Managing containers in private automation hub

Manage container image repositories with automation hub container registry
Red Hat Ansible Automation Platform 2.4 Managing containers in private automation hub

Manage container image repositories with automation hub container registry
Abstract

This guide provides administrator workflows and processes for configuring private automation hub container registry and repositories.
# Table of Contents

**WASSMAKING OPEN SOURCE MORE INCLUSIVE** .................................................. 3

**PROVIDING FEEDBACK ON RED HAT DOCUMENTATION** ................................ 4

**CHAPTER 1. MANAGING YOUR PRIVATE AUTOMATION HUB CONTAINER REGISTRY** .......... 5
  1.1. CONTAINER REGISTRIES .................................................. 5

**CHAPTER 2. CONFIGURING USER ACCESS FOR CONTAINER REPOSITORIES IN PRIVATE AUTOMATION HUB** ................................................................. 6
  2.1. PREREQUISITES .................................................. 6
  2.2. CONTAINER REGISTRY GROUP PERMISSIONS ................. 6
  2.3. CREATING A NEW GROUP ........................................ 6
  2.4. ASSIGNING PERMISSIONS TO GROUPS ....................... 7
  2.5. ADDING USERS TO GROUPS .................................. 8

**CHAPTER 3. POPULATING YOUR PRIVATE AUTOMATION HUB CONTAINER REGISTRY** .......... 9
  3.1. PREREQUISITES .................................................. 9
  3.2. OBTAINING IMAGES FOR USE IN AUTOMATION HUB ......... 9
  3.3. TAGGING IMAGES FOR USE IN AUTOMATION HUB ........ 10
  3.4. PUSHING A CONTAINER IMAGE TO PRIVATE AUTOMATION HUB ........................................ 10

**CHAPTER 4. SETTING UP YOUR CONTAINER REPOSITORY** .................................. 12
  4.1. PREREQUISITES .................................................. 12
  4.2. ADDING A README TO YOUR CONTAINER REPOSITORY ........ 12
  4.3. PROVIDING ACCESS TO YOUR CONTAINER REPOSITORY .... 12
  4.4. TAGGING CONTAINER IMAGES .................................. 13
  4.5. CREATING A CREDENTIAL IN AUTOMATION CONTROLLER .......... 13

**CHAPTER 5. PULLING IMAGES FROM A CONTAINER REPOSITORY** .......................... 15
  5.1. PREREQUISITES .................................................. 15
  5.2. PULLING AN IMAGE ........................................... 15
  5.3. SYNCING IMAGES FROM A CONTAINER REPOSITORY .......... 15
  5.4. ADDITIONAL RESOURCES ...................................... 16

**CHAPTER 6. WORKING WITH SIGNED CONTAINERS** ........................................ 17
  6.1. DEPLOYING YOUR SYSTEM FOR CONTAINER SIGNING ........ 17
  6.2. ADDING CONTAINERS REMOTELY TO AUTOMATION HUB .... 17
  6.3. ADDING AN EXECUTION ENVIRONMENT ......................... 18
  6.4. PUSHING CONTAINER IMAGES FROM YOUR LOCAL ........... 18
  6.5. USING POLICIES WITH SIGNED IMAGES ....................... 19
    6.5.1. Using podman to ensure an image is signed by a specific signature ......... 19
  6.6. CONFIGURING THE CLIENT TO VERIFY SIGNATURES .......... 20

**CHAPTER 7. DELETING A CONTAINER REPOSITORY** ........................................ 22
  7.1. DELETING A CONTAINER REPOSITORY .......................... 22
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

We appreciate your feedback on our technical content and encourage you to tell us what you think. If you’d like to add comments, provide insights, correct a typo, or even ask a question, you can do so directly in the documentation.

NOTE

You must have a Red Hat account and be logged in to the customer portal.

To submit documentation feedback from the customer portal, do the following:

1. Select the Multi-page HTML format.
2. Click the Feedback button at the top-right of the document.
3. Highlight the section of text where you want to provide feedback.
4. Click the Add Feedback dialog next to your highlighted text.
5. Enter your feedback in the text box on the right of the page and then click Submit.

We automatically create a tracking issue each time you submit feedback. Open the link that is displayed after you click Submit and start watching the issue or add more comments.
CHAPTER 1. MANAGING YOUR PRIVATE AUTOMATION HUB CONTAINER REGISTRY

Manage container image repositories in your Ansible Automation Platform infrastructure using the automation hub container registry. Automation hub provides features to govern who can access individual container repositories, change tags on images, view activity and image layers, and provide additional information related to each container repository.

1.1. CONTAINER REGISTRIES

The automation hub container registry is used for storing and managing container images. Once you have built or sourced a container image, you can push that container image to the registry portion of private automation hub to create a container repository.

Next steps

- Push a container image to the automation hub container registry.
- Create a group with access to the container repository in the registry.
- Add the new group to the container repository.
- Add a README to the container repository to provide users with information and relevant links.
CHAPTER 2. CONFIGURING USER ACCESS FOR CONTAINER REPOSITORIES IN PRIVATE AUTOMATION HUB

Configure user access for container repositories in your private automation hub to provide permissions that determine who can access and manage images in your Ansible Automation Platform.

2.1. PREREQUISITES

- You can create groups and assign permissions in private automation hub.

2.2. CONTAINER REGISTRY GROUP PERMISSIONS

User access provides granular controls to how users can interact with containers managed in private automation hub. Use the following list of permissions to create groups with the right privileges for your container registries.

<table>
<thead>
<tr>
<th>Permission name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new containers</td>
<td>Users can create new containers</td>
</tr>
<tr>
<td>Change container namespace permissions</td>
<td>Users can change permissions on the container repository</td>
</tr>
<tr>
<td>Change container</td>
<td>Users can change information on a container</td>
</tr>
<tr>
<td>Change image tags</td>
<td>Users can modify image tags</td>
</tr>
<tr>
<td>Pull private containers</td>
<td>Users can pull images from a private container</td>
</tr>
<tr>
<td>Push to existing container</td>
<td>Users can push an image to an existing container</td>
</tr>
<tr>
<td>View private containers</td>
<td>Users can view containers marked as private</td>
</tr>
</tbody>
</table>

2.3. CREATING A NEW GROUP

You can create and assign permissions to a group in automation hub that enables users to access specified features in the system. By default, there is an **admins** group in automation hub that has all permissions assigned and is available on initial login with credentials created when installing automation hub.

Prerequisites

- You have **groups** permissions and can create and manage group configuration and access in automation hub.

Procedure

1. Log in to your local automation hub.

3. Click Create.

4. Provide a Name and click Create.

You can now assign permissions and add users on the group edit page.

2.4. ASSIGNING PERMISSIONS TO GROUPS

You can assign permissions to groups in automation hub that enable users to access specific features in the system. By default, new groups do not have any assigned permissions. You can add permissions upon initial group creation or edit an existing group to add or remove permissions.

Prerequisites

- You have Change group permissions and can edit group permissions in automation hub.

Procedure

1. Log in to your local automation hub.


3. Click Add roles.

4. Click in the name field and fill in the role name.

5. Click in the description field and fill in the description.

6. Complete the Permissions section.

7. Click in the field for each permission type and select permissions that appear in the list.

8. Click Save when finished assigning permissions.


10. Click on a group name.

11. Click on the Access tab.

12. Click Add roles.

13. Select the role created in step 8.

14. Click Next to confirm the selected role.

15. Click Add to complete adding the role.

The group can now access features in automation hub associated with their assigned permissions.

Additional resources

- See Container registry group permissions to learn more about specific permissions.
2.5. ADDING USERS TO GROUPS

You can add users to groups when creating a group or manually add users to existing groups. This section describes how to add users to an existing group.

Prerequisites

- You have groups permissions and can create and manage group configuration and access in automation hub.

Procedure

1. Log in to automation hub.
3. Click on a Group name.
4. Navigate to the Users tab, then click Add.
5. Select users to add from the list and click Add.

You have added the users you selected to the group. These users now have permissions to use automation hub assigned to the group.
CHAPTER 3. POPULATING YOUR PRIVATE AUTOMATION HUB CONTAINER REGISTRY

By default, private automation hub does not include container images. To populate your container registry, you need to push a container image to it. The procedures in this section describe how to pull images from the Red Hat Ecosystem Catalog (registry.redhat.io), tag them, and push them to your private automation hub container registry.

IMPORTANT

Image manifests and filesystem blobs are served directly from registry.redhat.io and registry.access.redhat.com. However, from 1st of May 2023, filesystem blobs are served from quay.io instead. To avoid problems pulling container images, you must enable outbound connections to the following hostnames:

- cdn.quay.io
- cdn01.quay.io
- cdn02.quay.io
- cdn03.quay.io

This change should be made to any firewall configuration that specifically enables outbound connections to registry.redhat.io or registry.access.redhat.com.

Use the hostnames instead of IP addresses when configuring firewall rules.

After making this change you can continue to pull images from registry.redhat.io and registry.access.redhat.com. You do not require a quay.io login, or need to interact with the quay.io registry directly in any way to continue pulling Red Hat container images.

3.1. PREREQUISITES

- You have permissions to create new containers and push containers to private automation hub.

3.2. OBTAINING IMAGES FOR USE IN AUTOMATION HUB

Before you can push container images to your private automation hub, you must first pull them from an existing registry and tag them for use. This example details how to pull an image from the Red Hat Ecosystem Catalog (registry.redhat.io).

Prerequisites

You have permissions to pull images from registry.redhat.io.

Procedure

1. Log in to Podman using your registry.redhat.io credentials:

   $ podman login registry.redhat.io

2. Enter your username and password at the prompts.
3. Pull a container image:

   $ podman pull registry.redhat.io/<container_image_name>:<tag>

Verification

1. List the images in local storage:

   $ podman images

2. Verify that the image you recently pulled is contained in the list.

3. Verify that the tag is correct.

Additional resources

- See Red Hat Ecosystem Catalog Help for information on registering and getting images.

3.3. TAGGING IMAGES FOR USE IN AUTOMATION HUB

After you pull images from a registry, tag them for use in your private automation hub container registry.

Prerequisites

- You have pulled a container image from an external registry.
- You have the FQDN or IP address of the automation hub instance.

Procedure

- Tag a local image with the automation hub container repository

   $ podman tag registry.redhat.io/<container_image_name>:<tag> <automation_hub_URL>/<container_image_name>

Verification

1. List the images in local storage:

   $ podman images

2. Verify that the image you recently tagged with your automation hub information is contained in the list.

3.4. PUSHING A CONTAINER IMAGE TO PRIVATE AUTOMATION HUB

You can push tagged container images to private automation hub to create new containers and populate the container registry.

Prerequisites

- You have permissions to create new containers.
You have the FQDN or IP address of the automation hub instance.

Procedure

1. Log in to Podman using your automation hub location and credentials:

   ```bash
   $ podman login -u=<username> -p=<password> <automation_hub_url>
   ```

2. Push your container image to your automation hub container registry:

   ```bash
   $ podman push <automation_hub_url>/<container_image_name>
   ```

   **NOTE**
   
   The `--remove-signatures` flag is required when signed images from registry.redhat.io are pushed to the automation hub container registry. The `push` operation re-compresses image layers during the upload, which is not guaranteed to be reproducible and is client implementation dependent. This may lead to image-layer digest changes and a failed push operation, resulting in
   
   **Error: Copying this image requires changing layer representation, which is not possible (image is signed or the destination specifies a digest).**

Verification

1. Log in to your automation hub.

2. Navigate to **Container Registry**.

3. Locate the container in the container repository list.
CHAPTER 4. SETTING UP YOUR CONTAINER REPOSITORY

You can setup your container repository to add a description, include a README, add groups who can access the repository, and tag images.

4.1. PREREQUISITES

- You are logged in to a private automation hub with permissions to change the repository.

4.2. ADDING A README TO YOUR CONTAINER REPOSITORY

Add a README to your container repository to provide instructions to your users for how to work with the container. Automation hub container repositories support Markdown for creating a README. By default, the README will be empty.

Prerequisites

- You have permissions to change containers.

Procedure

1. Navigate to Execution Environments.
2. Select your container repository.
3. On the Detail tab, click Add.
4. In the Raw Markdown text field, enter your README text in Markdown.
5. Click Save when you are finished.

Once you add a README, you can edit it at any time by clicking Edit and repeating steps 4 and 5.

4.3. PROVIDING ACCESS TO YOUR CONTAINER REPOSITORY

Provide access to your container repository for users who need to work the images. Adding a group allows you to modify the permissions the group can have to the container repository. You can use this option to extend or restrict permissions based on what the group is assigned.

Prerequisites

- You have change container namespace permissions.

Procedure

1. Navigate to Execution Environments.
2. Select your container repository.
3. Click Edit at the top right of your window.
4. Under Groups with access, select a group or groups to grant access to.
4.4. TAGGING CONTAINER IMAGES

Tag images to add an additional name to images stored in your automation hub container repository. If no tag is added to an image, automation hub defaults to latest for the name.

Prerequisites

- You have change image tags permissions.

Procedure

1. Navigate to Execution Environments.
2. Select your container repository.
3. Click the Images tab.
4. Click the More Actions icon \( \cdots \), then click Manage tags.
5. Add a new tag in the text field and click Add.
   - Optional: Remove current tags by clicking x on any of the tags for that image.
6. Click Save.

Verification

- Click the Activity tab and review the latest changes.

4.5. CREATING A CREDENTIAL IN AUTOMATION CONTROLLER

Previously, you were required to deploy a registry to store execution environment images. On Ansible Automation Platform 2.0 and later, it is assumed that you already have a container registry up and running. Therefore, you are only required to add the credentials of a container registry of your choice to store execution environment images.

To pull container images from a password or token-protected registry, create a credential in automation controller:

Procedure

1. Navigate to automation controller.
2. In the side-menu bar, click Resources → Credentials.
3. Click Add to create a new credential.
4. Enter an authorization Name, Description, and Organization.
5. Select the Credential Type.
6. Enter the **Authentication URL**. This is the container registry address.

7. Enter the **Username** and **Password or Token** required to log in to the container registry.

8. Optionally, select **Verify SSL** to enable SSL verification.

9. Click **Save**.
CHAPTER 5. PULLING IMAGES FROM A CONTAINER REPOSITORY

Pull images from the automation hub container registry to make a copy to your local machine. Automation hub provides the `podman pull` command for each `latest` image in the container repository. You can copy and paste this command into your terminal, or use `podman pull` to copy an image based on an image tag.

5.1. PREREQUISITES

You must have permission to view and pull from a private container repository.

5.2. PULLING AN IMAGE

You can pull images from the automation hub container registry to make a copy to your local machine. Automation hub provides the `podman pull` command for each `latest` image in the container repository.

NOTE

If you need to pull container images from a password or token-protected registry, you must create a credential in automation controller before pulling the image.

Procedure

1. Navigate to Execution Environments.
2. Select your container repository.
3. In the Pull this image entry, click Copy to clipboard.
4. Paste and run the command in your terminal.

Verification

- Run `podman images` to view images on your local machine.

5.3. SYNCING IMAGES FROM A CONTAINER REPOSITORY

You can pull images from the automation hub container registry to sync an image to your local machine.

Prerequisites

You must have permission to view and pull from a private container repository.

Procedure

To sync an image from a remote container registry, you need to configure a remote registry.

1. Navigate to Execution Environments → Remote Registries.
2. Add `https://registry.redhat.io` to the registry.
3. Add any required credentials to authenticate.
Some container registries are aggressive with rate limiting. It is advisable to set a rate limit under Advanced Options.

1. Navigate to Execution Environments → Execution Environments.
2. Click Add execution environment in the page header.
3. Select the registry you wish to pull from. The "name" field displays the name of the image that will show up as on your local registry.

The "Upstream name" field is the name of the image on the remote server. For example if the upstream name is set to "alpine" and the "name" field to "local/alpine", the alpine image will be downloaded from the remote and renamed to local/alpine.

It is advisable to set a list of tags to include or exclude. Syncing images with a large number of tags is time consuming and will use a lot of disk space.

Additional resources

- See Red Hat Container Registry Authentication for a list of registries.

5.4. ADDITIONAL RESOURCES

- See the What is Podman? documentation for options to use when pulling images.
CHAPTER 6. WORKING WITH SIGNED CONTAINERS

6.1. DEPLOYING YOUR SYSTEM FOR CONTAINER SIGNING

Execution Environments are container images used by Ansible automation controller to run jobs. This content can be downloaded to the private automation hub, and published within your organization.

Automation hub is now implementing image signing so the user can rely on better security for the EE container images.

As an Ansible Automation Platform user, you can now confirm if an EE/Container is already signed, or how to use the proper tools to sign and verify the signature yourself. This section details how to deploy your system so it is ready for container signing.

Procedure

1. Deploy your system with support for container signing activated.

   
   _automation_hub:
   
   _automationhub_create_default_container_signing_service: true
   
   _automationhub_container_signing_service_key: _path/to/gpg.key_
   
   _automationhub_container_signing_service_script: _path/to/executable_

2. Navigate to automation hub.

3. In the navigation pane, select Signature Keys.

4. Ensure you have a key titled container-default, or container-anyname.

   NOTE

   The 'container-default' service is created by the Ansible Automation Platform installer.

6.2. ADDING CONTAINERS REMOTELY TO AUTOMATION HUB

There are two ways to add containers to automation hub:

- Create Remotes
- Execution Environment

Procedure

To add a remote registry:

1. In automation hub, click Execution Environments in the main menu pane. Two choices become available, Execution Environments, and Remote Registries.

2. Click Remote Registries.

3. Click Add Remote Registry in the main window.

   - In the Name field, enter the name of the registry where the container resides.
6.3. ADDING AN EXECUTION ENVIRONMENT

**Procedure**

1. Navigate to *Execution Environments*.

2. Enter the name of the execution environment.

3. Optional: Enter the upstream name.

4. Under the *Registry*, select the name of the registry from the drop-down menu.

5. Enter tags in the *Add tags to include* field. If the field is left blank, all of the tags will be passed. So it is important to pass repository specific tags.

6. The remaining fields are optional:
   - Currently included tags
   - Add tag(s) to exclude
   - Currently excluded tag(s)
   - Description
   - Groups with access

7. Click **Save**.

8. Synchronize the image.

6.4. PUSHING CONTAINER IMAGES FROM YOUR LOCAL

**Procedure**

1. From a terminal, log into podman, or any container client currently in use.

   ```
   > podman pull <__container-name__>
   ```

2. After the image is pulled, add tags:

   ```
   > podman tag <container-name> _<server-address>_/<container-name>:<tag name>
   ```

3. Sign the image after changes have been made, and push it back up:
4. Push the image without signing it:

```
> podman push _<server-address>_/<container-name>:<tag name>
--tls-verify=false
```

5. Navigate to the automation hub and click on **Execution Environments** if that window is not open.

6. Click the **Refresh** icon to refresh the page to show the new execution environment.

7. Click the name of the image.

In the details page, below the image name, will be displayed whether or not the image has been signed. In this case, it displays "Unsigned."

To sign the image from automation hub:

1. Click the image name to open the details page.

2. Click the three dots in the upper right hand corner of the details page. Three options are available:
   - Use in Controller
   - Delete
   - Sign

3. Click **sign** from the drop-down menu.

The signing service signs the image. Once the image is signed, the status changes to "signed."

### 6.5. Using Policies with Signed Images

Policies can be used by podman or other image clients to ensure the validity of the image by assigning specific policies to that signature.

#### 6.5.1. Using podman to ensure an image is signed by a specific signature

When ensuring a signature is signed by a specific signature(s), the signature must be on your local.

**Procedure**

1. In the navigation pane, select **Signature Keys**.

2. Click the three dots on the right hand side of the signature that you are using.

3. Select **Download key** from the drop-down menu. A new window opens.

4. In the **Name** field, enter the name of the key.
5. Click **Save**.

**6.6. CONFIGURING THE CLIENT TO VERIFY SIGNATURES**

**Prerequisites**

- The client must have sudo privileges configured to verify signatures.

**Procedure**

1. In a terminal type:

   ```bash
   > sudo <name of editor> __/etc/containers/policy.json__
   ```

   The file may look similar to this:

   ```json
   {
   "default": ["type": "reject"],
   "transports": {
   "docker": {
   "quay.io": ["type": "insecureAcceptAnything"],
   "docker.io": ["type": "insecureAcceptAnything"],
   "<server-address>": [ {
   "type": "signedBy",
   "keyType": "GPGKeys",
   "keyPath": "/tmp/containersig.txt"
   }]
   }
   }
   ```

   This shows there will be no verification from either **quay.io**, or **docker.io** since the type is **insecureAcceptAnything** which overrides the default type of **reject**. However, there will be verification from **<server-address>** as the parameter **type** has been set to "**signedBy**".

   **NOTE**

   The only **keyType** currently supported is GPG keys.

   1. Under the **<server-address>** entry, modify the **keyPath** to reflect the name of your key file.

   ```json
   {
   "default": ["type": "reject"],
   "transports": {
   "docker": {
   "quay.io": ["type": "insecureAcceptAnything"],
   "docker.io": ["type": "insecureAcceptAnything"],
   "<server-address>_1": [ {
   "type": "signedBy",
   "keyType": "GPGKeys",
   "keyPath": "/tmp/<key file name", 1
   "signedIdentity": {
   "type": "remapIdentity",
   ```
2. Save and close the file.

3. Pull the file using podman, or your client of choice:

```bash
> podman pull _<server-address>_/<container-name>:<tag name>
--tls-verify=false
```

This verifies the signature with no errors.
CHAPTER 7. DELETING A CONTAINER REPOSITORY

Delete a container repository from your local automation hub to manage your disk space. You can delete repositories from the Red Hat Ansible Automation Platform interface in the Container Repository list view.

7.1. DELETING A CONTAINER REPOSITORY

Prerequisites

- You have permissions to manage repositories.

Procedure

1. Navigate to Execution Environments.

2. On the container repository you would like to delete, click the More Actions icon ⋮, then click Delete.

3. When presented with the confirmation message, click the checkbox, then click Delete.

Verification

- Return to the Execution Environments list view. The container repository should be removed from the list.