

# 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`
- 

Revision #2

Created 2026-02-20 13:21:20 UTC by Amit Shukla

Updated 2026-02-20 13:31:23 UTC by Amit Shukla