args': {'extra_vars': {'hello': 'Fred'}}, 'retries': 1, 'delay': 10}
2023-06-22 17:39:11,682 - ansible_rulebook.builtin - INFO - running job template: hello, organization: Default
2023-06-22 17:39:11,682 - ansible_rulebook.builtin - INFO - ruleset: Hello Controller, rule Run job template
2023-06-22 17:39:11,688 - ansible_rulebook.engine - INFO - Broadcast shutdown to all source plugins
2023-06-22 17:39:11,689 - ansible_rulebook.engine - INFO - Broadcast to queues: [<Queue at 0x7f4069f386d0 maxsize=1 _getters[1]
```

▶ Rulebook Activation

- ▶ Once Rulebooks are activated It is possible to view the history output of that rulebook.
- ▶ This provides you with the ability to view the verbose output of ansible-rulebook for your specific rulebook.

Rule Audit

An Overview of events and actions

Rule Audit

Rule audit allows auditing of rules which have been triggered by incoming events.

Name	Status	Rulebook activation	Last fired date
Run job template	✓ Successful		22/06/2023, 19:39:11

Rule Audit > Run job template

Run job template

Details Events Actions

Rule name	Status	Rulebook activation ⓘ
Run job template	✓ Successful	Dev_test
Created	Fired date	
22/06/2023, 19:39:32	22/06/2023, 19:39:11	

▶ Audit Rulebook Activation

- ▶ Rule Audit allows you to view data around the events and actions that have taken place.
- ▶ Events will list the events that have matched within the rulebook.
- ▶ Actions provide a list of actions which have taken place

Event Audit

Event information and auditing

Rule Audit · Run job template

Run job template

Details **Events** Actions

▼ Name starts with →

Name	Source type	Timestamp
ansible.eda.range	ansible.eda.range	22/06/2023, 19:39:11

▶ Event list

- ▶ The Event list in Rule Audit provides a list of events that have been matched with your Rulebook conditions.
- ▶ The events are listed and provide the event source type where they were observed.
- ▶ These events can be selected to get the event information.

Source type Timestamp

ansible.eda.range 22/06/2023, 19:39:11

Event details

×

Name	Source type	Timestamp
ansible.eda.range	ansible.eda.range	22/06/2023, 19:39:11

Event log

```
{ "i": 1 }
```

Close

Action Audit

Action information and data

Rule Audit > Run job template

Run job template

Details Events **Actions**

🔍 Name starts with →

Name	Status	Last fired date
run_job_template		

Jobs > 188 - hello

Details

⏪ Back to Jobs Details Output

Job ID	188	Status	Successful	Started	22/06/2023, 19:39:12
Finished	22/06/2023, 19:39:23	Job Template	hello	Job Type	Playbook Run
Launched By	admin	Inventory	managed nodes	Project	Demo Project
Revision	347e44fea036c94d5f60e544de006453ee5c71ad	Playbook	hello_world.yml	Verbosity	0 (Normal)
Execution Environment	Default execution environment	Controller Node	controller-task-5c89b6bd-7fd6b	Container Group	default
Job Slice	0/1	Forks	0	Timeout	No timeout specified
Credentials	SSH: managed nodes				
Created	22/06/2023, 19:39:12 by admin	Last Modified	22/06/2023, 19:39:12		
Variables	YAML JSON				
1 {}					
Artifacts	YAML JSON				
1 {}					
<button>Relaunch</button> <button>Delete</button>					

▶ Actions

- ▶ Rule Audit provides the ability to access the Action history.
- ▶ If the action is to run_job_template in Automation Controller, the action will link back to the Job Status on Automation Controller.

Lab Time

Section 4 - Automate decision making with Event-Driven Ansible

 In this lab you will learn the basics of ansible-rulebook CLI and use Event-Driven Automation (EDA) Controller with real use cases.

 Approximate time: 40 mins



Red Hat Ansible Automation Platform

What you learned

- ▶ Writing playbooks using Ansible Lightspeed
- ▶ Introduction to Network Automation using AAP
- ▶ Using Automation Controller to operationalize network automation for your enterprise
- ▶ Automate decision making with Event-Driven Ansible
- ▶ Putting it all together! - Implementing real-world scenarios with what we learnt

NAF PRESENTS

AUTOCON 2

WORKSHOP SURVEY



NETWORKAUTOMATION.FORUM



Red Hat Ansible Automation Platform

Where to go next

Learn more

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

Get started

- ▶ [Evals](#)
- ▶ [Cloud.redhat.com](#)
- ▶ [Network Automation Guide](#)

Get serious

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

Chat with us

- **Slack**

<https://ansiblenetwork.slack.com>

Join by clicking here <http://bit.ly/ansibleslack>

- **Matrix**

Ansible Network Room

<https://matrix.to/#/#network:ansible.com>

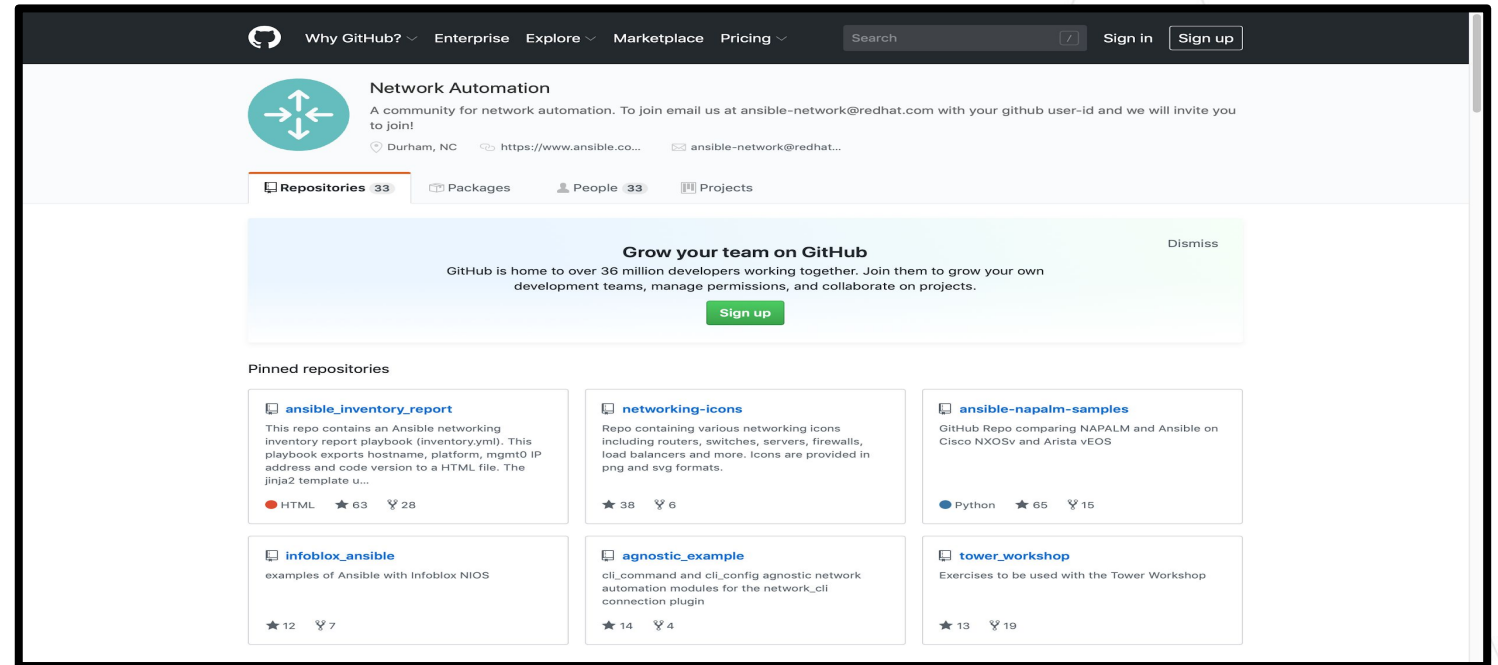
- **Forum**

Discourse Forum

<https://forum.ansible.com>

Bookmark the Github organization

- Examples, samples and demos
- Run network topologies right on your laptop



Ansible automation platform trial

<https://www.redhat.com/en/technologies/management/ansible/trial>

Thank you

 linkedin.com/company/red-hat

 youtube.com/AnsibleAutomation

 facebook.com/ansibleautomation

 twitter.com/ansible

 github.com/ansible