# Configuring a publishing source for your GitHub Pages site

**In this article**

You can configure your GitHub Pages site to publish when changes are pushed to a specific branch, or you can write a GitHub Actions workflow to publish your site.

**Who can use this feature**
People with admin or maintainer permissions for a repository can configure a publishing source for a GitHub Pages site.

> GitHub Pages is available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server.

## About publishing sources 🔗

You can publish your site when changes are pushed to a specific branch, or you can write a GitHub Actions workflow to publish your site. To use GitHub Actions as a publishing source for GitHub Pages, a site administrator must enable GitHub Actions for your GitHub Enterprise Server instance. For more information, see "[Enabling GitHub Actions for GitHub Enterprise Server](#)."

If you do not need any control over the build process for your site, we recommend that you publish your site when changes are pushed to a specific branch. You can specify which branch and folder to use as your publishing source. The source branch can be any branch in your repository, and the source folder can either be the root of the repository ( `/` ) on the source branch or a `/docs` folder on the source branch. Whenever changes are pushed to the source branch, the changes in the source folder will be published to your GitHub Pages site.

If you want to use a build process other than Jekyll or you do not want a dedicated branch to hold your compiled static files, we recommend that you write a GitHub Actions workflow to publish your site. GitHub Enterprise Server provides starter workflows for common publishing scenarios to help you write your workflow.

> **Warning**: If your site administrator has enabled Public Pages, GitHub Pages sites are publicly available on the internet, even if the repository for the site is private or internal. If you have sensitive data in your site's repository, you may want to remove the data before publishing. For more information, see "[Configuring GitHub Pages for your enterprise](#)" and "[About repositories](#)."

# Publishing from a branch 🔗

1. Make sure the branch you want to use as your publishing source already exists in your repository.

2. On GitHub Enterprise Server, navigate to your site's repository.

3. Under your repository name, click ⚙ **Settings**. If you cannot see the "Settings" tab, select the ··· dropdown menu, then click **Settings**.
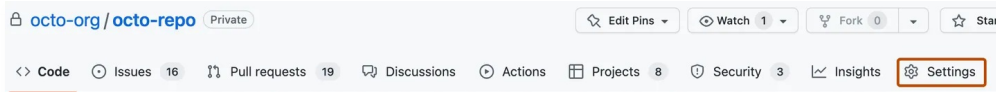


4. In the "Code and automation" section of the sidebar, click 🗔 **Pages**.

5. Under "Build and deployment", under "Source", select **Deploy from a branch**.

6. Under "Build and deployment", use the branch dropdown menu and select a publishing source.



7. Optionally, use the folder dropdown menu to select a folder for your publishing source.



8. Click **Save**.

## Troubleshooting publishing from a branch 🔗

> **Note:** If your repository contains symbolic links, you will need to publish your site using a GitHub Actions workflow. For more information about GitHub Actions, see "[GitHub Actions documentation](#)."

> **Notes:**
>
> - If you are publishing from a branch and your site has not published automatically, make sure someone with admin permissions and a verified email address has pushed to the publishing source.
>
> - Commits pushed by a GitHub Actions workflow that uses the `GITHUB_TOKEN` do not trigger a GitHub Pages build.

If you choose the `docs` folder on any branch as your publishing source, then later

remove the `/docs` folder from that branch in your repository, your site won't build and you'll get a page build error message for a missing `/docs` folder. For more information, see "[Troubleshooting Jekyll build errors for GitHub Pages sites](#)."

# Publishing with a custom GitHub Actions workflow 🔗

> **Note:** Publishing your GitHub Pages site with a custom GitHub Actions workflow is in beta and subject to change.

To configure your site to publish with GitHub Actions:

① On GitHub Enterprise Server, navigate to your site's repository.

② Under your repository name, click ⚙ **Settings**. If you cannot see the "Settings" tab, select the ··· dropdown menu, then click **Settings**.
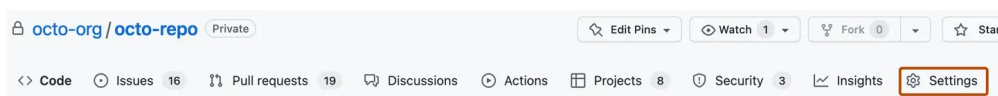


③ In the "Code and automation" section of the sidebar, click ▭ **Pages**.

④ Under "Build and deployment", under "Source", select **GitHub Actions**.

⑤ GitHub Enterprise Server will suggest several starter workflows. If you already have a workflow to publish your site, you can skip this step. Otherwise, choose one of the options to create a GitHub Actions workflow. For more information about creating your custom workflow, see "[Creating a custom GitHub Actions workflow to publish your site](#)."

GitHub Pages does not associate a specific workflow to the GitHub Pages settings. However, the GitHub Pages settings will link to the workflow run that most recently deployed your site.

# Creating a custom GitHub Actions workflow to publish your site 🔗

For more information about GitHub Actions, see "[GitHub Actions documentation](#)."

When you configure your site to publish with GitHub Actions, GitHub Enterprise Server will suggest starter workflows for common publishing scenarios. The general flow of a workflow is to:

① Trigger whenever there is a push to the default branch of the repository or whenever the workflow is run manually from the Actions tab.

② Use the `actions/checkout` action to check out the repository contents.

③ If required by your site, build any static site files.

④ Use the `actions/upload-pages-artifact` action to upload the static files as an artifact.

⑤ If the workflow was triggered by a push to the default branch, use the `actions/deploy-pages` action to deploy the artifact. This step is skipped if the workflow was triggered by a pull request.

The starter workflows use a deployment environment called `github-pages`. If your repository does not already include an environment called `github-pages`, the environment will be created automatically. We recommend that you add a deployment protection rule so that only the default branch can deploy to this environment. For more information, see "Using environments for deployment."

## Troubleshooting publishing with a custom GitHub Actions workflow 🔗

For information about how to troubleshoot your GitHub Actions workflow, see "About monitoring and troubleshooting."