

Troubleshooting creation and deletion of codespaces

In this article

Creating codespaces

Deleting codespaces

Container storage

Configuration

This article provides troubleshooting steps for common issues you may experience when creating or deleting a codespace, including storage and configuration issues.

Creating codespaces [↗](#)

No access to create a codespace [↗](#)

GitHub Codespaces is not available for all repositories. If the options for creating a codespace are not displayed, GitHub Codespaces may not be available for that repository.

If you have remaining monthly included usage of GitHub Codespaces on your personal account, or you have set up a payment method and a spending limit, you can create a codespace for any public repository.

You can also create a codespace for any private repository to which you have at least read access, provided this private repository is owned by a personal account. If a repository is private or internal and is owned by an organization, you may or may not be able to create a codespace for that repository, depending on the settings of the organization or its parent enterprise.

If you can't create a codespace for a repository, this may be due to one of the following organization or enterprise settings.

- Organization and enterprise owners can choose which users can access GitHub Codespaces in an organization's private and internal repositories. For more information, see "[Enabling or disabling GitHub Codespaces for your organization](#)."
- Organization and enterprise owners can disable forking for some or all of an organization's private and internal repositories. If you only have read access to a repository, and you cannot fork it, then you cannot create a codespace for that repository. For more information, see "[Managing the forking policy for your organization](#)."
- If you have a managed user account, you cannot create codespaces for your personal repositories, unless the repository is a fork of an organization-owned repository and you are allowed to use GitHub Codespaces at the organization's expense. For more information, see "[About Enterprise Managed Users](#)."

For information about other organization and enterprise settings that can affect whether you can create a codespace, see "[Enabling or disabling GitHub Codespaces for your](#)

[organization.](#)"

For more information about included usage for personal accounts, and setting a spending limit, see "[About billing for GitHub Codespaces](#)" and "[Managing the spending limit for GitHub Codespaces](#)."

Codespace does not open when created

If you create a codespace and it does not open:

- 1 Try reloading the page in case there was a caching or reporting problem.
- 2 Go to your GitHub Codespaces page: <https://github.com/codespaces> and check whether the new codespace is listed there. The process may have successfully created the codespace but failed to report back to your browser. If the new codespace is listed, you can open it directly from that page.
- 3 Retry creating the codespace for the repository to rule out a transient communication failure.

If you still cannot create a codespace for a repository where GitHub Codespaces is available, you may need to contact support. For more information, see "[Working with support for GitHub Codespaces](#)."

Codespace creation fails

If the creation of a codespace fails, it's likely to be due to a temporary infrastructure issue in the cloud - for example, a problem provisioning a virtual machine for the codespace. A less common reason for failure is if it takes longer than an hour to build the container. In this case, the build is canceled and codespace creation will fail.

Note: A codespace that was not successfully created is never going to be usable and should be deleted. For more information, see "[Deleting a codespace](#)."

If you create a codespace and the creation fails:

- 1 Check GitHub's [Status page](#) for any active incidents.
- 2 Go to [your GitHub Codespaces page](#), delete the codespace, and create a new codespace.
- 3 If the container is building, look at the logs that are streaming and make sure the build is not stuck. A container build that takes longer than one hour will be canceled, resulting in a failed creation.

One common scenario where this could happen is if you have a script running that is prompting for user input and waiting for an answer. If this is the case, remove the interactive prompt so that the build can complete non-interactively.

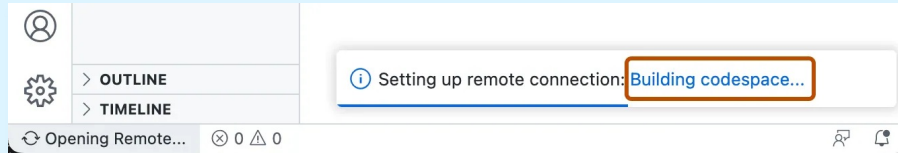
Note: To view the logs during a build:

- **In the browser**, if the initial steps of the build process take more than a few seconds, the "Setting up your codespace" page is displayed. Click **View logs**.

Setting up your codespace

```
✓ Image found.  
::: Building container...  
View logs
```

- In the VS Code desktop application, click **Building codespace** in the "Setting up remote connection" popup message that's displayed.



Log messages are printed to the Terminal in VS Code

- 4 If you have a container that takes a long time to build, consider using prebuilds to speed up codespace creations. For more information, see "[Configuring prebuilds](#)."

Deleting codespaces [🔗](#)

A codespace can only be deleted by:

- The person who created the codespace.
- An organization owner for an organization-owned codespace.
- Automatic deletion at the end of a retention period.

For more information, see "[Deleting a codespace](#)" and "[Configuring automatic deletion of your codespaces](#)."

Container storage [🔗](#)

When you create a codespace, it has a finite amount of storage and over time it may be necessary for you to free up space. Try running any of the following commands in the GitHub Codespaces terminal to free up storage space.

- Remove packages that are no longer used by using `sudo apt autoremove`.
- Clean the apt cache by using `sudo apt clean`.
- See the top 10 largest files in the codespace with `sudo find / -printf '%s %p\n' | sort -nr | head -10`.
- Delete unneeded files, such as build artifacts and logs.

Some more destructive options:

- Remove unused Docker images, networks, and containers by using `docker system prune` (append `-a` if you want to remove all images, and `--volumes` if you want to remove all volumes).
- Remove untracked files from working tree: `git clean -i`.

Configuration [🔗](#)

If changes to your dev container configuration cause a container error, your codespace

will run in recovery mode, and you will see an error message.

This codespace is currently running in recovery mode due to a container error.

Review the creation logs and update the dev container configuration as needed. For more information, see "[GitHub Codespaces logs](#)."

You can then try restarting the codespace, or rebuilding the container. For more information on rebuilding the container, see "[Introduction to dev containers](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)