

Using self-hosted runners in a workflow

Host your own runners

6 of 8 in learning path

Next: [Managing access to self-hosted runners using groups](#)

In this article

About self-hosted runner labels

About self-hosted runner groups

Viewing available runners for a repository

Using default labels to route jobs

Using custom labels to route jobs

Using groups to route jobs

Using labels and groups to route jobs

Routing precedence for self-hosted runners

To use self-hosted runners in a workflow, you can use labels or groups to specify the runner for a job.

You can target self-hosted runners for use in a workflow based on the labels assigned to the runners, or their group membership, or a combination of these.

About self-hosted runner labels

Labels allow you to send workflow jobs to specific types of self-hosted runners, based on their shared characteristics. For example, if your job requires a particular hardware component or software package, you can assign a custom label to a runner and then configure your job to only execute on runners with that label.

To specify a self-hosted runner for your job, configure `runs-on` in your workflow file with self-hosted runner labels.

All self-hosted runners have the `self-hosted` label. Using only this label will select any self-hosted runner. To select runners that meet certain criteria, such as operating system or architecture, we recommend providing an array of labels that begins with `self-hosted` (this must be listed first) and then includes additional labels as needed. When you specify an array of labels, jobs will be queued on runners that have all the labels that you specify.

Although the `self-hosted` label is not required, we strongly recommend specifying it when using self-hosted runners to ensure that your job does not unintentionally specify any current or future GitHub-hosted runners.

For information on creating custom and default labels, see "[Using labels with self-hosted runners](#)."

About self-hosted runner groups [↗](#)

For self-hosted runners defined at the organization level, you can group your runners with shared characteristics into a single runner group and then configure your job to target the runner group.


To specify a self-hosted runner group for your job, configure `runs-on.group` in your workflow file.

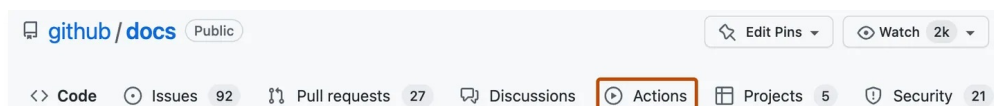
For information on creating and managing runner groups, see "[Managing access to self-hosted runners using groups](#)."


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 Click the **Self hosted** tab at the top of the list of runners.
- 5 Review the list of available self-hosted runners for the repository. This list includes both self-hosted runners and runner scale sets created with Actions Runner Controller. For more information, see "[About Actions Runner Controller](#)."
- 6 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 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 default labels to route jobs [↗](#)

A self-hosted runner automatically receives certain labels when it is added to GitHub Actions. These are used to indicate its operating system and hardware platform:

- `self-hosted` : Default label applied to all self-hosted runners.
- `linux` , `windows` , or `macOS` : Applied depending on operating system.
- `x64` , `ARM` , or `ARM64` : Applied depending on hardware architecture.

You can use your workflow's YAML to send jobs to a combination of these labels. In this example, a self-hosted runner that matches all three labels will be eligible to run the job:

```
runs-on: [self-hosted, linux, ARM64]
```

- `self-hosted` - Run this job on a self-hosted runner.
- `linux` - Only use a Linux-based runner.
- `ARM64` - Only use a runner based on ARM64 hardware.

The default labels are fixed and cannot be changed or removed. Consider using custom labels if you need more control over job routing.

Using custom labels to route jobs [↗](#)

You can create custom labels and assign them to your self-hosted runners at any time. Custom labels let you send jobs to particular types of self-hosted runners, based on how they're labeled.

For example, if you have a job that requires a specific type of graphics hardware, you can create a custom label called `gpu` and assign it to the runners that have the hardware installed. A self-hosted runner that matches all the assigned labels will then be eligible to run the job.

This example shows a job that combines default and custom labels:

```
runs-on: [self-hosted, linux, x64, gpu]
```

- `self-hosted` - Run this job on a self-hosted runner.
- `linux` - Only use a Linux-based runner.
- `x64` - Only use a runner based on x64 hardware.
- `gpu` - This custom label has been manually assigned to self-hosted runners with the GPU hardware installed.

These labels operate cumulatively, so a self-hosted runner must have all four labels to be eligible to process the job.

Using groups to route jobs [↗](#)

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:

```
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 and groups to route jobs [↗](#)

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:

```
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
```

Routing precedence for self-hosted runners

When routing a job to a self-hosted runner, GitHub looks for a runner that matches the job's `runs-on` labels and/or groups:

- If GitHub finds an online and idle runner that matches the job's `runs-on` labels and/or groups, the job is then assigned and sent to the runner.
 - If the runner doesn't pick up the assigned job within 60 seconds, the job is re-queued so that a new runner can accept it.
- If GitHub doesn't find an online and idle runner that matches the job's `runs-on` labels and/or groups, then the job will remain queued until a runner comes online.
- If the job remains queued for more than 24 hours, the job will fail.

Previous

[Using labels with self-hosted runners](#)

Next

[Managing access to self-hosted runners using groups](#)

Legal

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