Mozout

Generated on 15/11/2026 00:39

KNOWLEDGE BASE · PUBLIC DOCUMENT

Writing a Dockerfile for a Node.js or Python app

Docker & Containers

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-3432	v2026.10.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/3432

Writing a Dockerfile for a Node.js or Python app

Docker & Containers Created on **October 10, 2026** 3 min read



The **Dockerfile** is the recipe that turns your application into an **image**: a package with the code, the libraries and the environment, which runs the same on any server with Docker. **The short answer:** start from an official image, copy the dependency file first and the code after (so Docker can reuse its work), and run the application as a user that is not root. It is for a **VPS** with Docker installed: **how to install it**.

An example for each language

Node.js. The version numbers are examples: use the one your project supports.

```
FROM node:22-slim
WORKDIR /app
COPY package*.json ./
RUN npm ci --omit=dev
COPY . .
USER node
EXPOSE 3000
CMD ["node", "server.js"]
```

Python (Flask, with gunicorn):

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
RUN useradd --create-home appuser
USER appuser
EXPOSE 8000
CMD ["gunicorn", "--bind", "0.0.0.0:8000", "app:app"]
```

What each line does

Line	What it does
`FROM`	The starting image. Prefer the official ones and the smaller `slim` variants. Pin a version instead of `latest`.
`WORKDIR`	The working folder inside the image.
`COPY` the dependency file, `RUN` the install, `COPY` the code	In this order, Docker keeps the dependencies layer and only redoes it when that file changes. If you copy the code first, it reinstalls everything on every change.
`USER`	Runs the application as something other than root. If anything compromises it, it reaches less.
`EXPOSE`	Only documents the port. It does not open it: `-p` or Compose does the publishing.
`CMD`	The command that starts the application. Write it as a list, like above.

Build and test

1. **Create a `.dockerignore` file** beside the Dockerfile, one item per line: `node_modules`, `.git`, `.env`, `.venv`. Without it, the heavy folder and the secrets end up in the image.

2. **Build:**

```
docker build -t my-app .
```

3. **Run and test** on the server itself only:

```
docker run -d --name my-app -p 127.0.0.1:3000:3000 my-app
curl http://127.0.0.1:3000
```

4. **If it will not start,** see **what to look at when a container will not start**. For the application with its database, move on to Compose: **an application with its database**.

Inside the container, the application must listen on `0.0.0.0`, not `127.0.0.1`. On `127.0.0.1` only the container itself reaches it, and `-p` cannot connect anything to it. It is the most frequent cause of “the container runs but does not answer”. What keeps it on the server only is the `127.0.0.1` on the *left side* of `-p`.

Secrets never go in the image.

What you write in an `ENV` or copy inside stays recorded in the layers, and anyone with the image can read it. Pass passwords and keys at run time (`--env-file` or Compose). On an unmanaged VPS, the image and the container are yours: **how far our support goes**.

RECOMMENDED

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

See the VPS servers →

SEE ALSO

- **Installing Docker and Docker Compose on a VPS** →
- **Your first container, and what to look at when it will not start** →
- **Docker Compose: an application with its database** →

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-3432 · v2026.10.10 · Generated on 15/11/2026 00:39