



Red Hat Ansible Automation Platform 2.2

Managing Red Hat Certified and Ansible Galaxy collections in automation hub

Configure Automation Hub to deliver curated Red Hat Certified and Ansible Galaxy collections content to your users.

Red Hat Ansible Automation Platform 2.2 Managing Red Hat Certified and Ansible Galaxy collections in automation hub

Configure Automation Hub to deliver curated Red Hat Certified and Ansible Galaxy collections content to your users.

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

Providing Feedback: If you have a suggestion to improve this documentation, or find an error, please contact technical support at to create an issue on the Ansible Automation Platform Jira project using the Docs component.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. MANAGING RED HAT CERTIFIED COLLECTIONS SYNCLISTS IN AUTOMATION HUB	5
1.1. ABOUT RED HAT CERTIFIED COLLECTIONS SYNCLISTS	5
1.2. CREATING A SYNCLIST OF RED HAT CERTIFIED COLLECTIONS	5
CHAPTER 2. CONFIGURING AUTOMATION HUB REMOTE REPOSITORIES TO SYNC CONTENT FROM RED HAT CERTIFIED AND ANSIBLE GALAXY COLLECTIONS	7
2.1. ABOUT REMOTE REPOSITORIES	7
2.2. RETRIEVING YOUR RED HAT CERTIFIED COLLECTIONS SYNC URL AND API TOKEN.	7
2.3. CONFIGURING THE RH-CERTIFIED REMOTE REPOSITORY AND SYNCHRONIZING RED HAT ANSIBLE CERTIFIED CONTENT COLLECTION.	8
2.4. CONFIGURING THE COMMUNITY REMOTE REPOSITORY AND SYNCING ANSIBLE GALAXY COLLECTIONS	8
CHAPTER 3. COLLECTIONS AND CONTENT SIGNING IN PRIVATE AUTOMATION HUB	10
3.1. CONFIGURING CONTENT SIGNING ON PRIVATE AUTOMATION HUB	10
3.2. USING CONTENT SIGNING SERVICES IN PRIVATE AUTOMATION HUB	11
3.3. CONFIGURING ANSIBLE-GALAXY CLI TO VERIFY COLLECTIONS	12
CHAPTER 4. CONCLUSION	14

PREFACE

You can sync automation hub to use the Red Hat Certified Collections available through your Ansible Automation Platform subscription or community collections available through Ansible Galaxy.

Your organization can access and curate into a unique set of collections from all Red Hat Certified content in the automation hub service hosted on console.redhat.com.

You can also configure private automation hub for signing and publishing Ansible content collections tailored to meet your organization's unique needs.

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](#).

CHAPTER 1. MANAGING RED HAT CERTIFIED COLLECTIONS SYNCLISTS IN AUTOMATION HUB

You can use automation hub to distribute the relevant Red Hat Certified collections content to your users by creating synclists.

1.1. ABOUT RED HAT CERTIFIED COLLECTIONS SYNCLISTS

A synclist is a curated group of Red Hat Certified collections that is assembled by your organization administrator that syncs to your local automation hub. Use synclists to manage only the content that you want and exclude unnecessary collections. You can design and manage your synclist from the content available as part of Red Hat Certified collections on console.redhat.com

Each synclist has its own unique repository URL you can use to designate as a remote source for content in automation hub and is securely accessed using an API token.

1.2. CREATING A SYNCLIST OF RED HAT CERTIFIED COLLECTIONS

You can create a synclist of curated Red Hat Certified collections in automation hub on console.redhat.com. Your synclist repository is located under **Automation Hub** → **Repo Management**, which is updated whenever you choose to manage content within Red Hat Certified collections.

By default, all Red Hat Certified collections are included in your initial organization synclist.

Prerequisites

- You have a valid Ansible Automation Platform subscription.
- You have Organization Administrator permissions for console.redhat.com.
- Ensure that the following domain names are part of either the firewall or the proxy's allowlist for successful connection and download of collections from automation hub or Galaxy server:
 - **galaxy.ansible.com**
 - **cloud.redhat.com**
 - **console.redhat.com**
 - **sso.redhat.com**
- Automation hub resources are stored in Amazon Simple Storage. Add the following domain name to the allow list:
 - **automation-hub-prd.s3.us-east-2.amazonaws.com**
 - **ansible-galaxy.s3.amazonaws.com**
- SSL inspection is disabled either when using self signed certificates or for the Red Hat domains.

Procedure

1. Log in to console.redhat.com.
2. Navigate to **Automation Hub** → **Collections**.

3. Use the toggle switch on each collection to determine whether to exclude it from your synclist.

When you finish managing collections for your synclist, you can navigate to **Automation Hub** → **Repo Management** to initiate the remote repository sync to your local automation hub. If your remote repository is already configured, you can manually sync Red Hat Certified collections to your local automation hub to update the collections content that you made available to local users.

CHAPTER 2. CONFIGURING AUTOMATION HUB REMOTE REPOSITORIES TO SYNC CONTENT FROM RED HAT CERTIFIED AND ANSIBLE GALAXY COLLECTIONS

You can configure your local automation hub to sync with Red Hat Certified Collections hosted in your organization repository on console.redhat.com or to your choice of collections in Ansible Galaxy.

2.1. ABOUT REMOTE REPOSITORIES

You can configure your local automation hub to sync with Red Hat Certified Collections hosted in your organization repository on console.redhat.com and to your choice of collections in Ansible Galaxy by configuring remote repositories.

Each remote repository located in **Repo Management** → **Remote** provides information for both **community** and **rh-certified** about when the repo was **last updated** and when content was **last synced**. You can add new content to automation hub at any time using the **Edit** and **Sync** features included on the **Repo Management** → **Remote** page.



2.2. RETRIEVING YOUR RED HAT CERTIFIED COLLECTIONS SYNC URL AND API TOKEN.

You can sync Red Hat certified collections curated by your organization from console.redhat.com to your local automation hub.

Prerequisites

- You have Organization Administrator permissions for console.redhat.com.

Procedure

1. Log in to console.redhat.com as an Organization Administrator.
2. Navigate to **Automation Hub** → **Repo Management**.
3. Locate the **Sync URL** and click the **Copy to clipboard** icon (). Paste the **Sync URL** in a file to use when configuring the rh-certified remote.
4. Click the **More actions** icon () and click **Get token**.
5. On the **Token management** page, click **Load token**.
6. Click **Copy to clipboard** to copy the API token.
7. Paste the API token into a file and store in a secure location.



IMPORTANT

The API token is a secret token used to protect your content. Store your API token in a secure location.

2.3. CONFIGURING THE RH-CERTIFIED REMOTE REPOSITORY AND SYNCHRONIZING RED HAT ANSIBLE CERTIFIED CONTENT COLLECTION.

You can edit the **rh-certified** remote repository to sync collections from automation hub hosted on cloud.redhat.com to your local automation hub. By default, your local automation hub **rh-certified** repository includes the URL for the entire group of Red Hat Certified Collections available on cloud.redhat.com. To use only those collections specified by your organization, you must include a unique URL.


Prerequisites

- You have **Modify Ansible repo content** permissions. See [Managing user access in Automation Hub](#) for more information on permissions.
- You have retrieved the Sync URL and API Token from the automation hub hosted service on console.redhat.com.
- You have configured access to port 443. This is required for synchronizing certified collections. For more information, see the automation hub table in the [Network ports and protocols](#) chapter of the Red Hat Ansible Automation Platform Planning Guide.

Procedure

1. Log in to your local automation hub.
2. Navigate to **Repo Management**.
3. Click the **Remotes** tab.



4. In the **rh-certified** remote, click  and click **Edit**.
5. In the modal, paste the Sync URL and Token you acquired from cloud.redhat.com.
6. Click **Save**.

The modal closes and returns you to the **Repo Management** page. You can now synchronize collections between your organization synclist on console.redhat.com and your private automation hub.

+ . Click **Sync** to synchronize collections.

The **Sync status** notification updates to notify you of completion of Red Hat Certified collections sync.

Verification

You can confirm that your collections content has synced successfully by selecting **Red Hat Certified** from the collections content drop-down list.

2.4. CONFIGURING THE COMMUNITY REMOTE REPOSITORY AND SYNCING ANSIBLE GALAXY COLLECTIONS

You can edit the **community** remote repository to sync chosen collections from Ansible Galaxy to your local automation hub. By default, your local automation hub **community** repository directs to <https://galaxy.ansible.com/api/>.


Prerequisites

- You have **Modify Ansible repo content** permissions. See [Managing user access in Automation Hub](#) for more information on permissions.
- You have a **requirements.yml** file that identifies those collections to sync from Ansible Galaxy. See example below.

Requirements.yml example

```
collections:  
  # Install a collection from {Galaxy}.  
  - name: community.aws  
    version: 5.2.0  
    source: https://galaxy.ansible.com
```

Procedure

1. Log in to your local automation hub.
2. Navigate to **Repo Management**.
3. Click the **Remotes** tab.
4. In the **community** remote, click the **More Actions** icon  and click **Edit**.
5. In the modal, click **Browse** and locate the **requirements.yml** file on your local machine.
6. Click **Save**.

The modal closes and returns you to the **Repo Management** page. You can now sync collections identified in your **requirements.yml** file from Ansible Galaxy to your local automation hub.

1. Click **Sync** to sync collections from Ansible Galaxy and automation hub.

The **Sync status** notification updates to notify you of completion or failure of Ansible Galaxy collections sync to your automation hub.

Verification

You can confirm successful sync by selecting **Community** from the collections content drop-down list.

CHAPTER 3. COLLECTIONS AND CONTENT SIGNING IN PRIVATE AUTOMATION HUB

As an automation administrator for your organization, you can configure private automation hub for signing and publishing Ansible content collections from different groups within your organization.

For additional security, automation creators can configure Ansible-Galaxy CLI to verify these collections to ensure they have not been changed after they were uploaded to automation hub.

3.1. CONFIGURING CONTENT SIGNING ON PRIVATE AUTOMATION HUB

To successfully sign and publish Ansible Certified Content Collections, you must configure private automation hub for signing.

Prerequisites

- Your GnuPG key pairs have been securely set up and managed by your organization.
- Your public/private key pair has proper access for configuring content signing on private automation hub.

Procedure

1. Create a signing script that only accepts a filename.



NOTE

This script will act as the signing service and must generate an ascii-armored detached **gpg** signature for that file using the key specified through the **PULP_SIGNING_KEY_FINGERPRINT** environment variable.

The script then prints out a JSON structure with the following format.

```
{"file": "filename", "signature": "filename.asc"}
```

All the file names are relative paths inside the current working directory. The file name must remain the same for the detached signature, as shown.

The following example shows a script that produces signatures for content:

```
#!/usr/bin/env bash

FILE_PATH=$1
SIGNATURE_PATH="$1.asc"

ADMIN_ID="$PULP_SIGNING_KEY_FINGERPRINT"
PASSWORD="password"

# Create a detached signature
gpg --quiet --batch --pinentry-mode loopback --yes --passphrase \
  $PASSWORD --homedir ~/.gnupg/ --detach-sign --default-key $ADMIN_ID \
  --armor --output $SIGNATURE_PATH $FILE_PATH
```

```
# Check the exit status
STATUS=$?
if [ $STATUS -eq 0 ]; then
    echo {"file": "$FILE_PATH", "signature": "$SIGNATURE_PATH"}
else
    exit $STATUS
fi
```

After you deploy a private automation hub with signing enabled to your Ansible Automation Platform cluster, new UI additions display when you interact with collections.

2. Review the Ansible Automation Platform installer inventory file for options that begins with **automationhub_***.

```
[all:vars]
.
.
.
automationhub_create_default_collection_signing_service = True
automationhub_auto_sign_collections = True
automationhub_require_content_approval = True
automationhub_collection_signing_service_key = /abs/path/to/galaxy_signing_service.gpg
automationhub_collection_signing_service_script = /abs/path/to/collection_signing.sh
```

The two new keys (**automationhub_auto_sign_collections** and **automationhub_require_content_approval**) indicate that the collections must be signed and require approval after they are uploaded to private automation hub.

3.2. USING CONTENT SIGNING SERVICES IN PRIVATE AUTOMATION HUB

After you have configured content signing on your private automation hub, you can manually sign a new collection or replace an existing signature with a new one so that users who want to download a specific collection have the assurance that the collection is intended for them and has not been modified after certification.

Content signing on private automation hub provides solutions for the following scenarios:

- Your system does not have automatic signing configured and you must use a manual signing process to sign collections.
- The current signatures on the automatically configured collections are corrupted and must be replaced with new signatures.
- Additional signatures are required for previously signed content.
- You want to rotate signatures on your collections.

Procedure

1. Log in to your private automation hub instance in the automation hub UI.
2. In the left navigation, click **Collections** → **Approval**. The Approval dashboard is displayed with a list of collections.

3. Click **Sign and approve** for each collection you want to sign.
4. Verify that the collections you signed and approved manually are displayed in the Collections tab.

3.3. CONFIGURING ANSIBLE-GALAXY CLI TO VERIFY COLLECTIONS

You can configure Ansible-Galaxy CLI to verify collections. This ensures that collections you download are approved by your organization and have not been changed after they were uploaded to automation hub.

If a collection has been signed by automation hub, the server provides ASCII armored, GPG-detached signatures to verify the authenticity of **MANIFEST.json** before using it to verify the collection's contents. You must opt into signature verification by [configuring a keyring](#) for **ansible-galaxy** or providing the path with the **--keyring** option.

Prerequisites

- Signed collections are available in automation hub to verify signature.
- Certified collections can be signed by approved roles within your organization.
- Public key for verification has been added to the local system keyring.

Procedure

1. To import a public key into a non-default keyring for use with **ansible-galaxy**, run the following command.

```
gpg --import --no-default-keyring --keyring ~/.ansible/pubring.kbx my-public-key.asc
```



NOTE

In addition to any signatures provided by the automation hub, signature sources can also be provided in the requirements file and on the command line. Signature sources should be URIs.

2. Use the **--signature** option to verify the collection name provided on the CLI with an additional signature.

```
ansible-galaxy collection install namespace.collection
--signature https://examplehost.com/detached_signature.asc
--signature file:///path/to/local/detached_signature.asc --keyring ~/.ansible/pubring.kbx
```

You can use this option multiple times to provide multiple signatures.

3. Confirm that the collections in a requirements file list any additional signature sources following the collection's signatures key, as in the following example.

```
# requirements.yml
collections:
  - name: ns.coll
    version: 1.0.0
  signatures:
```


- `https://examplehost.com/detached_signature.asc`
- `file:///path/to/local/detached_signature.asc`

```
ansible-galaxy collection verify -r requirements.yml --keyring ~/.ansible/pubring.kbx
```

When you install a collection from automation hub, the signatures provided by the server are saved along with the installed collections to verify the collection's authenticity.

4. (Optional) If you need to verify the internal consistency of your collection again without querying the Ansible Galaxy server, run the same command you used previously using the **--offline** option.

CHAPTER 4. CONCLUSION

When you complete all of the previous procedures, you will have:

- created a synclist for Red Hat Certified collections content.
- synced that content to your local automation hub.
- designated community collections from Ansible Galaxy to distribute to your users.
- configured content signing on private automation hub.
- signed and approved collections for your organization's specific needs.
- configured Ansible-Galaxy CLI to verify collections before signing them.

Users can now view and download collections content from your local automation hub.