

About self-hosted runners

Host your own runners

1 of 8 in learning path

Next: [Adding self-hosted runners](#)

In this article

- About self-hosted runners
- Differences between GitHub-hosted and self-hosted runners
- Requirements for self-hosted runner machines
- Autoscaling your self-hosted runners
- Usage limits
- Workflow continuity for self-hosted runners
- Supported architectures and operating systems for self-hosted runners
- Communication between self-hosted runners and GitHub
- Self-hosted runner security

You can host your own runners and customize the environment used to run jobs in your GitHub Actions workflows.

About self-hosted runners

A self-hosted runner is a system that you deploy and manage to execute jobs from GitHub Actions on GitHub.com. For more information about GitHub Actions, see "[Understanding GitHub Actions](#)."

Self-hosted runners offer more control of hardware, operating system, and software tools than GitHub-hosted runners provide. With self-hosted runners, you can create custom hardware configurations that meet your needs with processing power or memory to run larger jobs, install software available on your local network, and choose an operating system not offered by GitHub-hosted runners. Self-hosted runners can be physical, virtual, in a container, on-premises, or in a cloud.

You can add self-hosted runners at various levels in the management hierarchy:

- Repository-level runners are dedicated to a single repository.
- Organization-level runners can process jobs for multiple repositories in an organization.
- Enterprise-level runners can be assigned to multiple organizations in an enterprise account.

Your runner machine connects to GitHub using the GitHub Actions self-hosted runner application. The GitHub Actions runner application is open source. You can contribute and file issues in the [runner](#) repository. When a new version is released, the runner application automatically updates itself when a job is assigned to the runner, or within a week of release if the runner hasn't been assigned any jobs.

A self-hosted runner is automatically removed from GitHub if it has not connected to GitHub Actions for more than 14 days. An ephemeral self-hosted runner is automatically removed from GitHub if it has not connected to GitHub Actions for more than 1 day.

For more information about installing and using self-hosted runners, see "[Adding self-hosted runners](#)" and "[Using self-hosted runners in a workflow](#)."

Differences between GitHub-hosted and self-hosted runners

GitHub-hosted runners offer a quicker, simpler way to run your workflows, while self-hosted runners are a highly configurable way to run workflows in your own custom environment.

GitHub-hosted runners:

- Receive automatic updates for the operating system, preinstalled packages and tools, and the self-hosted runner application.
- Are managed and maintained by GitHub.
- Provide a clean instance for every job execution.
- Use free minutes on your GitHub plan, with per-minute rates applied after surpassing the free minutes.

Self-hosted runners:

- Receive automatic updates for the self-hosted runner application only, though you may disable automatic updates of the runner. For more information about controlling runner software updates on self-hosted runners, see "[Autoscaling with self-hosted runners](#)." You are responsible for updating the operating system and all other software.
- Can use cloud services or local machines that you already pay for.
- Are customizable to your hardware, operating system, software, and security requirements.
- Don't need to have a clean instance for every job execution.
- Are free to use with GitHub Actions, but you are responsible for the cost of maintaining your runner machines.

Requirements for self-hosted runner machines

You can use any machine as a self-hosted runner as long as it meets these requirements:

- You can install and run the self-hosted runner application on the machine. For more information, see "[Supported architectures and operating systems for self-hosted runners](#)."
- The machine can communicate with GitHub Actions. For more information, see "[Communication between self-hosted runners and GitHub](#)."
- The machine has enough hardware resources for the type of workflows you plan to run. The self-hosted runner application itself only requires minimal resources.
- If you want to run workflows that use Docker container actions or service containers, you must use a Linux machine and Docker must be installed.

Autoscaling your self-hosted runners

You can automatically increase or decrease the number of self-hosted runners in your environment in response to the webhook events you receive. For more information, see "[Autoscaling with self-hosted runners](#)."

Usage limits [↗](#)

There are some limits on GitHub Actions usage when using self-hosted runners. These limits are subject to change.

- **Workflow run time** - Each workflow run is limited to 35 days. If a workflow run reaches this limit, the workflow run is cancelled. This period includes execution duration, and time spent on waiting and approval.
- **Job queue time** - Each job for self-hosted runners that has been queued for at least 24 hours will be canceled. The actual time in queue can reach up to 48 hours before cancellation occurs. If a self-hosted runner does not start executing the job within this limit, the job is terminated and fails to complete.
- **API requests** - You can execute up to 1,000 requests to the GitHub API in an hour across all actions within a repository. If requests are exceeded, additional API calls will fail which might cause jobs to fail.
- **Job matrix** - A job matrix can generate a maximum of 256 jobs per workflow run. This limit applies to both GitHub-hosted and self-hosted runners.
- **Workflow run queue** - No more than 500 workflow runs can be queued in a 10 second interval per repository. If a workflow run reaches this limit, the workflow run is terminated and fails to complete.
- **Registering self-hosted runners** - You can have a maximum of 10,000 self-hosted runners in one runner group. If this limit is reached, adding a new runner will not be possible.

Workflow continuity for self-hosted runners [↗](#)

If GitHub Actions services are temporarily unavailable, then a workflow run is discarded if it has not been queued within 30 minutes of being triggered. For example, if a workflow is triggered and the GitHub Actions services are unavailable for 31 minutes or longer, then the workflow run will not be processed.

Supported architectures and operating systems for self-hosted runners [↗](#)

The following operating systems are supported for the self-hosted runner application.

Linux [↗](#)

- Red Hat Enterprise Linux 7 or later
- CentOS 7 or later
- Oracle Linux 7 or later
- Fedora 29 or later
- Debian 9 or later
- Ubuntu 16.04 or later
- Linux Mint 18 or later
- openSUSE 15 or later
- SUSE Enterprise Linux (SLES) 12 SP2 or later

Windows [↗](#)

- Windows 7 64-bit
- Windows 8.1 64-bit
- Windows 10 64-bit

- Windows Server 2012 R2 64-bit
- Windows Server 2016 64-bit
- Windows Server 2019 64-bit
- Windows Server 2022 64-bit

macOS

- macOS 11.0 (Big Sur) or later

Architectures

The following processor architectures are supported for the self-hosted runner application.

- `x64` - Linux, macOS, Windows.
- `ARM64` - Linux, macOS, Windows (currently in beta).
- `ARM32` - Linux.

Communication between self-hosted runners and GitHub

The self-hosted runner connects to GitHub to receive job assignments and to download new versions of the runner application. The self-hosted runner uses an HTTPS *long poll* that opens a connection to GitHub for 50 seconds, and if no response is received, it then times out and creates a new long poll. The application must be running on the machine to accept and run GitHub Actions jobs.

Since the self-hosted runner opens a connection to GitHub.com, you do not need to allow GitHub to make inbound connections to your self-hosted runner.

You must ensure that the machine has the appropriate network access to communicate with the GitHub hosts listed below. Some hosts are required for essential runner operations, while other hosts are only required for certain functionality.

Note:

Some of the domains listed below are configured using `CNAME` records. Some firewalls might require you to add rules recursively for all `CNAME` records. Note that the `CNAME` records might change in the future, and that only the domains listed below will remain constant.

Needed for essential operations:

Shell

github.com

api.github.com

*.actions.githubusercontent.com

Needed for downloading actions:

Shell

codeload.github.com

Needed for uploading/downloading job summaries, logs, workflow artifacts, and caches:

Shell

```
actions-results-receiver-production.githubapp.com
*.blob.core.windows.net
```

Needed for runner version updates:

Shell

```
objects.githubusercontent.com
objects-origin.githubusercontent.com
github-releases.githubusercontent.com
github-registry-files.githubusercontent.com
```

Needed for retrieving OIDC tokens:

Shell

```
*.actions.githubusercontent.com
```

Needed for downloading or publishing packages or containers to GitHub Packages:

Shell

```
*.pkg.github.com
ghcr.io
```

In addition, your workflow may require access to other network resources.

If you use an IP address allow list for your GitHub organization or enterprise account, you must add your self-hosted runner's IP address to the allow list. For more information, see "[Managing allowed IP addresses for your organization](#)" or "[Enforcing policies for security settings in your enterprise](#)" in the GitHub Enterprise Cloud documentation.

You can also use self-hosted runners with a proxy server. For more information, see "[Using a proxy server with self-hosted runners](#)."

For more information about troubleshooting common network connectivity issues, see "[Monitoring and troubleshooting self-hosted runners](#)."

Self-hosted runner security

We recommend that you only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

This is not an issue with GitHub-hosted runners because each GitHub-hosted runner is always a clean isolated virtual machine, and it is destroyed at the end of the job execution.

Untrusted workflows running on your self-hosted runner pose significant security risks for your machine and network environment, especially if your machine persists its environment between jobs. Some of the risks include:

- Malicious programs running on the machine.
- Escaping the machine's runner sandbox.
- Exposing access to the machine's network environment.

- Persisting unwanted or dangerous data on the machine.

For more information about security hardening for self-hosted runners, see "[Security hardening for GitHub Actions](#)."

Restricting the use of self-hosted runners

Organization owners can disable the ability to create self-hosted runners at the repository level. For more information, see "[Disabling or limiting GitHub Actions for your organization](#)."

Next

[Adding self-hosted runners](#)

Legal

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