

# Deploy a CI/CD Pipeline via GitLab Self-Hosted

This guide walks you through deploying a CI/CD pipeline using a self-hosted GitLab instance on Elestio.

## Prerequisites

Before you begin, make sure you have:

- A running self-hosted GitLab instance (e.g. `https://gitlab.example.com`)
- A **Personal Access Token (PAT)** with the `api` scope (see note below)
- A repository hosted on your GitLab instance
- An active project in Elestio

⚠ **Important:** Only the `api` scope is required. Additional scopes like `read_repository` or `read_user` are not necessary when `api` is enabled.

You can follow our [documentation](#) to generate the PAT.

## Step 1: Select Deployment Method

1. Open your project in the Elestio dashboard
2. Navigate to **CI/CD** → **Create Pipeline**
3. Under **Deployment Method**, select **GitLab**



### Docker compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to...



### GitHub

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the...



### GitLab

GitLab Inc. is the open-core company that provides GitLab, the DevOps software that combines the ability to develop, secure,...

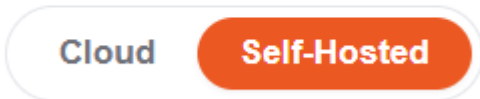
## Step 2: Choose Hosting Type

After selecting GitLab, choose your hosting type:

Option	Description
Cloud	Connect to GitLab.com via OAuth
Self-Hosted	Connect using your own GitLab instance and PAT

Select **Self-Hosted** to continue.

## 2. Hosting Type



Switching between Cloud and Self-Hosted will reset previously selected repositories and settings.

## Step 3: Connect Your Self-Hosted GitLab Instance

Enter your connection details:

### 1. GitLab Instance URL

- Example: `https://gitlab.example.com`
- Must include protocol (`http://` or `https://`)
- No trailing slash

### 2. Personal Access Token (PAT)

- Paste your token (`glpat-xxxxxxxxxxxx`)
- Field is masked for security


### 3. Click **Connect**




Import an existing project from a GitLab repository.

GitLab Instance URL

Personal Access Token (PAT)

 **Connect**

Expected result

-  Connected message:  
    **"Connected as [username] on [instance URL]."**
- Repository selection UI loads automatically

## If connection fails


Check:

- Instance is publicly reachable
- Token includes `api` scope
- Token has not expired

---

## Step 4: Select a Repository

Once connected, the **Import Git Repository** section appears.

1. **Git Account**
  - Your connected account is pre-selected
  - Use **Add Git Account** to connect another instance
2. **Git Scope** *(if available)*
  - Select a user, group, or organization
3. **Repository List**
  - Search or browse repositories
  - Private repos show a  icon
  - Click **"Import"** on your chosen repository


**Import Git Repository**

Git Account Git Scope

root root

Search repo

---

Reactjs  38 minutes ago **Import**

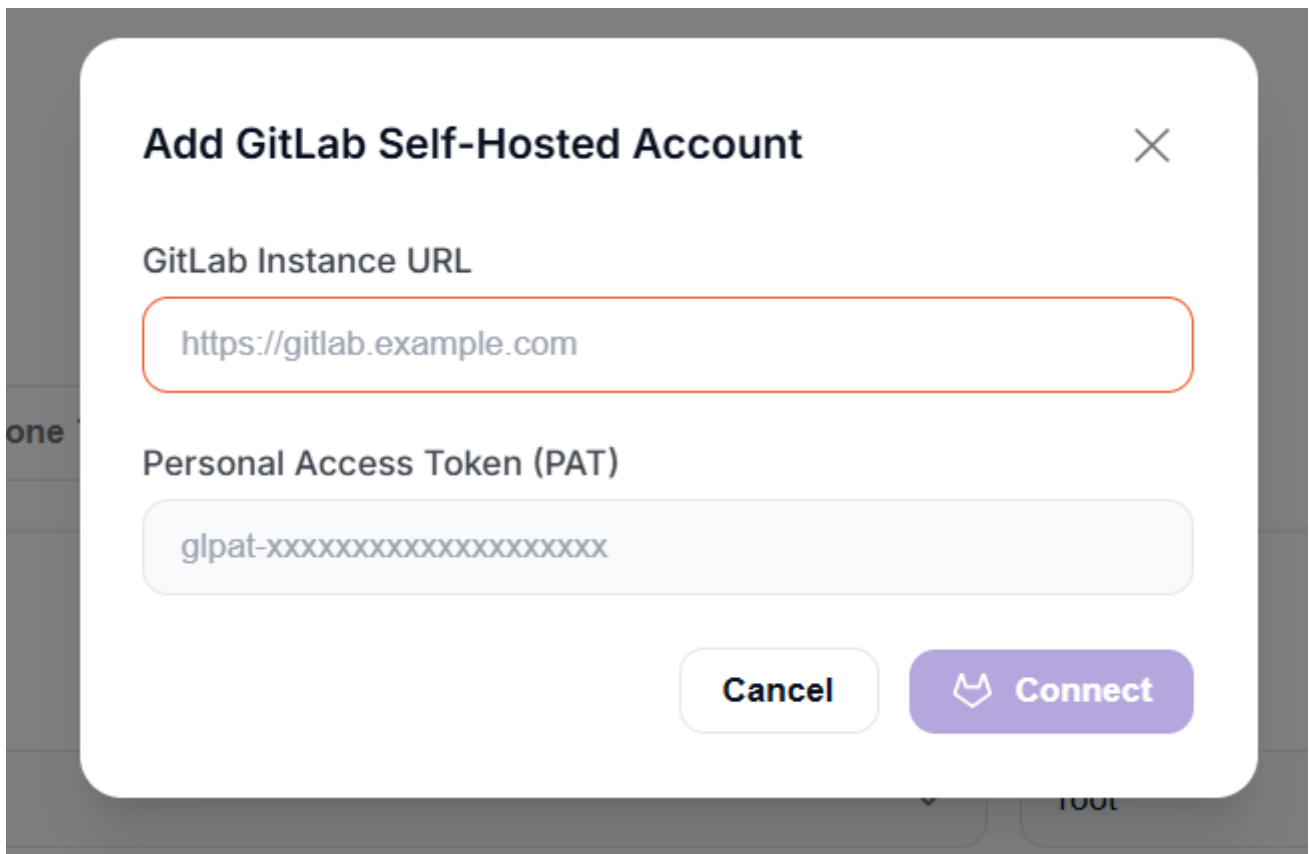
“ Now, click **"Next,"** then choose the cloud provider, region, and plan. After that, configure the pipeline settings by selecting the application type, runtime, and port configuration to proceed with the deployment.

## Adding Additional Self-Hosted Accounts

To connect another GitLab instance:

1. Open the **Git Account** dropdown
2. Click **Add Git Account**
3. Enter:
  - Instance URL
  - Personal Access Token ( `api` scope)
4. Click **Connect**

The new account becomes active immediately.



## Troubleshooting

Issue	Likely Cause	Fix
Connection fails	Invalid URL or token	Verify URL format and regenerate the PAT.
No repositories listed	Missing <code>api</code> scope	Recreate token with <code>api</code>
Branch not found	Empty repo or wrong branch	Ensure branch exists and has commits

Revision #1

Created 2026-05-15 15:03:52 UTC by Amit Shukla

Updated 2026-05-15 15:25:39 UTC by Amit Shukla