CI/CD Pipelines

Elestio CI/CD Pipelines let you deploy your own code from GitHub, GitLab, or a Docker registry onto a dedicated VM managed by Elestio.

Architecture

A CI/CD deployment consists of two parts:

1. **CI/CD Target VM:** A dedicated Elestio VM that hosts your pipelines
2. **Pipeline: A** build/run definition linked to your repository

---

Step 1: Deploy a CI/CD Target VM

```
elestio deploy cicd --project 112 --name my-cicd-target


# With provider/region/size options
elestio deploy cicd \
  --project 112 \
  --name prod-cicd \
  --provider hetzner \
  --region fsn1 \
  --size LARGE-4C-8G
```

Wait for it to be ready:

```
elestio wait <vmID>
```

Step 2a: Automated Pipeline (Recommended)

The CLI handles everything automatically: SSH key setup, Dockerfile generation, Docker image build, container start, and HTTP health check.

```
elestio cicd create --auto \
  --target <vmID> \
  --name my-app \
```

```
  --repo owner/repo \
  --mode github \
  --auth-id <githubAuthID>
```

Available modes:

Mode	Description
`github`	GitHub Static SPA (Vite, React, etc.)
`github-fullstack`	GitHub Full Stack (Node.js + frontend)
`gitlab`	GitLab Static SPA
`gitlab-fullstack`	GitLab Full Stack
`docker`	Custom Docker Compose (no Git)

Optional flags:

```
--branch main            # Source branch (default: main)
--build-cmd "npm run build"
--run-cmd "npm start"
--install-cmd "npm install"
--build-dir dist/
--framework react
--node-version 20
```

After completion, your app is live at: `https://<name>-u<userID>.vm.elestio.app/`

Step 2b: Manual Pipeline (from template)

For custom Docker deployments or advanced configurations.

```
# 1. Add SSH key so CLI can connect to the target VM
elestio ssh-keys add <vmID> --name "ci-key" --key "ssh-ed25519 AAAA..."

# 2. Generate a pipeline template
elestio cicd template docker > pipeline.json
# Other templates:
# elestio cicd template github
# elestio cicd template github-fullstack
# elestio cicd template gitlab

# 3. Edit pipeline.json with your config, then create the pipeline
```

```
elestio cicd create pipeline.json

# 4. SSH into the VM and configure your app
ssh root@<ipv4>
cd /opt/app/<pipeline-name>
# edit docker-compose.yml, add code, then:
docker-compose up -d
```

Managing Pipelines

```
# List all CI/CD target VMs
elestio cicd targets

# List pipelines on a target
elestio cicd pipelines <vmID>

# Get details of a specific pipeline
elestio cicd pipeline-info <vmID> <pipelineID>

# Restart a pipeline
elestio cicd pipeline-restart <vmID> <pipelineID>

# Stop a pipeline
elestio cicd pipeline-stop <vmID> <pipelineID>

# View build/runtime logs
elestio cicd pipeline-logs <vmID> <pipelineID>

# View deployment history
elestio cicd pipeline-history <vmID> <pipelineID>

# Delete a pipeline
elestio cicd pipeline-delete <vmID> <pipelineID> --force
```

Custom Domains for Pipelines

```
# List domains for a pipeline
elestio cicd domains <vmID> <pipelineID>
```

```
# Add a custom domain
elestio cicd domain-add <vmID> \
  --pipeline <pipelineID> \
  --domain myapp.example.com

# Remove a domain
elestio cicd domain-remove <vmID> \
  --pipeline <pipelineID> \
  --domain myapp.example.com
```

> Point your DNS A record to the CI/CD target VM's IPv4 before adding the domain.

Docker Registries

Connect private registries for pulling custom images:

```
# List connected registries
elestio cicd registries

# Add Docker Hub registry
elestio cicd registry-add \
--name X \
--username U \
--password P \
--url REPO

# Add GitLab.com registry
elestio cicd registry-add \
--name X \
--username U \
--password P \
--url REPO \
--registry-type registry.gitlab.com \
--repo-id ID

# Add self-hosted GitLab registry
elestio cicd registry-add \
--name X \
```

```
--username U \
--password P \
--url REPO \
--registry-type gitlab-self-hosted \
--repo-id ID \
--gitlab-url gitlab.company.com

# Add GitHub Container Registry
elestio cicd registry-add \
--name X \
--username U \
--password P \
--url REPO \
--registry-type ghcr.io
```

`registry-add` options:

Option	Description
`--name`	Unique identity nickname for the registry credential
`--username`	Registry username
`--password`	Registry password or access token
`--url`	Repository path (e.g. `myuser/myrepo`) **not** the registry host
`--registry-type`	Registry host: `docker.io` (default), `registry.gitlab.com`, `gitlab-self-hosted`, `ghcr.io`
`--repo-id`	GitLab project/repo ID — required for `registry.gitlab.com` and `gitlab-self-hosted`
`--gitlab-url`	Self-hosted GitLab hostname (e.g. `gitlab.company.com`) — required for `gitlab-self-hosted`

Troubleshooting Pipelines

Pipeline not starting

```
# Check logs
elestio cicd pipeline-logs <vmID> <pipelineID>
```

App not reachable

```
# SSH into the VM
ssh root@<ipv4>

# Check docker containers
cd /opt/app/<pipeline-name>
docker-compose ps
docker-compose logs
```

> Bind your app to `172.17.0.1:PORT` an internal Docker network, not `0.0.0.0:PORT`.

Trigger a manual rebuild

```
elestio cicd pipeline-restart <vmID> <pipelineID>
```

Revision #2
Created 2026-02-20 13:43:56 UTC by Amit Shukla
Updated 2026-04-03 11:02:42 UTC by Amit Shukla