

This version of GitHub Enterprise was discontinued on 2023-03-15. No patch releases will be made, even for critical security issues. For better performance, improved security, and new features, [upgrade to the latest version of GitHub Enterprise](#). For help with the upgrade, [contact GitHub Enterprise support](#).

Setting up the tool cache on self-hosted runners without internet access

In this article

About the included setup actions and the runner tool cache

Prerequisites

Populating the tool cache for a self-hosted runner

To use the included `actions/setup` actions on self-hosted runners without internet access, you must first populate the runner's tool cache for your workflows.

Note: GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the [GitHub public roadmap](#).

About the included setup actions and the runner tool cache

GitHub Actions on GitHub Enterprise Server is designed to work in environments without full internet access. By default, workflows cannot use actions from GitHub.com and [GitHub Marketplace](#).

Most official GitHub-authored actions are automatically bundled with GitHub Enterprise Server. However, self-hosted runners without internet access require some configuration before they can use the included `actions/setup-LANGUAGE` actions, such as `setup-node`.

The `actions/setup-LANGUAGE` actions normally need internet access to download the required environment binaries into the runner's tool cache. Self-hosted runners without internet access can't download the binaries, so you must manually populate the tool cache on the runner.

You can populate the runner tool cache by running a GitHub Actions workflow on GitHub.com that uploads a GitHub-hosted runner's tool cache as an artifact, which you can then transfer and extract on your internet-disconnected self-hosted runner.

Note: You can only use a GitHub-hosted runner's tool cache for a self-hosted runner that has an identical operating system and architecture. For example, if you are using a `ubuntu-22.04` GitHub-hosted runner to generate a tool cache, your self-hosted runner must be a 64-bit Ubuntu 22.04 machine. For more information on GitHub-hosted runners, see "[About GitHub-hosted runners](#)."

Prerequisites

- Determine which development environments your self-hosted runners will need. The following example demonstrates how to populate a tool cache for the `setup-node` action, using Node.js versions 10 and 12.
- Access to a repository on GitHub.com that you can use to run a workflow.
- Access to your self-hosted runner's file system to populate the tool cache folder.

Populating the tool cache for a self-hosted runner

- 1 On GitHub.com, navigate to a repository that you can use to run a GitHub Actions workflow.
- 2 Create a new workflow file in the repository's `.github/workflows` folder that uploads an artifact containing the GitHub-hosted runner's tool cache.

The following example demonstrates a workflow that uploads the tool cache for an Ubuntu 22.04 environment, using the `setup-node` action with Node.js versions 10 and 12.

```
name: Upload Node.js 10 and 12 tool cache
on: push
jobs:
  upload_tool_cache:
    runs-on: ubuntu-22.04
    steps:
      - name: Clear any existing tool cache
        run: |
          mv "${{ runner.tool_cache }}" "${{ runner.tool_cache }}.old"
          mkdir -p "${{ runner.tool_cache }}"
      - name: Setup Node 10
        uses: actions/setup-node@v2
        with:
          node-version: 10.x
      - name: Setup Node 12
        uses: actions/setup-node@v2
        with:
          node-version: 12.x
      - name: Archive tool cache
        run: |
          cd "${{ runner.tool_cache }}"
          tar -czf tool_cache.tar.gz *
      - name: Upload tool cache artifact
        uses: actions/upload-artifact@v2
        with:
          path: ${{runner.tool_cache}}/tool_cache.tar.gz
```

- 3 Download the tool cache artifact from the workflow run. For instructions on downloading artifacts, see "[Downloading workflow artifacts](#)."
- 4 Transfer the tool cache artifact to your self hosted runner and extract it to the local tool cache directory. The default tool cache directory is `RUNNER_DIR/_work/_tool`. If the runner hasn't processed any jobs yet, you might need to create the `_work/_tool` directories.

After extracting the tool cache artifact uploaded in the above example, you should have a directory structure on your self-hosted runner that is similar to the following example:

```
RUNNER_DIR
├─ ...
```

```
└─ _work
  └─ ...
    └─ tool
      └─ node
        └─ 10.22.0
          └─ 0 0 └─ ...
        └─ 12.18.3
          └─ ...
```

Your self-hosted runner without internet access should now be able to use the `setup-node` action. If you are having problems, make sure that you have populated the correct tool cache for your workflows. For example, if you need to use the `setup-python` action, you will need to populate the tool cache with the Python environment you want to use.

Legal

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