Releasing collections

Collection maintainers release all supported stable versions of the collections regularly, provided that there have been enough changes merged to release.

Preparing to release a collection

The collections under the ansible-collections organization follow semantic versioning when releasing. See :ref:`collection_versioning_and_deprecation` for details.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 19); *backlink*
>
> Unknown interpreted text role "ref".

To prepare for a release, a collection must have:

- A publicly available policy of releasing, versioning, and deprecation. This can be, for example, written in its README or in a dedicated pinned issue.
- A pinned issue when its release managers inform the community about planned or completed releases. This can be combined with the release policy issue mentioned above.
- A :ref:`changelog <collection_changelogs>`.

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 25); *backlink*
  >
  > Unknown interpreted text role "ref".

- Releases of the collection tagged in the collection's repository.
- CI pipelines up and running. This can be implemented by using GitHub Actions, Azure Pipelines, Zuul.
- All CI tests running against a commit that releases the collection. If they do not pass, the collection MUST NOT be released.

See :ref:`including_collection_ansible` if you plan on adding a new collection to the Ansible package.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 30); *backlink*
>
> Unknown interpreted text role "ref".

> **Note**
>
> Your collection must pass `ansible-test sanity` tests. See :ref:`testing_collections` for details.
>
> > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 34); *backlink*
> >
> > Unknown interpreted text role "ref".

Collection versioning and deprecation

> **Note**
>
> Collections MUST adhere to semantic versioning.

To preserve backward compatibility for users, every Ansible minor version series (5.1.x, 5.2.x, and so on) will keep the major version of a collection constant. For example, if Ansible 5.0.0 includes `community.general` 4.0.2, then each Ansible 5.X.x release will include the latest `community.general` 4.y.z release available at build time. Ansible 5.x.x will **never** include a `community.general` 5.y.x release, even if it is available. Major collection version changes will be included in the next Ansible major release (6.0.0 in this case). Ensure that the current major release of your collection included in 6.0.0 receives at least bugfixes as long as new Ansible 6.X.X releases are produced.

Since new minor releases are included, you can include new features, modules and plugins. You must make sure that you do not break backwards compatibility. See semantic versioning. for more details. This means in particular:

- You can fix bugs in **patch** releases but not add new features or deprecate things.
- You can add new features and deprecate things in **minor** releases, but not remove things or change behavior of existing features.
- You can only remove things or make breaking changes in **major** releases.

Ensure that if a deprecation is added in a collection version that is included in 5.x.y, the removal itself will only happen in a collection version included in 7.0.0 or later. Ensure that the policy of releasing, versioning, and deprecation is announced to contributors and users in some way. For an example of how to do this, see the announcement in community.general. You could also do this in the collection README file.

Collection changelogs

Collections MUST include a changelog. To give a consistent feel for changelogs across collections and ensure changelogs exist for collections included in the `ansible` package, we suggest you use antsibull-changelog to maintain and generate this.

Before releasing, verify the following for your changelogs:

- All merged pull requests since the last release, except ones related to documentation and new modules/plugins, have :ref:`changelog fragments <collection_changelog_fragments>`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 67);** ***backlink***
>
> Unknown interpreted text role "ref".

- New module and plugin pull requests, except jinja2 test and filter plugins, do **not** need a changelog fragment, they are auto-detected by the changelog generator by their `version_added` value.
- All the fragments follow the :ref:`changelog entry format <collection_changelogs_how_to_format>`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel)(docs)(docsite)(rst)(community)(collection_contributors)collection_releasing.rst`, line 69);** ***backlink***
>
> Unknown interpreted text role "ref".

Options for releasing a collection

There are several approaches on how to release a collection. If you are not aware of which approach to use, ask in the `#ansible-community` IRC channel or the `community` Matrix channel.

This section assumes that publishing the collection is done with Zuul and that antsibull-changelog is used for the changelog.

Releasing without release branches

Use releasing without release branches when:

- There are no prior major releases of the collection.
- There are no breaking changes introduced since the `1.0.0` release of the collection.

See :ref:`collection_release_without_branches` for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel) (docs) (docsite) (rst) (community) (collection_contributors) collection_releasing.rst, line 86); *backlink*

Unknown interpreted text role "ref".

When there is a need to introduce breaking changes, you can switch to the next approach.

Hybrid approach

In this approach, releases for the current major version are made from the `main` branch, while new releases for older major versions are made from release branches for these versions.

Releasing with release branches

Use releasing with release branches when breaking changes have been introduced. This approach is usually only used by the large community collections, `community.general` and `community.network`.

See :ref:`collection_release_with_branches` for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel) (docs) (docsite) (rst) (community) (collection_contributors) collection_releasing.rst, line 100); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\collection_contributors\(ansible-devel) (docs) (docsite) (rst) (community) (collection_contributors) collection_releasing.rst, line 102)

Unknown directive type "toctree".

```
.. toctree::
   :maxdepth: 1

   collection_release_without_branches
   collection_release_with_branches
```