Releasing collections without release branches

Since no release branches are used, this section does not distinguish between releasing a major, minor, or patch version.

- Release planning and announcement
- Creating the release branch
- Generating the changelog
- Publish the collection

Release planning and announcement

1. Examine the collection to determine if there are merged changes to release.
2. According to the changes made, choose an appropriate release version number. Keep in mind that the collections must follow the semantic versioning rules. 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_release_without_branches.rst`, **line 17**); ***backlink***
   >
   > Unknown interpreted text role "ref".

3. Announce your intention to release the collection in a corresponding pinned release issue or community pinboard of the collection and in the `community` :ref:`Matrix/IRC channel <communication_irc>`.

   > **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_release_without_branches.rst`, **line 19**); ***backlink***
   >
   > Unknown interpreted text role "ref".

Creating the release branch

1. Ensure you are in a default branch in your local fork. We use `main` in the following examples.

   ```
   git status
   git checkout main     # if needed
   ```

2. Update your local fork:

   ```
   git pull --rebase upstream main
   ```

3. Checkout a new release branch from the default branch:

   ```
   git checkout -b release_branch
   ```

4. Ensure the `galaxy.yml` contains the correct release version number.

Generating the changelog

1. Add a changelog fragment `changelogs/fragments/X.Y.Z.yml` with content:

   ```
   release_summary: |-
     Write some text here that should appear as the release summary for this version.
     The format is reStructuredText, but not a list as for regular changelog fragments.
     This text will be inserted into the changelog.
   ```

   For example:

   ```
   release_summary: |-
     This is the minor release of the ``community.mysql`` collection.
     This changelog contains all changes to the modules and plugins in this collection
     that have been made after the previous release.
   ```

2. If the content was recently moved from another collection (for example, migrating a module from one collection to another), ensure you have all related changelog fragments in the `changelogs/fragments` directory. If not, copy them previously.
3. Run `antsibull-changelog release --reload-plugins` . This package should previously be installed with `pip install antsibull-changelog`.
4. Verify that the `CHANGELOG.rst` looks as expected.
5. Commit and push changes to the `CHANGELOG.rst` and `changelogs/changelog.yaml`, and potentially deleted/archived fragments to the `origin` repository's `release_branch`.

```
git commit -a -m "Release VERSION commit"
git push origin release_branch
```

6. Create a pull request in the collection repository. If CI tests pass, merge it.
7. Checkout the default branch and pull the changes:

```
git checkout main
git pull --rebase upstream main
```

Publish the collection

1. Add an annotated tag to the release commit with the collection version. Pushing this tag to the `upstream` repository will make Zuul publish the collection on Ansible Galaxy.

```
git tag -n    # see current tags and their comments
git tag -a NEW_VERSION -m "comment here"    # the comment can be, for example,  "community.postgresql: 1
git push upstream NEW_VERSION
```

2. Wait until the new version is published on the collection's Ansible Galaxy page. It will appear in a list of tarballs available to download.

3. Update the version in the `galaxy.yml` file to the next **expected** version. Add, commit, and push to the `upstream`'s default branch.

4. Add a GitHub release for the new tag. Title should be the version and content `See https://github.com/ansible-collections/community.xxx/blob/main/CHANGELOG.rst for all changes.`

5. Announce the release through the Bullhorn Newsletter issue.

6. Announce the release in the pinned release issue/community pinboard of the collection mentioned in step 3 and in the `community` :ref:`Matrix/IRC channel <communication_irc>`.

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_release_without_branches.rst`, line 115); ***backlink***

Unknown interpreted text role "ref".