

Certification Workflow Guide

Red Hat Ansible Automation



Contents

Introduction	2
Certification Checklist	3
STEP 1: Review Certification Program Policies	3
STEP 2: Register with Red Hat Connect	3
STEP 3: Sign Ansible Certification Addendum	3
STEP 4: Register with TSANet	4
STEP 5: Develop your Ansible Collection	4
STEP 6: Test and Lint your Ansible Collection	5
STEP 7: Request a Galaxy namespace (if applicable)	7
STEP 8: Upload Collection to Ansible Galaxy	8
STEP 9: Request a Red Hat Partner Subscription (RHPS)	8
STEP 10: Request an Ansible Automation Hub Namespace	8
STEP 11: Upload Collection to Automation Hub	9
STEP 12: Publish Collection to Automation Hub	9
Life Cycle and Version Support	10
Ansible Partner Announcements	10
Red Hat Ansible Certified Content FAQ	10
Exceptions and Revisions	10
Feedback	10

Introduction

The Red Hat Ansible Certification Program is available for technology partners that develop and maintain Ansible Collections, and who agree to co-support their Collection with Red Hat for the benefit of joint customers.

The Certification process is intended to help joint customers using Ansible Certified Content and to inform them that the Partner devotes sufficient development and support resources to help resolve any issues that may arise with the Partner's Certified Ansible content used in production. All Ansible Certified Content is distributed via Ansible Automation Hub.

Automation Hub is the official location to discover and download Certified and supported Ansible Content Collections, included as part of the Ansible Automation Platform subscription. These Content Collections contain best-practice ways to consume automation and how-to guides for implementing them in your infrastructure.

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

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 review [the top-level business value summary](#) for the Ansible Certification Program, as well as the rest of the checklist contained in this document.

An Onboarding Guide on registration, legal agreements, support process, and subscription access can be found [here](#). This guide will be referenced several times in the steps below.

Ansible Automation Hub is where Partners upload new Collection versions and submit them for certification, and where Certified Content is distributed from. 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 the Red Hat Ansible Automation Certification Policy Guide, located [here](#).

STEP 2: Register with Red Hat Connect

Please visit [this site](#) for information on becoming a partner in order to move through the Certification process.

To access and sign necessary legal documents, access technical resources and more, [register as a partner with Red Hat Connect](#).

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.

A copy of the agreement is available for download [here](#).

STEP 4: Register with TSANet

TSANet (Technical Support Alliance Network) is the vendor-neutral collaborative support framework Red Hat uses to troubleshoot and resolve customer issues with certified products, facilitating 1:1 engagement with Red Hat Global Support Services.

This simplifies the administration and maintenance of the points of contact, as well as clarifies the support process details for both sides. Red Hat covers the cost of this limited subscription as part of the partnership agreement. Documentation can be found here on [how to register for a TSANet membership with Red Hat Partnership](#).

STEP 5: Develop your Ansible Collection

A Collection is a standardized schema for Ansible content. You can use Collections to package and distribute content such 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.

To ensure a consistent customer experience, we ask that maintainers use our [README template](#), which includes multiple sections with information relevant to the collection and its support.

The [Ansible Collections Checklist](#) is available to maintainers of Collections to provide them help, advice, and guidance on making sure their Collections follow best practices.

NOTE: Additional requirements 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.

Event-Driven Ansible Development

Event-Driven Ansible, or EDA, is a type of Ansible content available as of Ansible Automation Platform 2.4. With this type of content, Ansible users can trigger automation based on events, or a certain set of criteria that they can define and monitor.

For guidance on building and testing EDA content, refer to our [EDA content creation guide](#).

STEP 6: Test and Lint 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](#). Passing all sanity tests is required for Certification.

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 Collection's `tests/` directory. A brief explanation on sanity testing is provided below.

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 still recommended, but not required for Certification. Functional testing can be accomplished via unit or integration testing as part of `ansible-test`.

Please refer to the below commands for common ways to run `ansible-test`:



```
$ cd {...}/ansible_collections/{namespace}/{collection_name}/ # navigate to collection
```

```
$ ansible-test sanity # run all sanity tests
```

```
$ ansible-test sanity plugins/modules/files/template.py # run against specified files
```

```
$ ansible-test sanity --docker default # recommended if dependencies aren't installed
```

```
$ ansible-test sanity --test validate-modules # only run a single sanity test
```

```
$ ansible-test sanity --test validate-modules plugins/modules/files/template.py # run a single sanity test against a specified file
```

You can see all sanity tests in the [Sanity Testing section on the Developer Guide](#). For 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

It is possible to ignore some sanity tests for testing purposes. You can specify ignore files in the `tests/sanity` directory by using the naming convention:

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

Example:

```
ignore-2.14.txt  
ignore-2.15.txt
```

NOTE: This is not recommended. Ignoring some sanity tests may result in your Collection not working properly and could prevent certification. Ansible Lint checks for a specific list of sanity ignores, and errors will be displayed if the Collection contains disallowed ignores. More information about Ansible Lint is below.

Ignore All Unsupported Python Versions

To test only specific versions of Python, use the `tests/config.yml` file. A template for this file is located [here](#). Specify the lowest supported Python version, and `ansible-test sanity` will skip all older versions. This is recommended over ignoring unsupported Python versions in an ignore file.



Ansible-lint

Ansible Lint is a command-line tool for linting playbooks, roles and Collections aimed toward any Ansible users. Its main goal is to promote proven practices, patterns and behaviors while avoiding common pitfalls that can easily lead to bugs or make code harder to maintain. Ansible Lint has certain categories of tests, called *Profiles*, that allow content creators to progressively improve the quality of Ansible playbooks, roles, and collections as they increase the profile level.

For Certification, we require that the highest level of ansible-lint profile be passed: the [Production profile](#). This profile enforces all ansible-lint rules.

To [install](#) ansible lint, you can install from dnf, pip3, or from source.

```
$ pip3 install ansible-lint
```

```
$ dnf install ansible-lint
```

```
$ pip3 install git+https://github.com/ansible/ansible-lint
```

The Ansible Lint Github Action can be found [here](#).

Import Checks

When uploading a Collection to your Ansible Galaxy or Ansible Automation Hub namespace, the Collection is run through technical checks to verify that it has been packaged correctly. To duplicate the import checks prior to upload to Ansible Automation Hub (and Ansible Galaxy), you can use the [import tool](#) in your CI pipeline.

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 7: Request a Galaxy namespace (if applicable)

You can read the requirements for Ansible Galaxy namespaces [here](#).

Ansible Galaxy

Uploading your collection(s) to Ansible Galaxy is an optional step that will allow your collection to be accessed by the general public, though we do not require formal support outside of Automation Hub.

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. If for any reason you would like to skip this step, contact ansiblepartners@redhat.com and we can help walk you through ensuring export compliance requirements are met in alternative ways.

NOTE: If the namespace already exists, you can submit [a request to be added as an administrator of the namespace](#).

STEP 8: 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 9: Request a Red Hat Partner Subscription (RHPS)

Partners must request a Red Hat Partner Subscription (RHPS) as outlined [here](#), which provides access to [Ansible Automation Hub](#) after program registration (Steps 2 and 3) is complete.

Once your RHPS is active, you will be able to access [Automation Hub](#).

If you have any issues accessing Automation Hub or requesting your RHPS, please contact our Partner Acceleration Desk at partner-help@redhat.com.

STEP 10: Request an Ansible Automation Hub Namespace

When you have confirmed your access to Automation Hub, you will need to send an email to ansiblepartners@redhat.com with your requested namespace name (mirroring Galaxy, if applicable) and your Red Hat Connect account number. To see your Red Hat Connect account number, click your username drop-down in the top right corner of AutomationHub.

This account number will be added to the namespace to give administrator permissions.

STEP 11: Upload Collection to Automation Hub

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 on publishing with the Galaxy client can be found in the guide: [Getting Started with Automation Hub](#), and more information on Automation Hub uploads and releases can be found here: [Uploading Content to Automation Hub](#).

STEP 12: Publish Collection to Automation Hub

Once the Collection is uploaded, it is placed into a `staging` repository on Automation Hub. Until it is published and certified, it will not be visible on Automation Hub. The Partner Engineering team will review the collection and publish it if it is approved for certification, and it will then be visible on Automation Hub for download. This review and approval process typically



takes one to two business days. Partner Engineering will reach out to you via established channels if there are any concerns with certifying the Collection.

Life Cycle and Version Support

To remain certified, collections have to meet the following criteria:

- Collections must support at least two versions of ansible-core that are components attached to two supported AAP versions. Information on these versions can be found here: [Red Hat Ansible Automation Platform Life Cycle](#)
- Collections must have one or more releases every calendar year.
- Collections must respond to notifications from Partner Engineering about Life Cycle policy violations.
- Collections must be actively maintained and supported.
- Versions older than two years will be removed by Partner Engineering. Partners can request to remove content earlier, if desired.

If a Certified Collection violates these criteria, it is subject to deprecation and removal from the Certified Content Catalog. Partner Engineering will reach out to the Collection's maintainers and contacts before deprecation and removal. If maintainers do not respond or there is no plan to resolve the Life Cycle violations, the deprecation and removal process will proceed.

Ansible Partner Announcements

Subscribe to the Ansible Partner Announcements list by sending an email request to ansiblepartners@redhat.com for periodic updates from the Ansible Partner Team.

Red Hat Ansible Certified Content FAQ

For frequently asked questions and answers about Partnership and Ansible Certified Content, please check out our FAQ here: [Ansible Certification FAQ](#).

Exceptions and Revisions

Exceptions to this policy can be requested by emailing ansiblepartners@redhat.com. 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, typically via the Ansible Partner Announcements mailing list.



Feedback

Please send any comments or feedback on this document to ansiblepartners@redhat.com.