



# Red Hat Ansible Automation Platform 2.5

## Containerized installation

Install the containerized version of Ansible Automation Platform



# Red Hat Ansible Automation Platform 2.5 Containerized installation

---

Install the containerized version of Ansible Automation Platform

## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This guide helps you to understand the installation requirements and processes behind our containerized version of Ansible Automation Platform.

## Table of Contents

<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>3</b>
<b>CHAPTER 1. ANSIBLE AUTOMATION PLATFORM CONTAINERIZED INSTALLATION</b> .....	<b>4</b>
1.1. TESTED DEPLOYMENT TOPOLOGIES	4
1.2. SYSTEM REQUIREMENTS	4
1.3. PREPARING THE RED HAT ENTERPRISE LINUX HOST FOR CONTAINERIZED INSTALLATION	5
1.4. DOWNLOADING ANSIBLE AUTOMATION PLATFORM	6
1.5. INSTALLING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM	6
1.5.1. Inventory file for online installation for containerized growth topology (all-in-one)	7
1.5.2. Inventory file for online installation for containerized enterprise topology	9
1.5.3. Additional information for configuring your inventory file	11
1.5.4. Running the installation command	11
1.6. ACCESSING ANSIBLE AUTOMATION PLATFORM	12
1.7. USING CUSTOM TLS CERTIFICATES	12
1.8. USING CUSTOM RECEPTOR SIGNING KEYS	13
1.9. ENABLING AUTOMATION HUB COLLECTION AND CONTAINER SIGNING	13
1.10. ADDING EXECUTION NODES	13
1.11. ADDING A SAFE PLUGIN VARIABLE TO EVENT-DRIVEN ANSIBLE CONTROLLER	14
1.12. UPDATING CONTAINER-BASED ANSIBLE AUTOMATION PLATFORM	14
1.13. BACKING UP CONTAINER-BASED ANSIBLE AUTOMATION PLATFORM	16
1.14. UNINSTALLING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM	16
<b>CHAPTER 2. HORIZONTAL SCALING IN RED HAT ANSIBLE AUTOMATION PLATFORM</b> .....	<b>18</b>
2.1. HORIZONTAL SCALING IN EVENT-DRIVEN ANSIBLE CONTROLLER	18
2.1.1. Sizing and scaling guidelines	18
2.1.2. Setting up horizontal scaling for Event-Driven Ansible controller	19
<b>APPENDIX A. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM</b> .....	<b>20</b>
A.1. DIAGNOSING THE PROBLEM	20
A.2. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM INSTALLATION	22
A.3. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM CONFIGURATION	25
A.4. CONTAINERIZED ANSIBLE AUTOMATION PLATFORM REFERENCE	26
<b>APPENDIX B. INVENTORY FILE VARIABLES</b> .....	<b>32</b>
B.1. GENERAL VARIABLES	32
B.2. AUTOMATION HUB VARIABLES	35
B.3. AUTOMATION CONTROLLER VARIABLES	45
B.4. EVENT-DRIVEN ANSIBLE CONTROLLER VARIABLES	51
B.5. PLATFORM GATEWAY VARIABLES	57
B.6. DATABASE VARIABLES	61
B.7. IMAGE VARIABLES	63
B.8. RECEPTOR VARIABLES	64
B.9. ANSIBLE VARIABLES	66



## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you have a suggestion to improve this documentation, or find an error, you can contact technical support at <https://access.redhat.com> to open a request.

# CHAPTER 1. ANSIBLE AUTOMATION PLATFORM CONTAINERIZED INSTALLATION

Ansible Automation Platform is a commercial offering that helps teams manage complex multi-tier deployments by adding control, knowledge, and delegation to Ansible-powered environments.

This guide helps you to understand the installation requirements and processes behind the containerized version of Ansible Automation Platform.



## NOTE

Upgrades from 2.4 Containerized Ansible Automation Platform Tech Preview to 2.5 Containerized Ansible Automation Platform are unsupported at this time.

## Prerequisites

- A Red Hat Enterprise Linux (RHEL) 9.2 based host. Use a minimal operating system base install.
- A non-root user for the Red Hat Enterprise Linux host, with sudo or other Ansible supported privilege escalation (sudo recommended). This user is responsible for the installation of containerized Ansible Automation Platform.
- SSH public key authentication for the non-root user. For guidelines on setting up SSH public key authentication for the non-root user, see [How to configure SSH public key authentication for passwordless login](#).
  - SSH keys are only required when installing on remote hosts. If doing a self contained local VM based installation, you can use **ansible\_connection=local**.
- Internet access from the Red Hat Enterprise Linux host if you are using the default online installation method.
- The appropriate network ports are open if a firewall is in place. For more information about the ports to open, see [Container topologies](#) in *Tested deployment models*.

## 1.1. TESTED DEPLOYMENT TOPOLOGIES

Red Hat tests Ansible Automation Platform 2.5 with a defined set of topologies to give you opinionated deployment options. The supported topologies include infrastructure topology diagrams, tested system configurations, example inventory files, and network ports information.

For containerized Ansible Automation Platform, there are two infrastructure topology shapes:

1. Growth - (All-in-one) Intended for organizations that are getting started with Ansible Automation Platform. This topology allows for smaller footprint deployments.
2. Enterprise - Intended for organizations that require Ansible Automation Platform deployments to have redundancy or higher compute for large volumes of automation. This is a more future-proofed scaled out architecture.

For more information about the tested deployment topologies for containerized Ansible Automation Platform, see [Container topologies](#) in *Tested deployment models*.

## 1.2. SYSTEM REQUIREMENTS



Each virtual machine (VM) has the following system requirements:

Requirement	Minimum requirement
RAM	16 GB
CPUs	4
Local disk	60 GB
Disk IOPS	3000

### 1.3. PREPARING THE RED HAT ENTERPRISE LINUX HOST FOR CONTAINERIZED INSTALLATION

Containerized Ansible Automation Platform runs the component services as Podman based containers on top of a Red Hat Enterprise Linux host. Prepare the Red Hat Enterprise Linux host to ensure a successful installation.

#### Procedure

1. Log in to the Red Hat Enterprise Linux host as your non-root user.
2. Set a hostname that is a fully qualified domain name (FQDN):

```
sudo hostnamectl set-hostname <your_hostname>
```

3. Register your Red Hat Enterprise Linux host with **subscription-manager**:

```
sudo subscription-manager register
```

4. Run **sudo dnf repolist** to validate that only the BaseOS and AppStream repositories are setup and enabled on the host:

```
$ sudo dnf repolist
Updating Subscription Management repositories.
repo id                repo name
rhel-9-for-x86_64-appstream-rpms    Red Hat Enterprise Linux 9 for x86_64 -
AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms      Red Hat Enterprise Linux 9 for x86_64 -
BaseOS (RPMs)
```

5. Ensure that only these repositories are available to the Red Hat Enterprise Linux host. For more information about managing custom repositories, see: [Managing custom software repositories](#).
6. Ensure that the host has DNS configured and can resolve host names and IP addresses by using a fully qualified domain name (FQDN). This is essential to ensure services can talk to one another.
7. Install **ansible-core**:

```
sudo dnf install -y ansible-core
```

- Optional: You can install additional utilities that can be useful for troubleshooting purposes, for example **wget**, **git-core**, **rsync**, and **vim**:

```
sudo dnf install -y wget git-core rsync vim
```

- Optional: To have the installer automatically pick up and apply your Ansible Automation Platform subscription manifest license, follow the steps in [Obtaining a manifest file](#).

### Additional resources

- For more information about registering your RHEL system, see [Getting Started with RHEL System Registration](#).
- For information about configuring unbound DNS, see [Setting up an unbound DNS server](#).
- For information about configuring DNS using BIND, see [Setting up and configuring a BIND DNS server](#).

## 1.4. DOWNLOADING ANSIBLE AUTOMATION PLATFORM

Choose the installer you need based on your Red Hat Enterprise Linux environment internet connectivity and download the installer to your Red Hat Enterprise Linux host.

### Procedure

- Download the latest installer .tar file from the [Ansible Automation Platform download page](#).
  - For online installations: **Ansible Automation Platform 2.5 Containerized Setup**
  - For offline or bundled installations: **Ansible Automation Platform 2.5 Containerized Setup Bundle**
- Copy the installer .tar file and the optional manifest .zip file onto your Red Hat Enterprise Linux host.
- Decide where you want the installer to reside on the file system. Installation related files are created under this location and require at least 10 GB for the initial installation.
- Unpack the installer .tar file into your installation directory, and go to the unpacked directory.
  - To unpack the online installer:

```
$ tar xfvz ansible-automation-platform-containerized-setup-<version>.tar.gz
```

- To unpack the offline or bundled installer:

```
$ tar xfvz ansible-automation-platform-containerized-setup-bundle-<version>-<arch_name>.tar.gz
```

## 1.5. INSTALLING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM

You can control the installation of Ansible Automation Platform with inventory files. Inventory files define the hosts and containers used and created, variables for components, and other information needed to customize the installation.

Example inventory files are provided in this document that you can copy and change to quickly get started.

Inventory files for the growth and enterprise topology are also found in the downloaded installer package:

- The default one named **inventory** is for the enterprise topology pattern.
- If you want to deploy the growth or all-in-one pattern you need to copy over or use the **inventory-growth** file instead.

Additionally, you can find example inventory files in [Container topologies](#) in *Tested deployment models*.

To use the example inventory files, replace the < > placeholders with your specific variables, and update the host names. Refer to the **README.md** file in the installation directory for more information about optional and required variables.

### 1.5.1. Inventory file for online installation for containerized growth topology (all-in-one)

Use the example inventory file to perform an online installation for the containerized growth topology (all-in-one):

```
# This is the Ansible Automation Platform growth installer inventory file
# Please consult the docs if you are unsure what to add
# For all optional variables please consult the included README.md
# or the Red Hat documentation:
#
https://docs.redhat.com/en/documentation/red_hat_automation_platform/2.5/html/containerized
_installation

# This section is for your platform gateway hosts
# -----
[automationgateway]
aap.example.org

# This section is for your automation controller hosts
# -----
[automationcontroller]
aap.example.org

# This section is for your automation hub hosts
# -----
[automationhub]
aap.example.org

# This section is for your Event-Driven Ansible controller hosts
# -----
[automationeda]
aap.example.org

# This section is for the Ansible Automation Platform database
```

```
# -----  
[database]  
aap.example.org  
  
[all:vars]  
# Ansible  
ansible_connection=local  
  
# Common variables  
#  
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-general-inventory-variables  
# -----  
postgresql_admin_username=postgres  
postgresql_admin_password=<set your own>  
  
registry_username=<your RHN username>  
registry_password=<your RHN password>  
  
redis_mode=standalone  
  
# Platform gateway  
#  
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-gateway-variables  
# -----  
gateway_admin_password=<set your own>  
gateway_pg_host=aap.example.org  
gateway_pg_password=<set your own>  
  
# Automation controller  
#  
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-controller-variables  
# -----  
controller_admin_password=<set your own>  
controller_pg_host=aap.example.org  
controller_pg_password=<set your own>  
  
# Automation hub  
#  
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-hub-variables  
# -----  
hub_admin_password=<set your own>  
hub_pg_host=aap.example.org  
hub_pg_password=<set your own>  
  
# Event-Driven Ansible controller  
#  
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#event-driven-ansible-controller  
# -----  
eda_admin_password=<set your own>  
eda_pg_host=aap.example.org  
eda_pg_password=<set your own>
```

## 1.5.2. Inventory file for online installation for containerized enterprise topology

Use the example inventory file to perform an online installation for the containerized enterprise topology:

```
# This is the Ansible Automation Platform enterprise installer inventory file
# Please consult the docs if you are unsure what to add
# For all optional variables please consult the included README.md
# or the Red Hat documentation:
#
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation

# This section is for your platform gateway hosts
# -----
[automationgateway]
gateway1.example.org
gateway2.example.org

# This section is for your automation controller hosts
# -----
[automationcontroller]
controller1.example.org
controller2.example.org

# This section is for your Ansible Automation Platform execution hosts
# -----
[execution_nodes]
hop1.example.org receptor_type='hop'
exec1.example.org
exec2.example.org

# This section is for your automation hub hosts
# -----
[automationhub]
hub1.example.org
hub2.example.org

# This section is for your Event-Driven Ansible controller hosts
# -----
[automationeda]
eda1.example.org
eda2.example.org

[redis]
gateway1.example.org
gateway2.example.org
hub1.example.org
hub2.example.org
eda1.example.org
eda2.example.org

[all:vars]

# Common variables
#
```

```

https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-general-inventory-variables
# -----
postgresql_admin_username=<set your own>
postgresql_admin_password=<set your own>
registry_username=<your RHN username>
registry_password=<your RHN password>

# Platform gateway
#
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-gateway-variables
# -----
gateway_admin_password=<set your own>
gateway_pg_host=externaldb.example.org
gateway_pg_database=<set your own>
gateway_pg_username=<set your own>
gateway_pg_password=<set your own>

# Automation controller
#
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-controller-variables
# -----
controller_admin_password=<set your own>
controller_pg_host=externaldb.example.org
controller_pg_database=<set your own>
controller_pg_username=<set your own>
controller_pg_password=<set your own>

# Automation hub
#
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#ref-hub-variables
# -----
hub_admin_password=<set your own>
hub_pg_host=externaldb.example.org
hub_pg_database=<set your own>
hub_pg_username=<set your own>
hub_pg_password=<set your own>

# Event-Driven Ansible controller
#
https://docs.redhat.com/en/documentation/red\_hat\_ansible\_automation\_platform/2.5/html/containerized\_installation/appendix-inventory-files-vars#event-driven-ansible-controller
# -----
eda_admin_password=<set your own>
eda_pg_host=externaldb.example.org
eda_pg_database=<set your own>
eda_pg_username=<set your own>
eda_pg_password=<set your own>

```

## Redis configuration for an enterprise topology

- Redis can be colocated with any other node in a clustered installation.

- By default the **redis\_mode** is set to **cluster**.
  - **redis\_mode=cluster**
- For more information about Redis, see [Caching and queueing system](#) in *Planning your installation*.

### 1.5.3. Additional information for configuring your inventory file

#### Offline or bundled installation

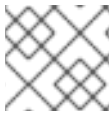
- To perform an offline installation, add the following under the **[all:vars]** group:

```
bundle_install=true
# The bundle directory must include /bundle in the path
bundle_dir=<full path to the bundle directory>
```

#### Configuring a HAProxy load balancer

- To configure a HAProxy load balancer in front of platform gateway with a custom CA cert, set the following inventory file variables under the **[all:vars]** group:

```
custom_ca_cert=<path_to_cert.crt>
gateway_main_url=<https://load_balancer_url>
```



#### NOTE

HAProxy SSL passthrough mode is not supported with platform gateway.

#### Configuring shared storage for automation hub

Shared storage is required when installing more than one instance of automation hub with a **file** storage backend. When installing a single instance of the automation hub, shared storage is optional.

- To configure shared storage for automation hub, set the following variable in the inventory file, ensuring your network file system (NFS) share has read, write, and execute permissions:

```
hub_shared_data_path=<path_to_nfs_share>
```

- To change the mount options for your NFS share, use the **hub\_shared\_data\_mount\_opts** variable. This variable is optional and the default value is **rw,sync,hard**.

#### Loading an automation controller license file

- To define the location of your automation controller license file, set the following variable in the inventory file:

```
controller_license_file=<full_path_to_your_manifest_zip_file>
```

### 1.5.4. Running the installation command

Use the following command to install containerized Ansible Automation Platform:

```
ansible-playbook -i inventory ansible.containerized_installer.install
```

- If your privilege escalation requires you to enter a password, append **-K** to the command line. You are then prompted for the **BECOME** password.
- You can use increasing verbosity, up to 4 v's (**-vvvv**) to see the details of the installation process. However, it is important to note that this can significantly increase installation time, so it is recommended that you use it only as needed or requested by Red Hat support.

## 1.6. ACCESSING ANSIBLE AUTOMATION PLATFORM

After the installation completes, the default protocol and ports used for Ansible Automation Platform are 80 (HTTP) and 443 (HTTPS).

You can customize the ports with the following variables:

```
envoy_http_port=80
envoy_https_port=443
```

If you want to disable HTTPS, set **envoy\_disable\_https** to **true**:

```
envoy_disable_https=true
```

### Accessing the platform UI

The platform UI is available by default at:

```
https://<gateway-node>:443
```

Log in as the admin user with the password you created for **gateway\_admin\_password**.

## 1.7. USING CUSTOM TLS CERTIFICATES

By default, the installer generates TLS certificates and keys for all services that are signed by a custom Certificate Authority (CA). You can provide a custom TLS certificate and key for each service. If that certificate is signed by a custom CA, you must provide the CA TLS certificate and key.

- Certificate Authority

```
ca_tls_cert=/full/path/to/tls/certificate
ca_tls_key=/full/path/to/tls/key
```

- Platform gateway

```
gateway_tls_cert=/full/path/to/tls/certificate
gateway_tls_key=/full/path/to/tls/key
```

- Automation controller

```
controller_tls_cert=/full/path/to/tls/certificate
controller_tls_key=/full/path/to/tls/key
```



- Automation hub

```
hub_tls_cert=/full/path/to/tls/certificate
hub_tls_key=/full/path/to/tls/key
```

- Event-Driven Ansible

```
eda_tls_cert=/full/path/to/tls/certificate
eda_tls_key=/full/path/to/tls/key
```

- PostgreSQL

```
postgresql_tls_cert=/full/path/to/tls/certificate
postgresql_tls_key=/full/path/to/tls/key
```

- Receptor

```
receptor_tls_cert=/full/path/to/tls/certificate
receptor_tls_key=/full/path/to/tls/key
```

## 1.8. USING CUSTOM RECEPTOR SIGNING KEYS

Receptor signing is enabled by default unless **receptor\_disable\_signing=true** is set, and an RSA key pair (public and private) is generated by the installer. However, you can give custom RSA public and private keys by setting the path variables:

```
receptor_signing_private_key=<full_path_to_private_key>
receptor_signing_public_key=<full_path_to_public_key>
```

## 1.9. ENABLING AUTOMATION HUB COLLECTION AND CONTAINER SIGNING

With automation hub you can sign Ansible collections and container images. This feature is not enabled by default, and you must provide the GPG key.

```
hub_collection_signing=true
hub_collection_signing_key=<full_path_to_collections_gpg_key>
hub_container_signing=true
hub_container_signing_key=<full_path_to_containers_gpg_key>
```

When the GPG key is protected by a passphrase, you must provide the passphrase.

```
hub_collection_signing_pass=<collections_gpg_key_passphrase>
hub_container_signing_pass=<containers_gpg_key_passphrase>
```

## 1.10. ADDING EXECUTION NODES

The containerized installer can deploy remote execution nodes. The **execution\_nodes** group in the inventory file handles this.

```
[execution_nodes]
<fqdn_of_your_execution_host>
```

An execution node is by default configured as an execution type running on port 27199 (TCP). This can be changed by the following variables:

```
receptor_port=27199
receptor_protocol=tcp
receptor_type=hop
```

The **receptor\_type** value can be either **execution** or **hop**, while the **receptor\_protocol** is either **tcp** or **udp**. By default, the nodes in the **execution\_nodes** group are added as peers for the controller node. However, you can change the peers configuration by using the **receptor\_peers** variable.

```
[execution_nodes]
fqdn_of_your_execution_host
fqdn_of_your_hop_host receptor_type=hop receptor_peers=["<fqdn_of_your_execution_host>"]
```

## 1.11. ADDING A SAFE PLUGIN VARIABLE TO EVENT-DRIVEN ANSIBLE CONTROLLER

When using `redhat.insights_eda` or similar plugins to run rulebook activations in Event-Driven Ansible controller, you must add a safe plugin variable to a directory in Ansible Automation Platform. This ensures connection between Event-Driven Ansible controller and the source plugin, and displays port mappings correctly.

### Procedure

1. Create a directory for the safe plugin variable: **`mkdir -p ./group_vars/automationedacontroller`**
2. Create a file within that directory for your new setting (for example, **`touch ./group_vars/automationedacontroller/custom.yml`**)
3. Add the variable **`automationedacontroller_additional_settings`** to extend the default **`settings.yml`** template for Event-Driven Ansible controller and add the **`SAFE_PLUGINS`** field with a list of plugins to enable. For example:

```
automationedacontroller_additional_settings:
  SAFE_PLUGINS:
    - ansible.eda.webhook
    - ansible.eda.alertmanager
```



### NOTE

You can also extend the **`automationedacontroller_additional_settings`** variable beyond **`SAFE_PLUGINS`** in the Django configuration file, `/etc/ansible-automation-platform/eda/settings.yml`

## 1.12. UPDATING CONTAINER-BASED ANSIBLE AUTOMATION PLATFORM

Perform a patch update for a container-based installation of Ansible Automation Platform from 2.5 to 2.5.x.

## Prerequisites

You have done the following:

- Reviewed the release notes for the associated patch release. For more information, see the [Ansible Automation Platform Release notes](#).
- Created a backup of your Ansible Automation Platform deployment. For more information, see [Backing up controller-based Ansible Automation Platform](#).

## Procedure

1. Download the latest version of the containerized installer from the [Ansible Automation Platform download](#).
  - a. For online installations **Ansible Automation Platform 2.5 Containerized Setup**
  - b. For offline or bundled installations: **Ansible Automation Platform 2.5 Containerized Setup Bundle**
2. Copy the installer **.tar** file onto your Red Hat Enterprise Linux host.
3. Decide where you want the installer to reside on the filesystem. Installation related files will be created under this location and require at least 10 GB for the initial installation.
4. Unpack the installer **.tar** file into your installation directory, and go to the unpacked directory.

- a. To unpack the online installer:

```
$ tar xfvz ansible-automation-platform-containerized-setup-<version>.tar.gz
```

- b. To unpack the offline or bundled installer:

```
$ tar xfvz ansible-automation-platform-containerized-setup-bundle-<version>-<arch name>.tar.gz
```

5. Edit the **inventory** file so that it matches your required configuration. You can keep the same parameters from your existing Ansible Automation Platform deployment or you can change the parameters to match any modifications to your environment.
6. Run the **install** command:

```
$ ansible-playbook -i inventory ansible.containerized_installer.install
```

- If your privilege escalation requires a password to be entered, append **-K** to the command. You will then be prompted for the **BECOME** password.
- You can use increasing verbosity, up to 4 v's (**-vvvv**) to see the details of the installation process. However it is important to note that this can significantly increase installation time, so it is recommended that you use it only as needed or requested by Red Hat support.

The installation will begin.

## 1.13. BACKING UP CONTAINER-BASED ANSIBLE AUTOMATION PLATFORM

Perform a back up of your container-based installation of Ansible Automation Platform.

### Procedure

1. Go to the Red Hat Ansible Automation Platform installation directory on your Red Hat Enterprise Linux host.
2. Run the **backup.yml** playbook:

```
$ ansible-playbook -i inventory ansible.containerized_installer.backup
```

This will backup the important data deployed by the containerized installer such as:

- PostgreSQL databases
- Configuration files
- Data files

By default, the backup directory is set to `~/backups`. You can change this by using the **backup\_dir** variable in your **inventory** file.

## 1.14. UNINSTALLING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM

To uninstall a containerized deployment, run the **uninstall.yml** playbook:

```
$ ansible-playbook -i inventory ansible.containerized_installer.uninstall
```

This stops all systemd units and containers and then deletes all resources used by the containerized installer such as:

- config and data directories and files
- systemd unit files
- Podman containers and images
- RPM packages

To keep container images, you can set the **container\_keep\_images** variable to **true**.

```
$ ansible-playbook -i inventory ansible.containerized_installer.uninstall -e  
container_keep_images=true
```

To keep PostgreSQL databases, you can set the **postgresql\_keep\_databases** variable to **true**.

```
$ ansible-playbook -i inventory ansible.containerized_installer.uninstall -e  
postgresql_keep_databases=true
```

**NOTE**

Use the Ansible Automation Platform secret key values rather than the autogenerated ones.

## CHAPTER 2. HORIZONTAL SCALING IN RED HAT ANSIBLE AUTOMATION PLATFORM

You can set up multi-node deployments for components across Ansible Automation Platform. Whether you require horizontal scaling for Automation Execution, Automation Decisions, or automation mesh, you can scale your deployments based on your organization's needs.

### 2.1. HORIZONTAL SCALING IN EVENT-DRIVEN ANSIBLE CONTROLLER

With Event-Driven Ansible controller, you can set up horizontal scaling for your events automation. This multi-node deployment enables you to define as many nodes as you prefer during the installation process. You can also increase or decrease the number of nodes at any time according to your organizational needs.

The following node types are used in this deployment:

#### API node type

Responds to the HTTP REST API of Event-Driven Ansible controller.

#### Worker node type

Runs an Event-Driven Ansible worker, which is the component of Event-Driven Ansible that not only manages projects and activations, but also executes the activations themselves.

#### Hybrid node type

Is a combination of the API node and the worker node.

The following example shows how you can set up an inventory file for horizontal scaling of Event-Driven Ansible controller on Red Hat Enterprise Linux VMs using the host group name **[automationeda]** and the node type variable **eda\_node\_type**:

```
[automationeda]
3.88.116.111
routable_hostname=automationeda-api.example.com eda_node_type=api

# worker node
3.88.116.112 routable_hostname=automationeda-api.example.com eda_node_type=worker
```

#### 2.1.1. Sizing and scaling guidelines

API nodes process user requests (interactions with the UI or API) while worker nodes process the activations and other background tasks required for Event-Driven Ansible to function properly. The number of API nodes you require correlates to the desired number of users of the application and the number of worker nodes correlates to the desired number of activations you want to run.

Since activations are variable and controlled by worker nodes, the supported approach for scaling is to use separate API and worker nodes instead of hybrid nodes due to the efficient allocation of hardware resources by worker nodes. By separating the nodes, you can scale each type independently based on specific needs, leading to better resource utilization and cost efficiency.

An example of an instance in which you might consider scaling up your node deployment is when you want to deploy Event-Driven Ansible for a small group of users who will run a large number of activations. In this case, one API node is adequate, but if you require more, you can scale up to three additional worker nodes.

To set up a multi-node deployment, follow the procedure in [Setting up horizontal scaling for Event-Driven Ansible controller](#).

## 2.1.2. Setting up horizontal scaling for Event-Driven Ansible controller

To scale up (add more nodes) or scale down (remove nodes), you must update the content of the inventory to add or remove nodes and rerun the installer.

### Procedure

1. Update the inventory to add two more worker nodes:

```
[automationedacontroller]

3.88.116.111 routable_hostname=automationedacontroller-api.example.com
eda_node_type=api

3.88.116.112 routable_hostname=automationedacontroller-api.example.com
eda_node_type=worker

# two more worker nodes
3.88.116.113 routable_hostname=automationedacontroller-api.example.com
eda_node_type=worker

3.88.116.114 routable_hostname=automationedacontroller-api.example.com
eda_node_type=worker
```

2. Re-run the installer.

# APPENDIX A. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM

Use this information to troubleshoot your containerized Ansible Automation Platform installation.

## A.1. DIAGNOSING THE PROBLEM

For general container-based troubleshooting, you can inspect the container logs for any running service to help troubleshoot underlying issues.

### Identifying the running containers

To get a list of the running container names run the following command:

```
$ podman ps --all --format "{{.Names}}"
```

Example output:

```
postgresql
redis-unix
redis-tcp
receptor
automation-controller-rsyslog
automation-controller-task
automation-controller-web
automation-eda-api
automation-eda-daphne
automation-eda-web
automation-eda-worker-1
automation-eda-worker-2
automation-eda-activation-worker-1
automation-eda-activation-worker-2
automation-eda-scheduler
automation-gateway-proxy
automation-gateway
automation-hub-api
automation-hub-content
automation-hub-web
automation-hub-worker-1
automation-hub-worker-2
```

### Inspecting the logs

To inspect any running container logs run the **journalctl** command:

```
$ journalctl CONTAINER_NAME=<container_name>
```

Example command with output:

```
$ journalctl CONTAINER_NAME=automation-gateway-proxy
```

```
Oct 08 01:40:12 aap.example.org automation-gateway-proxy[1919]: [2024-10-08 00:40:12.885][2]
```



```
[info][upstream] [external/envoy/source/common/upstream/cds_ap>
Oct 08 01:40:12 aap.example.org automation-gateway-proxy[1919]: [2024-10-08 00:40:12.885][2]
[info][upstream] [external/envoy/source/common/upstream/cds_ap>
Oct 08 01:40:19 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T00:40:16.753Z]
"GET /up HTTP/1.1" 200 - 0 1138 10 0 "192.0.2.1" "python->
```

To view the logs of a running container in real-time, run the **podman logs -f** command:

```
$ podman logs -f <container_name>
```

## Controlling container operations

You can control operations for a container by running the **systemctl** command:

```
$ systemctl --user status <container_name>
```

Example command with output:

```
$ systemctl --user status automation-gateway-proxy
● automation-gateway-proxy.service - Podman automation-gateway-proxy.service
   Loaded: loaded (/home/user/.config/systemd/user/automation-gateway-proxy.service; enabled;
   preset: disabled)
   Active: active (running) since Mon 2024-10-07 12:39:23 BST; 23h ago
     Docs: man:podman-generate-systemd(1)
   Process: 780 ExecStart=/usr/bin/podman start automation-gateway-proxy (code=exited,
   status=0/SUCCESS)
   Main PID: 1919 (common)
     Tasks: 1 (limit: 48430)
    Memory: 852.0K
       CPU: 2.996s
   CGroup: /user.slice/user-1000.slice/user@1000.service/app.slice/automation-gateway-
   proxy.service
           └─1919 /usr/bin/conmon --api-version 1 -c
           2dc3c7b2cecd73010bad1e0aaa806015065f92556ed3591c9d2084d7ee209c7a -u
           2dc3c7b2cecd73010bad1e0aaa80>
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:02.926Z]
"GET /api/galaxy/_ui/v1/settings/ HTTP/1.1" 200 - 0 654 58 47 ">
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:03.387Z]
"GET /api/controller/v2/config/ HTTP/1.1" 200 - 0 4018 58 44 "1">
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:03.370Z]
"GET /api/galaxy/v3/plugin/ansible/search/collection-versions/?>
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:03.405Z]
"GET /api/controller/v2/organizations/?role_level=notification_>
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:04.366Z]
"GET /api/galaxy/_ui/v1/me/ HTTP/1.1" 200 - 0 1368 79 40 "192.1">
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:04.360Z]
"GET /api/controller/v2/workflow_approvals/?page_size=200&statu>
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:04.379Z]
"GET /api/controller/v2/job_templates/7/ HTTP/1.1" 200 - 0 1356>
Oct 08 11:44:10 aap.example.org automation-gateway-proxy[1919]: [2024-10-08T10:44:04.378Z]
"GET /api/galaxy/_ui/v1/feature-flags/ HTTP/1.1" 200 - 0 207 81>
Oct 08 11:44:13 aap.example.org automation-gateway-proxy[1919]: [2024-10-08 10:44:13.856][2]
```

```
[info][upstream] [external/envoy/source/common/upstream/cds_ap>
Oct 08 11:44:13 aap.example.org automation-gateway-proxy[1919]: [2024-10-08 10:44:13.856][2]
[info][upstream] [external/envoy/source/common/upstream/cds_ap
```

## Getting container information about the execution plane

To get container information about automation controller, Event-Driven Ansible, and **execution\_nodes** nodes, prefix any Podman commands with either:

```
CONTAINER_HOST=unix:///run/user/<user_id>/podman/podman.sock
```

or

```
CONTAINERS_STORAGE_CONF=<user_home_directory>/aap/containers/storage.conf
```

Example with output:

```
$ CONTAINER_HOST=unix:///run/user/1000/podman/podman.sock podman images

REPOSITORY                                TAG      IMAGE ID   CREATED   SIZE
registry.redhat.io/ansible-automation-platform-25/ee-supported-rhel8 latest   59d1bc680a7c 6 days ago 2.24 GB
registry.redhat.io/ansible-automation-platform-25/ee-minimal-rhel8 latest   a64b9fc48094 6 days ago 338 MB
```

## A.2. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM INSTALLATION

The installation takes a long time, or has errors, what should I check?

1. Ensure your system meets the minimum requirements as outlined in the installation guide. Items such as improper storage choices and high latency when distributing across many hosts will all have a significant impact.
2. Check the installation log file located by default at **./aap\_install.log** unless otherwise changed within the local installer **ansible.cfg**.
3. Enable task profiling callbacks on an ad hoc basis to give an overview of where the installation program spends the most time. To do this, use the local **ansible.cfg** file. Add a callback line such as this under the **[defaults]** section:

```
$ cat ansible.cfg
[defaults]
callbacks_enabled = ansible.posix.profile_tasks
```

### Automation controller returns an error of 413

This error is due to **manifest.zip** license files that are larger than the **nginx\_client\_max\_body\_size** setting. If this error occurs, you will need to change the installation inventory file to include the following variables:

```
nginx_disable_hsts=false
```

```
nginx_http_port=8081
nginx_https_port=8444
nginx_client_max_body_size=20m
nginx_user_headers=[]
```

The current default setting of **20m** should be enough to avoid this issue.

### The installation failed with a “502 Bad Gateway” when going to the controller UI.

This error can occur and manifest itself in the installation application output as:

```
TASK [ansible.containerized_installer.automationcontroller : Wait for the Controller API to te ready]
*****
fatal: [daap1.lan]: FAILED! => {"changed": false, "connection": "close", "content_length": "150",
"content_type": "text/html", "date": "Fri, 29 Sep 2023 09:42:32 GMT", "elapsed": 0, "msg": "Status
code was 502 and not [200]: HTTP Error 502: Bad Gateway", "redirected": false, "server": "nginx",
"status": 502, "url": "https://daap1.lan:443/api/v2/ping/"}
```

- Check if you have an **automation-controller-web** container running and a systemd service.



#### NOTE

This is used at the regular unprivileged user not system wide level. If you have used **su** to switch to the user running the containers, you must set your **XDG\_RUNTIME\_DIR** environment variable to the correct value to be able to interact with the user **systemctl** units.

```
export XDG_RUNTIME_DIR="/run/user/$UID"
```

```
podman ps | grep web
systemctl --user | grep web
```

No output indicates a problem.

1. Try restarting the **automation-controller-web** service:

```
systemctl start automation-controller-web.service --user
systemctl --user | grep web
systemctl status automation-controller-web.service --user
```

```
Sep 29 10:55:16 daap1.lan automation-controller-web[29875]: nginx: [emerg] bind() to
0.0.0.0:443 failed (98: Address already in use)
Sep 29 10:55:16 daap1.lan automation-controller-web[29875]: nginx: [emerg] bind() to
0.0.0.0:80 failed (98: Address already in use)
```

The output indicates that the port is already, or still, in use by another service. In this case **nginx**.

2. Run:

```
sudo pkill nginx
```

3. Restart and status check the web service again.

Normal service output should look similar to the following, and should still be running:

```

Sep 29 10:59:26 daap1.lan automation-controller-web[30274]: WSGI app 0 (mountpoint= '/') ready in
3 seconds on interpreter 0x1a458c10 pid: 17 (default app)
Sep 29 10:59:26 daap1.lan automation-controller-web[30274]: WSGI app 0 (mountpoint= '/') ready in
3 seconds on interpreter 0x1a458c10 pid: 20 (default app)
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,043 INFO [-]
daphne.cli Starting server at tcp:port=8051:interface=127.0.>
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,043 INFO
Starting server at tcp:port=8051:interface=127.0.0.1
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,048 INFO [-]
daphne.server HTTP/2 support not enabled (install the http2 >
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,048 INFO
HTTP/2 support not enabled (install the http2 and tls Twisted ex>
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,049 INFO [-]
daphne.server Configuring endpoint tcp:port=8051:interface=1>
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,049 INFO
Configuring endpoint tcp:port=8051:interface=127.0.0.1
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,051 INFO [-]
daphne.server Listening on TCP address 127.0.0.1:8051
Sep 29 10:59:27 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:27,051 INFO
Listening on TCP address 127.0.0.1:8051
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: nginx entered RUNNING state, process has stayed up for > th>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: nginx entered RUNNING state, process has stayed up for > th>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: uwsgi entered RUNNING state, process has stayed up for > th>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: uwsgi entered RUNNING state, process has stayed up for > th>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: daphne entered RUNNING state, process has stayed up for > t>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: daphne entered RUNNING state, process has stayed up for > t>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: ws-heartbeat entered RUNNING state, process has stayed up f>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: ws-heartbeat entered RUNNING state, process has stayed up f>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: cache-clear entered RUNNING state, process has stayed up fo>
Sep 29 10:59:54 daap1.lan automation-controller-web[30274]: 2023-09-29 09:59:54,139 INFO
success: cache-clear entered RUNNING state, process has stayed up

```

You can run the installation program again to ensure everything installs as expected.

### When attempting to install containerized Ansible Automation Platform in Amazon Web Services you receive output that there is no space left on device

```

TASK [ansible.containerized_installer.automationcontroller : Create the receptor container]
*****
fatal: [ec2-13-48-25-168.eu-north-1.compute.amazonaws.com]: FAILED! => {"changed": false, "msg":
"Can't create container receptor", "stderr": "Error: creating container storage: creating an ID-mapped
copy of layer \"98955f43cc908bd50ff43585fec2c7dd9445eaf05eecd1e3144f93ffc00ed4ba\": error
during chown: storage-chown-by-maps: lchown usr/local/lib/python3.9/site-
packages/azure/mgmt/network/v2019_11_01/operations/__pycache__/_available_service_aliases_oper

```

```

ations.cpython-39.pyc: no space left on device: exit status 1\n", "stderr_lines": ["Error: creating
container storage: creating an ID-mapped copy of layer
\98955f43cc908bd50ff43585fec2c7dd9445eaf05eecd1e3144f93ffc00ed4ba\": error during chown:
storage-chown-by-maps: lchown usr/local/lib/python3.9/site-
packages/azure/mgmt/network/v2019_11_01/operations/__pycache__/_available_service_aliases_oper
ations.cpython-39.pyc: no space left on device: exit status 1"], "stdout": "", "stdout_lines": []}

```

If you are installing a **/home** filesystem into a default Amazon Web Services marketplace RHEL instance, it might be too small since **/home** is part of the root **/** filesystem. You will need to make more space available. The documentation specifies a minimum of 40GB for a single-node deployment of containerized Ansible Automation Platform.

### "Install container tools" task fails due to unavailable packages

This error occurs in the installation application output as:

```

TASK [ansible.containerized_installer.common : Install container tools]
*****
fatal: [192.0.2.1]: FAILED! => {"changed": false, "failures": ["No package crun available.", "No
package podman available.", "No package slirp4netns available.", "No package fuse-overlayfs
available."], "msg": "Failed to install some of the specified packages", "rc": 1, "results": []}
fatal: [192.0.2.2]: FAILED! => {"changed": false, "failures": ["No package crun available.", "No
package podman available.", "No package slirp4netns available.", "No package fuse-overlayfs
available."], "msg": "Failed to install some of the specified packages", "rc": 1, "results": []}
fatal: [192.0.2.3]: FAILED! => {"changed": false, "failures": ["No package crun available.", "No
package podman available.", "No package slirp4netns available.", "No package fuse-overlayfs
available."], "msg": "Failed to install some of the specified packages", "rc": 1, "results": []}
fatal: [192.0.2.4]: FAILED! => {"changed": false, "failures": ["No package crun available.", "No
package podman available.", "No package slirp4netns available.", "No package fuse-overlayfs
available."], "msg": "Failed to install some of the specified packages", "rc": 1, "results": []}
fatal: [192.0.2.5]: FAILED! => {"changed": false, "failures": ["No package crun available.", "No
package podman available.", "No package slirp4netns available.", "No package fuse-overlayfs
available."], "msg": "Failed to install some of the specified packages", "rc": 1, "results": []}

```

To fix this error, run the following command on the target hosts:

```
sudo subscription-manager register
```

## A.3. TROUBLESHOOTING CONTAINERIZED ANSIBLE AUTOMATION PLATFORM CONFIGURATION

Sometimes the post install for seeding my Ansible Automation Platform content errors out. This could manifest itself as output similar to this:

```

TASK [infra.controller_configuration.projects : Configure Controller Projects | Wait for finish the
projects creation] *****
Friday 29 September 2023 11:02:32 +0100 (0:00:00.443) 0:00:53.521 *****
FAILED - RETRYING: [daap1.lan]: Configure Controller Projects | Wait for finish the projects creation
(1 retries left).
failed: [daap1.lan] (item={'failed': 0, 'started': 1, 'finished': 0, 'ansible_job_id': '536962174348.33944',
'results_file': '/home/aap/.ansible_async/536962174348.33944', 'changed': False,
'__controller_project_item': {'name': 'AAP Config-As-Code Examples', 'organization': 'Default',
'scm_branch': 'main', 'scm_clean': 'no', 'scm_delete_on_update': 'no', 'scm_type': 'git'},

```

```
'scm_update_on_launch': 'no', 'scm_url': 'https://github.com/user/repo.git'}, 'ansible_loop_var':
'__controller_project_item'}) => {"__projects_job_async_results_item": {"__controller_project_item":
{"name": "AAP Config-As-Code Examples", "organization": "Default", "scm_branch": "main",
"scm_clean": "no", "scm_delete_on_update": "no", "scm_type": "git", "scm_update_on_launch": "no",
"scm_url": "https://github.com/user/repo.git"}, "ansible_job_id": "536962174348.33944",
"ansible_loop_var": "__controller_project_item", "changed": false, "failed": 0, "finished": 0,
"results_file": "/home/aap/.ansible_async/536962174348.33944", "started": 1}, "ansible_job_id":
"536962174348.33944", "ansible_loop_var": "__projects_job_async_results_item", "attempts": 30,
"changed": false, "finished": 0, "results_file": "/home/aap/.ansible_async/536962174348.33944",
"started": 1, "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}
```

The **infra.controller\_configuration.dispatch** role uses an asynchronous loop with 30 retries to apply each configuration type, and the default delay between retries is 1 second. If the configuration is large, this might not be enough time to apply everything before the last retry occurs.

Increase the retry delay by setting the **controller\_configuration\_async\_delay** variable to something other than 1 second. For example, setting it to 2 seconds doubles the retry time. The place to do this would be in the repository where the controller configuration is defined. It could also be added to the **[all:vars]** section of the installation program inventory file.

A few instances have shown that no additional modification is required, and re-running the installation program again worked.

## A.4. CONTAINERIZED ANSIBLE AUTOMATION PLATFORM REFERENCE

**Can you give details of the architecture for the Ansible Automation Platform containerized design?**

We use as much of the underlying native Red Hat Enterprise Linux technology as possible. Podman is used for the container runtime and management of services.

Use **podman ps** to list the running containers on the system:

```
$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
88ed40495117	registry.redhat.io/rhel8/postgresql-13:latest	run-postgresql	48
minutes ago	Up 47 minutes	postgresql	
8f55ba612f04	registry.redhat.io/rhel8/redis-6:latest	run-redis	47
minutes ago	Up 47 minutes	redis	
56c40445c590	registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8:latest	receptor	
/usr/bin/receptor...	47 minutes ago	Up 47 minutes	
f346f05d56ee	registry.redhat.io/ansible-automation-platform-24/controller-rhel8:latest	automation-controller-rsyslog	
/usr/bin/launch_a...	47 minutes ago	Up 45 minutes	
26e3221963e3	registry.redhat.io/ansible-automation-platform-24/controller-rhel8:latest	automation-controller-task	
/usr/bin/launch_a...	46 minutes ago	Up 45 minutes	
c7ac92a1e8a1	registry.redhat.io/ansible-automation-platform-24/controller-rhel8:latest	automation-controller-web	
/usr/bin/launch_a...	46 minutes ago	Up 28 minutes	

Use **podman images** to display information about locally stored images:

```
$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.redhat.io/ansible-automation-platform-24/ee-supported-rhel8	latest	b497bdbbee59e	10 days ago	3.16 GB
registry.redhat.io/ansible-automation-platform-24/controller-rhel8	latest	ed8ebb1c1baa	10 days ago	1.48 GB
registry.redhat.io/rhel8/redis-6	latest	78905519bb05	2 weeks ago	357 MB
registry.redhat.io/rhel8/postgresql-13	latest	9b65bc3d0413	2 weeks ago	765 MB

Containerized Ansible Automation Platform runs as rootless containers for enhanced security by default. This means you can install containerized Ansible Automation Platform by using any local unprivileged user account. Privilege escalation is only needed for certain root level tasks, and by default is not needed to use root directly.

The installation program adds the following files to the filesystem where you run the installation program on the underlying Red Hat Enterprise Linux host:

```
$ tree -L 1
.
├── aap_install.log
├── ansible.cfg
├── collections
├── galaxy.yml
├── inventory
├── LICENSE
├── meta
├── playbooks
├── plugins
├── README.md
├── requirements.yml
└── roles
```

The installation root directory includes other containerized services that make use of Podman volumes for example.

Here are some examples for further reference:

The **containers** directory includes some of the Podman specifics used and installed for the execution plane:

```
containers/
├── podman
├── storage
│   ├── defaultNetworkBackend
│   ├── libpod
│   ├── networks
│   ├── overlay
│   ├── overlay-containers
│   ├── overlay-images
│   ├── overlay-layers
│   ├── storage.lock
│   └── users.lock
└── storage.conf
```

The **controller** directory has some of the installed configuration and runtime data points:

```

controller/
├── data
│   ├── job_execution
│   ├── projects
│   └── rsyslog
├── etc
│   ├── conf.d
│   ├── launch_awx_task.sh
│   ├── settings.py
│   ├── tower.cert
│   └── tower.key
├── nginx
│   └── etc
├── rsyslog
│   └── run
├── supervisor
│   └── run

```

The **receptor** directory has the automation mesh configuration:

```

receptor/
├── etc
│   └── receptor.conf
├── run
│   ├── receptor.sock
│   └── receptor.sock.lock

```

After installation, you will also find other pieces in the local users home directory such as the **.cache** directory:

```

.cache/
├── containers
│   └── short-name-aliases.conf.lock
├── rhsm
│   └── rhsm.log

```

As services are run using rootless Podman by default, you can use other services such as running **systemd** as non-privileged users. Under **systemd** you can see some of the component service controls available:

The **.config** directory:

```

.config/
├── cni
│   ├── net.d
│   └── cni.lock
├── containers
│   ├── auth.json
│   └── containers.conf
├── systemd
│   └── user
│       ├── automation-controller-rsyslog.service
│       ├── automation-controller-task.service
│       ├── automation-controller-web.service
│       └── default.target.wants

```



```

├── podman.service.d
├── postgresql.service
├── receptor.service
├── redis.service
└── sockets.target.wants

```

This is specific to Podman and conforms to the Open Container Initiative (OCI) specifications. When you run Podman as the root user **/var/lib/containers** is used by default, for standard users the hierarchy under **\$HOME/.local** is used.

The **.local** directory:

```

.local/
├── share
│   └── containers
│       ├── cache
│       ├── podman
│       └── storage

```

As an example **.local/storage/volumes** contains what the output from **podman volume ls** provides:

```

$ podman volume ls

DRIVER    VOLUME NAME
local     d73d3fe63a957bee04b4853fd38c39bf37c321d14fdab9ee3c9df03645135788
local     postgresql
local     redis_data
local     redis_etc
local     redis_run

```

The execution plane is isolated from the control plane main services to ensure it does not affect the main services.

### Control plane services

Control plane services run with the standard Podman configuration and can be found in: **~/.local/share/containers/storage**.

### Execution plane services

Execution plane services (automation controller, Event-Driven Ansible and execution nodes) use a dedicated configuration found in **~/aap/containers/storage.conf**. This separation prevents execution plane containers from affecting the control plane services.

You can view the execution plane configuration with one of the following commands:

```
CONTAINERS_STORAGE_CONF=~/.aap/containers/storage.conf podman <subcommand>
```

```
CONTAINER_HOST=unix:///run/user/<user uid>/podman/podman.sock podman <subcommand>
```

### How can I see host resource utilization statistics?

- Run:

```
$ podman container stats -a
```

```

ID          NAME                CPU %    MEM USAGE / LIMIT  MEM %    NET IO    BLOCK
IO PIDS     CPU TIME  AVG CPU %
0d5d8eb93c18 automation-controller-web  0.23%    959.1MB / 3.761GB  25.50%    0B / 0B
0B / 0B    16        20.885142s 1.19%
3429d559836d automation-controller-rsyslog 0.07%    144.5MB / 3.761GB  3.84%    0B / 0B
0B / 0B    6         4.099565s 0.23%
448d0bae0942 automation-controller-task  1.51%    633.1MB / 3.761GB  16.83%    0B / 0B  0B
/ 0B      33        34.285272s 1.93%
7f140e65b57e receptor                0.01%    5.923MB / 3.761GB  0.16%    0B / 0B  0B / 0B
7         1.010613s 0.06%
c1458367ca9c redis                    0.48%    10.52MB / 3.761GB  0.28%    0B / 0B  0B / 0B  5
9.074042s 0.47%
ef712cc2dc89 postgresql                0.09%    21.88MB / 3.761GB  0.58%    0B / 0B  0B / 0B
21        15.571059s 0.80%
```

The previous is an example of a Dell sold and offered containerized Ansible Automation Platform solution (DAAP) install and utilizes ~1.8Gb RAM.

### How much storage is used and where?

The container volume storage is under the local user at **\$HOME/.local/share/containers/storage/volumes**.

1. To view the details of each volume run:

```
$ podman volume ls
```

2. Then run:

```
$ podman volume inspect <volume_name>
```

Here is an example:

```

$ podman volume inspect postgresql
[
  {
    "Name": "postgresql",
    "Driver": "local",
    "Mountpoint": "/home/aap/.local/share/containers/storage/volumes/postgresql/_data",
    "CreatedAt": "2024-01-08T23:39:24.983964686Z",
    "Labels": {},
    "Scope": "local",
    "Options": {},
    "MountCount": 0,
    "NeedsCopyUp": true
  }
]
```

Several files created by the installation program are located in **\$HOME/aap/** and bind-mounted into various running containers.

1. To view the mounts associated with a container run:

```
$ podman ps --format "{{.ID}}\t{{.Command}}\t{{.Names}}"
```

```
89e779b81b83 run-postgresql postgresql
4c33cc77ef7d run-redis redis
3d8a028d892d /usr/bin/receptor... receptor
09821701645c /usr/bin/launch_a... automation-controller-rsyslog
a2ddb5cac71b /usr/bin/launch_a... automation-controller-task
fa0029a3b003 /usr/bin/launch_a... automation-controller-web
20f192534691 gunicorn --bind 1... automation-eda-api
f49804c7e6cb daphne -b 127.0.0... automation-eda-daphne
d340b9c1cb74 /bin/sh -c nginx ... automation-eda-web
111f47de5205 aap-eda-manage rq... automation-eda-worker-1
171fcb1785af aap-eda-manage rq... automation-eda-worker-2
049d10555b51 aap-eda-manage rq... automation-eda-activation-worker-1
7a78a41a8425 aap-eda-manage rq... automation-eda-activation-worker-2
da9afa8ef5e2 aap-eda-manage sc... automation-eda-scheduler
8a2958be9baf gunicorn --name p... automation-hub-api
0a8b57581749 gunicorn --name p... automation-hub-content
68005b987498 nginx -g daemon o... automation-hub-web
cb07af77f89f pulpcore-worker automation-hub-worker-1
a3ba05136446 pulpcore-worker automation-hub-worker-2
```

2. Then run:

```
$ podman inspect <container_name> | jq -r .[].Mounts[].Source
```

```
/home/aap/.local/share/containers/storage/volumes/receptor_run/_data
/home/aap/.local/share/containers/storage/volumes/redis_run/_data
/home/aap/aap/controller/data/rsyslog
/home/aap/aap/controller/etc/tower.key
/home/aap/aap/controller/etc/conf.d/callback_receiver_workers.py
/home/aap/aap/controller/data/job_execution
/home/aap/aap/controller/nginx/etc/controller.conf
/home/aap/aap/controller/etc/conf.d/subscription_usage_model.py
/home/aap/aap/controller/etc/conf.d/cluster_host_id.py
/home/aap/aap/controller/etc/conf.d/insights.py
/home/aap/aap/controller/rsyslog/run
/home/aap/aap/controller/data/projects
/home/aap/aap/controller/etc/settings.py
/home/aap/aap/receptor/etc/receptor.conf
/home/aap/aap/controller/etc/conf.d/execution_environments.py
/home/aap/aap/tls/extracted
/home/aap/aap/controller/supervisor/run
/home/aap/aap/controller/etc/uwsgi.ini
/home/aap/aap/controller/etc/conf.d/container_groups.py
/home/aap/aap/controller/etc/launch_awx_task.sh
/home/aap/aap/controller/etc/tower.cert
```

3. If the **jq** RPM is not installed, install with:

```
$ sudo dnf -y install jq
```

## APPENDIX B. INVENTORY FILE VARIABLES

The following tables contain information about the variables used in Ansible Automation Platform's installation **inventory** files. The tables include the variables that you can use for RPM-based installation and container-based installation.

### B.1. GENERAL VARIABLES

RPM variable name	Container variable name	Description
	<b>bundle_dir</b>	The path to the bundle directory.  Default = <b>false</b>
	<b>bundle_install</b>	Use offline installation. Set to <b>true</b> to enable offline installation.  Default = <b>false</b>
	<b>ca_tls_cert</b>	Define a Certification Authority certificate here along with a matching key in <b>ca_tls_key</b> when you want the installer to create leaf certificates for each product for you.
	<b>ca_tls_key</b>	Define the key for a Certification Authority certificate here for the matching certificate in <b>ca_tls_cert</b> when you want the installer to create leaf certificates for each product for you.
	<b>ca_tls_remote</b>	TLS CA remote files.  Default = <b>false</b>
	<b>container_compress</b>	Container compression software.  Default = <b>gzip</b>
	<b>container_keep_images</b>	Keep container images.  Default = <b>false</b>
	<b>container_pull_images</b>	Pull newer container images.  Default = <b>true</b>

RPM variable name	Container variable name	Description
	<b>custom_ca_cert</b>	Define a custom Certification Authority certificate here when you have the leaf certificates created for each product and need the certificate trust to be established.
<b>enable_insights_collection</b>		<p>The default install registers the node to the Red Hat Insights for Red Hat Ansible Automation Platform for the Red Hat Ansible Automation Platform Service if the node is registered with Subscription Manager.</p> <p>Set to <b>False</b> to disable.</p> <p>Default = <b>true</b></p>
<b>nginx_tls_protocols</b>		<p>Defines support for <b>ssl_protocols</b> in NGINX.</p> <p>Values available <b>TLSv1, TLSv1.1, TLSv1.2, TLSv1.3</b>.</p> <p>The TLSv1.1 and TLSv1.2 parameters only work when OpenSSL 1.0.1 or higher is used.</p> <p>The TLSv1.3 parameter only works when OpenSSL 1.1.1 or higher is used.</p> <p>If <b>nginx_tls_protocols = ['TLSv1.3']</b> only TLSv1.3 is enabled. To set more than one protocol use <b>nginx_tls_protocols = ['TLSv1.2', 'TLSv1.3']</b>.</p> <p>Default = <b>TLSv1.2</b></p>
<b>nginx_user_http_config</b>		<p>List of NGINX configurations for <b>/etc/nginx/nginx.conf</b> under the http section.</p> <p>Each element in the list is provided into <b>http nginx config</b> as a separate line.</p> <p>Default = empty list</p>

RPM variable name	Container variable name	Description
	<b>registry_auth</b>	Use registry authentication.  Default = <b>true</b>
	<b>registry_ns_aap</b>	Ansible Automation Platform registry namespace.  Default = <b>ansible-automation-platform-25</b>
	<b>registry_ns_rhel</b>	RHEL registry namespace.  Default = <b>rhel8</b>
<b>redis_mode</b>		Redis can be colocated with platform gateway, automation hub, and Event-Driven Ansible controller nodes.  Default = cluster
<b>registry_password</b>	<b>registry_password</b>	This variable is only required if a non-bundle installer is used.  Password credential for access to <b>registry_url</b> .  Enter your Red Hat Registry Service Account credentials in <b>registry_username</b> and <b>registry_password</b> to link to the Red Hat container registry.  When <b>registry_url</b> is <b>registry.redhat.io</b> , username and password are required if not using a bundle installer.  For more information, see <a href="#">Setting registry_username and registry_password</a> .
	<b>registry_tls_verify</b>	Verify registry TLS.  Default = <b>true</b>
<b>registry_url</b>	<b>registry_url</b>	URL for the registry source.  Default = <b>registry.redhat.io</b>

RPM variable name	Container variable name	Description
<b>registry_username</b>	<b>registry_username</b>	<p>This variable is only required if a non-bundle installer is used.</p> <p>User credential for access to <b>registry_url</b>.</p> <p>Enter your Red Hat Registry Service Account credentials in <b>registry_username</b> and <b>registry_password</b> to link to the Red Hat container registry.</p> <p>For more information, see <a href="#">Setting registry_username and registry_password</a>.</p>
<b>routable_hostname</b>	<b>routable_hostname</b>	<p>This variable is used if the machine running the installer can only route to the target host through a specific URL. For example, if you use short names in your inventory, but the node running the installer can only resolve that host by using a FQDN.</p> <p>If <b>routable_hostname</b> is not set, it should default to <b>ansible_host</b>. If you do not set <b>ansible_host</b>, <b>inventory_hostname</b> is used as a last resort.</p> <p>This variable is used as a host variable for particular hosts and not under the <b>[all:vars]</b> section.</p> <p>For further information, see <a href="#">Assigning a variable to one machine: host variables</a>.</p>

## B.2. AUTOMATION HUB VARIABLES

RPM variable name	Container variable name	Description
-------------------	-------------------------	-------------

RPM variable name	Container variable name	Description
<b>automationhub_admin_password</b>	<b>hub_admin_password</b>	<p><i>Required</i></p> <p>Required passwords must be enclosed in quotes when they are provided in plain text in the inventory file.</p> <p>Use of special characters for this variable is limited. The password can include any printable ASCII character except <code>/</code>, <code>"</code>, or <code>@</code>.</p>
<b>automationhub_api_token</b>		<p>This variable can be used to provide the installer with an existing token.</p> <p>For example, a regenerated token in Hub UI will invalidate an existing token. Use <b>automationhub_api_token</b> to use that token in the installer the next time you run the installer.</p>
<b>automationhub_auto_sign_collections</b>	<b>hub_collection_auto_sign</b>	<p>If a collection signing service is enabled, collections are not signed automatically by default.</p> <p>Setting this parameter to <b>true</b> signs them by default.</p> <p>Default = <b>false</b></p>
<b>automationhub_backup_collections</b>		<p><i>Optional</i></p> <p>Ansible automation hub provides artifacts in <b>/var/lib/pulp</b>. Automation controller automatically backs up the artifacts by default.</p> <p>You can also set <b>automationhub_backup_collections</b> to <b>false</b> and the backup and restore process will not backup or restore <b>/var/lib/pulp</b>.</p> <p>Default = <b>true</b></p>



RPM variable name	Container variable name	Description
<b>automationhub_collection_download_count</b>		<p><i>Optional</i></p> <p>Determines whether download count is displayed on the UI.</p> <p>Default = <b>false</b></p>
<b>automationhub_collection_seed_repository</b>		<p>When you run the bundle installer, validated content is uploaded to the <b>validated</b> repository, and certified content is uploaded to the <b>rh-certified</b> repository.</p> <p>By default, both certified and validated content are uploaded.</p> <p>Possible values of this variable are <b>certified</b> or <b>validated</b>.</p> <p>If you do not want to install content, set <b>automationhub_seed_collections</b> to <b>false</b> to disable the seeding.</p> <p>If you only want one type of content, set <b>automationhub_seed_collections</b> to <b>true</b> and <b>automationhub_collection_seed_repository</b> to the type of content you do want to include.</p>
<b>automationhub_collection_signing_service_key</b>	<b>hub_collection_signing_key</b>	<p>If a collection signing service is enabled, you must provide this variable to ensure that collections can be properly signed.</p> <p><b>/absolute/path/to/key/to/sign</b></p>
<b>automationhub_collection_signing_service_script</b>		<p>If a collection signing service is enabled, you must provide this variable to ensure that collections can be properly signed.</p> <p><b>/absolute/path/to/script/that/signs</b></p>

RPM variable name	Container variable name	Description
<b>automationhub_container_signing_service_key</b>	<b>hub_container_signing_key</b>	If a container signing service is enabled, you must provide this variable to ensure that containers can be properly signed.  <b>/absolute/path/to/key/to/sign</b>
<b>automationhub_container_signing_service_script</b>		If a container signing service is enabled, you must provide this variable to ensure that containers can be properly signed.  <b>/absolute/path/to/script/that/signs</b>
<b>automationhub_create_default_collection_signing_service</b>	<b>hub_collection_signing_service</b>	Set this variable to <b>true</b> to create a collection signing service.  Default = <b>false</b>
<b>automationhub_create_default_container_signing_service</b>	<b>hub_container_signing_service</b>	Set this variable to <b>true</b> to create a container signing service.  Default = <b>false</b>
<b>automationhub_disable_hsts</b>	<b>hub_nginx_disable_hsts</b>	The default installation deploys a TLS enabled automation hub. Use this variable if you deploy automation hub with <i>HTTP Strict Transport Security</i> (HSTS) web-security policy enabled. This variable disables the HSTS web-security policy mechanism.  Default = <b>false</b>
<b>automationhub_disable_https</b>	<b>hub_nginx_disable_https</b>	<i>Optional</i>  If automation hub is deployed with HTTPS enabled.  Default = <b>false</b>

RPM variable name	Container variable name	Description
<b>automationhub_enable_analytics</b>		<p>A Boolean indicating whether to enable pulp analytics for the version of <b>pulpcore</b> used in automation hub in Ansible Automation Platform 2.5.</p> <p>To enable pulp analytics, set <b>automationhub_enable_analytics</b> to <b>true</b>.</p> <p>Default = <b>false</b></p>
<b>automationhub_enable_api_access_log</b>		<p>When set to <b>true</b>, this variable creates a log file at <b>/var/log/galaxy_api_access.log</b> that logs all user actions made to the platform, including their username and IP address.</p> <p>Default = <b>false</b></p>
<b>automationhub_enable_unauthenticated_collections</b>		<p>Set this variable to <b>true</b> to enable unauthorized users to view collections.</p> <p>Default = <b>false</b></p>
<b>automationhub_enable_unauthenticated_collections_download</b>		<p>Set this variable to <b>true</b> to enable unauthorized users to download collections.</p> <p>Default = <b>false</b></p>
<b>automationhub_importer_settings</b>	<b>hub_galaxy_importer</b>	<p><i>Optional</i></p> <p>Dictionary of setting to pass to galaxy-importer. At import time, collections can go through a series of checks.</p> <p>Behavior is driven by <b>galaxy-importer.cfg</b> configuration.</p> <p>Examples are <b>ansible-doc</b>, <b>ansible-lint</b>, and <b>flake8</b>.</p> <p>This parameter enables you to drive this configuration.</p>

RPM variable name	Container variable name	Description
<b>automationhub_pg_database</b>	<b>hub_pg_database</b>	The PostgreSQL database name.  RPM default = <b>automationhub</b>  Container default = <b>pulp</b>
<b>automationhub_pg_host</b>	<b>hub_pg_host</b>	Required if not using an internal database.  The hostname of the remote PostgreSQL database used by automation hub.  Default = <b>127.0.0.1</b>
<b>automationhub_pg_password</b>	<b>hub_pg_password</b>	Required if not using an internal database.  The password for the automation hub PostgreSQL database.  Use of special characters for this variable is limited. The <b>!, #, 0</b> and <b>@</b> characters are supported. Use of other special characters can cause the setup to fail.
<b>automationhub_pg_port</b>	<b>hub_pg_port</b>	Required if not using an internal database.  Default = <b>5432</b>
<b>automationhub_pg_sslmode</b>		<i>Required</i>  Default = <b>prefer</b>
<b>automationhub_pg_username</b>	<b>hub_pg_username</b>	The username for your automation hub PostgreSQL database  RPM default = <b>automationhub</b>  Container default = <b>pulp</b>

RPM variable name	Container variable name	Description
<b>automationhub_require_content_approval</b>		<p><i>Optional</i></p> <p>Value is <b>true</b> if automation hub enforces the approval mechanism before collections are made available.</p> <p>By default when you upload collections to automation hub, an administrator must approve it before they are made available to the users.</p> <p>If you want to disable the content approval flow, set the variable to <b>false</b>.</p> <p>Default = <b>true</b></p>
<b>automationhub_seed_collections</b>	<b>hub_seed_collections</b>	<p>A Boolean that defines whether or not pre-loading of collections is enabled. When you run the bundle installer, validated content is uploaded to the <b>validated</b> repository, and certified content is uploaded to the <b>rh-certified</b> repository. By default, certified content and validated content are both uploaded.</p> <p>If you do not want to install content, set this variable to <b>false</b> to disable the seeding.</p> <p>For the RPM-based installer, if you only want one type of content, set this variable to <b>true</b> and set the <b>automationhub_collection_seed_repository</b> variable to the type of content you want to include.</p> <p>Default = <b>true</b></p>
<b>automationhub_ssl_cert</b>	<b>hub_tls_cert</b>	<p><i>Optional</i></p> <p><b>/path/to/automationhub.cert</b></p> <p>Same as <b>web_server_ssl_cert</b> but for automation hub UI and API.</p>

RPM variable name	Container variable name	Description
<b>automationhub_ssl_key</b>	<b>hub_tls_key</b>	<p><i>Optional</i></p> <p><b>/path/to/automationhub.key.</b></p> <p>Same as <b>web_server_ssl_key</b> but for automation hub UI and API.</p>
<b>automationhub_user_headers</b>		<p>List of nginx headers for Ansible automation hub's web server.</p> <p>Each element in the list is provided to the web server's nginx configuration as a separate line.</p> <p>Default = empty list</p>
<b>ee_from_hub_only</b>		<p>When deployed with automation hub, the installer pushes execution environment images to automation hub and configures automation controller to pull images from the automation hub registry.</p> <p>To make automation hub the only registry to pull execution environment images from, set this variable to <b>true</b>.</p> <p>If set to <b>false</b>, execution environment images are also taken directly from Red Hat.</p> <p>Default = <b>true</b> when the bundle installer is used.</p>
<b>generate_automationhub_token</b>		<p>When performing a fresh installation, a new token will automatically be generated by default. If you want the installer to regenerate a new token, set <b>generate_automationhub_token=true</b> and the installer will use it in the installation process.</p>

RPM variable name	Container variable name	Description
<b>nginx_hsts_max_age</b>	<b>hub_nginx_hsts_max_age</b>	This variable specifies how long, in seconds, the system should be considered as an <i>HTTP Strict Transport Security</i> (HSTS) host. That is, how long HTTPS is used exclusively for communication.  Default = <b>63072000</b> seconds, or two years.
<b>pulp_db_fields_key</b>		Relative or absolute path to the Fernet symmetric encryption key that you want to import. The path is on the Ansible management node. It is used to encrypt certain fields in the database, such as credentials. If not specified, a new key will be generated.
	<b>hub_tls_remote</b>	Automation hub TLS remote files.  Default = <b>false</b>
	<b>hub_main_url</b>	Automation hub main URL.
	<b>hub_nginx_client_max_body_size</b>	NGINX maximum body size.  Default = <b>20m</b>
	<b>hub_nginx_http_port</b>	NGINX HTTP port.  Default = <b>8081</b>
	<b>hub_nginx_https_port</b>	NGINX HTTPS port.  Default = <b>8444</b>
	<b>hub_nginx_https_protocols</b>	NGINX HTTPS protocols.  Default = <b>[TLSv1.2, TLSv1.3]</b>
	<b>hub_pg_socket</b>	PostgreSQL Automation hub UNIX socket.
	<b>hub_secret_key</b>	Automation hub secret key.
	<b>hub_storage_backend</b>	Automation hub storage backend.
	<b>hub_workers</b>	Automation hub workers count.

RPM variable name	Container variable name	Description
	<b>hub_collection_signing</b>	Enable Automation hub collection signing.  Default = <b>false</b>
	<b>hub_container_signing</b>	Enable Automation hub container signing.  Default = <b>false</b>
	<b>hub_container_signing_pass</b>	Automation hub container signing passphrase.
	<b>hub_collection_signing_passes</b>	Automation hub collection signing passphrase.
	<b>hub_postinstall</b>	Enable Automation hub postinstall.  Default = <b>false</b>
	<b>hub_postinstall_async_delay</b>	Postinstall delay between retries.  Default = <b>1</b>
	<b>hub_postinstall_async_retries</b>	Postinstall number of retries to perform.  Default = <b>30</b>
	<b>hub_postinstall_dir</b>	Automation hub postinstall directory.
	<b>hub_postinstall_ignore_files</b>	Automation hub ignore files.
	<b>hub_postinstall_repo_ref</b>	Automation hub repository branch or tag.  Default = <b>main</b>
	<b>hub_postinstall_repo_url</b>	Automation hub repository URL.



RPM variable name	Container variable name	Description
	<b>hub_shared_data_path</b>	<p>Required when installing more than one instance of automation hub with a file storage backend. When installing a single instance of automation hub, it is optional.</p> <p>Path to a Network File System (NFS) share with read, write, and execute (RWX) access.</p>
	<b>hub_shared_data_mount_opts</b>	<p><i>Optional</i></p> <p>Mount options for NFS share.</p> <p>Default = <b>rw,sync,hard</b></p>

### B.3. AUTOMATION CONTROLLER VARIABLES

RPM variable name	Container variable name	Description
<b>admin_email</b>		The email address used for the admin user for automation controller.
<b>admin_password</b>	<b>controller_admin_password</b>	<p><i>Required</i></p> <p>Automation controller admin password.</p> <p>Passwords must be enclosed in quotes when they are provided in plain text in the inventory file.</p> <p>Use of special characters for this variable is limited. The password can include any printable ASCII character except <code>/</code>, <code>”</code>, or <code>@</code>.</p>
<b>admin_username</b>	<b>controller_admin_user</b>	<p>Automation controller admin user.</p> <p>Default = <b>admin</b></p>
<b>automation_controller_main_url</b>		Automation controller main URL.
<b>controller_tls_files_remote</b>	<b>controller_tls_remote</b>	<p>Automation controller TLS remote files.</p> <p>Default = <b>false</b></p>

RPM variable name	Container variable name	Description
<b>nginx_disable_hsts</b>	<b>controller_nginx_disable_hsts</b>	Disable NGINX HTTP Strict Transport Security (HSTS).  Default = <b>false</b>
<b>nginx_disable_https</b>	<b>controller_nginx_disable_https</b>	Disable NGINX HTTPS.  Default = <b>false</b>
<b>nginx_hsts_max_age</b>	<b>controller_nginx_hsts_max_age</b>	This variable specifies how long, in seconds, the system must be considered as an <i>HTTP Strict Transport Security</i> (HSTS) host. That is, how long HTTPS is used only for communication.  Default = <b>63072000</b> seconds, or two years.
<b>nginx_http_port</b>	<b>controller_nginx_http_port</b>	The NGINX HTTP server listens for inbound connections.  RPM default = <b>80</b>  Container default = <b>8080</b>
<b>nginx_https_port</b>	<b>controller_nginx_https_port</b>	The NGINX HTTPS server listens for secure connections.  RPM Default = <b>443</b>  Container default = <b>8443</b>
<b>nginx_user_headers</b>	<b>controller_nginx_user_headers</b>	List of NGINX headers for the automation controller web server.  Each element in the list is provided to the web server's NGINX configuration as a separate line.  Default = empty list

RPM variable name	Container variable name	Description
<b>node_state</b>		<p><i>Optional</i></p> <p>The status of a node or group of nodes. Valid options are <b>active</b>, <b>deprovision</b> to remove a node from a cluster, or <b>iso_migrate</b> to migrate a legacy isolated node to an execution node.</p> <p>Default = <b>active</b></p>
<b>node_type</b>		<p>For <b>[automationcontroller]</b> group.</p> <p>Two valid <b>node_types</b> can be assigned for this group.</p> <p>A <b>node_type=control</b> means that the node only runs project and inventory updates, but not regular jobs.</p> <p>A <b>node_type=hybrid</b> can run everything.</p> <p>Default for this group = <b>hybrid</b>.</p> <p>For <b>[execution_nodes]</b> group:</p> <p>Two valid <b>node_types</b> can be assigned for this group.</p> <p>A <b>node_type=hop</b> implies that the node forwards jobs to an execution node.</p> <p>A <b>node_type=execution</b> implies that the node can run jobs.</p> <p>Default for this group = <b>execution</b>.</p>

RPM variable name	Container variable name	Description
<b>peers</b>		<p><i>Optional</i></p> <p>The <b>peers</b> variable is used to indicate which nodes a specific host or group connects to. Wherever this variable is defined, an outbound connection to the specific host or group is established.</p> <p>This variable is used to add <b>tcp-peer</b> entries in the <b>receptor.conf</b> file used for establishing network connections with other nodes.</p> <p>The peers variable can be a comma-separated list of hosts and groups from the inventory. This is resolved into a set of hosts that is used to construct the <b>receptor.conf</b> file.</p>
<b>pg_database</b>	<b>controller_pg_database</b>	<p>The name of the PostgreSQL database.</p> <p>Default = <b>awx</b></p>
<b>pg_host</b>	<b>controller_pg_host</b>	<p><i>Required</i></p> <p>The PostgreSQL host, which can be an externally managed database.</p>

RPM variable name	Container variable name	Description
<b>pg_password</b>	<b>controller_pg_password</b>	<p><i>Required</i></p> <p>The password for the PostgreSQL database.</p> <p>Use of special characters for this variable is limited. The <b>!, #, 0</b> and <b>@</b> characters are supported. Use of other special characters can cause the setup to fail.</p> <p>NOTE</p> <p>You no longer have to provide a <b>pg_hashed_password</b> in your inventory file at the time of installation, because PostgreSQL 13 can now store user passwords more securely.</p> <p>When you supply <b>pg_password</b> in the inventory file for the installer, PostgreSQL uses the SCRAM-SHA-256 hash to secure that password as part of the installation process.</p>
<b>pg_port</b>	<b>controller_pg_port</b>	<p>The PostgreSQL port to use.</p> <p>Default = <b>5432</b></p>
<b>pg_username</b>	<b>controller_pg_username</b>	<p>Your PostgreSQL database username.</p> <p>Default = <b>awx</b>.</p>
<b>web_server_ssl_cert</b>	<b>controller_tls_cert</b>	<p><i>Optional</i></p> <p><b>/path/to/webserver.cert</b></p> <p>Same as <b>automationhub_ssl_cert</b> but for web server UI and API.</p>
<b>web_server_ssl_key</b>	<b>controller_tls_key</b>	<p><i>Optional</i></p> <p><b>/path/to/webserver.key</b></p> <p>Same as <b>automationhub_server_ssl_key</b> but for web server UI and API.</p>

RPM variable name	Container variable name	Description
	<b>controller_event_workers</b>	Automation controller event workers.  Default = <b>4</b>
	<b>controller_license_file</b>	The location of your automation controller license file.  For example:  <b>controller_license_file=/path/to/license.zip</b>  If you are defining this variable as part of the postinstall process ( <b>controller_postinstall = true</b> ), then you need to also set the <b>controller_postinstall_dir</b> variable.
	<b>controller_nginx_client_max_body_size</b>	NGINX maximum body size.  Default = <b>5m</b>
	<b>controller_nginx_https_protocols</b>	NGINX HTTPS protocols.  Default = <b>[TLSv1.2, TLSv1.3]</b>
	<b>controller_pg_socket</b>	PostgreSQL Controller UNIX socket.
	<b>controller_secret_key</b>	Automation controller secret key.
	<b>controller_uwsgi_listen_queue_size</b>	Automation controller uWSGI listen queue size.  Default = <b>2048</b>
	<b>controller_postinstall</b>	Enable or disable the postinstall feature of the containerized installer.  If set to <b>true</b> , then you also need to set <b>controller_license_file</b> and <b>controller_postinstall_dir</b> .  Default = <b>false</b>
	<b>controller_postinstall_dir</b>	The location of your automation controller postinstall directory.

RPM variable name	Container variable name	Description
	<b>controller_postinstall_async_delay</b>	Postinstall delay between retries.  Default = <b>1</b>
	<b>controller_postinstall_async_retries</b>	Postinstall number of tries to attempt.  Default = <b>30</b>
	<b>controller_postinstall_ignore_files</b>	Automation controller ignore files.
	<b>controller_postinstall_repo_ref</b>	Automation controller repository branch or tag.  Default = <b>main</b>
	<b>controller_postinstall_repo_url</b>	Automation controller repository URL.

## B.4. EVENT-DRIVEN ANSIBLE CONTROLLER VARIABLES

RPM variable name	Container variable name	Description
<b>automationedacontroller_activation_workers</b>	<b>eda_activation_workers</b>	<i>Optional</i>  Number of workers for ansible-rulebook activation pods in Event-Driven Ansible.  Default = (# of cores or threads) * 2 + 1
<b>automationedacontroller_admin_email</b>	<b>eda_admin_email</b>	<i>Optional</i>  Email address used by Django for the admin user for Event-Driven Ansible controller.  Default = <b>admin@example.com</b>

RPM variable name	Container variable name	Description
<b>automationedacontroller_admin_password</b>	<b>eda_admin_password</b>	<p><i>Required</i></p> <p>The admin password used by the Event-Driven Ansible controller instance.</p> <p>Passwords must be enclosed in quotes when they are provided in plain text in the <b>inventory</b> file.</p> <p>Use of special characters for this variable is limited. The password can include any printable ASCII character except <code>/</code>, <code>”</code>, or <code>@</code>.</p>
<b>automationedacontroller_admin_username</b>	<b>eda_admin_user</b>	<p>Username used by Django to identify and create the admin superuser in Event-Driven Ansible controller.</p> <p>Default = <b>admin</b></p>
<b>automationedacontroller_allowed_hostnames</b>		<p>List of additional addresses to enable for user access to Event-Driven Ansible controller.</p> <p>Default = empty list</p>
<b>automationedacontroller_controller_verify_ssl</b>		<p>Boolean flag used to verify automation controller’s web certificates when making calls from Event-Driven Ansible controller. Verified is <b>true</b> and not verified is <b>false</b>.</p> <p>Default = <b>false</b></p>
<b>automationedacontroller_disable_hsts</b>	<b>eda_nginx_disable_hsts</b>	<p><i>Optional</i></p> <p>Boolean flag to disable HSTS for Event-Driven Ansible controller.</p> <p>Default = <b>false</b></p>
<b>automationedacontroller_disable_https</b>	<b>eda_nginx_disable_https</b>	<p><i>Optional</i></p> <p>Boolean flag to disable HTTPS for Event-Driven Ansible controller.</p> <p>Default = <b>false</b></p>



RPM variable name	Container variable name	Description
<b>automationedacontroller_event_stream_path</b>	<b>eda_event_stream_prefix_path</b>	API prefix path used for Event-Driven Ansible event-stream through platform gateway.  Default = <b>/eda-event-streams</b>
<b>automationedacontroller_gunicorn_workers</b>	<b>eda_gunicorn_workers</b>	Number of workers for the API served through Gunicorn.  Default = (# of cores or threads) * 2 + 1
<b>automationedacontroller_max_running_activations</b>	<b>eda_max_running_activations</b>	<i>Optional</i>  The number of maximum activations running concurrently per node.  This is an integer that must be greater than 0.  Default = <b>12</b>
<b>automationedacontroller_nginx_tls_files_remote</b>	<b>eda_tls_remote</b>	Boolean flag to specify whether cert sources are on the remote host (true) or local (false).  Default = <b>false</b>
<b>automationedacontroller_pg_database</b>	<b>eda_pg_database</b>	<i>Optional</i>  The PostgreSQL database used by Event-Driven Ansible controller.  RPM default = <b>automationedacontroller</b>  Container default = <b>eda</b>
<b>automationedacontroller_pg_password</b>	<b>eda_pg_password</b>	<i>Required</i>  The password for the PostgreSQL database used by Event-Driven Ansible controller.  Use of special characters for this variable is limited. The <b>!, #, 0</b> and <b>@</b> characters are supported. Use of other special characters can cause the setup to fail.

RPM variable name	Container variable name	Description
<b>automationedacontroller_pg_port</b>	<b>eda_pg_port</b>	<p><i>Optional</i></p> <p>The port number of the PostgreSQL database used by Event-Driven Ansible controller.</p> <p>Default = <b>5432</b></p>
<b>automationedacontroller_pg_username</b>	<b>eda_pg_username</b>	<p><i>Optional</i></p> <p>The username for your Event-Driven Ansible controller PostgreSQL database.</p> <p>RPM default = <b>automationedacontroller</b></p> <p>Container default = <b>eda</b></p>
<b>automationedacontroller_redis_host</b>	<b>eda_redis_host</b>	The Redis hostname used by Event-Driven Ansible controller.
<b>automationedacontroller_redis_port</b>	<b>eda_redis_port</b>	The port used for the Redis host defined by <b>automationedacontroller_redis_host</b> for Event-Driven Ansible controller.
<b>automationedacontroller_rq_workers</b>		<p>Number of Redis Queue (RQ) workers used by Event-Driven Ansible controller. RQ workers are Python processes that run in the background.</p> <p>Default = (# of cores or threads) * 2 + 1</p>
<b>automationedacontroller_ssl_cert</b>	<b>eda_tls_cert</b>	<p><i>Optional</i></p> <p><b>/root/ssl_certs/eda.&lt;example&gt;.com.crt</b></p> <p>Same as <b>automationhub_ssl_cert</b> but for Event-Driven Ansible controller UI and API.</p>

RPM variable name	Container variable name	Description
<b>automationedacontroller_ssl_key</b>	<b>eda_tls_key</b>	<p><i>Optional</i></p> <p><b>/root/ssl_certs/eda.&lt;example&gt;.com.key</b></p> <p>Same as <b>automationhub_server_ssl_key</b> but for Event-Driven Ansible controller UI and API.</p>
<b>automationedacontroller_user_headers</b>	<b>eda_nginx_user_headers</b>	<p>List of additional NGINX headers to add to Event-Driven Ansible controller's NGINX configuration.</p> <p>Default = empty list</p>
<b>automationedacontroller_pg_host</b>	<b>eda_pg_host</b>	<p><i>Required</i></p> <p>The hostname of the PostgreSQL database used by Event-Driven Ansible controller, which can be an externally managed database.</p>
<b>eda_node_type</b>	<b>eda_type</b>	<p><i>Optional</i></p> <p>Event-Driven Ansible controller node type.</p> <p>Default = <b>hybrid</b></p>
	<b>eda_debug</b>	<p>Event-Driven Ansible controller debug.</p> <p>Default = <b>false</b></p>
	<b>eda_event_stream_url</b>	<p>Event-Driven Ansible controller event stream URL.</p>
	<b>eda_main_url</b>	<p>Event-Driven Ansible controller main URL.</p>
	<b>eda_nginx_client_max_body_size</b>	<p>NGINX maximum body size.</p> <p>Default = <b>1m</b></p>
	<b>eda_nginx_hsts_max_age</b>	<p>NGINX HSTS maximum age.</p> <p>Default = <b>63072000</b></p>

RPM variable name	Container variable name	Description
	<b>eda_nginx_http_port</b>	NGINX HTTP port. Default = <b>8082</b>
	<b>eda_nginx_https_port</b>	NGINX HTTPS port. Default = <b>8445</b>
	<b>eda_nginx_https_protocols</b>	NGINX HTTPS protocols. Default = <b>[TLSv1.2, TLSv1.3]</b>
	<b>eda_pg_socket</b>	PostgreSQL Event-Driven Ansible UNIX socket.
	<b>eda_redis_disable_tls</b>	Disable TLS Redis (for many nodes). Default = <b>false</b>
	<b>eda_redis_password</b>	Redis Event-Driven Ansible controller password (for many nodes).
	<b>eda_redis_tls_cert</b>	<i>Optional</i> <b>/path/to/edaredis.crt</b> Location of the Event-Driven Ansible controller Redis TLS certificate.
	<b>eda_redis_tls_key</b>	<i>Optional</i> <b>/path/to/edaredis.key</b> Location of the Event-Driven Ansible controller Redis TLS key.
	<b>eda_redis_username</b>	Redis Event-Driven Ansible controller username (for many nodes).
	<b>eda_safe_plugins</b>	Event-Driven Ansible controller safe plugins.
	<b>eda_secret_key</b>	Event-Driven Ansible controller secret key.

RPM variable name	Container variable name	Description
	<b>eda_workers</b>	Event-Driven Ansible controller workers count.  Default = <b>2</b>

## B.5. PLATFORM GATEWAY VARIABLES

RPM variable name	Container variable name	Description
<b>automationgateway_admin_email</b>	<b>gateway_admin_email</b>	The email address used for the admin user for platform gateway.
<b>automationgateway_admin_password</b>	<b>gateway_admin_password</b>	<i>Required</i>  The admin password used to connect to the platform gateway instance.  Passwords must be enclosed in quotes when they are provided in plain text in the <b>inventory</b> file.  Use of special characters for this variable is limited. The password can include any printable ASCII character except <i>/</i> , <i>"</i> , or <i>@</i> .
<b>automationgateway_admin_username</b>	<b>gateway_admin_user</b>	<i>Optional</i>  The username used to identify and create the admin superuser in platform gateway.  Default = <b>admin</b>
<b>automationgateway_disable_hsts</b>	<b>gateway_nginx_disable_hsts</b>	<i>Optional</i>  Disable NGINX HSTS.  Default = <b>false</b>
<b>automationgateway_disable_https</b>	<b>gateway_nginx_disable_https</b>	<i>Optional</i>  Disable NGINX HTTPS.  Default = <b>false</b>

RPM variable name	Container variable name	Description
<b>automationgateway_grpc_auth_service_timeout</b>	<b>gateway_grpc_auth_service_timeout</b>	Platform gateway auth server timeout.  Default = <b>30s</b>
<b>automationgateway_grpc_server_max_threads_per_process</b>	<b>gateway_grpc_server_max_threads_per_process</b>	Platform gateway auth server threads per process.  Default = <b>10</b>
<b>automationgateway_grpc_server_processes</b>	<b>gateway_grpc_server_processes</b>	Platform gateway auth server processes  Default = <b>5</b>
<b>automationgateway_main_url</b>	<b>gateway_main_url</b>	<i>Optional</i>  The main platform gateway URL that clients will connect to (e.g. <b>https://&lt;gateway_node&gt;</b> ).  If not specified, the first the first node in the <b>[automationgateway]</b> group will be used when needed.
<b>automationgateway_pg_database</b>	<b>gateway_pg_database</b>	<i>Optional</i>  The PostgreSQL database used by platform gateway.  RPM default = <b>automationgateway</b>  Container default = <b>gateway</b>
<b>automationgateway_pg_host</b>	<b>gateway_pg_host</b>	<i>Required</i>  The hostname of the PostgreSQL database used by platform gateway, which can be an externally managed database.

RPM variable name	Container variable name	Description
<b>automationgateway_pg_password</b>	<b>gateway_pg_password</b>	<p><i>Required</i></p> <p>The password for the PostgreSQL database used by platform gateway.</p> <p>Use of special characters for <b>automationgateway_pg_password</b> is limited. The <b>!</b>, <b>#</b>, <b>0</b> and <b>@</b> characters are supported.</p> <p>Use of other special characters can cause the setup to fail.</p>
<b>automationgateway_pg_port</b>	<b>gateway_pg_port</b>	<p><i>Optional</i></p> <p>The port number of the PostgreSQL database used by platform gateway.</p> <p>Default = <b>5432</b></p>
<b>automationgateway_pg_sslmode</b>	<b>gateway_pg_sslmode</b>	<p>Choose one of the two available modes: <b>prefer</b> and <b>verify-full</b>.</p> <p>Set to <b>verify-full</b> for client-side enforced SSL.</p> <p>Default = <b>prefer</b></p>
<b>automationgateway_pg_username</b>	<b>gateway_pg_username</b>	<p><i>Optional</i></p> <p>The username for your platform gateway PostgreSQL database.</p> <p>RPM default = <b>automationgateway</b></p> <p>Container default = <b>gateway</b></p>
<b>automationgateway_redis_host</b>	<b>gateway_redis_host</b>	<p>The Redis hostname used by platform gateway.</p>
<b>automationgateway_redis_port</b>	<b>gateway_redis_port</b>	<p>The Redis platform gateway port.</p> <p>Default = <b>6379</b></p>

RPM variable name	Container variable name	Description
<b>automationgateway_ssl_cert</b>	<b>gateway_tls_cert</b>	<i>Optional</i>  <b>/path/to/automationgateway.cert</b>  Same as <b>automationhub_ssl_cert</b> but for platform gateway UI and API.
<b>automationgateway_ssl_key</b>	<b>gateway_tls_key</b>	<i>Optional</i>  <b>/path/to/automationgateway.key</b>  Same as <b>automationhub_server_ssl_key</b> but for platform gateway UI and API.
	<b>gateway_nginx_client_max_body_size</b>	NGINX maximum body size.  Default = <b>5m</b>
	<b>gateway_nginx_hsts_max_age</b>	NGINX HSTS maximum age.  Default = <b>63072000</b>
	<b>gateway_nginx_http_port</b>	NGINX HTTP port.
	<b>gateway_nginx_https_port</b>	NGINX HTTPS port.
	<b>gateway_nginx_https_protocols</b>	NGINX HTTPS protocols.  Default = <b>[TLSv1.2, TLSv1.3]</b>
	<b>gateway_nginx_user_headers</b>	Custom NGINX headers.
	<b>gateway_redis_disable_tls</b>	Disable TLS Redis.  Default = <b>false</b>
	<b>gateway_redis_password</b>	Redis platform gateway password.



RPM variable name	Container variable name	Description
	<b>gateway_redis_tls_cert</b>	<i>Optional</i> <b>/path/to/gatewayredis.crt</b> Location of the platform gateway Redis TLS certificate.
	<b>gateway_redis_tls_key</b>	<i>Optional</i> <b>/path/to/gatewayredis.key</b> Location of the platform gateway Redis TLS key.
	<b>gateway_redis_username</b>	Redis platform gateway username. Default = <b>gateway</b>
	<b>gateway_secret_key</b>	Platform gateway secret key.
	<b>gateway_tls_remote</b>	Platform gateway TLS remote files. Default = <b>false</b>
	<b>gateway_uwsgi_listen_queue_size</b>	Platform gateway uWSGI listen queue size. Default = <b>4096</b>

## B.6. DATABASE VARIABLES

RPM variable name	Container variable name	Description
<b>pg_ssl_mode</b>		Choose one of the two available modes: <b>prefer</b> and <b>verify-full</b> .  Set to <b>verify-full</b> for client-side enforced SSL/TLS.  Default = <b>prefer</b>
<b>postgres_ssl_cert</b>	<b>postgresql_tls_cert</b>	Location of the PostgreSQL SSL/TLS certificate.  <b>/path/to/pgsql_ssl.cert</b>

RPM variable name	Container variable name	Description
<b>postgres_ssl_key</b>	<b>postgresql_tls_key</b>	Location of the PostgreSQL SSL/TLS key.  <b>/path/to/pgsql_ssl.key</b>
<b>postgres_use_cert</b>		Location of the PostgreSQL user certificate.  <b>/path/to/pgsql.crt</b>
<b>postgres_use_ssl</b>	<b>postgresql_disable_tls</b>	Determines if the connection between Ansible Automation Platform and the PostgreSQL database should use SSL/TLS. The default for this variable is <b>false</b> which means SSL/TLS is not used for PostgreSQL connections. When set to <b>true</b> , the platform connects to PostgreSQL by using SSL/TLS.
<b>postgres_max_connections</b>	<b>postgresql_max_connections</b>	Maximum database connections setting to apply if you are using installer-managed PostgreSQL.  See <a href="#">PostgreSQL database configuration and maintenance for automation controller</a> for help selecting a value.  Default = <b>1024</b>
	<b>postgresql_admin_database</b>	PostgreSQL admin database.  Default = <b>postgres</b>
	<b>postgresql_admin_username</b>	PostgreSQL admin user.  Default = <b>postgres</b>
	<b>postgresql_admin_password</b>	<i>Required</i>  PostgreSQL admin password.
	<b>postgresql_effective_cache_size</b>	PostgreSQL effective cache size.
	<b>postgresql_keep_databases</b>	Keep databases during uninstall.  Default = <b>false</b>

RPM variable name	Container variable name	Description
	<b>postgresql_log_destination</b>	PostgreSQL log file location. Default = <b>/dev/stderr</b>
	<b>postgresql_password_encryption</b>	PostgreSQL password encryption. Default = <b>scram-sha-256</b>
	<b>postgresql_shared_buffers</b>	PostgreSQL shared buffers.
	<b>postgresql_tls_remote</b>	PostgreSQL TLS remote files. Default = <b>false</b>
	<b>postgresql_port</b>	PostgreSQL port number. Default = <b>5432</b>

## B.7. IMAGE VARIABLES

RPM variable name	Container variable name	Description
	<b>controller_image</b>	Automation controller image. Default = <b>controller-rhel8:latest</b>
	<b>de_extra_images</b>	Decision environment extra images.
	<b>de_supported_image</b>	Decision environment supported image. Default = <b>de-supported-rhel8:latest</b>
	<b>eda_image</b>	Event-Driven Ansible image. Default = <b>eda-controller-rhel8:latest</b>
	<b>eda_web_image</b>	Event-Driven Ansible web image. Default = <b>eda-controller-ui-rhel8:latest</b>

RPM variable name	Container variable name	Description
	<b>ee_29_enabled</b>	Enable execution environment 29. Default = <b>false</b>
	<b>ee_29_image</b>	Execution environment 29 image. Default = <b>ee-29-rhel8:latest</b>
	<b>ee_extra_images</b>	Execution environment extra images.
	<b>ee_minimal_image</b>	Execution environment minimal image. Default = <b>ee-minimal-rhel8:latest</b>
	<b>ee_supported_image</b>	Execution environment supported image. Default = <b>ee-supported-rhel8:latest</b>
	<b>hub_image</b>	Automation hub image. Default = <b>hub-rhel8:latest</b>
	<b>hub_web_image</b>	Automation hub web image. Default = <b>hub-web-rhel8:latest</b>
	<b>postgresql_image</b>	PostgreSQL image. Default = <b>postgresql-15:latest</b>
	<b>receptor_image</b>	Receptor image. Default = <b>receptor-rhel8:latest</b>
	<b>redis_image</b>	Redis image. Default = <b>redis-6:latest</b>
	<b>pcp_image</b>	Performance Co-Pilot image. Default = <b>rhel8-pcp:latest</b>

## B.8. RECEPTOR VARIABLES

RPM variable name	Container variable name	Description
	<b>receptor_disable_signing</b>	Disable receptor signing. Default = <b>false</b>
	<b>receptor_disable_tls</b>	Disable receptor TLS. Default = <b>false</b>
	<b>receptor_log_level</b>	Receptor logging level. Default = <b>info</b>
	<b>receptor_mintls13</b>	Receptor TLS 1.3 minimal. Default = <b>false</b>
	<b>receptor_peers</b>	Receptor peers list.
<b>receptor_datadir</b>		<p>This variable configures the receptor data directory. By default, it is set to <b>/tmp/receptor</b>. To change the default location, run the installation script with "<b>-e receptor_datadir=</b>" and specify the target directory that you want.</p> <p><b>NOTES</b></p> <ul style="list-style-type: none"> <li>* The target directory must be accessible to <b>awx</b> users.</li> <li>* If the target directory is a temporary file system <b>tmpfs</b>, ensure it is remounted correctly after a reboot. Failure to do so results in the receptor no longer having a working directory.</li> </ul>
<b>receptor_listener_port</b>	<b>receptor_port</b>	Receptor port number. Default = <b>27199</b>
<b>receptor_listener_protocol</b>	<b>receptor_protocol</b>	Receptor protocol. Default = <b>tcp</b>
	<b>receptor_signing_private_key</b>	Receptor signing private key.

RPM variable name	Container variable name	Description
	<b>receptor_signing_public_key</b>	Receptor signing public key.
	<b>receptor_signing_remote</b>	Receptor signing remote files. Default = <b>false</b>
	<b>receptor_tls_cert</b>	Receptor TLS certificate.
	<b>receptor_tls_key</b>	Receptor TLS key.
	<b>receptor_tls_remote</b>	Receptor TLS remote files. Default = <b>false</b>
	<b>receptor_type</b>	Receptor node type. Default = <b>execution</b>

## B.9. ANSIBLE VARIABLES

The following variables control how Ansible Automation Platform interacts with remote hosts.

For more information about variables specific to certain plugins, see the documentation for [Ansible.Builtin](#).

For a list of global configuration options, see [Ansible Configuration Settings](#).

Variable	Description
<b>ansible_connection</b>	The connection plugin used for the task on the target host.  This can be the name of any of Ansible connection plugins. SSH protocol types are <b>smart</b> , <b>ssh</b> or <b>paramiko</b> .  Default = <b>smart</b>
<b>ansible_host</b>	The IP or name of the target host to use instead of <b>inventory_hostname</b> .
<b>ansible_port</b>	The connection port number.  Default: 22 for SSH
<b>ansible_user</b>	The user name to use when connecting to the host.

Variable	Description
<b>ansible_password</b>	<p>The password to authenticate to the host.</p> <p>Never store this variable in plain text.</p> <p>Always use a vault.</p>
<b>ansible_ssh_private_key_file</b>	Private key file used by SSH. Useful if using multiple keys and you do not want to use an SSH agent.
<b>ansible_ssh_common_args</b>	This setting is always appended to the default command line for <b>sftp</b> , <b>scp</b> , and <b>ssh</b> . Useful to configure a ProxyCommand for a certain host or group.
<b>ansible_sftp_extra_args</b>	This setting is always appended to the default <b>sftp</b> command line.
<b>ansible_scp_extra_args</b>	This setting is always appended to the default <b>scp</b> command line.
<b>ansible_ssh_extra_args</b>	This setting is always appended to the default <b>ssh</b> command line.
<b>ansible_ssh_pipelining</b>	Determines if SSH <b>pipelining</b> is used. This can override the <b>pipelining</b> setting in <b>ansible.cfg</b> . If using SSH key-based authentication, the key must be managed by an SSH agent.
<b>ansible_ssh_executable</b>	<p>Added in version 2.2.</p> <p>This setting overrides the default behavior to use the system SSH. This can override the <b>ssh_executable</b> setting in <b>ansible.cfg</b>.</p>
<b>ansible_shell_type</b>	The shell type of the target system. Do not use this setting unless you have set the <b>ansible_shell_executable</b> to a non-Bourne (sh) compatible shell. By default commands are formatted using sh-style syntax. Setting this to <b>csh</b> or <b>fish</b> causes commands executed on target systems to follow the syntax of those shells instead.
<b>ansible_shell_executable</b>	<p>This sets the shell that the Ansible controller uses on the target machine and overrides the executable in <b>ansible.cfg</b> which defaults to <b>/bin/sh</b>.</p> <p>Do not change this variable unless <b>/bin/sh</b> is not installed on the target machine or cannot be run from sudo.</p>

Variable	Description
<b>inventory_hostname</b>	<p>This variable takes the hostname of the machine from the inventory script or the Ansible configuration file.</p> <p>You cannot set the value of this variable.</p> <p>Because the value is taken from the configuration file, the actual runtime hostname value can vary from what is returned by this variable.</p>