

# About larger runners

## In this article

- Overview of larger runners
  - Machine sizes for larger runners
  - About runner groups
  - Architectural overview of larger runners
  - Autoscaling larger runners
  - Networking for larger runners
- 

GitHub offers runners with more RAM, CPU, and disk space.

### Who can use this feature

Larger runners are only available for organizations and enterprises using the GitHub Team or GitHub Enterprise Cloud plans.

## Overview of larger runners

---

In addition to the [standard GitHub-hosted runners](#), GitHub offers customers on GitHub Team and GitHub Enterprise Cloud plans a range of managed virtual machines with more RAM, CPU, and disk space. These runners are hosted by GitHub and have the runner application and other tools preinstalled.

GitHub offers larger runners with macOS, Ubuntu, or Windows operating systems, and different features are available depending on which operating system you use. For more information, see "[Additional features for larger runners](#)."

### About Ubuntu and Windows larger runners

Larger runners with Ubuntu or Windows operating systems are configured in your organization or enterprise. When you add a larger runner, you are defining a type of machine from a selection of available hardware specifications and operating system images. GitHub will then create multiple instances of this runner that scale up and down to match the job demands of your organization, based on the autoscaling limits you define. For more information, see "[Managing larger runners](#)."

Ubuntu and Windows larger runners offer autoscaling capabilities and the ability to assign the runners static IP addresses from a specific range. They can also be managed using runner groups, which enables you to control access to the larger runners. For more information, see "[Additional features for larger runners](#)."

### About macOS larger runners

Larger runners with a macOS operating system are used by updating the YAML workflow label to the desired runner image. To run your workflows on a macOS larger runner, update the `runs-on` key to use one of the GitHub-defined macOS larger runner labels. No additional configuration is required. For more information, see "[Running jobs on larger runners](#)."

The following machines sizes are available for macOS larger runners.

Runner Size	Architecture	Processor (CPU)	Memory (RAM)	Storage (SSD)	YAML workflow label
Large	Intel	12	30 GB	14 GB	macos-latest-large , macos-12-large , macos-13-large [Beta]
XLarge	arm64 (M1)	6 CPU and 8 GPU	14 GB	14 GB	macos-latest-xlarge [Beta], macos-13-xlarge [Beta]

Limitations for macOS larger 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. For more information, see "[Running jobs on larger runners](#)."
- Due to a limitation of Apple's Virtualization Framework, which our hypervisor uses, nested-virtualization is not supported by arm64 runners.

Additional features for larger runners

Compared to standard GitHub-hosted runners, larger runners have additional features, and their availability varies depending on the larger runner's operating system.

Operating system	Ubuntu	Windows	macOS
Hardware acceleration for Android SDK tools	✓	×	×
Static IP addresses	✓	✓	×
Autoscaling	✓	✓	×
Runner groups	✓	✓	×

These features can enhance your CI/CD pipelines in the following ways.

- Hardware acceleration for the Android SDK tools 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.
- Assigning larger runners static IP addresses from a specific range enables you to use this range to configure a firewall allowlist. For more information, see "[Networking for larger runners](#)."
- Autoscaling enables larger runners to scale up to a maximum limit set by you, so your workflows can run concurrently. For more information, see "[Autoscaling larger runners](#)."
- Runner groups allow you to control access to larger runners for your organizations, repositories, and workflows. For more information, see "[Controlling access to larger runners](#)."

For a full list of included tools for each runner operating system, see the [GitHub Actions Runner Images](#) repository.

## Understanding billing [↗](#)

**Note:** Larger runners are not eligible for the use of included minutes on private repositories. For both private and public repositories, when larger runners are in use, they will always be billed at the per-minute rate.

Compared to standard GitHub-hosted runners, larger runners are billed differently. Larger runners are only billed at the per-minute rate for the amount of time workflows are executed on them. There is no cost associated with creating a larger runner that is not being used by a workflow. For more information, see "[About billing for GitHub Actions](#)."

## Machine sizes for larger runners [↗](#)

Processor (CPU)	Memory (RAM)	Storage (SSD)	Operating system (OS)
6	14 GB	14 GB	macOS
12	30 GB	14 GB	macOS
4	16 GB	150 GB	Ubuntu
8	32 GB	300 GB	Ubuntu, Windows
16	64 GB	600 GB	Ubuntu, Windows
32	128 GB	1200 GB	Ubuntu, Windows
64	256 GB	2040 GB	Ubuntu, Windows

## About runner groups [↗](#)

**Note:** Only larger runners with Linux or Windows operating systems can be assigned to runner groups.

Runner groups enable administrators to control access to runners at the organization and enterprise levels. With runner groups, you can collect sets of runners and create a security boundary around them. You can then decide which organizations or repositories are permitted to run jobs on those sets of machines. During the larger runner deployment process, the runner can be added to an existing group, otherwise it will join a default group. You can create a group by following the steps in "[Controlling access to larger runners](#)."

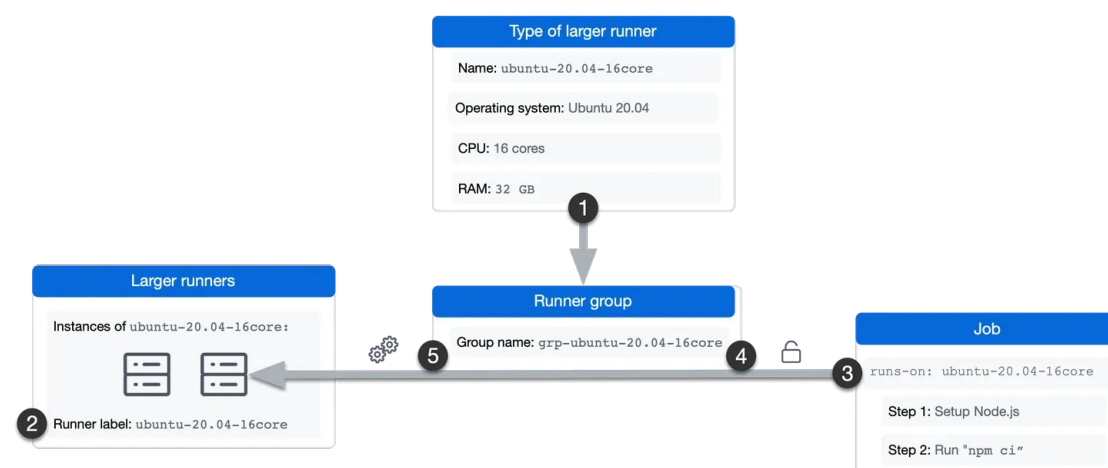
## Architectural overview of larger runners [↗](#)

**Note:** This architecture diagram only applies to larger runners with Linux or Windows operating systems.

Larger runners are managed at the organization level, where they are arranged into groups that can contain multiple instances of the runner. They can also be created at the enterprise level and shared with organizations in the hierarchy. Once you've created a group, you can then add a runner to the group and update your workflows to target

either the group name or the label assigned to the larger runner. You can also control which repositories are permitted to send jobs to the group for processing. For more information about groups, see "[Controlling access to larger runners](#)."

In the following diagram, a class of hosted runner named `ubuntu-20.04-16core` has been defined with customized hardware and operating system configuration.



- 1 Instances of this runner are automatically created and added to a group called `grp-ubuntu-20.04-16core`.
- 2 The runners have been assigned the label `ubuntu-20.04-16core`.
- 3 Workflow jobs use the `ubuntu-20.04-16core` label in their `runs-on` key to indicate the type of runner they need to execute the job.
- 4 GitHub Actions checks the runner group to see if your repository is authorized to send jobs to the runner.
- 5 The job runs on the next available instance of the `ubuntu-20.04-16core` runner.

## Autoscaling larger runners [↗](#)

**Note:** Autoscaling is only available for larger runners with Linux or Windows operating systems.

Larger runners can automatically scale to suit your needs. You can provision machines to run a specified maximum number of jobs when jobs are submitted for processing. Each machine only handles one job at a time, so these settings effectively determine the number of jobs that can be run concurrently.

You can configure the maximum job concurrency, which allows you to control your costs by setting the maximum parallel number of jobs that can be run using this set. A higher value here can help avoid workflows being blocked due to parallelism. For more information, see "[Managing larger runners](#)."

## Networking for larger runners [↗](#)

**Note:** Assigning static IP addresses to runners is only available for larger runners with Linux or Windows operating systems.

By default, larger runners receive a dynamic IP address that changes for each job run. Optionally, GitHub Enterprise Cloud customers can configure their larger runners to receive a static IP address from GitHub's IP address pool. For more information, see

"[About GitHub's IP addresses](#)."

When enabled, instances of the larger runner will receive IP addresses from specific ranges that are unique to the runner, allowing you to use the ranges to configure a firewall allowlist. You can use up to 10 larger runners with static IP address ranges for the larger runners created at the enterprise level. In addition, you can use up to 10 larger runners with static IP address ranges for the larger runners created at the organization level, for each organization in your enterprise. For more information, see "[Managing larger runners](#)."

If you would like to use more than 10 larger runners with static IP address ranges, please contact us through the [GitHub Support portal](#).

**Note:** If runners are unused for more than 30 days, their IP address ranges are automatically removed and cannot be recovered.

## Legal

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