

DocsAPI Reference ▾

Getting Started>Supported Frameworks>Incremental MigrationProduction ChecklistKnowledge BaseDocs llms-full.txt🔗

Access>AI>Build & Deploy>CDN>CLI>Collaboration>Compute>Multi-tenant>Observability>Platform▾Project Configuration▾

Git settings

Last updated November 25, 2025

Once you have [connected a Git repository](#), select the **Git** menu item from your [project settings page](#) to edit your project's Git settings. These settings include:

- Managing Git Large File Storage (LFS)
- Creating Deploy Hooks
- Ignoring the build step when a commit is pushed to the Git repository

Disconnect your Git repository

To disconnect your Git repository from your Vercel project:

- Choose a project from the [dashboard](#)
- Select the **Settings** tab and then select the **Git** menu item
- Under **Connected Git Repository**, select the **Disconnect** button.

Git Large File Storage (LFS)

If you have [LFS objects](#) in your repository, you can enable or disable support for them from the [project settings](#). When support is enabled, Vercel will pull the LFS objects that are used in your repository.

 You must [redeploy your project](#) after turning Git LFS on.

Deploy Hooks

Vercel supports **deploy hooks**, which are unique URLs that accept HTTP POST requests and trigger deployments. Check out [our Deploy Hooks documentation](#) to learn more.


Ignored Build Step

By default, Vercel creates a new [deployment](#) and build (unless the Build Step is [skipped](#)) for every commit pushed to your connected Git repository.

Each commit in Git is assigned a unique hash value commonly referred to as SHA. If the SHA of the commit was already deployed in the past, no new Deployment is created. In that case, the last Deployment matching that SHA is returned instead.

To ignore the build step:

- Choose a project from the [dashboard](#)
- Select the **Settings** tab and then select the **Git** menu item
- In the **Ignored Build Step** section, select the behavior you would like. This behavior provides a command that outputs a code, which tells Vercel whether to issue a new build or not. The command is executed within the [Root Directory](#) and can access all [System Environment Variables](#):
 - Automatic:** Each commit will issue a new build
 - Only build production:** When the `VERCEL_ENV` is production, a new build will be issued
 - Only build preview:** When the `VERCEL_ENV` is preview, a new build will be issued
 - Only build if there are changes:** A new build will be issued only when the Git diff contains changes
 - Only build if there are changes in a folder:** A new build will be issued only when the Git diff contains changes in a folder that you specify
 - Don't build anything:** A new build will never be issued
 - Run my Bash script:** [Run a Bash script](#) from a location that you specify
 - Run my Node script:** [Run a Node script](#) from a location that you specify
 - Custom:** You can enter any other command here, for example, only building an Nx app (`npm nx-ignore <project-name>`)
- When your deployment enters the `BUILDING` state, the command you've entered in the **Ignored Build Step** section will be run. The command will always exit with either code `1` or `0` :
 - If the command exits with code `1`, the build continues as normal
 - If the command exits with code `0`, the build is immediately aborted, and the deployment state is set to `CANCELED`

 Canceled builds are counted as full deployments as they execute a build command in the build step. This means that any canceled builds initiated using the ignore build step will still count towards your [deployment quotas](#) and [concurrent build slots](#). You may be able to optimize your deployment queue by [skipping builds](#) for projects within a monorepo that are unaffected by a change.

To learn about more advanced usage see the ["How do I use the Ignored Build Step field on Vercel?"](#) guide.

Ignore Build Step on redeploy

If you have set an ignore build step command or [script](#), you can also skip the build step when redeploying your app:

- From the Vercel dashboard, select your project
- Select the **Deployments** tab and find your deployment
- Click the ellipses (...) and from the context menu, select **Redeploy**
- Uncheck the **Use project's Ignore Build Step** checkbox

Verified Commits

Vercel allows you to require verified commits for deployments. This is only available for GitHub projects. Learn more about [verified commits on GitHub](#).

To enable verified commits:

- From the Vercel dashboard, select your project
- Select the **Settings** tab and then select the **Git** menu item
- Under **Require Verified Commits**, select the **Enable** checkbox

When enabled, Vercel will only create deployments for commits that have been verified by GitHub. For all other commits, the deployment will be automatically canceled.

Was this helpful?