

# How to Create a Pipeline for Git Submodule Repo

Deploying code from a repository that includes Git submodules can be a bit more complex than a standard repository. This article will guide you through the process of deploying a submodule Git repository on the Elestio CI/CD pipeline.

Check out our [example](#) repository to see how the process works.

The overall deployment process is similar to deploying a standard repository, as detailed in the [Elestio documentation](#). However, there are a few additional considerations when deploying a submodule repository with Elestio.

## Configuring .gitmodules

For a repository to handle submodules correctly, it must include a `.gitmodules` file. This file contains the configuration for each submodule, specifying its path, URL, and optionally the branch to be used.

If you want to deploy a submodule Git repository on a particular branch, you need to specify the branch in the `.gitmodules` file. Here's an example configuration:

```
[submodule "reactjs"]
  path = reactjs
  url = https://github.com/elestio-examples/reactjs.git
  branch = master
```

This configuration tells Git to check out the `master` branch of the reactjs` repository.`

## Webhook Considerations

When working with submodules, it's important to note that Git only sends webhooks for changes in the main repository, not for changes only in the submodules.

Therefore, if you make any changes in the submodule repository, you need to sync and commit those changes in the main repository to trigger the webhook. Alternatively, if you commit any changes in the main repository as well, the Elestio automation will automatically fetch the latest changes from the submodule repository.

**Note:** In your pipeline, installing dependencies and starting the server will only affect the main repository. The submodule's dependencies will not be installed, nor will its server be started by default. To manage this, you need to create a custom Dockerfile or Docker Compose configuration within the main repository. This setup will ensure that the pipeline deployment accounts for both the main repository and its submodules. Select either Docker or Docker Compose as the runtime environment based on your project's configuration.

---

Revision #16

Created 26 July 2024 13:17:10 by Amit

Updated 26 July 2024 14:22:59 by Amit