Mozout

Gerado a 21/10/2026 13:26

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Docker Compose: uma aplicação com o seu banco de dados

Node.js, Python e Docker

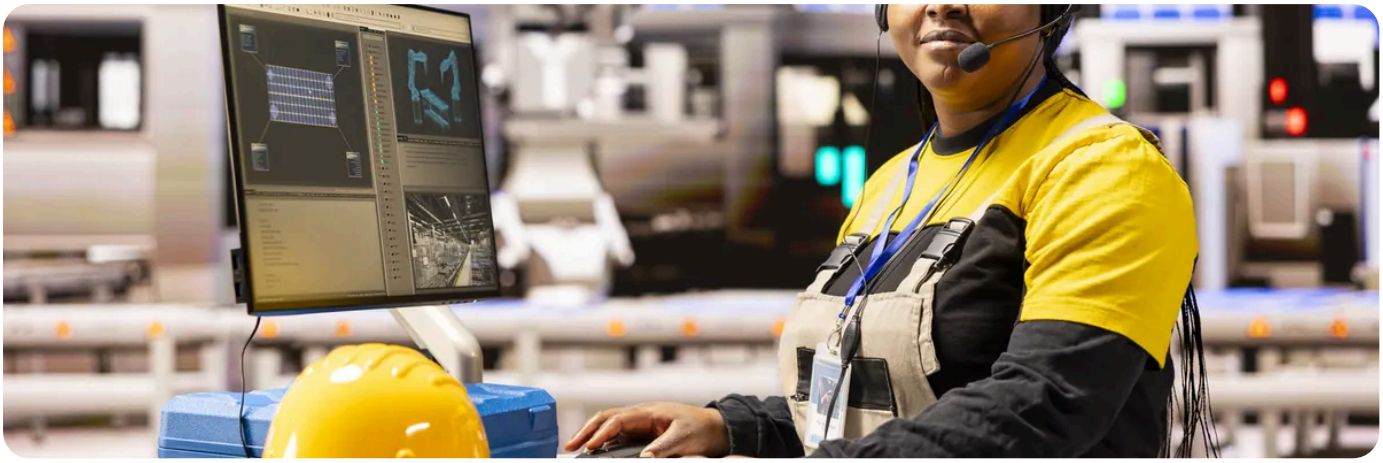
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-2730	v2026.10.10	Brasil · Português (Brasil)

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/2730

Docker Compose: uma aplicação com o seu banco de dados

Node.js, Python e Docker Criado em **10 de outubro de 2026** 3 min de leitura



O **Docker Compose** descreve num único arquivo uma aplicação inteira: o programa, o banco de dados, as redes entre eles e os volumes onde os dados ficam. Um comando, `docker compose up -d`, levanta tudo. Fica aqui um exemplo com a aplicação e o PostgreSQL, num **VPS** com o Docker já instalado (**veja como**). A mesma estrutura serve para MariaDB ou MySQL: muda a imagem e as variáveis.

Montar o projeto

1. **Crie uma pasta só para o projeto** e entre nela: `mkdir meu-projecto && cd meu-projecto`. O nome da pasta torna-se o nome do projeto no Compose.

2. **Crie o arquivo** `.env` com as senhas, e proteja-o:

```
DB_PASSWORD=ponha-aqui-uma-palavra-passe-longa-e-aleatoria
```

`chmod 600 .env`. Se usar Git, ponha o `.env` no `.gitignore`. Nunca o publique.

3 **Crie o arquivo `compose.yaml`** (o nome `docker-compose.yml` também funciona):

```
services:
  app:
    build: .
    restart: unless-stopped
    env_file: .env
    environment:
      DB_HOST: db
    ports:
      - "127.0.0.1:3000:3000"
    depends_on:
      db:
        condition: service_healthy
  db:
    image: postgres:16
    restart: unless-stopped
    environment:
      POSTGRES_USER: appuser
      POSTGRES_PASSWORD: ${DB_PASSWORD}
      POSTGRES_DB: appdb
    volumes:
      - db_data:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U appuser -d appdb"]
      interval: 10s
      timeout: 5s
      retries: 5
volumes:
  db_data:
```

4 **Levante tudo** e veja o estado:

```
docker compose up -d
docker compose ps
docker compose logs -f app
```

O Compose cria uma rede privada para o projeto.

O que cada parte faz

Parte	O que faz	Por quê assim
`DB_HOST: db`	A aplicação encontra o banco de dados pelo **nome do serviço**.	Dentro da rede do Compose não se usa `localhost` nem o IP: cada serviço é um computador à parte.
`127.0.0.1:3000:3000`	Publica a porta da aplicação **só no próprio servidor**.	Quem a mostra ao mundo é um proxy com HTTPS: veja **nginx como proxy reverso**.

Parte	O que faz	Por quê assim
Nenhuma porta no `db`	O banco de dados não é alcançável de fora.	É o desejado. Os outros serviços chegam-lhe pela rede interna.
`db_data` (volume)	Onde ficam os dados do banco de dados.	Sem volume, apagar o contêiner apaga os dados.
`healthcheck` e `service_healthy`	A aplicação só inicia quando o banco de dados já aceita conexões.	`depends_on` sozinho só espera que o contêiner *exista*, não que esteja pronto.

O endereço de conexão, dentro da aplicação, fica assim: `postgres://appuser:SENHA@db:5432/appdb`. Se a senha tiver caracteres especiais (`@` , `:` , `/`), têm de ir codificados no endereço, ou escolha uma sem eles.

As variáveis `POSTGRES_*` só valem na primeira vez.
Servem para criar o banco de dados quando o volume está vazio. Se mudar a senha no arquivo depois de a base existir, o banco de dados **não muda**, e a aplicação deixa de entrar. Muda-se com um comando dentro do banco de dados. E o `docker compose down -v` **apaga os volumes**, ou seja, os dados: sem o `-v` , um `down` só remove os contêineres.

Escolha uma versão da imagem do banco de dados e fixe-a
(`postgres:16` em vez de `postgres:latest`). Passar de uma versão principal para a seguinte não se faz mudando só o número: exige uma cópia e uma restauração. Veja **cópias com mysqldump e pg_dump**, e como atualizar sem perder nada em **atualizar uma aplicação Docker**. A escolha entre motores está em **PostgreSQL ou MySQL**.

RECOMENDADO
O VPS tem de ser seu, com root. Se ainda não tem, veja os planos: o Docker e a aplicação ficam à sua conta, e a rede e o servidor à nossa.
Ver os servidores VPS →

VEJA TAMBÉM
Nginx como proxy reverso à frente de um contêiner →
Atualizar uma aplicação Docker sem perder dados →

Cópias com mysqldump e pg_dump, e restaurar a cópia

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

RLX-BR-KB-2730 · v2026.10.10 · Gerado a 21/10/2026 13:26

Versão online deste documento