Certification workflow guide

Red Hat Ansible Automation

Red Hat

Certified Technology

Version 2
March 2026

Contents

Introduction

The Red Hat Ansible Certification Program is available for technology partners that develop and maintain Ansible Collections and agree to co-support their Collections 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 Ansible Certified Content used in production. All Ansible Certified Content is distributed through Ansible automation hub.

Automation hub is the official location to discover and download Certified and supported Ansible Content Collections, which are included as part of the Ansible Automation Platform subscription. These Content Collections contain best practices for consuming automation and how-to guides for implementing them in your infrastructure.

All Ansible Certified Content on automation hub is co-supported by Red Hat and our partners to help joint customers with any issues they may face during their automation journey.

This step-by-step workflow guide describes the Ansible certification workflow to partners who are interested in certifying their Ansible Collections.

Red Hat

Certified Technology

Version 2

March 2026

Certification checklist

STEP 1: Review certification program policies

The certification process ensures that a Certified Collection meets all the requirements of Ansible Automation Platform and is jointly supported by Red Hat and your organization.

For certification policies and requirements, see Red Hat Ansible Automation Certification policy guide.

For further information on the Ansible Certification Program, including benefits of partnering with Red Hat Ansible, see Red Hat Ansible Automation Certification.

STEP 2: Register with Red Hat Connect

You must register as a partner with Red Hat Connect in order to access and sign necessary legal documents, access technical resources, and more. To complete your registration, see Red Hat partner onboarding.

STEP 3: Sign the Ansible Certification appendix

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

To view and accept the Ansible Certification Appendix, see Legal Agreements.

STEP 4: Register with Technical Support Alliance Network

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

Use of TSANet simplifies administration and clarifies the support process details for both sides. Red Hat provides this limited, free of charge subscription as part of the partnership agreement.

For more information on membership benefits and details on how to register, see TSANet Membership.

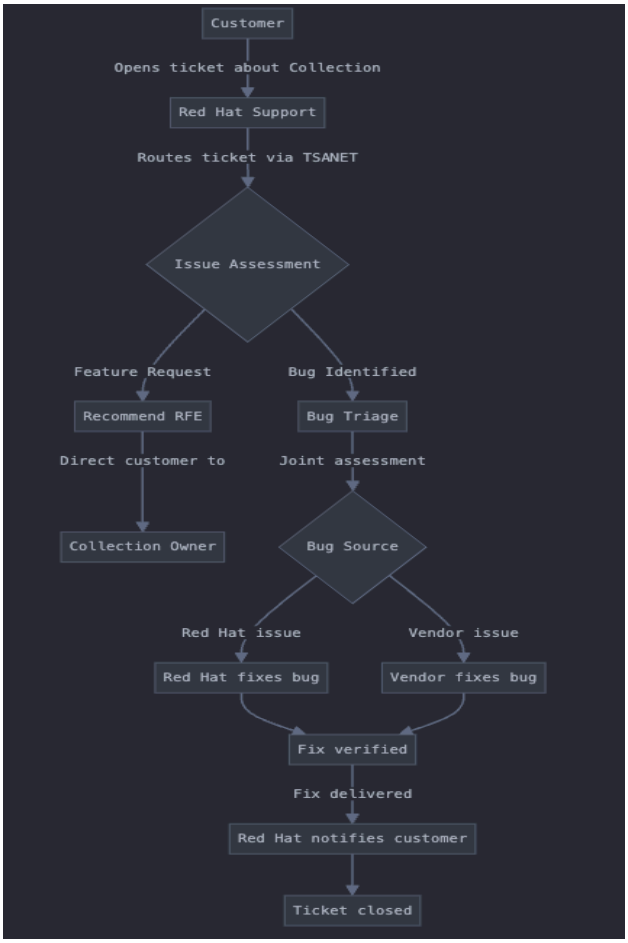

Figure 1: TSANet customer support process

STEP 5: Develop your Ansible Collection

A Collection is a standardized schema for Ansible Content. Collections can be used to package and distribute Content including Playbook examples, Roles, modules, plugins, documentation, and more. Collections may be published and accessed through Ansible Galaxy for community consumption and Ansible automation hub for Certified and validated consumption.

For more information on developing Collections, see Developing Collections.

When developing Collections, use our README template to ensure a consistent customer experience.

For more information on best practices for developing and maintaining Collections, see Ansible community package Collections requirements.

Node counting for Ansible Automation Platform

Your collection affects how Ansible Automation Platform counts managed nodes for customers inventories. Node counting enables customers to track individual automated systems.

Understanding node counting is essential when your collection:

- Interacts with multiple systems which utilize a controller, for example, Azure, Cisco Catalyst Center, or VMWare.
- Orchestrates workflows across platforms
- Manages infrastructure at scale

For detailed information about node counting methodologies, automation execution patterns, and how different types of automation tasks affect node counts, see the Indirect Node Counting Taxonomy Guide.

STEP 6: Test and lint your Ansible Collection

ansible-test

`ansible-test` is a command line tool included with `ansible-core`. The `ansible-test` program provides the capability to run different test classifications on top of your Collection, such as sanity, unit, or integration tests. Your Collection must pass all sanity tests to be Certified.

To use `ansible-test` to test your Collection, you must ensure your Collection is stored using the following directory structure on your local machine:

```
{...}/ansible_Collections/{namespace}/{Collection_name}/
```

For further information on testing, see Testing Collection Contributions.

Sanity testing

Sanity testing includes scripts and tools used to perform static code analysis. The primary purpose of sanity testing is to enforce Ansible coding standards and requirements. For the purposes of certification, Ansible enforces correctness, not functionality. Functional testing is highly recommended, but not required. Functional testing can be accomplished through unit or integration testing as part of `ansible-test`.

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.

Use the following commands 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
```

For more information on and further examples of sanity testing, see Sanity Tests.

Ignore unsupported versions of Python

Use the `tests/config.yml` file to specify which versions of Python you would like to test.

If you specify the lowest supported Python version, then `ansible-test` sanity will skip testing for all previous versions. This method is preferred over using an ignore file to skip testing unsupported versions of Python.

For a template using the `tests/config.yml` file, see config.yml.

Ansible-lint

Ansible-lint is a command line tool for linting Playbooks, Roles, and Collections for Ansible users. Using ansible-lint promotes proven practices, patterns, and behaviors and prevents common pitfalls that can create bugs or make code harder to maintain. Ansible-lint profiles allow content creators to progressively improve the quality of Ansible Playbooks, Roles, and Collections as they increase the profile level. For more information on ansible-lint, see Ansible Lint Documentation.

You must pass the production profile to attain certification. The production profile is the highest level of ansible-lint profile, and it enforces all ansible-lint rules. For more information on the production profile, see Profiles.

You can install ansible-lint from dnf or pip3 using the following commands:

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

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

For more information, see Installing and Using ansible-lint as a GitHub Action.

The certification approval process uses import checks, which include a specific version of ansible-lint. These versions are updated regularly. For more information on which versions you should be using in your tests and CI, see Ansible Certified Content FAQ.

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. You can use the galaxy-importer tool in your CI pipeline to duplicate the import checks prior to uploading your Collection to hub and Galaxy. For more information, see galaxy-importer.

Ansible automation hub tagging

The `galaxy.yml` file you submit to automation hub must contain at least one of the following tags:

- application
- cloud
- database
- eda
- infrastructure
- linux
- monitoring
- networking
- security
- storage
- tools
- window

Step 7: Verify that your Collection meets certification requirements

Certified Collections must meet the following requirements:

- Collection dependencies must come from another Certified Collection in automation hub. The Certified Collection cannot have a dependency on a Collection that is only found on Galaxy.
- Collections must follow semantic versioning, meaning that your Collection must be a minimum of version 1.0.0. For more information, see Semantic Versioning.
    - **Allowed:**
        - 1.0.0
        - 1.5.3
    - **Not allowed:**
        - 0.1.0 Less than 1.0.0
        - 1.2.3-beta1 Beta tag

    - 1.0.0+20130 Doesn't match MAJOR.MINOR.PATCH
    - 1.2.3rc1 Release Candidates (RC) aren't GA

- Collections must be licensed under an OSI license. This license must be included in the Collection tarball or referenced from the Collection `README` file. For more information, see Licenses.
- Collections must have a `requires_ansible` value, which includes a supported `ansible-core` version. For more information, see requires_ansible settings and Red Hat Ansible Automation Platform life cycle.
- External Python dependencies must be listed in a `requirements.txt` file and must be within an open range (`>=`). This prevents issues during the creation of execution environments, particularly when bundling Collections that have conflicting versions of the same dependencies.

  The following style of external Python dependencies are **preferred and allowed**:

  - jinja2 >= 1.2.3 (minimum version, no upper bound)
  - jinja2 (no minimum or maximum version)

  The following style of external Python dependencies are **not preferred, but allowed**:

  - dep <= 1.0.0 (prevents using with newer versions)

  The following style of external Python dependencies are **not allowed**:

  - jinja2 == 3.1.0
- Collections must pass `ansible-test sanity`.
- Collections must pass the ansible-lint production profile with no warnings or errors.
- The Collection `README` file must include the following:
  - Information on the changelog or release notes for the Collection.
  - A support section that clarifies how customers can open a support issue
  - URLs formatted using markdown, for example: [link title](full url).

NOTE: GitHub-relative links do not work inside automation hub.

For more information, see Ansible Certified Collections README Template.

- The Collection must not contain any binary files. Only plugins written in Python or Powershell will be certified.

`requires_ansible` settings

Certified Collections must specify a value for the `requires_ansible` key in the `meta/runtime.yml` file. The value must correspond to one of the `ansible-core` versions that Red Hat Ansible Automation Platform supports, as follows:

- `ansible-core` 2.15 through `ansible-core` 2.18

`ansible-core` 2.19 is in technical preview only and cannot be utilized in Certified Content.

The following table provides upcoming end-of-life (EOL) dates for `ansible-core` versions in Ansible Automation Platform. Red Hat will not certify Collections if the `required_ansible` key specifies an `ansible-core` version that has reached the EOL date.

Ansible Automation Platform Version	**`ansible-core` version**	EOL date
Both	2.18	Sept 30, 2027
Both	2.17	November, 2025
2.5	2.16	Sept 30, 2027
2.4	2.15	June 30, 2026

Table 1: EOL dates for Ansible Automation Platform and `ansible-core`

NOTE: The EOL dates listed in the `ansible-core` support matrix in community documentation do not apply to Certified Collections. These EOL dates are different to the EOL dates for Ansible Automation Platform. Certified Collections must specify `ansible-core` versions in the `requires_ansible` key based on the EOL dates for Ansible Automation Platform only.

STEP 8: Request a Galaxy namespace (Optional)

Uploading your Collection to Ansible Galaxy is an optional step that will allow your Collection to be accessed by the general public. We do not require formal support outside of the automation hub.

Ansible Galaxy is a community repository for Ansible Collections that are available to add directly into your Playbooks to streamline your automation projects.

> **NOTE:** If the namespace already exists, you can submit a request to be added as an administrator of the namespace. To submit this request, see namespace issue request.

For more information on the requirements for Ansible Galaxy namespaces, see galaxy_ng.

STEP 9: Upload Collection to Ansible Galaxy

Upload your Collection using one of the following methods:

- galaxy.ansible.com Web UI
- ansible-galaxy CLI tool

For procedural information on uploading your Collection using the Web UI or CLI tool, see Publishing your Collection.

STEP 10: Request a Red Hat Partner Subscription

Partners must request a Red Hat Partner Subscription (RHPS), which provides access to Ansible automation hub after program registration is complete. For more information, see Software Access.

If you have any issues requesting your RHPS or accessing automation hub, contact our Partner Acceleration Desk by emailing partner-help@redhat.com.

STEP 11: Request an Ansible automation hub namespace

Email ansiblepartners@redhat.com to request an Ansible automation hub namespace. Include the following information in your request:

- Your requested namespace name (mirroring Galaxy, if applicable)
- Your Red Hat Connect account number

  NOTE: To view your Red Hat Connect account number, navigate to the username drop-down menu in automation hub. This account number provides administrator permissions.

STEP 12: Upload your Collection to automation hub

Log in to automation hub, and upload a Collection from the web UI or using the `ansible-galaxy` CLI command. For more information on publishing with the Galaxy client, see Managing automation content, and for more information on automation hub uploads and releases, see Getting started with Ansible Automation Platform.

STEP 13: Publish your Collection to automation hub

Once the Collection is uploaded, it is placed into a `staging` repository on automation hub. It will not be visible until it is published and Certified. The Partner Engineering team will review the Collection. If it is approved for certification, it will be published and available on automation hub for download. Partner Engineering will contact you through the established channels if there are any concerns with certifying your Collection.

Life Cycle and version support

Collections must meet the following criteria to maintain certification:

- Collections must support at least two versions of `ansible-core` attached to two supported versions of Ansible Automation Platform. For more information on versions, see Red Hat Ansible Automation Platform Life Cycle.
- Collections must have at least one release every calendar year.
- Partners must respond to notifications from Partner Engineering about Life Cycle policy violations.
- Collections must be actively maintained and supported.
- Versions that are two years old or older will be removed by Partner Engineering. Partners may request to remove content at any time.

If a Certified Collection violates these criteria, it is subject to deprecation and removal from the Certified Content catalog. Partner Engineering will contact the Collection's maintainers to address Life Cycle violations. The deprecation and removal process will only proceed if there is no response from the maintainer or if the maintainer does not intend to resolve the violations.

Ansible partner announcements

Subscribe to the Ansible partner announcements list for periodic updates from the Ansible partner team. Email your subscription request to ansiblepartners@redhat.com.

Red Hat Ansible Certified Content FAQ

For frequently asked questions and answers about Partnership and Ansible Certified Content, see Ansible Certified Content 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.

Feedback

Email ansiblepartners@redhat.com with feedback or questions regarding this document.