Red Hat

What's New

Ansible Automation Platform 2.6

May 12, 2026

Contents

1 What's new

Release notes

Ansible Automation Platform unifies comprehensive automation capabilities, a robust ecosystem, and flexible deployment options into one strategic solution. It enables customers to automate and orchestrate workflows across domains for efficient, resilient, and consistent IT operations at scale.

From the introduction of Ansible Automation Platform 2, our promise has been to deliver an automation platform experience that allows our customers to create the automation content they need with more efficiency, manage the resulting automated workflows more effectively, and scale those automated workflows with ease across domains and environments. We continue to strive to deliver on that promise with the release of Ansible Automation Platform 2.6.

Automate at scale on proven foundation. Red Hat Ansible Automation Platform 2.6 delivers new features, platform enhancements, and strategic integrations that will help you continue to build a resilient, trusted foundation for the next generation of IT operations.

- **Unlock more value** with the new automation dashboard, which allows you to securely measure automation ROI, with customized tracking and reporting.
- **Operate more efficiently** with the new Ansible Lightspeed intelligent assistant, which harnesses generative AI to provide on-demand support, for a more intuitive platform experience.
- **Achieve new levels of scale** with the self-service automation portal, which enables platform admins to quickly and easily scale automation service delivery to new users and teams.

Because your team is not just preparing for the future, you are automating for it.

Platform services in 2.6

These are the major service versions in Ansible Automation Platform 2.6. For more current information, see the entire Life Cycle.

Ansible-core version	Ansible Automation Platform UI version	Automation controller (automation execution) version	Automation hub (automation content) version	Event-Driven Ansible controller (automation decisions) version	Platform gateway version
2.16	2.6.1	4.7.1	4.11.0	1.2.0	2.6.202510 01

New features and enhancements

Review the new features and enhancements in Red Hat Ansible Automation Platform 2.6 to maximize your automation capabilities. This release introduces the Ansible Lightspeed intelligent assistant, the self-service automation portal, and unified role-based access control.

General availability of Ansible Lightspeed intelligent assistant

The Ansible Lightspeed intelligent assistant is now generally available on Ansible Automation Platform 2.6 on OpenShift Container Platform. It is an intuitive chat interface embedded within the Ansible Automation Platform, utilizing generative artificial intelligence (AI) to answer questions about the Ansible Automation Platform.

The chat experience in the Ansible Lightspeed intelligent assistant interacts with users in their natural language prompts in English, and uses large language models (LLMs) to generate quick, accurate, and personalized responses. These responses empower users to work more efficiently, thereby improving productivity and the overall quality of their work.

To access and install Ansible Lightspeed intelligent assistant, you will need the following:

- Ansible Automation Platform 2.6 on OpenShift Container Platform
- An LLM service that is hosted on either Red Hat Enterprise Linux AI or Red Hat OpenShift AI

For more information, see Deploying the Ansible Lightspeed intelligent assistant on OpenShift Container Platform in Installing on OpenShift Container Platform.

General availability of Ansible self-service automation portal

Ansible self-service automation portal is now generally available as part of the Ansible Automation Platform subscription. The new self-service automation portal empowers platform admins to provide a streamlined "point-and-click" Ansible automation experience to a broader set of users within the organization. Users who are not Ansible experts now have a dedicated self-service portal from which they can launch a range of automation jobs.

- Installation: Deployment of self-service automation portal requires Red Hat OpenShift Container Platform using a Helm chart. A future deployment of self-service automation

portal on Red Hat Enterprise Linux 10 is planned for Technology Preview in a future asynchronous release of Ansible Automation Platform 2.6.

- Synchronizes existing automation content: Extend the reach and impact of your automation job templates, while maintaining full control and compliance.
- Seamless Integration: Uses your existing Ansible Automation Platform configuration—same logins, same security controls, same automation logic.
- Simplified Interface: A distinct, user-friendly web interface designed for business users, not automation experts.
- Guided Workflows: Step-by-step forms that walk users through automation requests without technical complexity - automatically generated from your existing job templates.
- Smart Forms: Real-time field validation, conditional and dynamic forms, and dropdown fields for Ansible Automation Platform artifacts, such as Ansible Automation Platform inventories.

General availability of Ansible automation dashboard

Automation dashboard is now generally available as part of the Ansible Automation Platform subscription. Automation dashboard is a utility you can connect to one or more Ansible Automation Platform deployments to visualize automation usage data, determine time savings, track ROI, and drive increased visibility into automation strategy, resource allocation, and prioritization of automation projects. Benefits include:

- Installation: Deployment of automation dashboard is via containerized installation only.
- Secure on-premise deployment: Simplified deployment as a self-contained, on-premise utility that runs on a dedicated RHEL 9 x86 and ARM host.
- Easy Integration: Integrates into Ansible Automation Platform 2.4, 2.5 and 2.6 instances with OAuth2 token for read-only access to pull data.
- Automated data sync: Once configured, the dashboard automatically syncs and visualizes data from connected Ansible Automation Platform instances.
- Flexible Reporting: Dashboard allows to generate and share customized PDF reports and export raw CSV data for flexible ingestion into BI tools.

For more information, see Using automation dashboard.

Configuration as Code

The ansible.platform collection now provides unified, platform-wide Role-Based Access Control (RBAC) management across Ansible Automation Platform components. New or enhanced modules include `Organization`, `Team`, `User`, `Role definitions`, `Role Assignments` (team/user). Additionally:

- You can declare the RBAC state as code and apply idempotently across services.
- Ansible collections now use a standard global environment variable prefix across components. Automation controller, Automation hub, and Event-Driven Ansible all use a new standard of “AAP_” instead of "COMPONENT_". For example, `aap_hostname`. See the documentation in Automation hub for more information.

Service accounts

- Service accounts, created in console.redhat.com, can now be used to manage subscriptions in Ansible Automation Platform. Manifest files and basic authentication may still be used for this purpose as well.
- Service accounts are now required in order to send data to automation analytics.

Event-Driven Ansible (Automation decisions)

Event-Driven Ansible includes several key enhancements in the Ansible Automation Platform 2.6 release that improve performance, simplify operations, and expand the platform’s capabilities across security, networking, and event processing.

- **External secret management**: Event-Driven Ansible now supports external secret management systems, achieving parity with Automation controller. This includes support for HashiCorp Vault, CyberArk, Microsoft Azure Key Vault, and AWS Secrets Manager.
- **Editable project URLs**: You can now edit the source control URL for existing Event-Driven Ansible projects, providing greater flexibility to adapt to repository changes.
- **Improved job auditing**: A new label is automatically added to jobs triggered by Event-Driven Ansible, along with support for custom labels. This allows for more efficient tracing and auditing of event-triggered automations.
- **Kafka enhancements**: The Kafka source plugin now supports multiple topics and allows the use of regular expressions and wildcards. Additionally, it now supports GSSAPI for enhanced authentication.

- **New event filter**: A new filter plugin, `event_splitter`, is available to handle and process nested events more effectively.
- **Rulebook concurrency key**: Rulebooks now support a concurrency key, enabling you to group events by resource to ensure they are processed sequentially.

Installation updates

Containerized installation

Updated system requirements for containerized installation of Ansible Automation Platform include:

- The Red Hat Enterprise Linux 9.2 operating system requirement was updated to 9.4 or later minor versions of Red Hat Enterprise Linux 9. Red Hat Enterprise Linux 10 system requirements are unchanged.
- PostgreSQL 15, 16, and 17 are now supported for customer provided (external) databases.

> **NOTE:**
> External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.

For more information see System requirements in *Containerized installation*.

Operator installation

Updated system requirements for Ansible Automation Platform Operator on Red Hat OpenShift Container Platform include:

- The Red Hat Enterprise Linux 9.2 operating system requirement was updated to 9.4 or later minor versions of Red Hat Enterprise Linux 9. Red Hat Enterprise Linux 10 system requirements are unchanged.
- PostgreSQL 16 and 17 are now supported for customer-provided (external) databases.

> **NOTE:**
> External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.

For more information about the Ansible Automation Platform Operator system requirements, see Choose a deployment method and topology.

RPM installation

Updated system requirements for RPM installation of Ansible Automation Platform 2.6 include:

- Ansible Automation Platform RPM installer was deprecated in 2.5 and will be removed in Ansible Automation Platform 2.7. The RPM installer will be supported for RHEL 9 during the lifecycle of Ansible Automation Platform 2.6 to support migrations to existing supported topologies. See the support matrix for more information on upgrade and migration paths.
- Red Hat Enterprise Linux 9.2 operating system requirement was updated to 9.4 or later minor versions of Red Hat Enterprise Linux 9. Red Hat Enterprise Linux 8 is no longer supported.
- Red Hat Enterprise Linux 10 is not supported for RPM installations. See support matrix for more information on supported upgrade and migration paths.
- PostgreSQL 16 and 17 are now supported for customer-provided (external) databases.

> **NOTE:**
> External databases using PostgreSQL 16 or 17 must rely on external backup and restore processes. Backup and restore functionality is dependent on utilities provided with PostgreSQL 15.

For more information, see System requirements in *RPM installation*.

Upgrade paths

The following table outlines the supported upgrade paths for Ansible Automation Platform 2.6.

> **NOTE:**
> The RPM-based upgrade paths are deprecated and will be removed in Ansible Automation Platform 2.7.

Starting Deployment	Upgrade Deployment
2.4 RPM single automation controller node	2.6 RPM growth
2.4 RPM single node automation controller and automation hub	2.6 RPM growth
2.4 RPM multi node automation controller	2.6 RPM enterprise
2.4 RPM multi node automation controller and automation hub	2.6 RPM enterprise
2.5 RPM growth	2.6 RPM growth

Starting Deployment	Upgrade Deployment
2.5 RPM enterprise	2.6 RPM enterprise
2.5 Container growth	2.6 Container growth
2.5 Container enterprise	2.6 Container enterprise
2.4 Operator single automation controller node	2.6 Operator growth
2.4 Operator single node automation controller and automation hub	2.6 Operator growth
2.4 Operator multi node automation controller	2.6 Operator enterprise
2.4 Operator multi node automation controller and automation hub	2.6 Operator enterprise
2.5 Operator growth	2.6 Operator growth
2.5 Operator enterprise	2.6 Operator enterprise

Migration paths

The following table outlines the supported migration paths for Ansible Automation Platform 2.6. Migration involves transitioning between deployment types, such as from an RPM to a containerized installation. This process is exclusively supported between identical versions (for example, 2.6 to 2.6).

Source environment	Target environment
RPM-based Ansible Automation Platform	Container-based Ansible Automation Platform
RPM-based Ansible Automation Platform	OpenShift Container Platform
RPM-based Ansible Automation Platform	Managed Ansible Automation Platform
Container-based Ansible Automation Platform	OpenShift Container Platform
Container-based Ansible Automation Platform	Managed Ansible Automation Platform

Overview of upgrade improvements

Changes in 2.6 improve the overall upgrade experience, as detailed in the following sections:

- Upgrading from 2.5 to 2.6
- Upgrading from 2.4 to 2.6

NOTE:
You must be on the latest version of 2.4 or 2.5 before you upgrade to 2.6.

Upgrading from 2.5 to 2.6

Upgrading from 2.5 to 2.6 does not involve changes to the platform infrastructure requirements, architecture, or services. The improvements described in the 2.4 to 2.6 upgrade path are also present in the 2.5 to 2.6 upgrade path; however, the platform gateway service is already in place in 2.5.

Additionally, note the following:

- If you upgraded from 2.4 to 2.5, you must migrate your authentication methods and users before upgrading to 2.6 as that legacy authenticator functionality was removed.
- When you upgrade to 2.6, the system removes any users that the 2.4 to 2.5 upgrade did not fully migrate. The users that have previously merged their user records while on 2.5 will remain to function as is for 2.6.
- Upgrading to 2.6 prevents 2.4 automation controller users who never successfully logged into 2.5 from logging into the platform-gateway. These users retain backwards compatibility for direct Automation Execution access but cannot access the full platform. Ensure all users planning to leverage 2.6 have successfully logged into 2.5 prior to upgrading.
- Unified RBAC management across Ansible Automation Platform components: All Ansible Automation Platform collections, which support the Configuration-as-Code (CaC) approach, now use a standard global environment variable name and module variable name across Ansible Automation Platform components. For more details, see What's new around RBAC in 2.6, What's changed around RBAC for users moving from 2.5 to 2.6, and `ansible.platform` documentation in automation hub.

For more information about upgrading, see the upgrade document for your deployment type:

- Containerized
- RPM
- OpenShift Container Platform

NOTE:
Upgrades from the latest 2.5 version to 2.6 are supported with all deployment types: RPM, containerized, and OpenShift Container Platform deployments.

Upgrading from 2.4 to 2.6

Note the following when upgrading from 2.4 to 2.6:

- **Upgrades from 2.4**: Ansible Automation Platform supports upgrading directly from the latest 2.4 version to 2.6. Directly upgrading to 2.6 is the recommended upgrade path from 2.4, as a number of improvements in 2.6 simplify and improve the upgrade experience.

  NOTE:
  You can upgrade directly from the latest 2.4 version to 2.6 with RPM and OpenShift Container Platform deployments. However, upgrading Event-Driven Ansible 2.4 or from the 2.4 containerized deployment is not supported, as both features were Tech Preview in 2.4.

  For more information, see the upgrade document for your deployment type. Either RPM, or OpenShift Container Platform.

- **Infrastructure changes**: Ansible Automation Platform RPM deployments require additional infrastructure compared with 2.4, due to the addition of the platform gateway service. Infrastructure needs vary depending on factors such as whether you implement a growth or an enterprise deployment.
  For details about infrastructure and inventory file changes in various upgrade scenarios, see Infrastructure changes for RPM deployments.

- **Authentication changes**: Enterprise authentication configuration and mappings, including SAML, LDAP, and OIDC, automatically move from automation controller 2.4 to platform gateway 2.6 during the upgrade. Although these settings migrate automatically within Ansible Automation Platform, you must update the callback URLs in your external Identity Provider (IdP) settings to point to platform gateway. For more information, see Authentication provider migration behavior.
  See Access management and authentication for information about authentication options in general.

  NOTE:
  Authentication upgrade improvements apply to RPM and OpenShift Container Platform deployments. Upgrades from the 2.4 containerized deployment Tech Preview release are not supported. Additionally, upgrading Event-Driven Ansible 2.4 is not supported.

- **Identify access management changes**: All automation controller Identity Access Management (IAM) data moves from automation controller 2.4 to the platform gateway in 2.6 as part of the upgrade process. With automation controller 2.4 as the default

source of IAM data for the platform gateway in 2.6, users retain their memberships and are assigned appropriate platform-level roles in 2.6.
As part of the upgrade process:

- Users, teams, organizations, their memberships, and common roles in 2.4 move from automation controller 2.4 to the platform gateway in 2.6.
- Administrators in automation controller 2.4 become platform gateway administrators in 2.6.
- Controller admins in 2.4 become platform gateway admins in 2.6.
  The more organizations, teams, and users being migrated during an upgrade, the longer the upgrade takes. As an example, upgrading and migrating 4,000 users, 400 teams, and 40 organizations may take close to two hours.

> **NOTE:**
> Identity access management changes apply to RPM and OpenShift Container Platform deployments. Upgrades from the 2.4 containerized deployment Tech Preview release are not supported.

See Data movement during upgrade to 2.6 for more information.

- **API changes**: Some APIs are being deprecated in 2.6. See API changes for more information.
- **Unified RBAC management across Ansible Automation Platform components**: All Ansible Automation Platform collections, which support the Configuration-as-Code (CaC) approach, now use a standard global environment variable name and module variable name across Ansible Automation Platform components. For more details, see What's new around RBAC in 2.6, What's changed around RBAC for users moving from 2.5 to 2.6, and `ansible.platform` documentation in automation hub.

Platform UI

Ansible Automation Platform 2.6 was delivered with the goal to simplify the UI, improve the relationship between user interface elements, and maintain the association between users, organizations, teams, and roles.

Within the Platform UI, the role based access controls (RBAC) have been centralized to give administrators control of users across the entire platform. The centralized RBAC has introduced additional APIs and expanded the scope of those APIs to allow the assignment of roles across any of the platform resources. The details of these changes are reflected within the API changes.

The UI has also been updated to the latest version of Patternfly, which brings significant updates and refinements aiming to enhance user experience, performance, and developer efficiency.

Related information
Support matrix for upgrade scenarios

RPM upgrade
Updating containerized Ansible Automation Platform
Upgrading Red Hat Ansible Automation Platform Operator on Red Hat OpenShift Container Platform
Ansible Automation Platform migration

Deprecated features

Deprecated functionality is still included in Ansible Automation Platform and continues to be supported during this version's support cycle. However, the functionality will be removed in a future release of Ansible Automation Platform and is not recommended for new deployments.

The following table provides information about features that were deprecated in Ansible Automation Platform 2.5:

Component	Feature
Access to service APIs for automation controller, automation hub, and Event-Driven Ansible	With the addition of platform gateway, a number of service-specific API endpoints are deprecated because their functionality will be removed or superseded with other capabilities in a future release. Ansible Automation Platform 2.5 and 2.6 expose API access to individual services (automation controller, private automation hub, Event-Driven Ansible) to maintain compatibility with existing REST API integrations. This access will be removed in a future release. For detailed information, see API changes in *Planning your upgrade*.
Installer	The Ansible Automation Platform installer using Red Hat Enterprise Linux RPMs was deprecated (announced) in 2.5 and will be removed in Ansible Automation Platform 2.7. The RPM installer will be supported for Red Hat Enterprise Linux 9 during the lifecycle of Ansible Automation Platform 2.6 to support migrations to existing supported topologies. Users are encouraged to migrate to the containerized topology on Red Hat Enterprise Linux or to the OpenShift Container Platform Operator installation method. See the support matrix for more information on upgrade and migration paths.

Component	Feature
Ansible-core	<pre>uri module: - Using 'yes' or 'no' for 'follow_redirects' parameter is deprecated. yum_repository: - deprecated parameters: - 'keepcache' - 'async' - "deltarpm_metadata_percentage" - "gpgcakey" - "http_caching" - "keepalive" - "metadata_expire_filter" - "mirrorlist_expire" - "protect" - "ssl_check_cert_permissions" - "ui_repoid_vars"` url lookup:     - Using `yes` or `no` for `follow_redirects` parameter is deprecated.</pre>
Execution environment	Removing `cisco.asa` from ee-supported as it is being deprecated Removing `ibm.qradar` from ee-supported as it is being deprecated
Certified Collections	An `ansible.platform` collection is available as the preferred collection to replace the service-specific `ansible.controller`, `ansible.hub`, and `ansible.eda` collections. These service-specific collections will be replaced by `ansible.platform` after 2.6.
Ansible code bot code bot	The code bot (as described in the Red Hat Ansible Lightspeed with IBM watsonx Code Assistant user guide) is being deprecated, and will be retired on December 31, 2025.
Ansible Content	Deprecation of the Notification Service for ServiceNow, which will not be supported on the ServiceNow Zurich and later releases. Support will end when the Yokohama release is end-of-life.

Removed features

Removed features are those that were deprecated in earlier releases. They are now removed from the Ansible Automation Platform 2.5, and will no longer be supported.

Component	Feature
Event-Driven Ansible controller	Removal of `max_running_activations setting in eda-controller`
Platform gateway	Legacy Authenticators that were added during an upgrade from 2.4 to 2.5 will no longer be present

Changed features

Changed features are not deprecated and will continue to be supported until further notice.

The following table provides information about features that are changed in Ansible Automation Platform 2.6:

Component	Feature
Platform gateway	The determination for matching to existing user records upon login has changed from previous versions. The new process leverages email address as the primary matching criteria for existing user accounts across multiple authentication methods. See Configure central authentication for Ansible Automation Platform for more details. Within 2.5, each authentication method would result in a user record being created regardless of the email matching from the different IdP sources.
Platform-operator, Ansible Automation Platform Hub Operator	Added `postgres_extra_settings` to Ansible Automation Platform operators to apply PostgreSQL configuration file level changes to managed Postgres.
Platform-operator, Event-Driven Ansible	Added `postgres_extra_settings` to Ansible Automation Platform operators to apply PostgreSQL configuration file level changes to managed Postgres.

Component	Feature
Platform-operator, gateway-operator	Added `postgres_extra_settings` to Ansible Automation Platform operators to apply PostgreSQL configuration file level changes to managed Postgres.

Known issues

This section provides information about known issues in Ansible Automation Platform 2.6.

- For role based authentication mappings, the role list includes all roles within the platform. Only the role assignments of Org Admin, Org Member, Team Admin, Team Member, and Platform Auditor are supported at this time. The list will be limited to only those that can be applied at a platform level in a subsequent release.
- If you have an existing deployment of Red Hat Ansible Lightspeed on Ansible Automation Platform 2.5, upgrading to Ansible Automation Platform 2.6 will cause your Red Hat Ansible Lightspeed deployment to fail. To avoid this failure, do not upgrade to Ansible Automation Platform 2.6 until a forthcoming patch is released on October 22, 2025. However, new deployments of Red Hat Ansible Lightspeed will work correctly on Ansible Automation Platform 2.6.(AAP-54064)
  For more information, see Ansible Lightspeed upgrade fails when upgrading Ansible Automation Platform 2.5 to 2.6.
- Automation controller in Ansible Automation Platform 2.4 allowed customers to enter an encrypted private key in SAML configuration without raising an error. If request signing was not enabled in the authenticator and the SAML IdP, then the Ansible Automation Platform administrator would not know that encrypted keys were not supported. Encrypted keys not supported in Ansible Automation Platform 2.6 authenticators. The platform alerts users that encrypted keys are not supported. However, when upgrading from Ansible Automation Platform 2.4 to 2.6, customers must replace encrypted private keys with unencrypted private keys in their SAML authenticators to prevent migration errors for the authenticator to platform gateway. If you skip this step, the authenticator is not migrated as part of the upgrade. The SAML authenticator must then be recreated manually by a local administrator to re-enable authentication. This might delay SSO users from logging back into the platform after the upgrade.

Fixed issues

This section provides information about fixed issues in Ansible Automation Platform 2.6.

Ansible Automation Platform

NOTE:
Ansible Automation Platform 2.6 also includes the fixes from the latest 2.5 patch release. For more information, see Ansible Automation Platform patch release September 23, 2025.

Ansible Automation Platform

- The `SOCIAL_AUTH_USERNAME_IS_FULL_EMAIL` configuration parameter now functions as expected, allowing social auth logins to set the platform gateway username to the user's email when enabled.(AAP-49736)

RPM-based Ansible Automation Platform

- Fixed an issue where installer managed CA certificates were discovered but not used by the installer.(AAP-53335)

Patch releases

The following release notes detail the updates for the Ansible Automation Platform patch releases.

Security, bug fixes, and enhancements for Ansible Automation Platform are released as asynchronous erratas. All Ansible Automation Platform erratas are available on the Download Red Hat Ansible Automation Platform page.

As a Red Hat Customer Portal user, you can enable errata notifications in the account settings for Red Hat Subscription Management (RHSM). When errata notifications are enabled, you receive notifications through email whenever new erratas relevant to your registered systems are released.

NOTE:
Red Hat Customer Portal user accounts must have systems registered and consuming Ansible Automation Platform entitlements for Ansible Automation Platform errata notification emails to generate.

The patch releases section of the release notes will be updated over time to give notes on enhancements and bug fixes for patch releases of Ansible Automation Platform.

Related information
Red Hat Ansible Automation Platform Life Cycle
What is a CVE?
Red Hat CVE Database

Ansible Automation Platform patch release May 4, 2026

The following release notes detail the updates for the Ansible Automation Platform patch released on May 4, 2026

This release includes the following components and versions:

Release Date	Component versions
May 4, 2026	• Automation controller 4.7.11 • Automation hub 4.11.8 • Event-Driven Ansible 1.2.8 • Container-based installer Ansible Automation Platform (bundle) 2.6-8 • Container-based installer Ansible Automation Platform (online) 2.6-8 • Receptor 1.6.4 • RPM-based installer Ansible Automation Platform (bundle) 2.6-6.1 • RPM-based installer Ansible Automation Platform (online) 2.6-6

CSV Versions in this release:

- Namespace-scoped bundle: aap-operator.v2.6.0-0.1777410689
- Cluster-scoped bundle: aap-operator.v2.6.0-0.1777410680

Overview

This Ansible Automation Platform 2.6 async (20260422) release includes a set of targeted enhancements across installation and platform UX, plus a large batch of security (CVE) remediations and bug fixes across multiple AAP components.

Enhancements

Automation hub

- Added verification that Hub supports Execution Environments with PQC signatures. (AAP-71606)

Container-based installer Ansible Automation Platform

- Fixed the preflight check to allow hop nodes to run on systems with less than 16GB of RAM. (AAP-71341)

Red Hat Ansible Lightspeed

- Support for llama-stack 0.4.3.(AAP-69996)
- Support for llama-stack 0.4.3.(AAP-65012)

Ansible Automation Platform Operator

- Allows the ability to disable backup db compression per component using the use_db_compression parameter (default: true). (AAP-69747)

Ansible Automation Platform ui

- Private flags only appear in UI when enabled - this applies uniformly to both runtime and install-time private flags. Private runtime flags can be toggled off via the UI, which causes them to disappear. This prevents users from easily discovering feature flags that are not meant to be advertised to all customers.(AAP-69669)
- Added a Feature Flags page under Settings that allows platform administrators to view feature flags and toggle runtime flags on or off without restarting services.(AAP-69001)

Automation controller

- Sets `XDG_CONFIG_HOME=/tmp/.config` in the `Containerfile` so podman-remote can write its config at runtime.
- Fixes `handle_removed_image` task failing with `RuntimeError`: Error running command in containerized installer deployments. (AAP-68260)

Deprecated

Ansible Automation Platform Operator

- `old_postgres_configuration_secret` has been deprecated for automation controller and event-driven ansible.
- `postgres_migrant_configuration_secret` has been deprecated for automation hub. (AAP-68604)

Receptor

- Address CVE-2025-68121.(AAP-65759)

CVE

General

- CVE-2026-6266: Account hijacking and unauthorized access via unverified email linking. This affects the following components:
  - `automation-controller` for {PlatformNameShort} 2.5 and 2.6.
  - `automation-gateway` for {PlatformNameShort} 2.5 and 2.6.
  - `python3.12-django-ansible-base` for {PlatformNameShort} 2.5 and 2.6.
  - `ansible-automation-platform-26/controller-rhel9` for Ansible Automation Platform 2.6 only.

  - `ansible-automation-platform-26/gateway-rhel9` for Ansible Automation Platform 2.6 only.

Execution Environment

- CVE-2026-23490 - pyasn1: Denial of Service due to memory exhaustion from malformed RELATIVE-OID in:
  - `ansible-automation-platform-26/ee-supported-rhel9` for Ansible Automation Platform 2.6. AAP-72593
- CVE-2026-27459 - pyOpenSSL: DTLS cookie callback buffer overflow in:
  - ansible-automation-platform-26/ee-supported-rhel9 for Ansible Automation Platform 2.6. AAP-68956
- CVE-2026-32274 - Black: Arbitrary file writes from unsanitized user input in cache file name in:
  - `ansible-automation-platform-26/ee-minimal-rhel9` for Ansible Automation Platform 2.6. AAP-68419
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
  - ansible-automation-platform-26/ee-supported-rhel9 for Ansible Automation Platform 2.6. AAP-68399

Automation controller

- CVE-2025-14550 - Django: Denial of Service via crafted request with duplicate headers in:
  - automation-controller for Ansible Automation Platform 2.6. AAP-64818
- CVE-2025-69534 - markdown: Denial of Service via malformed HTML-like sequences in:
  - automation-controller for Ansible Automation Platform 2.6. AAP-67446
- CVE-2026-26007 - cryptography: Subgroup Attack due to missing subgroup validation for SECT curves in:
  - automation-controller for Ansible Automation Platform 2.6. AAP-65413
- CVE-2026-27459 - pyOpenSSL: DTLS cookie callback buffer overflow in:
  - automation-controller for Ansible Automation Platform 2.6. AAP-68960
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
  - automation-controller for Ansible Automation Platform 2.6. AAP-68405

Automation hub

- CVE-2026-27459 - pyOpenSSL: DTLS cookie callback buffer overflow in:
  - ansible-automation-platform-26/hub-rhel9 for Ansible Automation Platform 2.6. AAP-68957
- CVE-2026-32274 - Black: Arbitrary file writes from unsanitized user input in cache file name in:
  - ansible-automation-platform-26/hub-rhel9 for Ansible Automation Platform 2.6. AAP-68421
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
  - ansible-automation-platform-26/hub-rhel9 for Ansible Automation Platform 2.6. AAP-68401

Platform Gateway

- CVE-2026-27459 - pyasn1: Denial of Service via unbounded recursion in ASN.1 decoding in:
  - ansible-automation-platform-26/gateway-rhel9 for Ansible Automation Platform 2.6. AAP-69035
- CVE-2026-27606 - Rollup: Remote Code Execution via Path Traversal Vulnerability in:
  - ansible-automation-platform-26/gateway-rhel9 for Ansible Automation Platform 2.6. AAP-66536
- CVE-2026-29074 - SVGO: Denial of Service via XML entity expansion in:
  - automation-gateway for Ansible Automation Platform 2.6. AAP-68531
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
  - ansible-automation-platform-26/gateway-rhel9 for Ansible Automation Platform 2.6. AAP-68400
- CVE-2026-33154 - Dynaconf: Arbitrary code execution via Server-Side Template Injection in:
  - ansible-automation-platform-26/gateway-rhel9 for Ansible Automation Platform 2.6. AAP-69466

Ansible Automation Platform UI

- CVE-2026-26996 - minimatch: Denial of Service via specially crafted glob patterns in:
  - automation-platform-ui for Ansible Automation Platform 2.6. AAP-66292
- CVE-2026-27606 - Rollup: Remote Code Execution via Path Traversal Vulnerability in:
  - automation-platform-ui for Ansible Automation Platform 2.6. AAP-66535

Event-Driven Ansible

- CVE-2026-24049 - wheel: Privilege escalation or arbitrary code execution via malicious wheel file unpacking in:
  - ansible-automation-platform-26/eda-controller-rhel9-operator for Ansible Automation Platform 2.6. AAP-63863
- CVE-2026-26007 - cryptography: Subgroup Attack due to missing subgroup validation for SECT curves in:
  - ansible-automation-platform-26/eda-controller-rhel9 for Ansible Automation Platform 2.6. AAP-65406
- CVE-2026-27459 - pyOpenSSL: DTLS cookie callback buffer overflow in:
  - ansible-automation-platform-26/eda-controller-rhel9 for Ansible Automation Platform 2.6. AAP-68954
- CVE-2026-30922 - pyasn1: Denial of Service via unbounded recursion in:
  - ansible-automation-platform-26/eda-controller-rhel9 for Ansible Automation Platform 2.6. AAP-69032
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
  - ansible-automation-platform-26/eda-controller-rhel9 for Ansible Automation Platform 2.6. AAP-68398
- CVE-2026-33154 - Dynaconf: Arbitrary code execution via Server-Side Template Injection in:
  - ansible-automation-platform-26/eda-controller-rhel9 for Ansible Automation Platform 2.6. AAP-69465

Red Hat Ansible Lightspeed

- CVE-2025-69227 - aiohttp: Denial of Service via specially crafted POST request in:

    - ansible-automation-platform/ansible-lightspeed-service-container(2.6) for Ansible Automation Platform 2.6. AAP-65586
    - ansible-automation-platform/ansible-lightspeed-chatbot-container(2.6) for Ansible Automation Platform 2.6. AAP-65585
- CVE-2025-69228 - aiohttp: Denial of Service via memory exhaustion from crafted POST request in:
    - ansible-automation-platform-26/ansible-lightspeed-service-container(2.6) for Ansible Automation Platform 2.6. AAP-65629
    - ansible-automation-platform/ansible-lightspeed-chatbot-container(2.6) for Ansible Automation Platform 2.6. AAP-65627
- CVE-2026-0598 - Broken Object Level Authorization leading to cross-user AI conversation context injection in:
    - ansible-automation-platform/ansible-wisdom-service for Ansible Automation Platform 2.6. AAP-64145
- CVE-2026-26007 - cryptography: Subgroup Attack due to missing subgroup validation for SECT curves in:
    - ansible-automation-platform-26/mcp-tools-rhel9 for Ansible Automation Platform 2.6. AAP-71204
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-71203
    - ansible-automation-platform-26/lightspeed-chatbot-rhel9 for Ansible Automation Platform 2.6. AAP-71202
- CVE-2026-27459 - pyOpenSSL: DTLS cookie callback buffer overflow in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-68958
- CVE-2026-29074 - SVGO: Denial of Service via XML entity expansion in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-68528
- CVE-2026-30922 - pyasn1: Denial of Service via unbounded recursion in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-69041
- CVE-2026-31812 - quinn-proto: Denial of Service via crafted QUIC Initial packet in:
    - ansible-automation-platform-26/lightspeed-chatbot-rhel9 for Ansible Automation Platform 2.6. AAP-68140
- CVE-2026-32597 - PyJWT accepts unknown crit header extensions (RFC 7515 §4.1.11 MUST violation) in:
    - ansible-automation-platform-26/mcp-tools-rhel9 for Ansible Automation Platform 2.6. AAP-68404

    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-68403
    - ansible-automation-platform-26/lightspeed-chatbot-rhel9 for Ansible Automation Platform 2.6. AAP-68402
- CVE-2026-33154 - Dynaconf: Arbitrary code execution via Server-Side Template Injection in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-69468
- CVE-2026-39373 - JWCrypto: Memory exhaustion via crafted compressed JWE tokens in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-71150
- CVE-2026-4800 - lodash: Arbitrary code execution via untrusted input in template imports in:
    - ansible-automation-platform-26/lightspeed-rhel9 for Ansible Automation Platform 2.6. AAP-70458

Ansible Automation Platform security

- CVE-2026-35029 - LiteLLM: Remote code execution and privilege escalation via unrestricted proxy configuration endpoint in:
    - redhat-user-workloads/lightspeed-chatbot-rhel9 for Ansible Automation Platform 2.6. AAP-70909
- CVE-2026-35030 - LiteLLM: Authentication bypass and privilege escalation via OIDC userinfo cache key collision in:
    - redhat-user-workloads/lightspeed-chatbot-rhel9 for Ansible Automation Platform 2.6. AAP-70913
- CVE-2026-4926 - path-to-regexp: Denial of Service via crafted regular expressions in:
    - ansible-automation-platform-tech-preview/mcp-server-rhel9 for Ansible Automation Platform 2.6. AAP-70022

Receptor

- CVE-2026-25679 - Incorrect parsing of IPv6 host literals in net/url in:
    - ansible-automation-platform-26/receptor-rhel9 for Ansible Automation Platform 2.6. AAP-68747
    - receptor for Ansible Automation Platform 2.6. AAP-68731

- CVE-2026-27137 - Incorrect enforcement of email constraints in crypto/x509 in:
  - ansible-automation-platform-26/receptor-rhel9 for Ansible Automation Platform 2.6. AAP-68737

Bug fixes

Platform gateway

- Fixed an issue where organization administrators could not view, modify, or remove permissions on teams outside of their organization.(AAP-72502)

Automation Hub

- Fixed an issue where the Automation Hub OpenAPI specification was missing service_index endpoints.(AAP-72227)
- Fixed an issue where artifact download view counting could return an error instead of correctly using name/namespace.(AAP-71346)

Red Hat Lightspeed

- Fixed an issue where the containerized {RHLightspeed} install did not correctly configure the Azure OpenAI provider base URL for Llama Stack 0.4.3.(AAP-72046)
- Fixed an issue where the containerized {RHLightspeed} install did not correctly configure the Azure OpenAI provider base URL for Llama Stack 0.4.3.(AAP-71979)
- Fixed an issue where the /api/lightspeed/v1/ai/chat endpoint response schema could deviate from the documented API specification.(AAP-70666)
- Fixed an issue where MCP-enabled prompts could fail due to max_tokens handling and provider defaults in lightspeed-stack-providers.(AAP-70396)
- Fixed an issue where the wisdom-manage shell command output was impacted by the Django 5.2 verbosity level change.(AAP-69164)
- Fixed an issue where ALIA/Lightspeed backups were abnormally large due to unnecessary files being included.(AAP-68774)
- Fixed an issue where ALIA/Lightspeed backups were abnormally large due to unnecessary files being included.(AAP-67911)

Container-based installer Ansible Automation Platform

- Fixed an issue where component TLS certificates were not regenerated on certain CA certificate changes.(AAP-71956)
- Fixed an issue where the Redis hostname could fail to be set in disconnected containerized installer environments.(AAP-71493)
- Fixed an issue where the 2.6 bundle installer could fail when PCP was enabled with a metrics service host in inventory, by ensuring the PCP image is loaded on Automation Metrics nodes.(AAP-71026)

Django ansible base

- Fixed an issue where a fresh installation could immediately show a “RoleDefinition matching query does not exist” error during resource sync.(AAP-71868)
- Fixed an issue where periodic resource sync between Controller and Gateway could delete valid role assignments when pagination failed mid-fetch.(AAP-71775)

Content

- Fixed an issue where the ansible.controller collection job_template module did not support Bitbucket webhooks.(AAP-71827)

Event-Drive Ansible

- Fixed an issue where projects could be deleted while a project sync was running. (AAP-71406)
- Fixed an issue where the EDA event-stream node tag in gateway config could be incorrect, causing routing issues to EDA event-stream.(AAP-69827)

Automation controller

- Fixed an issue where nested workflows could apply incorrect variable precedence when set_stats artifacts were passed via extra_vars.(AAP-70756)
- Fixed an issue where object creation could be significantly slower in organizations with large numbers of resources, by reducing RoleEvaluation object creation overhead.(AAP-70752)
- Fixed an issue where inventory imports with large numbers of changes could take an excessive amount of time.(AAP-70377)
- Fixed an issue where concurrent jobs could incorrectly clear host facts due to a race condition.(AAP-69262)
- Fixed an issue where job cancellation did not reliably propagate to dependent jobs in workflows.(AAP-68975)
- Fixed an issue where project_update.yml could fail with a jinja2 error when using custom execution environment images with newer ansible-core versions.(AAP-68783)

Ansible Automation Platform Operator

- Fixed an issue where the Gateway Operator stored database passwords unencrypted, by removing postgresql-init ConfigMap and switching to runtime-executed postgresql modules.(AAP-70404)
- Fixed an issue where Automation Hub backup ignored postgres_image and postgres_image_version, causing it to always use the default PostgreSQL image. (AAP-69856)
- Fixed an issue where operator event creation could fail with a time-parsing error that masked the underlying error message.(AAP-69634)
- Fixed an issue where CRD validation for _image and _image_version fields was missing for installer operators.(AAP-68765)
- Fixed an issue where users could not override nested restore parameters (including `no_log`) in `AnsibleAutomationPlatformRestore.(` AAP-68242)

Ansible Automation Platform ui

- Fixed an issue where unthrottled WebSocket refresh events caused excessive Jobs list API requests, leading to queued requests and an unresponsive UI under high concurrency. (AAP-70349)

- Fixed an issue where the Assign Roles wizard did not correctly show “System” as a resource type when assigning custom roles.(AAP-67506)
- Fixed an issue where OAuth authorization could fail to redirect correctly after Keycloak SSO because the next parameter was not preserved.(AAP-59343)

Receptor

- Fixed an issue where the work results command could emit misleading warnings during connection shutdown.(AAP-43847)

Ansible Automation Platform patch release March 25, 2026

The following release notes detail the updates for the Ansible Automation Platform patch released on March 25, 2026.

This release includes the following components and versions:

Release Date	Component versions
March 25, 2026	• Automation controller 4.7.19 • Automation hub 4.11.7 • Event-Driven Ansible 1.2.7 • Container-based installer Ansible Automation Platform (bundle) 2.6-7 • Container-based installer Ansible Automation Platform (online) 2.6-7 • Receptor 1.6.4 • RPM-based installer Ansible Automation Platform (bundle) 2.6-6 • RPM-based installer Ansible Automation Platform (online) 2.6-6

CSV Versions in this release:

- Namespace-scoped bundle: aap-operator.v2.6.0-0.1774648945

- Cluster-scoped bundle: aap-operator.v2.6.0-0.1774648973

Overview

This asynchronous update for Red Hat Ansible Automation Platform 2.6 (2.6.20260325) provides targeted enhancements, security updates, and bug fixes across automation controller, platform gateway, automation hub, Event-Driven Ansible, Lightspeed, execution environments, platform operators, and both platform and containerized installers.

This release focuses on expanding audit coverage for administrative actions, upgrading core services to Django 5.2 LTS, addressing multiple CVEs across the stack, and improving reliability, performance, and observability. It also refines user experience in the web UI and gateway and improves diagnostics through clearer logging and traceability.

Highlights

Expanded audit and access logging

- Introduces and extends audit logging for users, teams, organizations, role assignments, dynamic preferences, and direct component access, improving traceability of administrative and configuration changes.
  AAP-67043, AAP-66919, AAP-66800, AAP-66668

Platform-wide move to Django 5.2 LTS

- Upgrades Django for gateway, hub, controller, and Lightspeed components to Django 5.2 LTS, aligning with a supported, more secure framework baseline.
  AAP-68587, AAP-68135, AAP-60155, AAP-59873, AAP-60388, AAP-64430

Security hardening through CVE remediation

- Resolves multiple vulnerabilities in UI, controller, gateway proxy, automation hub, Lightspeed, and packaging, including issues in Axios, Authlib, Pillow, pyasn1, cryptography, jsonpath, AIOHTTP, express-rate-limit, and Go's `crypto/tls` and `net/url` libraries.
  AAP-69040, AAP-68686, AAP-68683, AAP-68529, AAP-68526, AAP-67735, AAP-67503, AAP-66903, AAP-66695, AAP-66655, AAP-66636, AAP-65713, AAP-65711, AAP-65695, AAP-65507, AAP-65506, AAP-65505, AAP-65475, AAP-65474, AAP-65473, AAP-65472, AAP-65412, AAP-65411, AAP-65410, AAP-65409, AAP-65224, AAP-64902, AAP-61921

Improved stability and performance across services

- Addresses issues impacting UI responsiveness, containerized installer behavior after Django upgrades, constructed inventory and facts handling, credential validation in Event-Driven

Ansible, database restore flows in platform operators, and certificate handling in execution environments.
AAP-69005, AAP-68843, AAP-68842, AAP-68841, AAP-68135, AAP-68079, AAP-67759, AAP-67749, AAP-67579, AAP-67552, AAP-67550, AAP-67549, AAP-67548, AAP-67498, AAP-67460, AAP-67371, AAP-67230, AAP-67081, AAP-67080, AAP-67079, AAP-67078, AAP-67038, AAP-66864, AAP-66845, AAP-66806, AAP-66706, AAP-66579, AAP-66400, AAP-66106, AAP-66105, AAP-66104, AAP-66102, AAP-65109, AAP-65081, AAP-64996, AAP-64630, AAP-64146, AAP-60313, AAP-60238, AAP-58769, AAP-58535, AAP-22149

- This update rebases the containerized installer to ansible.platform collection version 2.6.20260306, aligning the installer with the current Ansible Automation Platform 2.6 collection release.
AAP-67548

Features

Controller

- This update improves compatibility with the receptor control tooling used by automation controller by updating the pinned `receptorctl` version for Tower 4.7 / Ansible Automation Platform 2.6.
AAP-66806

Enhancements

Ansible Automation Platform

- This update extends audit logging for identity lifecycle operations in the gateway by recording creation, modification, and deletion of users, teams, and organizations.
AAP-66919
- This update adds audit logging for dynamic preference changes so that updates to registered preferences and settings are tracked over time.
AAP-66800
- This update refines the login experience by removing the “show password” eye icon so that the password field remains masked during entry.
AAP-67230
- This update improves diagnostics for connectivity issues with automation controller by enhancing logging behind the “Error connecting to Controller API” banner.
AAP-64146

Containerized-installer

- This update improves compatibility of the containerized installer after the Django 5.2 upgrade, preventing controller install failures caused by changes in Django behavior and output.
AAP-68587
- This update keeps TLS configuration accurate by ensuring the gateway certificate is regenerated when certificate data changes so that gateway_main_url and related fields are updated.
AAP-66579

- This update improves observability for direct component access in containerized deployments by adding nginx log markers for controller, hub, and Event-Driven Ansible in the containerized installer.
  AAP-66106

Controller

- This update increases observability for direct API access to automation controller by adding nginx log markers for requests containing `X-Trusted-Proxy` and `X-DAB-JW-TOKEN` headers.
  AAP-66102
- This update aligns automation controller with the supported framework baseline by upgrading its Django dependency to version 5.2 LTS.
  AAP-59873

Django-ansible-base

- This update extends audit logging coverage by adding audit entries for user and team role assignment changes, improving visibility into permission updates.
  AAP-67042

Event-driven-ansible

- This update improves observability for API traffic to Event-Driven Ansible by adding nginx log markers for direct API access.
  AAP-66105

Hub

- This update improves the robustness of the automation hub container registry by setting gunicorn and proxy timeouts to better handle varied workloads and network conditions.
  AAP-67759
- This update enhances logging parity across services by adding nginx log markers for direct API access to hub so that traffic bypassing the gateway can be detected.
  AAP-66104
- This update prepares for future token management changes by adding a deprecation warning for the ah_token module in the ansible.hub collection on AAP 2.6 (Hub 4.11) behind Ansible Automation Platform gateway.
  AAP-65109
- This update modernizes automation hub by upgrading its Django dependency to version 5.2 LTS.
  AAP-60388

CVE

Ansible Automation Platform UI

- CVE-2026-29074 – SVGO denial of service via XML entity expansion in:
  - `automation-platform-ui`. AAP-68529
  - `gateway-rhel9 image`. AAP-68526
- CVE-2026-27904 – Minimatch denial of service via catastrophic backtracking in glob expressions in:

    - `automation-platform-ui`. AAP-66695
- CVE-2025-69873 – Regular expression denial of service (ReDoS) via `$data` references in:
    - `automation-platform-ui` for Ansible Automation Platform 2.6. AAP-65713
    - `gateway-rhel9 image`. AAP-65711
    - `lightspeed-rhel9`. AAP-66655
- CVE-2026-25639 – Axios denial of service `via __proto__ handling` in `mergeConfig` in:
    - `automation-platform-ui`. AAP-65475
    - `gateway-rhel9 image`. AAP-65472
    - `lightspeed-rhel9`. AAP-65473

Automation gateway

- CVE-2025-68121 – Unexpected session resumption in Go crypto/tls in:
    - `automation-gateway-proxy` for Ansible Automation Platform 2.6. AAP-65695
- CVE-2025-61726 – Memory exhaustion via query parameter parsing in Go `net/url` in:
    - `automation-gateway-proxy` for Ansible Automation Platform 2.6. AAP-64902

Lightspeed / MCP / RAG

- CVE-2026-30922 – `pyasn1` denial of service via unbounded recursion in:
    - `lightspeed-chatbot-rhel9` image for Ansible Automation Platform 2.6. AAP-69040
- CVE-2026-28498 – `Authlib` authentication bypass via forged OpenID Connect ID tokens in:
    - `lightspeed-chatbot-rhel9` image for Ansible Automation Platform 2.6. AAP-68686
- CVE-2026-28802 – `Authlib` signature verification bypass allowing unauthorized access via malicious JWTs in:
    - `lightspeed-chatbot-rhel9` image. AAP-67503
- CVE-2026-25990 – Pillow out-of-bounds write via specially crafted PSD images in:
    - `lightspeed-chatbot-rhel9`. AAP-65506
    - hub-rhel9. AAP-65505
- CVE-2026-26007 – cryptography subgroup attack due to missing subgroup validation for SECT curves in:
    - `mcp-tools-rhel9`. AAP-65412
    - `lightspeed-rhel9`. AAP-65411
    - `lightspeed-chatbot-rhel9`. AAP-65410
- CVE-2026-1615 – jsonpath arbitrary code execution via unsafe JSON Path evaluation in:
    - `lightspeed-service-container`. AAP-65224

- CVE-2025-69223 – AIOHTTP HTTP parser `auto_decompress` vulnerability exploitable with zip bombs in:
  - `lightspeed-chatbot-rhel9`. AAP-61921
- CVE-2026-30827 – `express-rate-limit denial` of service for IPv4 clients due to incorrect IPv6 subnet masking in:
  - `aap-mcp-server-rhel9`. AAP-67735

Pillow / Image processing

- CVE-2026-25990 – Out-of-bounds write via specially crafted PSD images in:
  - `hub-rhel9`. AAP-65505

Bug fixes

Ansible Automation Platform

- Fixed an issue where the "Organization Admins Can Manage Users and Teams" setting did not correctly disable the create-team button in the UI when turned off, so organization admins now see the correct state. AAP-68843
- Fixed an issue where organization administrators were still able to delete or modify teams when "Organization Admins Can Manage Users and Teams" was disabled, so this setting now enforces the intended restrictions.
  AAP-68842
- Fixed an issue where teams from other organizations were not visible to organization administrators as expected when organization-wide visibility was enabled.
  AAP-68841
- Fixed an issue where an organization administrator could not assign team access to projects in Ansible Automation Platform 2.6, preventing proper delegation of permissions.
  AAP-65081
- Fixed an issue where list views in the gateway UI loaded slowly because of excessive duplicate API requests and aggressive polling intervals, improving responsiveness.
  AAP-67460
- Fixed an issue where redirects using the next URL parameter failed when the value included a plus sign (+), whether encoded or unencoded, so redirects now work correctly.
  AAP-64996
- Fixed an issue where creating Event-Driven Ansible projects concurrently from multiple users could result in server errors when handling project creation.
  AAP-67749
- Fixed an issue where general project creation flows in Django Ansible Base could lead to errors when invoked by multiple users, improving stability.
  AAP-60238

Containerized-installer

- Fixed an issue where containerized controller installs could fail after the Django 5.2 upgrade because Django output changed and broke parsing in the installer.
  AAP-68135

- Fixed an issue where Podman's pids_limit could be set to an extremely large value on nodes with large memory, exceeding system-supported limits, by capping the value.
  AAP-67579

Controller

- Fixed an issue where facts could become inconsistent when running job templates with fact storage enabled, particularly when multiple inventories had same-name hosts or concurrent jobs updated facts.
  AAP-67371
- Fixed an issue where constructed inventories could not be saved when verbosity was greater than 2, so higher verbosity levels are now supported.
  AAP-66864
- Fixed an issue where job events missing an event type caused uncaught exceptions in the job events children summary view, improving reliability.
  AAP-64630

Event-driven-ansible

- Fixed an issue where Decision Environment credential validation rejected container registry credentials when the password came from an external credential provider unless placeholder text was used, allowing those credentials to be attached without workarounds. AAP-69005
- Fixed an issue where Jinja2 variable substitution in rule names failed in Event-Driven Ansible controller worker mode even though the same variables worked in action `extra_vars`, aligning behavior with the CLI. AAP-67038
- Fixed an issue where Event-Driven Ansible server could not sync git projects using `ssh://` or `git+ssh://` URL schemes, restoring project sync behavior. AAP-66353

Execution-environments

- Fixed an issue where a change in the `certifi` package affected default trust store paths in Ansible Automation Platform 2.6 execution environments by switching to `system-certifi` to restore expected behavior. AAP-58769

Hub

- Fixed an issue where the X-Forwarded-Proto header could be incorrectly set in conjunction with the `alter_hostname_settings` configuration on Azure when passing traffic from gateway to hub. AAP-66706

Lightspeed

- Fixed an issue where OAuth2 authentication on containerized installer deployments could fail when the Lightspeed port was set to 443 because of incorrect URL handling and default port logic.
  `AAP-66845`
- Fixed an issue where the platform configuration MCP server exposed the `settings_list` tool twice, causing API errors in clients, by renaming the tools to `controller-settings_list` and `gateway-settings_list`.
  AAP-66400

- Fixed an issue where the /check endpoint of the Ansible Lightspeed API container reported an incorrect commit version and SHA, improving diagnostics.
  AAP-60313

Platform-operator

- Fixed an issue where deleting a restored Ansible Automation Platform object did not delete the associated deployment or pods, leaving orphaned resources.
  AAP-68079
- Fixed an issue where IRSA-based S3 authentication support from galaxy-operator was not available in automation hub operator for stable-2.6, allowing S3 access-key fields to be optional.
  AAP-67498
- Fixed an issue where Galaxy operator restores with `force_drop_db` failed due to missing `CREATEDB` privileges and partitioned index handling, causing `pg_restore` to fail during restores.
  AAP-67081
- Fixed an issue where Event-Driven Ansible operator restores with `force_drop_db` failed because the managed PostgreSQL user lacked permissions to recreate databases, causing failures on restore.
  AAP-67080
- Fixed an issue where gateway operator restores with `force_drop_db` failed because required privileges were missing and partitioned indexes caused errors during `pg_restore`.
  AAP-67079
- Fixed an issue where AWX operator restores with `force_drop_db` were ignored, preventing databases from being dropped and recreated as expected.
  AAP-67078

Receptor

- Fixed an issue where receptor reported “Error locating unit” when running in controller because cancelled work units were deleted prematurely across restarts.
  AAP-22149

Known issues

Lightspeed

- This update documents that validation of Lightspeed enablement in related ATF pipelines is part of ongoing work, with pipelines verified for coverage.
  AAP-66885

Developer preview

Controller

- This update introduces a developer preview of the dispatcherd feature flag for automation controller in Ansible Automation Platform 2.6, allowing early evaluation of the new task system engine ahead of general availability.
  AAP-58535

Rebase

Platform-installer

- This update rebases the platform installer to `ansible.platform` collection version 2.6.20260306, aligning installer content with the current collection version. AAP-67549

Platform-operator

- This update rebases the platform operator to `ansible.platform` collection version 2.6.20260306, ensuring operator-managed resources use the current collection baseline. AAP-67550

Ansible Automation Platform patch release February 25, 2026

The following release notes detail the updates for the Ansible Automation Platform patch released on February 25, 2026

Components and versions

This release includes the following components and versions:

Release Date	Component versions
Component versions	• Automation controller 4.7.9 • Automation hub 4.11.6 • Event-Driven Ansible 1.2.6 • Container-based installer Ansible Automation Platform (bundle) 2.6-6 • Container-based installer Ansible Automation Platform (online) 2.6-6 • Receptor 1.6.3 • RPM-based installer Ansible Automation Platform (bundle) 2.6-5 • RPM-based installer Ansible Automation Platform (online) 2.6-5

CSV Versions in this release:

- Namespace-scoped Bundle: ap-operator.v2.6.0-0.1772585537
- Cluster-scoped Bundle: aap-operator.v2.6.0-0.1772583722

CVE

- CVE-2026-24486 `ansible-automation-platform-26/lightspeed-chatbot-rhel9`: Python-Multipart has Arbitrary File Write via Non-Default Configuration.(AAP-64188)
- CVE-2026-24486 `ansible-automation-platform-26/mcp-tools-rhel9`: Python-Multipart has Arbitrary File Write via Non-Default Configuration.(AAP-64186)
- <xref href="https://access.redhat.com/security/cve/cve-2025-13465" format="html" scope="external">CVE-2025-13465</xref> <codeph>automation-platform-ui</codeph>: prototype pollution in <codeph>_.unset</codeph> and <codeph>_.omit</codeph> functions.(AAP-64106)
- CVE-2025-13465 `ansible-automation-platform-26/lightspeed-rhel9`: prototype pollution in `_.unset` and `_.omit` functions.(AAP-64104)
- CVE-2025-13465 `ansible-automation-platform-26/gateway-rhel9`: prototype pollution in `_.unset` and `_.omit` functions.(AAP-64103)
- CVE-2026-24049 `automation-controller`: `wheel`: Privilege Escalation or Arbitrary Code Execution via malicious wheel file unpacking.(AAP-63877)
- CVE-2026-24049 `ansible-automation-platform-26/de-supported-rhel9`: `wheel` Privilege Escalation or Arbitrary Code Execution via malicious wheel file unpacking. (AAP-63861)
- CVE-2026-24049 `ansible-automation-platform-26/de-minimal-rhel9: wheel`: Privilege Escalation or Arbitrary Code Execution via malicious wheel file unpacking. (AAP-63860)
- CVE-2025-59057 `automation-platform-ui` React Router has XSS Vulnerability. (AAP-62544)
- CVE-2025-59057 <codeph> `ansible-automation-platform-26/gateway-rhel9`: React Router has XSS Vulnerability.(AAP-62543)
- CVE-2026-21884 `automation-platform-ui`: React Router SSR XSS in ScrollRestoration. (AAP-62542)
- CVE-2026-21884 `ansible-automation-platform-26/gateway-rhel9`: React Router SSR XSS in ScrollRestoration.(AAP-62541)
- CVE-2026-22029 `automation-platform-ui`: React Router vulnerable to XSS via Open Redirects.(AAP-62524)
- CVE-2026-22029 `ansible-automation-platform-26/gateway-rhel9`: React Router vulnerable to XSS via Open Redirects.(AAP-62523)

- CVE-2026-21441 `ansible-automation-platform-26/hub-web-rhel9`: urllib3 vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API). (AAP-62449)
- CVE-2026-21441 `ansible-automation-platform-26/hub-rhel9`: urllib3 vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API). (AAP-62448)
- CVE-2025-69223<codeph> `python3.11-aiohttp`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-62286)
- CVE-2025-66471 `ansible-automation-platform-26/hub-web-rhel9`: `urllib3` Streaming API improperly handles highly compressed data.(AAP-62081)
- CVE-2025-66471 `ansible-automation-platform-26/hub-rhel9-operator`: `urllib3` Streaming API improperly handles highly compressed data.(AAP-62080)
- CVE-2025-66471 `ansible-automation-platform-26/hub-rhel9`: `urllib3` Streaming API improperly handles highly compressed data.(AAP-62079)
- CVE-2025-69223 `ansible-automation-platform-26/hub-rhel9`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-61920)
- CVE-2025-69223 `ansible-automation-platform-26/ee-supported-rhel9`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-61919)
- CVE-2025-69223 `ansible-automation-platform-26/ee-minimal-rhel9`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-61918)
- CVE-2025-53643 `ansible-automation-platform-26/ee-supported-rhel9`: AIOHTTP HTTP Request/Response Smuggling.(AAP-54841)
- CVE-2026-23490 `automation-controller`: pyasn1 has a DoS vulnerability in decoder. (AAP-63123)
- CVE-2025-61140 `ansible-automation-platform-26/lightspeed-rhel9`: `jsonpath`: Prototype Pollution vulnerability in the value function.(AAP-64332)
- CVE-2026-0994 `python3.11-protobuf`: Denial of Service in Python Protobuf. (AAP-64072)

Ansible Automation Platform

Bug Fixes

- Fixed an issue where there was double logging in Gateway/DAB. Fixed unit tests. (AAP-65216)
- Fixed an issue where an organization administrator could not delegate permissions to objects within their organization.(AAP-65081)

- If the Project’s source control branch is overridden by a Template or template Schedule, it is now displayed on the schedule detail and schedule edit form review step. (AAP-60920)
- Fixed an issue preventing reordering more than 50 authentication mappings. (AAP-59119)
- Restored a bug fix so that the feature flags table is created/updated as expected even on partial migrations.(AAP-65815)

Ansible Automation Platform Operator

Enhancements

- Increased envoy request timeout from 1 second to 5 seconds.(AAP-64420)

Bug Fixes

- Fixed an issue with Automation Hub file data not restored in the correct directory. (AAP-65961)
- Fixed an issue where custom PostgreSQL settings could not be applied to the AAP Operator. Added command configuration to PostgreSQL statefulset configuration when postgres_extra_args is defined.(AAP-65487)
- Fixed an issue where there was a missing resource_requirement in the nginx container configured in the EDA event stream deployment.(AAP-64007)
- Fixed an issue where Kubernetes Secret values were being printed in operator logs. (AAP-62943)
- Fixed an issue with an extra_settings to allow customizing the LOGGING level for the Gateway Operator (AAP-62938)

Automation controller

Enhancements

- Fixed the job list endpoint to no longer load the job artifacts, resulting in better performance.(AAP-63489)
- Upgraded to Django 5.2.(AAP-59873)

Bug Fixes

- Fixed missing `RoleUserAssignment` openapi schema component.(AAP-60826)

- Fixed an issue where the AWX CLI failed to authenticate to AAP 2.5 using username/password. This resolves the Valid credentials were not provided errors when connecting to Gateway environments.(AAP-46830)

Automation hub

Features

- Added a static OpenAPI spec to galaxy that focuses the potential endpoints users can call.(AAP-66415)
- Improved documentation for Automation Hub OpenAPI specifications.(AAP-66410)

Enhancements

- Added concise descriptions to API Endpoints for AAP MCP server.(AAP-66412)

Container-based Ansible Automation Platform

Enhancements

- Increased envoy request timeout from 1 second to 5 seconds.(AAP-64323)
- Added a retry mechanism when trying to get the Automation Controller status (AAP-64291)
- Increased envoy request timeout from 1 second to 5 seconds.(AAP-64008)
- Fixed a compatibility issue when `jinja2` native is enabled on ansible-core.(AAP-62878) URL anchors in the inventory samples reflect official documentation.(AAP-55780)

Bug Fixes

- Restored a bug fix so that the feature flags table is created/updated as expected even on partial migrations.(AAP-65815)
- Fixed automation gateway preflight check which doesn't require `ansible_host` to be defined anymore.(AAP-65370)
- Fixed an issue where the installer did not make use of `ansible_user_dir` for receptor. (AAP-64452)
- Fixed an issue where disabling TLS on envoy no longer causes a controller connection error when running Merge organization task.(AAP-62904)
- Fixed an issue where the TLS verification when pushing container images to the Automation Hub registry and TLS was enabled.(AAP-62864)

RPM-based Ansible Automation Platform

Enhancements

- Increased envoy request timeout from 1 second to 5 seconds.(AAP-64008)

Event-Driven Ansible

Enhancements

- The content of the `de-minimal` and `de-supported` images of the decision environment changes. There are new names for existing plugins, and the old names are still available albeit deprecated. In most of the cases only a change of the used event source or event filter is needed.(AAP-48005)
- For example:

```
----
- name: Production ruleset
  sources:
    - ansible.eda.pg_listener:
        postgres_params:
          host: postgresql_hostname
          port: postgresql_port
          dbname: postgresql_database
        channels:
          - my_events
          - my_alerts
[...]
----


* The event source name will need to be changed as follows:


----
- name: Production ruleset
  sources:
    - eda.builtin.pg_listener:
        postgres_params:
          host: postgresql_hostname
          port: postgresql_port
          dbname: postgresql_database
        channels:
          - my_events
          - my_alerts
[...]
----
```

Bug Fixes

- Fixed an issue where the activation worker failed to reconnect to redis after disconnection.Override RQ's default heartbeat to call `register_birth`, allowing worker re-registration in case of worker disconnects from Redis, and also eliminating ghost workers. Upgraded rq version to 2.6.1.(AAP-56872)

Execution Environments

Enchancements

- ee-minimal and ee-supported have been updated to use Python 3.12. The version number has been updated to 2.0.0.(AAP-56549)

Red Hat Ansible Lightspeed

Enhancements

- Upgrade to Python 3.12.(AAP-61048)

Bug Fixes

- Fixed an issue where Lightspeed timed out connecting to chatbot in Testing CI containerized installer. Added chatbot and mcp tools ports to firewalld.(AAP-65319)
- Fixed an issue where ChatGPT 5.1 produced blank ALIA responses while using a supported model provider. Added custom config variable to be added to the llama-stack agent configuration.(AAP-63538)
- Fixed an issue where navigating away from the chatbot while a request was in progress would interrupt the process, often resulting in errors like duplicated messages. This issue has been resolved this by ensuring that outstanding requests continue processing even when the browser focus changes (AAP-62685)
- ChatGPT 5.1 produces blank ALIA responses, while using a supported model provider.

Ansible Automation Platform patch release January 21, 2026

The following release notes detail the updates for the Ansible Automation Platform patch released on January 21, 2026.

This release includes the following components and versions:

Release Date	Component versions
January 21, 2026	• automation controller 4.7.8 • Automation hub 4.11.5 • Event-Driven Ansible 1.2.4

Release Date	Component versions
	• Container-based installer Ansible Automation Platform (bundle) 2.6-5 • Container-based installer Ansible Automation Platform (online) 2.6-5 • Receptor 1.6.3 • RPM-based installer Ansible Automation Platform (bundle) 2.6-4 • RPM-based installer Ansible Automation Platform (online) 2.6-4

CSV Versions in this release:

- Namespace: aap-operator.v2.6.0-0.1768951397
- Cluster: aap-operator.v2.6.0-0.1768951413

CVE

With this update, the following CVEs have been addressed:

- CVE-2025-69223 `automation-controller` : AIOHTTP's HTTP Parser auto_decompress feature is vulnerable to zip bomb.(AAP-61927)
- CVE-2025-69223 `ansible-automation-platform-26/controller-rhel9` : AIOHTTP's HTTP Parser auto_decompress feature is vulnerable to zip bomb.(AAP-61915)
- CVE-2025-4565 `python3.11-protobuf` : Unbounded recursion in Python Protobuf. (AAP-60498)
- CVE-2025-66031 `ansible-ai-connect-service` : `node-forge` ASN.1 Unbounded Recursion.(AAP-59983)
- CVE-2025-66416 `ansible-automation-platform-26/mcp-tools-rhel9` : DNS Rebinding Protection Disabled by Default in Model Context Protocol PythonSDK.(AAP-59952)
- CVE-2025-64459 `ansible-automation-platform-26/hub-rhel9` : Django SQL injection. (AAP-58111)
- CVE-2025-53643 `automation-controller` : AIOHTTP HTTP Request/Response Smuggling.(AAP-54877)
- CVE-2025-61729 `ansible-automation-platform-26/receptor-rhel9` : Excessive resource consumption when printing error string for host certificate validation in `crypto/x509` . (AAP-61230)

- CVE-2025-64460 `python3.11-django` : `Django` : Algorithmic complexity in XML deserializer leads to denial of service.(AAP-61780)
- CVE-2025-64460 `automation-controller` : `Django` : Algorithmic complexity in XML Deserializer leads to denial of service.(AAP-60961)
- CVE-2026-21441 `ansible-automation-platform-26/lightspeed-rhel9` : `urllib3` vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API).(AAP-62341)
- CVE-2025-66471 `python3.11-urllib3` : `urllib3` streaming API improperly handles highly compressed data.(AAP-62290)
- CVE-2025-66471 `python3.11-urllib3` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62290)
- CVE-2025-66471 `automation-controller` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62090)
- CVE-2025-66471 `ansible-automation-platform-26/controller-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62068)
- CVE-2025-66471 `ansible-automation-platform-25/lightspeed-rhel8` : `urllib3` : Streaming API improperly handles highly compressed data.(AAP-62030)
- CVE-2025-66471 `ansible-automation-platform-26/mcp-tools-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62085)
- CVE-2025-66471 `ansible-automation-platform-26/lightspeed-rhel9-operator` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62084)
- CVE-2025-66471 `ansible-automation-platform-26/lightspeed-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62083)
- CVE-2025-66471 `ansible-automation-platform-26/lightspeed-chatbot-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62082)
- CVE-2025-66471 `ansible-automation-platform-26/controller-rhel9-operator` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62069)
- CVE-2025-61729 `receptor` : Excessive resource consumption when printing error string for host certificate validation in `crypto/x509` .(AAP-61235)
- CVE-2025-62706 `ansible-automation-platform-26/lightspeed-chatbot-rhel9` : `Authlib` : JWE `zip=DEF` decompression bomb enables DoS.(AAP-60596)
- CVE-2025-66471 `ansible-automation-platform-26/gateway-rhel9: `urllib3` Streaming API improperly handles highly compressed data.(AAP-62077)
- CVE-2025-66471 `ansible-automation-platform-26/eda-controller-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62072)
- CVE-2025-66471 `ansible-automation-platform-26/de-supported-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62071)
- CVE-2025-66471 `ansible-automation-platform-26/de-minimal-rhel9` : `urllib3` Streaming API improperly handles highly compressed data.(AAP-62070)

- CVE-2026-21441 `ansible-automation-platform-26/gateway-rhel9`: `urllib3` vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API). (AAP-62446)
- CVE-2026-21441 `ansible-automation-platform-26/eda-controller-rhel9`: `urllib3` vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API).(AAP-62443)
- CVE-2026-21441 `ansible-automation-platform-26/de-minimal-rhel9`: `urllib3` vulnerable to decompression-bomb safeguard bypass when following HTTP redirects (streaming API).(AAP-62383)
- CVE-2025-69223 `ansible-automation-platform-26/de-supported-rhel9`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-61917)
- CVE-2025-69223 `ansible-automation-platform-26/de-minimal-rhel9`: AIOHTTP's HTTP Parser `auto_decompress` feature is vulnerable to zip bomb.(AAP-61916)

Ansible Automation Platform

Features

- Page titles now reflect the current page content.(AAP-61754)
- Ansible Automation Platform now provides support for IPv6 single-stack and dual-stack (IPv4 and IPv6) deployments in container-based environments. IPv6 is now supported across all Ansible Automation Platform deployment methods. The `FEATURE_GATEWAY_IPV6_USAGE_ENABLED` feature flag has been removed and IPv6 support is enabled by default.(ANSTRAT-1575)
- Red Hat now collects anonymized telemetry data from the Ansible MCP server. The telemetry data includes metrics related to MCP server performance, adoption trends, and usage patterns. For more information, see the Telemetry data collection for the Ansible MCP server.(ANSTRAT-1792)

Enhancements

Improves labels and descriptions for Authenticator Mappings details.(AAP-51295)

Updated modal warning message and layout when enabling a copied Rulebook Activation. (AAP-42574)

- Added dedicated aap.auth_audit logger with specialized formatters and handlers.
  - Source IP address.
  - User agent from HTTP requests.
    - Introduced new logs for authentication events, all of which are both present in logs following their original patterns as well as logs under the `aap.auth_audit` logger, including all of the original information.

IMPORTANT:
In a future release, all authentication logs introduced and moved from their existing logger to the aap.auth_audit logger will be removed from all but the `aap.auth_audit` logger.

- Specific log changes:
  - `OAuth2` Token Lifecycle Tracking
    - Log token creation, modification, and usage with associated OAuth2 application.
    - Track authenticated API requests with user, method, path, and token details.
    - Improved activity stream with selective field diffing to reduce noise from trivial updates.
- SSO Authentication Logging
  - Log SSO redirect initiation with authenticator identification and sanitized redirect URLs.
  - Track social auth failures and exceptions.
  - Enhanced SAML authenticator logging.
- Authentication Event Logging
  - Log successful and failed authentication attempts.
  - Track cases where no authenticator can validate a user.
  - Include authenticator type in login success messages.
  - Log access denials from claims processing.
- Improved Error Logging
  - Enhanced Redis connection error messages to include underlying exceptions.
  - Better diagnostic information for troubleshooting.
    (AAP-60364)
    - Reduce cognitive complexity `in _sync_user_superuser_flag`.
      (AAP-62771)

Bug Fixes

- Fixed an issue preventing gateway from working in a pure IPv4 single stack environment when IPv6 is enabled.(AAP-60478)
- Fixed an issue where the UI did not allow for full search in resource dropdowns.
(AAP-57712)
- Fixed an issue that occasionally showed a bad request status when navigating between different pages in Ansible Automation Platform.(AAP-56701)
- Fixed an issue where searching for a collection by name returned incorrect results. Fixed the filtering by name in the **Collections** page.(AAP-56529)

- Fixed an issue where the user could not edit the **Client Certificate** and/or **Client Key** fields of a credential once set.(AAP-55296)
- Fixed an issue where the workflow job templates node credentials were missing after save for job template nodes that have a default credential that is promptable. (AAP-52638)
- Fixed an issue where the platform gateway UI reset the order of an authentication mapping when the entity was edited by the user.(AAP-52258)
- Fixed an issue where automation controller unavailability rendered the entire Ansible Automation Platform UI inaccessible. The UI now remains functional during automation controller outages, displaying a notification banner to alert users.(AAP-50106)
- Fixed an issue where the descriptions for the **Remotes** and **Remote Registries** were not accurate.(AAP-49838)
- Fixed an issue where the survey text area **Default Answer** field did not properly accept newlines when pressing Enter.(AAP-49820)
- Fixed an issue where the **Review** page of the add **Approval** node in workflow job template did not load.(AAP-49433)
- Fixed an issue where **Days of Data to Keep** is missing from **Edit Cleanup Job** schedule page.(AAP-48972)
- Fixed an issue where editing and saving credentials that use external credential lookup plugins (such as CyberArk) failed with an error message. Users can now successfully modify and save these credentials as expected.(AAP-44813)
- Fixed an issue where the SAML Service Provider extra configuration data field could not be cleared in the UI, as it would automatically reset to the default value. Users can now set this field to null, which is required for compatibility with certain identity providers. (AAP-43661)
- Fixed an issue where ad-hoc commands failed with a **Bad Request** error when using credentials configured with prompt on launch for password fields. The **Run Command** wizard now correctly displays a credential passwords step to collect required passwords before executing the command.(AAP-43603)

Deprecated

- Feature flag `FEATURE_GATEWAY_IPV6_USAGE_ENABLED` has been removed. IPv6 is now supported by default.(AAP-61805)

Automation controller

Features

- Added runtime feature flags.(AAP-62686)

Automation hub

Bug Fixes

- Fixed an issue where the password field on the automation hub Django REST framework authentication page was missing the autocomplete attribute. As a consequence, the field did not align with security best practices regarding browser autofill. With this update, the password field uses the autocomplete="new-password" attribute. As a result, the automation hub API authentication page now complies with recommended security settings.(AAP-59912)

Container-based Ansible Automation Platform

Enhancements

- Added ITLS support to lightspeed chatbot service.(AAP-60900)

Bug Fixes

- Fixed an issue where the system-prompt optimized for granite and OpenAI models. (AAP-60898)
- Fixed an issue where the containerized installer could not properly configure Redis in the IPv6 only environment. Added IPv6 support for different Ansible Automation Platform components within the containerized installer collection.(AAP-60532)
- Fixed an issue where the ansiblemcp uninstall is failing to stop containers.

Event-Driven Ansible

Features

- Added x-ai-description field to the activation PATCH method.(AAP-61969)
- With this update, activations in the "workers offline" status on the Event-Driven Ansible server are now protected from accidental deletion or disabling. This enhancement adds a warning banner to the User Interface (UI) for the "restart", "disable", and "delete" workflows, providing users with a clear warning before performing these actions. The feature also supports a force flag for the "disable" and "delete" operations, giving users the option to bypass the warning if necessary.(AAP-51378)

Bug Fixes

- Fixed an issue where the IPv6 support in the Event-Driven Ansible operator configmaps was missing the extra listen **NGINX** directive. We added the required directive so the event streams pod now has **NGINX** bound to its IPv6 interface.(AAP-62001)

Red Hat Ansible Lightspeed

Enhancements

- Enabled chatbot authentication.(AAP-61015)
- Supports Red Hat Ansible Lightspeed chatbot authentication.(AAP-59478)
- Enabled chatbot authentication.(AAP-59476)
- The MCP service / endpoint now displays a banner that explains how to connect to the service, this banner replaces an error message.(AAP-59334)
- Implemented authentication between ansible-lightspeed and ansible-lightspeed-chatbot.(AAP-58796)
- Added chatbot authentication capabilities.(AAP-58794)
- Added Lightspeed chatbot TLS support.(AAP-54412)

Bug Fixes

- Fixed an issue where there was an error on the cursor editor. When the error occurred, the MCP server configuration on the Cursor Settings panel and Output view displayed error messages.(AAP-62002)
- Fixed an issue where the Lightspeed Chatbot `PROVIDER_VECTOR_DB_ID` set to literal string caused degraded responses in production.(AAP-61118)
- Fixed bug where VS Code would throw a warning when connecting to Ansible Automation Platform MCP servers due to a period character being in the tool name. (AAP-59293)
- Fixed an issue where Ansible Lightspeed intelligent assistant showed raw tool_call output in answers.(AAP-57513)
- Fixed an issue where long tool names in the Ansible Automation Platform MCP server exceeding the 60-character limit in Cursor's MCP server resulted in the MCP server tools to being hidden in Cursor, impacting tool usage. With this release, the MCP server now supports tool names exceeding 60 characters, resolving the visibility and usability issues for all tools in the Ansible Automation Platform MCP server.(AAP-61149)
- Fixed an issue where ALIA was experiencing Database or Disk is Full errors.(AAP-53081)
- Fixed an issue where the `CHATBOT_API_KEY` environment variable was not passed to the container when the provider was azure/openai.(AAP-63292)

Deprecated

- Removed ansible-risk-insights dependency from ansible-ai-connect-service. (AAP-60336)

Ansible Automation Platform patch release January 6, 2026

The following release notes detail the updates for the Ansible Automation Platform patch released on January 6, 2026.

This release includes the following components and versions:

Release Date	Component versions
January 6, 2026	• Automation controller 4.7.6 • Automation hub 4.11.4 • Event-Driven Ansible 1.2.3 • Container-based installer Ansible Automation Platform (bundle) 2.6-4.1 • Container-based installer Ansible Automation Platform (online) 2.6-4 • Receptor 1.6.2 • RPM-based installer Ansible Automation Platform (bundle) 2.6-3.2 • RPM-based installer Ansible Automation Platform (online) 2.6-3

CSV Versions in this release:

- Namespace: aap-operator.v2.6.0-0.1767630689
- Cluster: aap-operator.v2.6.0-0.1767630627

CVE

With this update, the following CVEs have been addressed:

- CVE-2025-14025 `automation-gateway` : Read-only Personal Access Token (PAT) bypasses write restrictions.(AAP-60068)

- CVE-2025-68664 `ansible-automation-platform-26/lightspeed-rhel9: LangChain` : Arbitrary code execution via serialization injection.(AAP-61313)

Ansible Automation Platform patch release December 10, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on December 10, 2025.

This release includes the following components and versions:

Release Date	Component versions
December 10, 2025	• Automation controller 4.7.6 • Automation hub 4.11.4 • Event-Driven Ansible 1.2.3 • Container-based installer Ansible Automation Platform (bundle) 2.6-4 • Container-based installer Ansible Automation Platform (online) 2.6-4 • Receptor 1.6.2 • RPM-based installer Ansible Automation Platform (bundle) 2.6-3.1 • RPM-based installer Ansible Automation Platform (online) 2.6-3

CSV Versions in this release:

- namespace: aap-operator.v2.6.0-0.1764966733
- cluster: aap-operator.v2.6.0-0.1764966767

General

Stakeholder Notification: Ansible Lightspeed Release 2.6 Sign-Off Issue

We are writing to inform you of an issue discovered during the final sign-off process for the Ansible Lightspeed 2.6 release that impacts a subset of customers.

Issue Description

Ansible Lightspeed customers who deploy using the Red Hat OpenShift Container Platform (OCP) installer will encounter a specific error when attempting to use the Chatbot feature to retrieve information about Ansible Automation Platform. This manifests as a **403 Forbidden error**.

A similar issue also affects customers who utilize the **containerized installer** with the **Enterprise topology**. Crucially, this issue **does not affect** customers using the containerized installer with the **Growth or All-in-One topology**.

Root Cause and Impact

The 403 error is caused by a Cross-Site Request Forgery (CSRF) protection mechanism not correctly recognizing the origin of the Chatbot's request in the affected deployment configurations. This prevents the Chatbot from functioning as intended for these specific customer segments.

Workaround

We have identified a viable workaround that can be implemented by affected customers immediately:

Customers must edit their `AnsibleAutomationPlatform` kind value in the configuration to include a new `extra_setting` variable and wait for redeployment.

Setting Name	Value	Action Required
`CSRF_TRUSTED_ORIGINS`	`<Ansible Automation Platform Gateway FQDN>`	Modify the `AnsibleAutomationPlatform` kind value and then wait for the changes to become effective.

Example:

```
kind: AnsibleAutomationPlatform
  spec:
    lightspeed:
      extra_settings:
        - setting: CSRF_TRUSTED_ORIGINS
          value: 'https://gateway-onprem-signoff-26.apps.aap-test2.w6n5.p1.openshiftapps.com'
```

NOTE:
The `<Ansible Automation Platform Gateway FQDN>` should be replaced with the customer's actual Fully Qualified Domain Name for the platform gateway.

Next Steps

Our engineering team is prioritizing a permanent fix for this issue, which will be included in a forthcoming patch release. We will provide updates on the timeline for the fix shortly.

Please disseminate this information to your relevant customer-facing and support teams so they can proactively assist customers encountering this issue with the provided workaround.

Thank you for your understanding and continued support.

CVE

With this update, the following CVEs have been addressed:

- CVE-2025-64459 `ansible-automation-platform-26/gateway-rhel9` : Django SQL injection.(AAP-58110)
- CVE-2025-64459 `automation-controller` : Django SQL injection.(AAP-58117)
- CVE-2025-64459 `ansible-automation-platform-26/controller-rhel9` : Django SQL injection.(AAP-58104)
- CVE-2025-62727 `ansible-automation-platform-26/mcp-tools-rhel9` : Starlette DoS via Range header merging.(AAP-57017)
- CVE-2025-62727 `ansible-automation-platform-26/lightspeed-chatbot-rhel9` : Starlette DoS via Range header merging.(AAP-57011)
- CVE-2025-64459 `ansible-automation-platform-26/lightspeed-rhel9` : Django SQL injection.(AAP-58112)
- CVE-2025-64459 `python3.11-django` : Django SQL injection.

Ansible Automation Platform

Features

- Ansible Automation Platform now provides support for IPv6 single-stack and dual-stack (IPv4 and IPv6) deployments in Red Hat OpenShift Container Platform, and RPM-based environments. Support for container-based environments will be introduced in a future patch release. To enable IPv6 in Ansible Automation Platform, set the `FEATURE_GATEWAY_IPV6_USAGE_ENABLED` feature flag to True. For more information about using feature flags, see How to set feature flags for Red Hat Ansible Automation Platform.(ANSTRAT-1575)

Availability to deploy and configure Ansible MCP servers

- Organization administrators can now deploy an Ansible Model Context Protocol (MCP) server on an Operator-based or containerized installation of Ansible Automation Platform 2.6. This functionality is available as a Technology Preview release.

- Model Context Protocol (MCP) is an open standard that enables AI models to use external AI tools and services via a unified interface.
- The following are the key capabilities:
  - Using the Ansible MCP server, you can connect your Ansible Automation Platform with your preferred external AI tool (such as Claude, Cursor, or ChatGPT). The AI tools can access key information about your Ansible Automation Platform environment and perform tasks.
  - Ansible users can query information, execute workflows, and perform automation tasks using natural language prompts directly within their preferred AI tool.

NOTE:
Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.

For more information about deploying the Ansible MCP server, see Deploy Ansible MCP server on Ansible Automation Platform. (ANSTRAT-1567)

Enhancements

- Fixed an issue with missing Ansible Automation Platform 2.6 repositories for Red Hat Enterprise Linux 10, which previously prevented the successful build of devtools RPMs. This resulted in devtools failing to mirror Ansible Automation Platform 2.6 on Red Hat Enterprise Linux 10. With this release, we have built the devtools RPMs for Red Hat Enterprise Linux 10 on a dedicated channel which are now accessible to users. (AAP-53866)

Bug Fixes

- Fixed an issue where the python build dependencies wheel files were stored in container images.(AAP-59254)
- Fixed an issue where the job template did not remain editable after the associated project was deleted.(AAP-58467)
- Fixed a server error that could happen when assigning permissions via the `/api/eda/` or `/api/controller/` endpoints.(AAP-58622)
- Fixed an issue where the job template did not remain editable after the associated project was deleted.(AAP-58467)
- Fixed an issue where the project status update link on the job details page was broken. (AAP-57215)
- Fixed an issue where the brand logo was missing in the **About** page when accessing it from the **Overview** page.(AAP-57133)

- Fixed an issue where the **Resource Type** filter in the **Roles** page did not correctly filter by resource types like Credential, Project, and Execution Environment, that are found in both Automation Execution and Automation Decisions.(AAP-56691)
- Fixed an issue where the **Launched by** field appeared blank in the UI when the project update is triggered automatically, such as through **Update revision on launch** or other automated conditions.(AAP-56643)
- Fixed an issue where the playbook select dropdown did not automatically select a playbook if there was only one in the project.(AAP-56279)
- Fixed an issue where the source control **Branch** option was missing from the Inventory source.(AAP-56149)
- Fixed an issue where the Azure AD name was inconsistently called **Azuread** in the user interface. This has been corrected to **Azure AD**.(AAP-55677)
- Fixed an issue where the OpenAPI specification for the platform gateway REST API was not providing comprehensive documentation and detailed request/response schemas. (AAP-53643)
- Fixed an issue where the edit form for a survey would not display in the UI if the survey was created without a default value using the ansible.controller collection.(AAP-51548)
- Fixed an issue where the text on **Remotes** and **Remote Registries** pages was not accurate.(AAP-49838)
- Fixed an issue where the source control branch for the **Project Sync Job Details** was missing.(AAP-49450)
- Fixed an issue where the collection hyperlink was broken in card view in private automation hub.(AAP-49006)
- Fixed an issue where the **Search** function failed to narrow results when adding host to group.(AAP-47510)
- Fixed an issue where the custom login text had poor legibility and did not allow for HTML markup such as links.(AAP-47462)
- Fixed an issue where the filtering by host name did not work as expected in the **Add Existing Host** dialog.(AAP-45534)
- Fixed an issue where the email notification URL for the workflow job template displayed a blank page.(AAP-43796)
- Fixed an issue where creating a new template from **Project** or **Inventory** did not auto-populate the Project field.(AAP-41725)
- Fixed an issue where the **Permission Denied** message on the templates tab, when the user has permission, was misleading. Updated the messaging when job template creation is not available: Job template creation requires project access and the user is not currently assigned to any projects.(AAP-40800)
- Fixed an issue where the repository URL in the **Details** page was incorrect.(AAP-40160)

- Fixed a survey validation issue where the minimum length value of a question could be set to greater than the maximum length value.(AAP-39932)
- Fixed a survey validation issue where text was being treated as a number when evaluating its length.(AAP-39931)
- Fixed an issue where the user was unable to create a schedule for **Constructed inventory** synchronization. The **Create Schedule** UI no longer presents the option to select a constructed inventory when the resource type is Inventory source.(AAP-38660)
- Fixed an issue where the survey answers were not being saved when editing or creating a schedule.(AAP-37923)
- Fixed an issue where the instance groups on a schedule could not be edited. (AAP-37872)
- Fixed an issue where there was a **404** error message on a validated repo sync on private automation hub.
    - Introduces an **Options** section for the checkboxes **Signed collections only** and **Sync all dependencies**.
    - Adds an info message about syncing dependencies outside the repository. (AAP-36592)
- Fixed an issue where there was an inconsistency in the task timestamps between the **Overview** and **Detail** views.(AAP-36588)
- Fixed an issue where an unchecked SSL verification caused `ImagePullBackOff` errors. This caused failed job launches due to SSL certificate verification issues. With this release, SSL certificate verification is bypassed for **Container Registry** type credentials.(AAP-33889)
- Fixed an issue where users were encountering an issue with `extra-vars` `number_list` containing more than 21 digits in non-quoted integer format, experiencing a UI display problem. Previously, the user interface incorrectly converted long numbers to scientific notation, making input difficult.(AAP-31805)

Deprecated

- The following endpoints have been deprecated for Ansible Automation Platform, MCP, and MVP in the OpenAPI Specifications;
    - `UserViewSet` and `DeprecatedRelatedUserViewSet` are deprecated.
    - `UserTeamViewSet` and `UserOrganizationViewSet` are deprecated.
    - `authenticators` and `authenticator_uid` fields are deprecated in `UserSerializer`.(AAP-58322)

Ansible Automation Platform Operator

Enhancements

- Event-Driven Ansible event-stream mTLS configuration has been added to the installer. (AAP-58343)
- Added `spec_overrides` field to the restore CR spec:
  - Added support for overriding Controller-specific settings via `spec_overrides.controller`.
  - Added support for overriding automation hub specific settings via `spec_overrides.hub`.
  - Added support for overriding Event-Driven Ansible specific settings via `spec_overrides.eda`.
  - Added support for overriding database-specific settings via `spec_overrides.database`.(AAP-60024)

Bug Fixes

- Fixed an issue with object storage secrets that were not included in the Automation Hub backup.(AAP-59610)
- Fixed the conditional failure for `AnsibleWorkflow` job launch when using the `AnsibleWorkflow` CR in Ansible Automation Platform 2.6.(AAP-59106)
- Fixed an issue where there was an OpenShift Container Platform resource runner python library dependency missing from the container image.(AAP-59032)
- Fixed a server error that could happen when assigning permissions via the `/api/eda/` or `/api/controller/` endpoints.(AAP-58622)

Automation controller

Features

- Receptor collection version bumped to 2.0.8, which is compatible with Red Hat Enterprise Linux 10.(AAP-58421)
- Added `x-ai-description` to controller schema to provide AI friendly description of each endpoint.(AAP-59819)

Bug Fixes

- Fixed an issue where project update failed with no output, and project deletions failed. Automation controller now uses the force flag when syncing a project which has **Allow branch override** enabled.(AAP-58533)
- In this update, users attempting to install a software package on an unsupported architecture may encounter issues due to incorrect data in reminders. This has been resolved.(AAP-59728)
- Fixed an issue where the project update failed with no output and project deletions also failed.(AAP-58533)

- Fixed an issue where the OpenAPI specification for the Automation controller was incomplete, impeding MCP server integration development. This limited the seamless MCP server integration with the Ansible Automation Platform. The Automation controller's REST API is now complete and accessible.(AAP-53640)

Automation hub

Bug Fixes

- Fixed an issue with Automation hub authentication failure for users with the **Team Admin** role:
  - Users assigned the **Team Admin** role can now successfully authenticate to Automation hub. Previously, these users would receive a **401** error when accessing Automation hub API endpoints due to an incompatibility between the **Team Admin** role and Automation hub's internal permission system.(AAP-58898)
- Fixed an issue where the password field on the Automation hub Django REST framework authentication page was missing the autocomplete attribute. As a consequence, the field did not align with security best practices regarding browser autofill. With this update, the password field uses the `autocomplete="new-password"` attribute. As a result, the Automation hub API authentication page complies with recommended security settings. (AAP-59910)
- Previously, upgrades from Ansible Automation Platform 2.5 to 2.6 failed when API access logging was enabled. This occurred due to an incorrect import path in the galaxy-ng package. This release corrects the import path.(AAP-59886)

Container-based Ansible Automation Platform

Enhancements

- Configured podman PID limits, `sysctls` for `inotify` and kernel keys, and ulimit `nofile` for user running Ansible Automation Platform service containers based on system resources.(AAP-59438)

Bug Fixes

- Fixed an issue where after uninstall/re-install of receptor jobs were unable to start due to stale exited containers with the same name were still present.(AAP-59609)

Event-Driven Ansible

Enhancements

- Added concise descriptions to API endpoints for Ansible Automation Platform MCP MVP endpoints (`x-ai-description`).(AAP-58431)

Bug Fixes

- Fixed an issue where the OpenAPI specification for Event-Driven Ansible was not offering comprehensive documentation and detailed request/response schemas. Previously, developers integrating with Event-Driven Ansible via the MCP server had to manually explore APIs and format API calls without proper guidance, which impeded seamless integration. With this release, the OpenAPI specification for the Event-Driven Ansible REST API is now complete and well-documented. This enhancement enables seamless integration with the MCP server using Event-Driven Ansible.(AAP-53642)

Lightspeed

Technical Preview

- This developer preview introduces support for Ansible MCP Servers.(AAP-57303)
- This new feature enables users to access the Ansible Automation Platform 2.6 API directly from AI tools like Claude Code.(AAP-57217)

Features

Added the new Ansible Automation Platform MCP Server to the 2.6 stream.(AAP-58863)

Bug Fixes

- Fixed an issue of Ansible Automation Platform Multi-Channel Platform (MCP) servers crashing due to incomplete Epic and System Design Plan (SDP) creation, leading to unclear work requirements. As a result, the Ansible Automation Platform MCP Servers have created the System Design Plan and related tasks, addressing ANSTRAT-1567. This enhancement improves the efficiency of feature development for end users by completing the Ansible Automation Platform MCP Servers system design plan in an active manner.(AAP-53087)

Receptor

Features

- Receptor collection version bumped to 2.0.8, which supports Red Hat Enterprise Linux 10 mesh nodes.(AAP-57987)

Bug Fixes

- Fixed an issue where the receptor-collection was not up to date with Automation hub standards. There is now an up to date Changelog included in receptor-collection. (AAP-58434)

Ansible Automation Platform patch release November 19, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on November 19, 2025.

This release includes the following components and versions:

Release Date	Component versions
November 19, 2025	• Automation controller 4.7.5 • Automation hub 4.11.3 • Event-Driven Ansible 1.2.2 • Container-based installer Ansible Automation Platform (bundle) 2.6-3 • Container-based installer Ansible Automation Platform (online) 2.6-3 • Receptor 1.6.2 • RPM-based installer Ansible Automation Platform (bundle) 2.6-3 • RPM-based installer Ansible Automation Platform (online) 2.6-3

CSV Versions in this release:

- Namespace: aap-operator.v2.6.0-0.1763137334
- Cluster: aap-operator.v2.6.0-0.1763137355

CVE

With this update, the following CVEs have been addressed:

- CVE-2025-9909 `automation-gateway` : improper path validation in gateway allows credential exfiltration.(AAP-53584)
- CVE-2025-59530 `receptor` : `quic-go` crash due to premature `HANDSHAKE_DONE` frame. (AAP-55973)

Ansible Automation Platform

Features

- Allows for Event-Driven Ansible to add CA Certificates in gateway which can then be used by **Envoy** to do certificate based authorization for mTLS `EventStreams`. (AAP-56770)

Enhancements

- Red Hat Ansible Lightspeed section has been removed from the left navigation bar. (AAP-53006)
- Added fallback-authenticator feature, which allows users to configure `fallback_authentication` for running custom logic in the event local authentication fails.
  - Set all existing local authenticators and those created on initial install to fallback to controller credentials.
  - The ability to clear the preset if the user does not want to fallback to controller authorization anymore.(AAP-56919)
- Ansible Lightspeed intelligent assistant has expanded its support for third-party Large Language Model (LLM) providers, and now includes OpenAI and Microsoft Azure. Third-party LLM support is available for both OpenShift Container Platform operator installation and containerized installation.
  - For more information, see Deploying the Ansible Lightspeed intelligent assistant on Red Hat OpenShift Container Platform and Deploying the Ansible Lightspeed intelligent assistant on containerized installation.(ANSTRAT-1673)

Bug Fixes

- Fixed a significant performance regression in response time for GET requests to */ role_definitions/* and related endpoints.(AAP-56868)
- Fixed an issue where users who existed in Ansible Automation Platform 2.5 with controller legacy authentication, but never logged in were unable to attempt authentication with controller in Ansible Automation Platform 2.6, and were left in an unusable state.(AAP-56388)
- Fixed issue in which superuser status would sync from platform gateway to other components if set to `True`, but not if set to `False`, where administrator privileges were not removed from the other components in all cases.(AAP-56296)

- Fixed an issue where platform auditors were not able to view all platform level settings. (AAP-55608)
- Fixed an issues where the **Team** input field on the authentication mapping form was not hidden when an organization role was selected.(AAP-55602)
- Fixed an issue where the workflow visualizer CSS was displaying the incorrect height. (AAP-55164)
- Fixed an issue using the and condition with multiple attributes. Previously the authentication map would skip the missing attributes, now, the map will be applied only if all attributes are present and the condition(s) are met.(AAP-53612)
- Fixed an issue where the `LOGIN_REDIRECT_OVERRIDE` did not allow for a bypass URL. A login page has been added at */login* to bypass the `LOGIN_REDIRECT_OVERRIDE` setting when it is misconfigured.(AAP-53471)
- Fixed the Subscription Usage chart where it did not always display at full height. (AAP-52218)
- Fixed an issue that was preventing users from viewing complete survey question choices that contained a colon.(AAP-50290)
- Fixed an issue where a warning message was not available when a user tried to restart an activation in the **workers offline** status.(AAP-24009)
- Fixed an issue where filtering platform resources by special characters did not work as expected.(AAP-52360)
- Fixed an issue where, applying a domains filter on the Jobs tab and navigating to the **Projects** section, then selecting a project with multiple templates, caused the system to display only the job template that was filtered by the domain, hiding other templates and showing a misleading message.(AAP-48031)
- Fixed an issue where there was no limit filtering to the jobs page.(AAP-45218)
- Fixed a form validation issue on the **Login redirect override** field in platform gateway settings.(AAP-40517)
- Fixed an execution environment deletion warning.(AAP-55135)

Red Hat Ansible Lightspeed

Features

- Added support for 3rd party model providers OpenAI.(AAP-58291)
- Added support for 3rd party model providers Azure.(AAP-58290)

Enhancements

- Upgraded Lightspeed Core Stack to 0.3.0.(AAP-55681)

- Added ALIA support `lightspeed-stack` 0.3.0 and `llama-stack` 0.2.22.(AAP-58136)
- Upgraded Ansible Lightspeed intelligent assistant to `Lightspeed-core` 0.3.0. (AAP-56629)
- Added ALIA support for Azure provider.(AAP-56511)
- Added ALIA support for OpenAI provider.(AAP-56509)
- Made changes required to support `llama-stack` 0.2.22.(AAP-58361)

Bug Fixes

- Fixes an issue where the Red Hat Ansible Lightspeed assistant returned raw `tool_call` JSON instead of natural language answers due to improper processing in Ansible Automation Platform 2.6 with granite-3.3-8b. This compromised user experience by exposing internal details.(AAP-57513)
- Fixed an issue where the user would be scrolled to the bottom of the chat history if they clicked **thumbs up/thumbs down** on a previous message.(AAP-58438)
- Fixed an issue where during the upgrade of `chatbot-api`, the new one is stuck in pending state waiting until PVC is removed.(AAP-57376)

Known Issues

- If you are using an IBM Granite 3.3 AI model series in your Ansible Lightspeed intelligent assistant deployment, there may be a delay of ~1 minute in receiving a chat response. As a workaround, restart the chat session.(AAP-58186)

Automation controller

Features

- Receptor collection version updated to 2.0.6, which is compatible with ansible-core 2.19. (AAP-42617)

Bug Fixes

- Fixed an issue where the migrating team mappers which did not include a users field is now supported.(AAP-56395)
- Fixed the following migration error for the migration `0200_template_name_constraint.py` when there was a job template or project with duplicate name in the same organization.(AAP-56222)

Error Message

```
django.db.utils.ProgrammingError: column main_unifiedjobtemplate.org_unique does not
exist
```

- Fixed an issue where some edge cases caused JSON to fail to parse a line from the worker stream with the error: **Expecting value: line 1 column 1 (char 0) Line with invalid JSON data: b**. Updated the pinned version for `receptorctl` in automation controller to address this issue. This effects Tower 4.7.(AAP-58412)
- Fixed an issue where some edge cases caused JSON to fail to parse a line from the worker stream with the error: **Expecting value: line 1 column 1 (char 0) Line with invalid JSON data: b**. Updated the pinned version for `receptorctl` in automation controller to address this issue. This effects Tower 4.6.(AAP-58415)
- Fixed an issue where there was not a meaningful error message whenever the streaming of logs was aborted. Update `ansble-runner` to 2.4.2 to address this issue.(AAP-58390)
- Fixes an issue where jobs failed on `fapolicyd` enabled systems where python3.9 was not installed by default. Updates `automation-controller-fapolicyd` from python3.9 to python3.11 to address this issue.(AAP-55790)

Automation hub

Bug Fixes

- Fixed an upgrade error, `AttributeError` or `ValueError`, **content type mismatch** in the migration that happens when upgrading if any role is assigned to a group globally before the migration.(AAP-58299)

Container-based Ansible Automation Platform

Enhancements

- Added ALIA support lightspeed-stack 0.3.0 and llama-stack 0.2.22.(AAP-58295)
- Added ALIA support for Azure provider.(AAP-58206)
- Added ALIA support for OpenAI provider.(AAP-58197)

Bug Fixes

- Fixed a compatibility issue with PostgreSQL 17 when using an external database and admin credentials.(AAP-57431)
- Fixed an issue with the chatbot response about the latest Ansible Automation Platform version.(AAP-57385)
- Fixed an issue with the monitoring image on Red Hat Ansible Lightspeed nodes when using the bundle deployment.(AAP-57167)

RPM-based Ansible Automation Platform

Enhancements

Event-Driven Ansible event-stream mTLS configuration added to installer.(AAP-46070)

Bug Fixes

- Fixed an issue where the installer failed during the execution environment image upload when there was no automation hub node in inventory.(AAP-56892)
- Fixed an issue with extra log content. platform gateway logs in */var/log/ansible-automation-platform/gateway* have been refactored, there is now more separation of the logs for various components:
  - *control-plane-supervisor.log* ← Messages from `supervisorctl` about the control-plane (new)
  - *control-plane.log* ← Django logs for the control-plane (new, extracted from gateway.log)
  - *gateway.log* ← Django logs for gateway (existing, had items removed)
  - *uwsgi.log* ← UWSGI logs for the {Gateeway} (new, extracted from gateeay.log)
  - *envoy.log* ← The proxy log (existing, unchanged).(AAP-30549)

Event-Driven Ansible

Features

- Enhancement to support mTLS event streams.(AAP-57375)
- Added the ca_certificates module and the enable_mtls attribute to route objects. (AAP-48345)
- Added a credential type for mTLS event stream.(AAP-46054)

Enhancements

- Event-Driven Ansible event-stream mTLS configuration added to the installer, (AAP-57434)

Receptor

Features

- Addresses edge cases that could cause JSON failure to parse a line from the worker stream. It also raises the versions of go dependencies and other minor functionality changes.(AAP-57253)

Ansible Automation Platform patch release November 5, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on November 5, 2025.

This release includes the following components and versions:

Release Date	Component versions
November 5, 2025	• Automation controller 4.7.4 • Automation hub 4.11.2 • Event-Driven Ansible 1.2.1 • Container-based installer Ansible Automation Platform (bundle) 2.6-2 • Container-based installer Ansible Automation Platform (online) 2.6-2 • Receptor 1.6.0 • RPM-based installer Ansible Automation Platform (bundle) 2.6-2 • RPM-based installer Ansible Automation Platform (online) 2.6-2

CSV Versions in this release:

- Namespace-scoped Bundle: aap-operator.v2.6.0-0.1761384532
- Cluster-scoped Bundle: aap-operator.v2.6.0-0.1761384578

Red Hat Ansible Lightspeed

Bug Fixes

- A typo in the `containerfile` caused the **nginx** configuration file to be copied to a non-existent directory in operator-based installations, leading to the Lightspeed API service being unavailable due to incorrect port configuration. With this release, the typo has been fixed, ensuring the Lightspeed API service now listens on the correct port in operator-based installations, improving API endpoint accessibility.(AAP-56712)

Technical note

Red Hat Ansible Lightspeed

RFC 2818 is now enforced between the lightspeed service and the identity provider (gateway) during the OAuth2 authorisation.

Container-based Ansible Automation Platform

Bug Fixes

- Fixed an issue with the chatbot response about the latest Ansible Automation Platform version.(AAP-57385)

Ansible Automation Platform patch release October 28, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on October 28, 2025.

This release includes the following components and versions:

Release Date	Component versions
October 28, 2025	• Automation controller 4.7.4 • Automation hub 4.11.2 • Event-Driven Ansible 1.2.1 • Container-based installer Ansible Automation Platform (bundle) 2.6-2

Release Date	Component versions
	• Container-based installer Ansible Automation Platform (online) 2.6-2 • Receptor 1.6.0 • RPM-based installer Ansible Automation Platform (bundle) 2.6-2 • RPM-based installer Ansible Automation Platform (online) 2.6-2

CSV Versions in this release:

- Namespace: aap-operator.v2.6.0-0.1762261205
- Cluster: aap-operator.v2.6.0-0.1762261209

CVE

With this update, the following CVEs have been addressed:

CVE-2025-59682 `python-django` : Potential partial directory-traversal via archive.extract(). (AAP-54755)

CVE-2025-9908 `event-driven-ansible` : Sensitive internal headers disclosure in Ansible Automation Platform Event-Driven Ansible event streams.(AAP-53582)

CVE-2025-9907 `event-driven-ansible` : Event stream test mode exposes sensitive headers in Ansible Automation Platform Event-Driven Ansible.(AAP-53580)

CVE-2025-59343 `automation-platform-ui` : tar-fs symlink validation bypass.(AAP-54392)

CVE-2025-58754 `automation-platform-ui` : Axios DoS via lack of data size check.(AAP-53718)

Ansible Automation Platform

Features

- Added a step in the subscription wizard that allows the user to configure automation analytics.(AAP-55094)
- Added two new toggle options on the subscription wizard to allow for fetching subscriptions using basic authentication.(AAP-47865)

Bug Fixes

- Fixed an issue where the `ansible-builder` and `ansible-navigator` did not use execution environment images from ansible-automation-platform-26 namespace by default.(AAP-54934)
- Fixed an issue where the settings did not display **Red Hat** consistently in the API and UI. (AAP-54276)
- Fixed an issue where the decision environment dropdown was broken. Replaced the dropdown type for decision environments in the rulebook activation form so that when there are no decision environments available, the dropdown displays No results found instead of an empty dropdown.(AAP-53844)
- Fixed an issue where creating resources with `cookie/xcrf` token failed. Aligned dependency versions between Konflux build and component repository.(AAP-53561)
- Fixed an issue where the component label for the Platform Auditor role did not display all components.(AAP-53551)
- Fixed an issue where empty strings were displayed in the extra variables field on the **Jobs > Details** page.(AAP-49448)
- Fixed an issue where the **Load More** in authentication mapping role dropdown did not work.(AAP-54049) HubName
- Fixed an issue where the user was unable to create Event-Driven Ansible or automation hub roles when creating a custom role and selecting the **Automation Decisions** project or credential types because the UI displayed only the automation controller permissions. (AAP-54756) ControllerName
- Fixed an issue where the PatternFly 6 Upgrade broke the Ansible Automation Platform topology layout and fullscreen mode.(AAP-51106)
- Fixed an issue where some fields were missing `autocomplete = new-password` setting. (AAP-55783)
- Fixed an issue where the user was unable to select the default execution environment in the automation settings page.(AAP-39321)
- Fixed an issue where the LDAP Group Type parameters failed to save user preferences when the language was initially set to `es_ES`, resulting in a wrong version displayed on the user interface due to an uninitialized translation object.(AAP-56356)
- Fixed an issue that prevented SAML and AzureAD authentication when local user accounts share the same email address.(AAP-56518)

Deprecated

- Subscription credentials can no longer be viewed/edited from the system settings page. (AAP-55014)

Ansible Automation Platform Operator

Bug Fixes

- Fixed an issue where the Ansible Lightspeed API version did not work during Ansible Automation Platform idle.(AAP-54174)
- Fixed an issue that caused a failure to gather the job data from the controller API. (AAP-55632)
- Fixed a bug where the user could set an image without the respective version, causing the installation to enter an error loop.(AAP-55642)
- Fixed an issue where the backup and restore Ansible Automation Platform instance failed, from cluster A to cluster B, when restoring an upgraded AAP environment from 2.4.(AAP-55648)

Red Hat Ansible Lightspeed

Features

Ability to deploy Red Hat Ansible Lightspeed on new containerized installations of Ansible Automation Platform 2.6

You can deploy and use Red Hat Ansible Lightspeed when you install or upgrade to a containerized installation of Ansible Automation Platform 2.6.

Red Hat Ansible Lightspeed includes two main components that enhance your automation experience with generative artificial intelligence (AI):

Ansible Lightspeed intelligent assistant, which is an AI-powered, intuitive chat interface embedded within the Ansible Automation Platform.

The integration of Red Hat Ansible Lightspeed intelligent assistant with the Model Context Protocol (MCP) server is available as a Technology Preview release. This integration enhances the user experience by delivering relevant, dynamically sourced data results to your queries.

NOTE:
Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.

Ansible Lightspeed coding assistant, which is a generative AI service that works with IBM watsonx Code Assistant to help developers create and maintain Ansible content more efficiently.

For more information, see Deploy Red Hat Ansible Lightspeed on containerized Ansible Automation Platform.

Enhancements

- Added `postgres_extra_settings` to Ansible Automation Platform operators to apply PostgreSQL configuration file level changes to managed postgres.(AAP-55053)

Automation controller

Enhancements

- Added support for Red Hat username and password for the subscription management API.(AAP-54975)

Bug Fixes

- Fixes the `system_administrator` role creation race condition which most commonly happened on new Openshift deployments resulting in the default instance group not being created.(AAP-54963)
- Fixed an issue where the Controller container file was missing the metrics utility in version 2.6.(AAP-54948)
- Fixed an issue where the `awx.awx.license` appeared to succeed when given an invalid *pool/subscription*.(AAP-54768)
- Fixed an issue where the `ansible.platform` collection did not work with the default Red Hat Ansible Automation Platform credential type.(AAP-41000)
- Fixed an issue where there was a duplicate value (`subsystem_metrics_pipe_execute_seconds`) detected under *api/controller/v2/metrics/* on Ansible Automation Platform 2.5.(AAP-55621)
- Fixed an issue where the platform auditor did not have access to controller settings. (AAP-55607)

Automation hub

Enhancements

- Fixed an **HTTP 500** error when getting */api/galaxy/_ui/v2/users/3/*.(AAP-54260)

Bug Fixes

- Fixed an HTTP 500 error when getting /api/galaxy/_ui/v2/users/3/.(AAP-54260)

Container-based Ansible Automation Platform

Enhancements

- Implemented preflight ansible-core version validation.(AAP-54932)

Bug Fixes

- Fixed an issue where `REDHAT_CANDLEPIN_VERIFY` was not being used for the correct CA permissions so that the controller could not make requests to **subscription.rhsm.redhat.com**.(AAP-55180)

RPM-based Ansible Automation Platform

Bug Fixes

- Fixed an issue where setting `automationgateway_disable_https=false` resulted in install failure.(AAP-55466)
- Fixed an issue where `RESOURCE_KEY SECRET_KEY` was not updated when restoring from a different environment.(AAP-54942)
- Fixed an issue where Event-Driven Ansible DE credentials failed to populate on initial installation.(AAP-54519)

Fixed an issue where the `envoy.log` for automation gateway did not receive logs after it was rotated.(AAP-51779)

Fixed an issue where `REDHAT_CANDLEPIN_VERIFY` was not being used for the correct CA permissions so that the controller could not make requests to **subscription.rhsm.redhat.com**. (AAP-55183)

Event-Driven Ansible

Features

- Changes in the deployment and nginx configuration now allow for gunicorn and daphne to bind to `::` as well, essentially allowing for seamlessly binding to IPv4 and IPv6 (dual-stack) addresses, while also enabling the operator to run in single-stack IPv6 or IPv4 scenarios.(AAP-56192)

Receptor

Bug Fixes

Fixed an issue where there was stability issue on long-running jobs, clusters under heavy load, and network flakiness.(AAP-53742)

Ansible Automation Platform patch release October 16, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on October 16, 2025.

This release includes the following components and versions:

Release Date	Component versions
October 16, 2025	• Automation controller 4.7.2 • Automation hub 4.11.1 • Event-Driven Ansible 1.2.0 • Container-based installer Ansible Automation Platform (bundle) 2.6-1.1 • Container-based installer Ansible Automation Platform (online) 2.6-1 • Receptor 1.5.7 • RPM-based installer Ansible Automation Platform (bundle) 2.6-1.1 • RPM-based installer Ansible Automation Platform (online) 2.6-1

CSV Versions in this release:

- Namespace-scoped Bundle: aap-operator.v2.6.0-0.1760139263
- Cluster-scoped Bundle: aap-operator.v2.6.0-0.1760139657

Ansible Automation Platform

Bug Fixes

- Fixed an issue where the claims processing failed to migrate services during the post-migrate upgrade process.(AAP-55631)

Automation controller

Bug Fixes

- Fixed an issue where the Ansible Automation Platform upgrade would be marked as failed if a single authenticator failed to migrate.(AAP-55629)

Automation hub

Bug Fixes

- Fixed a global galaxy team role migration issue that could occur during the post-migrate upgrade process.(AAP-55304)
- Fixed an issue caused by a constraint violation during migrations.(AAP-55309)
- Fixed an issue from `aap-gateway-manage, migrate_service_data`, that states **Role definition content type must be null to assign globally**, which was due to permissions in hub that failed validation.(AAP-55639)

Ansible Automation Platform patch release October 6, 2025

The following release notes detail the updates for the Ansible Automation Platform patch released on October 6, 2025.

This release includes the following components and versions:

Release Date	Component versions
October 6, 2025	• Automation controller 4.7.1

Release Date	Component versions
	• Automation hub 4.11.0 • Event-Driven Ansible 1.2.0 • Container-based installer Ansible Automation Platform (bundle) 2.6-1 • Container-based installer Ansible Automation Platform (online) 2.6-1 • Receptor 1.5.7 • RPM-based installer Ansible Automation Platform (bundle) 2.6-1 • RPM-based installer Ansible Automation Platform (online) 2.6-1

CSV Versions in this release:

- Namespace-scoped Bundle: aap-operator.v2.6.0-0.1759764484
- Cluster-scoped Bundle: aap-operator.v2.6.0-0.1759764962

Automation hub

- Fixed an issue where the automation hub collections in 2.6 could not be pulled with Ansible Galaxy due to incorrect dynamic http logic. This issue only affects the Red Hat Ansible Automation Platform Operator installation.(AAP-55099)

Technology Preview

Technology Preview features in Red Hat Ansible Automation Platform provide early access to experiment with new tools, provide feedback, and prepare your teams. Evaluate how new features fit into your automation strategy and plan implementations ahead of general availability.

Using Technology Preview features helps you to:

- **Test new capabilities early:** Experiment with upcoming features in non-production environments to understand their potential value and fit for your use case.
- **Provide feedback:** Share your experience with Red Hat to help shape the final implementation of features before they reach general availability.
- **Plan ahead:** Identify integration points, training needs, and process changes before features become production-ready.

How Ansible Automation Platform supports Technology Preview

Technology Preview features are clearly marked in the documentation so that you can easily distinguish them from generally available features.

Key characteristics of Technology Preview features:

- Clearly marked with "Technology Preview" labels in the documentation.
- Subject to change based on customer feedback and product development needs.
- Not recommended for production use.

NOTE: Hat does not support Technology Preview features with production service level agreements (SLAs) and they might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features so you can test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features - Scope of Support.

Ansible Lightspeed intelligent assistant with MCP servers

Integrate Ansible Automation Platform with an MCP server to safely use your existing AI tools to manage and run automation.

You can deploy an Ansible Model Context Protocol (MCP) server on an operator-based installation or container-based installation of Ansible Automation Platform. This functionality is available as a Technology Preview release.

NOTE: Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Related information
Deploy Ansible MCP server on Ansible Automation Platform

ansible-core 2.19

The ansible-core 2.19 release includes an overhaul of the templating system and a new feature labeled Data Tagging.

NOTE: Ansible Automation Platform does not include ansible-core 2.19 by default, but it is compatible with 2.19. See related links for more information.

Changes in ansible-core enable reporting of numerous problematic behaviors that went undetected in previous releases, with wide-ranging positive effects on security, performance, and user experience. Backward compatibility has been preserved where practical, but some breaking changes were necessary. This section describes some common problem scenarios with example content, error messages, and suggested solutions.We recommend you test your playbooks and roles in a staging environment with this release to determine where you may need to make changes.

Related information
Red Hat Ansible Automation Platform Life Cycle
Technical preview description
Porting guide

Access preconfigured development tools with Ansible development workspaces

Ansible development workspaces provide a fully supported browser-based development environment that includes Ansible development tools for creating and testing Ansible playbooks, roles, and collections. The workspaces run as containers within Red Hat OpenShift Dev Spaces.

Introduction to Ansible development workspaces

Ansible development workspaces offer centralized management and policy enforcement, giving administrators better control and governance over secure, consistent automation development environments. Developers benefit by avoiding local application installs, especially in locked-down settings.

Ansible development tools and runtimes are pre-configured in Ansible development workspaces. Developers can start creating projects for automation content quickly by logging in to Ansible development workspaces in a browser.

The development tools in Ansible development workspaces are based on Ansible recommended practices, which improves the quality and reliability of your automation content. As a component of your Red Hat Ansible Automation Platform subscription, Ansible development workspaces are fully supported.

To ensure that your automation content files persist when you quit Ansible development workspaces, you push your projects to a git repository in a source control manager (SCM) that is linked to your workspace.

Ansible development workspaces components

Each Ansible development workspace is a project-agnostic full development environment. Dependencies are satisfied for all the tools in the environment.

The following applications are pre-installed.

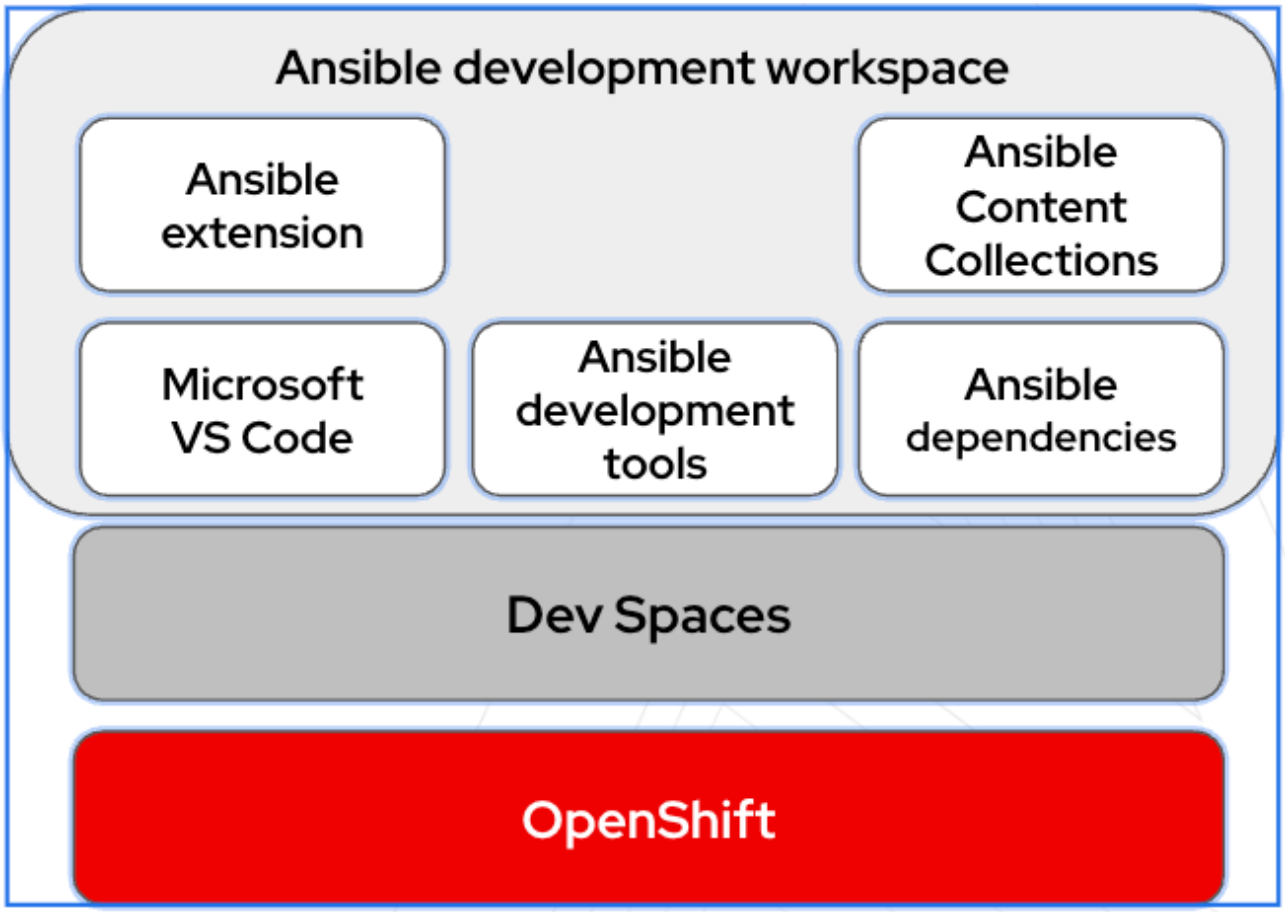
- Microsoft VS Code
- Python
- `ansible-core`
- Ansible development tools (ADT) package, which includes:
  - `ansible-creator` for scaffolding directory structure for your automation content
  - `ansible-lint` for identifying stylistic errors and anti-patterns
  - `molecule` for running functional tests on your automation content

About the Ansible dev spaces image

Red Hat OpenShift Dev Spaces is a containerized cloud development environment (CDE) that provides pre-configured, consistent workspaces running on OpenShift Container Platform. It provisions ready-to-code workspaces on demand.

The Ansible dev spaces image is the container image for Ansible development workspaces. It replaces an existing Ansible demo within OpenShift Dev Spaces and is fully supported by Red Hat.

The following diagram illustrates the relationship between OpenShift Container Platform, OpenShift Dev Spaces, and Ansible development workspaces.



Related information
Ansible Automation Platform devspaces image page

Set up and install Red Hat OpenShift dev spaces to run your Ansible container

An administrator must install Red Hat OpenShift Dev Spaces to create a OpenShift Dev Spaces dashboard from where developers can launch Ansible development workspaces.

Prerequisites

Review the prerequisites listed here before beginning the installation of Red Hat OpenShift Dev Spaces. Meeting these requirements helps ensure a successful setup.

- You have access to a web-browser and network connectivity.
- You have installed Red Hat OpenShift Container Platform.
- You have an active Red Hat OpenShift cluster.

- You have a valid subscription to Red Hat Ansible Automation Platform.
- You have set up a version control system such as Git.

Install Red Hat OpenShift dev spaces

An administrator must install Red Hat OpenShift Dev Spaces to generate an OpenShift Dev Spaces dashboard. The dashboard is the entry point for developers to launch Ansible development workspaces.

Procedure

1. Follow the steps in Installing Dev Spaces on OpenShift using the web console in the *Red Hat OpenShift Dev Spaces Administration guide* to install OpenShift Dev Spaces.

   This process includes the following steps:

   a. Log in to your OpenShift cluster as an administrator.
   b. Install the OpenShift Dev Spaces operator from the OperatorHub.
   c. Create an instance of the OpenShift Dev Spaces operator.
2. Share the URL for the OpenShift Dev Spaces dashboard with the users who need to launch Ansible development workspaces.

Create and launch an Ansible development workspace

An administrator installs Red Hat OpenShift Dev Spaces. After installation, developers can use the provided OpenShift Dev Spaces dashboard to create Ansible development workspaces that include a web-based version of VS Code.

Authentication

Ansible dev spaces must be able to authenticate with your Git source control manager (SCM).

- If your organization has integrated Git source control OAuth authentication with Ansible dev spaces, you do not need to configure authentication between OpenShift Dev Spaces and your Git SCM.
- If your organization has not set up OAuth authentication, you must generate personal access tokens for authentication between OpenShift Dev Spaces and your Git SCM.

Configure Git personal access token authentication

You must create a personal access token (PAT) in your Git source control manager (SCM), and add it to OpenShift Dev Spaces to enable access to your repositories from your Ansible development workspace.

Procedure

1. Create a personal access token in your Git SCM and save it.
    - See Managing your personal access tokens in the GitHub documentation.
    - See Personal access tokens in the Gitlab documentation.
2. In a browser, navigate to the OpenShift Dev Spaces dashboard provided by your administrator, and log in.
3. Expand the dropdown menu under your login name and select **User Preferences**.
4. Select **Personal Access Tokens**.
5. Click **+Add Token**.
6. Complete the **Add Personal Access Token** form:
    - **Token Name**: Enter a name for your token
    - **Token**: Enter your personal access token for your Git repository.
7. Click **Add** to save the personal access token.

Create a Git repository for an Ansible development workspace

To launch an Ansible development workspace, you must provide a link to a Git repository that defines the development environment. The repository also stores the automation content you create in Ansible dev spaces.

Procedure

1. If your administrator provides an example repository for your team, fork the repository to create your own copy.
2. If you do not have access to an example repository, you must create your own repository.
    a. Create a directory for your new repository and use `git init` to initialize it as a Git repository.
    b. Add a `devfile.yaml` file to the repository to define the Ansible dev spaces image that you want to use for your Ansible development workspace. See Creating a devfile for Ansible development workspaces.
    c. Add a `.code-workspace` file to the repository to specify the VS Code extensions for your Ansible development workspace. See Creating a `.code-workspace` file for Ansible development workspaces.

Create a devfile for an Ansible development workspace

To ensure your Ansible development workspace launches with the correct Ansible dev spaces image, you must add a `devfile` to your git repository. A `devfile` is a YAML file that defines the development environment for a project in Red Hat OpenShift Dev Spaces.

Procedure

1. In your Git repository for your Ansible development workspace, create a new file named `devfile.yaml`.
2. Copy and paste the following sample code into the `devfile.yaml` file:

```
---
# cspell: disable=devspaces
schemaVersion: 2.2.2
metadata:
  name: ansible-devspaces-devfile
components:
  - name: tooling-container
    container:
      image: registry.redhat.io/ansible-automation-platform-tech-preview/
ansible-devspaces-rhel9:latest
      memoryRequest: 256M
      memoryLimit: 6Gi
      cpuRequest: 250m
      cpuLimit: 2000m
      args: ["tail", "-f", "/dev/null"]
      env:
        - name: "ANSIBLE_COLLECTIONS_PATH"
          value: "~/.ansible/collections:/usr/share/ansible/collections"
        - name: KUBEDOCK_ENABLED
          value: "true"
...
```

3. Modify the image value to the name of your specific Ansible image.
4. Add the `devfile.yaml` file to your Git repository and push the changes to your source control manager (SCM).

Create a code-workspace file for an Ansible development workspace

To configure VS Code extensions that are included in your Ansible development workspace, you must add a `.code-workspace` JSON file to your git repository.

Procedure

1. In your Git repository for your Ansible development workspace, create a new file named `.code-workspace`.
2. Copy and paste the following sample code into the `.code-workspace` file:

```
{
    "settings": {
        "ansible.lightspeed.suggestions.enabled": true,
        "ansible.lightspeed.enabled": true
    },
    "extensions": {
        "recommendations": [
                        "redhat.ansible",
                        "redhat.vscode-yaml",
                        "redhat.vscode-openshift-connector",
                        "eamodio.gitlens",
        ]
    },
}
```

3. If you want to add extra extensions to your Ansible development workspace, add them in the `extensions.recommendations` section of the file.
4. Add the `.code-workspace` file to your Git repository and push the changes to your source control manager (SCM).

Launch an Ansible dev spaces workspace

Launch your Ansible development workspace by providing the URL for your prepared Git repository in the OpenShift Dev Spaces dashboard. This opens your VS Code environment in a browser.

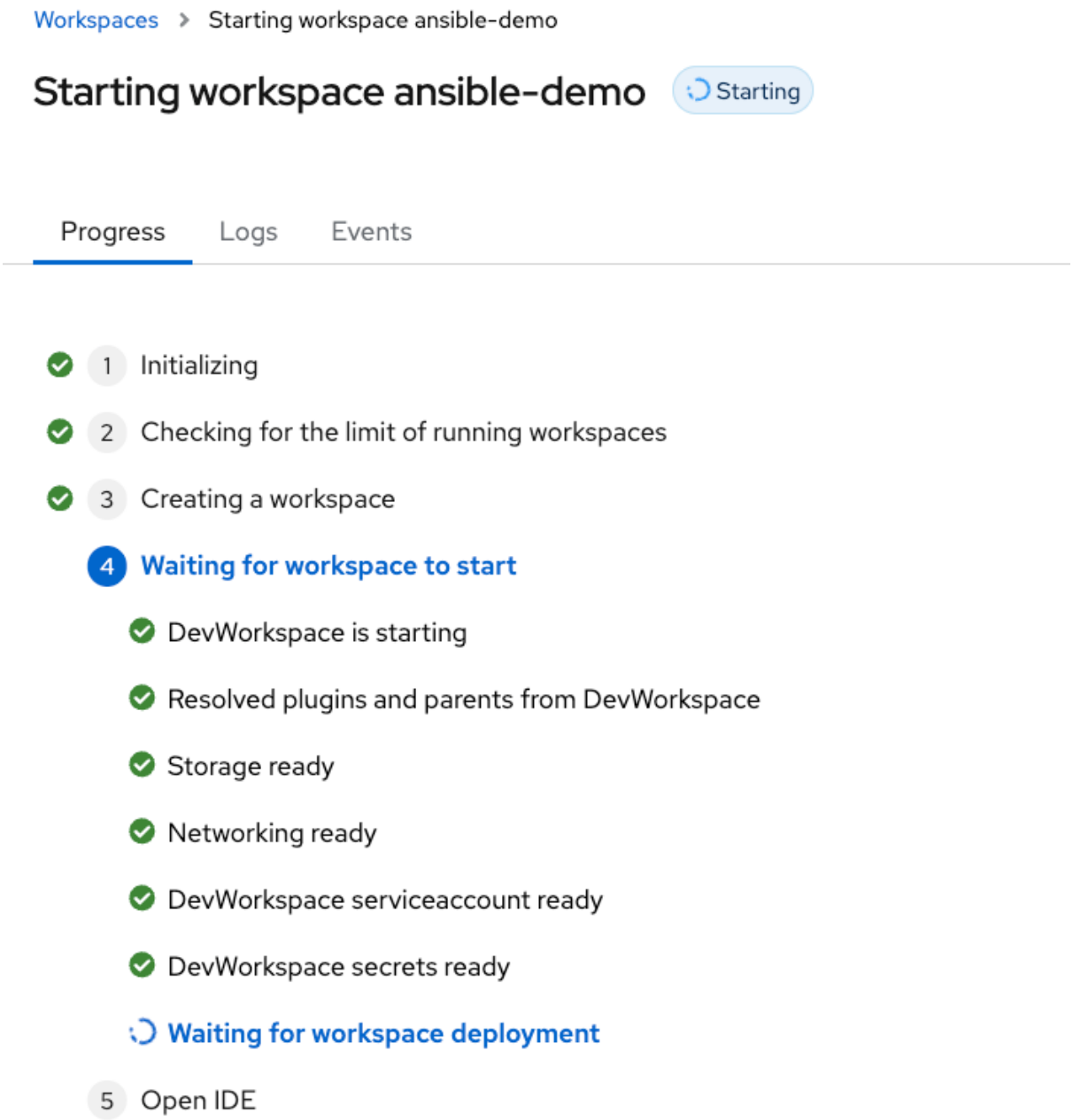
Before you begin

- Your administrator has provided a URL for a OpenShift Dev Spaces dashboard.

- You have prepared a git repository that contains the `devfile.yaml` and `.code-workspace` files that define the Ansible development workspace configuration.

Procedure

1. In a browser, navigate to the OpenShift Dev Spaces dashboard and log in.
2. Select **Create Workspace** in the navigation pane.
3. In the **Import from Git** field of the **Create Workspace** form, enter the URL for the Git repository that contains your `devfile.yaml` and `.code-workspace` files.
4. Click **Create & Open**.
5. OpenShift Dev Spaces displays the progress for the provisioning process of your Ansible development workspace.



After the Ansible development workspace launches, a VS Code environment opens in your browser.

6. To open a terminal for executing commands and viewing `ansible-lint` suggestions in VS Code, click the main menu icon in the **Activity** bar and select **Terminal › New Terminal**.

   For more information about working in a VS Code terminal, see Getting started with the terminal in the VS Code documentation.

Develop automation content in your workspace

The Ansible development tools are installed as part of the Ansible extension in the Ansible development workspace. You can use Ansible development tools to scaffold directories for automation content in your repository.

Using the Ansible extension ensures that best practices for directory structure are met.

Red Hat recommends that you create only one collection per repository, so that each collection has a clear, specific purpose. This approach promotes reusability, as each collection is a self-contained unit of content. A one-to-one relationship between a collection and its repository also improves manageability by simplifying dependency management, maintenance, and release cycles.

Related information
Developing automation content

Create collections and playbooks in your Ansible development workspace

Use the Ansible extension in VS Code to use Ansible development tools to scaffold directories and files for your automation content. You can use Red Hat Ansible Lightspeed with IBM watsonx Code Assistant to help you write playbooks, and `ansible-lint` to debug them.

Procedure

1. In the OpenShift Dev Spaces dashboard. select the Ansible development workspace where you want to develop automation content.
2. In the **Activity** bar of VS Code, select the Ansible icon to open Ansible development tools.
3. Select **Connect** in the Ansible Lightspeed section to log in to Ansible Lightspeed.
4. Select an option in the **initialize** section of **Ansible Development tools** to scaffold files and directories for a collection project or a playbook project.

   For more information on creating projects, see:

   - *Scaffold a playbook project* in Auto-generate the structure and files for your automation project
   - *Scaffold a collection for your roles* in Package and distribute automation content with collections

5. Select options in the **Add** section of Ansible development tools to add files for playbooks or roles to your project. Alternatively, you can use the options in the **Ansible Lightspeed** section to generate playbooks or roles.
6. Save your work:
   a. Click the main menu icon in the **Activity** bar and select **Terminal > New Terminal**.

b. Use `git add` and `git commit` commands to stage the changed files and commit your changes to the local repository in the workspace.

c. Use the `git push` command to push your updates to your repository in your source control manager.

Edit and debug automation content in your Ansible development workspace

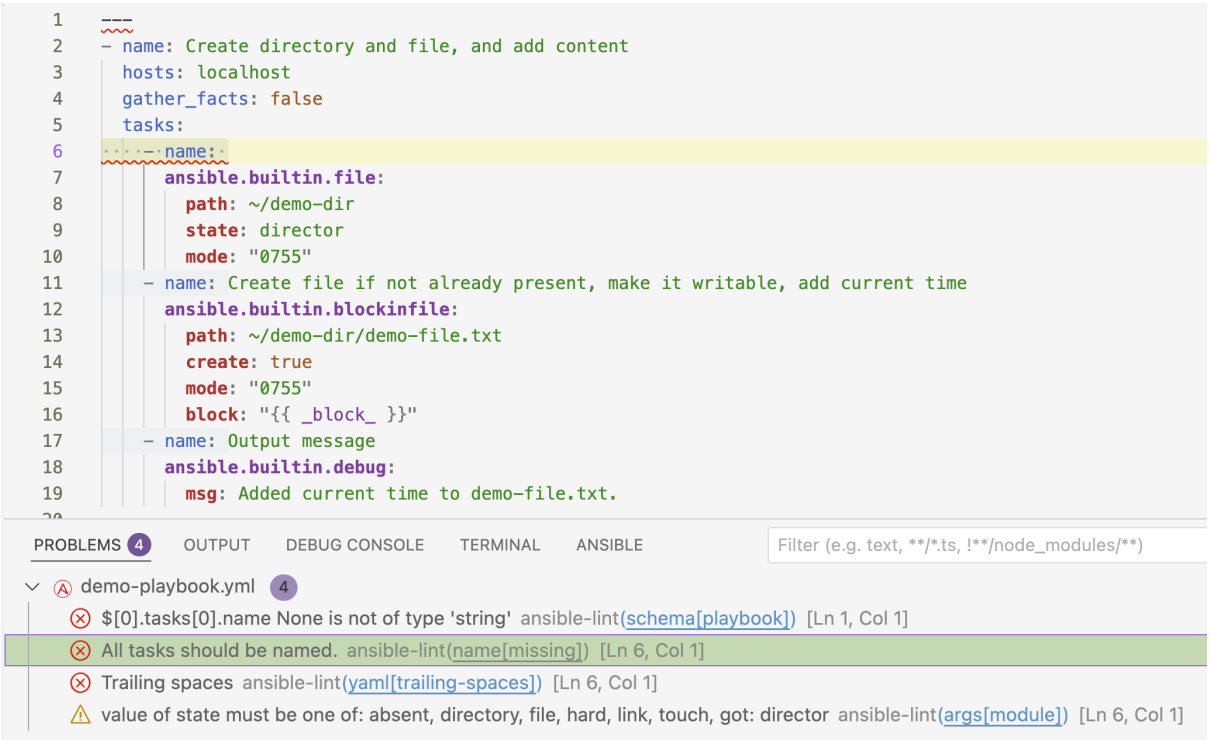
You can continue to work in an existing workspace. Workspaces that are inactive might be paused due to an administrator-set timeout to free up resources. However, they will automatically relaunch when you select them from the OpenShift Dev Spaces dashboard.

The administrator in OpenShift Dev Spaces configures the duration of this inactivity timeout. Additionally, Ansible lint will identify errors within your playbooks.

Procedure

1. To display your previously created workspaces, select **Workspaces** in your OpenShift Dev Spaces dashboard.
2. Select **Open** next to the workspace that you want to use.
3. Select the **Explorer** icon in the **Activity** bar to open the file explorer, and open the file you want to edit.
4. While you are editing, the Ansible extension provides suggestions. Select a suggestion from the dropdown list to include it in your playbook.
5. To view documentation for a keyword or a module, hover your mouse over it.
6. Open the terminal in VS Code: click the main menu icon in the **Activity** bar and select **Terminal › New Terminal**.
7. Select the **Problems** tab in the terminal to view issues that `ansible-lint` has identified.

   In the following example, one error is selected in the **Problems** tab, and the corresponding line in the playbook is highlighted.



8. When you have resolved the problems in your playbook, a message is displayed in the **Problems** tab of the terminal.

Execute playbooks in your Ansible development workspace

Execute your playbooks efficiently using the integrated Ansible extension within the VS Code environment of your workspace.

- To execute a playbook in your Ansible development workspace, right-click on a playbook name in the file explorer and select **Run ansible playbook via > Run ansible playbook via ansible playbook**.

> **NOTE:**
> You cannot use execution environments in Ansible development workspaces. Do not use `ansible-navigator` to execute playbooks.

Share your work

Share your automation content and collaborate with colleagues by working together from a single shared Git repository and submitting pull requests.

Procedure

1. To contribute to a colleague's project, request the URL for the Git repository that corresponds to your colleague's Ansible development workspace.
2. Launch a workspace using the repository URL that your colleague shared.

3. Work within a new git branch and contribute to your colleague's repository by creating a merge or pull request.

Delete an Ansible development workspace

Manage your Ansible development workspaces by understanding how to delete them from OpenShift Dev Spaces when they are no longer needed.

Deleting unused workspaces helps free up cluster resources and maintains a clean development environment within OpenShift Dev Spaces.

Related information
Red Hat OpenShift Dev Spaces Authentication guide

Delete an Ansible development workspace

To delete the contents of an Ansible development workspace, you delete the workspace itself. This action removes all the pods, storage, and other resources associated with that specific workspace, effectively wiping its contents.

Before you begin

- You know the name of the workspace you want to delete.

Procedure

1. Stop the Ansible development workspace that you want to delete.
    - To stop the workspace in the Dev Spaces dashboard, select the workspace that you want to delete and select **actions › Stop Workspace**.
    - To stop the workspace using OpenShift `oc` commands, follow the steps in Stopping workspaces in the Red Hat OpenShift Dev Spaces *User Guide*.
2. Delete the workspace:
    - To delete the workspace from the Dev Spaces dashboard, select the workspace that you want to delete and select **actions › Delete Workspace**.
    - To delete a workspace using OpenShift `oc` commands, follow the steps in Removing workspaces in the Red Hat OpenShift Dev Spaces *User Guide*.

Uninstall OpenShift Dev Spaces

Uninstall OpenShift Dev Spaces completely when it is no longer required. Remember that this action removes all related Ansible dev spaces user data.

- To uninstall OpenShift Dev Spaces, follow the steps in the Uninstalling Dev Spaces chapter of the Red Hat OpenShift Dev Spaces *Administration Guide*.

> **NOTE:**
> Uninstalling Ansible dev spaces removes all Ansible dev spaces-related user data.

Manage edge devices by integrating with Red Hat Edge Manager

Red Hat Edge Manager simplifies the management of edge devices and applications through a declarative approach. It automatically implements and maintains configurations across your entire fleet by defining a desired state for OS versions, host configurations, and application deployments.

> **IMPORTANT:**
> The Red Hat Edge Manager documentation has moved to a standalone location. For the most current documentation, see Red Hat Edge Manager 1.0. The content in this guide is deprecated.

The Red Hat Edge Manager on Ansible Automation Platform offers elevated integration with your automations. You can then focus more on orchestrating the environment without worrying about updating the operating system.

Related information
Red Hat Edge Manager 1.0

Architecture

You can manage individual devices or an entire fleet by using the Red Hat Edge Manager. The Red Hat Edge Manager uses an agent-based architecture that allows for a scalable and robust device management, even with limited network conditions.

By deploying a Red Hat Edge Manager agent to a device, the agent autonomously manages and monitors the device while periodically communicating with the Red Hat Edge Manager service to check for new configurations and to report device status.

The Red Hat Edge Manager supports image-based operating systems. You can include the Red Hat Edge Manager agent and the agent configuration in the image that is distributed to the devices.

Image-based operating systems allow the agent to start a transactional update of the image and to roll back to the earlier version in case of an update error.

The Red Hat Edge Manager architecture has the following main features:

- Agent
- Service
- Image-based operating system
- API server
- Database
- Device
- Device fleet

Related information
Red Hat Edge Manager agent and service
Red Hat Edge Manager API server

Agent and service

The Red Hat Edge Manager agent is a process running on each managed device that periodically communicates with the Red Hat Edge Manager service. The agent is responsible for the following tasks:

- Enrolling devices into the service
- Periodically checking with the service for changes in the device specification, such as changes to the operating system, configuration, and applications
- Applying any updates independently from the service
- Reporting status of the device and the applications

The Red Hat Edge Manager service is responsible for the following tasks:

- Authenticating and authorizing users and agents
- Enrolling devices
- Managing device inventory
- Reporting status from individual devices or fleets

The service also communicates with a database that stores the device inventory and the target device configuration. When communicating with the service, the agent polls the service for

changes in the configuration. If the agent detects that the current configuration deviates from the target configuration, the agent attempts to apply the changes to the device.

When the agent receives a new target configuration from the service, the agent does the following tasks:

1. To avoid depending on network connectivity during the update, the agent downloads all required resources, such as the operating system image and application container images, over the network to disk.
2. The agent updates the operating system image by delegating to `bootc`.
3. The agent updates configuration files on the file system of the device by overlaying a set of files that the service sends to the device.
4. If necessary, the agent reboots into the new operating system. Otherwise, the agent signals system services and applications to reload the updated configuration.
5. The agent updates applications running on Podman.

If the update fails or the system does not return online after rebooting, the agent automatically rolls back to the earlier operating system image and configuration.

NOTE:
You can keep fleet definitions in Git. The Red Hat Edge Manager periodically syncs with the fleet definitions in the database.

API server

The API server is a core part of the Red Hat Edge Manager service that gives users and agents an option to communicate with the service.

The API server exposes the following endpoints:

User-facing API endpoint

Users can connect to the user-facing API endpoint from the CLI or the web console. Users must authenticate on the platform gateway to obtain a JSON Web Token (JWT) to make HTTPS requests.

Agent-facing API endpoint

Agents connect to the agent-facing endpoint, which is mTLS-protected. The service authenticates devices by using the X.509 client certificates.

The Red Hat Edge Manager service also communicates with various external systems to authenticate and authorize users, get mTLS certificates signed, or query configuration for managed devices.

Enroll devices

To manage your devices with the Red Hat Edge Manager, you must enroll the devices to the Red Hat Edge Manager service.

The first time the Red Hat Edge Manager agent runs on a device, the agent prepares for the enrollment process by generating a cryptographic key pair. The cryptographic key pair serves as the unique cryptographic identity of the device. The key pair consists of a public and a private key. The private key never leaves the device, so that the device cannot be duplicated or impersonated.

When the device is not yet enrolled, the agent performs service discovery to find its Red Hat Edge Manager service instance. Then, the device establishes a secure, mTLS-protected network connection to the service. The device uses its X.509 enrollment certificate that the device acquired during image building or device provisioning. The device submits an enrollment request to the service that includes the following:

- a description of the device hardware and operating system
- an X.509 Certificate Signing Request which includes the cryptographic identity of the device to obtain the initial management certificate

The device is not considered trusted and remains quarantined in a device lobby until an authorized user approves or denies the request.

For more information, see the following sections:

Related information
Optional: Request an enrollment certificate for early binding

Enrollment methods

You can provision the enrollment endpoint and certificate to the device in the following ways:

Early binding

You can build an operating system image that includes the enrollment endpoint and certificate. Devices that use an early binding image can automatically connect to the defined service to request enrollment, without depending on any provisioning infrastructure. The devices share the same long-lived X.509 client certificate. However, in this case, the devices are bound to a specific service and owner.

Late binding

You can define the enrollment endpoint and certificate at provisioning time instead of including them in the operating system image. Devices that use a late binding image are not bound to a single owner or service and can have device-specific, short-lived X.509 client certificates. However, late binding requires virtualization or bare-metal provisioning infrastructure that can request device-specific enrollment endpoints and certificates from the

Red Hat Edge Manager service and inject them into the provisioned system by using mechanisms such as *cloud-init*, *Ignition*, or *kickstart*.

NOTE:
The enrollment certificate is only used to secure the network connection for submitting an enrollment request. The enrollment certificate is not involved in the actual verification or approval of the enrollment request. The enrollment certificate is no longer used with enrolled devices, as the devices rely on device-specific management certificates instead.

Related information
cloud-init
Ignition
kickstart

Install Red Hat Edge Manager

Install the Red Hat Edge Manager to manage edge devices and applications at scale. This guide focuses on a standalone deployment of the Red Hat Edge Manager on Red Hat Enterprise Linux alongside Ansible Automation Platform.

Install the Red Hat Edge Manager RPM package

Prepare your Red Hat Enterprise Linux host for the installation of the Red Hat Edge Manager by enabling the necessary repositories, installing the `flightctl-services` package, configuring the baseDomain, and then starting and verifying the running services.

Before you begin

- An active Ansible Automation Platform subscription with a running instance and the necessary API URLs and OAuth credentials.
- A separate machine from Ansible Automation Platform to install the Red Hat Edge Manager on.
- Podman installed for managing containers.
- A Red Hat Enterprise Linux host with:
  - Minimal installation
  - 4 cores and 16GB RAM (recommended)
  - Administrative access (root or sudo-capable user)
  - SSH access

Procedure

1. SSH into your Red Hat Enterprise Linux host.
2. Authenticate and log in to the Red Hat Container Registry:

```
sudo podman login registry.redhat.io
```

3. Install the necessary repositories and packages:

   - Ensure that the Ansible Automation Platform repositories are enabled by running the following example command based on the version of Red Hat Enterprise Linux and architecture of your host:

```
sudo subscription-manager repos --enable ansible-automation-
platform-2.5-for-rhel-9-x86_64-rpms
```

   - Install the Red Hat Edge Manager service by running:

```
sudo dnf install -y flightctl-services
```

4. Update the installed `/etc/flightctl/service-config.yaml` to set the `baseDomain`:

```
sudo vi /etc/flightctl/service-config.yaml
```

IMPORTANT:
Ensure that you set the `baseDomain` in the service configuration correctly. By default, the installation process attempts to automatically set this value based on the IP address of your Red Hat Enterprise Linux host.

However, if your environment uses a specific domain name to access this host, for example `rhem-example.com`, it is recommended that you manually update the `baseDomain` in `/etc/flightctl/service-config.yaml` to this hostname.

Setting the `baseDomain` correctly ensures that all generated URLs, certificates, and internal configurations within the Red Hat Edge Manager are accurate for your network setup. This is especially important for integration with Ansible Automation Platform and for ensuring that the UI is accessible through the intended domain name.

You can check the currently configured `baseDomain` using:

```
grep baseDomain: /etc/flightctl/service-config.yaml
```

5. Enable and start the services:

```
sudo systemctl enable flightctl.target
sudo systemctl start flightctl.target
```

6. Verify that services are running:

```
sudo systemctl list-units flightctl-*.service
```

   You should see these 7 services running:

   - flightctl-db
   - flightctl-kv
   - flightctl-api
   - flightctl-periodic
   - flightctl-worker
   - flightctl-ui
   - flightctl-cli-artifacts

7. Go to the UI at the `baseDomain` stored in the service configuration file:

```
grep baseDomain: /etc/flightctl/service-config.yaml
```

   Visit the displayed `baseDomain` in your web browser to access the UI.

If your services do not run correctly, use the following log command to troubleshoot further and remediate:

```
journalctl -u flightctl-<impacted service> -b --no-pager
```

Set up the OAuth application for Ansible Automation Platform

You have two options for setting up the OAuth application in Ansible Automation Platform, either manually or automatically in the Ansible Automation Platform UI.

Set up the OAuth application automatically

Automatic setup of an OAuth application by generating an OAuth token within Ansible Automation Platform and adding it to your configuration file. Upon service startup, the application is automatically created, and the client ID updated.

Procedure

1. Generate an OAuth token in Ansible Automation Platform:
    a. From the navigation panel, select **Access Management > Users**.
    b. Select a user with write permissions to the **Default** organization (admin user recommended).
    c. Click the **Tokens** tab for that user.
    d. Click **Create token** and enter the relevant details.
        i. **Scope**: Select **Write**.
2. Go to the Integrate with Ansible Automation Platform section for the steps to edit your `service-config.yaml` file and complete setting up the OAuth application automatically.

Set up the OAuth application manually

Manually set up an OAuth application within your Ansible Automation Platform instance. This is important for enabling token-based authentication and integrating external applications such as the Red Hat Edge Manager.

Procedure

1. From the navigation panel on your Ansible Automation Platform instance, go to **Access Management > OAuth Applications**.
2. Click **Create OAuth application**.
3. Enter the following details:
    - **Name**: Enter a name such as "Red Hat Edge Manager". This is the name visible in the Ansible Automation Platform UI.
    - **URL**: The `baseDomain` of your Red Hat Edge Manager UI with `https://`.
    - **Organization**: Select **Default**.
    - **Authorization grant type**: Select **Authorization code**.
    - **Client**: Select **Public**.
    - **Redirect URIs**:
        - The redirect configured for your UI is your `baseDomain` with a /callback route appended, such as `https://your-edge-manager-ip-or-domain:443/callback`.

If you have more than one URI, enter them in this field separated by a space, not commas or other delimiters.

- To provide a redirect for CLI usage (`flightctl login`), configure a redirect URI, such as `http://127.0.0.1/callback` .

4. Click **Create OAuth application**. An **Application Links** section is now visible in the navigation panel.
5. Copy the **Client ID** as you need it to update the **oAuthApplicationClientId** in your `service-config.yaml` file with this value.
6. Go to the Integrating with Ansible Automation Platform section for the steps to edit your `service-config.yaml` file and complete setting up the OAuth application manually.

Related information
Configuring access to external applications with token-based authentication

Integrate with Ansible Automation Platform

Integrate the Red Hat Edge Manager with your Ansible Automation Platform instance by modifying the `service-config.yaml` file to include authentication type, API URLs, OAuth client ID, and an optional OAuth token, followed by restarting the services.

Procedure

1. Stop the flightctl services before editing your `service-config.yaml` file:

```
sudo systemctl stop flightctl.target
```

2. Configure the integration settings by editing the configuration file:

```
sudo vi /etc/flightctl/service-config.yaml
```

3. Update the configuration file to integrate with Ansible Automation Platform:

```
global:
  baseDomain: <your-edge-manager-ip-or-domain>
  auth:
    type: aap
    insecureSkipTlsVerify: false
    aap:
      apiUrl: https://your-aap-instance.example.com
      externalApiUrl: https://your-aap-instance.example.com
      oAuthApplicationClientId: <client-id-from-oauth-app>
      oAuthToken: <your-oauth-token>
```

baseDomain

The domain name or IP for the host. This is the only mandatory field.

type

Set this to `aap` to enable Ansible Automation Platform authentication.

insecureSkipTlsVerify

Set to `false`. Only set this to `true` to skip TLS certificate verification for the Ansible Automation Platform URLs. For production environments, consider configuring a CA certificate (see the Self-signed certificates section).

apiUrl

The internal facing API URL for the running Ansible Automation Platform instance that makes requests against.

externalApiUrl

The externally accessible URL of your running Ansible Automation Platform instance.

oAuthApplicationClientId

This is the Client ID of the OAuth application configured in Ansible Automation Platform for the Red Hat Edge Manager. This is not necessary if you are using the automatic method.

oAuthToken

This is an OAuth token with write permissions for the "Default" organization. This is only needed if you want the setup process to automatically create the OAuth application. This is not necessary if you are using the manual method.

4. Start the services:

```
sudo systemctl start flightctl.target
```

Self-signed certificates

The Red Hat Edge Manager services automatically generate and store self-signed certificates in the `/etc/flightctl/pki` directory. These include:

- `/etc/flightctl/pki/ca.crt`
- `/etc/flightctl/pki/ca.key`
- `/etc/flightctl/pki/client-enrollment.crt`
- `/etc/flightctl/pki/client-enrollment.key`
- `/etc/flightctl/pki/server.crt`
- `/etc/flightctl/pki/server.key`

You can use your own custom certificates by placing them in the following locations:

- Custom Server Certificate/Key Pair:
  - `/etc/flightctl/pki/server.crt`
  - `/etc/flightctl/pki/server.key`
- Custom CA Certificate for Ansible Automation Platform authentication:
  - `/etc/flightctl/pki/auth/ca.crt`

NOTE:
Ensure that you adjust the `insecureSkipTlsVerify` setting in the `service-config.yaml` if you use a custom CA certificate for your Ansible Automation Platform instance.

Understand bootable container images

Image-based operating systems allow the operating system and its configuration and applications to be versioned, deployed, and updated as a single unit. Using an image-based operating system reduces operational risks by doing the following:

- Minimizing potential drift between what is tested and what is deployed to a large number of devices.
- Minimizing the risk of failed updates that require expensive maintenance or replacement through transactional updates and rollbacks.

The Red Hat Edge Manager focuses on image-based Linux operating systems that run bootable container images (`bootc`).

IMPORTANT:
The `bootc` tool does not update package-based operating systems.

Related information
bootc

The image building process

Create the bootable container (bootc) images you need to provision Edge Manager devices.

Before you begin

- Podman
- Skopeo
- bootc-image-builder

Procedure

1. Choose a base `bootc` operating system image, such as a Fedora, CentOS, or RHEL image.
2. Create a container file that layers the following items onto the base `bootc` image:
    - The Red Hat Edge Manager agent and configuration.
    - Optional: Any drivers specific to your target deployment environment.
    - Optional: Host configuration, for example, certificate authority bundles, and application workloads that are common to all deployments.
3. Build, publish, and sign a `bootc` operating system image using `podman` and `skopeo`.
4. Create an operating system disk image by using `bootc-image-builder`.
5. Build, publish, and sign an operating system disk image using `skopeo`.

NOTE:
The operating system disk image has partitions, volumes, the file system, and the initial `bootc` image. You only need to create the operating system disk image once, during provisioning. For later device updates, you only need the `bootc` operating system image, which has the files in the file system.

Related information
Building a operating system image for the Red Hat Edge Manager
Special considerations for building images

Special considerations for building images

Follow these key considerations when building Red Hat Edge Manager operating system images to keep consistency and stability across your device fleet. These guidelines specify how and where to implement configuration details during the image creation process.

- Build-time configuration over dynamic runtime configuration
- Configuration in the `/usr` directory
- Drop-in directories
- Operating system images with scripts

Build-time configuration over dynamic runtime configuration

Apply configuration directly to the operating system image at build time to ensure your configurations are tested, distributed, and updated together. If build-time configuration is not feasible, you can use Red Hat Edge Manager to dynamically configure devices at runtime instead.

Dynamic runtime configuration is preferable in the following cases:

- You have a configuration that is deployment or site-specific, such as a hostname or a site-specific network credential.
- You have secrets that are not secure to distribute with the image.
- You have application workloads that need to be added, updated, or deleted without reboot or they are on a faster cadence than the operating system.

Configuration in the `/usr` directory

Place configuration files in the `/usr` directory if the configuration is static and the application or service supports that configuration. By placing the configuration in the `/usr` directory, the configuration remains read-only and fully defined by the image.

Do not place the configuration in the `/usr` directory in the following cases:

- The configuration is deployment or site-specific.
- The application or service only supports reading configuration from the `/etc` directory.
- The configuration might need to be changed at runtime.

Drop-in directories

Use drop-in directories to add, replace, or remove configuration files that the service aggregates. Do not directly edit your configuration files because it can cause deviations from the target configuration.

NOTE:
You can identify drop-in directories by the `.d/` at the end of the directory name. For example, `/etc/containers/certs.d`, `/etc/cron.d`, and `/etc/NetworkManager/conf.d`.

Operating system images with scripts

Avoid executing scripts or commands that change the file system. The `bootc` or the Red Hat Edge Manager can overwrite the changed files that can cause a deviation or failed integrity checks.

Instead, run such scripts or commands during image building so changes are part of the image. You can also use the configuration management mechanisms of the Red Hat Edge Manager.

Related information
Generic guidance for building images

Build a *bootc* operating system image for Red Hat Edge Manager

With `bootc`, your operating system becomes a container image that lets your device be managed by the Red Hat Edge Manager.

To prepare your device to be managed by the Red Hat Edge Manager, build a `bootc` operating system image that has the Red Hat Edge Manager agent. Then build an operating system disk image for your devices.

Prerequisites

See the following prerequisites for building a `bootc` operating system image:

- Install `podman` version 5.0 or later and `skopeo` version 1.14 or later.
- Install `bootc-image-builder`.

Related information

Getting container tools
Installing bootc-image-builder

Install the Red Hat Edge Manager CLI

To install the Red Hat Edge Manager CLI, complete the following steps:

Procedure

1. Enable the subscription manager for the repository appropriate for your system by running the following command:

```
sudo subscription-manager repos --enable ansible-automation-platform-2.5-for-rhel-9-x86_64-rpms
```

2. Install the `flightctl` CLI with your package manager by running the following command:

```
sudo dnf install flightctl
```

If you set up the OAuth application manually, you also need to make sure that one utility `xdg-open`, `x-www-browser`, or `www-browser` is available, for example, by installing `xdg-utils`.

Related information
Configuring container pull secrets when building Red Hat Edge Manager operating system images for different target platforms

Log in to the Red Hat Edge Manager through the CLI

How you log in the Red Hat Edge Manager depends on whether you choose the automatic or manual method when you initially set up the application.

- If you use the automatic setup you can create a personal access token, even only with Read scope (under the profile icon in the top right corner of your Ansible Automation Platform UI > **User details** > **Tokens** tab) and then use this token to log in directly through the CLI, with the following example syntax:

```
flightctl login https://<your-edge-manager-ip-or-domain>:3443 --token=<your-aap-oauth-token> --insecure-skip-tls-verify
```

- If you use the manual setup, use the **Client ID** to log in through a web-based process, with the following example syntax:

```
flightctl login https://<your-edge-manager-ip-or-domain>:3443 --web --client-id=<your-aap-client-id> --insecure-skip-tls-verify
```

- This opens in a web browser and asks you to approve.
  The `--insecure-skip-tls-verify` parameter is used only if you have not generated your own valid certificates.

Next steps

Use the following commands to help you with the CLI:

- To output a list of available commands, use:

```
flightctl
```

- To output both the flightctl CLI version and the back-end Red Hat Edge Manager version, use:

```
flightctl version
```

IMPORTANT:
To ensure supportability and proper functionality, the version of the flightctl CLI must match the version of the Red Hat Edge Manager in use. Mismatched versions are not supported.

Optional: Request an enrollment certificate for early binding

If you want to include an agent configuration in the image, complete the following steps:

Procedure

1. Log in to the flightctl CLI by following the steps in Logging into the Red Hat Edge Manager through the CLI.

NOTE:
The CLI uses the certificate authority pool of the host to verify the identity of the Red Hat Edge Manager service. The verification can lead to a TLS verification error when using self-signed certificates, if you do not add your certificate authority certificate to the pool. You can bypass the server verification by adding the `--insecure-skip-tls-verify` flag to your command.

2. Get the enrollment credentials in the format of an agent configuration file by running the following command:

```
flightctl certificate request --signer=enrollment --expiration=365d --
output=embedded > config.yaml
```

NOTE:

- The `--expiration=365d` option specifies that the credentials are valid for a year.
- The `--output=embedded` option specifies that the output is an agent configuration file with the enrollment credentials embedded.

The returned `config.yaml` contains the URLs of the Red Hat Edge Manager service, the certificate authority bundle, and the enrollment client certificate and key for the agent. See the following example:

```
enrollment-service:
  authentication:
    client-certificate-data: LS0tLS1CRUdJTiBD...
    client-key-data: LS0tLS1CRUdJTiBF...
  service:
    certificate-authority-data: LS0tLS1CRUdJTiBD...
    server: https://agent-api.flightctl.127.0.0.1.nip.io:7443
  enrollment-ui-endpoint: https://ui.flightctl.127.0.0.1.nip.io:8081
```

Optional: Use image pull secrets

If your device relies on containers from a private repository, you must configure a pull secret for the registry. Complete the following steps:

Procedure

1. Depending on the kind of container image you use, place the pull secret in one or both of the following system paths on the device:
   - Operating system images use the `/etc/ostree/auth.json` path.
   - Application container images use the `/root/.config/containers/auth.json` path.

   IMPORTANT:
   The pull secret must exist on the device before the secret can be consumed.

2. Ensure that the pull secrets use the following format:

```
{
  "auths": {
    "registry.example.com": {
      "auth": "base64-encoded-credentials"
    }
  }
}
```

Related information
[Configuring container pull secrets when building Red Hat Edge Manager operating system images for different target platforms]

Build the operating system image with *bootc*

Build the operating system image with the `bootc` that contains the Red Hat Edge Manager agent. You can optionally include the following items in your operating system image:

- The agent configuration for early binding
- Any drivers
- Host configuration
- Application workloads that you need

Complete the following steps:

Procedure

1. Create a `Containerfile` file with the following content to build a RHEL 9-based operating system image that includes the Red Hat Edge Manager agent and configuration:
2. Build instructions for the bootc image:

```
FROM registry.redhat.io/rhel9/rhel-bootc:<required_os_version>

RUN dnf --enablerepo ansible-automation-platform-2.5-for-rhel-9-x86_64-rpms -y
install flightctl-agent-0.7.2-1.el9fc  && \

    dnf -y clean all && \

    systemctl enable flightctl-agent.service && \

    systemctl mask bootc-fetch-apply-updates.timer
```

<required_os_version>

The base image referenced in `FROM` is a bootable container (`bootc`) image that already has a Linux kernel. You can reuse existing standard container build tools and workflows.

systemctl mask bootc-fetch-apply-updates.timer

This part of the `RUN` command disables the default automatic updates. The updates are managed by the Red Hat Edge Manager.

IMPORTANT:
If your device relies on containers from a private repository, you must place the device pull secret in the `/etc/ostree/auth.json` path. The pull secret must exist on the device before the secret can be consumed.

- **Optional:** To enable `podman-compose` application support, add the following section to the `Containerfile` file:

```
RUN dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm && \
    dnf -y install podman-compose && \
    dnf -y clean all && \
    systemctl enable podman.service
```

- **Optional:** If you created the `config.yaml` for early binding, add the following section to the `Containerfile`:

```
ADD config.yaml /etc/flightctl/
```

For more information, see Optional: Requesting an enrollment certificate for early binding.

3. Define the Open Container Initiative (OCI) registry by running the following command:

```
OCI_REGISTRY=registry.redhat.io
```

4. Define the image repository that you have permissions to write to by running the following command:

```
OCI_IMAGE_REPO=${OCI_REGISTRY}/<your_org>/<your_image>
```

5. Define the image tag by running the following command:

```
OCI_IMAGE_TAG=v1
```

6. Build the operating system image for your target platform:

```
sudo podman build -t ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG} .
```

Sign and publish the bootc operating system image by using Sigstore

To sign the `bootc` operating system image by using Sigstore, complete the following steps:

Procedure

1. Generate a Sigstore key pair named `signingkey.pub` and `signingkey.private`:

```
skopeo generate-sigstore-key --output-prefix signingkey
```

2. Configure container tools such as Podman and Skopeo to upload Sigstore signatures together with your signed image to your OCI registry:

```
sudo tee "/etc/containers/registries.d/${OCI_REGISTRY}.yaml" > /dev/null <<EOF
docker:
    ${OCI_REGISTRY}:
        use-sigstore-attachments: true
EOF
```

3. Log in to your OCI registry by running the following command:

```
sudo podman login ${OCI_REGISTRY}
```

4. Sign and publish the operating system image by running the following command:

```
sudo podman push \
    --sign-by-sigstore-private-key ./signingkey.private \
    ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Build the operating system disk image

Build the operating system disk image that has the file system for your devices.

Procedure

1. Create a directory called `output` by running the following command:

```
mkdir -p output
```

2. Use `bootc-image-builder` to generate an operating system disk image of type `iso` from your operating system image by running the following command:

```
sudo podman run --rm -it --privileged --pull=newer \
    --security-opt label=type:unconfined_t \
    -v "${PWD}/output":/output \
    -v /var/lib/containers/storage:/var/lib/containers/storage \
    registry.redhat.io/rhel9/bootc-image-builder:latest \
    --type iso \
    ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Result

When the `bootc-image-builder` completes, you can find the ISO disk image at the `${PWD}/output/bootiso/install.iso` path.

Optional: Sign and publish the operating system disk image to an Open Container Initiative registry

Sign and publish disk images to your Open Container Initiative (OCI) registry. Optionally compress them as OCI artifacts for unified distribution with bootc images. To publish an ISO, use a repository named after your `bootc` image with `/diskimage-iso` appended.

Before you begin

- You created a private key by using Sigstore. See Signing and publishing the *bootc* operating system image by using Sigstore.

Sign and publish your disk image to your OCI registry by completing the following steps:

Procedure

1. Change the owner of the directory where the ISO disk image is located from `root` to your current user by running the following command:

```
sudo chown -R $(whoami):$(whoami) "${PWD}/output"
```

2. Define the `OCI_DISK_IMAGE_REPO` environmental variable to be the same repository as your `bootc` image with `/diskimage-iso` appended by running the following command:

```
OCI_DISK_IMAGE_REPO=${OCI_IMAGE_REPO}/diskimage-iso
```

3. Create a manifest list by running the following command:

```
sudo podman manifest create \
    ${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

4. Add the ISO disk image to the manifest list as an OCI artifact by running the following command:

```
sudo podman manifest add \
    --artifact --artifact-type application/vnd.diskimage.iso \
    --arch=amd64 --os=linux \
    ${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG} \
    "${PWD}/output/bootiso/install.iso"
```

5. Sign the manifest list with your private Sigstore key and push the image to the registry by running the following command:

```
sudo podman manifest push --all \
     --sign-by-sigstore-private-key ./signingkey.private \
    ${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG} \
    docker://${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Additional resources

Access supplementary documentation for detailed guidance on building Red Hat Edge Manager operating system images for different target platforms, including how to configure container pull secrets.

- For more information about building the operating system image on different target platforms, see Configuring container pull secrets.

Requirements for specific target platforms

Review the specific requirements and procedures necessary to prepare operating system images for deployment onto target platforms such as Red Hat OpenShift Virtualization and VMware vSphere.

- Building images for Red Hat OpenShift Virtualization
- Building images for VMware vSphere

Build images for Red Hat OpenShift Virtualization

When building operating system images and disk images for Red Hat OpenShift Virtualization, you can follow the generic image building process with the following changes:

- Using late binding by injecting the enrollment certificate or the agent configuration through `cloud-init` when provisioning the virtual device.
- Adding the `open-vm-tools` guest tools to the image.
- Building a disk image of type `qcow2` instead of `iso`.

Complete the generic steps with changes to the following steps:

Procedure

1. Build an operating system image based on RHEL 9 that includes the Red Hat Edge Manager agent and VM guest tools but excludes the agent configuration.
2. Create a file named `Containerfile` with the following content:

```
 FROM registry.redhat.io/rhel9/bootc-image-builder:latest

RUN subscription-manager repos --enable ansible-automation-platform-2.5-for-
rhel-9-x86_64-rpms

    dnf -y install flightctl-agent && \

    dnf -y clean all && \

    systemctl enable flightctl-agent.service

RUN dnf -y install cloud-init open-vm-tools && \

    dnf -y clean all && \

    ln -s ../cloud-init.target /usr/lib/systemd/system/default.target.wants &&
\

    systemctl enable vmtoolsd.service
```

3. **Optional:** To enable `podman-compose` application support, add the following section to the `Containerfile` file:

```
RUN dnf install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-9.noarch.rpm && \
    dnf -y install podman-compose && \
    dnf -y clean all && \
    systemctl enable podman.service
```

Build the bootc image

Build, sign, and publish the `bootc` operating system image by following the generic image building process:

Procedure

1. Create a directory called `output` by running the following command:

```
mkdir -p output
```

2. Generate an operating system disk image of type `vmdk` from your operating system image by running the following command:

```
sudo podman run --rm -it --privileged --pull=newer \
    --security-opt label=type:unconfined_t \
    -v "${PWD}/output":/output \
    -v /var/lib/containers/storage:/var/lib/containers/storage \
    registry.redhat.io/rhel9/bootc-image-builder:latest \
    --type qcow2 \
    ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Next steps

When the `bootc-image-builder` completes, you can find the disk image under `${PWD}/output/vmdk/disk.vmdk`.

Build the QCoW2 disk image

Red Hat OpenShift Virtualization can download disk images from an OCI registry but it expects a container disk image instead of an OCI artifact.

Complete the following steps to build, sign, and upload the QCoW2 disk image:

Procedure

1. Create a file called `Containerfile.qcow2` with the following content:

```
FROM registry.access.redhat.com/ubi9/ubi:latest AS builder

ADD --chown=107:107 output/qcow2/disk.qcow2 /disk/

RUN chmod 0440 /disk/*

FROM scratch

COPY --from=builder /disk/* /disk/
```

ADD --chown=107:107 output/qcow2/disk.qcow2 /disk/

Adds the QCoW2 disk image to a builder container to set the required `107` file ownership, which is the QEMU user.

RUN chmod 0440 /disk/*

Sets the required `0440` file permissions.

COPY --from=builder /disk/* /disk/

Copies the file to a scratch image.

2. Build, sign, and publish your disk image by running the following command:

```
sudo chown -R $(whoami):$(whoami) "${PWD}/output"

OCI_DISK_IMAGE_REPO=${OCI_IMAGE_REPO}/diskimage-qcow2

sudo podman build -t ${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG} -f
Containerfile.qcow2 .

sudo podman push --sign-by-sigstore-private-key ./signingkey.private $
{OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Build images for VMware vSphere

When building operating system images and disk images for VMware vSphere, you can follow the generic image building process with the following changes:

- Using late binding by injecting the enrollment certificate or the agent configuration through `cloud-init` when provisioning the virtual device.
- Adding the `open-vm-tools` guest tools to the image.
- Building a disk image of type `vmdk` instead of `iso`.

Complete the generic steps with changes to the following steps:

Procedure

1. Build an operating system image based on RHEL 9 that includes the Red Hat Edge Manager agent and VM guest tools but excludes the agent configuration.
2. Create a file named `Containerfile` with the following content:

```
FROM registry.redhat.io/rhel9/bootc-image-builder:latest

RUN subscription-manager repos --enable ansible-automation-platform-2.5-for-rhel-9-x86_64-rpms

    dnf -y install flightctl-agent && \

    dnf -y clean all && \

    systemctl enable flightctl-agent.service && \

RUN dnf -y install cloud-init open-vm-tools && \

    dnf -y clean all && \

    ln -s ../cloud-init.target /usr/lib/systemd/system/default.target.wants && \

    systemctl enable vmtoolsd.service
```

3. Create a directory called `output` by running the following command:

```
mkdir -p output
```

4. Generate an operating system disk image of type `vmdk` from your operating system image by running the following command:

```
sudo podman run --rm -it --privileged --pull=newer \

    --security-opt label=type:unconfined_t \

    -v "${PWD}/output":/output \

    -v /var/lib/containers/storage:/var/lib/containers/storage \

    registry.redhat.io/rhel9/bootc-image-builder:latest \

    --type vmdk \

    ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG}
```

Next steps

When the `bootc-image-builder` completes, you can find the disk image under `${PWD}/output/vmdk/disk.vmdk`.

Provision edge devices

You can provision devices with the Red Hat Edge Manager in different environments. Use the operating system image or disk image that you built for use with the Red Hat Edge Manager. Depending on your target environment, provision a physical or virtual device.

Related information
Provision physical devices
Provision devices with OpenShift Virtualization

Provision physical devices

When you build an International Organization for Standardization (ISO) disk image from an operating system image by using the `bootc-image-builder` tool, the image is similar to the RHEL ISOs available for download. However, your operating system image content is embedded in the ISO disk image.

Related information
Deploying a custom ISO container image
Deploying an ISO *bootc* image over PXE boot

Provision devices with OpenShift Virtualization

You can provision a virtual machine on OpenShift Virtualization by using a QCoW2 container disk image that is hosted on an OCI container registry.

If your operating system image does not already contain the Red Hat Edge Manager agent enrollment configuration, you can inject the configuration through the `cloud-init` user data at provisioning.

Create the `cloud-init` configuration

The `cloud-init` configuration customizes a virtual machine instance on its first boot, allowing you to automatically enroll it as a new agent in your Red Hat Edge Manager service.

Before you begin

- You installed the `flightctl` CLI and logged in to your Red Hat Edge Manager service instance.
- You installed the `oc` CLI, used it to log in to your OpenShift cluster instance, and changed to the project in which you want to create your virtual machine.

Procedure

1. Request a new Red Hat Edge Manager agent enrollment configuration and store it in a file called `config.yaml` by running the following command:

```
flightctl certificate request --signer=enrollment --expiration=365d --
output=embedded > config.yaml
```

2. Create a cloud configuration user data file called `cloud-config.yaml` that places the agent configuration in the correct location on the first boot by running the following command:

```
cat <<EOF > cloud-config.yaml
#cloud-config
write_files:
- path: /etc/flightctl/config.yaml
  content: $(cat config.yaml | base64 -w0)
  encoding: b64
EOF
```

3. Create a Kubernetes `Secret` that contains the cloud configuration user data file:

```
oc create secret generic enrollment-secret --from-file=userdata=cloud-
config.yaml
```

Create the virtual machine

Create a virtual machine that has its primary disk populated from your QCoW2 container disk image and a `cloud-init` configuration drive that is populated from your enrollment secret.

Complete the following steps:

Procedure

1. Create a file that has the `VirtualMachine` resource manifest by running the following command:

```
cat <<EOF > my-bootc-vm.yaml
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: my-bootc-vm
spec:
  runStrategy: RerunOnFailure
  template:
    spec:
      domain:
        cpu:
          cores: 1
        memory:
          guest: 1024M
        devices:
          disks:
            - name: containerdisk
              disk:
                bus: virtio
            - name: cloudinitdisk
              disk:
                bus: virtio
      volumes:
        - name: containerdisk
          containerDisk:
            image: ${OCI_DISK_IMAGE_REPO}:${OCI_IMAGE_TAG}
        - name: cloudinitdisk
          cloudInitConfigDrive:
            secretRef:
              name: enrollment-secret
EOF
```

2. Apply the resource manifest to your cluster by running the following command:

```
oc apply -f my-bootc-vm.yaml
```

Related information
Cloud-init documentation
Building images for Red Hat OpenShift Virtualization

Enroll and view devices

The Red Hat Edge Manager manages the device lifecycle from enrollment to decommissioning of a device. The device lifecycle also includes device management, such as organizing, monitoring, and updating your devices with the Red Hat Edge Manager.

You can manage your devices individually or in a fleet. With the Red Hat Edge Manager you can manage a whole fleet of devices as a single object instead of managing many devices individually.

You only need to specify the required configuration once, and then the Red Hat Edge Manager applies the configuration to all devices in the fleet.

Understanding individual device management is the foundation for managing devices in a fleet. You might want to manage your devices individually in the following scenarios:

- If a few devices have different configurations.
- If you use external automation for updating the device.

The following sections focus on managing individual devices:

Related information
Enroll and view devices
Labels and label selectors
Update labels on the CLI
Update the operating system
Operating system configuration for edge devices

Enroll devices

To manage your devices with the Red Hat Edge Manager, you must enroll the devices to the Red Hat Edge Manager service.

The first time the Red Hat Edge Manager agent runs on a device, the agent prepares for the enrollment process by generating a cryptographic key pair. The cryptographic key pair serves as the unique cryptographic identity of the device. The key pair consists of a public and a private key. The private key never leaves the device, so that the device cannot be duplicated or impersonated.

When the device is not yet enrolled, the agent performs service discovery to find its Red Hat Edge Manager service instance. Then, the device establishes a secure, mTLS-protected network connection to the service. The device uses its X.509 enrollment certificate that the device acquired during image building or device provisioning. The device submits an enrollment request to the service that includes the following:

- a description of the device hardware and operating system
- an X.509 Certificate Signing Request which includes the cryptographic identity of the device to obtain the initial management certificate

The device is not considered trusted and remains quarantined in a device lobby until an authorized user approves or denies the request.

For more information, see the following sections:

Related information
Optional: Request an enrollment certificate for early binding

Enroll devices on the CLI

You must enroll devices into the Red Hat Edge Manager service before you can manage them.

Before you begin

- You must install the Red Hat Edge Manager CLI. See Installing the Red Hat Edge Manager CLI.
- You must log in to the Red Hat Edge Manager service.

Procedure

1. List all devices that are currently waiting for approval by running the following command:

```
flightctl get enrollmentrequests --field-selector="status.approval.approved != true"
```

See the following example:

```
NAME           APPROVAL   APPROVER   APPROVED LABELS
<device_name>  Pending    <none>     <none>
```

NOTE:
The unique device name is generated by the agent and you cannot change it. The agent chooses a base32-encoded hash of its public key as the device name.

2. Approve an enrollment request by specifying the name of the enrollment request. Optionally, you can add labels to the device by using the `--label` or `-l` flags. See the following example:

```
flightctl approve -l region=eu-west-1 -l site=factory-berlin
enrollmentrequest/54shovu028bvj6stkovjcvovjgo0r48618khdd5huhdjfn6raskg
```

See the following example output:

```
NAME            APPROVAL   APPROVER   APPROVED LABELS
<device_name>   Approved   user       region=eu-west-1,site=factory-berlin
```

Next steps

After you approve the enrollment request, the service issues the management certificate for the device and registers the device in the device inventory. You can then manage the device.

View devices

To get more information about the devices in your inventory, you can use the Red Hat Edge Manager CLI.

View device inventory and device details on the web UI

You can view details for enrolled devices, including their status and health, on the Red Hat Edge Manager web UI.

Before you begin

- You must install the Red Hat Edge Manager CLI. See Installing the Red Hat Edge Manager CLI.
- You must enroll at least one device.

Procedure

1. From the navigation panel, select **Application Links > Edge Manager**. This opens the external Edge Manager instance.
2. From the navigation panel, select **Devices** where you can view your device inventory, details, and decommission devices.

View device inventory and device details on the CLI

View the device inventory and retrieve detailed information by using the `flightctl` command.

Procedure

1. View the devices in the device inventory by running the following command:

```
flightctl get devices
```

See the following example output:

```
NAME           ALIAS    OWNER    SYSTEM  UPDATED     APPLICATIONS  LAST SEEN
<device_name>  <none>   <none>  Online  Up-to-date  <none>        3 seconds
ago
```

2. View the details of this device in YAML format by running the following command:

```
flightctl get device/<device_name> -o yaml
```

See the following example output:

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: <device_name>
  labels:
    region: eu-west-1
    site: factory-berlin
spec:
  os:
    image: quay.io/flightctl/rhel:9.5
  config:
  - name: my-os-configuration
    configType: GitConfigProviderSpec
    gitRef:
      path: /configuration
      repository: my-configuration-repo
      targetRevision: production
status:
  os:
    image: quay.io/flightctl/rhel:9.5
```

```
  config:
    renderedVersion: "1"
  applications:
    data: {}
    summary:
      status: Unknown
  resources:
    cpu: Healthy
    disk: Healthy
    memory: Healthy
  systemInfo:
    architecture: amd64
    bootID: 037750f7-f293-4c5b-b06e-481eef4e883f
    operatingSystem: linux
  summary:
    info: ""
    status: Online
  updated:
    status: UpToDate
  lastSeen: "2024-08-28T11:45:34.812851905Z"
# [...]
```

labels

User-defined labels assigned to the device.

spec:os:image

The target operating system image version of the device.

spec:config

The target operating system configuration of the device.

status:os:image

The current operating system image version of the device.

status:config:renderedVersion

The current operating system configuration version of the device.

status:applications:data

The current list of deployed applications of the device.

status:applications:summary

The health status of applications on the device.

status:resources

The availability of CPU, disk, and memory resources.

status:systemInfo

Basic system information.

status:summary:status

The health status of the device.

status:updated:status

The update status of the device.

status:lastSeen

The last check-in time and date of the device.

Labels and label selectors

You can organize resources by assigning labels for location, hardware, or purpose. The Red Hat Edge Manager labels follow the same syntax, principles, and operators as Kubernetes labels and label selectors. Use these labels to select devices or apply operations to devices in the inventory.

Labels follow the `key=value` format. You can use the key to group devices. For example, if your labels follow the `site=<location>` naming convention, you can group your devices by site. You can also use labels that only consist of keys.

Labels must adhere to the following rules to be valid:

- Keys and value must each be 63 characters or less.
- Keys and values can consist of alphanumeric characters (`a-z` , `A-Z` , `0-9`).
- Keys and values can also contain dashes (`-`), underscores (`_`), dots (`.`) but not as the first or last character.
- Value can be omitted.

You can apply labels to devices in the following ways:

- Define a set of default labels during image building that are automatically applied to all devices during deployment.
- Assign initial labels during enrollment.
- Assign labels post-enrollment.

When resources are labeled, you can select a subset of devices by creating a label selector. A label selector is a comma-separated list of labels for selecting devices that have the same set of labels.

See the following examples:

Example label selector	Selected devices
`site=factory-berlin`	All devices with a `site` label key and a `factory-berlin` label value.
`site!=factory-berlin`	All devices with a `site` label key but where the label value is not `factory-berlin`.
`site in (factory-berlin,factory-madrid)`	All devices with a `site` label key and where the label value is either `factory-berlin` or `factory-madrid`.

Related information
Labels and Selectors

View devices and their labels on the web UI

View devices and their associated labels on the web UI. You can use labels to organize your devices and device fleets.

Procedure

1. From the navigation panel, select **Application Links > Edge Manager**. This opens the external Edge Manager instance.
2. From the navigation panel, select **Devices**.
3. Select the device you want to manage. In the **Details** tab you can view the associated labels under **Labels**.

View devices and their labels on the CLI

View devices and their associated labels. You can use labels to organize your devices and device fleets.

Complete the following steps:

Procedure

1. View devices in your inventory with their labels by using the `-o wide` option:

```
flightctl get devices -o wide
```

See the following example output:

```
NAME            ALIAS    OWNER   SYSTEM  UPDATED     APPLICATIONS  LAST SEEN
   LABELS

<device1_name>  <none>   <none>  Online  Up-to-date  <none>        3 seconds
ago  region=eu-west-1,site=factory-berlin

<device2_name>  <none>   <none>  Online  Up-to-date  <none>        1 minute
ago   region=eu-west-1,site=factory-madrid
```

2. View devices in your inventory with a specific label or set of labels by using the `-l <key=value>` option:

```
flightctl get devices -l site=factory-berlin -o wide
```

See the following example output:

```
NAME            ALIAS    OWNER   SYSTEM  UPDATED     APPLICATIONS  LAST SEEN
   LABELS

<device1_name>  <none>   <none>  Online  Up-to-date  <none>        3 seconds
ago  region=eu-west-1,site=factory-berlin
```

Update labels on the CLI

You can update the labels on your devices by using the Red Hat Edge Manager CLI.

Complete the following steps:

Procedure

1. Export the current definition of the device into a file by running the following command:

```
flightctl get device/<device1_name> -o yaml > my_device.yaml
```

2. Use your preferred editor to edit the `my_device.yaml` file. See the following example:

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  labels:
    some_key: some_value
    some_other_key: some_other_value
  name: <device1_name>
spec:
[...]
```

3. Save the file and apply the updated device definition by running the following command:

```
flightctl apply -f my_device.yaml
```

4. Verify your changes by running the following example output:

```
NAME            ALIAS    OWNER   SYSTEM  UPDATED     APPLICATIONS  LAST SEEN
   LABELS

<device1_name>  <none>   <none>  Online  Up-to-date  <none>        3 minutes
ago  some_key=some_value,some_other_key=some_other_value

<device2_name>  <none>   <none>  Online  Up-to-date  <none>        4 minutes
ago  region=eu-west-1,site=factory-madrid
```

Filter a list with field selectors

Field selectors filter a list of Red Hat Edge Manager resources based on specific resource field values. They follow the same syntax, principles, and operators as Kubernetes Field and Label selectors, with additional operators available for more advanced search use cases.

Supported fields

Red Hat Edge Manager resources give a set of metadata fields that you can select.

Each resource supports the following metadata fields:

- `metadata.name`
- `metadata.owner`

- `metadata.creationTimestamp`

NOTE:
To query labels, use Label Selectors for advanced and flexible label filtering.

For more information, see Labels and label selectors.

List of additional supported fields

In addition to the metadata fields, each resource has its own unique set of fields that you can select, offering further flexibility in filtering and selection based on resource-specific attributes.

The following table lists the fields supported for filtering for each resource kind:

Kind	Fields
Certificate Signing Request	`status.certificate`

Device	`status.summary.status`
	`status.applicationsSummary.status`
	`status.updated.status`
	`status.lastSeen`
	`status.lifecycle.status`

Enrollment Request	`status.approval.approved` `status.certificate`
Fleet	`spec.template.spec.os.image`
Repository	`spec.type` `spec.url`
Resource Sync	`spec.repository`

Fields discovery

Some Red Hat Edge Manager resources might expose additional supported fields. You can discover the supported fields by using `flightctl` with the `--field-selector` option. If you try to use an unsupported field, the error message lists the available supported fields.

See the following examples:

```
flightctl get device --field-selector='text'
```

```
Error: listing devices: 400, message: unknown or unsupported selector: unable to
resolve selector name "text". Supported selectors are: [metadata.alias
metadata.creationTimestamp metadata.name metadata.nameoralias metadata.owner
status.applicationsSummary.status status.lastSeen status.summary.status
status.updated.status]
```

The field `text` is not a valid field for filtering. The error message provides a list of supported fields that you can use with `--field-selector` for the `Device` resource.

You can then use one of the supported fields:

```
flightctl get devices --field-selector 'metadata.alias contains cluster'
```

The `metadata.alias` field is checked with the containment operator `contains` to see if it has the value `cluster`.

Examples

Example 1: Excluding a specific device by name

The following command filters out a specific device by its name:

```
flightctl get devices --field-selector 'metadata.name!
=c3tkb18x9fw32fzx5l556n0p0dracwbl4uiojxu19g2'
```

Example 2: Filter by owner, labels, and creation timestamp

This command retrieves devices owned by `Fleet/pos-fleet`, located in the `us` region, and created in 2024:

```
flightctl get devices --field-selector 'metadata.owner=Fleet/pos-fleet,
metadata.creationTimestamp >= 2024-01-01T00:00:00Z, metadata.creationTimestamp < //
2025-01-01T00:00:00Z' -l 'region=us'
```

Example 3: Filter by Owner, Labels, and Device Status

This command retrieves devices owned by `Fleet/pos-fleet`, located in the `us` region, and with a `status.updated.status` of either `Unknown` or `OutOfDate`:

```
flightctl get devices --field-selector 'metadata.owner=Fleet/pos-fleet,
status.updated.status in (Unknown, OutOfDate)' -l 'region=us'
```

Supported operators

Learn the operators and corresponding symbols you can use to construct sophisticated field selectors when querying or filtering Red Hat Edge Manager resources. This enables precise and flexible control over resource selection.

Operator	Symbol	Description
Exists	`exists`	Checks if a field exists
DoesNotExist	`!`	Checks if a field does not exist
Equals	`=`	Checks if a field is equal to a value
DoubleEquals	`==`	Another form of equality check
NotEquals	`!=`	Checks if a field is not equal to a value
GreaterThan	`>`	Checks if a field is greater than a value
GreaterThanOrEquals	`>=`	Checks if a field is greater than or equal to a value
LessThan	`<`	Checks if a field is less than a value
LessThanOrEquals	`⇐`	Checks if a field is less than or equal to a value
In	`in`	Checks if a field is within a list of values
NotIn	`notin`	Checks if a field is not in a list of values
Contains	`contains`	Checks if a field has a value
NotContains	`notcontains`	Checks if a field does not contain a value

Operators usage by field type

Each field type supports a specific subset of operators:

Field Type	Supported Operators	Value
String	`Equals` : Matches if the field value is an exact match to the specified string. `DoubleEquals` : Matches if the field value is an exact match to the specified string (alternative to `Equals`). `NotEquals` : Matches if the field value is not an exact match to the specified string. `In` : Matches if the field value matches at least one string in the list. `NotIn` : Matches if the field value does not match any of the strings in the list. `Contains` : Matches if the field value has the specified substring. `NotContains` : Matches if the field value does not contain the specified substring. `Exists` : Matches if the field is present. `DoesNotExist` : Matches if the field is not present.	Text string
Timestamp	`Equals` : Matches if the field value is an exact match to the specified timestamp. `DoubleEquals` : Matches if the field value is an exact match to the specified	RFC 3339 format

Field Type	Supported Operators	Value
	timestamp (alternative to `Equals`). `NotEquals`: Matches if the field value is not an exact match to the specified timestamp. `GreaterThan`: Matches if the field value is after the specified timestamp. `GreaterThanOrEquals`: Matches if the field value is after or equal to the specified timestamp. `LessThan`: Matches if the field value is before the specified timestamp. `LessThanOrEquals`: Matches if the field value is before or equal to the specified timestamp. `In`: Matches if the field value matches at least one timestamp in the list. `NotIn`: Matches if the field value does not match any of the timestamps in the list. `Exists`: Matches if the field is present. `DoesNotExist`: Matches if the field is not present.	
Number	`Equals`: Matches if the field value equals the specified number. `DoubleEquals`: Matches if the field value equals the specified number (alternative to `Equals`).	Number format

Field Type	Supported Operators	Value
	`NotEquals` : Matches if the field value does not equal to the specified number. `GreaterThan` : Matches if the field value is greater than the specified number. `GreaterThanOrEquals` : Matches if the field value is greater than or equal to the specified number. `LessThan` : Matches if the field value is less than the specified number. `LessThanOrEquals` : Matches if the field value is less than or equal to the specified number. `In` : Matches if the field value equals at least one number in the list. `NotIn` : Matches if the field value does not equal any numbers in the list. `Exists` :Matches if the field is present. `DoesNotExist` : Matches if the field is not present.	
Boolean	`Equals` : Matches if the value is `true` or `false` . `DoubleEquals` : Matches if the value is `true` or `false` (alternative to `Equals`). `NotEquals` : Matches if the value is the opposite of the specified value.	Boolean format (`true` , `false`)

Field Type	Supported Operators	Value
	`In` : Matches if the value (`true` or `false`) is in the list. **NOTE:** The list can only contain `true` or `false` , so this operator is limited in use. `NotIn` : Matches if the value is not in the list. `Exists` : Matches if the field is present. `DoesNotExist` : Matches if the field is not present.	
Array	`Contains` : Matches if the array has the specified value. `NotContains` : Matches if the array does not contain the specified value. `In` : Matches if the array overlaps with the specified values. `NotIn` : Matches if the array does not overlap with the specified values. `Exists` : Matches if the field is present. `DoesNotExist` :Matches if the field is not present. **NOTE:** Using `Array[Index]` treats the element as the type defined for the array elements. For example string, timestamp, number, or boolean.	Array element

Update the operating system

Update a device's operating system by changing the target operating system image name or version in the device specification. The agent detects the requested update upon communicating with the server and automatically begins downloading and verifying the new operating system in the background.

The Red Hat Edge Manager agent schedules the actual system update that is performed according to the update policy. At the scheduled update time, the agent installs the new version without disrupting the currently running operating system. Finally, the device reboots into the new version.

The Red Hat Edge Manager currently supports the following image type and image reference format:

Image Type	Image Reference
bootc	An OCI image reference to a container registry. Example: `quay.io/flightctl-example/rhel:9.5`

During the process, the agent sends status updates to the service. You can check the update process by viewing the device status.

Related information
Enroll and view devices

Update the operating system on the CLI

You can update the operating system on an individual device by specifying a new target image in the device manifest using the Red Hat Edge Manager CLI. This initiates a secure, transactional update process managed automatically by the device agent.

Procedure

1. Get the current resource manifest of the device by running the following command:

```
flightctl get device/<device_name> -o yaml > my_device.yaml
```

2. Edit the `Device` resource to specify the new operating system name and version target.

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: <device_name>
spec:
[...]
  os:
    image: quay.io/flightctl/rhel:9.5
[...]
```

3. Apply the updated `Device` resource by running the following command:

```
flightctl apply -f <device_name>.yaml
```

Operating system configuration for edge devices

You can include an operating system-level host configuration in the image to give maximum consistency and repeatability. To update the configuration, create a new operating system image and update devices with the new image.

However, updating devices with a new image can be impractical in the following cases:

- The configuration is missing in the image.
- The configuration needs to be specific to a device.
- The configuration needs to be updateable at runtime without updating the operating system image and rebooting.

For these cases, you can declare a set of configuration files that are present on the file system of the device. The Red Hat Edge Manager agent applies updates to the configuration files while ensuring that either all files are successfully updated in the file system, or rolled back to their pre-update state. If the user updates both an operating system and configuration set of a device at the same time, the Red Hat Edge Manager agent updates the operating system first. It then applies the specified set of configuration files.

You can also specify a list of configuration sets that the Red Hat Edge Manager agent applies in sequence. In case of a conflict, the last applied configuration set is valid.

IMPORTANT:
After the Red Hat Edge Manager agent updates the configuration on the disk, the running applications need to reload the new configuration into memory for the configuration to become effective. If the update involves a reboot, `systemd` automatically restarts the applications with the new configuration and in the correct order. If the update does not involve a reboot, many applications can detect changes to their configuration files and automatically reload the files. When an application does not support change detection, you can use device lifecycle hooks to run scripts or commands if certain conditions are met.

Configuration providers

You can provide configuration from many sources, called configuration providers, in Red Hat Edge Manager. The Red Hat Edge Manager currently supports the following configuration providers:

Git Config Provider

Fetches device configuration files from a Git repository.

Kubernetes Secret Provider

Fetches a secret from a Kubernetes cluster and writes the content to the file system of the device.

HTTP Config Provider

Fetches device configuration files from an HTTP(S) endpoint.

Inline Config Provider

Allows specifying device configuration files inline in the device manifest without querying external systems.

Configuration from a Git repository

You can store device configuration in a Git repository such as GitHub or GitLab. You can then add a Git Config Provider so that the Red Hat Edge Manager synchronizes the configuration from the repository to the file system of the device.

The Git Config Provider takes the following parameters:

Parameter	Description
`Repository`	The name of a `Repository` resource defined in the Red Hat Edge Manager.
`TargetRevision`	The branch, tag, or commit of the repository to checkout.
`Path`	The absolute path to the directory in the repository from which files and subdirectories are synchronized to the file system of the device.

Parameter	Description
	The `Path` directory corresponds to the root directory (`/`) on the device, unless you specify the `MountPath` parameter.
`MountPath`	Optional. The absolute path to the directory in the file system of the device to write the content of the repository to. By default, the value is the file system root (`/`).

The `Repository` resource defines the Git repository, the protocol, and the access credentials that the Red Hat Edge Manager must use. You only need to set up the repository once. After setting up, you can use the repository to configure individual devices or device fleets.

Secrets from a Kubernetes cluster

The Red Hat Edge Manager can query only the Kubernetes cluster that the Red Hat Edge Manager is running on for a Kubernetes secret. You can write the content of that secret to a path on the device file system.

The Kubernetes Secret Provider takes the following parameters:

Parameter	Description
`Name`	The name of the secret.
`NameSpace`	The namespace of the secret.
`MountPath`	The directory in the file system of the device to write the secret contents to.

NOTE:
The Red Hat Edge Manager needs permission to access secrets in the defined namespace. For example, creating a `ClusterRole` and `ClusterRoleBinding` allows the `flightctl-worker` service account to get and list secrets in that namespace.

Configuration from an HTTP server

The Red Hat Edge Manager can query an HTTP server for configuration. The HTTP server can serve static or dynamically generated configuration for a device.

The HTTP Config Provider takes the following parameters:

Parameter	Description
`Repository`	The name of a `Repository` resource defined in the Red Hat Edge Manager.
`Suffix`	The suffix to append to the base URL defined in the `Repository` resource. The suffix can include path and query parameters, for example `/path/to/endpoint?query=param`.
`FilePath`	The absolute path to the file in the file system of the device to write the response of the HTTP server to.

The `Repository` resource specifies the HTTP server for the Red Hat Edge Manager to connect to, and the protocol and access credentials to use. You must set up the repository needs once, and then you can use the repository to configure many devices or device fleets.

Configuration inline in the device specification

You can specify configuration inline in a device specification. When you use the inline device specification, the Red Hat Edge Manager does not need to connect to external systems to fetch the configuration.

The Inline Config Provider takes a list of file specifications, where each file specification takes the following parameters:

Parameter	Description
`Path`	The absolute path to the file in the file system of the device to write the content to. If a file already exists in the specified path, the file is overwritten.
`Content`	The UTF-8 or base64-encoded content of the file.
`ContentEncoding`	Defines how the contents are encoded. Must be either `plain` or `base64`. Default value is set to `plain`.
`Mode`	Optional. The permission mode of the file. You can specify the octal with a leading zero, for example `0644`, or as a decimal without a leading zero, for example `420`. The `setuid`, `setgid`, and `sticky` bits are supported. If not specified, the permission mode for files defaults to `0644`.
`User`	Optional. The owner of the file. Specified either as a name or numeric ID. Default value is set to `root`.
`Group`	Optional. The group of the file. Specified either as a name or numeric ID.

Related information
Use device lifecycle hooks

Manage the device configuration from a Git repository on the CLI

Integrate your device configuration files with standard Git workflows by defining a Repository resource by using the CLI. This enables the Red Hat Edge Manager to automatically synchronize configuration file updates from the repository onto the device's file system.

Procedure

1. Create a file, for example `site-settings-repo.yaml`, that has the following definition for a `Repository` resource, named `site-settings`:

```
apiVersion: flightctl.io/v1alpha1
kind: Repository
metadata:
  name: site-settings
spec:
  type: git
  url: https://github.com/<your_org>/<your_repo>.git
```

2. Create the `Repository` resource by running the following command:

```
flightctl apply -f site-settings-repo.yaml
```

3. Verify that the resource has been correctly created and is accessible by Red Hat Edge Manager running the following command:

```
flightctl get repository/site-settings
```

See the following example output:

```
NAME           TYPE  REPOSITORY URL                                  ACCESSIBLE
site-settings  git   https://github.com/<your_org>/<your_repo>.git  True
```

4. Apply the `example-site` configuration to a device by updating the device specification:

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: <device_name>
spec:
[...]
  config:
  - name: example-site
    configType: GitConfigProviderSpec
    gitRef:
      repository: site-settings
      targetRevision: production
      path: /etc/example-site
[...]
```

config

The example configuration takes all the files from the `example-site` directory from the `production` branch of the `site-settings` repository and places the files in the root directory (`/`).

gitRef:path

Ensure that the target path is writeable by creating your directory structure. The root directory (`/`) is not writeable in `bootc` systems.

Run user-defined commands with device lifecycle hooks

The Red Hat Edge Manager agent uses lifecycle hooks to run user-defined commands at specific stages. For example, you can add a backup script to back up application data that must be completed before an operating system update can begin.

As another example, certain applications or system services do not automatically reload their configuration file when the file changes on the disk. You can manually reload the configuration file by specifying a command as another hook, which is called after the agent completes the update process.

The following device lifecycle hooks are supported:

Lifecycle Hook	Description
`beforeUpdating`	This hook is called after the agent completed preparing for the update and before actually making changes to the system. If an action in this hook returns with failure, the agent cancels the update.
`afterUpdating`	This hook is called after the agent has written the update to disk. If an action in this hook returns with failure,the agent cancels and rolls back the update.
`beforeRebooting`	This hook is called before the system reboots. The agent blocks the reboot until running the action has completed or timed out. If any action in this hook returns with failure, the agent cancels and rolls back the update.
`afterRebooting`	This hook is called when the agent first starts after a reboot. If any action in this hook returns with failure, the agent reports this but continues starting up.

Rule files

You can define device lifecycle hooks by adding rule files to one of the following locations in the device file system:

- Rules in the `/usr/lib/flightctl/hooks.d/<lifecycle_hook_name>/` drop-in directory are read-only. To add rules to the `/usr` directory, you must add them to the operating system image during image building.
- Rules in the `/etc/flightctl/hooks.d/<lifecycle_hook_name>/` drop-in directory are read-writable. You can update the rules at runtime by using several methods.

When creating and placing the files, you must consider the following practices:

- The name of the rule must be all lower case.
- If you define rules in both locations, the rules are merged.
- If you add more than one rule files to a lifecycle hook directory, the files are processed in lexical order of the file names.
- If you define files with identical file names in both locations, the file in the `/etc` folder takes precedence over the file of the same name in the `/usr` folder.

A rule file is written in YAML format and has a list of one or more actions. An action can be an instruction to run an external command.

When you specify many actions for a hook, the actions are performed in sequence, finishing one action before starting the next.

If an action returns with a failure, the following actions are skipped.

A `run` action takes the following parameters:

Parameter	Description
`Run`	The absolute path to the command to run, followed by any flags or arguments, for example `/usr/bin/nmcli connection reload`. The command is not executed in a shell, so you cannot use shell variables, such as `$PATH` or `$HOME`, or chain commands, such as `|` or `;`. If necessary, you can start a shell by specifying the shell as command to run, for example `/usr/bin/bash -c 'echo $SHELL $HOME $USER'`.
`EnvVars`	Optional. A list of key-value pairs to set as environment variables for the command.
`WorkDir`	Optional. The directory the command is run from.
`Timeout`	Optional. The maximum duration that is allowed for the action to complete. Specify the duration as a single positive integer followed by a time unit. The `s`, `m`, and `h` units are supported for seconds, minutes, and hours.
`If`	Optional. A list of conditions that must be true for the action to be run. If not provided, actions run unconditionally.

By default, the system performs actions every time the hook is triggered. However, for the `afterUpdating` hook, you can use the `If` parameter to add conditions that must be true for an action to be performed. Otherwise, the action is skipped.

For example, to run an action only if a given file or directory changes during the update, you can define a path condition that takes the following parameters:

Parameter	Description
`Job type`	An absolute path to a file or directory that must change during the update as a condition for the action to be performed. Specify paths by using forward slashes (/): • If the path is to a directory, it must end with a forward slash (`/`). • If you specify a path to a file, the file must have changed to satisfy the condition

Parameter	Description
	• If you specify a path to a directory, a file in that directory or any of its subdirectories must have changed to satisfy the condition
`Op`	A list of file operations, such as `created`, `updated`, and `removed`, to limit the type of changes to the specified path as a condition for the action to be performed.

If you specify a path condition for an action in the `afterUpdating` hook, you have the following variables that you can include in arguments to your command and are replaced with the absolute paths to the changed files:

Variable	Description
`${ Path }`	The absolute path to the file or directory specified in the path condition.
`${ Files }`	A space-separated list of absolute paths of the files that changed during the update and are covered by the path condition.
`${ CreatedFiles }`	A space-separated list of absolute paths of the files that were created during the update and are covered by the path condition.
`${ UpdatedFiles }`	A space-separated list of absolute paths of the files that were updated during the update and are covered by the path condition.
`${ RemovedFiles }`	A space-separated list of absolute paths of the files that were removed during the update and are covered by the path condition.

The Red Hat Edge Manager agent includes a built-in set of rules defined in `/usr/lib/flightctl/hooks.d/afterupdating/00-default.yaml`. The following commands are executed if certain files are changed:

File	Command	Description
`/etc/systemd/system/`	`systemctl daemon-reload`	Changes to `systemd` units are activated by signaling the `systemd` daemon to reload the `systemd` manager configuration. This reruns all generators, reloads all unit files, and re-creates the entire dependency tree.
`/etc/NetworkManager/system-connections/`	`nmcli conn reload`	Changes to `NetworkManager` system connections are activated by signaling the `NetworkManager` daemon to reload all connections. For more information, see the *Additional resources* section.

File	Command	Description
`/etc/firewalld/`	`firewall-cmd --reload`	Changes to the permanent configuration of `firewalld` are activated by signaling `firewalld` to reload firewall rules as new runtime configuration.

Related information
Configuring and managing networking

Monitor device resources

You can set up monitors for device resources and define alerts when the use of these resources crosses a defined threshold. When the agent alerts the Red Hat Edge Manager service, the service sets the device status to "degraded" or "error" (depending on the severity level).

Resource monitors take the following parameters:

Parameter	Description
MonitorType	The resource to monitor. Currently supported resources are "CPU", "Memory", and "Disk".
SamplingInterval	The interval in which the monitor samples use, specified as positive integer followed by a time unit ("s" for seconds, "m" for minutes, "h" for hours).
AlertRules	A list of alert rules.
Path	(Disk monitor only) The absolute path to the directory to monitor. Utilization reflects the filesystem containing the path, similar to df, even if it's not a mount point.

Alert rules take the following parameters:

Parameter	Description
Severity	The alert rule's severity level out of "Info", "Warning", or "Critical". Only one alert rule is allowed per severity level and monitor.
Duration	The duration that resource use is measured and averaged over when sampling, specified as positive integer followed by a time unit

Parameter	Description
	("s" for seconds, "m" for minutes, "h" for hours). It must be smaller than the sampling interval.
Percentage	The use threshold that triggers the alert, as percentage value (range 0 to 100 without the "%" sign).
Description	A human-readable description of the alert. This is useful for adding details about the alert that might help with debugging. By default it populates the alert as : load is above >% for more than.

Monitor device resources on the CLI

Monitor the resource use of your Red Hat Edge Manager devices on the CLI by configuring detailed monitors and threshold-based alert rules. This enables automatic status reporting, helping you keep stability and troubleshoot performance issues.

- Add resource monitors in the `resources:` section of the device's specification.

  For example, add the following monitor for your disk:

```
 apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: <device_name>
spec:
[...]
  resources:
  - monitorType: Disk
    samplingInterval: 5s
    path: /application_data
    alertRules:
    - severity: Warning
      duration: 30m
      percentage: 75
      description: Disk space for application data is >75% full for over 30m.
    - severity: Critical
      duration: 10m
      percentage: 90
      description: Disk space for application data is >90% full over 10m.
[...]
```

samplingInterval

Samples usage every 5 seconds.

path

Checks disk usage on the file system that is associated with the `/applications_data` path.

alertRules[severity: Warning]

Initiates a warning if the average usage exceeds 75% for more than 30 minutes.

alertRules[severity: Critical]

Initiates a critical alert if the average usage exceeds 90% for over 10 minutes.

Manage applications on an edge device

Modify the application list in the device specification to deploy, update, or remove applications. The Red Hat Edge Manager agent detects changes upon check-in, downloads new or updated Open Container Initiative (OCI) packages and images, and manages their deployment or removal at runtime.

The Red Hat Edge Manager supports the `podman-compose` tool as the application runtime and format.

Related information
Building a *bootc* operating system image for use with the Red Hat Edge Manager

Build an application package image

The Red Hat Edge Manager can download application packages from an Open Container Initiative (OCI) compatible registry. You can build an OCI container image that includes your application package in the `podman-compose` format and push the image to your OCI registry.

Before you begin

- You must install the Red Hat Edge Manager CLI.
- You must log in to the Red Hat Edge Manager service.
- Your device must run an operating system image with the `podman-compose` tool installed.

Procedure

1. Define the functionality of the application in a file called `podman-compose.yaml` that follows the Podman Compose specification:
    - Create a file called `Containerfile` with the following content:

```
FROM scratch
COPY podman-compose.yaml /podman-compose.yaml
LABEL appType="compose"
```

FROM scratch

Embeds the compose file in a `scratch` container.

LABEL appType="compose"

Adds the `appType=compose` label.

2. Build and push the container image to your OCI registry:

a. Define the image repository that you have permissions to write to by running the following command:

```
OCI_IMAGE_REPO=quai.io/<your_org>/<your_image>
```

b. Define the image tag by running the following command:

```
OCI_IMAGE_TAG=v1
```

c. Build the application container image by running the following command:

```
podman build -t ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG} .
```

d. Push the container image by running the following command:

```
podman push ${OCI_IMAGE_REPO}:${OCI_IMAGE_TAG} .
```

Specify applications inline in the device specification

Application manifests are specified inline in a device's specification, so you do not need to build an OCI registry application package.

The inline application provider accepts a list of application content with the following parameters:

Parameter	Description
Path	The relative path to the file on the device. Note that any existing file is overwritten.
Content (Optional)	The plain text (UTF-8) or base64-encoded content of the file.
ContentEncoding	How the contents are encoded. Must be either "plain" or "base64". Defaults to "plain".

Example

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: some_device_name
spec:
[...]
  applications:
    - name: my-app
      appType: compose
      inline:
        - content: |
            version: "3.8"
            services:
              service1:
                image:  quay.io/flightctl-tests/alpine:v1
                command: ["sleep", "infinity"]
          path: podman-compose.yaml
[...]
```

NOTE:
Inline compose applications can have two paths at most. You must name the first one `podman-compose.yaml`, and the second (override) `podman-compose.override.yaml`.

Deploy applications to a device using the CLI

Deploy application packages securely from an OCI registry onto a target device by using the Red Hat Edge Manager CLI. Specifying the container image reference in the device manifest automatically triggers the transactional deployment through the device agent.

Procedure

1. Specify the application package that you want to deploy in the `spec.applications` field in the `Device` resource:

```
apiVersion: flightctl.io/v1alpha1
kind: Device
metadata:
  name: <device_name>
spec:
[...]
  applications:
  - name: wordpress
    image: quay.io/rhem-demos/wordpress-app:latest
    envVars:
      WORDPRESS_DB_HOST: <database_host>
      WORDPRESS_DB_USER: <user_name>
      WORDPRESS_DB_PASSWORD: <password>
[...]
```

name

A user-defined name for the application that is used when the web console and the CLI list applications.

image

A reference to an application package in an OCI registry.

envVars

Optional. A list of key-value pairs that are passed to the deployment tool as environment variables or command line flags.

> **NOTE:**
> For each application in the `applications` section of the device specification, you can find the corresponding device status information.

2. Verify the status of an application deployment on a device by inspecting the device status information by running the following command:

```
flightctl get device/<your_device_id> -o yaml
```

See the following example output:

```
[...]
spec:
  applications:
  - name: example-app
    image: quay.io/flightctl-demos/example-app:v1
status:
  applications:
  - name: example-app
    ready: 3/3
    restarts: 0
    status: Running
  applicationsSummary:
    info: All application workloads are healthy.
    status: Healthy
[...]
```

Manage a large number of devices with device fleets

The Red Hat Edge Manager simplifies the management of a large number of devices and workloads through *device fleets*. A fleet is a resource that defines a group of devices governed by a common device template and management policies.

When you make a change to the device template, all devices in the fleet receive the changes when the Red Hat Edge Manager agent detects the new target specification.

Device monitoring in a fleet is also simplified because you can check the status summary of the whole fleet.

Fleet-level management offers the following advantages:

- Scales your operations because you perform operations only once for each fleet instead of once for each device.
- Minimizes the risk of configuration mistakes and configuration drift.
- Automatically applies the target configuration when you add devices to the fleet or replace devices in the fleet. The fleet specification consists of the following features:

  Label selector

  Determines which devices are part of the fleet.

Device template

Defines the configuration that the Red Hat Edge Manager enforces on devices in the fleet.

Policies

Governs how devices are managed, for example, how changes to the device template are rolled out to the devices.

You can have both individually managed and fleet-managed devices at the same time.

When you select a device into a fleet, the Red Hat Edge Manager creates the device specification for the new device based on the device template. If you update the device template for a fleet or a new device joins the fleet, the Red Hat Edge Manager enforces the new specification in the fleet.

If a device is not selected into any fleets, the device is considered user-managed or unmanaged. For user-managed devices, you must update the device specification either manually or through an external automation.

IMPORTANT:
A device cannot be a member of more than one fleet at the same time.

Related information
Labels and label selectors

Device selection into a fleet

By default, devices are not assigned to a fleet. Instead, each fleet uses a selector that defines which labels a device must have to be added to the fleet.

To understand how to use labels in a fleet, see the following example.

The following list shows point-of-sales terminal devices and their labels:

Device	Labels
A	`type: pos-terminal`, `region: east`, `stage: production`
B	`type: pos-terminal`, `region: east`, `stage: development`
C	`type: pos-terminal`, `region: west`, `stage: production`
D	`type: pos-terminal`, `region: west`, `stage: development`

If all point-of-sale terminals use the same configuration and are managed by the same operations team, you can define a single fleet called `pos-terminals` with the `type=pos-terminal` label selector. Then, the fleet contains devices A, B, C, and D.

However, you might want to create separate fleets for the different organizations for development or production. You can define a fleet for development with the `type=pos-terminal, stage=development` label selector, which selects devices C and D. Then, you can define another fleet for production with the `type=pos-terminal, stage=production` label selector. By using the correct label selectors, you can manage both fleets independently.

IMPORTANT:
You must define selectors in a way that two fleets do not select the same device. For example, if one fleet selects `region=east`, and another fleet selects `stage=production`, both fleets try to select device A. If two fleets try to select the same device, the Red Hat Edge Manager keeps the device in the currently assigned fleet, if any, and sets the `OverlappingSelectors` condition on the affected fleets to `true`.

Device templates

A device template of a fleet has a device specification that is applied to all devices in the fleet when the template is updated.

For example, you can specify in the device template of a fleet that all devices in the fleet must run the `quay.io/flightctl/rhel:9.5` operating system image.

The Red Hat Edge Manager service then rolls out the target specification to all devices in the fleet, and the Red Hat Edge Manager agents update each device.

You can change other specification items in the device template and the Red Hat Edge Manager applies the changes in the same way.

However, sometimes not all of the devices in the fleet need to have the exact same specification. The Red Hat Edge Manager allows templates to contain placeholders that are populated based on the device name or label values.

The syntax of the placeholders matches that of Go templates. However, you can only use simple text and actions.

The use of conditionals or loops in the placeholders is not supported.

You can reference anything from the metadata of a device, such as `{{ .metadata.labels.key }}` or `{{ .metadata.name }}`.

You can also use the following functions in your placeholders:

- The `upper` function changes the value to uppercase. For example, the function is `{{ upper .metadata.name }}`.
- The `lower` function changes the value to lowercase. For example, the function is `{{ lower .metadata.labels.key }}`.
- The `replace` function replaces all occurrences of a substring with another string. For example, the function is `{{ replace "old" "new" .metadata.labels.key }}`.

- The `getOrDefault` function returns a default value if accessing a missing label. For example, the function is `{{ getOrDefault .metadata.labels "key" "default" }}`. You can combine the functions in pipelines, for example, a combined function is `{{ getOrDefault .metadata.labels "key" "default" | upper | replace " " "-" }}`.

NOTE:
Ensure you are using proper Go template syntax. For example, `{{ .metadata.labels.target-revision }}` is not valid because of the hyphen. Instead, you must refer to the field as `{{ index .metadata.labels "target-revision" }}`.

You can use the placeholders in device templates in the following ways:

- You can label devices by deployment stage, for example, stage labels are `stage: testing` and `stage: production`. Then, you can use the label with the `stage` key as placeholder when referencing the operating system image to use, for example, use `quay.io/myorg/myimage:latest-{{ .metadata.labels.stage }}` or when referencing a configuration folder in a Git repository.
- You can label devices by deployment site, for example, deployment sites are `site: factory-berlin` and `site: factory-madrid`.
- Then, you can use the label with the `site` key as parameter when referencing the secret with network access credentials in Kubernetes. The following fields in device templates support placeholders:

Field	Placeholders supported in
Operating System Image	repository name, image name, image tag
Git Config Provider	target revision, path
HTTP Config Provider	URL suffix, path
Inline Config Provider	content, path

Add devices to a fleet on the web UI

Define the label selector for a device fleet by using the Red Hat Edge Manager web UI to automatically include devices that match your specified criteria. This streamlines fleet management by applying a common device template and ensuring consistent policies across all enrolled devices.

Procedure

1. From the navigation panel, select **Application Links › Edge Manager**. This opens the external Edge Manager instance.
2. From the navigation panel, select **Fleets**. Select the fleet that you want to add devices to.

3. Click **Actions** and select **Edit fleet**.
4. In the **General info** tab, click **Add label** under the **Device selector** option.
5. Add the label to select devices for your fleet. Any devices with that label are added to the fleet.

Add devices to a fleet on the CLI

Use the Red Hat Edge Manager CLI to define the label selectors for a fleet resource, automatically enrolling devices that match the specified criteria. This streamlines fleet management by enabling consistent configuration and policy enforcement across a defined group of devices.

Procedure

1. Run the following command to verify that the label selector returns the devices that you want to add to the fleet:

```
flightctl get devices -l type=pos-terminal -l stage=development
```

2. If running the command returns the expected list of devices, you can define a fleet that selects the devices by using the following YAML file:

```
apiVersion: flightctl.io/v1alpha1
kind: Fleet
metadata:
  name: my_fleet
spec:
  selector:
    matchLabels:
      type: pos-terminal
      stage: development
[...]
```

3. Apply the change by running the following command:

```
flightctl apply -f my_fleet.yaml
```

4. Check for any overlaps with the selector of other fleets by running the following command:

```
flightctl get fleets/my_fleet -o json | jq -r '.status.conditions[] |
select(.type=="OverlappingSelectors").status'
```

See the following example output:

```
False
```

Rollout device selection

When performing a rollout by using `flightctl`, you must manage which devices participate in the rollout and how much disruption is acceptable. The device selection process and the rollout disruption budget concept ensure controlled and predictable rollouts.

The process and configuration for selecting devices during a rollout includes targeting strategies, batch sequencing, and success criteria for controlled software deployment.

Device targeting

A rollout applies only to devices that belong to a fleet. Each device can belong to only a single fleet. Since rollout definitions are done at the fleet level, the selection process determines which devices within a fleet that participate in a batch rollout based on label criteria.

After processing all batches, all fleet devices are rolled out.

- **Labels**: Devices with specific metadata labels can be targeted for rollouts.
- **Fleet membership**: Rollouts apply only to devices within the specified fleet.

Device selection strategy

The Red Hat Edge Manager supports only the `BatchSequence` strategy for device selection. This strategy defines a stepwise rollout process where devices are added in batches based on specific criteria. Batches are executed sequentially.

After each batch completes, execution proceeds to the next batch only if the success rate of the previous batch meets or exceeds the configured success threshold.

The success rate is determined as:

```
# of successful rollouts in the batch / # of devices in the batch >= success
threshold
```

In a batch sequence, the final batch is an implicit batch and it is not specified in the batch sequence. It selects all devices in a fleet that have not been selected by the explicit batches in the sequence.

Limit in device selection

Each batch in the `BatchSequence` strategy might use an optional `limit` parameter to define how many devices should be included in the batch. You can specify the limit can in two ways:

- **Absolute number**: A fixed number of devices to be selected.
- **Percentage**: The percentage of the total matching device population to be selected.
    - If you provide a `selector` with labels, the percentage is calculated based on the number of devices that match the label criteria within the fleet.
    - If you do not provide a `selector`, the percentage is applied to all devices in the fleet.

Success threshold

The `successThreshold` defines the percentage of successfully updated devices required to continue the rollout. If the success rate falls below this threshold, the rollout might be paused to prevent further failures.

Example

The following shows an example YAML configuration for a fleet specification:

```
apiVersion: v1alpha1
kind: Fleet
metadata:
  name: default
spec:
  selector:
    matchLabels:
      fleet: default
  rolloutPolicy:
    deviceSelection:
      strategy: 'BatchSequence'
      sequence:
        - selector:
            matchLabels:
              site: madrid
          limit: 1  # Absolute number
        - selector:
            matchLabels:
              site: madrid
          limit: 80%  # Percentage of devices matching the label criteria within the
fleet
        - limit: 50%  # Percentage of all devices in the fleet
        - selector:
            matchLabels:
              site: paris
        - limit: 80%
        - limit: 100%
    successThreshold: 95%
```

In this example, there are 6 explicit batches and 1 implicit batch:

- The first batch selects 1 device having a label **site:madrid**.
- With the second batch 80% of all devices having the label **site:madrid** are either selected for rollout in the current batch or were previously selected for rollout.
- With the third batch 50% of all devices are either selected for rollout in the current batch or were previously selected for rollout.
- With the fourth batch all devices that were not previously selected and have the label **site:paris** are selected.

- With the fifth batch 80% of all devices are either selected for rollout in the current batch or were previously selected for rollout.
- With the sixth batch 100% of all devices are either selected for rollout in the current batch or were previously selected for rollout.
- The last implicit batch selects all devices that have not been selected in any previous batch (might be none).

Define a rollout disruption budget

A rollout disruption budget defines the acceptable level of service impact during a rollout. This ensures that a deployment does not take down too many devices at once, maintaining overall system stability.

Disruption budget parameters

Configure rollout disruption parameters, such as grouping criteria (`groupBy`) and availability limits (`minAvailable` , `maxUnavailable`), to control the maximum acceptable service impact during fleet updates and keep overall system stability.

- `groupBy` : Defines how devices are grouped when applying the disruption budget. The grouping is done by label keys.
- `minAvailable` : Specifies the minimum number of devices that must remain available during a rollout.
- `maxUnavailable` : Limits the number of devices that can be unavailable at the same time.

Example

The following shows an example YAML configuration for a fleet specification:

```
apiVersion: v1alpha1
kind: Fleet
metadata:
  name: default
spec:
  selector:
    matchLabels:
      fleet: default
  rolloutPolicy:
    disruptionBudget:
      groupBy: ['site', 'function']
      minAvailable: 1
      maxUnavailable: 10
```

In this example, the grouping is performed on 2 label keys: **site** and **function**. A group for disruption budget consists of all devices in a fleet having the same label values for the preceding label keys. For every such group the conditions defined in this specification are continuously enforced.

Red Hat product documentation legal notices

Copyright © Red Hat

Except as otherwise noted below, the text of and illustrations in this documentation are licensed by Red Hat under the Creative Commons Attribution–Share Alike 3.0 Unported license. If you distribute this document or an adaptation of it, you must provide the URL for the original version. Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

XFS is a trademark or registered trademark of Hewlett Packard Enterprise Development LP or its subsidiaries in the United States and other countries.

The OpenStack® Word Mark and OpenStack logo are trademarks or registered trademarks of the Linux Foundation, used under license.

All other trademarks are the property of their respective owners.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. link:https://fsf.org/. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If

such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions. "This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code. The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component,

or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions. All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law. No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies. You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions. You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so. A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms. You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms. "Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination. You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies. You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients. Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents. A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom. If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License. Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License. The Free Software Foundation may publish revised and/ or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES

SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16. If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see link:https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read link:https://www.gnu.org/licenses/why-not-lgpl.html.

Apache license

Version 2.0, January 2004

http://www.apache.org/licenses/

Terms and Conditions for use, reproduction, and distribution

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **"control"** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or **"Your"**) shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **"submitted"** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **"Not a Contribution."**

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a **"NOTICE"** text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications,

or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2026. All rights reserved.

www.redhat.com