Mozout

Gerado a 14/11/2026 22:20

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Escrever um Dockerfile para uma aplicação Node.js ou Python

Docker e contêineres

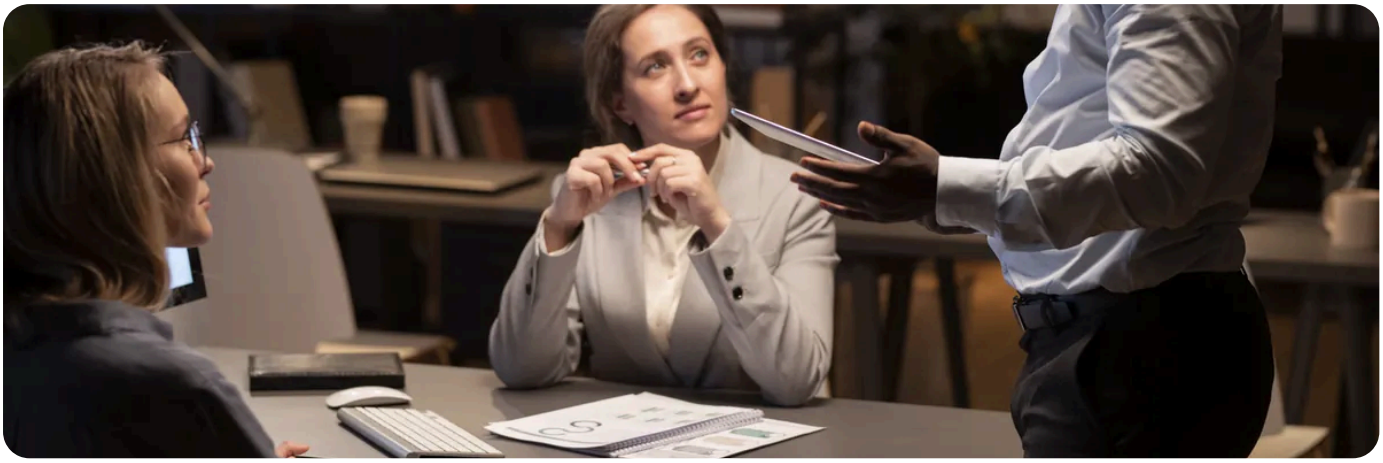
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-3432	v2026.10.10	Brasil · Português (Brasil)

MozOut Brasil
RLX Holdings (Pty) Ltd. · Grupo RLX Holdings
N.º fiscal (SARS) 943 907 6234 · N.º de registo (CIPC) 2025/667953/07

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

Escrever um Dockerfile para uma aplicação Node.js ou Python

Docker e contêineres Criado em **10 de outubro de 2026** 3 min de leitura



O **Dockerfile** é a receita que transforma a sua aplicação numa **imagem**: um pacote com o código, as bibliotecas e o ambiente, que corre igual em qualquer servidor com Docker. **A resposta curta:** comece de uma imagem oficial, copie primeiro o arquivo das dependências e só depois o código (para o Docker reaproveitar o trabalho), e rode a aplicação com um usuário que não seja o root. É para um **VPS** com Docker instalado: **como instalar**.

Um exemplo para cada linguagem

Node.js. Os números das versões são exemplos: use a que o seu projeto suporta.

```
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, com 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"]
```

O que faz cada linha

Linha	O que faz
`FROM`	A imagem de partida. Prefira as oficiais e as variantes `slim`, menores. Fixe uma versão em vez de `latest`.
`WORKDIR`	A pasta de trabalho dentro da imagem.
`COPY` do arquivo de dependências, `RUN` da instalação, `COPY` do código	Por esta ordem, o Docker guarda a camada das dependências e só a refaz quando esse arquivo muda. Se copiar o código antes, reinstala tudo a cada alteração.
`USER`	Roda a aplicação sem ser root. Se algo a comprometer, alcança menos.
`EXPOSE`	Só documenta a porta. Não a abre: quem publica é o `-p` ou o Compose.
`CMD`	O comando que inicia a aplicação. Escreva-o em lista, como acima.

Construir e testar

1. **Crie o arquivo** `.dockerignore` ao lado do Dockerfile, com uma linha por item: `node_modules`, `.git`, `.env`, `.venv`. Sem ele, a pasta pesada e os segredos vão parar à imagem.

2. **Construa:**

```
docker build -t minha-app .
```

3. **Corra e teste** só no próprio servidor:

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

4. **Se não inicia,** veja **o que olhar quando um contêiner não inicia**. Para a aplicação com o seu banco de dados, passe ao Compose: **uma aplicação com o seu banco de dados**.

Dentro do contêiner, a aplicação tem de ouvir em `0.0.0.0`, não em `127.0.0.1`. Em `127.0.0.1` só o próprio contêiner a alcança, e o `-p` não consegue ligar nada a ela. É a causa mais frequente de «o contêiner corre mas não responde». Quem fica a ouvir só no servidor é o `127.0.0.1` do *lado esquerdo* do `-p`.

Segredos nunca na imagem.

O que escrever num `ENV` ou copiar para dentro fica gravado nas camadas, e quem tiver a imagem lê-o. Passe senhas e chaves ao rodar (`--env-file` ou o Compose). Num VPS não gerenciado, a imagem e o contêiner são seus: **até onde vai o nosso suporte**.

RECOMENDADO

Precisa de um servidor para os seus contêineres? Veja os VPS.

Ver os servidores VPS →

VEJA TAMBÉM

- **Instalar o Docker e o Docker Compose num VPS** →
- **O seu primeiro contêiner, e o que ver quando não inicia** →
- **Docker Compose: uma aplicação com o seu banco de dados** →

Mozout

MozOut Brasil

RLX Holdings (Pty) Ltd. · Grupo RLX Holdings

N.º FISCAL (SARS)	943 907 6234
N.º DE REGISTO (CIPC)	2025/667953/07
SEDE	72 York Road, Kensington Joanesburgo 2094, Gauteng, África do Sul
CONTACTOS	info@mozout.com · +27 78 823 3325 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-BR-KB-3432 · v2026.10.10 · Gerado a 14/11/2026 22:20