

Deploy examples & apps on Elestio using the "Import Third-Party Git Repository" feature

This tutorial will walk you through the process of deploying GitHub, GitLab, or Bitbucket public repositories in CI/CD using our **Import Third-Party Git Repository** feature. In our example, we'll use a Simple JavaScript web application public repository, but you can deploy it, in the same way, using any public repository from [GitHub](#), [GitLab](#), or [Bitbucket](#). ☐

What is the Import Third-Party Git Repository?

A feature of Elestio's CI/CD **Import Third-Party Git Repository** lets you deploy any type of public git repository from [GitHub](#), [GitLab](#), or [Bitbucket](#) in the Elestio cloud.

What is the difference between Import Git Repository And Import Third-Party Git Repository?

Only public or private repositories that are already in your own git account can be deployed using the **Import Git Repository** feature. Depending on the deployment method you select—for example if you choose GITHUB—you can only deploy repositories that are already in your github account. On the other hand, you can deploy any only public repository from your or anyone else's account from github, gitlab, or bitbucket using the **Import Third-Party Git Repository** feature.

“ In these **Import Third-Party Git Repository** features, Elestio will create a repository from your entered repository URL in your git account and deploy it to the cloud.

You probably heard about Kubernetes (and all its complexity) or various options to deploy your apps like Heroku, Render Fly, or Railways. They all have something in common, those products are building your own source code on every commit from your GIT repository.

Elestio is doing the same ... **but different!** Instead of deploying your app to a shared cluster, we deploy to dedicated VMs.

To learn more about the elestio CI/CD, go [here](#).

If you're new, sign up for [Elestio](#), otherwise, login to your existing account.

Deploy Any Public repository Apps to the cloud using the CI/CD **Import Third-Party Git Repository** feature.

Step 1:

Go to CI/CD from the left sidebar.

Step 2:

Now, select the deployment source.

Create CI/CD pipeline

1 Source ————— 2 Target ————— 3 Configuration

1. Deployment Method



Select the deployment method of your Service between Github, Gitlab and Docker.

When using the Github/Gitlab deployment method, each time a change is pushed to your repository, a new deployment of your service will occur.



Github

Git-Hub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management...



Gitlab

GitLab Inc. is the open-core company that provides GitLab, the DevOps software that combines the ability to develop, secure, and operate software in a single application



Docker compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services

In this tutorial, I'm deploying using GITHUB, but you can also use GITLAB if you want to deploy your repository there.

Step 3:

Click the Import a Third-Party Git Repository

Import a Third-Party Git Repository



Enter the URL of a Git repository to deploy it:

`https://some-provider.com/some-organization/some-project`

Back

Continue

The URL of the public repository you want to deploy should now be entered here.

Now click the **Continue** button to proceed.

If you have already authenticated your GITHUB or GITLAB account in CI/CD for repository access, you can fill up the below details directly. Otherwise, you must first authenticate your GIT account with elestio CI/CD for repository creation into your account.

We require GIT authentication in order to create these example template repositories in your GIT account.

[← Back](#)

Git Account

amitshuklabag

Create Git Repository

☐ Create private Git Repository

To ensure you can easily update your project after deploying it, a Git repository must be created. Every push to that Git repository will be deployed automatically.

Git Scope

elestio-examples

Repository Name

static

Next

Here you can check the checkbox **Create private Git Repository** if you want to make these repo private otherwise leave it unchecked and click the **Next button** for further steps.

Step 4:

Choose Deployment Targets

1. Choose Deployment Targets



Where do you want this service to be deployed?

Deploy on a new VM

Deploy on existing VM

Deployment mode

Single Node

Cluster

Elestio offers two types of deployment targets "**Deploy on a new VM**" and "**Deploy on an existing VM**".

You are allowed to set up n pipelines on each elestio CI/CD target/VM. According to the project configuration you select and the project you're deploying, the number of pipelines varies.

If you want to deploy these projects as a pipeline on a new Target/VM or don't have any installed targets, choose "Deploy on a new VM." If you already have any installed or previously configured CI/CD targets/VMs, choose "Deploy on an existing VM," and then choose the existing target from the targets dropdown.

Follow the steps below only if you select "**Deploy on a new VM**," otherwise click the next button to proceed.

CI/CD Pipelines by Elestio are available with our 5 cloud partners (AWS Lightsail, Digital Ocean, Vultr, Linode & Hetzner) in 85 locations over 27 countries but also on any cloud (AWS, Azure, Google, Oracle, ...) and on-premise with [BYOVM](#).


- Select Service Cloud Provider



- Select Service Cloud Region

Europe North America


fsn1

 Germany - Falkenstein

hel1





 Finlande - Helsinki

nbg1





 Germany - Nuremberg

- Select Service Plan





SMALL-1C-2G

 1 CPU  2 GB RAM  20 GB Storage  20 TB Bandwidth included





SMALL-2C-2G

 2 CPU  2 GB RAM  40 GB Storage  20 TB Bandwidth included

MEDIUM-2C-4G

 2 CPU  4 GB RAM  40 GB Storage  20 TB Bandwidth included

MEDIUM-3C-4G

 3 CPU  4 GB RAM  80 GB Storage  20 TB Bandwidth included

- Now Customize the target name and project (where the CICD Target will be created).

5. Provide a name for your new CICD Target instance



The CICD Target name cannot be changed afterwards. This is the name of the instance that will contain one or more pipelines.

Name*

cicd-xttin

6. Select the target project where the CICD Target will be created


elestio-services

“ If you want to deploy it with a different name and a different project, you can customize it. By default, we configure it with a dynamic target name and the current project.

Step 5:

Configure your Project

Configure your static Project

 Define your project build and run

Project Name	Branch
<input type="text" value="static"/>	<input type="text" value="main"/>
Runtime	
<input type="text" value="Static Website"/>	
Framework	Root Directory
<input type="text" value="No Framework"/>	<input type="text" value="/"/>

You can configure the project details by filling up the project name, branch, run time, version, framework, and root directory.

Build and Output Setting

Install command	Run command
<input type="text" value="`yarn install`, `pnpm install`, or `npm install`"/>	<input type="text"/>
Build command	Build Output dir
<input type="text"/>	<input type="text" value="/"/>

You can configure your project install, run, and build commands in the Build and output setting.

Life cycle scripts (optional)

Pre install command	Post install command
<input type="text" value="./scripts/preInstall.sh"/>	<input type="text" value="./scripts/postInstall.sh"/>
Pre Backup command	Post Backup command
<input type="text" value="./scripts/preBackup.sh"/>	<input type="text" value="./scripts/postBackup.sh"/>
Pre Restore command	Post Restore command
<input type="text" value="./scripts/preRestore.sh"/>	<input type="text" value="./scripts/postRestore.sh"/>

The configuration of life cycle scripts is always optional; they should only be used if you want to execute a specific command before and after building your project. Otherwise, leave them empty.

Environment variables



Use Environment variables to store configuration values. API keys and secrets. You can access them in your service like regular environment variables.

.env

1 ENV=production

2

You can list all of your project's API keys and secrets here if they were saved in ENV

Exposed Ports



Configure the ports which should be exposed. When the ports are not exposed publicly, they are only accessible to other Services of the same App via the internal network.

Interface

172.17.0.1

Host Protocol

HTTP



Host Port

3000

Container Port

80



+ Add Another

Reverse proxy configuration



Configure the ports which should be exposed. When the ports are not exposed publicly, they are only accessible to other Services of the same App via the internal network.

Listen

Target

Protocol

HTTPS

Port

443



Protocol

HTTP

IP

172.17.0.1

Port

3000

Path

/



☐ Require Basic Auth

+ Add Another

The final step is to configure the exposed port and reverse proxy settings. You can specify the port on which your project will run here.

“ If your project includes **elestio.yml**, Elestio will auto-fill all of these fields. As in this tutorial, we're using our Simple Javascript web application **elestio** example repository, so you can see in the above images that all of our fields are auto-filled.

Refer to these links to learn how to create our own [elestio.yml](#) for the project.

A sample **elestio.yml** for our deploying Simple Javascript web application is shown below. check it out on [github](#)

```
config:
  runTime: "static"
  version: ""
  framework: ""
  buildCommand: ""
  buildDir: ""
  runCommand: ""
ports:
  - protocol: "HTTPS"
    targetProtocol: "HTTP"
    listeningPort: "443"
    targetPort: "3000"
    targetIP: "172.17.0.1"
    public: true
    path: "/"
    isAuth: false
    login: ""
    password: ""
exposedPorts:
  - protocol: "HTTP"
    hostPort: "3000"
    containerPort: "80"
    interface: "172.17.0.1"
```

Step 6:

Click the **Create CI/CD pipeline** button to deploy your pipeline.



Provider

Hetzner Cloud

Region

Europe, Germany
Falkenstein

Estimated Monthly Price*

\$10

*Estimated monthly price is based on
730 hours of usage.

Create CI/CD pipeline

In a couple of moments, your application was successfully deployed on elestio .



static (cicd-xttin)

CI/CD

Deployed

Success

Open terminal

Delete Pipeline

Details

Tools

Backups

Build & Deploy

History

Domain Management

Termination protection

Disabled. Pipeline can be terminated.

Protection deactivated



Pipeline Details

CNAME: static-u76.vm.appdrag.net
Deployment method: Github
Runtime: Static website
Framework: No Framework
Branch name: main
Repo url: <https://github.com/amitshuklabag/static>

View Instructions

View Instructions from README

View Instructions

Manage Stack

View running logs

Restart Stack

Stop Stack

Move Pipeline

Move pipeline to new and existing target

Move Pipeline

Clone Pipeline

Clone pipeline to new and existing target

Clone Pipeline



Welcome to Elestio

Deploy your apps quickly with the easiest CI/CD system

This Host static-u76.vm.appdrag.net

Your IP 117.207.213.159

Your Location [IN, Indore](https://IN,Indore)

Latency to server [142 ms](https://142ms)

Deploy on Elestio

You can now view your deployed URL and access your application by going to desired application pipeline details.

Please let us know by contacting our support [email](#) or [ticketing](#) system if you give it a shot and encounter any problems or if anything goes wrong.

Join us on [discord](#) to know more.

Revision #21

Created 15 November 2022 11:47:04 by Amit Shukla

Updated 16 November 2022 18:09:47 by Amit Shukla