



Red Hat Ansible Automation Platform 2.3

Managing Red Hat Certified and Ansible Galaxy collections in automation hub

Configure automation hub to deliver curated Red Hat Certified and Ansible Galaxy collections to your users.

Red Hat Ansible Automation Platform 2.3 Managing Red Hat Certified and Ansible Galaxy collections in automation hub

Configure automation hub to deliver curated Red Hat Certified and Ansible Galaxy collections to your users.

Legal Notice

Copyright © 2023 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 ANSIBLE CONTENT COLLECTION SYNCLISTS IN AUTOMATION HUB	5
1.1. ABOUT RED HAT ANSIBLE CERTIFIED CONTENT COLLECTIONS SYNCLISTS	5
1.2. CREATING A SYNCLIST OF RED HAT ANSIBLE CERTIFIED CONTENT COLLECTIONS	5
CHAPTER 2. CONFIGURING ANSIBLE AUTOMATION HUB REMOTE REPOSITORIES TO SYNC CONTENT	7
2.1. REASONS TO CONFIGURE REMOTE REPOSITORIES	7
2.2. RETRIEVING THE SYNC URL AND API TOKEN FOR YOUR RED HAT CERTIFIED COLLECTION	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. PRIVATE AUTOMATION HUB	10
3.1. REQUIRED SHARED FILESYSTEM	10
3.2. SETTING UP THE SHARED FILESYSTEM	10
3.3. ENABLING FIREWALL SERVICES	10
CHAPTER 4. COLLECTIONS AND CONTENT SIGNING IN PRIVATE AUTOMATION HUB	12
4.1. CONFIGURING CONTENT SIGNING ON PRIVATE AUTOMATION HUB	12
4.2. USING CONTENT SIGNING SERVICES IN PRIVATE AUTOMATION HUB	13
4.3. DOWNLOADING SIGNATURE PUBLIC KEYS	14
4.4. CONFIGURING ANSIBLE-GALAXY CLI TO VERIFY COLLECTIONS	14
CHAPTER 5. FREQUENTLY ASKED QUESTIONS ABOUT RED HAT ANSIBLE CERTIFIED CONTENT	16
5.1. WHY CERTIFY ANSIBLE COLLECTIONS?	16
5.2. HOW DO I GET A COLLECTION CERTIFIED?	16
5.3. WHAT'S THE DIFFERENCE BETWEEN ANSIBLE GALAXY AND ANSIBLE AUTOMATION HUB?	16
5.4. HOW DO I REQUEST A NAMESPACE ON ANSIBLE GALAXY?	16
5.5. ARE THERE ANY RESTRICTIONS FOR ANSIBLE GALAXY NAMESPACE NAMING?	17
5.6. ARE THERE ANY RECOMMENDATIONS FOR COLLECTION NAMING?	17
5.7. HOW DO I GET A NAMESPACE ON ANSIBLE AUTOMATION HUB?	17
5.8. HOW DO I RUN SANITY TESTS ON MY COLLECTION?	17
5.9. DOES ANSIBLE GALAXY HOUSE THE SOURCE CODE FOR MY COLLECTION?	17
5.10. DOES RED HAT OFFICIALLY SUPPORT COLLECTIONS DOWNLOADED AND INSTALLED FROM ANSIBLE GALAXY	17
5.11. HOW DOES THE JOINT SUPPORT AGREEMENT ON CERTIFIED COLLECTIONS WORK?	18
5.12. CAN I CREATE AND CERTIFY A COLLECTION CONTAINING ONLY ANSIBLE ROLES?	18
CHAPTER 6. ANSIBLE VALIDATED CONTENT	19
6.1. CONFIGURING VALIDATED OR CERTIFIED COLLECTIONS WITH THE INSTALLER	19
6.2. ANSIBLE VALIDATED CONTENT	19
CHAPTER 7. CONCLUSION	23

PREFACE

Red Hat Ansible Certified Content Collection is included in your subscription to Red Hat Ansible Automation Platform. Red Hat Ansible Certified Content Collection 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.

You can also synchronize Ansible automation hub to create a custom list of collections from Red Hat Ansible Certified Content, Ansible validated content, or Ansible Galaxy community content.

At present, Ansible validated collections are only available in your private automation hub Ansible validated content as part of 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.

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](#).

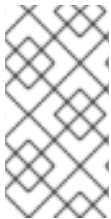
CHAPTER 1. MANAGING ANSIBLE CONTENT COLLECTION SYNCLISTS IN AUTOMATION HUB

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

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.



NOTE

Initially, Ansible validated content is only available through the private automation hub installer. Organization Administrators can preload all Ansible validated content collections at install time and put them into a staging state where they can be reviewed to avoid uploading unnecessary collections.

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** → **Repo Management**, 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**
 - **sso.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 → Repo Management** 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 SYNC CONTENT

You can configure your private automation hub to synchronize with Ansible Certified Content Collections hosted in your organization repository on **console.redhat.com** or to your choice of collections in Ansible Galaxy.

2.1. REASONS TO CONFIGURE REMOTE REPOSITORIES

By configuring remote repositories, you can set your private automation hub to synchronize Red Hat Certified Collections hosted in your organization's repository on **console.redhat.com** and your choice of collections in Ansible Galaxy.

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

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 **Repo Management**.
3. Click the **Remotes** tab.
4. In the **rh-certified** remote repository, click  and click **Edit**.
5. In the modal, paste the **Sync URL** and Token you acquired from console.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.
7. Click **Sync** to synchronize collections.

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 SYNCING 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:

Requirements.yml example

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

Procedure

1. Log in to your Ansible 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 synchronize collections identified in your **requirements.yml** file from Ansible Galaxy to your private automation hub.
7. Click **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
```



```
# 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 AAP installer inventory file for options that begin 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.

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 left navigation, click **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.

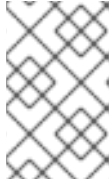
Prerequisites

- 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.

```
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 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 OR CERTIFIED 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:

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

6.2. ANSIBLE VALIDATED CONTENT



NOTE

- Ansible validated content is only available with a valid subscription to Red Hat Ansible Automation Platform.
- Unlike Red Hat Ansible Certified Content, Ansible validated content is not supported by Red Hat or our partners.
- From the Red Hat Ansible Automation Platform 2.3 release, Ansible validated content is preloaded into private automation hub and can be updated manually by downloading the packages.

Entity	Collection name	Description	Published by
Ansible	cloud.azure_roles.load_balancer	A role to manage Azure Load Balancer	Red Hat Ansible
Ansible	cloud.azure_roles.managed_postgresql	A role to manage Azure PostgreSQL Database	Red Hat Ansible

Entity	Collection name	Description	Published by
Ansible	cloud.azure_roles.network_interface	A role to manage Azure Network Interface	Red Hat Ansible
Ansible	cloud.azure_roles.networking_stack	A role to manage Azure Networking Stack	Red Hat Ansible
Ansible	cloud.azure_roles.resource_group	A role to manage Azure Resource Group	Red Hat Ansible
Ansible	cloud.azure_roles.security_group	A role to manage Azure Security Group	Red Hat Ansible
Ansible	cloud.azure_roles.virtual_machine	A role to manage Azure Virtual Machine	Red Hat Ansible
Ansible	network.base	A validated content collection to configure base config related implementation that would be used by other validated content	Red Hat Ansible
Ansible	network.base	A validated content collection to configure bgp and provide capabilities to do operational state/healthchecks	Red Hat Ansible
Ansible	network.acls	A validated content collection to configure acls and provide capabilities to do operational state/healthchecks	Red Hat Ansible
Ansible	network.interfaces	A validated content collection to configure interfaces and provide capabilities to do operational state/healthchecks	Red Hat Ansible
Ansible	network.ospf	A validated content collection to configure ospf and provide capabilities to do operational state/healthchecks	Red Hat Ansible
Ansible	<name yet to decide>	Connectivity between on-prem network device (for ex CSR) and cloud gateway(for ex AWS)	Red Hat Ansible

Entity	Collection name	Description	Published by
Ansible	<name yet to decide>	A validated content (potentially a role) on Inventory Report that returns statistics and IDs of different edge nodes and servers	Red Hat Ansible
Ansible		Network device Inventory report using html	Red Hat Ansible
Ansible		Network config backup	Red Hat Ansible
Ansible		Network config restore	Red Hat Ansible
Ansible		IOS Updater	Red Hat Ansible
Ansible		NXOS Updater	Red Hat Ansible
Ansible		EOS Updater	Red Hat Ansible
Ansible		Firewall policy automation - A validated content to take care of FW policy hygiene	Red Hat Ansible
Ansible		OSbuilder for RHEL Edge disconnected (customer request)	Red Hat Ansible
Ansible		Middleware collection	Red Hat Ansible
Ansible		Windows and Linux Compliance	Red Hat Ansible
Ansible		SAP Deployment	Red Hat Ansible
Ansible		Automation controller configuration	Red Hat Ansible
Ansible		Execution Environment Utilities	Red Hat Ansible
Ansible		Automation hub configuration	Red Hat Ansible
Ansible		AAP utilities	Red Hat Ansible

Entity	Collection name	Description	Published by
Ansible		Role to deploy and migrate a web application on <i>Amazon Web Services (AWS)</i>	Red Hat Ansible
Ansible		Role to deal with AWS orphaned instances by tag	Red Hat Ansible
Ansible		Role to create a customized <i>Amazon Machine Images (AMI)</i>	Red Hat Ansible
Ansible		Role to detach and delete <i>AWS Internet Gateway (IGW)s</i>	Red Hat Ansible
Ansible		Role to configure a multi-region CloudTrail	Red Hat Ansible
Ansible		Role to configure CloudTrail encryption	Red Hat Ansible
Ansible		Role to troubleshoot EC2 instances failing to join an ECS cluster	Red Hat Ansible
Ansible		Role to troubleshoot <i>Relational database Service (RDS)</i> connectivity from an instance	Red Hat Ansible
Ansible		Role to troubleshoot <i>Virtual Private Cloud (VPC)</i> connectivity issues	Red Hat Ansible

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.