

# Certification Workflow Guide

Red Hat Ansible Automation



# Contents

<b>Introduction</b>	<b>2</b>
<b>Certification Checklist</b>	<b>2</b>
STEP 1: Review Certification Program Policies	2
STEP 2: Register with Red Hat Connect	3
STEP 3: Sign Ansible Certification Addendum	3
STEP 4: Develop your Ansible Collection	3
STEP 5: Test your Ansible Collection	4
STEP 6: Request a Galaxy namespace (if applicable)	5
STEP 7: Upload Collection to Ansible Galaxy	6
STEP 8: Request an Ansible Automation Hub Namespace	6
STEP 9: Upload Collection to Automation Hub	7
STEP 10: Publish Collection to Automation Hub	7
<b>Exceptions and Revisions</b>	<b>7</b>
<b>Feedback</b>	<b>7</b>

# Introduction

The Red Hat Ansible Certification Program is available for technology partners that develop and maintain Ansible Collections, and agree to co-support them for joint customers of Red Hat and the partner.

The certification process is intended to help joint customers, using Ansible certified content, to inform them that the partner is an active member of the Ansible Community, and devotes sufficient development and support resources to help resolve any issues that may arise with Partner Ansible content used in production. All Ansible certified content is distributed via Ansible Automation Hub.

This step-by-step workflow guide will help describe the Ansible Certification workflow to partners who are interested in certifying their Ansible Collections.

# Certification Checklist

## **STEP 1: Review Certification Program Policies**

Please visit <http://www.ansible.com/partners> to review the top-level business value summary of the Ansible Certification Program, as well as the rest of the checklist contained in this document.

Partners are expected to also request a Not-for-resale (NFR) subscription via the process outlined in the following [Knowledgebase article](#), which provides access to Ansible Automation Hub on [cloud.redhat.com](http://cloud.redhat.com) once program registration (Step 2 and Step 3) is complete.

Ansible Automation Hub is used to upload new Collection versions and submit them for certification. The certification process assures that a certified Collection meets all the requirements of an Ansible Platform and is jointly supported by Red Hat and your organization.

The certification specific policies and requirements are covered in Red Hat Ansible Automation Certification Policy Guide, located at <http://www.ansible.com/partners>.

## STEP 2: Register with Red Hat Connect

Please visit the following site for information on becoming a partner in order to move through the certification process:

<https://connect.redhat.com/en/partner-with-us/red-hat-ansible-automation-certification>

A step-by-step guide is also available:

<https://redhat-connect.gitbook.io/red-hat-partner-connect-general-guide/initial-onboarding/register>

**NOTE:** It is recommended to register as a New Company during the registration process unless you specifically know who the Organization Admin is for your company (who can authorize your access).

## STEP 3: Sign Ansible Certification Appendix

Before an Ansible Collection can legally be certified, the partner must agree to and sign the Ansible Certification Addendum agreement. .

The 'Legal Agreements' section of the Partner Guide provides instructions on how to view and accept the Ansible Certification Appendix from within the Connect portal.

<https://redhat-connect.gitbook.io/red-hat-partner-connect-general-guide/initial-onboarding/legal-agreements>

[If you need to download a copy of the agreement to review](#), a copy of the agreement can be found at:

<https://www.redhat.com/licenses/Ansible-Certification-Program-Appendix-20210323.pdf>

## STEP 4: Develop your Ansible Collection

A Collection is a standardized schema for Ansible content. You can use Collections to package and distribute such items as playbook examples, roles, modules, plugins, documentation and more. You can publish and use collections through Ansible Galaxy for community consumption and Ansible Automation Hub for certified consumption.



The [Collections Developer Guide](#) will help guide you through the entire process of developing Collections, how the Collection structure is organized and what each folder inside a collection means and should contain.

Also, the [Ansible Collections Checklist](#) is for maintainers of Collections to provide them help, advice, and guidance on making sure their Collections are validated for correctness.

Additional requirements to note when developing your collection for certification:

- Collection Dependencies must come from Automation Hub, the certified collection cannot have a dependency on a collection that is only found on Galaxy
- Automation Hub follows Semantic Versioning explicitly and does not accept Beta or alpha versions of a collection, unlike its upstream counterpart Galaxy. With this, note that your collection must be a minimum of version 1.0.0

## STEP 5: Test your Ansible Collection

`ansible-test`

`ansible-test` is a command line tool that is included with Ansible 2.9 and newer. The `ansible-test` program provides the capability to test your collection by running different test classifications on top of it, such as [sanity, unit, or integration tests](#). For the purposes of Ansible certification, tests of type sanity only are required for inclusion in the Collection.

For `ansible-test` to test your Collection make sure your Collection is present in the following directory structure on your local machine:

```
{...}/ansible_collections/{namespace}/{collection_name}/
```

A Collection can be tested with tests present as part of the Ansible distribution or by additional tests inside the tests directory present inside the Collection. A brief explanation on sanity testing is provided below:

```
pip install ansible==2.9 #or a pre-release version of ansible 2.10
cd {...}/ansible_collections/{namespace}/{collection_name}/

# Run all sanity tests
ansible-test sanity

# Run all sanity tests against against certain files
ansible-test sanity plugins/modules/files/template.py

# Run all tests inside docker (good if you don't have dependencies installed)
ansible-test sanity --docker default

# Run validate-modules test only
ansible-test sanity --test validate-modules

# Run validate-modules against a specific file
ansible-test sanity --test validate-modules lib/ansible/modules/files/template.py
```

## Sanity Tests

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. For purposes of certification, Ansible is enforcing correctness, not functionality. Functional testing is indeed recommended, but not required for certification. Functional testing can be accomplished via unit or integration testing as part of `ansible-test`.

For the purpose of certification, a Collection must only pass sanity testing when uploaded to Ansible Automation Hub, hence, they are the most important test classification from that point of view. Please refer to the below commands to run the Ansible sanity tests.

All variants of sanity tests are available via the [Sanity Testing section on the Developer Guide](#). As a Collection developer passing all sanity tests ensures that your Collection adheres to Ansible coding standards and requirements and also checks if your Collection content can be discovered and used by a local Ansible installation.



## Sanity Ignore Files

**NOTE:** It is possible to ignore some sanity tests for testing purposes. By specifying ignore files under the `tests/sanity` folder by naming them as:

```
ignore-<ansible major.minor version>.txt
```

Example:

```
ignore-2.9.txt
```

```
ignore-2.10.txt
```

For certification purposes this is not recommended as ignoring some sanity tests may result in your Collection not working properly, nor being certified.

## Import Checks

When uploading a collection to your Ansible Galaxy or Ansible Automation Hub namespace, the import is run through technical checks. To duplicate the import checks prior to upload to Ansible Automation Hub (and Ansible Galaxy), there is an import tool (<https://github.com/ansible/galaxy-importer>) that can be downloaded and included in your CI infrastructure.

## Ansible Automation Hub Tagging

For submission to Ansible Automation Hub, the `galaxy.yml` file must contain at least one of the available tags added from the Ansible Automation Hub tag list.

Available tags:

- application
- cloud
- database
- infrastructure
- linux
- monitoring
- networking
- security
- storage
- tools
- windows

## STEP 6: Request a Galaxy namespace (if applicable)

Refer to the following documentation on Galaxy namespacing:

<https://galaxy.ansible.com/docs/contributing/namespaces.html>

### Ansible Galaxy

Ansible Galaxy is a community repository for Ansible Collections that are available to drop directly into your Playbooks to streamline your automation projects. Before certifying a Collection it must be made available to the community as a whole first.

In order to certify an Ansible Collection it must reside in the vendor's Ansible Galaxy namespace upstream.

NOTE: The namespace may already be created, in which case [a request should be made only to be added as an administrator of the namespace](#).

## STEP 7: Upload Collection to Ansible Galaxy

There are two methods for publishing:

1. From the galaxy.ansible.com Web UI
2. From the ansible-galaxy CLI tool

Once you have access to your namespace on Galaxy, you can upload a Collection by following the Publishing Collections section of the [Collections Developer Guide](#).

## STEP 8: Request an Ansible Automation Hub Namespace

This new service is the official location to discover and download supported Ansible Content Collections, included as part of your Ansible Automation Platform subscription. These content Collections contain ways to consume automation, and how-to guides to implement them in your infrastructure.





As an Ansible Partner, Automation Hub is the place where you can publish your Ansible content. Content published on Automation Hub is certified by Red Hat.

Certified content on Automation Hub is co-supported by Red Hat and Red Hat Partners to help joint customers with any issues they face in their automation journey.

Once you have access to Automation Hub [check prerequisites], you can send an email to [ansiblepartners@redhat.com](mailto:ansiblepartners@redhat.com) to create a namespace on Automation Hub.

We would need your account details, that you will get as part of the signup process to associate the namespace with your account.

## STEP 9: Upload Collection to Automation Hub

You would need to request a namespace on Automation Hub and then get admin access to the namespace to upload a Collection.

To get access to Automation Hub, please follow the [process](#) to get an NFR licence and then email the partner engineering team([ansiblepartners@redhat.com](mailto:ansiblepartners@redhat.com)) about the account number that gets created. (see step 8)

You should now be able to login to Automation Hub and upload a Collection from the Web UI or via the `ansible-galaxy` CLI command (similar to Galaxy). Full information can be found at following the guide: [Uploading content to Red Hat Automation Hub](#).

## STEP 10: Publish Collection to Automation Hub

Once the Collection is uploaded, someone from the Partner Engineering team will review and certify the Collection or reply if there are any concerns in certifying the Collection.

## Exceptions and Revisions

Exceptions to this policy can be requested via <https://www.ansible.com/partners/contact>. Exceptions are granted at Red Hat's discretion. Red Hat may change the requirements for



obtaining or maintaining Red Hat certification at any time. Red Hat will give partners advance notification of policy changes.

## Feedback

Please share any comments or feedback on this document to <https://www.ansible.com/partners/contact>.