Manually syncing actions from GitHub.com

In this article

For users that need access to actions from GitHub.com, you can sync specific actions to your enterprise.

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.

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.

The recommended approach of enabling access to actions from GitHub.com is to enable automatic access to all actions. You can do this by using GitHub Connect to integrate GitHub Enterprise Server with GitHub Enterprise Cloud. For more information, see "Enabling automatic access to GitHub.com actions using GitHub Connect."

However, if you want stricter control over which actions are allowed in your enterprise, you can follow this guide to use GitHub's open source `actions-sync` tool to sync individual action repositories from GitHub.com to your enterprise.

About the `actions-sync` tool

The `actions-sync` tool must be run on a machine that can access the GitHub.com API and your GitHub Enterprise Server instance's API. The machine doesn't need to be connected to both at the same time.

If your machine has access to both systems at the same time, you can do the sync with a single `actions-sync sync` command. If you can only access one system at a time, you can use the `actions-sync pull` and `push` commands.

The `actions-sync` tool can only download actions from GitHub.com that are stored in public repositories.

Note: The `actions-sync` tool is intended for use in systems where GitHub Connect is not enabled. If you run the tool on a system with GitHub Connect enabled, you may see the error `The repository <repo_name> has been retired and cannot be reused`. This indicates that a workflow has used that action directly on GitHub.com and the namespace is retired on your GitHub Enterprise Server instance. For more information, see "Enabling automatic access to GitHub.com actions using GitHub Connect."

Prerequisites

- Before using the `actions-sync` tool, you must ensure that all destination organizations already exist in your enterprise. The following example demonstrates how to sync actions to an organization named `synced-actions`. For more information, see "[Creating a new organization from scratch](#)."
- You must create a personal access token on your enterprise that can create and write to repositories in the destination organizations. For more information, see "[Managing your personal access tokens](#)."
- If you want to sync the bundled actions in the `actions` organization on your GitHub Enterprise Server instance, you must be an owner of the `actions` organization.

  > **Note:** By default, even site administrators are not owners of the bundled `actions` organization.

  Site administrators can use the `ghe-org-admin-promote` command in the administrative shell to promote a user to be an owner of the bundled `actions` organization. For more information, see "[Accessing the administrative shell (SSH)](#)" and "[Command-line utilities](#)."

  ```
  ghe-org-admin-promote -u USERNAME -o actions
  ```

Example: Using the `actions-sync` tool

This example demonstrates using the `actions-sync` tool to sync an individual action from GitHub.com to an enterprise instance.

> **Note:** This example uses the `actions-sync sync` command, which requires concurrent access to both the GitHub.com API and your enterprise instance's API from your machine. If you can only access one system at a time, you can use the `actions-sync pull` and `push` commands. For more information, see the [`actions-sync` README](#).

1. Download and extract the latest [`actions-sync` release](#) for your machine's operating system.
2. Create a directory to store cache files for the tool.
3. In your terminal, change to the `bin` directory within the release directory you just extracted. For example:

   ```
   cd PATH/TO/gh_DATETIME_OS_ARCHITECTURE/bin
   ```

4. Run the `actions-sync sync` command:

   ```
   ./actions-sync sync \
     --cache-dir "cache" \
     --destination-token "aabbccddeeffgg" \
     --destination-url "https://my-ghes-instance" \
     --repo-name "actions/stale:synced-actions/actions-stale"
   ```

   The above command uses the following arguments:

   - `--cache-dir`: The cache directory on the machine running the command.
   - `--destination-token`: A personal access token for the destination enterprise instance.
   - `--destination-url`: The URL of the destination enterprise instance.

- `--repo-name`: The action repository to sync. This takes the format of `owner/repository:destination_owner/destination_repository`.
  - The above example syncs the [`actions/stale`](#) repository to the `synced-actions/actions-stale` repository on the destination enterprise instance. You must create the organization named `synced-actions` in your enterprise before running the above command.
  - If you omit `:destination_owner/destination_repository`, the tool uses the original owner and repository name for your enterprise. Before running the command, you must create a new organization in your enterprise that matches the owner name of the action. Consider using a central organization to store the synced actions in your enterprise, as this means you will not need to create multiple new organizations if you sync actions from different owners.
  - You can sync multiple actions by replacing the `--repo-name` parameter with `--repo-name-list` or `--repo-name-list-file`. For more information, see the [`actions-sync` README](#).

5. After the action repository is created in your enterprise, people in your enterprise can use the destination repository to reference the action in their workflows. For the example action shown above:

```
uses: synced-actions/actions-stale@v1
```

For more information, see "[Workflow syntax for GitHub Actions](#)."