Running jobs on larger runners

In this article

You can speed up your workflows by configuring them to run on larger runners.

> **Who can use this feature**
> Larger runners are only available for organizations and enterprises using the GitHub Team or GitHub Enterprise Cloud plans.

Mac Windows Linux

Running jobs on your runner

Once your runner type has been defined, you can update your workflow YAML files to send jobs to your newly created runner instances for processing. You can use runner groups or labels to define where your jobs run.

Note: Larger runners are automatically assigned a default label that corresponds to the runner name. You cannot add custom labels to larger runners, but you can use the default labels or the runner's group to send jobs to specific types of runners.

Only owner or administrator accounts can see the runner settings. Non-administrative users can contact the organization owner to find out which runners are enabled. Your organization owner can create new runners and runner groups, as well as configure permissions to specify which repositories can access a runner group. For more information, see "Managing larger runners."

Once your runner type has been defined, you can update your workflow YAML files to send jobs to your newly created runner instances for processing. You can use runner groups or labels to define where your jobs run.

Note: Larger runners are automatically assigned a default label that corresponds to the runner name. You cannot add custom labels to larger runners, but you can use the default labels or the runner's group to send jobs to specific types of runners.

Only owner or administrator accounts can see the runner settings. Non-administrative users can contact the organization owner to find out which runners are enabled. Your organization owner can create new runners and runner groups, as well as configure permissions to specify which repositories can access a runner group. For more information, see "Managing larger runners."

Once your runner type has been defined, you can update your workflow YAML files to send jobs to runner instances for processing. To run jobs on macOS larger runners, update the `runs-on` key in your workflow YAML files to use one of the GitHub-defined labels for macOS runners. For more information, see "Available macOS larger runners."

Available macOS larger runners

Use the labels in the table below to run your workflows on the corresponding macOS larger runner.

Runner Size	Architecture	Processor (CPU)	Memory (RAM)	Storage (SSD)	OS (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]

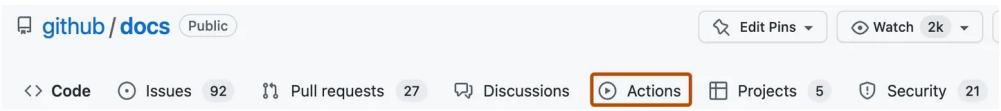
Note: For macOS larger runners, the `-latest` runner label uses the macOS 12 runner image. For macOS Xlarge, the `-latest` runner label uses the macOS 13 runner image

Viewing available runners for a repository

Note: This feature is currently in beta and subject to change.

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

1. On GitHub.com, 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 runners for the repository.
5. Optionally, to copy a runner's label to use it in a workflow, click ··· to the right of the runner, then click **Copy label**.

Note: Enterprise and organization owners with privileges to create runners have the option to create new runners from this page. If you are an enterprise or organization owner, 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."

Using groups to control where jobs are run

In this example, Ubuntu runners have been added to a group called `ubuntu-runners`. The `runs-on` key sends the job to any available runner in the `ubuntu-runners` group:

```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      group: ubuntu-runners
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Using groups to control where jobs are run

In this example, Ubuntu runners have been added to a group called `ubuntu-runners`. The `runs-on` key sends the job to any available runner in the `ubuntu-runners` group:

```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      group: ubuntu-runners
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Using labels to control where jobs are run

In this example, a runner group is populated with Ubuntu 16-core runners, which have also been assigned the label `ubuntu-20.04-16core`. The `runs-on` key sends the job to any available runner with a matching label:

```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      labels: ubuntu-20.04-16core
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
```

```
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Using labels to control where jobs are run

In this example, a runner group is populated with Windows 16-core runners, which have also been assigned the label `windows-2022-16core`. The `runs-on` key sends the job to any available runner with a matching label:

```
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      labels: windows-2022-16core
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Targeting macOS larger runners in a workflow

To run your workflows on macOS larger runners, set the value of the `runs-on` key to a label associated with a macOS larger runner. For a list of macOS larger runner labels, see "Available macOS larger runners."

In this example, the workflow uses a label that is associated with macOS XL runners, which is `macos-latest-xl -arm64`. The `runs-on` key sends the job to any available runner with a matching label:

```
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      labels: macos-latest-xlarge -arm64
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '16'
      - run: npm install -g bats
      - run: bats -v
```

Using labels and groups to control where jobs are run

When you combine groups and labels, the runner must meet both requirements to be eligible to run the job.

In this example, a runner group called `ubuntu-runners` is populated with Ubuntu runners, which have also been assigned the label `ubuntu-20.04-16core`. The `runs-on` key combines `group` and `labels` so that the job is routed to any available runner within the group that also has a matching label:

```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      group: ubuntu-runners
      labels: ubuntu-20.04-16core
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Using labels and groups to control where jobs are run

When you combine groups and labels, the runner must meet both requirements to be eligible to run the job.

In this example, a runner group called `ubuntu-runners` is populated with Ubuntu runners, which have also been assigned the label `ubuntu-20.04-16core`. The `runs-on` key combines `group` and `labels` so that the job is routed to any available runner within the group that also has a matching label:

```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on:
      group: ubuntu-runners
      labels: ubuntu-20.04-16core
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

Troubleshooting larger runners

If you notice the jobs that target your larger runners are delayed or not running, there are several factors that may be causing this.

- **Concurrency settings**: You may have reached your maximum concurrency limit. If you would like to enable more jobs to run in parallel, you can update your autoscaling settings to a larger number. For more information, see "Managing larger runners."
- **Repository permissions**: Ensure you have the appropriate repository permissions enabled for your larger runners. By default, enterprise runners are not available at the repository level and must be manually enabled by an organization administrator. For more information, see "Managing larger runners."
- **Billing information**: You must have a valid credit card on file in order to use larger runners. After adding a credit card to your account, it can take up to 10 minutes to enable the use of your larger runners. For more information, see "Adding or editing a payment method."
- **Spending limit**: Your GitHub Actions spending limit must be set to a value greater than zero. For more information, see "Managing your spending limit for GitHub

Actions."

- **Fair use policy**: GitHub has a fair use policy that begins to throttle jobs based on several factors, such as how many jobs you are running or how many jobs are running across the entirety of GitHub Actions.

If you notice the jobs that target your larger runners are delayed or not running, there are several factors that may be causing this.

- **Concurrency settings**: You may have reached your maximum concurrency limit. If you would like to enable more jobs to run in parallel, you can update your autoscaling settings to a larger number. For more information, see "Managing larger runners."
- **Repository permissions**: Ensure you have the appropriate repository permissions enabled for your larger runners. By default, enterprise runners are not available at the repository level and must be manually enabled by an organization administrator. For more information, see "Managing larger runners."
- **Billing information**: You must have a valid credit card on file in order to use larger runners. After adding a credit card to your account, it can take up to 10 minutes to enable the use of your larger runners. For more information, see "Adding or editing a payment method."
- **Spending limit**: Your GitHub Actions spending limit must be set to a value greater than zero. For more information, see "Managing your spending limit for GitHub Actions."
- **Fair use policy**: GitHub has a fair use policy that begins to throttle jobs based on several factors, such as how many jobs you are running or how many jobs are running across the entirety of GitHub Actions.

Because macOS arm64 does not support Node 12, macOS larger runners automatically use Node 16 to execute any JavaScript action written for Node 12. Some community actions may not be compatible with Node 16. If you use an action that requires a different Node version, you may need to manually install a specific version at runtime.

For example, the `setup-ruby` action must be modified before you can use it on macOS larger runners. The following example shows how to install a specific version of Ruby, if you replace `RUBY_VERSION` with the desired version of Ruby.

YAML

```yaml
- name: Setup Ruby
  run: |
    brew install ruby-build
    ruby-build RUBY_VERSION /Users/runner/hostedtoolcache/Ruby/RUBY_VERSION/arm64
    touch /Users/runner/hostedtoolcache/Ruby/RUBY_VERSION/arm64.complete
- name: Setup Ruby (with Bundler cache)
  uses: ruby/setup-ruby@ec02537da5712d66d4d50a0f33b7eb52773b5ed1
  with:
    ruby-version: RUBY_VERSION
    bundler-cache: true
```

Legal

© 2023 GitHub, Inc. Terms Privacy Status Pricing Expert services Blog