Red Hat
Ansible Automation
Platform

Get started

with Red Hat Ansible Automation Platform

Contents

Introduction

Chapter 1:
Learn the basics

- 1.1 Platform installation
- 1.2 Key platform components
- 1.3 Ansible Playbooks
- 1.4 Ansible Roles
- 1.5 Inventories
- 1.6 Ansible Content Collections

Chapter 2:
Get started with common use cases

- 2.1 Provision cloud resources
- 2.2 Deploy virtual machines at scale
- 2.3 Manage services
- 2.4 Perform compliance audits
- 2.5 Manage system configurations
- 2.6 Deploy applications
- 2.7 Configure network devices
- 2.8 Upgrade operating systems

Chapter 3:
Explore advanced automation use cases

- 3.1 Event-driven automation
- 3.2 Automation at the edge
- 3.3 End-to-end cloud workflows
- 3.4 Organization-wide automation

Resources and information

Introduction

Automation is a critical technology for managing today's increasingly complex IT environments.

Through IT automation, you can save staff time, increase product quality, encourage cross-team collaboration, improve employee satisfaction, and reduce costs throughout your organization.

A foundation for building and operating automation services at scale, **Red Hat® Ansible® Automation Platform** delivers all the tools and features you need to implement organization-wide automation. The platform provides a flexible, stable, and security-focused foundation for deploying end-to-end automation workflows—from IT processes, to hybrid cloud, to edge locations. With Ansible Automation Platform, you can create, manage, and scale automation across your organization.

This e-book provides guidance for getting started with Ansible Automation Platform. We'll review key automation concepts, product features and capabilities, use cases for gaining value quickly, and strategies for implementing organization-wide automation.

The value of automation by the numbers

Red Hat Ansible Automation Platform unites people and processes on a flexible foundation to deliver value across your organization:

36%
higher development team productivity[1]

68%
faster deployment of new compute resources[1]

23%
faster time to market for new products and services[1]

61%
reduction in unplanned downtime[1]

668%
3-year return on investment (ROI)[1]

US$8.54 million
higher revenue per organization per year[1]

*1 IDC White Paper, sponsored by Red Hat. "**The business value of Red Hat Ansible Automation Platform**." Document #US51839824. March 2024.*

Chapter 1:

Learn the basics

Before getting started with Red Hat Ansible Automation Platform, it's important to understand what the platform includes, how its components work together, and a few key automation concepts like playbooks, roles, and inventories.

In this chapter:

1.1 Platform installation

1.2 Key platform components

1.3 Ansible Playbooks

1.4 Ansible Roles

1.5 Inventories

1.6 Ansible Content Collections

Install Red Hat Ansible Automation Platform

Installing and setting up Red Hat Ansible Automation Platform is simple and fast. The installation program gives you the flexibility to install Ansible Automation Platform—along with the optional Event-Driven Ansible controller—using several supported installation scenarios. See these installation guides for specific details on each scenario:

- Traditional virtual machine
- Red Hat OpenShift®
- Cloud environments
- Containerized environments

Regardless of which installation scenario you choose, installing Ansible Automation Platform involves these steps:

1. **Edit the Ansible Automation Platform installer inventory file** to specify your installation scenario and describe host deployments to Ansible.
2. **Run the Ansible Automation Platform installer setup script** and install your private automation hub using parameters defined in the installer inventory file.
3. **Verify successful installation** by logging in to the automation controller, automation hub, and optional Event-Driven Ansible controller.

Understand key components of the platform

Red Hat Ansible Automation Platform includes several key components that you'll interact with while creating, managing, and scaling automation across your organization.

Ansible automation hub

Available with your Ansible Automation Platform subscription, **Ansible automation hub** is a central repository to discover, download, and manage **Ansible Content Collections** and **validated content**. **Private automation hub** is an on-site repository that allows enterprises with disconnected environments to manage, share, and curate their own automation content and control access to certified and validated Red Hat and partner-created content.

Learn more about Ansible Content Collections on **page 10**.

Automation controller

Automation controller is the control plane for Ansible Automation Platform. It lets you deploy, initiate, delegate, and audit automation across your organization. Automation controller includes a user interface, role-based access control (RBAC), and workflows to help you scale your automation with efficiency and flexibility. Manage inventories, launch and schedule workflows, track changes, and generate reports, all from a centralized user interface and **RESTful application programming interface (API)**.

Read the **getting started guide**, **administration guide**, and **user guide** to learn about configuring and using automation controller.

Automation mesh

Automation mesh is an overlay network that simplifies the distribution of automation tasks across a collection of nodes using existing connectivity. Execution nodes run the automation execution environments that perform the tasks defined in Ansible playbooks. Automation mesh creates peer-to-peer connections between these execution nodes, increasing the resiliency of your automation workloads to network latency and connection disruptions. This also permits more flexible architectures and provides rapid, independent scaling of control and execution capacity.

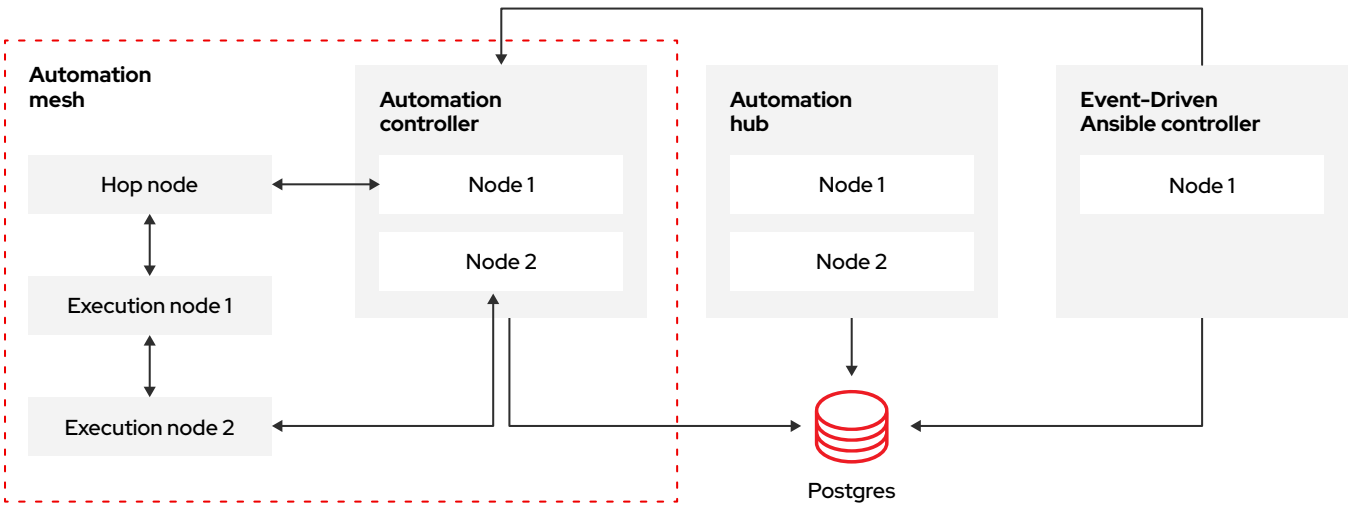
Event-Driven Ansible

Event-Driven Ansible provides event-handling capabilities for delivering an automated response to changing IT conditions in any IT domain. It can process events containing discrete intelligence about conditions in your IT environment, determine the appropriate response to the event, then execute automated actions to address or remediate the event. IT service management tasks like ticket enhancement, remediation, and user management are ideal starting points, but Event-Driven Ansible is flexible enough to automate a diverse set of tasks across your IT environment.

Learn more about Event-Drive Ansible on page 23.

Red Hat Ansible Automation Platform 2.4 architecture

Each of these components works together within the overall Ansible Automation Platform architecture.



Get to know Ansible Playbooks

Playbooks provide instructions for configuring, deploying, and orchestrating IT assets through Red Hat Ansible Automation Platform. They consist of sets of commands called plays that define automation across an inventory of hosts. Each play includes one or more tasks that target one, many, or all hosts in the inventory. Each task calls a module that performs a specific function like collecting information, managing configurations, or validating connectivity. Playbooks can be shared and reused by multiple teams to create repeatable automation.

This example shows the common parts of an Ansible Playbook.

```
1  ---
2  - name: Add VLANs
3    hosts: arista
4    gather_facts: false
5
6    vars:
7      vlans:
8        - name: desktops
9          vlan_id: 20
10       - name: servers
11         vlan_id: 30
12       - name: DMZ
13         vlan_id: 50
14
15   tasks:
16     - name: Add VLAN configuration
17       arista.eos.eos_vlans:
18         state: merged
19         config: "{{ vlans }}"
```

Indicates the start of a playbook

Calls a device or group of devices named `arista`

Optional parameter for retrieving facts about the specified devices

Variable definition

In this playbook, we define variable values directly.

Via automation controller, you can also create a survey to prompt users for variable values when they run your playbook.

Tasks

Tasks and Ansible modules have a 1:1 correlation. This section calls modules to configure virtual local area networks (VLANs) for each of the 3 variables defined in the `vars` section.

What is YAML?

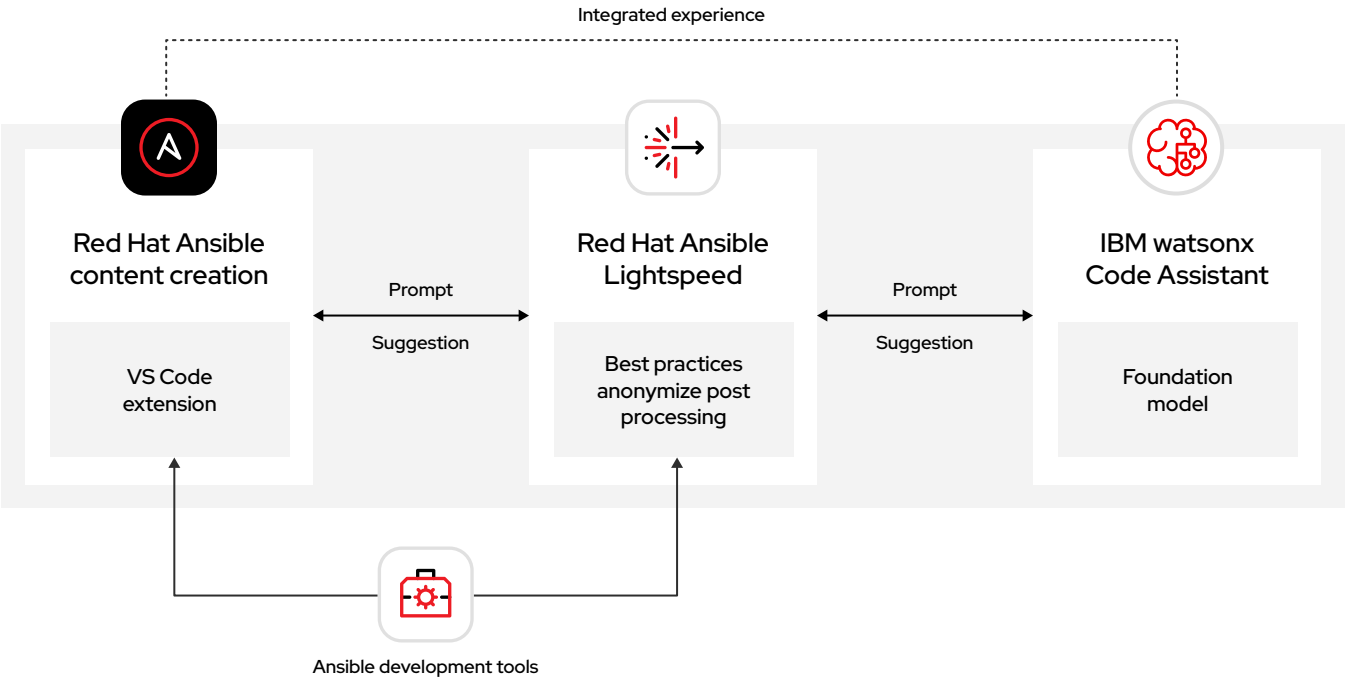
YAML is a human-friendly data serialization language that is designed to be easy to read, write, and understand. Unlike many programming languages, there are no usual format symbols like braces, square brackets, closing tags, or quotation marks. YAML files are simpler to read since they use indentation with space characters to determine the structure and indicate nesting. Because of its flexibility and accessibility, Ansible Playbooks, Rulebooks, and inventories use YAML to fully define all automation content.

Simplify playbook creation

While writing playbooks in YAML is straightforward, there are 2 tools available to simplify and streamline the creation of playbooks.

The **Ansible VS Code Extension** adds language support for Ansible to Visual Studio Code and OpenVSX compatible editors running on operating systems that support `ansible` and `ansible-lint`. Ansible keywords, module names, and module options, as well as standard YAML elements, are recognized and highlighted. While you type, the syntax of your Ansible scripts is verified and feedback is provided instantaneously. On opening and saving a document, `ansible-lint` executes in the background and reports any potential issues. Smart autocompletion detects whether the cursor is on a play, block, or task and provides relevant suggestions.

Red Hat Ansible Lightspeed with IBM watsonx Code Assistant is a generative artificial intelligence (AI) service that helps developers create Ansible content more efficiently. It accepts prompts entered by a user and then interacts with IBM watsonx foundation models to generate code recommendations based on Ansible best practices. Because it is context aware, Ansible Lightspeed with watsonx Code Assistant can also provide suggestions based on other words found in the playbook, including play names and other modules.



Watch this video to see how to enable Ansible Lightspeed with watsonx Code Assistant with the Ansible VS Code Extension.

Create reusable automation content with roles

Sharing and reusing automation content allows teams to work more efficiently and avoid duplicate efforts. Red Hat Ansible Automation Platform provides technologies to help you create reusable content and quickly distribute it across your organization.

Ansible Roles let you build reusable components by grouping and encapsulating related automation artifacts, like configuration files, templates, tasks, and handlers. Because roles isolate these components, it's easier to reuse them and share them with other people. You can also make your roles configurable by exposing variables that users can set when calling the role, allowing them to configure their system according to specific requirements.

Read this article to learn more about creating reusable playbooks with roles.

Understand inventories

An inventory is a collection of hosts that may be acted on using Ansible commands and playbooks. Inventory files organize hosts into groups and can serve as a source of trust for your IT assets. These files can be formatted as simple INI or YAML. Many organizations choose to write their inventories in YAML for consistency with their playbooks. Using an inventory file, a single playbook can maintain hundreds of devices with a single command.

This section explains how to build an inventory file. You can also find a sample inventory report on GitHub.

Create a basic INI-formatted inventory

First, group your inventory logically. Best practices are to group servers and devices by their ***what*** (application, stack, or microservice), ***where*** (datacenter or region), and ***when*** (development stage). Here are some examples:

- **What**: `db, web, leaf, spine`
- **Where**: `east, west, floor_19, building_A`
- **When**: `dev, test, staging, prod`

This example code, in INI format, illustrates a basic group structure for a very small datacenter. You can group groups using the syntax `[metagroupname:children]` and listing groups as members of the metagroup.

Here, the group `network` includes all leafs and all spines. The group `datacenter` includes all network devices plus all webservers.

Read the Build your inventory section of the Ansible documentation to learn more.

```
[leafs]
leaf01
leaf02

[spines]
spine01
spine02

[network:children]
leafs
spines

[webservers]
webserver01
webserver02

[datacenter:children]
network
webservers
```

Anatomy of a YAML-formatted inventory

```
1  ---
2  all:
3    vars:
4      ansible_user: admin
5      ansible_password: password123
6      ansible_become_pass: password123
7      ansible_become: True
8      ansible_become_method: enable
9      ansible_network_cli_ssh_type: libssh
10   children:
11     routers:
12       children:
13         arista:
14         cisco:
15         juniper:
16     arista:
17       hosts:
18         rtr2:
19           ansible_host: 172.16.100.2
20         rtr4:
21           ansible_host: 172.16.100.4
22       vars:
23         ansible_network_os: arista.eos.eos
24         ansible_connection: ansible.netcommon.network_cli
25     cisco:
26       hosts:
27         rtr1:
28           ansible_host: 172.16.100.1
29       vars:
30         ansible_network_os: cisco.ios.ios
31         ansible_connection: ansible.netcommon.network_cli
32     juniper:
33       hosts:
34         rtr3:
35           ansible_host: 172.16.100.3
36       vars:
37         ansible_network_os: junipernetworks.junos.junos
38         ansible_connection: ansible.netcommon.netconf
```

Indicates the start of a playbook

Defines variables that apply to all hosts within the inventory, regardless of group

Group hierarchy

Lines 10-15 identify the host groups within this inventory. In this case, the group `routers` contains 3 subgroups: `arista`, `cisco`, and `juniper`.

Group definition

The `hosts` command defines which hosts belong to each group. In this example, the group `arista` contains 2 hosts that are identified by their IP addresses.

Group variables

Each group can have its own set of variables. This inventory defines the operating system and connection type for each group. All of these group variables point to items contained in content collections.

Reduce frustrations through version control

Keeping your playbooks, roles, inventories, and variables files in **version control systems** with meaningful commit messages and comments can simplify how you create, review, and revise automation content. By tracking modifications made to files, version control tools let you trace and audit the content development over time and quickly revert to previous versions if an issue occurs. And because version control tools support branching and merging, multiple team members can work on the same automation content without impacting each other.

Read **A beginner's guide to Git version control** to learn more about this popular version control system.

Get started in less time with Ansible Content Collections

Ansible Content Collections are a standardized distribution format for automation content. Collections can contain playbooks, rulebooks, roles, modules, other plugins, inventories, and documentation.

You can take advantage of 2 types of ready-to-use collections:

- Red Hat Ansible Certified Content includes collections built, supported, and maintained by Red Hat and our technology partners. Available directly from Ansible automation hub, these collections focus on integrations between Red Hat and partner platforms so you can automate across different IT domains and technologies.
- Ansible validated content offers expert guidance for using Ansible Automation Platform to perform operations or tasks in Red Hat and partner platforms. Typically delivered as roles and documentation, validated content contains customizable, opinionated use cases and may be based on Red Hat Ansible Certified Content. Validated content can provide additional operational capabilities related to the entire Ansible Automation Platform, including configuration of the automation controller and automation hub and creation of execution environments.

You can also develop internal collections and upload them to private automation hub for distribution within your organization. Once a collection is published, internal users can download it for use.

Learn more about **Ansible Automation Platform Certified and Validated Content** on the Red Hat Customer Portal.

Chapter 2:

Get started with common use cases

Adopting automation is a journey, not an all-or-nothing proposition. You can get started with a single use case and expand at a pace that works for your organization.

In this chapter:

2.1 Provision cloud resources

2.2 Deploy virtual machines at scale

2.3 Manage services

2.4 Perform compliance audits

2.5 Manage system configurations

2.6 Deploy applications

2.7 Configure network devices

2.8 Upgrade operating systems

This chapter gives an overview of common automation use cases to help you explore the benefits of automation within your organization. Each of these use cases varies in the amount of time and effort required to automate. We recommend creating a flowchart of your current processes to help you understand the complexities of your automation project and how to best apply these use cases. Successful automation adoption journeys often follow a progression: teams start small, show value, and expand the range and complexity of their efforts in an iterative manner.

By measuring and analyzing the time and cost of performing a task manually versus automating the same task, you can determine the total automation savings across your organization. The **automation calculator** provides graphs, metrics, and calculations to help you determine the total savings on your automation investment.

You can use **automation analytics and Red Hat Insights for Red Hat Ansible Automation Platform** to plan, measure, manage, and expand your automation based on actionable data. Included with your subscription, these 2 tools provide rich reporting and observability metrics to help you track your automation efforts. The automation analytics tool lets you measure the business impact of Ansible Automation Platform, including your return on investment (ROI) and how your automation is performing. Red Hat Insights lets you monitor your automation infrastructure, including system configuration and health status. You can find and resolve problems like infrastructure performance issues, system availability, and security vulnerabilities. You can also use notifications from Red Hat Insights as a data source to initiate automatic remediation through Event-Driven Ansible.

Use case: Provision cloud resources

Provisioning cloud resources is a time-consuming, error-prone process for many organizations. You can use Red Hat Ansible Automation Platform to simplify virtual machine provisioning in public cloud environments. Create playbooks using Ansible Certified Content to allocate storage, set up networks and subnetworks, and provision virtual machine instances. Add variables for configuration options like instance types, zones, and security groups to make your playbook reusable so you can deploy virtual machines anywhere.

Here are 3 example tasks for provisioning cloud resources in public cloud environments.

Example: Create a Google Cloud instance with disks and network interfaces

```
---
- name: create a instance
  google.cloud.gcp_compute_instance:
    name: test_object
    machine_type: n1-standard-1
    disks:
    - auto_delete: 'true'
      boot: 'true'
      source: "{{ disk }}"
    - auto_delete: 'true'
      interface: NVME
      type: SCRATCH
      initialize_params:
        disk_type: local-ssd
      labels:
      environment: production
    network_interfaces:
    - network: "{{ network }}"
      access_configs:
      - name: External NAT
        nat_ip: "{{ address }}"
        type: ONE_TO_ONE_NAT
    zone: us-central1-a
    project: test_project
    auth_kind: serviceaccount
    state: present
```

Learn more about **Red Hat Certified Content for Google Cloud** on the Red Hat Ecosystem Catalog.

Example: Create an AWS EC2 instance with a public IP address

```
---
- name: Provision AWS EC2 instance
  amazon.aws.ec2_instance:
    name: “public-compute-instance”
    key_name: “prod-ssh-key”
    vpc_subnet_id: subnet-5ca1ab1e
    instance_type: c5.large
    security_group: default
    network:
      assign_public_ip: true
    image_id: ami-123456
    tags:
      Environment: Testing
```

Learn more about **Red Hat Certified Content for AWS** on the Red Hat Ecosystem Catalog.

Example: Create a Microsoft Azure virtual machine with managed disk

```
---
- name: Provision Microsoft Azure instance
  azure_rm_virtualmachine:
    name: vm-managed-disk
    vm_size: Standard_D4
    resource_group: myResourceGroup
    admin_username: "{{ username }}"
    availability_set: avs-managed-disk
    managed_disk_type: Standard_LRS
    image:
      offer: RHEL
      publisher: RedHat
      sku: '8-lvm-gen2'
      version: latest
```

Learn more about **Red Hat Certified Content for Microsoft Azure** on the Red Hat Ecosystem Catalog.

Use case: Deploy virtual machines at scale

Deploying virtual machines using manual processes can result in misconfigurations or unexpected issues that lead to downtime and service disruptions. With Red Hat Ansible Automation Platform, you can automatically create and maintain standardized virtual machine templates that let you provision virtual machines consistently across your VMware vSphere environment. Create a template based on a static virtual machine image using the `vmware.vmware_rest.vcenter_vmtemplate_libraryitems` module from the VMware collection. Then, use the same module to consistently deploy new virtual machines based on the template.

Example: Use a template to deploy virtual machines with VMware

```
---
- name: Deploy a new VM based on the template
  vmware.vmware_rest.vcenter_vmtemplate_libraryitems:
    name: vm-from-template
    library: "{{ nfs_lib.id }}"
    template_library_item: "{{ my_template_item.id }}"
    placement:
      cluster: "{{ lookup('vmware.vmware_rest.cluster_moid',
                   '/my_dc/host/my_cluster') }}"
      folder: "{{ lookup('vmware.vmware_rest.folder_moid',
                  '/my_dc/vm') }}"
      resource_pool: "{{ lookup('vmware.vmware_rest.resource_pool_moid',
                          '/my_dc/host/my_cluster/Resources') }}"
    state: deploy
```

Read **Managing a VMware template life cycle with Ansible** to learn more. Access the **VMware collection** on the Red Hat Ecosystem Catalog.

Use case: Manage services

Known problematic services that require frequent restarts can be challenging to manage. Red Hat Ansible Automation Platform can help you respond quickly to recurring issues with applications and services. Built-in modules—including `ansible.builtin.systemd` and `ansible.builtin.sysvinit`—let you control services on remote hosts via a selection of service managers. The `ansible.builtin.service` module acts as a proxy to service manager modules so you can manage diverse environments without creating a specific task for each service manager. As a result, you can create simple playbooks that automatically gather information on impacted systems and application layers, and restart services as soon as an issue is reported.

Example: Start services

```
1  ---
2  - name: Start service httpd, if not started
3    ansible.builtin.service:
4      name: httpd
5      state: started
```

Example: Stop services

```
1  ---
2  - name: Stop service httpd, if started
3    ansible.builtin.service:
4      name: httpd
5      state: stopped
```

Example: Restart services

```
1  ---
2  - name: Restart service httpd
3    ansible.builtin.service:
4      name: httpd
5      sleep: 60
6      state: restarted
```

Learn about **managing services with the `ansible.builtin.service` module** in the Ansible Automation Platform documentation.

Use case: Perform compliance audits

Most environments contain many different platforms and devices, making manual compliance audits difficult and time consuming. Red Hat Ansible Automation Platform simplifies and standardizes how you audit resources across your IT environment. Write playbooks using Ansible Certified Content to query, store, and report system configurations with less manual effort. And if a system configuration is not in the expected state, Ansible Automation Platform can automatically log a service ticket and optionally remediate the configuration.

Example: Gather network facts

```
---
- name: Use Cisco IOS facts module
  hosts: cisco
  gather_facts: false # this is not the cisco facts module

  tasks:
    - name: retrieve facts
      cisco.ios.ios_facts:

    - name: display version
      ansible.builtin.debug:
        msg: “{{ ansible_net_version }}”

    - name: display serial number
      ansible.builtin.debug:
        msg: “{{ ansible_net_serialnum }}”
```

Example: Retrieve network resource information

```
---
- name: Retrieve interface information
  hosts: cisco
  gather_facts: false # this is not the cisco facts module

  tasks:
    - name: use state gathered
      cisco.ios.ios_interfaces:
        state: gathered
      register: interfaces_info

    - name: print interface information
      ansible.builtin.debug:
        msg: “{{ interfaces_info }}”
```

Example: Back up network configurations

```
---
- hosts: cisco
  gather_facts: false

  tasks:
  - name: Back up config
    cisco.ios.ios_config:
      backup: yes
```

Use case: Manage system configurations

Keeping resources current with the latest security standards helps protect systems and reduce vulnerabilities. Red Hat Enterprise Linux® System Roles is a collection of Ansible Certified Content that provides a stable and consistent configuration interface to automate and manage multiple releases of Red Hat Enterprise Linux. Create and review playbooks using these roles to automatically update system configurations whenever security standards change.

Read **Introduction to Red Hat Enterprise Linux System Roles** in the Ansible Automation Platform documentation for more details. Access **system roles and documentation** from the Red Hat Hybrid Cloud Console (**console.redhat.com**).

Example: Update kernel settings

```
---
- name: Manage kernel settings
  hosts: all
  vars:
    kernel_settings_sysctl:
      - name: fs.epoll.max_user_watches
        value: 785592
      - name: fs.file-max
        value: 379724
      - name: kernel.threads-max
        state: absent

    kernel_settings_sysfs:
      - name: /sys/kernel/debug/x86/pti_enabled
        value: 0
      - name: /sys/kernel/debug/x86/retp_enabled
        value: 0
      - name: /sys/kernel/debug/x86/ibrs_enabled
        value: 0

    kernel_settings_systemd_cpu_affinity: “1,3,5,7”
    kernel_settings_transparent_hugepages: madvise
    kernel_settings_transparent_hugepages_defrag: defer
  roles:
    - linux-system-roles.kernel_settings
```

Use case: Deploy applications

Manual application deployment processes can be error-prone and result in increased security risks and decreased application performance. Red Hat Ansible Automation Platform includes built-in modules that let you write reusable playbooks for installing and configuring applications simply and consistently across your environment. Use certified modules to install web servers using YUM (Yellowdog Updater Modified) or DNF (Dandified YUM), set default home pages, start servers, and configure firewalls, all in a single, easy-to-read playbook.

Example: Deploy a web server

```
---
- name: Setup the web server
  hosts: "{{ hosts }}"
  become: true
  tasks:
    - name: httpd installed
      ansible.builtin.yum:
        name: httpd
        state: latest

    - name: custom index.html
      ansible.builtin.copy:
        dest: /var/www/html/index.html
        content: | Custom Web Page

    - name: httpd service enabled
      ansible.builtin.service:
        name: httpd
        enabled: true
        state: started

    - name: open firewall
      ansible.posix.firewalld:
        service: http
        state: enabled
        immediate: true
        permanent: true
```

Use case: Configure network devices

Manual approaches to network configuration and updates are too slow to effectively support modern application and data transfer requirements. Red Hat Ansible Certified Content helps you automate many common network tasks across your hybrid cloud. Write playbooks to configure router hostnames and domain name system (DNS) servers, and create and propagate virtual local area network (VLAN) configurations across your environment.

Example: Configure routers

```
---
- name: configure cisco routers
  hosts: routers
  connection: ansible.netcommon.network_cli
  gather_facts: false
  vars:
    dns: “8.8.8.8 8.8.4.4”

  tasks:
   - name: configure hostname
     cisco.ios.ios_config:
       lines: hostname {{ inventory_hostname }}

   - name: configure DNS
     cisco.ios.ios_config:
       lines: ip name-server {{dns}}
```

Read **Use Ansible network roles** in the Ansible Automation Platform documentation to learn more about automating your network.

Example: Add a VLAN

```
---
- name: add vlans
  hosts: arista
  gather_facts: false

  vars:
    vlans:
      - name: desktops
        vlan_id: 20
      - name: servers
        vlan_id: 30
      - name: DMZ
        vlan_id: 50

  tasks:
    - name: add VLAN configuration
      arista.eos.eos_vlans:
        state: merged
        config: "{{ vlans }}"
```

Use case: Upgrade operating systems

Infrastructure maintenance tasks like operating system upgrades often require large teams of IT staff members working outside normal business hours. With Red Hat Ansible Automation Platform, you can create complex automation workflows to perform Red Hat Enterprise Linux operating system upgrades across your environment. Write playbooks that download and install new operating system versions, conditionally reboot virtual machines, and automatically create reports describing the installed services and packages.

Example: Patch a Red Hat Enterprise Linux installation

```
---
- name: Upgrade all packages (yum)
 ansible.builtin.yum:
   name: '*'
   state: latest
   update_only: true
 when: ansible_pkg_mgr == "yum"
 register: patchingresult_yum


- name: Upgrade all packages (dnf)
 ansible.builtin.dnf:
   name: '*'
   state: latest
   update_only: true
 when: ansible_pkg_mgr == "dnf"
 register: patchingresult_dnf


- name: Check to see if we need a reboot
 ansible.builtin.command: needs-restarting -r
 register: result
 changed_when: result.rc == 1
 failed_when: result.rc > 1
 check_mode: false


- name: Reboot Server if Necessary
 ansible.builtin.reboot:
 when:
   - result.rc == 1
   - allow_reboot == true
```

Chapter 3:

Explore advanced automation use cases

Once you've gained experience automating common use cases and have started to deliver value through automation, you can expand your knowledge and processes to more advanced use cases throughout your organization.

This chapter discusses next steps for advancing your automation projects and goals with Red Hat Ansible Automation Platform.

In this chapter:

3.1 Event-driven automation

3.2 Automation at the edge

3.3 End-to-end cloud workflows

3.4 Organization-wide automation

Save time and effort with event-driven automation

Event-driven automation is the next step in the journey to end-to-end IT automation. It responds automatically when specific events or conditions occur in your IT environment. Event-driven automation receives information from observability and other tools, decides which actions to take, and initiates predefined actions based on conditional rules. By automating responses to events like network or system slowdowns, configuration drift, changing infrastructure conditions, and new service ticket entries, you gain the flexibility to create innovative, complex workflows across your environment.

Red Hat Ansible Automation Platform includes powerful event-driven automation capabilities. Event-Driven Ansible lets you respond in a predetermined way to observed events and conditions in your IT environment, without manual intervention. Simply define ***if-then*** rules, event sources, and automated actions in Ansible Rulebooks. The platform matches events received from third-party monitoring and observability tools to the applicable rulebook, determines the appropriate action, and then performs that action.

What is a rulebook?

Rulebooks are sets of conditional rules that Event-Driven Ansible uses to perform actions. They define 1 or more event sources, conditional rules, and corresponding actions. Rulebooks are written in YAML and use ***if-then*** rules to link specific events to automated actions.

5 ways to use Event-Driven Ansible

1. **Proactively remediate issues.** Automatically identify and remediate potential issues like performance degradation, configuration drift, or security vulnerabilities before they impact operations and users.
2. **Accelerate troubleshooting.** Simplify and speed troubleshooting activities by automating initial response actions based on factors like the type and severity of the incident, the frequency of similar incidents, and established corporate policies.
3. **Handle user administration requests.** Evaluate and respond automatically to user administration requests like recovering passwords or managing access based on information including the user's role and request type.
4. **Proactively manage systems.** Watch for configuration drift and automatically apply updates to maintain the expected states of your IT systems across your entire infrastructure.
5. **Scale and tune systems.** Scale and tune your infrastructure automatically to meet user and application demand based on data—including network bandwidth and latency and processor and storage use—reported by your infrastructure monitoring tools.

Read these checklists to learn more about how you can use Event-Driven Ansible:

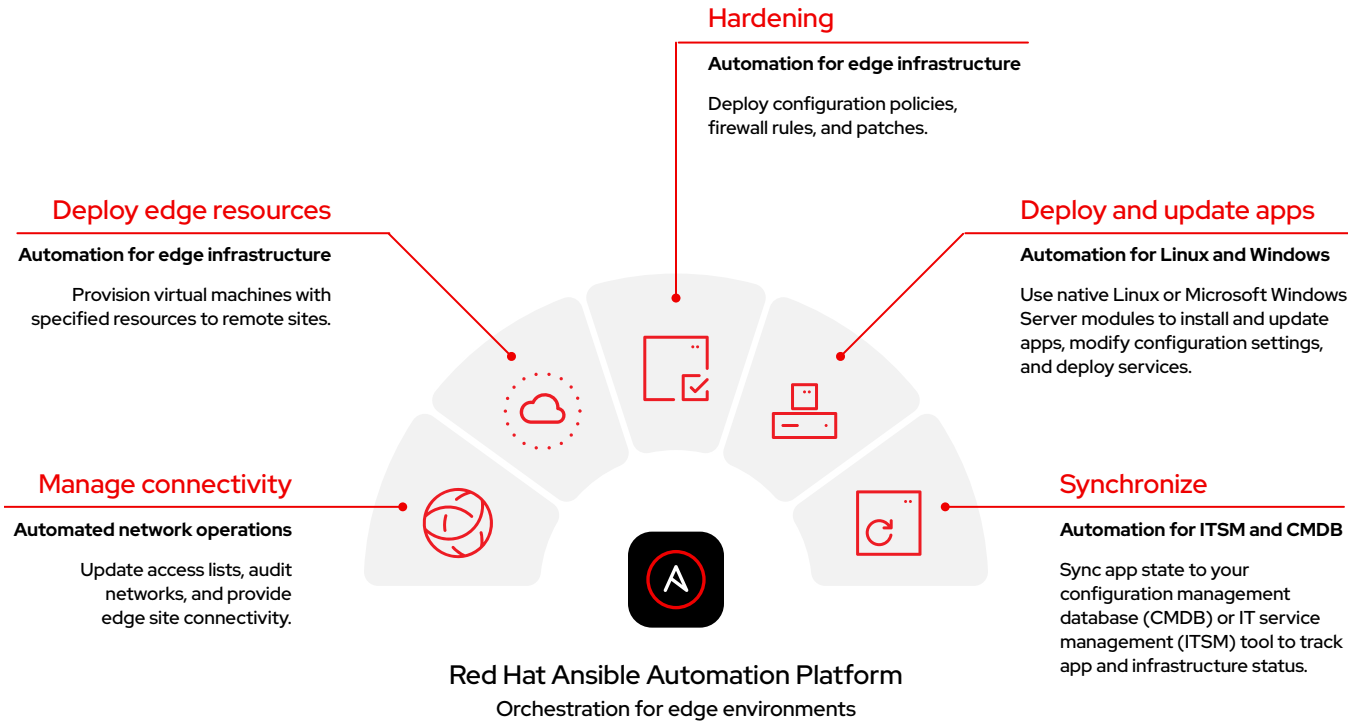
- 5 reasons to include event-driven automation in your IT strategy
- 5 ways Event-Driven Ansible automation helps you achieve more

Deploy automation at the edge

Edge computing moves processing power closer to data sources, allowing you to deploy latency-sensitive applications, gather data from a multitude of devices, and create sites that can operate even when a connection to the datacenter or cloud is lost. **Automation at the edge** can help you manage, orchestrate, and maintain your entire IT environment in a consistent and standardized manner.

Red Hat Ansible Automation Platform uses containerization to distribute and run automation across environments, helping operations teams standardize configuration and deployment from datacenters, across clouds, and all the way to edge locations. It offers a single, consistent view of your IT landscape, so your teams can reliably manage thousands of sites, network devices, and clusters. This unified approach to automation can help you lower operational expenses and deliver smoother customer experiences at edge sites, where resources are limited.

Read **Automation at the edge: 7 industry use cases and examples** to learn about industry-specific use cases for edge automation.



6 benefits of automating at the edge

1. **Focus on security and safety.** Run updates, patches, and required maintenance automatically without sending a technician to the site.
2. **Reduce downtime.** Simplify network management, reduce network failure, and boost your bottom line.
3. **Improve efficiency.** Increase performance and reduce human error with automated analysis, monitoring, and alerting.
4. **Increase scalability.** Apply configurations consistently across your infrastructure and quickly scale edge devices.
5. **Respond rapidly.** Deliver optimized user experiences with automated workflows based on real-time data and events.
6. **Ensure compliance.** Ensure systems and applications operate as defined for security and audit assessments and to meet both regulatory requirements and internal standards.

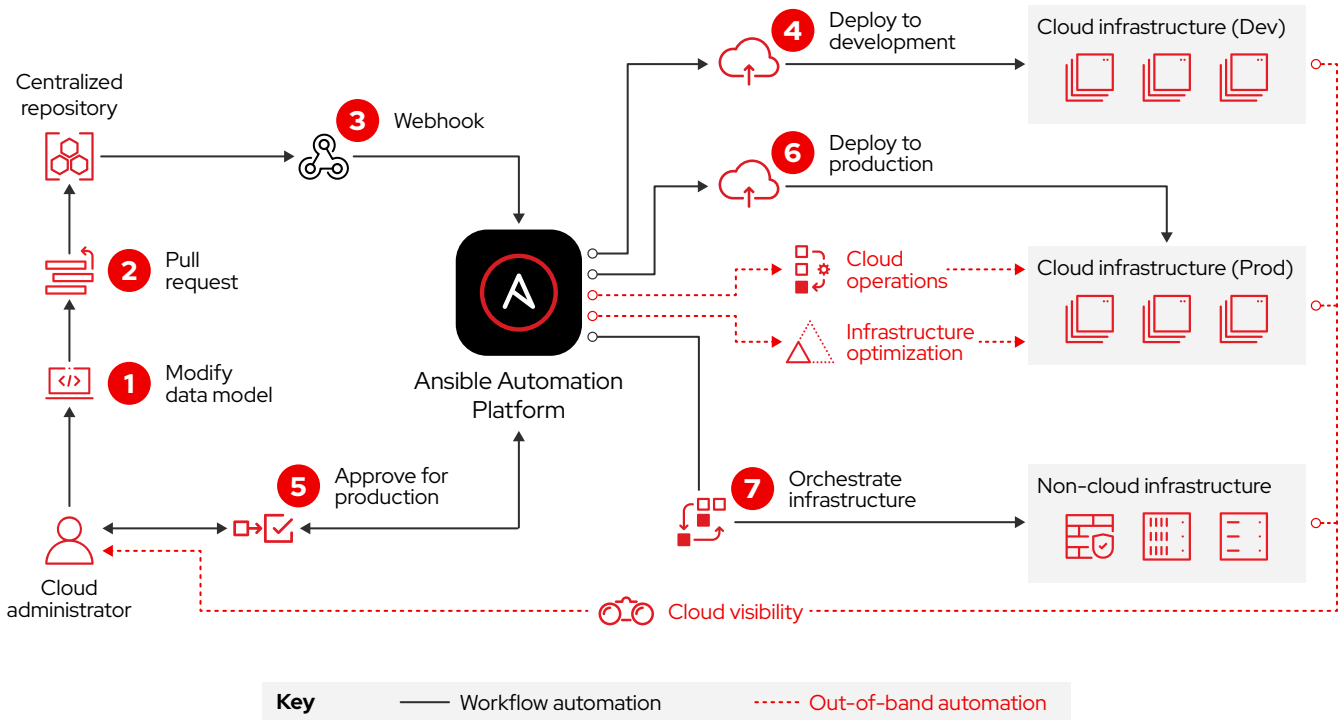
Orchestrate complete cloud workflows

Automation helps you operationalize entire hybrid and multicloud environments—from on-site datacenters to public cloud infrastructure—with streamlined orchestration and workflows. Using cloud automation, you can document, assess, and codify tasks so that they can be combined reliably and repeatedly into workflows to achieve predictable business outcomes. Cloud automation also helps you create a consistent operational framework across all IT and cloud domains.

Red Hat Ansible Automation Platform lets you automate and orchestrate all aspects of your hybrid cloud environment, from cloud resources and services to operating systems, applications, and security. It connects your existing automation, configuration, and cloud tools and processes with a common language. As a result, you can create a consistent operational framework across all cloud domains, processes, and roles, and locate your automation closer to target endpoints.

Read **Automate your hybrid cloud at scale** to learn more about automating cloud environments.

This example of a complete cloud automation workflow shows how you can use Ansible Automation Platform to orchestrate cloud resource and application life cycles using a GitOps approach.



Workflow automation

1. A cloud administrator modifies a resource definition or playbook.
2. The cloud administrator commits the changed definition or playbook to a centralized repository.
3. Webhook integration in Ansible Automation Platform allows it to notice the change and start any necessary automation.
4. Ansible Automation Platform redeploys the cloud resources to a development environment.
5. The cloud administrator approves the automated production request.
6. Ansible Automation Platform deploys the cloud resources to the production environment.
7. Ansible Automation Platform sets up and orchestrates any other off-cloud resources needed for production deployment.

Out-of-band automation

- **Cloud operations:** Ansible Automation Platform performs Day 1-2 operations, including modifications and updates, as needed.
- **Infrastructure optimization:** Ansible Automation Platform optimizes infrastructure and resources as needed.
- **Cloud visibility:** Ansible Automation Platform takes snapshots of the infrastructure for visibility and insights as needed.

Promote automation across your organization

Organization-wide automation does not happen instantly, and automation is not an all-or-nothing proposition. You need a sustainable automation strategy to guide your journey. Building your strategy requires assessment, planning, and adaptation.

Identify your business objectives

Connect automation efforts with business challenges and goals. This helps you identify where to automate and create top-down requirements for success. For example, you could automate patching to boost system security and stability and meet business needs for higher uptime.

Encourage cross-team collaboration

Use incentives to promote collaboration across your organization. Coordination allows teams to create complete automation workflows that deliver more value. Working with others also helps to cultivate shared ownership and accountability for automation.

Build trust throughout your organization

Build a centralized repository for trusted automation content. Each team should create automation content in their area of expertise and contribute it to the repository for use by other teams. Staff can include boundaries that allow others to use their content more confidently.

Share knowledge and success

Create a core team of stakeholders—often called a community of practice (CoP) or center of excellence (CoE)—that share automation best practices, experiences, and accomplishments across your organization. These teams should also help others along their automation journey.

Centralize your automation content

Choose an automation platform that provides a unified foundation for collaboration, tools, and content across your organization. Sharing tools and content in a single, trusted location allows teams to automate more efficiently and avoid duplicate efforts.

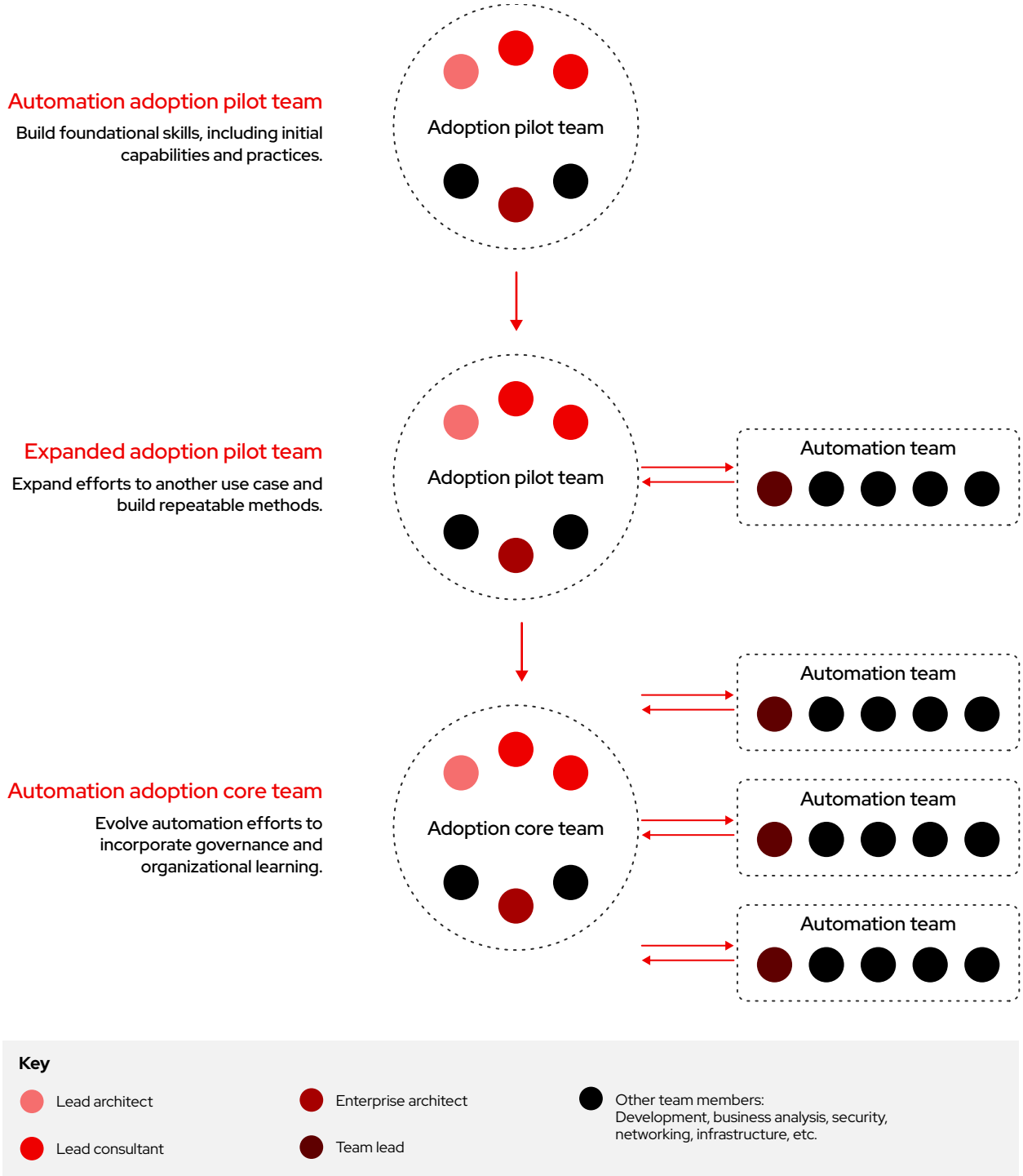
Define success

There is no single way to measure automation success—each team has unique characteristics and ambitions. Create realistic goals that align with your teams' current skills while encouraging staff to expand their abilities. Examples of long-term automation success include:

- **Adoption** across your enterprise, from vision to execution, with an emphasis on simplicity and shared knowledge.
- **Accountability** with each staff member taking responsibility for their individual goals.
- **Governance** through prescriptive processes that accomplish automation goals and produce repeatable results.
- **Security** with a simplified pipeline, repeatable and reusable practices, proactive vulnerability resolution, and automated investigation and response to incidents.
- **Standards** that provide the foundation and extensibility needed to achieve organizational and team goals.

Read **The automated enterprise** to learn more.

People are at the core of any enterprise-wide initiative, and automation is no different. In order to adopt automation across your organization, all teams—including line of business, network, security, operations, development, and infrastructure—must be on board and ready to learn new concepts and skills. Create a COP or COE team to support organization-wide adoption of automation best practices.



Resources and information

Red Hat offers many resources to help you get started and progress on your automation journey.

Try free interactive labs

These hands-on learning scenarios give you a preconfigured environment to experiment, practice, and see how automation works.

Access labs

Read product documentation

Find documentation for Ansible Automation Platform, including release notes, installation guides, and operational information.

Read documentation

Learn good practices

The Ansible good practices guide gathers recommendations from Ansible practitioners at Red Hat, consultants, developers, and others.

Access the guide

Take a free online course

The Ansible Basics: Automation Technical Overview course is a series of on-demand videos that introduce you to Ansible Automation Platform.

Take the course

Find the resources you need

Red Hat offers many resources to help get you started—and keep you learning. These are collected on the Ansible get started webpage.

Find resources

Join the Ansible community

Join the Ansible community to discuss IT challenges, collaborate on solutions, meet with industry peers, and connect on social networks.

Check it out

1088058_0424_KVM

Red Hat