

About GitHub-hosted runners

GitHub offers hosted virtual machines to run workflows. The virtual machine contains an environment of tools, packages, and settings available for GitHub Actions to use.

In this article

[Overview of GitHub-hosted runners](#)

[Using a GitHub-hosted runner](#)

[Viewing available runners for a repository](#)

[Supported runners and hardware resources](#)

[Runner Images](#)

[Cloud hosts used by GitHub-hosted runners](#)

[Workflow continuity](#)

[Administrative privileges](#)

[IP addresses](#)

[Communication requirements for GitHub-hosted runners](#)

[The etc/hosts file](#)

[File systems](#)

[Further reading](#)

Overview of GitHub-hosted runners

Runners are the machines that execute jobs in a GitHub Actions workflow. For example, a runner can clone your repository locally, install testing software, and then run commands that evaluate your code.

GitHub provides runners that you can use to run your jobs, or you can [host your own runners](#). Each GitHub-hosted runner is a new virtual machine (VM) hosted by GitHub with the runner application and other tools preinstalled, and is available with Ubuntu Linux, Windows, or macOS operating systems. When you use a GitHub-hosted runner, machine maintenance and upgrades are taken care of for you.

You can choose one of the standard GitHub-hosted runner options or, if you are on the GitHub Team or GitHub Enterprise Cloud plan, you can provision a runner with more cores, or a runner that's powered by a GPU processor. These machines are referred to as "larger runner." For more information, see [About larger runners](#).

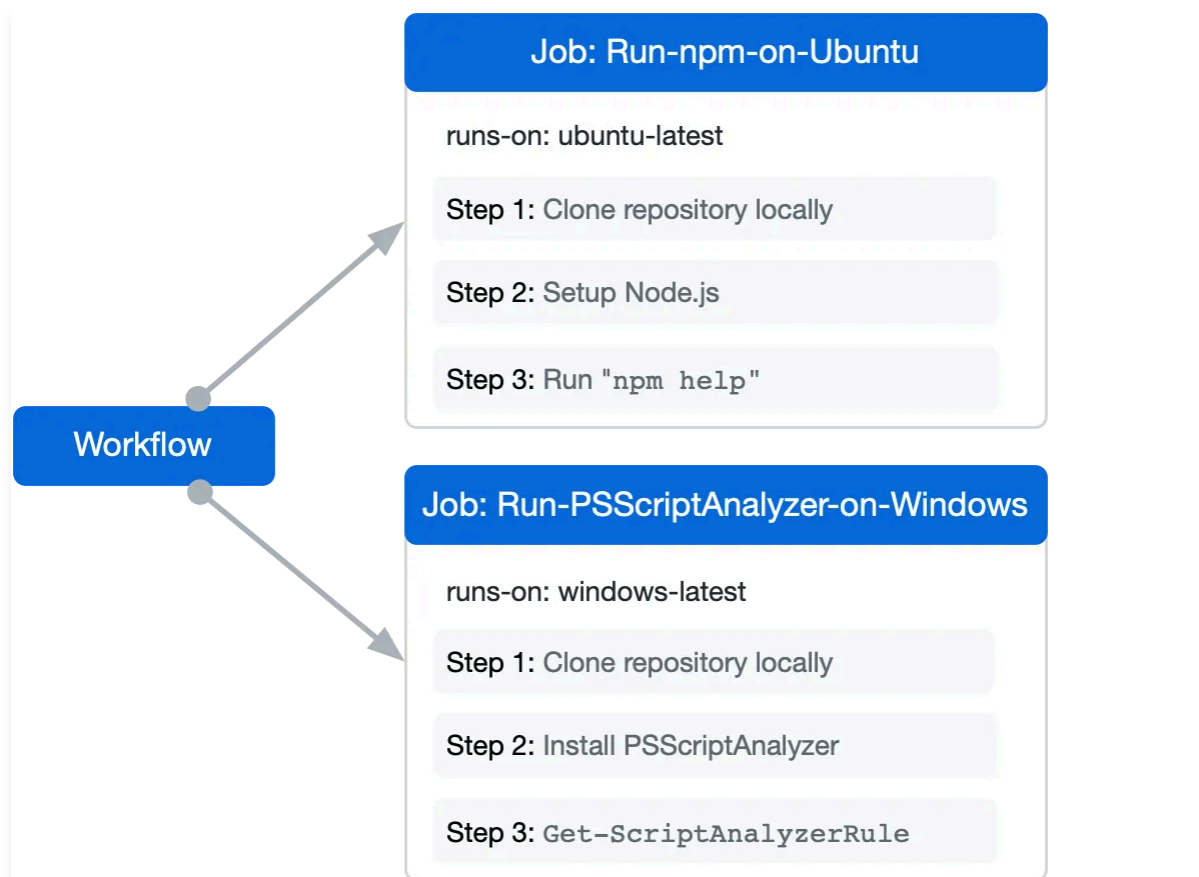
Using GitHub-hosted runners requires network access with at least 70 kilobits per second upload and download speeds.

Using a GitHub-hosted runner

To use a GitHub-hosted runner, create a job and use `runs-on` to specify the type of runner that will process the job, such as `ubuntu-latest`, `windows-latest`, or `macos-latest`. For the full list of runner types, see [About GitHub-hosted runners](#). If you have `repo: write` access to a repository, you can view a list of the runners available to use in workflows in the repository. For more information, see [Viewing available runners for a repository](#).

When the job begins, GitHub automatically provisions a new VM for that job. All steps in the job execute on the VM, allowing the steps in that job to share information using the runner's filesystem. You can run workflows directly on the VM or in a Docker container. When the job has finished, the VM is automatically decommissioned.

The following diagram demonstrates how two jobs in a workflow are executed on two different GitHub-hosted runners.



The following example workflow has two jobs, named `Run-npm-on-Ubuntu` and `Run-PSScriptAnalyzer-on-Windows`. When this workflow is triggered, GitHub provisions a new virtual machine for each job.

- The job named `Run-npm-on-Ubuntu` is executed on a Linux VM, because the job's `runs-on:` specifies `ubuntu-latest`.
- The job named `Run-PSScriptAnalyzer-on-Windows` is executed on a Windows VM, because the job's `runs-on:` specifies `windows-latest`.

YAML



```
name: Run commands on different operating systems
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  Run-npm-on-Ubuntu:
    name: Run npm on Ubuntu
    runs-on: ubuntu-latest
    steps:
```

```
- uses: actions/checkout@v4
- uses: actions/setup-node@v4
  with:
    node-version: '14'
- run: npm help
```

Run-PSScriptAnalyzer-on-Windows:

```
name: Run PSScriptAnalyzer on Windows
runs-on: windows-latest
steps:
- uses: actions/checkout@v4
- name: Install PSScriptAnalyzer module
  shell: pwsh
  run: |
    Set-PSRepository PSGallery -InstallationPolicy Trusted
    Install-Module PSScriptAnalyzer -ErrorAction Stop
- name: Get list of rules
  shell: pwsh
  run: |
    Get-ScriptAnalyzerRule
```

While the job runs, the logs and output can be viewed in the GitHub UI:

The screenshot displays the GitHub Actions interface. On the left, a sidebar shows the 'Summary' tab selected, with a list of jobs under 'Jobs'. The job 'Run PSScriptAnalyzer on Wi...' is highlighted. Below this, 'Run details' are visible, including 'Usage' and 'Workflow file'. The main area on the right shows a dark-themed panel for the selected job, titled 'Run PSScriptAnalyzer on Windows', which 'succeeded now in 1m 23s'. A list of steps follows, each with a green checkmark icon and a right-pointing chevron: 'Set up job', 'Run actions/checkout@v3', 'Install PSScriptAnalyzer module', 'Get list of rules', 'Post Run actions/checkout@v3', and 'Complete job'.

Summary

Jobs

- Run npm on Ubuntu
- Run PSScriptAnalyzer on Wi...**

Run details

- Usage
- Workflow file

Run PSScriptAnalyzer on Windows


succeeded now in 1m 23s

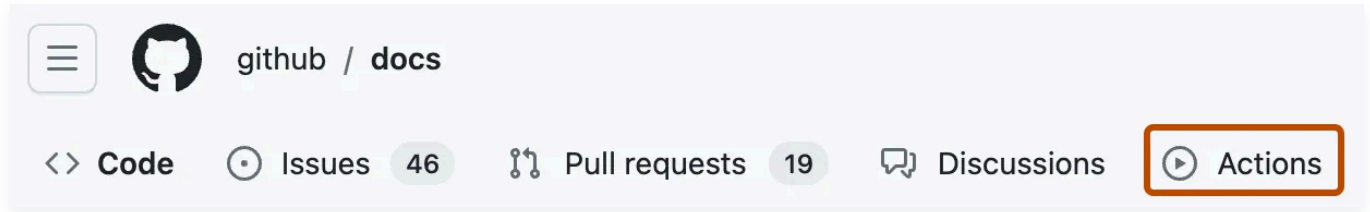
- > ✓ Set up job
- > ✓ Run actions/checkout@v3
- > ✓ Install PSScriptAnalyzer module
- > ✓ Get list of rules
- > ✓ Post Run actions/checkout@v3
- > ✓ Complete job


The GitHub Actions runner application is open source. You can contribute and file issues in the [runner](#) repository.

Viewing available runners for a repository

If you have `repo: write` access to a repository, you can view a list of the runners available to the repository.

- 1 On GitHub, navigate to the main page of the repository.
- 2 Under your repository name, click  **Actions**.



- 3 In the left sidebar, under the "Management" section, click  **Runners**.
- 4 Review the list of available GitHub-hosted runners for the repository.
- 5 Optionally, to copy a runner's label to use it in a workflow, click \cdots to the right of the runner, then click **Copy label**.

Note

Enterprise and organization owners can create runners from this page. To create a new runner, click **New runner** at the top right of the list of runners to add runners to the repository.

For more information, see [Managing larger runners](#) and [Adding self-hosted runners](#).

Supported runners and hardware resources

Ranges of GitHub-hosted runners are available for use in public and private repositories.

For lists of available runners, see:

- [Standard runners for public repositories](#)
- [Standard runners for private repositories](#)

GitHub-hosted Linux runners support hardware acceleration for Android SDK tools, which makes running Android tests much faster and consumes fewer minutes. For more information on Android hardware acceleration, see [Configure hardware acceleration for the Android Emulator](#) in the Android Developers documentation.

Note

The `-latest` runner images are the latest stable images that GitHub provides, and might not be the most recent version of the operating system available from the operating system vendor.

Warning

Beta and Deprecated Images are provided "as-is", "with all faults" and "as available" and are excluded from the service level agreement and warranty. Beta Images may not be covered by customer support.

Standard GitHub-hosted runners for public repositories

For public repositories, jobs using the workflow labels shown in the table below will run on virtual machines with the associated specifications. The use of these runners on public repositories is free and unlimited.

Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	4	16 GB	14 GB	x64	ubuntu-latest , ubuntu-24.04 , ubuntu-22.04 , ubuntu-20.04
Windows	4	16 GB	14 GB	x64	windows-latest , windows-2025 [Public preview], windows-2022 , windows-2019
Linux [Public preview]	4	16 GB	14 GB	arm64	ubuntu-24.04-arm , ubuntu-22.04-arm
macOS	4	14 GB	14 GB	Intel	macos-13

Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
macOS	3 (M1)	7 GB	14 GB	arm64	macos-latest , macos-14 , macos-15 [Public preview]

Note

The arm64 Linux runner is in public preview and subject to change.

Standard GitHub-hosted runners for private repositories

For private repositories, jobs using the workflow labels shown in the table below will run on virtual machines with the associated specifications. These runners use your GitHub account's allotment of free minutes, and are then charged at the per minute rates. For more information, see [About billing for GitHub Actions](#).

Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	2	7 GB	14 GB	x64	ubuntu-latest , ubuntu-24.04 , ubuntu-22.04 , ubuntu-20.04
Windows	2	7 GB	14 GB	x64	windows-latest , windows-2025 [Public preview], windows-2022 , windows-2019
macOS	4	14 GB	14 GB	Intel	macos-13
macOS	3 (M1)	7 GB	14 GB	arm64	macos-latest , macos-14 , macos-15 [Public preview]

Workflow logs list the runner used to run a job. For more information, see [Viewing workflow run history](#).

Limitations for arm64 macOS runners

- All actions provided by GitHub are compatible with arm64 GitHub-hosted runners. However, community actions may not be compatible with arm64 and need to be manually installed at runtime.
- Nested-virtualization and Metal Performance Shaders (MPS) are not supported due to the limitation of Apple's Virtualization Framework.
- Networking capabilities such as Azure private networking and assigning static IPs are not currently available for macOS larger runners.
- The arm64 macOS runners do not have a static UUID/UDID assigned to them because Apple does not support this feature. However, Intel MacOS runners are assigned a static UDID, specifically 4203018E-580F-C1B5-9525-B745CECA79EB . If you are building and signing on the same host you plan to test the build on, you can sign with a [development provisioning profile](#). If you do require a static UDID, you can use Intel runners and add their UDID to your Apple Developer account.

Larger runners

Customers on GitHub Team and GitHub Enterprise Cloud plans can choose from a range of managed virtual machines that have more resources than the [standard GitHub-hosted runners](#). These machines are referred to as "larger runner." They offer the following advanced features:

- More RAM, CPU, and disk space
- Static IP addresses
- Azure private networking
- The ability to group runners
- Autoscaling to support concurrent workflows
- GPU-powered and ARM-powered runners

These larger runners are hosted by GitHub and have the runner application and other tools preinstalled.

For more information, see [Using larger runners](#).

Runner Images

GitHub maintains our own set of VM images for our standard hosted runners. This includes the images for macOS, x64 linux and Windows images. The list of images and their included tools are managed in the [actions/runner-images](https://github.com/actions/runner-images) repository. Our arm64 linux images are partner images, and those are managed in the [actions/partner-runner-images](https://github.com/actions/partner-runner-images) repository.

Preinstalled software for GitHub-owned images

The software tools included in our GitHub-owned images are updated weekly. The update process takes several days, and the list of preinstalled software on the `main` branch is updated after the whole deployment ends.

Workflow logs include a link to the preinstalled tools on the exact runner. To find this information in the workflow log, expand the `Set up job` section. Under that section, expand the `Runner Image` section. The link following `Included Software` will describe the preinstalled tools on the runner that ran the workflow.

For more information, see [Viewing workflow run history](#).

GitHub-hosted runners include the operating system's default built-in tools, in addition to the packages listed in the above references. For example, Ubuntu and macOS runners include `grep`, `find`, and `which`, among other default tools.

You can also view a software bill of materials (SBOM) for each build of the Windows and Ubuntu runner images. For more information, see [Security hardening for GitHub Actions](#).

Using preinstalled software

We recommend using actions to interact with the software installed on runners. This approach has several benefits:

- Usually, actions provide more flexible functionality like version selection, ability to pass arguments, and parameters
- It ensures the tool versions used in your workflow will remain the same regardless of software updates

If there is a tool that you'd like to request, please open an issue at [actions/runner-images](https://github.com/actions/runner-images). This repository also contains announcements about all major software updates on runners.

Installing additional software

You can install additional software on GitHub-hosted runners. For more information, see [Customizing GitHub-hosted runners](#).

Cloud hosts used by GitHub-hosted runners

GitHub hosts Linux and Windows runners on virtual machines in Microsoft Azure with the GitHub Actions runner application installed. The GitHub-hosted runner application is a fork of the Azure Pipelines Agent. Inbound ICMP packets are blocked for all Azure virtual machines, so ping or traceroute commands might not work. GitHub hosts macOS runners in Azure data centers.

For Linux and Windows x64 runners, GitHub uses `Dadsv5-series` virtual machines. For more information, see [Dasv5 and Dadsv5-series](#) in the Microsoft Azure documentation.

For linux arm64 runners, GitHub uses `Dpdsv6-series` virtual machines. For more information, see [Dpdsv6 series](#) in the Microsoft Azure documentation.

GPU runners use `NCasT4_v3-series` virtual machines. For more information, see [NCasT4_v3-series](#) in the Microsoft Azure documentation.

Workflow continuity

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.

In addition, if the workflow run has been successfully queued, but has not been processed by a GitHub-hosted runner within 45 minutes, then the queued workflow run is discarded.

Administrative privileges

The Linux and macOS virtual machines both run using passwordless `sudo`. When you need to execute commands or install tools that require more privileges than the current user, you can use `sudo` without needing to provide a password. For more information, see the [Sudo Manual](#).

Windows virtual machines are configured to run as administrators with User Account Control (UAC) disabled. For more information, see [How User Account Control works](#) in the Windows documentation.

IP addresses

To get a list of IP address ranges that GitHub Actions uses for GitHub-hosted runners, you can use the GitHub REST API. For more information, see the `actions` key in the response of the `GET /meta` endpoint. For more information, see [REST API endpoints for meta data](#).

Windows and Ubuntu runners are hosted in Azure and subsequently have the same IP address ranges as the Azure datacenters. macOS runners are hosted in GitHub's own macOS cloud.

Since there are so many IP address ranges for GitHub-hosted runners, we do not recommend that you use these as allowlists for your internal resources. Instead, we recommend you use larger runners with a static IP address range, or self-hosted runners. For more information, see [Using larger runners](#) or [About self-hosted runners](#).

The list of GitHub Actions IP addresses returned by the API is updated once a week.

Communication requirements for GitHub-hosted runners

A GitHub-hosted runner must establish connections to GitHub-owned endpoints to perform essential communication operations. In addition, your runner may require access to additional networks that you specify or utilize within an action.

To ensure proper communications for GitHub-hosted runners between networks within your configuration, ensure that the following communications are allowed.

Note

Some of the domains listed 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 will remain constant.

Needed for essential operations:

Shell



```
github.com
api.github.com
*.actions.githubusercontent.com
```

Needed for downloading actions:

Shell



codeload.github.com
pkg.actions.githubusercontent.com

Needed for publishing immutable actions:

Shell



ghcr.io

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

Shell



results-receiver.actions.githubusercontent.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
pkg-containers.githubusercontent.com
ghcr.io
```

Needed for Git Large File Storage

Shell



```
github-cloud.githubusercontent.com
github-cloud.s3.amazonaws.com
```

Needed for jobs for Dependabot updates

Shell



```
dependabot-actions.githubapp.com
```

The etc/hosts file

GitHub-hosted runners are provisioned with an `etc/hosts` file that blocks network access to various cryptocurrency mining pools and malicious sites. Hosts such as MiningMadness.com and cpu-pool.com are rerouted to localhost so that they do not present a significant security risk.

File systems

GitHub executes actions and shell commands in specific directories on the virtual machine. The file paths on virtual machines are not static. Use the environment variables GitHub provides to construct file paths for the `home`, `workspace`, and `workflow` directories.

Directory	Environment variable	Description
home	HOME	Contains user-related data. For example, this directory could contain credentials from a login attempt.
workspace	GITHUB_WORKSPACE	Actions and shell commands execute in this directory. An action

Directory	Environment variable	Description
		can modify the contents of this directory, which subsequent actions can access.
workflow/event.json	GITHUB_EVENT_PATH	The POST payload of the webhook event that triggered the workflow. GitHub rewrites this each time an action executes to isolate file content between actions.

For a list of the environment variables GitHub creates for each workflow, see [Store information in variables](#).

Docker container filesystem

Actions that run in Docker containers have static directories under the `/github` path. However, we strongly recommend using the default environment variables to construct file paths in Docker containers.

GitHub reserves the `/github` path prefix and creates three directories for actions.

- `/github/home`
- `/github/workspace` - **Note:** GitHub Actions must be run by the default Docker user (root). Ensure your Dockerfile does not set the `USER` instruction, otherwise you will not be able to access `GITHUB_WORKSPACE`.
- `/github/workflow`

Further reading

- [Managing billing for GitHub Actions](#)
- You can use a matrix strategy to run your jobs on multiple images. For more information, see [Running variations of jobs in a workflow](#).

Legal