Red Hat Ansible Automation Platform 2.4

Managing Red Hat Certified, validated, and Ansible Galaxy content in automation hub

Configure and distribute curated collections with automation hub
Configure and distribute curated collections with automation hub
Abstract

This guide shows how to configure automation hub to deliver curated Red Hat Certified and Ansible Galaxy collections to your users.
# Table of Contents

**PREFACE** .................................................................................................................. 3

**MAKING OPEN SOURCE MORE INCLUSIVE** ................................................................. 4

**PROVIDING FEEDBACK ON RED HAT DOCUMENTATION** .............................................. 5

**CHAPTER 1. MANAGING ANSIBLE CONTENT COLLECTION SYNCLISTS IN AUTOMATION HUB** ................................. 6
   1.1. ABOUT RED HAT ANSIBLE CERTIFIED CONTENT COLLECTIONS SYNCLISTS .......................... 6
   1.2. CREATING A SYNCLIST OF RED HAT ANSIBLE CERTIFIED CONTENT COLLECTIONS .................. 6

**CHAPTER 2. CONFIGURING ANSIBLE AUTOMATION HUB REMOTE REPOSITORIES TO SYNCHRONIZE CONTENT** .......... 8
   2.1. REASONS TO CREATE REMOTE CONFIGURATIONS .................................................. 8
   2.2. RETRIEVING THE SYNC URL AND API TOKEN FOR YOUR RED HAT CERTIFIED COLLECTION ........... 8
   2.3. CONFIGURING THE RH-CERTIFIED REMOTE REPOSITORY AND SYNCHRONIZING RED HAT ANSIBLE CERTIFIED CONTENT COLLECTION. ................................................. 9
   2.4. CONFIGURING THE COMMUNITY REMOTE REPOSITORY AND SYNCING ANSIBLE GALAXY COLLECTIONS .......... 10

**CHAPTER 3. PRIVATE AUTOMATION HUB** .................................................................... 11
   3.1. REQUIRED SHARED FILESYSTEM .................................................................. 11
   3.2. SETTING UP THE SHARED FILESYSTEM ..................................................... 11
   3.3. ENABLING FIREWALL SERVICES .................................................................. 11

**CHAPTER 4. COLLECTIONS AND CONTENT SIGNING IN PRIVATE AUTOMATION HUB** ............... 13
   4.1. CONFIGURING CONTENT SIGNING ON PRIVATE AUTOMATION HUB ......................... 13
   4.2. USING CONTENT SIGNING SERVICES IN PRIVATE AUTOMATION HUB ....................... 14
   4.3. DOWNLOADING SIGNATURE PUBLIC KEYS ................................................... 15
   4.4. CONFIGURING ANSIBLE-GALAXY CLI TO VERIFY COLLECTIONS ....................... 15

**CHAPTER 5. FREQUENTLY ASKED QUESTIONS ABOUT RED HAT ANSIBLE CERTIFIED CONTENT** ...................... 17
   5.1. WHY CERTIFY ANSIBLE COLLECTIONS? ..................................................... 17
   5.2. HOW DO I GET A COLLECTION CERTIFIED? ............................................... 17
   5.3. WHAT’S THE DIFFERENCE BETWEEN ANSIBLE GALAXY AND ANSIBLE AUTOMATION HUB? ............... 17
   5.4. HOW DO I REQUEST A NAMESPACE ON ANSIBLE GALAXY? ................................. 17
   5.5. ARE THERE ANY RESTRICTIONS FOR ANSIBLE GALAXY NAMESPACE NAMING? .................. 18
   5.6. ARE THERE ANY RECOMMENDATIONS FOR COLLECTION NAMING? .......................... 18
   5.7. HOW DO I GET A NAMESPACE ON ANSIBLE AUTOMATION HUB? ......................... 18
   5.8. HOW DO I RUN SANITY TESTS ON MY COLLECTION? ...................................... 18
   5.9. DOES ANSIBLE GALAXY HOUSE THE SOURCE CODE FOR MY COLLECTION? .................... 18
   5.10. DOES RED HAT OFFICIALLY SUPPORT COLLECTIONS DOWNLOADED AND INSTALLED FROM ANSIBLE GALAXY ....................... 18
   5.11. HOW DOES THE JOINT SUPPORT AGREEMENT ON CERTIFIED COLLECTIONS WORK? ............ 19
   5.12. CAN I CREATE AND CERTIFY A COLLECTION CONTAINING ONLY ANSIBLE ROLES? ............ 19

**CHAPTER 6. ANSIBLE VALIDATED CONTENT** .................................................................. 20
   6.1. CONFIGURING VALIDATED COLLECTIONS WITH THE INSTALLER ......................... 20
   6.2. INSTALLING VALIDATED CONTENT USING THE TARBALL .................................. 20

**CHAPTER 7. CONCLUSION** ......................................................................................... 22
Red Hat Ansible Certified Content Collection is included in your subscription to Red Hat Ansible Automation Platform. Red Hat Ansible content includes two types of content: Ansible Certified Content Collections and Ansible validated content. Using Ansible automation hub, you can access and curate a unique set of collections from all forms of Ansible content.

Ansible validated collections are available in your private automation hub through the Platform Installer. When you download Red Hat Ansible Automation Platform with the bundled installer, validated content is pre-populated into the private automation hub by default, but only if you enable the private automation hub as part of the inventory.

If you are not using the bundle installer, a Red Hat supplied Ansible playbook can be used to install validated content. For further information, see Ansible validated content

You can update these collections manually by downloading their packages.
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 ANSIBLE CONTENT COLLECTION
SYNCLISTS IN AUTOMATION HUB

IMPORTANT

As of the 2.4 release you can still synchronize content, but synclists are deprecated, and will be removed in a future version.

From Ansible Automation Platform 2.4 a private automation hub administrator can go to the rh-certified remote and upload a manually-created requirements file.

Remotes are configurations that allow you to synchronize content to your custom repositories from an external collection source.

You can use Ansible automation hub to distribute the relevant Red Hat Certified collections content to your users by creating synclists or a requirements file. For more information about using requirements files, see Install multiple collections with a requirements file in the Ansible Galaxy User Guide.

1.1. ABOUT RED HAT ANSIBLE CERTIFIED CONTENT COLLECTIONS
SYNCLISTS

A synclist is a curated group of Red Hat Certified collections that is assembled by your organization administrator that synchronizes with your local Ansible automation hub. You can 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 content on console.redhat.com.

Each synclist has its own unique repository URL that 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 ANSIBLE CERTIFIED
CONTENT COLLECTIONS

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

All Ansible Certified Content Collections are included by default in your initial organization synclist.

Prerequisites

- You have a valid Ansible Automation Platform subscription.
- You have Organization Administrator permissions for console.redhat.com.
- 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
Ansible automation hub resources are stored in Amazon Simple Storage and the following domain name is in 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.
4. When you finish managing collections for your synclist, navigate to Automation Hub → Repositories to initiate the remote repository synchronization to your private automation hub.
5. Optional: If your remote repository is already configured, you can manually synchronize Red Hat Ansible Certified Content Collections to your private automation hub to update the collections content that you made available to local users.
CHAPTER 2. CONFIGURING ANSIBLE AUTOMATION HUB REMOTE REPOSITORIES TO SYNCHRONIZE CONTENT

You can configure your private automation hub to synchronize with Ansible Certified Content Collections hosted on console.redhat.com or to your choice of collections in Ansible Galaxy, using remote configurations.

IMPORTANT

As of the 2.4 release you can still synchronize content, but synclists are deprecated, and will be removed in a future version.

From Ansible Automation Platform 2.4 a private automation hub administrator can go to the rh-certified remote and upload a manually-created requirements file.

Remotes are configurations that allow you to synchronize content to your custom repositories from an external collection source.

2.1. REASONS TO CREATE REMOTE CONFIGURATIONS

Each remote configuration located in Automation Hub → Remotes provides information for both the community and rh-certified repository about when the repository was last updated. You can add new content to Ansible automation hub at any time using the Edit and Sync features included on the Automation Hub → Repositories page.

2.2. RETRIEVING THE SYNC URL AND API TOKEN FOR YOUR RED HAT CERTIFIED COLLECTION

You can synchronize Ansible Certified Content Collections curated by your organization from console.redhat.com to your private automation hub.

Prerequisites

- You have organization administrator permissions to create the synclist on console.redhat.com.

Procedure

1. Log in to console.redhat.com as an organization administrator.
2. Navigate to Automation Hub → Connect to Hub.
3. Under Offline token, click Load token.
4. Click Copy to clipboard to copy the API token.
5. 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.
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 synchronize collections from automation hub hosted on console.redhat.com to your private automation hub. By default, your private automation hub rh-certified repository includes the URL for the entire group of Ansible Certified Content Collections.

To use only those collections specified by your organization, a private automation hub administrator can upload manually-created requirements files from the rh-certified remote.

For more information about using requirements files, see Install multiple collections with a requirements file in the Ansible Galaxy User Guide.

If you have collections A, B, and C in your requirements file, and a new collection X is added to console.redhat.com that you want to use, you must add X to your requirements file for private automation hub to synchronize it.

Prerequisites

- You have valid 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 private automation hub.
2. Navigate to Automation Hub → Remotes.
3. In the rh-certified remote repository, click the More Actions icon ⋮ and click Edit.
4. In the modal, paste the Sync URL and Token you acquired from console.redhat.com.
5. Click Save.
   The modal closes and returns you to the Remotes page. You can now synchronize collections between your organization synclist on console.redhat.com and your private automation hub.
6. Click the More Actions icon ⋮ and select Sync.

The Sync status notification updates to notify you of completion of the Red Hat Certified Content Collections synchronization.

Verification

- Select Red Hat Certified from the collections content drop-down list to confirm that your collections content has synchronized successfully.
2.4. CONFIGURING THE COMMUNITY REMOTE REPOSITORY AND SYNCKING ANSIBLE GALAXY COLLECTIONS

You can edit the community remote repository to synchronize chosen collections from Ansible Galaxy to your private automation hub. By default, your private automation hub community repository directs to 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 synchronize from Ansible Galaxy as in the following example:

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

Requirements.yml example

Procedure

1. Log in to your Ansible automation hub.
2. Navigate to Automation Hub → Remotes.
3. In the Community remote, click the More Actions icon ⋮ and select Edit.
4. In the modal, click Browse and locate the requirements.yml file on your local machine.
5. Click Save.
   The modal closes and returns you to the Remotes page. You can now synchronize collections identified in your requirements.yml file from Ansible Galaxy to your private automation hub.
6. Click the More Actions icon ⋮ and select Sync to sync collections from Ansible Galaxy and Ansible automation hub.

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

Verification

- Select Community from the collections content drop-down list to confirm successful synchronization.
CHAPTER 3. PRIVATE AUTOMATION HUB

Ansible automation hub is the central repository place for the certified collections, and functions as the main source of trusted, tested and supported content.

With private automation hub, automation developers can collaborate and publish their own automation content and deliver Ansible code more easily within their organization. It is also the central repository for Ansible validated content, which is not supported, but is trusted and tested by Red Hat and our partners.

3.1. REQUIRED SHARED FILESYSTEM

A high availability automation hub requires you to have a shared file system, such as NFS, already installed in your environment. Before you run the Red Hat Ansible Automation Platform installer, verify that you installed the `/var/lib/pulp` directory across your cluster as part of the shared file system installation. The Red Hat Ansible Automation Platform installer returns an error if `/var/lib/pulp` is not detected in one of your nodes, causing your high availability automation hub setup to fail.

3.2. SETTING UP THE SHARED FILESYSTEM

You must mount the shared file system on each automation hub node:

Procedure

1. Create the `/var/lib/pulp` directory.
   ```
   # mkdir /var/lib/pulp
   ```

2. Mount the shared filesystem (this reference environment uses an NFS share).
   ```
   # mount -t nfs4 <nfs_share_ip_address>:/ /var/lib/pulp
   ```

3. Confirm that the shared filesystem is successfully mounted:
   ```
   $ df -h
   ```

3.3. ENABLING FIREWALL SERVICES

Because of the requirement of using a shared filesystem as part of a highly available Ansible automation hub environment, the following firewall services must be enabled to ensure that the filesystem is successfully mounted.

On each Ansible automation hub node, you must:

1. Ensure the following `firewalld` services (`nfs, mountd, rpc-bind`) are enabled.
   ```
   # firewall-cmd --zone=public --add-service=nfs
   # firewall-cmd --zone=public --add-service=mountd
   # firewall-cmd --zone=public --add-service=rpc-bind
   ```

2. Reload `firewalld` for changes to take effect.
   ```
   # firewall-cmd --reload
   ```
3. Verify the **firewalld** services are enabled.

    # firewall-cmd --get-services
CHAPTER 4. 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.

4.1. 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 accepts only a filename.

```
NOTE

This script acts 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
```
2. Review the Ansible Automation Platform installer inventory file for options that begin with `automationhub_*`.

```yaml
[all:vars]
  ...
  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.

**4.2. 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 manually approved are displayed in the **Collections** tab.

### 4.3. DOWNLOADING SIGNATURE PUBLIC KEYS

After you sign and approve collections, download the signature public keys from the automation hub UI. You must download the public key before you add it to the local system keyring.

**Procedure**

1. Log in to your private automation hub instance in the automation hub UI.

2. In the navigation pane, select **Signature Keys**. The Signature Keys dashboard displays a list of multiple keys: collections and container images.
   - To verify collections, download the key prefixed with `collections-`.
   - To verify container images, download the key prefixed with `container-`.

3. Choose one of the following methods to download your public key:
   - Select the menu icon and click **Download Key** to download the public key.
   - Select the public key from the list and click the **Copy to clipboard** icon.
   - Click the drop-down menu under the **Public Key** tab and copy the entire public key block.

Use the public key that you copied to verify the content collection that you are installing.

### 4.4. 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.
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.

```bash
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.

```yaml
# requirements.yml
collections:
- name: ns.coll
  version: 1.0.0
  signatures:
  - https://examplehost.com/detached_signature.asc
  - file:///path/to/local/detached_signature.asc
```

```bash
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 5. FREQUENTLY ASKED QUESTIONS ABOUT RED HAT ANSIBLE CERTIFIED CONTENT

The following is a list of Frequently Asked Questions for the Red Hat Ansible Automation Platform Certification Program. If you have any questions regarding the following items, email ansiblepartners@redhat.com.

5.1. WHY CERTIFY ANSIBLE COLLECTIONS?

The Ansible certification program enables a shared statement of support for Red Hat Ansible Certified Content between Red Hat and the ecosystem partner. An end customer, experiencing trouble with Ansible and certified partner content, can open a support ticket, for example, a request for information, or a problem with Red Hat, and expect the ticket to be resolved by Red Hat and the ecosystem partner.

Red Hat offers go-to-market benefits for Certified Partners to grow market awareness, demand generation and collaborative selling.

Red Hat Ansible Certified Content Collections are distributed through Ansible automation hub (subscription required), a centralized repository for jointly supported Ansible Content. As a certified partner, publishing collections to Ansible automation hub provides end customers the power to manage how trusted automation content is used in their production environment with a well-known support life cycle.

For more information about getting started with certifying a solution, see Red Hat Partner Connect.

5.2. HOW DO I GET A COLLECTION CERTIFIED?

Refer to Red Hat Partner Connect for the Ansible certification policy guide to understand how to certify your collection.

5.3. WHAT'S THE DIFFERENCE BETWEEN ANSIBLE GALAXY AND ANSIBLE AUTOMATION HUB?

Collections published to Ansible Galaxy are the latest content published by the Ansible community and have no joint support claims associated. Ansible Galaxy is the recommended frontend directory for the Ansible community accessing all content.

Collections published to Ansible automation hub are targeted for joint customers of Red Hat and selected partners. Customers need an Ansible subscription to access and download collections on Ansible automation hub. A certified collection means that Red Hat and partners have a strategic relationship in place and are ready to support joint customers, and may have had additional testing and validation done against them.

5.4. HOW DO I REQUEST A NAMESPACE ON ANSIBLE GALAXY?

After you request a namespace through an Ansible Galaxy GitHub issue, send an email to ansiblepartners@redhat.com You must provide us with the GitHub username that you used to sign up on Ansible Galaxy, and you must have logged in at least once for the system to validate. When users are added as administrators of the namespace, then additional administrators can be added by the self-serve process.
5.5. ARE THERE ANY RESTRICTIONS FOR ANSIBLE GALAXY NAMESPACE NAMING?

Collection namespaces must follow python module name convention. This means collections should have short, all lowercase names. You can use underscores in the collection name if it improves readability.

5.6. ARE THERE ANY RECOMMENDATIONS FOR COLLECTION NAMING?

A general suggestion is to create a collection with `company_name.product` format. This way multiple products may have different collections under the company namespace.

5.7. HOW DO I GET A NAMESPACE ON ANSIBLE AUTOMATION HUB?

By default namespaces used on Ansible Galaxy are also used on Ansible automation hub by the Ansible partner team. For any queries and clarifications contact ansiblepartners@redhat.com.

5.8. HOW DO I RUN SANITY TESTS ON MY COLLECTION?

Ansible sanity tests are made up of scripts and tools used to perform static code analysis. The primary purpose of these tests is to enforce Ansible coding standards and requirements. Ansible collections must be in a specific path, such as the following example:

```
{...}/ansible_collections/{namespace}/{collection}/
```

Ensure that your collection is in that specific path, and that you have three directories:

- An empty directory named `ansible_collections`
- A directory for the namespace
- A directory for the collection itself

5.9. DOES ANSIBLE GALAXY HOUSE THE SOURCE CODE FOR MY COLLECTION?

No, Ansible Galaxy does not house the source for the collections. The actual collection source must be housed outside of Ansible Galaxy, for example, in GitHub. Ansible Galaxy contains the collection build tarball to publish the collection. You can include the link to the source for community users in the `galaxy.yml` file contained in the collection. This shows users where they should go if they want to contribute to the collection or even file issues against it.

5.10. DOES RED HAT OFFICIALLY SUPPORT COLLECTIONS DOWNLOADED AND INSTALLED FROM ANSIBLE GALAXY

No, collections downloaded from Galaxy do not have any support claims associated and are 100% community supported. Users and contributors of any such collection must contact the collection developers directly.
5.11. HOW DOES THE JOINT SUPPORT AGREEMENT ON CERTIFIED COLLECTIONS WORK?

If a customer raises an issue with the Red Hat support team about a certified collection, Red Hat support assesses the issue and checks whether the problem exists within Ansible or Ansible usage. They also check whether the issue is with a certified collection. If there is a problem with the certified collection, support teams transfer the issue to the vendor owner of the certified collection through an agreed upon tool such as TSANet.

5.12. CAN I CREATE AND CERTIFY A COLLECTION CONTAINING ONLY ANSIBLE ROLES?

You can create and certify collections that contain only roles. Current testing requirements are focused on collections containing modules, and additional resources are currently in progress for testing collections only containing roles. Please contact ansiblepartners@redhat.com for more information.
CHAPTER 6. ANSIBLE VALIDATED CONTENT

Red Hat Ansible Automation Platform includes Ansible validated content, which complements existing Red Hat Ansible Certified Content.

Ansible validated content provides an expert-led path for performing operational tasks on a variety of platforms including both Red Hat and our trusted partners.

6.1. CONFIGURING VALIDATED COLLECTIONS WITH THE INSTALLER

When you download and run the bundle installer, certified and validated collections are automatically uploaded. Certified collections are uploaded into the `rh-certified` repository. Validated collections are uploaded into the `validated` repository.

You can change to default configuration by using two variables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>automationhub_seed_collections</td>
<td>A boolean that defines whether or not preloading is enabled.</td>
</tr>
<tr>
<td>automationhub_collection_seed_repository</td>
<td>If <code>automationhub_seed_collections</code> is set to <code>true</code>, this variable enables you to specify the type of content to upload. Possible values are <code>certified</code> or <code>validated</code>. If missing both content sets will be uploaded.</td>
</tr>
</tbody>
</table>

6.2. INSTALLING VALIDATED CONTENT USING THE TARBALL

If you are not using the bundle installer, a standalone tarball, `ansible-validated-content-bundle-1.tar.gz` can be used instead. The standalone tarball can also be used later to update validated contents later in any environment, when a newer tarball becomes available, and without having to re-run the bundle installer.

To obtain the tarball, navigate to the Red Hat Ansible Automation Platform download page and select Ansible Validated Content.

You require the following variables to run the playbook:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>automationhub_admin_password</td>
<td>Your administration password.</td>
</tr>
<tr>
<td>automationhub_api_token</td>
<td>The API token generated for your automation hub.</td>
</tr>
<tr>
<td>automationhub_main_url</td>
<td>For example, <code>https://automationhub.example.com</code></td>
</tr>
</tbody>
</table>
### automationhub_require_content_approval

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>automationhub_require_content_approval</td>
<td>Boolean (true or false)</td>
</tr>
<tr>
<td></td>
<td>This must match the value used during automation hub deployment.</td>
</tr>
<tr>
<td></td>
<td>This variable is set to true by the installer.</td>
</tr>
</tbody>
</table>

**NOTE**

Use either `automationhub_admin_password` or `automationhub_api_token`, not both.

Upload the content and define the variables (this example uses `automationhub_api_token`):

```bash
ansible-playbook collection_seed.yml
-e automationhub_api_token=<api_token>
-e automationhub_main_url=https://automationhub.example.com
-e automationhub_require_content_approval=true
```

For more information on running ansible playbooks, see `ansible-playbook`.

When complete, the collections are visible in the validated collection section of private automation hub.
CHAPTER 7. CONCLUSION

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

- created a synclist for Red Hat Ansible Certified Content content.
- synchronized that content to your private 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 private automation hub.