Elestio CLI & Skill

Elestio CLI, a full command-line tool for managing the Elestio platform, and the Elestio Skill, which teaches AI agents how to use it

Overview

What is the Elestio CLI?

The **Elestio CLI** is the official command-line interface for the Elestio platform, installable as a global npm package:

```
npm install -g elestio
```

It exposes every platform capability as a command: deploy services, manage firewall rules, configure SSL certificates, create CI/CD pipelines, handle backups, and monitor billing, all from your terminal without ever opening a browser.

Key capabilities

Category	What you can do
Catalog deployment	Deploy any of 400+ open-source templates in one command
CI/CD	Create automated pipelines from GitHub/GitLab repos
Service management	Power on/off, reboot, resize, lock/unlock services
Firewall	Add, update, and remove firewall rules
SSL & Domains	Attach custom domains with auto-provisioned certificates
Backups	Local, remote, and S3-based backup automation
Volumes & Snapshots	Create and manage persistent storage (provider-dependent)
Billing	Query costs per project or per service

What is the Elestio Agent skill?

The **Elestio Agent Skill** is an open-format instruction file that teaches AI coding agents (Claude Code, Cursor, Codex, and OpenCode) how to use the Elestio CLI on your behalf.

Once installed, your AI agent gains the ability to:

- Decide which template to deploy based on your natural language request
- Walk you through provider, region, and size selection interactively
- Execute the correct CLI commands end-to-end
- Verify deployment success and return your service credentials

The skill follows the Agent Skills open format, making it portable across all supported AI tools.

How they work together

```
You (natural language)
          ↓
AI Agent reads SKILL.md → knows CLI commands, interactive procedures, and provider rules
          ↓
AI Agent runs: elestio deploy postgresql --project 112 --name my-db
          ↓
Elestio CLI → calls Elestio REST API → provisions VM on your chosen cloud
          ↓
Service is live at https://my-db-uxxx.vm.elestio.app
```

Supported AI Agents

Agent	Skills Directory
Claude Code	`~/.claude/skills/elestio/`
OpenAI Codex	`~/.codex/skills/elestio/`
Cursor	`~/.cursor/rules/`
OpenCode	`~/.config/opencode/skills/elestio/`

Quick links

- **Dashboard:** https://dash.elest.io
- **Elestio CLI (npm):** https://www.npmjs.com/package/elestio
- **Agent Skill (GitHub):** https://github.com/elestio/elestio-skill
- **API Docs:** https://api-doc.elest.io
- **Templates (400+):** https://elest.io/fully-managed-services

Installation and Setup

Requirements

- **Node.js** >= 18 (check with `node --version`)
- An active **Elestio account** with a verified email, payment method, and API token
- **Git** (required only for the Agent Skill installer)

Step 1 – Create an Elestio account (manual)

You need to do these steps yourself, manually, using a web browser. They cannot be done automatically or through the command line (CLI).

1. Sign up at Dashboard
2. Verify your email address
3. Add a credit card at https://dash.elest.io/account/payment-options
4. Wait for account approval (usually instant; can take up to 30 minutes to 48 hours for new accounts)

Step 2 – Generate an API token

1. Go to https://dash.elest.io/account/security
2. Click **Manage API Tokens → Create Token**
3. Copy the token value (it is shown only once)

Step 3 – Install the Elestio CLI

```
npm install -g elestio
```

Verify the installation:

```
elestio --version
```

Step 4 – Configure credentials

```
elestio login --email "you@example.com" --token "your_api_token"
```

Test that authentication works:

```
elestio auth test
# Expected: [SUCCESS] Authenticated as you@example.com
```

Step 5 – (Optional) Set a default project

Avoid passing `--project` on every command by setting a default:

```
# List your projects to find the ID
elestio projects

# Set the default
elestio config --set-default-project 112
```

Installing the Agent Skill

Automatic (recommended)

```
curl -fsSL https://raw.githubusercontent.com/elestio/elestio-skill/main/scripts/install.sh | bash
```

The installer will:

1. Detect which AI tools you have installed (Claude Code, Codex, Cursor, OpenCode)
2. Install the Elestio CLI globally via npm
3. Copy `SKILL.md` to each detected agent's skills directory

Via skills.sh

```
npx skills add elestio/elestio-skill
```

Manual installation

```
# 1. Install the CLI
npm install -g elestio
```

```
# 2. Clone the skill repo
git clone https://github.com/elestio/elestio-skill.git /tmp/elestio-skill

# 3. Copy SKILL.md to your agent
# Claude Code
mkdir -p ~/.claude/skills/elestio
cp /tmp/elestio-skill/SKILL.md ~/.claude/skills/elestio/

# Cursor
mkdir -p ~/.cursor/rules
cp /tmp/elestio-skill/SKILL.md ~/.cursor/rules/elestio.md
```

Update the skill

Re-run the installer at any time to get the latest version:

```
curl -fsSL https://raw.githubusercontent.com/elestio/elestio-skill/main/scripts/install.sh | bash
```

Verify everything is working

```
# Authentication
elestio auth test

# List your projects
elestio projects

# Browse the template catalog (no auth required)
elestio templates search postgresql
```

Using the Elestio CLI – Command Reference

Authentication & Configuration

```
elestio login --email X --token Y       # Store credentials locally
elestio auth test                        # Verify current credentials
elestio whoami                           # Show current user info
elestio config                           # Show current configuration
elestio config --set-default-project X  # Set a default project ID
```

Catalog (no authentication required)

```
elestio templates                        # List all 400+ templates
elestio templates search <query>         # Search by name (e.g. "postgresql")
elestio templates info <name>            # Detailed template information
elestio categories                       # List all software categories
elestio sizes                            # All provider/region/size combinations
elestio sizes --provider netcup          # Filter sizes by provider
elestio sizes --provider hetzner
```

Projects

```
elestio projects                         # List all projects
elestio projects create <name>           # Create a new project
elestio projects delete <id> --force     # Delete a project
elestio projects members <id>            # List project members
elestio projects add-member <id> <email>
elestio projects remove-member <id> <memberId>
```

Services

```
elestio services                          # List all services (all projects)
elestio services --project 123            # Filter by project
elestio service <vmID>                    # Get service details
elestio deploy <template> --project X --name Y   # Deploy from catalog
elestio deploy cicd --project X --name Y         # Deploy a CI/CD target VM
elestio delete-service <vmID> --force     # Delete a service
elestio move-service <vmID> <projectId>   # Move service to another project
elestio wait <vmID>                       # Wait until deployment completes
```

Deploy with all options:

```
elestio deploy postgresql \
  --project 112 \
  --name prod-postgres \
  --provider hetzner \
  --region fsn1 \
  --size LARGE-4C-8G \
  --support level2
```

Default values (when flags are omitted):

Setting	Default
Provider	netcup
Region	nbg (Germany)
Size	MEDIUM-2C-4G
Support	level1

Power Management

```
elestio reboot <vmID>                     # Graceful reboot
elestio reset <vmID>                      # Hard reset (power cycle)
elestio shutdown <vmID>                   # Graceful shutdown (ACPI)
elestio poweroff <vmID>                   # Force power off
elestio poweron <vmID>                    # Power on
elestio restart-stack <vmID>              # Restart Docker containers only (fastest)
elestio lock <vmID>                       # Enable termination protection
```

```
elestio unlock <vmID>                    # Disable termination protection
elestio resize <vmID> --size LARGE-4C-8G  # Resize VM (upgrade or downgrade)
```

Firewall

```
elestio firewall get <vmID>              # List current firewall rules

elestio firewall enable <vmID> --rules '[
  {"type":"INPUT","port":"22","protocol":"tcp","targets":["0.0.0.0/0"]},
  {"type":"INPUT","port":"443","protocol":"tcp","targets":["0.0.0.0/0"]}
]'

elestio firewall update <vmID> --rules '[...]'   # Replace all rules
elestio firewall disable <vmID>                  # Disable firewall
```

SSL & Custom Domains

```
elestio ssl list <vmID>                  # List domains for a service
elestio ssl add <vmID> myapp.example.com # Add domain (auto-provisions SSL)
elestio ssl remove <vmID> myapp.example.com
```

SSH Keys

```
elestio ssh-keys list <vmID>
elestio ssh-keys add <vmID> --name "deploy-key" --key "ssh-ed25519 AAAA..."
elestio ssh-keys remove <vmID> --name "deploy-key"
```

> **Note:** Provide only the key type and key data. Do **not** include the trailing comment/email.

Auto-Updates

```
# OS-level updates
elestio updates system-enable <vmID> --day 0 --hour 5 --security-only
elestio updates system-disable <vmID>
elestio updates system-now <vmID>        # Run OS update immediately
```

```
# Application-level updates
elestio updates app-enable <vmID> --day 0 --hour 3
elestio updates app-disable <vmID>
elestio updates app-now <vmID>           # Run app update immediately

# Change application version
elestio change-version <vmID> 15         # e.g., PostgreSQL 15
```

Alerts

```
elestio alerts get <vmID>
elestio alerts enable <vmID> --rules '{...}' --cycle 60
elestio alerts disable <vmID>
```

Access & Credentials

```
elestio credentials <vmID>               # App URL + admin login
elestio ssh <vmID>                       # SSH web terminal URL
elestio ssh <vmID> --direct              # Direct SSH command (copy-paste)
elestio vscode <vmID>                    # VSCode in browser URL
elestio files <vmID>                     # File manager URL
```

Billing

```
elestio billing                          # Total cost across all projects
elestio billing project <id>             # Per-service cost breakdown
```

ID Reference

Term	Where to find it	Used in
`vmID`	`elestio services` → `.vmID`	Most commands
`serverID`	`elestio services` → `.id`	Backup endpoints, notes
`projectID`	`elestio projects` → `.projectID`	`--project` flag
`templateID`	`elestio templates` → `.id`	`elestio deploy`

Term	Where to find it	Used in
`pipelineID`	`elestio cicd pipelines` → `.pipelineID`	CI/CD commands
`volumeID`	`elestio volumes` → `.id`	Volume commands

> **Common mistake:** `vmID` and `serverID` are different numbers for the same service. Most commands use `vmID`.

Using the Agent Skill with AI Agents

How it works

The Elestio Agent Skill is a `SKILL.md` file placed in your AI agent's skills directory. When you ask your agent to deploy or manage infrastructure, it reads `SKILL.md` to understand:

- Which CLI commands to run
- The correct interactive deployment procedure (what to ask you before deploying)
- Provider capabilities and limitations
- Error-handling strategies

The agent then runs the appropriate `elestio` commands on your behalf.

Trigger phrases

Your agent will activate the Elestio skill when you say things like

What you say	What the agent does
"Deploy PostgreSQL"	`elestio deploy postgresql` with guided setup
"I need a Redis cache"	`elestio deploy redis` with guided setup
"Set up n8n for automation"	`elestio deploy n8n` with guided setup
"Deploy my app from GitHub"	Automated CI/CD pipeline workflow
"What services are running?"	`elestio services`
"How much is this costing?"	`elestio billing`
"Reboot my PostgreSQL server"	`elestio reboot <vmID>`
"Add a firewall rule for port 8080"	`elestio firewall update <vmID>`

The Interactive Deployment Procedure

The skill **mandates** that the agent ask you for each deployment parameter individually. It will never deploy with default values without your confirmation.

Parameters asked (in order):

1. Cloud Provider

Netcup, Hetzner, AWS, Azure, GCP, DigitalOcean, Vultr, Scaleway, and Linode.

> Recommended: Netcup, Hetzner, AWS, and Azure.

2. Region *(based on provider chosen)*

- Netcup: `nbg` (Europe – Germany) | `mns` (North America – US)
- Hetzner: `fsn1`, `nbg1`, `hel1` (EU), `ash` (US)
- AWS: `us-east-1`, `eu-west-1`, etc.

3. Service Plan / Size

Starts at $14/month in the EU ($16 in the US) for a dedicated VM with 2 CPUs, 4 GB RAM, and 60 GB NVMe storage on Netcup.

You can check more here: https://elest.io/pricing#pricing

4. Service Name

The agent suggests a default (e.g., "prod-postgresql") and waits for your text input.

5. Admin Email

The agent proposes your email and lets you confirm or change it.

Example conversation

```
You:   "Deploy PostgreSQL"
Agent: Which cloud provider? (Netcup recommended for value, Hetzner for features)
You:   "Netcup"
Agent: Which region? (nbg = Europe Germany, mns = North America US)
You:   "nbg"
Agent: Which plan? SMALL ~$7/mo · MEDIUM ~$14/mo · LARGE ~$28/mo · XL ~$55/mo
You:   "MEDIUM"
Agent: What name for this service? Suggestion: prod-postgresql
You:   "demo-postgres"
Agent: Admin email? Suggestion: you@company.com
You:   "Confirmed"
Agent: [runs elestio deploy postgresql --project 112 --name demo-postgres ...]
Agent: ✓ Service deployed! Credentials: ...
```

What the agent CANNOT do for you

The skill explicitly prohibits the agent from performing these steps (you must do them in a browser):

- Creating an Elestio account (signup)
- Adding a payment method
- Waiting for account approval

After deployment: what the agent will do next

Once a service is deployed, the agent follows this sequence automatically:

1. **Get credentials:** `elestio credentials <vmID>`
2. **Enable auto-backups:** `elestio backups auto-enable <vmID>`
3. **Add custom domain** (if requested): `elestio ssl add <vmID> myapp.example.com`
4. **Configure firewall** (if requested): `elestio firewall enable <vmID> --rules [...]`
5. **Enable auto-updates:** `elestio updates system-enable <vmID> --security-only`

Supported agents & skill locations

Agent	Skill file location
Claude Code	`~/.claude/skills/elestio/SKILL.md`
OpenAI Codex	`~/.codex/skills/elestio/SKILL.md`
Cursor	`~/.cursor/rules/elestio.md`
OpenCode	`~/.config/opencode/skills/elestio/SKILL.md`

After installing or updating the skill, **restart your agent tool** to load the new instructions.

Version & source

- **Skill version:** 2.0
- **GitHub:** https://github.com/elestio/elestio-skill
- **Specification:** https://agentskills.io/specification

Deploying Services

Elestio supports two deployment modes:

- **Catalog Deployment** (400+ pre-built open-source templates)
- **CI/CD Deployment** (your own custom code from GitHub or GitLab).

Catalog Deployment

Use this when you need a standard open-source service (database, monitoring tool, CMS, etc.).

Find a template

```
# Search by name
elestio templates search postgresql
elestio templates search redis
elestio templates search wordpress

# Browse all templates
elestio templates

# Get template details (ID, versions, supported providers)
elestio templates info postgresql
```

Deploy a template

```
# Minimal deploy (uses defaults: netcup / nbg / MEDIUM-2C-4G)
elestio deploy postgresql --project 112 --name my-db

# Full options
elestio deploy postgresql \
  --project 112 \
  --name prod-postgres \
  --provider hetzner \
  --region fsn1 \
  --size LARGE-4C-8G \
  --support level2
```

Wait for the deployment to complete

```
elestio wait <vmID>
# Polls until deploymentStatus = "Deployed"
```

Get your service credentials

```
elestio credentials <vmID>
# Returns: URL, admin username, admin password
```

Common catalog examples

```
# Databases
elestio deploy postgresql --project 112 --name prod-pg
elestio deploy mysql --project 112 --name prod-mysql
elestio deploy redis --project 112 --name cache
elestio deploy mongodb --project 112 --name docs-db

# Analytics & monitoring
elestio deploy grafana --project 112 --name metrics
elestio deploy metabase --project 112 --name bi

# Automation
elestio deploy n8n --project 112 --name automation
elestio deploy temporal --project 112 --name workflows

# CMS & web
elestio deploy wordpress --project 112 --name blog
elestio deploy ghost --project 112 --name newsletter
```

CI/CD Deployment (Custom Code)

Use this when you want to deploy your own application from a GitHub or GitLab repository.

Decision tree

```
Is your app in a GitHub/GitLab repo?
├── YES → Automated CI/CD (recommended)
│         Step 1: elestio deploy cicd --project 112 --name my-cicd
│         Step 2: elestio cicd create --auto --target <vmID> --name my-app --repo owner/repo
```

```
--mode github
│
└── NO (custom Docker) → Manual CI/CD
          Step 1: elestio deploy cicd --project 112 --name my-cicd
          Step 2: elestio ssh-keys add <vmID> --name "agent-key" --key "ssh-ed25519 ..."
          Step 3: elestio cicd template docker > pipeline.json  (edit this file)
          Step 4: elestio cicd create pipeline.json
          Step 5: SSH in and configure docker-compose.yml
```

Automated GitHub/GitLab deploy

```
# 1. Deploy a CI/CD target VM
elestio deploy cicd --project 112 --name my-cicd-target

# 2. Auto-create the pipeline (CLI handles SSH, Dockerfile, build, start)
elestio cicd create --auto \
  --target <vmID> \
  --name my-app \
  --repo owner/repo \
  --mode github \
  --auth-id <githubAuthID>

# Site is live at: https://my-app-u<userID>.vm.elestio.app/
```

Modes available: `github`, `github-fullstack`, `gitlab`, `gitlab-fullstack`, `docker`

Optional flags:

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

What does auto-create do automatically

1. Discovers the Git account linked to the auth ID
2. Finds the repository and selects the correct branch
3. Creates the pipeline via the Elestio API
4. Adds an SSH key to the CI/CD target VM

5. Writes a correct multi-stage Dockerfile (Node build + Nginx serve)
6. Builds the Docker image
7. Starts the container
8. Verifies HTTP 200 on the service URL

Total time from `cicd create --auto` to live URL: **~2 minutes** after the CI/CD target is deployed.

Available Providers & Sizes

```
# See all available provider/region/size combinations with pricing
elestio sizes

# Filter by provider
elestio sizes --provider netcup
elestio sizes --provider hetzner
elestio sizes --provider aws
```

Common sizes

Size	CPU	RAM	~Price/mo
MEDIUM-2C-4G	2 cores	4 GB	~$14
LARGE-4C-8G	4 cores	8 GB	~$26
XL-8C-16G	8 cores	16 GB	~$49

You can find more at https://elest.io/pricing#pricing.

Managing Services

Listing & Inspecting Services

```
# List all services across all projects
elestio services

# Filter by project
elestio services --project 112

# Get detailed info for one service
elestio service <vmID>
```

Power Management

```
elestio reboot <vmID>          # Graceful OS reboot (safest)
elestio restart-stack <vmID>   # Restart Docker containers only (fastest, no OS reboot)
elestio shutdown <vmID>        # Graceful shutdown via ACPI signal
elestio poweroff <vmID>        # Force power off (like unplugging)
elestio poweron <vmID>         # Power on a stopped VM
elestio reset <vmID>           # Hard power cycle
```

> **Tip:** Prefer `restart-stack` When you just need to restart the application, it is much faster than a full `reboot` and avoids OS startup time.

Resizing a Service

```
# Upgrade or downgrade VM size
elestio resize <vmID> --size LARGE-4C-8G

# Check available sizes first
elestio sizes --provider netcup
```

> **Important:** Not all providers support downgrades. Only **Netcup, AWS, Azure, and Scaleway** allow downgrading instance sizes. Attempting a downgrade on Hetzner or GCP

will be blocked automatically by the CLI.

Termination Protection

Prevent accidental deletion:

```
elestio lock <vmID>            # Enable termination protection
elestio unlock <vmID>          # Disable termination protection
```

Firewall Management

```
# View current rules
elestio firewall get <vmID>

# Enable firewall with rules
elestio firewall enable <vmID> --rules '[
  {"type":"INPUT","port":"22","protocol":"tcp","targets":["0.0.0.0/0"]},
  {"type":"INPUT","port":"443","protocol":"tcp","targets":["0.0.0.0/0"]},
  {"type":"INPUT","port":"8080","protocol":"tcp","targets":["10.0.0.0/8"]}
]'

# Update all rules (replaces existing ruleset)
elestio firewall update <vmID> --rules '[...]'

# Disable firewall
elestio firewall disable <vmID>
```

Firewall rule format

```
{
  "type": "INPUT",
  "port": "22",
  "protocol": "tcp",
  "targets": ["0.0.0.0/0", "::/0"]
}
```

- `port`: single port (`"22"`) or range (`"8000-8080"`)
- `protocol`: `"tcp"` or `"udp"`
- `targets`: list of CIDR ranges; `"0.0.0.0/0"` means all IPv4

SSL & Custom Domains

```
# List current domains
elestio ssl list <vmID>

# Add a custom domain (SSL auto-provisioned via Let's Encrypt)
elestio ssl add <vmID> myapp.example.com

# Remove a domain
elestio ssl remove <vmID> myapp.example.com
```

> Point your domain's DNS A record to the service's IPv4 address before running `ssl add`.

SSH Keys

```
elestio ssh-keys list <vmID>
elestio ssh-keys add <vmID> --name "my-laptop" --key "ssh-ed25519 AAAA..."
elestio ssh-keys remove <vmID> --name "my-laptop"
```

> Provide only the key type + key data. **Do not** include the trailing comment (e.g., `user@host`).

SSH & Web Access

```
elestio ssh <vmID>             # Open SSH web terminal URL in browser
elestio ssh <vmID> --direct    # Print the direct SSH command
elestio vscode <vmID>          # Open VSCode in browser
elestio files <vmID>           # Open web file manager
elestio credentials <vmID>     # Print application URL + admin credentials
```

Auto-Updates

```
# OS security updates only, every Sunday at 05:00
elestio updates system-enable <vmID> --day 0 --hour 5 --security-only

# All OS updates
elestio updates system-enable <vmID> --day 0 --hour 5
```

```
# Disable OS auto-updates
elestio updates system-disable <vmID>

# Trigger an OS update now
elestio updates system-now <vmID>

# Application-level updates (the Docker image)
elestio updates app-enable <vmID> --day 0 --hour 3
elestio updates app-disable <vmID>
elestio updates app-now <vmID>

# Upgrade/downgrade application version
elestio change-version <vmID> 16    # e.g., PostgreSQL 16
```

`--day` **values:** 0 = Sunday, 1 = Monday, ... 6 = Saturday

Moving a Service

```
elestio move-service <vmID> <targetProjectId>
```

Deleting a Service

```
elestio delete-service <vmID> --force
```

> Always use `elestio lock <vmID>` first if you want to protect a service from accidental deletion.

VM Architecture

Every service runs on a dedicated VM with this layout:

```
/opt/elestio/nginx/          ← Reverse proxy (managed by Elestio, do not modify)
/opt/app/                    ← Your application
    ├── docker-compose.yml   ← For catalog services
    └── <pipeline_name>/     ← For CI/CD pipelines
```

- A reverse proxy handles HTTPS termination automatically

- SSL is auto-renewed via Let's Encrypt
- Internal Docker network: `172.17.0.1`
- Bind application ports to `172.17.0.1:PORT` not `0.0.0.0`

Backups & Storage

Elestio provides three backup tiers and two storage extension types (volumes and snapshots).

Local Backups (Application-Level)

Application-consistent backups are stored on the service VM itself. Created using pre/post-backup scripts specific to each template.

```
# List local backups
elestio backups local-list <vmID>

# Create a local backup now
elestio backups local-take <vmID>

# Restore from a local backup
elestio backups local-restore <vmID> /opt/app-backups/backup.zst

# Delete a local backup
elestio backups local-delete <vmID> /opt/app-backups/backup.zst
```

Remote Backups (Elestio Managed)

Backups are stored in Elestio's managed storage infrastructure, separate from the VM.

```
# List remote backups
elestio backups remote-list <vmID>

# Take a remote backup now
elestio backups remote-take <vmID>

# Restore from a remote backup
elestio backups remote-restore <vmID> <snapshot-name>

# Enable automatic remote backups (daily)
```

```
elestio backups auto-enable <vmID>

# Disable automatic remote backups
elestio backups auto-disable <vmID>
```

S3 External Backups

Store backups in your own S3-compatible bucket (AWS S3, MinIO, Backblaze B2, etc.).

```
# Test S3 connectivity before enabling
elestio s3-backup verify <vmID> \
  --key ACCESS_KEY \
  --secret SECRET_KEY \
  --bucket my-backups \
  --endpoint s3.amazonaws.com

# Enable S3 backups
elestio s3-backup enable <vmID> \
  --key ACCESS_KEY \
  --secret SECRET_KEY \
  --bucket my-backups \
  --endpoint s3.amazonaws.com

# Take an S3 backup now
elestio s3-backup take <vmID>

# List S3 backups
elestio s3-backup list <vmID>

# Restore from S3
elestio s3-backup restore <vmID> <backup-key>

# Delete an S3 backup
elestio s3-backup delete <vmID> <backup-key>

# Disable S3 backups
elestio s3-backup disable <vmID>
```

Snapshots (Provider-Level)

Point-in-time disk snapshots. Support varies by provider. Hetzner has full snapshot support; **Netcup** has limited support.

```
# List all snapshots
elestio snapshots list <vmID>

# Create a manual snapshot
elestio snapshots take <vmID>

# Restore a snapshot (use 0 for the most recent)
elestio snapshots restore <vmID> <snapshotId>
elestio snapshots restore <vmID> 0

# Delete a snapshot
elestio snapshots delete <vmID> <snapshotId>

# Enable automatic snapshots
elestio snapshots auto-enable <vmID>

# Disable automatic snapshots
elestio snapshots auto-disable <vmID>
```

Volumes (Block Storage)

Persistent storage volumes that can be attached to services. **Hetzner** provides full volume operations; other providers may have limited or no support.

```
# List all volumes in a project
elestio volumes

# Create a standalone volume
elestio volumes create --name my-volume --size 20

# List volumes attached to a service
elestio volumes service-list <vmID>

# Create and attach a volume to a service
elestio volumes service-create <vmID> --name app-data --size 50

# Resize a volume
```

```
elestio volumes resize <vmID> <volumeID> --size 100

# Detach a volume from a service
elestio volumes detach <vmID> <volumeID>

# Delete a volume
elestio volumes delete <vmID> <volumeID>

# Protect a volume from accidental deletion
elestio volumes protect <vmID> <volumeID>
```

Recommended Backup Strategy

Tier	Frequency	Retention	Use for
Local	Manual	No	Fast restore, same VM
Remote Borg (auto)	Daily	Based on the support plan	VM failure, corruption, restore
S3 external	Daily	Customizable	Compliance, cross-region

Enable all three for production services:

```
# Enable auto remote backups
elestio backups auto-enable <vmID>

# Enable S3 backups
elestio s3-backup enable <vmID> --key X --secret Y --bucket Z --endpoint S

# Enable auto snapshots (Hetzner only)
elestio snapshots auto-enable <vmID>
```

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>
```

Cloud Provider Guide

Elestio supports 9 cloud providers across 100+ regions. Each provider has different features, pricing, and capabilities.

Provider Comparison

Feature	Netcup	Hetzner	AWS	Azure	Scaleway	Vultr	Linode	DO
Catalog deploy	✓	✓	✓	✓	✓	✓	✓	✓
CI/CD deploy	✓	✓	✓	✓	✓	✓	✓	✓
Volumes	✗	✓	✓	✓	✓	✓	✓	✓
Snapshots	✗	✓	✓	✓	✓	✓	✓	✓
Size downgrade	✓	Only If last upgraded without disk	✓	✓	✓	—	—	—
Firewall	✓	✓	✓	✓	✓	✓	✓	✓
SSL/Domains	✓	✓	✓	✓	✓	✓	✓	✓

Choosing a Provider

Use Netcup when:

- You want the **best price/performance ratio**
- You don't need block volumes or full snapshots
- You may need to **downgrade** your instance size later
- EU-based (Germany) is acceptable

Use Hetzner when:

- You need **block volumes** for persistent storage
- You need reliable **snapshots** and power management
- You want a strong EU-based infrastructure (Germany, Finland)
- Note: Hetzner does **not** support size downgrades

Use AWS when:

- You need global reach (many regions worldwide)
- You need size-downgrade flexibility
- You're already in the AWS ecosystem

Use Azure when:

- You need Microsoft cloud infrastructure
- You need size-downgrade flexibility
- You need specific Azure regions (e.g., `germanywestcentral`)

Available Regions

```
# See all regions for a provider
elestio sizes --provider netcup
elestio sizes --provider hetzner
elestio sizes --provider aws
elestio sizes --provider azure
```

Common regions

Netcup:

Region	Location
`nbg`	Nuremberg, Germany (EU)
`mns`	Manassas, Virginia, USA

Hetzner:

Region	Location
`fsn1`	Falkenstein, Germany
`nbg1`	Nuremberg, Germany
`hel1`	Helsinki, Finland
`ash`	Ashburn, Virginia, USA

AWS: `us-east-1`, `us-west-2`, `eu-west-1`, `eu-central-1`, `ap-southeast-1`, `ap-northeast-1`, and more.

Specifying a Provider at Deploy Time

```
elestio deploy postgresql \
  --project 112 \
  --name my-db \
  --provider hetzner \
  --region fsn1 \
  --size LARGE-4C-8G
```

Size Downgrade Warning

Attempting to downgrade on an unsupported provider (Hetzner, GCP) can **permanently block the service** and require Elestio support intervention. The CLI will automatically refuse downgrade requests on unsupported providers.

Only these providers support downgrading:

- Netcup
- AWS
- Azure
- Scaleway

Support Tiers

Available on all providers:

Tier	Monthly Cost	Response Time	Channel
`level1`	Included	48 hours	Email
`level2`	+$50/service/mo	24 hours	Priority email
`level3`	+$200/service/mo	4 hours	Dedicated engineer

```
elestio deploy postgresql --project 112 --support level2
```

Troubleshooting & Error Reference

Error Reference Table

Error message	Cause	Solution
`Not configured`	No credentials stored	Run `elestio login --email X --token Y`
`Authentication failed`	Wrong credentials or expired token	Regenerate token at dash.elest.io/account/security
`Account not approved`	New account without approval	Wait for approval or contact support@elest.io
`Template not found`	Wrong name or ID	Run `elestio templates search <name>`
`Invalid serverType`	Provider/region/size combo invalid	Check `elestio sizes --provider X`
`Service not found`	Wrong vmID or wrong project	Run `elestio services --project X`
`Deployment timeout`	Deployment is taking too long	Check dashboard; contact support if > 10 min
`Project not found`	Wrong projectId	Run `elestio projects` to find the correct ID

Authentication Issues

"Authentication failed" repeatedly

```
# Step 1: Check current config
elestio config

# Step 2: Re-generate token in the Elestio dashboard
# Go to: https://dash.elest.io/account/security → Manage API Tokens → Create Token

# Step 3: Update credentials
elestio login --email "you@example.com" --token "new_token"

# Step 4: Verify
```

```
elestio auth test
```

The wrong user is being used

```
elestio whoami   # Check which account is active
elestio config   # See stored configuration
```

Deployment Issues

Service stuck in "Deploying" state

1. Normal deployment takes **2–5 minutes**
2. CI/CD target deployment may take up to **10 minutes**
3. If > 10 minutes: check the Elestio dashboard for error messages
4. Contact support@elest.io if still stuck after 15+ minutes

```
# Poll until deployed
elestio wait <vmID>

# Check current status
elestio service <vmID>
```

"Service not found," but it exists

```
# Verify you are using vmID (not serverID)
elestio services --project 112

# Both IDs are shown — ensure you use the correct one:
# vmID  → most commands (firewall, ssl, power, etc.)
# id    → backup endpoints
```

"Invalid serverType" when deploying

```
# Check valid combinations for your provider
elestio sizes --provider netcup
elestio sizes --provider hetzner
```

Pipeline Issues

Pipeline not building

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

# Restart the pipeline
elestio cicd pipeline-restart <vmID> <pipelineID>
```

App deployed but not reachable in the browser

1. Check that your app is binding to the internal network, not `0.0.0.0`:
   - Correct: `172.17.0.1:3000`
   - Wrong: `0.0.0.0:3000`
2. SSH in and check containers:

```
ssh root@<ipv4>
cd /opt/app/<pipeline-name>
docker-compose ps
docker-compose logs
```

Pipeline fails after code push

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

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

Firewall Issues

Can't reach the service after enabling the firewall

1. Ensure port 443 (HTTPS) is in the rules
2. Ensure port 22 (SSH) is included if you need SSH access
3. Re-check rules:

```
elestio firewall get <vmID>
```

4. Temporarily disable to test:

```
elestio firewall disable <vmID>
```

SSL/Domain Issues

SSL certificate not provisioning

1. Ensure your DNS A record points to the service's IPv4 **before** running `ssl add`
2. DNS propagation can take up to 48 hours (typically 5–30 minutes)
3. Check: `elestio ssl list <vmID>`
4. Remove and re-add if stuck: `elestio ssl remove <vmID> <domain>` then re-add

Resize/Downgrade Issues

Service blocked after downgrade attempt

This happens when trying to downgrade on a provider that doesn't support it (Hetzner, GCP).

Supported for downgrade: Netcup, AWS, Azure, and Scaleway only.

Resolution: Contact support@elest.io with the vmID. They can restore the service.

General Debugging Flow

```
# 1. Verify authentication
elestio auth test

# 2. Check service status
elestio service <vmID>

# 3. Restart the application stack (fastest recovery)
elestio restart-stack <vmID>

# 4. If pipeline involved, check logs
elestio cicd pipeline-logs <vmID> <pipelineID>

# 5. Full reboot if restart-stack doesn't help
elestio reboot <vmID>
```

Getting Help

Channel	Details
Email support	support@elest.io

Channel	Details
Dashboard	https://dash.elest.io
API Docs	https://api-doc.elest.io
Templates	https://elest.io/fully-managed-services