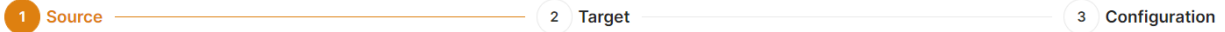


Overview

Elestio is the simplest and fastest way to deploy your source code from github/gitlab/docker registry to production.

Create CI/CD pipeline



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

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



Gitlab

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



Docker compose

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

2. Git repository

Import Git Repository

Clone Template

Import Git Repository

Git Account

amitshuklabag

Git Scope

elestio-examples

Search repo

airbyte

19 days ago

Import

angular

3 months ago

Import

astro

3 months ago

Import

bookstack

a month ago

Import

brunch

3 months ago

Import

Import Third-Party Git Repository →

In 3 steps you can deploy any kind of project, frontend, backend, API, DB, you name it.

1. Select your source code in Github / Gitlab or from a Docker registry (private registries are supported)
2. Select the target where you want to deploy
3. Configure in the UI env vars, volumes, life cycle events, reverse proxy with SSL

Wait a few minutes and **your app is live!**

We support **ANY** Linux tech stack and we provide [samples for the most popular stacks & frameworks \(54 Apps & Frameworks\)](#)

Select a template

Search templates by name



Static

A Static Java Script Web Application



Angular.js

A simple Angular app, created with the Angular CLI



React.js

A client-side React app created with create-react-app



Next.js

A Next.js App created using npx create-next-app



Vue.js

Sample Template of Vue.js, created with the vue CLI



Django

A Simple Django web App template



Node.js-Express

An App built using backend Node.js and Express Framework



Java

A Basic Application for Spring Boot framework



Svelte

A basic Svelte app using the simple template



Dotnet

A Simple web-app created using the command dotnet new



Go

A basic application created using go mod init



Ruby

A server-side web app created using Ruby on Rails



Remix

A new Remix app --the result of running 'npx create-remix'



Nuxt.js

A sample Nuxt.js app, created with create-nuxt-app



Gatsby.js

A Gatsby starter app created with gatsby new



Vuepress

A simple web application created using command create vuepress-site



TinaCMS

A TinaCMS application created with a Bare Bones Starter



Pelican

A simple Pelican App created using Python



MkDocs

A MkDocs starter app created using command mkdocs new



Astro

A simple web application created using command create astro@latest



Gridsome

A Gridsome App created on the basis of Jamstack



Brunch

A simple Brunch App created using command brunch new



Docusaurus

A Docusaurus App created using command create-docusaurus@latest



Jigsaw

A sample starter app created using Jigsaw



Umi.js

A Client-side Umi.js App created with create umi



Jekyll

A server-side Jekyll app created using Ruby Gem



Quasar

A sample Quasar App based on Vue.js



EmberApp

A sample Ember App created using the command ember new



Eleventy

A simple Eleventy App created using @11ty/eleventy



Hugo

A sample App created using the Hugo website Engine



Expo

A sample starter App created using Expo



Strapi-SQLite

A Strapi App with sqlite Database



Strapi-Postgres

A Strapi App with Postgres Database



Directus-Postgres

A Directus App with Postgres Database



Cloudgate

A simple Cloudgate App created using create-cloudgate-app



Directus

Docker Compose Example of Directus App



Bookstack

Docker Compose Example of Bookstack App



MySQL + PhpMyAdmin

MySQL Docker Compose Example



Metabase

Metabase Docker Compose Example

Deploy to 132 datacenters in 29 countries + BringYourOwnVM

CI/CD Pipelines by Elestio are available with our 5 cloud partners (AWS Lightsail, Digital Ocean, Vultr, Linode & Hetzner) in 137 locations over 29 countries but also on any cloud (AWS, Azure, Google, Oracle, ...) and on-premise with [BYOVM](#).

Vertical & Horizontal scaling

Scale from 1 CPU/2GB Ram up to 48 CPU/256GB Ram, you can also scale horizontally with a [load balancer](#) and clustered infrastructure. All are fully managed on any cloud.

Full isolation or Shared infrastructure

You can decide to deploy CI/CD pipelines in full isolation mode (1 pipeline on 1 VM) or deploy several pipelines to the same CI/CD target (Multiple pipelines on 1 VM). It's up to you if you are not sure we recommend going with full isolation to avoid one pipeline affecting the performance of others.

What is included

- Automated source code build & deployment from Github/Gitlab & docker registries
- Live & history logs, quickly edit your config, and redeploy when needed
- Deploy multiple pipelines on a single target
- You can move a pipeline from a target to another one in a click (nearly instant migration)
- Reverse proxy with managed SSL, additional custom domains are supported
- Tools: ssh, web terminal, VS Code, File explorer all in the context of your pipeline
- Automated Backups/Restore to S3 Bucket

Pricing

- We don't charge for build minutes
- We don't charge per user
- We don't charge per pipeline
- We don't charge per project
- We don't charge per X, Y, or Z BS metrics

You pay only for the CICD Targets that you launch, the price is per hour and details are available on the [Elestio pricing page](#)

Example: If you need to deploy 2 CI/CD pipelines, 1 for the frontend of your app (React.js) and 1 for the backend (Node.js) you can deploy both of them to the same CI/CD target instance. Now if you need to deploy 5 projects to the same CI/CD target the price won't change. Of course, at some point, you might have to upgrade your instance to get more RAM/CPU to satisfy all your apps.

Philosophy & Vision

Low overhead: we regularly see orchestration, networking, monitoring & logging systems using 80% of the hardware resources. Leaving not much for the apps to perform well or pushing customers to over-provision everything to get decent performances at a huge cost. We have created in-house a super low overhead management system. This allows us to offer a very efficient way to deploy and manage a lot of projects at scale.

Portability: your pipelines can be moved from one CI/CD target to another with a click. You can also download a backup and run it locally. There is no lock-in. We want customers to stay with us because of the quality of service, support, and time-saving compared to doing all the DevOps tasks by yourself.

KISS: an acronym for keeping it simple and stupid. The principle states that most systems work best if they are kept simple rather than made complicated; therefore, simplicity should be a key goal in design, and unnecessary complexity should be avoided.

Revision #53

Created 29 June 2022 08:36:29 by Joseph Benguira

Updated 27 April 2025 14:58:45 by Joseph Benguira