

Managing releases in a repository

In this article

- About release management
- Creating a release
- Editing a release
- Deleting a release

You can create releases to bundle and deliver iterations of a project to users.

Who can use this feature

Repository collaborators and people with write access to a repository can create, edit, and delete a release.

GitHub CLI Web browser

About release management

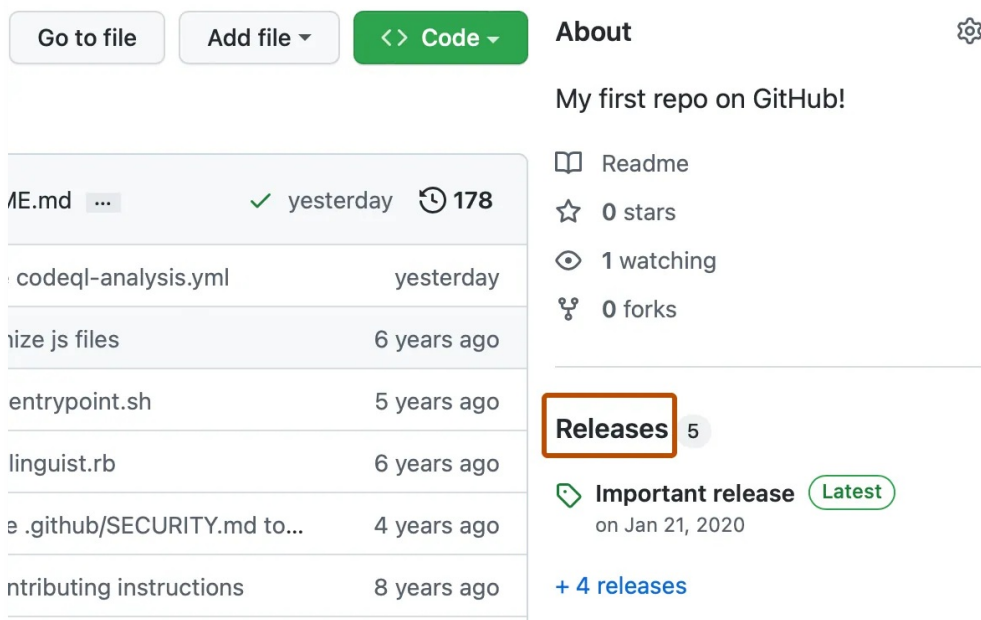
You can create new releases with release notes, @mentions of contributors, and links to binary files, as well as edit or delete existing releases. You can also create, modify, and delete releases by using the Releases API. For more information, see "[Releases](#)" in the REST API documentation.

You can also publish an action from a specific release in GitHub Marketplace. For more information, see "[Publishing actions in GitHub Marketplace](#)."

You can choose whether Git Large File Storage (Git LFS) objects are included in the ZIP files and tarballs that GitHub Enterprise Cloud creates for each release. For more information, see "[Managing Git LFS objects in archives of your repository](#)."

Creating a release

- 1 On GitHub.com, navigate to the main page of the repository.
- 2 To the right of the list of files, click **Releases**.



- 3 At the top of the page, click **Draft a new release**.
- 4 To choose a tag for the release, select the **Choose a tag** dropdown menu.
 - To use an existing tag, click the tag.
 - To create a new tag, type a version number for your release, then click **Create new tag**.
- 5 If you created a new tag, select the **Target** dropdown menu, then click the branch that contains the project you want to release.
- 6 Optionally, above the description field, select the **Previous tag** dropdown menu, then click the tag that identifies the previous release.

- 7 In the "Release title" field, type a title for your release.
- 8 In the "Describe this release" field, type a description for your release. If you @mention anyone in the description, the published release will include a **Contributors** section with an avatar list of all the mentioned users. Alternatively, you can automatically generate your release notes by clicking **Generate release notes**.
- 9 Optionally, to include binary files such as compiled programs in your release, drag and drop or manually select files in the binaries box.
- 10 Optionally, to notify users that the release is not ready for production and may be unstable, select **This is a pre-release**.
- 11 Optionally, select **Set as latest release**. If you do not select this option, the latest release label will automatically be assigned based on semantic versioning.

- 12 Optionally, if GitHub Discussions is enabled for the repository, create a discussion for the release.
 - Select **Create a discussion for this release**.
 - Select the **Category** dropdown menu, then click a category for the release discussion.
- 13 If you're ready to publicize your release, click **Publish release**. To work on the release later, click **Save draft**. You can then view your published or draft releases in the releases feed for your repository. For more information, see "[Viewing your repository's releases and tags](#)."

To learn more about GitHub CLI, see "[About GitHub CLI](#)."

- 1 To create a release, use the `gh release create` subcommand. Replace `tag` with the desired tag for the release.

```
gh release create TAG
```

- 2 Follow the interactive prompts. Alternatively, you can specify arguments to skip these prompts. For more information about possible arguments, see [the GitHub CLI manual](#). For example, this command creates a prerelease with the specified title and notes.

```
gh release create v1.3.2 --title "v1.3.2 (beta)" --notes "this is a beta release" --prerelease
```

If you @mention any GitHub Enterprise Cloud users in the notes, the published release on GitHub.com will include a **Contributors** section with an avatar list of all the mentioned users.

Editing a release [↗](#)

- 1 On GitHub.com, navigate to the main page of the repository.
- 2 To the right of the list of files, click **Releases**.

Go to file
Add file
Code
About

My first repo on GitHub!

ME.md ...
✓ yesterday
🕒 178

codeql-analysis.yml
yesterday

size js files
6 years ago

entrypoint.sh
5 years ago

linguist.rb
6 years ago

e.github/SECURITY.md to...
4 years ago


ntributing instructions
8 years ago

Readme
0 stars
1 watching
0 forks



Releases 5

Important release Latest
on Jan 21, 2020

+ 4 releases

- Next to the release you want to edit, click .

Important release
Latest

This releases important things ⚡

▼ Assets 2

Source code (zip)
Nov 15, 2019

Source code (tar.gz)
Nov 15, 2019

- Edit the details for the release in the form, then click **Update release**. If you add or remove any @mentions of GitHub users in the description, those users will be added or removed from the avatar list in the **Contributors** section of the release.

Releases cannot currently be edited with GitHub CLI.

Deleting a release

- On GitHub.com, navigate to the main page of the repository.
- To the right of the list of files, click **Releases**.

Go to file
Add file
Code
About


My first repo on GitHub!

Readme
0 stars
1 watching
0 forks



Releases 5

Important release Latest
on Jan 21, 2020
+ 4 releases

ME.md ...	✓ yesterday	🕒 178
codeql-analysis.yml	yesterday	
size js files	6 years ago	
entrypoint.sh	5 years ago	
linguist.rb	6 years ago	
e.github/SECURITY.md to...	4 years ago	
contributing instructions	8 years ago	

3 On the right side of the page, next to the release you want to delete, click .

Important release
Latest

This releases important things ⚡

Assets 2

Source code (zip)	Nov 15, 2019
Source code (tar.gz)	Nov 15, 2019

4 Click **Delete this release**.

1 To delete a release, use the `gh release delete` subcommand. Replace `tag` with the tag of the release to delete. Use the `-y` flag to skip confirmation.

```
gh release delete TAG -y
```

Legal