

Getting the most out of your included usage

In this article

About Codespaces compute

About Codespaces storage

Understanding your Codespaces usage

Tips for making your allowed usage go further

Find out about the free use of GitHub Codespaces that's included with personal accounts.

Personal GitHub accounts include a quota of free use of GitHub Codespaces every month.

Note: Free use of GitHub Codespaces is included in personal accounts only. It is not included in organization or enterprise accounts.

There are two types of Codespaces usage: compute and storage. During your monthly billing period, as you use Codespaces, your compute and storage usage is deducted from the quota of free usage that's included in your personal GitHub account, until either compute or storage is consumed. Once one of those limits is reached, your use of GitHub Codespaces will be restricted, unless you've set up a spending limit and a payment method. You will not be able to create new codespaces or open existing codespaces, that you would be billed for, until your quota renews. If you have access to repositories owned by an organization that pays for use of GitHub Codespaces, you may still be able to create codespaces for those repositories.

The amount of free usage provided on your personal account every month is designed to allow you to make open source contributions, or to work on side projects, free of charge. It is not intended to be enough for you to do everyday work free of charge.











About Codespaces compute [↗](#)

Codespaces compute is counted in core hours, which is the sum of the time a codespace is active, multiplied by the multiplier for the codespace's machine type: for example, a multiplier of 2 for a 2-core machine, or a multiplier of 8 for an 8-core machine. A codespace becomes active when you create it or start it. A codespace stops being active when you stop it or delete it, or when it is stopped or deleted automatically.

The default idle timeout, which stops a codespace after a period of inactivity, is 30 minutes. You can reduce this if required. For more information, see the "About billing for compute usage" section of "[About billing for GitHub Codespaces](#)."

About Codespaces storage [↗](#)

You can see the storage usage for each of your codespaces on the "Your codespaces" page at github.com/codespaces.

Owned by octo-org	
 octo-org/test-repo refactored space palm-tree trial-branch No changes	 16-core • 64GB RAM • 128GB •  5.07 GB •  Active • ...
 octo-org/rails-repo Project X - work in progress feature3000* This codespace has uncommitted changes	 8-core • 32GB RAM • 64GB •  5.27 GB • Last used 1 day ago • ...
 Created from microsoft/vscode-remote-try-php congenial umbrella	 2-core • 8GB RAM • 32GB •  0.86 GB • Last used 1 day ago • ...

Note: If the dev container for a codespace was built from the default image, the size of the codespace shown on this page does not include the size of the base dev container. Storage for the base dev container is provided free of charge. For more information, see "[Storage usage for your base dev container](#)".

For billing purposes, Codespaces storage is counted in GB-months. This is a cumulative measure of the total storage each codespace consumes from creation to deletion, plus the storage for prebuilds. For more information, see the "Billing for storage usage" section of "[About billing for GitHub Codespaces](#)".


Where did my monthly storage go?

Deleting codespaces you're not using will avoid using up the free storage included in your personal account unnecessarily. However, if you have set up prebuild configurations, your included storage may continue to diminish during your monthly billing cycle.

Although prebuilds are not listed on the "Your codespaces" page, prebuilds created for a repository consume storage even if you do not currently have any codespaces for that repository. To avoid this, you can delete the prebuilds configurations you set up. For more information, see "[About GitHub Codespaces prebuilds](#)" and "[Managing prebuilds](#)".

Understanding your Codespaces usage

You can check the cumulative GitHub Codespaces usage for your current monthly billing cycle in your GitHub settings. For more information, see "[Viewing your GitHub Codespaces usage](#)".

 Codespaces Included quotas reset in 20 days. See billing documentation		
>	Usage hours 153.06 of 180.00 included core hours used <div> <div></div> </div>	\$0.00
>	Storage 20.00 of 20.00 included GB-month used <div> <div></div> </div>	\$0.00
\$0.00 monthly spending limit Set up a spending limit		\$0.00

You can expand the "Usage hours" and "Storage" section for more information - for example, to see how much of the consumed storage was used by prebuilds.

For more specific information - for example, if you want to know which repositories have prebuilds that are consuming storage - you can generate a usage report. The usage report is a CSV file that's emailed to you. For more information on how to generate a usage report, see "[Viewing your GitHub Codespaces usage](#)".

To see your Codespaces usage, filter the report to show only rows that mention "Codespaces" in the `Product` column.

A	B	C	D	E	F	G	H	I	J	K
Date	Product	SKU	Quantity	Unit Type	Price Per	Multiplier	Owner	Repository Slug	Username	Actions Workflow
11/10/2022	Codespaces - Linux	Compute - 4 core	1.0555	hour	0.36	1	monalisa	my-demo-repo	monalisa	
11/10/2022	Codespaces - Linux	Storage	0.2627	gb-month	0.07	1	monalisa	my-demo-repo	monalisa	
11/10/2022	Codespaces - Linux	Storage	0.3233	gb-month	0.07	1	monalisa	sample-app-repo	monalisa	
13/10/2022	Codespaces - Linux	Prebuild storage	1.5845	gb-month	0.07	1	monalisa	my-demo-repo		Create Codespaces Prebuilds
14/10/2022	Codespaces - Linux	Prebuild storage	1.9392	gb-month	0.07	1	monalisa	my-demo-repo		Create Codespaces Prebuilds
11/10/2022	Codespaces - Linux	Compute - 4 core	1.9842	hour	0.36	1	monalisa	java-repo	monalisa	
11/10/2022	Codespaces - Linux	Storage	0.4956	gb-month	0.07	1	monalisa	java-repo	monalisa	
11/10/2022	Codespaces - Linux	Compute - 4 core	1.0031	hour	0.36	1	monalisa	sample-app-repo	monalisa	
11/10/2022	Codespaces - Linux	Storage	0.1604	gb-month	0.07	1	monalisa	my-first-app	monalisa	
11/10/2022	Codespaces - Linux	Compute - 4 core	0.5099	hour	0.36	1	monalisa	my-first-app	monalisa	
12/10/2022	Codespaces - Linux	Compute - 4 core	3.9845	hour	0.36	1	monalisa	my-first-app	monalisa	
12/10/2022	Codespaces - Linux	Storage	0.42	gb-month	0.07	1	monalisa	my-first-app	monalisa	

Storage usage for your base dev container [↗](#)

If you don't add a dev container configuration to your repository, or if your configuration does not specify an image to use, then GitHub creates a container from a default Linux image. Storage of base dev containers built from the default Linux image is free of charge and does not consume your included storage. Your storage usage will be based only on the files in your repository, and any files you subsequently add to the codespace, including VS Code extensions. If you use an alternative base image, then the resulting container and all of the files in the codespace will be counted as used storage. For information about what's included in the default Linux image, see the [devcontainers/images](#) repository.

You can check which image was used to create a codespace's dev container. In the Terminal of your codespace, run this command.

Shell

devcontainer-info

If the dev container for the current codespace was built from the default image, the output of this command will contain the following information.

- Definition ID: universal
- Source code repository: https://github.com/devcontainers/images

Tips for making your allowed usage go further [↗](#)


- Your codespaces consume compute usage while they are running. If you're not using a codespace, stopping the codespace prevents unnecessary compute usage. For more information, see "[Stopping and starting a codespace](#)."
- You can reduce the idle timeout for Codespaces in your personal settings to less than the default 30 minutes. This will shorten the period of inactivity before your codespaces are automatically stopped. This can save on compute usage. For more information, see "[Setting your timeout period for GitHub Codespaces](#)."
- Your codespaces consume storage while they exist. You should delete a codespace you have finished using and know that you will not use again. For more information, see "[Deleting a codespace](#)."

Note: Deleting a codespace will not reduce your used storage amount for the billing current month as this is a cumulative figure.

- Configure your retention period to ensure codespaces you forget to delete are

deleted automatically. The default retention period is 30 days. For more information, see "[Configuring automatic deletion of your codespaces](#)."

- Visual Studio Code extensions consume storage. Make sure you are only installing extensions that you need. You can find out how much space is being used by extensions by running this command in your codespace.

Shell 

```
du -h -s ~/.vscode-remote/extensions
```

- Monitor your compute and storage usage by going to your billing page on GitHub.com, <https://github.com/settings/billing>, and reviewing the figures in the "Codespaces" section.

Note: Storage is calculated hourly and added to your existing storage usage. Consumed storage is therefore cumulative for the duration of your month-long billing cycle. This means that, during the billing period, the value you see on your billing page will only increase or remain the same. Usage will be reset to zero when a new billing cycle starts. Deleting a codespace, or a prebuild, will not reduce the usage figure for the current month, but it will reduce the rate at which storage usage accumulates.

- Ensure that you are using prebuilds for only as many versions and as many regions as you need. For more information, see "[About GitHub Codespaces prebuilds](#)" and "[About billing for GitHub Codespaces](#)."

Note: If your included storage usage is exhausted, new prebuilds are disabled until you set up a spending limit or your included usage quota renews.

- If you have configured prebuilds in a repository's settings, but you're not using GitHub Codespaces for that repository, consider deleting the prebuild configuration to avoid prebuilds for that repository consuming your included storage allowance unnecessarily. When you delete a prebuild configuration all the associated prebuilds are deleted, reducing your storage consumption from that point onward. For more information, see "[Managing prebuilds](#)."

You can check for prebuild configurations in the "Codespaces" page of a repository's settings. For more information, see "[Configuring prebuilds](#)."

Alternatively, you can check which repositories have prebuilds by reviewing a usage report. For more information, see "[Understanding your Codespaces usage](#)" above.

- Storage of containers built from the default Linux image for codespaces is free of charge and does not reduce your included storage. You can therefore avoid your storage allowance being consumed by your dev container by using the default image in your dev container configuration, rather than specifying a more specialized image. For more information, see "[Introduction to dev containers](#)" and "[Storage usage for your base dev container](#)" above.

Legal