Releases and maintenance

This section describes release cycles, rules, and maintenance schedules for both Ansible community projects: the Ansible community package and `ansible-core`. The two projects have different versioning systems, maintenance structures, contents, and workflows.

Ansible community package	**ansible-core**
Uses new versioning (2.10, then 3.0.0)	Continues "classic Ansible" versioning (2.11, then 2.12)
Follows semantic versioning rules	Does not use semantic versioning
Maintains only one version at a time	Maintains latest version plus two older versions
Includes language, runtime, and selected Collections	Includes language, runtime, and builtin plugins
Developed and maintained in Collection repositories	Developed and maintained in ansible/ansible repository

Many community users install the Ansible community package. The Ansible community package offers the functionality that existed in Ansible 2.9, with more than 85 Collections containing thousands of modules and plugins. The `ansible-core` option is primarily for developers and users who want to install only the collections they need.

Release cycle overview

The two community releases are related - the release cycle follows this pattern:

1. Release of a new ansible-core major version, for example, ansible-core 2.11
   - New release of ansible-core and two prior versions are now maintained (in this case, ansible-base 2.10, Ansible 2.9)
   - Work on new features for ansible-core continues in the `devel` branch
2. Collection freeze (no new Collections or new versions of existing Collections) on the Ansible community package
3. Release candidate for Ansible community package, testing, additional release candidates as necessary
4. Release of a new Ansible community package major version based on the new ansible-core, for example, Ansible 4.0.0 based on ansible-core 2.11
   - Newest release of the Ansible community package is the only version now maintained
   - Work on new features continues in Collections
   - Individual Collections can make multiple minor and/or major releases
5. Minor releases of three maintained ansible-core versions every three weeks (2.11.1)
6. Minor releases of the single maintained Ansible community package version every three weeks (4.1.0)
7. Feature freeze on ansible-core
8. Release candidate for ansible-core, testing, additional release candidates as necessary
9. Release of the next ansible-core major version, cycle begins again

Ansible community package release cycle

The Ansible community team typically releases two major versions of the community package per year, on a flexible release cycle that trails the release of `ansible-core`. This cycle can be extended to allow for larger changes to be properly implemented and tested before a new release is made available. See :ref:`ansible_roadmaps` for upcoming release details. Between major versions, we release a new minor version of the Ansible community package every three weeks. Minor releases include new backwards-compatible features, modules and plugins, as well as bug fixes.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel)(docs)(docsite)(rst)(reference_appendices)release_and_maintenance.rst`, **line 53**); ***backlink***

Unknown interpreted text role "ref".

Starting with version 2.10, the Ansible community team guarantees maintenance for only one major community package release at a time. For example, when Ansible 4.0.0 gets released, the team will stop making new 3.x releases. Community members may maintain older versions if desired.

Note

Older, unmaintained versions of the Ansible community package might contain unfixed security vulnerabilities (*CVEs*). If you are using a release of the Ansible community package that is no longer maintained, we strongly encourage you to upgrade as soon as possible in order to benefit from the latest features and security fixes.

Each major release of the Ansible community package accepts the latest released version of each included Collection and the latest released version of ansible-core. For specific schedules and deadlines, see the :ref:`ansible_roadmaps` for each version. Major releases of the Ansible community package can contain breaking changes in the modules and other plugins within the included Collections and/or in core features.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 61); ***backlink***

Unknown interpreted text role "ref".

The Ansible community package follows semantic versioning rules. Minor releases of the Ansible community package accept only backwards-compatible changes in included Collections, in other words, not Collections major releases. Collections must also use semantic versioning, so the Collection version numbers reflect this rule. For example, if Ansible 3.0.0 releases with community.general 2.0.0, then all minor releases of Ansible 3.x (such as Ansible 3.1.0 or Ansible 3.5.0) must include a 2.x release of community.general (such as 2.8.0 or 2.9.5) and not 3.x.x or later major releases.

Work in Collections is tracked within the individual Collection repositories.

You can refer to the :ref:`Ansible package porting guides<porting_guides>` for tips on updating your playbooks to run on newer versions of Ansible. For Ansible 2.10 and later releases, you can install the Ansible package with `pip`. See :ref:`intro_installation_guide` for details. For older releases, you can download the Ansible release from https://releases.ansible.com/ansible/.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 67); ***backlink***

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 67); ***backlink***

Unknown interpreted text role "ref".

Ansible community changelogs

This table links to the changelogs for each major Ansible release. These changelogs contain the dates and significant changes in each minor release.

Ansible Community Package Release	Status
6.0.0	In development (unreleased)
5.x Changelogs	Current
4.x Changelogs	End of life after 4.10
3.x Changelogs	Unmaintained (end of life)
2.10 Changelogs	Unmaintained (end of life)

ansible-core release cycle

`ansible-core` is developed and released on a flexible release cycle. This cycle can be extended in order to allow for larger changes to be properly implemented and tested before a new release is made available. See :ref:`ansible_core_roadmaps` for upcoming release details.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 93); ***backlink***

Unknown interpreted text role "ref".

`ansible-core` has a graduated maintenance structure that extends to three major releases. For more information, read about the :ref:`development_and_stable_version_maintenance_workflow` or see the chart in :ref:`release_schedule` for the degrees to which current releases are maintained.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel)(docs)(docsite)(rst)(reference_appendices)release_and_maintenance.rst`, line 95); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel)(docs)(docsite)(rst)(reference_appendices)release_and_maintenance.rst`, line 95); *backlink*

Unknown interpreted text role "ref".

Note

Older, unmaintained versions of `ansible-core` can contain unfixed security vulnerabilities (*CVEs*). If you are using a release of `ansible-core` that is no longer maintained, we strongly encourage you to upgrade as soon as possible to benefit from the latest features and security fixes. `ansible-core` maintenance continues for 3 releases. Thus the latest release receives security and general bug fixes when it is first released, security and critical bug fixes when the next `ansible-core` version is released, and **only** security fixes once the follow on to that version is released.

You can refer to the :ref:`core_porting_guides` for tips on updating your playbooks to run on newer versions of `ansible-core`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel)(docs)(docsite)(rst)(reference_appendices)release_and_maintenance.rst`, line 103); *backlink*

Unknown interpreted text role "ref".

You can install `ansible-core` with `pip`. See :ref:`intro_installation_guide` for details.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel)(docs)(docsite)(rst)(reference_appendices)release_and_maintenance.rst`, line 105); *backlink*

Unknown interpreted text role "ref".

`ansible-core` changelogs

This table links to the changelogs for each major `ansible-core` release. These changelogs contain the dates and significant changes in each minor release.

`ansible-core`/`ansible-base`	Status	Expected end of life
Release		
devel	In development (ansible-core 2.13 unreleased, trunk)	TBD
2.12 ansible-core Changelogs	Maintained (security **and** general bug fixes)	May 2023
2.11 ansible-core Changelogs	Maintained (security **and** critical bug fixes)	Nov 2022
2.10 ansible-base Changelogs	Maintained (security fixes only)	May 2022
2.9 Changelogs	Maintained (pre-collections) (security fixes only)	May 2022
2.8 Changelogs	Unmaintained (end of life)	EOL
2.7 Changelogs	Unmaintained (end of life)	EOL
2.6 Changelogs	Unmaintained (end of life)	EOL
2.5 Changelogs	Unmaintained (end of life)	EOL
<2.5	Unmaintained (end of life)	EOL

Preparing for a new release

Feature freezes

During final preparations for a new release, core developers and maintainers focus on improving the release candidate, not on adding or reviewing new features. We may impose a feature freeze.

A feature freeze means that we delay new features and fixes that are not related to the pending release so we can ship the new release as soon as possible.

Release candidates

We create at least one release candidate before each new major release of Ansible or `ansible-core`. Release candidates allow the Ansible community to try out new features, test existing playbooks on the release candidate, and report bugs or issues they find.

Ansible and `ansible-core` tag the first release candidate (RC1) which is usually scheduled to last five business days. If no major bugs or issues are identified during this period, the release candidate becomes the final release.

If there are major problems with the first candidate, the team and the community fix them and tag a second release candidate (RC2). This second candidate lasts for a shorter duration than the first. If no problems have been reported for an RC2 after two business days, the second release candidate becomes the final release.

If there are major problems in RC2, the cycle begins again with another release candidate and repeats until the maintainers agree that all major problems have been fixed.

Development and maintenance workflows

In between releases, the Ansible community develops new features, maintains existing functionality, and fixes bugs in `ansible-core` and in the collections included in the Ansible community package.

Ansible community package workflow

The Ansible community develops and maintains the features and functionality included in the Ansible community package in Collections repositories, with a workflow that looks like this:

- Developers add new features and bug fixes to the individual Collections, following each Collection's rules on contributing.
- Each new feature and each bug fix includes a changelog fragment describing the work.
- Release engineers create a minor release for the current version every three weeks to ensure that the latest bug fixes are available to users.
- At the end of the development period, the release engineers announce which Collections, and which major version of each included Collection, will be included in the next release of the Ansible community package. New Collections and new major versions may not be added after this, and the work of creating a new release begins.

We generally do not provide fixes for unmaintained releases of the Ansible community package, however, there can sometimes be exceptions for critical issues.

Some Collections are maintained by the Ansible team, some by Partner organizations, and some by community teams. For more information on adding features or fixing bugs in Ansible-maintained Collections, see :ref:`contributing_maintained_collections`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices) release_and_maintenance.rst`, line 193); *backlink***
>
> Unknown interpreted text role "ref".

ansible-core workflow

The Ansible community develops and maintains `ansible-core` on GitHub, with a workflow that looks like this:

- Developers add new features and bug fixes to the `devel` branch.
- Each new feature and each bug fix includes a changelog fragment describing the work.
- The development team backports bug fixes to one, two, or three stable branches, depending on the severity of the bug. They do not backport new features.
- Release engineers create a minor release for each maintained version every three weeks to ensure that the latest bug fixes are available to users.
- At the end of the development period, the release engineers impose a feature freeze and the work of creating a new release begins.

We generally do not provide fixes for unmaintained releases of `ansible-core`, however, there can sometimes be exceptions for critical issues.

For more information on adding features or fixing bugs in `ansible-core` see :ref:`community_development_process`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 208); *backlink*

Unknown interpreted text role "ref".

Generating changelogs

We generate changelogs based on fragments. When creating new features for existing modules and plugins or fixing bugs, create a changelog fragment describing the change. A changelog entry is not needed for new modules or plugins. Details for those items will be generated from the module documentation.

To add changelog fragments to Collections in the Ansible community package, we recommend the antsibull-changelog utility.

To add changelog fragments for new features and bug fixes in `ansible-core`, see the :ref:`changelog examples and instructions<changelogs_how_to>` in the Community Guide.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 221); *backlink*

Unknown interpreted text role "ref".

Deprecation cycles

Sometimes we remove a feature, normally in favor of a reimplementation that we hope does a better job. To do this we have a deprecation cycle. First we mark a feature as 'deprecated'. This is normally accompanied with warnings to the user as to why we deprecated it, what alternatives they should switch to and when (which version) we are scheduled to remove the feature permanently.

Ansible community package deprecation cycle

Since Ansible is a package of individual collections, the deprecation cycle depends on the collection maintainers. We recommend the collection maintainers deprecate a feature in one Ansible major version and do not remove that feature for one year, or at least until the next major Ansible version. For example, deprecate the feature in 3.1.0, and do not remove the feature until 5.0.0, or 4.0.0 at the earliest. Collections should use semantic versioning, such that the major collection version cannot be changed within an Ansible major version. Thus the removal should not happen before the next major Ansible community package release. This is up to each collection maintainer and cannot be guaranteed.

ansible-core deprecation cycle

The deprecation cycle in `ansible-core` is normally across 4 feature releases (2.x.y, where the x marks a feature release and the y a bugfix release), so the feature is normally removed in the 4th release after we announce the deprecation. For example, something deprecated in 2.9 will be removed in 2.13, assuming we do not jump to 3.x before that point. The tracking is tied to the number of releases, not the release numbering.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\reference_appendices\(ansible-devel) (docs) (docsite) (rst) (reference_appendices)release_and_maintenance.rst`, line 238)

Unknown directive type "seealso".

```
.. seealso::

   :ref:`community_committer_guidelines`
       Guidelines for Ansible core contributors and maintainers
   :ref:`testing_strategies`
       Testing strategies
   :ref:`ansible_community_guide`
       Community information and contributing
   `Development Mailing List <https://groups.google.com/group/ansible-devel>`_
       Mailing list for development topics
   :ref:`communication_irc`
       How to join Ansible chat channels
```