



VOLUMES E PERSISTÊNCIA DE DADOS NO DOCKER

Responsável: João Pedro Toledo Gonçalves.

Data: 26/01/2026

Código: ITGINF 0010/26 | **Classificação:** RESTRITO

Responsável: João Pedro Toledo Gonçalves | **Data:** 26/01/2026

1. HISTÓRICO DE REVISÃO

NOTA

REGRA DE OURO:

1. **Autor:** João Pedro Toledo Gonçalves.
2. **Descrição:** Criação do documento.

Data	Versão	Descrição	Autor
26/01/2026	1.0	Criação Inicial	João Pedro Toledo Gonçalves

2. OBJETIVO

Definir estratégias de persistência de dados em containers (Stateful), diferenciando Volumes gerenciados de Bind Mounts e garantindo backup e permissões corretas.

3. PRÉ-REQUISITOS

- ☐ Docker Engine instalado.
- ☐ Entendimento básico de permissões Linux (chmod/chown).

4. PASSO A PASSO (EXECUÇÃO)

Etapas 1: Volumes Gerenciados (Recomendado)

NOTA

NOTA: Volumes são gerenciados pelo Docker em `/var/lib/docker/volumes`. São mais fáceis de fazer backup e migrar.

1. ☒ Crie um volume explícito.

```
docker volume create dados-postgres
```

2. ☒ Use o volume em um container.

```
docker run -d --name db-prod -v dados-postgres:/var/lib/postgresql/data postgres:1
```

5

Etapa 2: Bind Mounts (Caminhos do Host)

IMPORTANTE

IMPORTANTE: Use Bind Mounts para arquivos de configuração (`nginx.conf`) ou código fonte em desenvolvimento.

1. [x] Mapeie um diretório local absoluto para dentro do container.

```
docker run -d \  
  --name web-server \  
  -v /home/user/meu-site:/usr/share/nginx/html:ro \  
  nginx
```

> `:ro` = Read Only (O container não pode alterar os arquivos do host).

Etapa 3: Inspeccionando Volumes

1. [x] Descubra onde os dados estão realmente salvos no disco.

```
docker volume inspect dados-postgres
```

Etapa 4: Backup de Volume

1. [x] Use um container temporário para tar/zipar o conteúdo de um volume.

```
docker run --rm \  
  -v dados-postgres:/volume \  
  -v $(pwd):/backup \  
  alpine tar cvf /backup/backup.tar /volume
```

5. SOLUÇÃO DE PROBLEMAS (TROUBLESHOOTING)

Problema 1: "Permission Denied" ao escrever no volume

- **Causa:** O UID do processo dentro do container é diferente do UID do dono da pasta no host (Bind Mount).
- **Solução:**
 1. [x] Descubra o UID interno: `docker exec container id`.
 2. [x] Ajuste o dono no host: `sudo chown -R 1000:1000 /pasta/local`.

Problema 2: Volume órfão (Dados "perdidos")

- **Causa:** Remover container sem remover volume mantém os dados (o que é bom), mas pode confundir.
- **Solução:**

1. [x] Liste volumes: ``docker volume ls``.
2. [x] Remova não usados: ``docker volume prune``.

6. DADOS TÉCNICOS

Campo	Valor	Descrição
Path Linux	<code>/var/lib/docker/volumes</code>	Local padrão dos volumes
Driver	local	Driver padrão

7. VALIDAÇÃO FINAL

- ☐ O comando ``docker volume ls`` lista o volume criado?
- ☐ Se remover o container (``docker rm -f``), os dados persistem?
- ☐ O container consegue ler/escrever no diretório mapeado?