interweb

Generated on 15/11/2026 01:46

KNOWLEDGE BASE · PUBLIC DOCUMENT

How to run your own program as a systemd service that starts at boot

Administration & Fixes

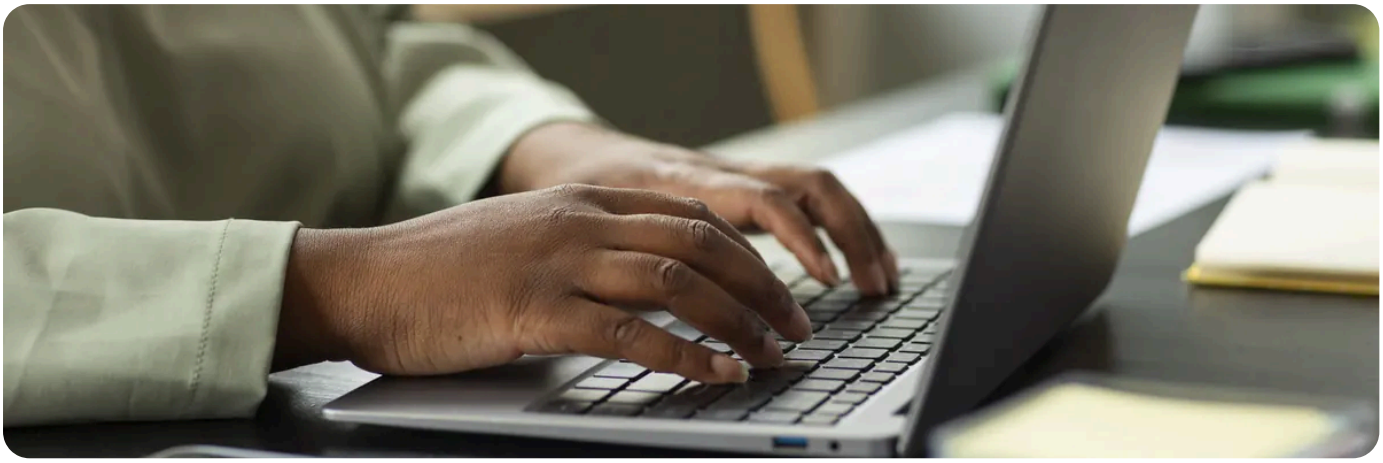
CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-1386	v2026.10.10	Mozambique · English

Interweb Mozambique
RLX Holdings, S.A. · Group RLX Holdings
Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

https://www.meu.interweb.co.mz/knowledgebase/1386

How to run your own program as a systemd service that starts at boot

Administration & Fixes Created on **October 10, 2026** 3 min read



For a program of yours (a Node.js app, a Python script, a bot) to start with the server and come back by itself if it dies, you write a **unit file** in `/etc/systemd/system/`, reload systemd and enable the service with `systemctl enable --now`. It is the native way on any modern Linux, with nothing extra to install. For Node.js apps there is also PM2 (**keeping a Node.js app running with PM2**); systemd covers everything else, and Node too.

Step by step

1. **Run the program by hand first.** If it does not start in the terminal, it will not start as a service. Note the program's full path (`which node` or `which python3`) and the folder it runs from.

2. **Create an unprivileged user** for the service instead of running it as root: `sudo useradd --system --home /opt/myapp --shell /usr/sbin/nologin myapp`. If the program is compromised, the damage stays tied to that user.

3 **Write the unit** with `sudo nano /etc/systemd/system/myapp.service`

```
[Unit]
Description=My application
After=network.target

[Service]
User=myapp
WorkingDirectory=/opt/myapp
ExecStart=/usr/bin/node /opt/myapp/server.js
Restart=on-failure
RestartSec=5
EnvironmentFile=/etc/myapp.env

[Install]
WantedBy=multi-user.target
```

4 **Keep the secrets in a separate file,** `/etc/myapp.env`, one `NAME=value` line per variable, and protect it: `sudo chmod 600 /etc/myapp.env`. See **environment variables and secrets**.

5 **Tell systemd there is a new unit** and start it: `sudo systemctl daemon-reload` then `sudo systemctl enable --now myapp`. "enable" makes it start with the server; "--now" starts it right away.

6 **Confirm.** `systemctl status myapp` should say "active (running)", and `journalctl -u myapp -f` shows what the program writes, live.

7 **Prove the reboot.** Restart the server at a quiet hour and see whether the service returns. That is the only way to know it works.

The unit's lines, in words

Line	What it is for
`After=network.target`	Starts only after the network is ready.
`User=`	Who runs the program. Never root, unless it truly needs it.
`ExecStart=`	The command, with the program's **full path**. systemd uses neither your PATH nor your terminal.
`Restart=on-failure`	Starts again if it ends with an error; does not if you stopped it on purpose.
`EnvironmentFile=`	Reads variables from a file instead of writing them in the unit (which any user can read).

Line	What it is for
`WantedBy=multi-user.target`	What makes `enable` start it at the server's normal boot.

`ExecStart` is not a terminal.
Redirections, `&&` and `~` do not work as they do at the prompt; use absolute paths and, if you need several commands, put them in a script and run the script. If you see "status=203/EXEC", the path is wrong: **reading systemctl status**.

Changed the unit and nothing changed?
You are missing `sudo systemctl daemon-reload`. And a unit that gets "Permission denied" reading a folder is permissions or SELinux: **file permissions**.

Scheduling with a timer instead of cron:
Systemd schedules too (`.timer` files), but for simple jobs cron is still the most direct: **when cron does not run**.

NEED A HAND?

Does the program run by hand and fail as a service, and you have read the log? Show us the error text and what changed since it worked.

Open a support ticket →

SEE ALSO

- A service will not start: reading systemctl status →
- Where the logs are on a Linux VPS →
- Keeping a Node.js app running with PM2 →
- How far our support goes →

interweb

Interweb Mozambique

RLX Holdings, S.A. · Group RLX Holdings

TAX ID (NUIT)	401 191 607
LEGAL ENTITY NO. (NUEL)	105 045 974
ECONOMIC ACTIVITY CODE (CAE)	62022
REGISTERED OFFICE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT	info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-1386 · v2026.10.10 · Generated on 15/11/2026 01:46