Private Partner Automation Hub 2.4

Private Partner Automation Hub Installation Guide

Private Partner Automation Hub Installation Guide

Last Updated: 2023-12-11

Private Partner Automation Hub 2.4 Private Partner Automation Hub Installation Guide

Private Partner Automation Hub Installation Guide

Legal Notice

Abstract

Private Partner Automation Hub Installation Guide

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you have a suggestion to improve this documentation, or find an error, please contact technical support at https://access.redhat.com to create an issue on the Ansible Automation Platform Jira project using the **docs-product** component.

CHAPTER 1. PRIVATE PARTNER AUTOMATION HUB INSTALLATION GUIDE

Red Hat Ansible private partner automation hub enables Red Hat partners to curate, host, and serve their own Ansible content to their consultants, customers, or users. Private partner automation hub gives partners the ability to serve content to multiple different parties simultaneously using comprehensive role-based access controls. This offering is designed to help partners expedite their solutions using Ansible's proven automation technology.

This guide contains a collection of Ansible playbooks that install a containerized private partner automation hub. You can choose from two types of installation based on your organization's needs:

- a development option
- a production option

Prerequisites

- A RHEL 9.2 based host. A minimal OS base install is recommended.
- A non-root user for the RHEL host, with sudo or other Ansible supported privilege escalation (sudo recommended). This user is responsible for the installation of the private partner automation hub.
- It is recommended to set up an **SSH public key authentication**for the non-root user. For guidelines on setting up an 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**, which does not require SSH.

1.1. SYSTEM REQUIREMENTS

Your system must meet the following minimum system requirements to install and run private partner automation hub.

Memory	8Gb RAM at minimum
CPU	2 CPU at minimum
Disk space	60Gb, with a minimum of 40Gb dedicated to /**var** for collection storage
Disk IOPs	1500

1.2. PREPARING THE RHEL HOST FOR CONTAINERIZED INSTALLATION

Procedure

Containerized Ansible Automation Platform runs the component services as podman-based containers on top of a RHEL host. The installer takes care of this once the underlying host has been prepared. Use the following instructions to prepare the host.

1. Log into your RHEL host as your non-root user.
2. Run **dnf repolist** to validate only the BaseOS and appstream repos are setup and enabled on the host:

```
$ 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)
```

3. Ensure that these repos and only these repos are available to the host OS. If you need to know how to do this use this guide: Chapter 10. Managing custom software repositories Red Hat Enterprise Linux
4. Ensure that the host has DNS configured and can resolve hostnames and IPs using a fully qualified domain name (FQDN). This is essential to ensure services can talk to one another.
   To configure unbound DNS refer to Chapter 2. Setting up an unbound DNS server Red Hat Enterprise Linux 9.

   To configure DNS using BIND refer to Chapter 1. Setting up and configuring a BIND DNS server Red Hat Enterprise Linux 9.

1.3. INSTALLING ANSIBLE-CORE

Procedure

1. Install ansible-core and other tools:

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

2. Set a fully qualified hostname:

```
sudo hostnamectl set-hostname your-FQDN-hostname
```

1.4. DOWNLOADING ANSIBLE AUTOMATION PLATFORM

Procedure

1. Download the latest installer tarball from access.redhat.com. This can be done directly within the RHEL host, which saves time.
2. If you have downloaded the tarball and optional manifest zip file onto your laptop, copy them onto your RHEL host.
   Decide where you would like the installer to reside on the filesystem. Installation related files will be created under this location and require at least 10Gb for the initial installation.

3. Unpack the installer tarball into your installation directory, and cd into the unpacked directory.

   a. online installer

   ```
   $ tar xfvz ansible-automation-platform-partner-hub-setup-bundle-latest.tar.gz
   ```

   b. bundled installer

   ```
   $ tar xfvz ansible-automation-platform-partner-hub-setup-bundle-xxx-<arch name>.tar.gz
   ```

   Ansible collections will already be installed inside the directory called **collections**. You will have to set **ANSIBLE_COLLECTIONS_PATH** environment variable to the directory path to consume the ansible collections.

4. Set **ANSIBLE_COLLECTIONS_PATH**:

```
$ export ANSIBLE_COLLECTIONS_PATH=/path/to/ansible-automation-platform-partner-hub-setup-
bundle-xxx/collections
```

1.5. ONLINE AND OFFLINE INSTALLATION

There are two ways to run the containerized installer:

- Online: The container images will be pulled from a registry.
- Offline: The container images will be imported from a tarball.

This workflow is controlled through the **bundle_install** variable.

1.5.1. Online installation

Online installation is the default install scenario (**bundle_install: false**) and pulls the container images from **registry.redhat.io**, which requires authentication.

Procedure

1. Provide the registry username and password for authentication.

```
registry_username: foo@ansible.com
registry_password: bar
```

1.5.2. Offline installation

Procedure

1. Set **bundle_install: true** in in your Ansible configuration and **bundle_dir`** with the path to the bundle directory without the images subdirectory:

```
$ ansible-playbook -i </path/to/inventory> ansible.containerized_installer.install -e bundle_install=true
-e bundle_dir=$(pwd)/bundle
```

1.6. INSTALLING AUTOMATION HUB FOR DEVELOPMENT

Use the playbook **install_devel** to install an all-in-one automation hub for development.

NOTE

If you do not set the **hub_admin_password** parameter, the installer sets one and prints it as part of its output.

Procedure

1. Run the installation playbook using the following command:

```
ansible-playbook -i </path/to/inventory>
ansible.private_partner_automation_hub_installer.install_devel
```

NOTE

The development option installs automation hub and the database on the same host, so there is no need to assign the host to a group in the inventory file.

1.7. INSTALLING A PRODUCTION AUTOMATION HUB

Use the playbook **install_prod** to install a production-ready, highly available automation hub. This option requires the user to manage all supporting services.

Prerequisites

- External PostgreSQL server
- LDAP server
- NFS server

The playbook requires the following parameters. For more detail, refer to the **production-sample-values** template in the tarball.

- PostgreSQL
- LDAP: **hub_authentication_backend** must be set to **ldap**
- NFS

Procedure

1. Run the installation playbook using the following command:

```
ansible-playbook -i </path/to/inventory> -e @</path/to/values>
ansible.private_partner_automation_hub_installer.install_prod
```

NOTE

The production option mandates a three-node installation, and installs automation hub on all hosts, so there is no need to assign the nodes to a group in the inventory file.

1.8. ACCESSING THE DEVELOPMENT HUB

The protocol and ports default values are:

- https protocol
- ports 80/443

The automation hub UI is available by default at `https://<hub node>`