interweb

Generated on 15/11/2026 01:44

KNOWLEDGE BASE · PUBLIC DOCUMENT

Running Flask or Django with gunicorn and systemd on a VPS

Python & Other Languages

CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-1108	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/1108

Running Flask or Django with gunicorn and systemd on a VPS

Python & Other Languages Created on **October 10, 2026** 3 min read



The server that ships with Flask and Django (`flask run` , `manage.py runserver`) is for development, not for the public. On a VPS, the usual combination is **gunicorn** (which runs the application with several processes) managed by **systemd** (which starts it with the server and brings it back if it dies), with **nginx** in front handling HTTPS. This article builds the first two pieces. It is for a VPS with root; on shared hosting you use “Setup Python App” instead.

Step by step

1 **Create the virtual environment and install gunicorn** in the application folder, with a user of its own (not root):

```
cd /home/appuser/app
python3 -m venv .venv
.venv/bin/pip install -r requirements.txt gunicorn
```

If `pip` complains about “externally-managed-environment”, see **that error**.

2 **Test by hand first.** Flask:

```
.venv/bin/gunicorn --bind 127.0.0.1:8000 app:app
```

`app:app` means “the file `app.py` , the variable `app` ”. In Django, use the project’s WSGI module: `myproject.wsgi:application` . If it answers `curl http://127.0.0.1:8000` , stop with Ctrl+C and move on.

3 **Create the service**, with `sudo nano /etc/systemd/system/my-app.service` :

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

[Service]
User=appuser
Group=appuser
WorkingDirectory=/home/appuser/app
EnvironmentFile=/home/appuser/app/.env
ExecStart=/home/appuser/app/.venv/bin/gunicorn --workers 3 --bind 127.0.0.1:8000
app:app
Restart=on-failure

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

4 **Enable and start it:**

```
sudo systemctl daemon-reload
sudo systemctl enable --now my-app
sudo systemctl status my-app
```

`enable` makes it start with the server; `--now` starts it right away.

5 **Read the logs** when something fails: `journalctl -u my-app -n 100` . After updating the code: `sudo systemctl restart my-app` .

The points that usually cause trouble

Point	What to do
Number of processes (`--workers`)	The 3 above is an example. Each process uses memory: start with few and see what the VPS can take.
The `bind` address	`127.0.0.1` only accepts requests from the server itself, which is what you want behind a proxy. Do not use `0.0.0.0` without a reason.
Secrets	In the file pointed to by `EnvironmentFile` , with `chmod 600` , not written into the service or the code.
Static files (Django)	gunicorn does not serve them well: leave them to nginx, or see **Django static files**.
The service will not start	`systemctl status` and `journalctl` tell you why: **reading systemctl status**.

Always use the virtual environment's gunicorn

in `ExecStart`, with the full path. systemd does not activate the environment: if you write just `gunicorn`, either it uses the system's (and cannot find your libraries) or it finds none at all. And never run the application as root.

HTTPS and the domain are still missing.

Gunicorn behind nginx is done as in **nginx as a reverse proxy**: `proxy_pass` points to `http://127.0.0.1:8000`. For the general picture of services of your own, see **running your own program as a service**. On an unmanaged VPS the service is yours to maintain: **how far our support goes**.

RECOMMENDED

Need a server for your Python application? Have a look at our VPS plans.

See the VPS servers →

SEE ALSO

- **Running a Flask or FastAPI application** →
- **How to run your own program as a systemd service that starts at boot** →
- **Nginx as a reverse proxy in front of a container** →

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-1108 · v2026.10.10 · Generated on 15/11/2026 01:44