Red Hat

Migrate

Ansible Automation Platform 2.6

May 12, 2026

Contents

1 Migrate

Migrate from existing deployment topologies

Learn about supported migration paths between RPM-based, container-based, OpenShift Container Platform, and Managed Ansible Automation Platform deployments, including step-by-step workflows and migration requirements.

Migration between different Ansible Automation Platform deployment types for Ansible Automation Platform 2.6 requires specific steps and considerations.

The supported migration paths include:

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

Migrations outside of those listed are not supported at this time.

WARNING: To upgrade to Ansible Automation Platform 2.7, you must be running a containerized or OpenShift Container Platform deployment. RPM-based deployments are not supported as an upgrade path to 2.7. If you are running an RPM-based deployment, migrate to a containerized or OpenShift Container Platform deployment before you upgrade.

Supported migration workflows:

- Document all components and configurations that require migration between Ansible Automation Platform platforms
- Provide step-by-step migration workflows for different deployment scenarios
- Identify potential challenges and unknowns that require further investigation

Out of scope

Understand which Ansible Automation Platform components and configurations require manual re-creation in the target environment and are not covered by the migration process.

Migration covers core Ansible Automation Platform components. Some components and configurations are out of scope and require manual re-creation in the target environment:

- Event-Driven Ansible: Manually recreate configuration and content for Event-Driven Ansible in the target environment.
- Instance groups: Manually recreate instance group configurations after migration.
- Hub content: Manually re-import or reconfigure content hosted in automation hub.
- Custom Certificate Authority (CA) for receptor mesh: Manually reconfigure custom CA configurations for receptor mesh.
- Disconnected environments: The migration process does not cover disconnected environments.
- Execution environments (other than the default one): Manually rebuild or re-import custom execution environments.

Manually re-create, import, or configure these items in the target environment.

Migration process overview

Understand the complete migration workflow including preparation, export, artifact creation, import, reconciliation, and validation steps for moving between Ansible Automation Platform installation types.

IMPORTANT:
You can only migrate to a different installation type of the same Ansible Automation Platform version. For example, you can migrate from RPM version 2.6 to containerized 2.6, but not from RPM version 2.4 to containerized 2.6.

You can only migrate to a different installation type of the same Ansible Automation Platform version. For example, you can migrate from containerized 2.7 to OpenShift Container Platform 2.7, but not from containerized 2.6 to OpenShift Container Platform 2.7.

WARNING: If you are running an RPM-based deployment, complete your migration to a containerized or OpenShift Container Platform deployment before upgrading to Ansible Automation Platform 2.7. RPM-based deployments are not supported as an upgrade path to 2.7.

The migration between Ansible Automation Platform installation types follows this general workflow:

1. Prepare and assess the source environment
2. Export the source environment
3. Create and verify the migration artifact
4. Prepare and assess the target environment
5. Import the migration content to the target environment
6. Reconcile the target environment post-import
7. Validate the target environment

Migration prerequisites

Prerequisites for migrating your Ansible Automation Platform deployment. For your specific migration path, ensure that you meet all necessary conditions before proceeding.

WARNING: To upgrade to Ansible Automation Platform 2.7, you must first migrate from your RPM-based deployment to a containerized or OpenShift Container Platform deployment. RPM-based deployments are not supported as an upgrade path to 2.7.

RPM to containerized migration prerequisites

Before migrating from an RPM-based deployment to a container-based deployment, ensure you meet the following prerequisites:

NOTE: Completing this migration is a required step if you plan to upgrade to Ansible Automation Platform 2.7. RPM-based deployments are not supported as an upgrade path to 2.7.

- You have a source RPM-based deployment of Ansible Automation Platform.
- The source RPM-based deployment is on the latest async release of the version you are on.
- You have a target environment prepared for a container-based deployment of Ansible Automation Platform.
- You have downloaded the containerized installation program for the latest release of the Ansible Automation Platform version you are on.
- You have enough storage for database dumps and backups.
- There is network connectivity between the source and target environments.

RPM to OpenShift Container Platform migration prerequisites

Before migrating from an RPM-based deployment to an OpenShift Container Platform deployment, ensure you meet the following prerequisites:

NOTE: Completing this migration is a required step if you plan to upgrade to Ansible Automation Platform 2.7. RPM-based deployments are not supported as an upgrade path to 2.7.

- You have a source RPM-based deployment of Ansible Automation Platform.
- The source RPM-based deployment is on the latest async release of the version you are on.
- You have a target OpenShift Container Platform environment ready.
- You have Ansible Automation Platform Operator available for the latest release of the Ansible Automation Platform version you are on.
- You have made a decision on internal or external database configuration.
- You have made a decision on internal or external Redis configuration.
- There is network connectivity between the source and target environments.

RPM to Managed Ansible Automation Platform migration prerequisites

Before migrating from an RPM-based deployment to a Managed Ansible Automation Platform deployment, ensure you meet the following prerequisites:

NOTE: Completing this migration is a required step if you plan to upgrade to Ansible Automation Platform 2.7. RPM-based deployments are not supported as an upgrade path to 2.7.

- You have a source RPM-based deployment of Ansible Automation Platform.
- The source deployment is on the latest release of the Ansible Automation Platform version you are on.
- You have a target Managed Ansible Automation Platform deployment.
- You have enabled local authentication on the source deployment before the migration.

- A local administrator account must be functional on the source deployment before migration. Verify this by performing a successful login to the source deployment.
- You have a plan to retain a backup throughout the migration process and to ensure that your existing Ansible Automation Platform deployment remains active until your migration has completed successfully.
- You have a plan for any environment changes based on the migration from a self-hosted Ansible Automation Platform deployment to a Managed Ansible Automation Platform deployment:
    - Job log retention changes from a customer-configured option to 30 days.
    - Network changes occur when moving the control plane to the managed service.
    - Automation mesh requires reconfiguration.
- You must reconfigure or re-create Single Sign-On (SSO) identity providers post-migration to account for URL changes.

Containerized to OpenShift Container Platform migration prerequisites

Before migrating from a container-based deployment to an OpenShift Container Platform deployment, ensure that you meet the following prerequisites:

- You have a source container-based deployment of Ansible Automation Platform.
- The source deployment is on the latest async release of the version you are on.
- You have a target OpenShift Container Platform environment ready.
- You have an Ansible Automation Platform Operator available for the latest release of the Ansible Automation Platform version you are on.
- You have decided between internal or external database configuration.
- You have decided between internal or external Redis configuration.
- There is network connectivity between the source and target environments.

Containerized to Managed Ansible Automation Platform migration prerequisites

Before migrating from a container-based deployment to a Managed Ansible Automation Platform deployment, ensure that you meet the following prerequisites:

- You have a source container-based deployment of Ansible Automation Platform.
- The source deployment is on the latest release of the Ansible Automation Platform version you are on.
- You have a target Managed Ansible Automation Platform deployment.
- You have enabled local authentication on the source deployment before the migration.
- A local administrator account must be functional on the source deployment before migration. Verify this by performing a successful login to the source deployment.
- You have a plan to retain a backup throughout the migration process and to ensure that your existing Ansible Automation Platform deployment remains active until your migration has completed successfully.
- You have a plan for any environment changes based on the migration from a self-hosted Ansible Automation Platform deployment to a Managed Ansible Automation Platform deployment:
    - Job log retention changes from a customer-configured option to 30 days.
    - Network changes occur when moving the control plane to the managed service.
    - Automation mesh requires reconfiguration.
- You must reconfigure or re-create Single Sign-On (SSO) identity providers post-migration to account for URL changes.

Contents of the migration artifact

The migration artifact packages all necessary data and configurations from your source environment. Verify its structure and contents to ensure a successful migration.

Artifact structure

The migration artifact is a comprehensive package containing all necessary components to transfer your Ansible Automation Platform deployment.

Structure the artifact as follows:

```
/
  manifest.yml
  secrets.yml
  sha256sum.txt

  -> controller:
     controller.pgc
     -> custom_configs:
        foo.py
        bar.py
  -> gateway:
     gateway.pgc
  -> hub:
     hub.pgc
```

Manifest file

The `manifest.yml` file serves as the primary metadata document for the migration artifact. It contains critical versioning and component information from your source environment.

Structure the manifest as follows:

```
---
aap_version: X.Y # The version being migrated
platform: rpm # The source platform type
components:
  - name: controller
    version: x.y.z
  - name: hub
    version: x.y.z
  - name: gateway
    version: x.y.z
```

Secrets file

The `secrets.yml` file in the migration artifact includes essential Django `SECRET_KEY` values required for authentication between services.

Structure the secrets file as follows:

```
controller_pg_database: <redacted>
controller_secret_key: <redacted>
gateway_pg_database: <redacted>
gateway_secret_key: <redacted>
hub_pg_database: <redacted>
hub_secret_key: <redacted>
hub_db_fields_encryption_key: <redacted>
```

NOTE:
Ensure the `secrets.yml` file is encrypted and kept in a secure location.

Migration artifact creation checklist

Use this checklist to verify the migration artifact.

- Database dumps: Include complete database dumps for each component.
    - Ensure the automation controller database (`controller.pgc`) is present in the artifact.

    - Ensure the automation hub database (`hub.pgc`) is present in the artifact.
    - Ensure the platform gateway database (`gateway.pgc`) is present in the artifact.
- Secret dumps: Export and include all security-related information.
    - Validate that all secret values are present in the `secrets.yml` file.
- Custom configurations: Package all customizations from the source environment.
    - Validate that any custom Python scripts or modules (for example `foo.py` , `bar.py`) are present on the artifact.
    - Document any non-standard configurations or environment-specific settings.
- Database information: Document database details.
    - Include the database names for all components.
    - Document database users and required permissions.
    - Note any database-specific configurations or optimizations.
- Verification: Ensure artifact integrity and completeness.
    - Verify that all required files are included in the artifact.
    - Verify that checksums exist for all included database files.
    - Test the artifact's structure and accessibility.
    - Consider encrypting the artifact for secure transfer to the target environment.
    - Document any known limitations or special considerations.

Prepare and export data from the source environment

Prepare and export data from your existing Ansible Automation Platform deployment. The exported data forms a critical migration artifact, which you use to configure your new environment.

Prepare and export data from an RPM-based environment

Prepare and export data from your RPM-based Ansible Automation Platform deployment.

Prepare and assess the source environment

Before beginning your migration, document your current RPM deployment to use as a reference throughout the migration process and when configuring your target environment.

Procedure

1. Document the full topology of your current RPM deployment:
    a. Map out all servers, nodes, and their roles (for example control nodes, execution nodes, database servers).
    b. Note the hostname, IP address, and function of each server in your deployment.
    c. Document the network configuration between components.
2. Ansible Automation Platform version information:
    a. Record the exact Ansible Automation Platform version (X.Y) currently deployed.
3. Document the specific version of each component:
    a. Automation controller version
    b. Automation hub version
    c. Platform gateway version
4. Database configuration:
    a. Database names for each component
    b. Database users and roles
    c. Connection parameters and authentication methods
    d. Any custom PostgreSQL configurations or optimizations

Export the source environment

From your source environment, export the data and configurations needed for migration.

Procedure

1. Verify the PostgreSQL database version is PostgreSQL version 15.

    You can verify your current PostgreSQL version by connecting to your database server and running the following command as the `postgres` user:

```
$ psql -c 'SELECT version();'
```

IMPORTANT:
PostgreSQL version 15 is a strict requirement for the migration process to succeed. If running PostgreSQL 13 or earlier, upgrade to version 15 before proceeding with the migration.

If using an Ansible Automation Platform managed database, re-run the installation program to upgrade the PostgreSQL version. If using a customer provided (external) database, contact your database administrator or service provider to confirm the version and arrange for an upgrade if required.

2. Create a complete backup of the source environment:

```
$ ./setup.sh -e 'backup_dest=/path/to/backup_dir/' -b
```

3. Get the connection settings from one node from each of the component groups.

   For each command, access the host and become the `root` user.

   - Access the automation controller node and run:

     ```
     # awx-manage print_settings | grep '^DATABASES'
     ```

   - Access the automation hub node and run:

     ```
     # grep '^DATABASES' /etc/pulp/settings.py
     ```

   - Access the platform gateway node and run:

     ```
     # aap-gateway-manage print_settings | grep '^DATABASES'
     ```

4. Stage the manually created artifact on the platform gateway node.

```
# mkdir -p /tmp/backups/artifact/{controller,gateway,hub}
```

```
# mkdir -p /tmp/backups/artifact/controller/custom_configs
```

```
# touch /tmp/backups/artifact/secrets.yml
```

```
# cd /tmp/backups/artifact/
```

5. Validate the database size and make sure you have enough space on the filesystem for the `pg_dump`.

   You can verify the database sizes by connecting to your database server and running the following command as the `postgres` user:

   ```
   $ psql -c '\l+'
   ```

   Adjust the filesystem size or mount an external filesystem as needed before performing the next step.

   NOTE:
   These commands send all target files to the `/tmp` filesystem. Adjust the commands to match your environment's needs.

6. Perform database dumps of all components on the platform gateway node within the artifact you created.

   ```
   # psql -h <pg_hostname> -U <component_pg_user> -d <database_name> -t -c 'SHOW server_version;' # ensure connectivity to the database
   ```

   ```
   # pg_dump -h <pg_hostname> -U <component_pg_user> -d <component_pg_name> --clean --create -Fc -f <component>/<component>.pgc
   ```

   ```
   # ls -ld <component>/<component>.pgc
   ```

   ```
   # echo "<component>_pg_database: <database_name>" >> secrets.yml ## Add the database name for the component to the secrets file
   ```

7. Export secrets from the RPM environment from one node of each component group.

   For each of the following steps, use the `root` user to run the commands.

   - Access the automation controller node, gather the secret key, and add it to the `controller_secret_key` value in the `secrets.yml` file.

     ```
     # cat /etc/tower/SECRET_KEY
     ```

   - Access the automation hub node, gather the secret key, and add it to the `hub_secret_key` value in the `secrets.yml` file.

```
# grep '^SECRET_KEY' /etc/pulp/settings.py | awk -F'=' '{ print $2 }'
```

- Access the automation hub node, gather the `database_fields.symmetric.key` value, and add it to the `hub_db_fields_encryption_key` value in the `secrets.yml` file.

```
# cat /etc/pulp/certs/database_fields.symmetric.key
```

- Access the platform gateway node, gather the secret key, and add it to the `gateway_secret_key` value in the `secrets.yml` file.

```
# cat /etc/ansible-automation-platform/gateway/SECRET_KEY
```

8. Export automation controller custom configurations.

   If any custom settings exist on the `/etc/tower/conf.d`, copy them to `/tmp/backups/artifact/controller/custom_configs`.

   Configuration files on automation controller that are managed by the installation program and not considered custom:

   - `/etc/tower/conf.d/postgres.py`
   - `/etc/tower/conf.d/channels.py`
   - `/etc/tower/conf.d/caching.py`
   - `/etc/tower/conf.d/cluster_host_id.py`

9. Package the artifact.

```
# cd /tmp/backups/artifact/
```

```
# [ -f sha256sum.txt ] && rm -f sha256sum.txt; find . -type f -name "*.pgc"
-exec sha256sum {} \; >> sha256sum.txt
```

```
# cat sha256sum.txt
```

```
# cd ..
```

```
# tar cf artifact.tar artifact
```

```
# sha256sum artifact.tar > artifact.tar.sha256
```

```
# sha256sum --check artifact.tar.sha256
```

```
# tar tvf artifact.tar
```

Example output of `tar tvf artifact.tar`:

```
drwxr-xr-x ansible/ansible     0 2025-05-08 16:48 artifact/
drwxr-xr-x ansible/ansible     0 2025-05-08 16:33 artifact/controller/
-rw-r--r-- ansible/ansible 732615 2025-05-08 16:26 artifact/controller/
controller.pgc
drwxr-xr-x ansible/ansible      0 2025-05-08 16:33 artifact/controller/
custom_configs/
drwxr-xr-x ansible/ansible      0 2025-05-08 16:11 artifact/gateway/
-rw-r--r-- ansible/ansible 231155 2025-05-08 16:28 artifact/gateway/
gateway.pgc
drwxr-xr-x ansible/ansible      0 2025-05-08 16:26 artifact/hub/
-rw-r--r-- ansible/ansible 29252002 2025-05-08 16:26 artifact/hub/hub.pgc
-rw-r--r-- ansible/ansible      614 2025-05-08 16:24 artifact/secrets.yml
-rw-r--r-- ansible/ansible      338 2025-05-08 16:48 artifact/sha256sum.txt
```

10. Download the `artifact.tar` and `artifact.tar.sha256` to your local machine or transfer to the target node with the `scp` command.

Related information
Back up containerized Ansible Automation Platform

Prepare and export data from a container-based environment

Prepare and export data from your container-based Ansible Automation Platform deployment.

Prepare and assess the source environment

Document your current containerized deployment configuration, topology, and components to create a comprehensive reference for migration.

Procedure

1. Document the full topology of your current containerized deployment:
    a. Map out all servers, nodes, and their roles (for example control nodes, execution nodes, database servers).
    b. Note the hostname, IP address, and function of each server in your deployment.
    c. Document the network configuration between components.
2. Ansible Automation Platform version information:
    a. Record the exact Ansible Automation Platform version (X.Y) currently deployed.
3. Document the specific version of each component:
    a. Automation controller version
    b. Automation hub version
    c. Platform gateway version
4. Database configuration:
    a. Database names for each component
    b. Database users and roles
    c. Connection parameters and authentication methods
    d. Any custom PostgreSQL configurations or optimizations
5. Identify all custom configurations and settings
6. Document container resource allocations and volumes

Export the source environment

Export databases, secrets, and custom configurations from your source containerized Ansible Automation Platform deployment to create the migration artifact.

Procedure

1. Create a complete backup of the source environment:

```
$ ansible-playbook -i <path_to_inventory>
ansible.containerized_installer.backup
```

2. Get the connection settings from one node in each of the component groups.
   - Access the automation controller node and run:

     ```
     $ podman exec -it automation-controller-task bash -c 'awx-manage
     print_settings | grep '^DATABASES'
     ```

   - Access the automation hub node and run:

     ```
     $ podman exec -it automation-hub-api bash -c "pulpcore-manager
     diffsettings | grep '^DATABASES'"
     ```

   - Access the platform gateway node and run:

     ```
     $ podman exec -it automation-gateway bash -c "aap-gateway-manage
     print_settings | grep '^DATABASES'"
     ```

3. Validate the database size and make sure you have enough space on the filesystem for the `pg_dump`.

   You can verify the database sizes by connecting to your database server and running the following command as the `postgres` user:

   ```
   $ podman exec -it postgresql bash -c 'psql -c "\l+"'
   ```

   Adjust the filesystem size or mount an external filesystem as needed before performing the next step.

   NOTE:
   These commands send all target files to the `/tmp` filesystem. Adjust the commands to match your environment's needs.

4. Stage the manually created artifact on the platform gateway node.

   ```
   # mkdir -p /tmp/backups/artifact/{controller,gateway,hub}
   ```

   ```
   # mkdir -p /tmp/backups/artifact/controller/custom_configs
   ```

```
# touch /tmp/backups/artifact/secrets.yml
```

```
# cd /tmp/backups/artifact/
```

5. Perform database dumps of all components on the platform gateway node within the artifact created previously.

   To run the `psql` and `pg_restore` commands, you must create a temporary container and run the commands inside of it. This command must be run from the database node.

```
$ podman run -it --rm --name postgresql_restore_temp --network host --volume
~/aap/tls/extracted:/etc/pki/ca-trust/extracted:z --volume ~/aap/postgresql/
server.crt:/var/lib/pgsql/server.crt:ro,z --volume ~/aap/postgresql/
server.key:/var/lib/pgsql/server.key:ro,z --volume /tmp/backups/artifact:/var/
lib/pgsql/backups:ro,z registry.redhat.io/rhel8/postgresql-15:latest bash
```

```
$ podman run -it --rm --name postgresql_restore_temp --network host --volume
~/aap/tls/extracted:/etc/pki/ca-trust/extracted:z --volume ~/aap/postgresql/
server.crt:/var/lib/pgsql/server.crt:ro,z --volume ~/aap/postgresql/
server.key:/var/lib/pgsql/server.key:ro,z --volume /tmp/backups/artifact:/var/
lib/pgsql/backups:ro,z registry.redhat.io/rhel9/postgresql-15:latest bash
```

NOTE:
This command assumes the image `registry.redhat.io/rhel8/postgresql-15:latest`. If you are missing the image, check the available images for the user with `podman images ls`.

This command assumes the image `registry.redhat.io/rhel9/postgresql-15:latest`. If you are missing the image, check the available images for the user with `podman images ls`.

The command above opens a shell inside the container named `postgresql_restore_temp` and has the artifact mounted into `/var/lib/pgsql/backups`. Also, this command is mounting the PostgreSQL certificates to ensure that you can resolve the correct certificates.

```
bash-4.4$ cd /var/lib/pgsql/backups

bash-4.4$ psql -h <pg_hostname> -U <component_pg_user> -d <database_name> -t
-c 'SHOW server_version;' # ensure connectivity to db

bash-4.4$ pg_dump -h <pg_hostname> -U <component_pg_user> -d
<component_pg_name> --clean --create -Fc -f <component>/<component>.pgc

bash-4.4$ ls -ld <component>/<component>.pgc

bash-4.4$ echo "<component>_pg_database: <database_name>" >> secrets.yml ##
Add the DB name for the component to the secrets file
```

After collecting this data, exit from this temporary container.

6. Export the secrets from the containerized environment from one node of each component group.

   For each step below, use the `root` user to run the commands.

   a. Access the automation controller node and gather the secret key and add to the `controller_secret_key` value in `secrets.yaml` file.

   ```
   $ podman secret inspect --showsecret --format "{{.SecretData}}"
   controller_secret_key
   ```

   b. Access the automation hub node and gather the secret key and add to the `hub_secret_key` value in `secrets.yaml` file.

   ```
   $ podman secret inspect --showsecret --format "{{.SecretData}}"
   hub_secret_key
   ```

   c. Access the automation hub node and gather the `database_fields.symmetric.key` value and add to the `hub_db_fields_encryption_key` value in `secrets.yaml` file.

   ```
   $ podman secret inspect --showsecret --format "{{.SecretData}}"
   hub_database_fields
   ```

   d. Access the platform gateway node and gather the secret key and add to the `gateway_secret_key` value in `secrets.yaml` file.

   ```
   $ podman secret inspect --showsecret --format "{{.SecretData}}"
   gateway_secret_key
   ```

7. Export automation controller custom configurations.

   If any `extra_settings` exist in your containerized installation inventory, copy them into a new file and saving them under `/tmp/backups/artifact/controller/custom_configs`.

8. Package the artifact.

```
 # cd /tmp/backups/artifact/

# [ -f sha256sum.txt ] && rm -f sha256sum.txt; find . -type f -name "*.pgc"
-exec sha256sum {} \; >> sha256sum.txt

# cat sha256sum.txt

# cd ..

# tar cf artifact.tar artifact

# sha256sum artifact.tar > artifact.tar.sha256

# sha256sum --check artifact.tar.sha256

# tar tvf artifact.tar
```

Example output of `tar tvf artifact.tar`:

```
 drwxr-xr-x ansible/ansible     0 2025-05-08 16:48 artifact/

drwxr-xr-x ansible/ansible     0 2025-05-08 16:33 artifact/controller/

-rw-r--r-- ansible/ansible 732615 2025-05-08 16:26 artifact/controller/
controller.pgc

drwxr-xr-x ansible/ansible      0 2025-05-08 16:33 artifact/controller/
custom_configs/

drwxr-xr-x ansible/ansible      0 2025-05-08 16:11 artifact/gateway/

-rw-r--r-- ansible/ansible 231155 2025-05-08 16:28 artifact/gateway/
gateway.pgc

drwxr-xr-x ansible/ansible      0 2025-05-08 16:26 artifact/hub/

-rw-r--r-- ansible/ansible 29252002 2025-05-08 16:26 artifact/hub/hub.pgc

-rw-r--r-- ansible/ansible      614 2025-05-08 16:24 artifact/secrets.yml

-rw-r--r-- ansible/ansible      338 2025-05-08 16:48 artifact/sha256sum.txt
```

9. Download the `artifact.tar` and `artifact.tar.sha256` to your local machine or transfer to the target node with the `scp` command.

Related information
Back up containerized Ansible Automation Platform

Prepare, configure, and validate the target environment

Prepare, configure, and validate your target Ansible Automation Platform environment.

Prepare the container-based target environment and import migration content

Prepare and assess your target container-based Ansible Automation Platform environment, and import and reconcile your migrated content.

Prepare and assess the target environment

Transfer the migration artifact, install containerized Ansible Automation Platform, and configure the inventory file to match your source environment topology and database settings.

Procedure

1. Validate the file system home folder size and make sure it has enough space to transfer the artifact.
2. Transfer the artifact to the nodes where you will be working by using `scp` or any preferred file transfer method. It is recommended that you work from the platform gateway node as it has access to most systems. However, if you have access or file system space limitations due to the PostgreSQL dumps, work from the database node instead.
3. Download the latest version of containerized Ansible Automation Platform from the Ansible Automation Platform download page.
4. Validate the artifact checksum.
5. Extract the artifact on the home folder for the user running the containers.

```
$ cd ~
```

```
$ sha256sum --check artifact.tar.sha256
```

```
$ tar xf artifact.tar
```

```
$ cd artifact
```

```
$ sha256sum --check sha256sum.txt
```

6. Generate an inventory file for your containerized deployment.

   Configure the inventory file to match the same topology as the source environment. Configure the component database names and the `secret_key` values from the artifact's `secrets.yml` file.

   You can do this in two ways:

   - Set the extra variables in the inventory file.
   - Use the `secrets.yml` file as an additional variables file when running the installation program.

     a. Option 1: Extra variables in the inventory file

     ```
     $ egrep 'pg_database|_key' inventory
     controller_pg_database=<redacted>
     controller_secret_key=<redacted>
     gateway_pg_database=<redacted>
     gateway_secret_key=<redacted>
     hub_pg_database=<redacted>
     hub_secret_key=<redacted>
     __hub_database_fields=<redacted>
     ```

     NOTE:
     The `__hub_database_fields` value comes from the `hub_db_fields_encryption_key` value in your secret.

     b. Option 2: Additional variables file

     ```
     $ ansible-playbook -i inventory
     ansible.containerized_installer.install -e @~/artifact/secrets.yml
     -e "__hub_database_fields='{{ hub_db_fields_encryption_key }}'"
     ```

7. Install and configure the containerized target environment.
8. Verify PostgreSQL database version is on version 15.
9. Create a backup of the initial containerized environment.

   ```
   $ ansible-playbook -i <path_to_inventory>
   ansible.containerized_installer.backup
   ```

10. Verify the fresh installation functions correctly.

Related information

Import the migration content to the target environment

To import your migration content into the target environment, stop the containerized services, import the database dumps, and then restart the services.

Procedure

1. Stop the containerized services, except the database.
   a. In all nodes, if Performance Co-Pilot is configured, run the following command:

```
$ systemctl --user stop pcp
```

   b. Access the automation controller node and run:

```
$ systemctl --user stop automation-controller-task automation-
controller-web automation-controller-rsyslog

$ systemctl --user stop receptor
```

   c. Access the automation hub node and run:

```
$ systemctl --user stop automation-hub-api automation-hub-content
automation-hub-web automation-hub-worker-1 automation-hub-worker-2
```

   d. Access the Event-Driven Ansible node and run:

```
$ systemctl --user stop automation-eda-scheduler automation-eda-daphne
automation-eda-web automation-eda-api automation-eda-worker-1 automation-
eda-worker-2 automation-eda-activation-worker-1 automation-eda-
activation-worker-2
```

   e. Access the platform gateway node and run:

```
$ systemctl --user stop automation-gateway automation-gateway-proxy
```

f. Access the platform gateway node when using standalone Redis, or all nodes from the Redis group in your inventory file when using clustered Redis, and run:

```
$ systemctl --user stop redis-unix redis-tcp
```

NOTE:
In an enterprise deployment, the components run on different nodes. Run the commands on each component node.

2. Import database dumps to the containerized environment.

a. If you are using an Ansible Automation Platform managed database, you must create a temporary container to run the `psql` and `pg_restore` commands. Run this command from the database node:

```
$ podman run -it --rm --name postgresql_restore_temp --network host --
volume ~/aap/tls/extracted:/etc/pki/ca-trust/extracted:z --volume ~/aap/
postgresql/server.crt:/var/lib/pgsql/server.crt:ro,z --volume ~/aap/
postgresql/server.key:/var/lib/pgsql/server.key:ro,z --volume ~/
artifact:/var/lib/pgsql/backups:ro,z registry.redhat.io/rhel8/
postgresql-15:latest bash
```

NOTE:
The command above opens a shell inside the container named `postgresql_restore_temp` with the artifact mounted at `/var/lib/pgsql/backups`. Additionally, it mounts the PostgreSQL certificates to ensure that you can resolve the correct certificates.

The command assumes the image `registry.redhat.io/rhel8/postgresql-15:latest` is available. If you are missing the image, check the available images for the user with `podman images ls`.

It also assumes that the artifact is located in the current user's home folder. If the artifact is located elsewhere, change the `~/artifact` with the required path.

b. If you are using a customer-provided (external) database, you can run the `psql` and `pg_restore` commands from any node that has these commands installed and that has access to the database. Reach out to your database administrator if you are unsure.

c. From inside the container, access the database and ensure the users have the `CREATEDB` role.

```
bash-4.4$ psql -h <pg_hostname> -U postgres

postgres=# \l

          Name           |     Owner     | Encoding |   Collate   |
 Ctype    | ICU Locale | Locale Provider |   Access privileg

es

-------------------------+---------------+----------+-------------
+-------------+------------+-----------------+-----------------

-----

 automationedacontroller | eda           | UTF8     | en_US.UTF-8 |
en_US.UTF-8 |            | libc            |

 automationhub           | automationhub | UTF8     | en_US.UTF-8 |
en_US.UTF-8 |            | libc            |

 awx                     | awx           | UTF8     | en_US.UTF-8 |
en_US.UTF-8 |            | libc            |

 gateway                 | gateway       | UTF8     | en_US.UTF-8 |
en_US.UTF-8 |            | libc            |

...
```

d. For each component name, add the `CREATEDB` role to the `Owner`. For example:

```
postgres=# ALTER ROLE awx WITH CREATEDB;

postgres=# \q
```

Replace `awx` with the database owner.

e. With the `CREATEDB` in place, access the path where the artifact is mounted, and run the `pg_restore` commands.

```
bash$ cd /var/lib/pgsql/backups

bash$ pg_restore --clean --create --no-owner -h <pg_hostname> -U
<component_pg_user> -d template1 <component>/<component>.pgc
```

f. After the restore, remove the permissions from the user. For example:

```
postgres=# ALTER ROLE awx WITH NOCREATEDB;

postgres=# \q
```

Replace `awx` with each user containing the role.

3. Start the containerized services, except the database.

NOTE:
In an enterprise deployment, the components run on different nodes. Run the commands on each component node.

a. In all nodes, if Performance Co-Pilot is configured, run the following command:

```
$ systemctl --user start pcp
```

b. Access the automation controller node and run:

```
$ systemctl --user start automation-controller-task automation-
controller-web automation-controller-rsyslog

$ systemctl --user start receptor
```

c. Access the automation hub node and run:

```
$ systemctl --user start automation-hub-api automation-hub-content
automation-hub-web automation-hub-worker-1 automation-hub-worker-2
```

d. Access the Event-Driven Ansible node and run:

```
$ systemctl --user start automation-eda-scheduler automation-eda-daphne
automation-eda-web automation-eda-api automation-eda-worker-1 automation-
eda-worker-2  automation-eda-activation-worker-1 automation-eda-
activation-worker-2
```

e. Access the platform gateway node and run:

```
$ systemctl --user start automation-gateway automation-gateway-proxy
```

f. Access the platform gateway node when using standalone Redis, or all nodes from the Redis group in your inventory when using clustered Redis, and run:

```
$ systemctl --user start redis-unix redis-tcp
```

Reconcile the target environment post-import

Perform the following post-import reconciliation steps to verify your target environment functions correctly.

Procedure

1. Deprovision the platform gateway configuration.
   - To deprovision platform gateway configuration, SSH to the host serving an `automation-gateway` container as the same rootless user from 4.2.6 and run the following to remove the platform gateway proxy configuration:

```
 $ podman exec -it automation-gateway bash

$ aap-gateway-manage migrate

$ aap-gateway-manage shell_plus

>>> HTTPPort.objects.all().delete(); ServiceNode.objects.all().delete();
ServiceCluster.objects.all().delete()
```

2. Transfer custom configurations and settings.
   - Edit the inventory file and apply any relevant `extra_settings` to each component by using the `component_extra_settings`.
3. Remove all resource server key secrets to be repopulated by the installer:

```
 $ for i in `podman secret ls | egrep 'resource_server' | awk '{print $2}'`; do
podman secret rm $i; done
```

4. Re-run the installation program on the target environment by using the same inventory from the installation.
5. Sync platform gateway resources if Event-Driven Ansible is present:

```
$ podman exec -it automation-eda-api bash
```

```
$ aap-eda-manage resource_sync
```

6. Validate instances for automation execution.

a. SSH to the host serving an `automation-controller-task` container as the rootless user, and run the following commands to validate and remove instances that are orphaned from the source artifact:

```
$ podman exec -it automation-controller-task bash
```

```
$ awx-manage list_instances
```

b. Find nodes that are no longer part of this cluster. A good indicator is nodes with 0 capacity as they have failed their health checks:

```
[ungrouped capacity=0]

    [DISABLED] node1.example.org capacity=0 node_type=hybrid
version=X.Y.Z heartbeat="..."

    [DISABLED] node2.example.org capacity=0 node_type=execution
version=ansible-runner-X.Y.Z heartbeat="..."
```

c. Remove those nodes with `awx-manage`, leaving only the `aap-controller-task` instance:

```
awx-manage deprovision_instance --hostname=node1.example.org

awx-manage deprovision_instance --hostname=node2.example.org
```

7. Repair orphaned automation hub content links for Pulp.

   - Run the following command from any host that has direct access to the automation hub address:

```
$ curl -d '{\"verify_checksums\": true }' -X POST -k https://<gateway
url>/api/galaxy/pulp/api/v3/repair/ -u
<gateway_admin_user>:<gateway_admin_password>
```

8. Reconcile instance groups configuration:
   a. Go to **Automation Execution > Infrastructure > Instance Groups**.
   b. Select the **Instance Group** and then select the **Instances** tab.
   c. Associate or disassociate instances as required.
9. Reconcile decision environments and credentials:
   a. Go to **Automation Decisions > Decision Environments**.
   b. Edit each decision environment which references a registry URL either unrelated or no longer accessible to this new environment. For example, the automation hub decision environment might require modification for the target automation hub environment.

c. Select each associated credential to these decision environments and ensure their addresses align with the new environment.

10. Reconcile execution environments and credentials:
    a. Go to **Automation Execution > Infrastructure > Execution Environments**.
    b. Check each execution environment image and verify their addresses against the new environment.
    c. Go to **Automation Execution > Infrastructure > Credentials**.
    d. Edit each credential and ensure that all environment specific information aligns with the new environment.
11. Verify any further customizations or configurations after the migration, such as RBAC rules with instance groups.

Validate the target environment

After completing the migration, validate that all components in your target environment function correctly.

Procedure

1. Verify all migrated components function correctly.
    a. Platform gateway: Access the Ansible Automation Platform URL at `https://<gateway_hostname>/` and verify that the dashboard loads correctly. Check that the platform gateway service is running and connected to automation controller.
    b. Automation controller: Under **Automation Execution**, check that projects, inventories, and job templates are present and configured.
    c. Automation hub: Under **Automation Content**, verify that collections, namespaces, and their contents are visible.
    d. Event-Driven Ansible (if applicable): Under **Automation Execution Decisions**, verify that rule audits, rulebook activations, and projects are accessible.
    e. For each component, check the logs to ensure there are no startup errors or warnings:

```
podman logs <container_name>
```

2. Test workflows and automation processes.
    a. Run job templates: Run several key job templates, including those with dependencies on various credential types.
    b. Test workflow templates: Run workflow templates to ensure that workflow nodes run in the correct order and that the workflow completes successfully.
    c. Verify execution environments: Ensure that jobs run in the appropriate execution environments and can access required dependencies.

   d. Check job artifacts: Verify that job artifacts are properly stored and accessible.
   e. Validate job scheduling: Test scheduled jobs to ensure they run at the expected times.
3. Validate user access and permissions.
   a. User authentication: Test login functionality with various user accounts to ensure authentication works correctly.
   b. Role-based access controls: Verify that users have appropriate permissions for organizations, projects, inventories, and job templates.
   c. Team memberships: Confirm that team memberships and team-based permissions are intact.
   d. API access: Test API tokens and ensure that API access is functioning properly.
   e. SSO integration (if applicable): Verify that Single Sign-On authentication is working correctly.
4. Confirm content synchronization and availability.
   a. Collection synchronization: Check that you can synchronize collections from a remote.
   b. Collection Upload: Check that you can upload collections.
   c. Collection repositories: Verify that automation hub makes collections available and that execution environments can use them.
   d. Project synchronization: Check that projects can sync content from source control repositories.
   e. External content sources: Test synchronization from automation hub and Ansible Galaxy (if configured).
   f. Execution environment availability: Confirm that all required execution environments exist and that execution nodes can access them.
   g. Content dependencies: Verify that the system correctly resolves content dependencies when running jobs.

Prepare the OpenShift Container Platform target environment and import migration content

Prepare and assess your target OpenShift Container Platform environment, and import and reconcile your migrated content.

Prepare and assess the target environment

Transfer the migration artifact, create an OpenShift Container Platform project, and deploy Ansible Automation Platform using the Operator with configurations matching your source environment.

Procedure

1. Configure Ansible Automation Platform Operator for an Ansible Automation Platform deployment.
2. Set up the database configuration (internal or external).
3. Set up the Redis configuration (internal or external).
4. Install Ansible Automation Platform using Ansible Automation Platform Operator.
5. Create a backup of the initial OpenShift Container Platform deployment.
6. Verify the fresh installation functions correctly.

Import the migration content to the target environment

To import your environment, scale down Ansible Automation Platform components, restore databases, replace encryption secrets, and scale services back up.

NOTE:
The import process requires the latest version of Ansible Automation Platform named `aap` in the default `aap` namespace and all default database names and database users.

Procedure

1. Scale down Ansible Automation Platform components.
   a. Begin by scaling down the Ansible Automation Platform deployment by using `idle_aap`:

```
oc patch ansibleautomationplatform aap --type merge -p '{"spec":
{"idle_aap":true}}'
```

   b. Wait for component pods to stop. Only the 6 Operator pods will remain running.

```
NAME
 READY    STATUS      RESTARTS    AGE

pod/aap-controller-migration-4.6.13-5swc6                              0/1
    Completed   0          160m

pod/aap-gateway-operator-controller-manager-6b75c95458-4zrxv           2/2
    Running     0          26h

pod/ansible-lightspeed-operator-controller-manager-b674c55b8-qncjp     2/2
    Running     0          45h

pod/automation-controller-operator-controller-manager-6b79d48d4cchn    2/2
    Running     0          45h

pod/automation-hub-operator-controller-manager-5cd674c984-5njfj        2/2
    Running     0          45h

pod/eda-server-operator-controller-manager-645f4db5-d2flt              2/2
    Running     0          45h

pod/resource-operator-controller-manager-86b8f7bb54-cvz6d              2/2
    Running     0          45h
```

c. Scale down the Ansible Automation Platform Gateway Operator and Ansible Automation Platform Controller Operator:

```
 oc scale --replicas=0 deployment aap-gateway-operator-controller-manager
automation-controller-operator-controller-manager
```

Example output:

```
 deployment.apps/aap-gateway-operator-controller-manager scaled

deployment.apps/automation-controller-operator-controller-manager scaled
```

2. Scale up the idled Postgres `StatefulSet`.

```
oc scale --replicas=1 statefulset.apps/aap-postgres-15
```

3. Prepare a temporary environment for the database restore.
   a. Create a temporary Persistent Volume Claim (PVC) with appropriate settings and sizing.

      `aap-temp-pvc.yaml`

```
 ---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: aap-temp-pvc
  namespace: aap
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 200Gi
```

```
oc create -f aap-temp-pvc.yaml
```

b. Obtain the existing PostgreSQL image to use for temporary deployment:

```
 echo $(oc get pod/aap-postgres-15-0 -o
jsonpath="{.spec.containers[].image}")
```

c. Create a temporary PostgreSQL deployment with the mounted temporary PVC:

`aap-temp-postgres.yaml`

```
---
kind: Deployment
apiVersion: apps/v1
metadata:
  name: aap-temp-postgres
spec:
  replicas: 1
  selector:
    matchLabels:
      app: aap-temp-postgres
  template:
    metadata:
      labels:
        app: aap-temp-postgres
    spec:
      containers:
        - name: aap-temp-postgres
          image: <postgres image from previous step>
          command:
            - /bin/sh
            - '-c'
            - sleep infinity
          imagePullPolicy: Always
          securityContext:
            runAsNonRoot: true
            allowPrivilegeEscalation: false
          volumeMounts:
            - name: aap-temp-pvc
              mountPath: /tmp/aap-temp-pvc
      volumes:
        - name: aap-temp-pvc
          persistentVolumeClaim:
            claimName: aap-temp-pvc
```

```
oc create -f aap-temp-postgres.yaml
```

4. Copy the export artifact to the temporary PostgreSQL pod.
    a. First, obtain the pod name and set it as an environment variable:

```
export AAP_TEMP_POSTGRES=$(oc get pods --no-headers -o custom-
columns="metadata.name" | grep aap-temp-postgres)
```

    b. Test the environment variable:

```
echo $AAP_TEMP_POSTGRES
```

    Example output:

```
aap-temp-postgres-7b6c57f87f-s2ldp
```

    c. Copy the artifact and checksum to the PVC:

```
oc cp artifact.tar $AAP_TEMP_POSTGRES:/tmp/aap-temp-pvc/

oc cp artifact.tar.sha256 $AAP_TEMP_POSTGRES:/tmp/aap-temp-pvc/
```

5. Restore databases to Ansible Automation Platform PostgreSQL by using the temporary PostgreSQL pod.
    a. First, obtain the PostgreSQL passwords for all three databases and the PostgreSQL admin password:

```
echo "" && for secret in aap-controller-postgres-configuration aap-hub-
postgres-configuration aap-gateway-postgres-configuration

do

echo $secret

echo "PASSWORD: `oc get secrets $secret -o jsonpath="{.data['password']}"
| base64 -d`"

echo "USER: `oc get secrets $secret -o jsonpath="{.data['username']}" |
base64 -d`"

echo "DATABASE: `oc get secrets $secret -o jsonpath="{.data['database']}"
| base64 -d`"

echo ""

done && echo "POSTGRES ADMIN PASSWORD: `oc get secrets aap-gateway-
postgres-configuration -o jsonpath="{.data['postgres_admin_password']}" |
base64 -d`"
```

    b. Enter into the temporary PostgreSQL deployment and change directory to the mounted PVC containing the copied artifact:

```
oc exec -it deployment.apps/aap-temp-postgres -- /bin/bash
```

c. Inside the pod, change directory to `/tmp/aap-temp-pvc` and list its contents:

```
cd /tmp/aap-temp-pvc && ls -l
```

Example output:

```
total 2240
-rw-r--r--. 1 1000900000 1000900000 2273280 Jun 13 17:41 artifact.tar
-rw-r--r--. 1 1000900000 1000900000      79 Jun 13 17:42
artifact.tar.sha256
drwxrws---. 2 root       1000900000   16384 Jun 13 17:40 lost+found
```

d. Verify the archive:

```
sha256sum --check artifact.tar.sha256
```

Example output:

```
artifact.tar: OK
```

e. Extract the artifact and verify its contents:

```
tar xf artifact.tar && cd artifact && sha256sum --check sha256sum.txt
```

Example output:

```
 ./controller/controller.pgc: OK
./gateway/gateway.pgc: OK
./hub/hub.pgc: OK
```

f. Drop the automation controller database:

```
dropdb -h aap-postgres-15 automationcontroller
```

g. Alter the user temporarily with the `CREATEDB` role:

```
postgres=# ALTER USER automationcontroller WITH CREATEDB;
```

h. Create the database:

```
createdb -h aap-postgres-15 -U automationcontroller automationcontroller
```

i. Revert temporary user permission:

```
postgres=# ALTER USER automationcontroller NOCREATEDB;
```

j. Restore the automation controller database:

```
pg_restore --clean --create --no-owner -h aap-postgres-15 -U
automationcontroller -d automationcontroller controller/controller.pgc
```

k. Restore the automation hub database:

```
pg_restore --clean --create --no-owner -h aap-postgres-15 -U
automationhub -d automationhub hub/hub.pgc
```

l. Restore the platform gateway database:

```
pg_restore --clean --create --no-owner -h aap-postgres-15 -U gateway -d
gateway gateway/gateway.pgc
```

m. Exit the pod:

```
exit
```

6. Replace database field encryption secrets and clean up temporary resources.
   a. Replace database field encryption secrets:

```
oc set data secret/aap-controller-secret-key secret_key="<unencoded
controller_secret_key value from secrets.yml>"
```

```
oc set data secret/aap-db-fields-encryption-secret
secret_key="<unencoded gateway_secret_key value from secrets.yml>"
```

```
oc set data secret/aap-hub-db-fields-encryption
database_fields.symmetric.key="<unencoded hub_db_fields_encryption_key
value from secrets.yml>"
```

b. Clean up the temporary PostgreSQL and PVC:

```
oc delete -f aap-temp-postgres.yaml
```

```
oc delete -f aap-temp-pvc.yaml
```

7. Scale Ansible Automation Platform components back up.

a. Scale the platform gateway and automation controller Operators back up and wait for the platform gateway Operator reconciliation loop to complete:

The PostgreSQL `StatefulSet` returns to idle.

```
oc scale --replicas=1 deployment aap-gateway-operator-controller-manager
automation-controller-operator-controller-manager
```

Example output:

```
deployment.apps/aap-gateway-operator-controller-manager scaled

deployment.apps/automation-controller-operator-controller-manager scaled
```

```
oc logs -f $(oc get pods  --no-headers -o custom-
columns=":metadata.name" | grep aap-gateway-operator)
```

b. Wait for reconciliation to stop.

Example output:

```
 META: ending play

{"level":"info","ts":"2025-06-12T15:41:29Z","logger":"runner","msg":"Ansi
ble-runner exited
successfully","job":"5672263053238024330","name":"aap","namespace":"aap"}



----- Ansible Task Status Event StdOut (aap.ansible.com/v1alpha1,
Kind=AnsibleAutomationPlatform, aap/aap) -----




PLAY RECAP
*********************************************************************

localhost                  : ok=45   changed=0    unreachable=0
 failed=0    skipped=63   rescued=0    ignored=0
```

c. Scale Ansible Automation Platform back up using `idle_aap`:

```
 oc patch ansibleautomationplatform aap --type=merge -p '{"spec":
{"idle_aap":false}}'
```

Example output:

```
 ansibleautomationplatform.aap.ansible.com/aap patched
```

8. Wait for the `aap-gateway` pod to be running and clean up old service endpoints.

Example output:

```
 pod/aap-gateway-6c989b846c-47b91 2/2 Running 0 45s
```

```
 for i in HTTPPort Route ServiceNode; do oc exec -it deployment.apps/aap-
gateway -- aap-gateway-manage shell -c 'from aap_gateway_api.models import
'$i';print('$i'.objects.all().delete())'; done
```

Example output:

```
 (23, {'aap_gateway_api.ServiceAPIRoute': 4, 'aap_gateway_api.AdditionalRoute':
7, 'aap_gateway_api.Route': 11, 'aap_gateway_api.HTTPPort': 1})

(0, {})

(4, {'aap_gateway_api.ServiceNode': 4})
```

9. Run `awx-manage` to deprovision instances.

a. Obtain the automation controller pod:

```
export AAP_CONTROLLER_POD=$(oc get pods --no-headers -o custom-
columns=":metadata.name" | grep aap-controller-task)
```

b. Test the environment variable:

```
echo $AAP_CONTROLLER_POD
```

Example output:

```
aap-controller-task-759b6d9759-r59q9
```

c. Enter into the automation controller pod:

```
oc exec -it $AAP_CONTROLLER_POD -- /bin/bash

awx-manage list_instances
```

Example output:

```
bash-4.4$

[controlplane capacity=642 policy=100%]

    aap-controller-task-759b6d9759-r59q9 capacity=642 node_type=control
version=4.6.15 heartbeat="2025-06-12 21:39:48"

    node1.example.org capacity=0 node_type=hybrid version=4.6.13
heartbeat="2025-05-30 17:22:11"



[default capacity=0 policy=100%]

    node1.example.org capacity=0 node_type=hybrid version=4.6.13
heartbeat="2025-05-30 17:22:11"

    node2.example.org capacity=0 node_type=execution version=ansible-
runner-2.4.1 heartbeat="2025-05-30 17:22:08"
```

d. Remove old nodes with `awx-manage`, leaving only `aap-controller-task`:

```
awx-manage deprovision_instance --hostname=node1.example.org

awx-manage deprovision_instance --hostname=node2.example.org
```

10. Remove the `aap-resource-server` secret and allow the deployments to reconcile which will recreate the resource service keys and secret for the components:

```
$ oc delete secret/aap-resource-server
```

11. Run the `curl` command to repair automation hub filesystem data.

```
curl -d '{\"verify_checksums\": true }' -X POST -k https://<aap url>/api/
galaxy/pulp/api/v3/repair/ -u <admin_user>:<restored_admin_password>
```

Reconcile the target environment post-import

After importing your migration artifact, perform the following steps to reconcile your target environment.

Procedure

1. Modify the Django `SECRET_KEY` secrets to match the source platform.
2. Deprovision and reconfigure platform gateway service nodes.
3. Re-run platform gateway nodes and services register logic.
4. Convert container-specific settings to OpenShift Container Platform-appropriate formats.
5. Reconcile container resource allocations to OpenShift Container Platform resources.

Validate the target environment

Verify that all Ansible Automation Platform services are running, credentials work correctly, and migrated content like projects, inventories, and job templates are accessible on OpenShift Container Platform.

Procedure

1. Verify all migrated components are functional.
2. Test workflows and automation processes.
3. Validate user access and permissions.
4. Confirm content synchronization and availability.

5. Test integration with OpenShift Container Platform-specific features.

Prepare to migrate to Managed Ansible Automation Platform

Prepare and migrate your source environment to a Managed Ansible Automation Platform deployment, and reconcile the target environment post-migration.

Migrate to Managed Ansible Automation Platform

Submit a support ticket on the Red Hat Customer Portal to request a migration to Managed Ansible Automation Platform.

Before you begin

- You have a migration artifact from your source environment.

Procedure

1. Submit a support ticket on the Red Hat Customer Portal requesting a migration to Managed Ansible Automation Platform.

   The support ticket should include:

   - Source installation type (RPM, Containerized, OpenShift)
   - Managed Ansible Automation Platform URL or deployment name
   - Source version (installer or Operator version)
2. The Ansible Site Reliability Engineering (SRE) team provides instructions in the support ticket on how to upload the resulting migration artifact to secure storage for processing.
3. The Ansible SRE team imports the migration artifact into the identified target instance and notifies the customer through the support ticket.
4. The Ansible SRE team notifies customers of successful migration.

Reconcile the target environment post-migration

Update necessary configurations after migrating to Managed Ansible Automation Platform.

Procedure

1. Log in to the Managed Ansible Automation Platform instance by using the local administrator account to confirm that data was imported.
2. Perform the following actions based on the configuration of the source deployment:
    a. Reconfigure Single Sign-On (SSO) authenticators and mappings to reflect the new URLs.
    b. Update private automation hub content to reflect the new URLs.
        i. Run the following command to update the automation hub repositories:

```
curl -d '{\"verify_checksums\": true }' -X POST -k https://
<platform url>/api/galaxy/pulp/api/v3/repair/ -u
<admin_user>:<admin_password>
```

        ii. Perform a sync on any repositories configured in automation hub.
        iii. Push any custom execution environments from the source automation hub to the target automation hub.
    c. Reconfigure automation mesh.
3. After migration, you can request standard Site Reliability Engineering (SRE) tasks through support tickets, such as configuration of custom certificates, a custom domain, or connectivity through private endpoints.

Red Hat product documentation legal notices

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

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

www.redhat.com