

Tutorial para rodar containers e importar dados para o banco MySQL usando Docker Compose

Este tutorial vai mostrar como rodar 3 serviços (db, backend e frontend) usando Docker Compose, e depois importar um arquivo de backup chamado `vendinha_dump.sql` para o banco de dados MySQL.

O que você vai precisar:

- **Um computador com o Docker e Docker Compose instalados.** Se você não tem o Docker instalado, você pode baixar <https://www.docker.com/products/docker-desktop/>.
- **O arquivo de backup** `vendinha_dump.sql` (já deve estar na pasta `db` do seu projeto).
- **Conhecimento básico de como abrir o terminal ou prompt de comando.**
 - Estar com o terminal aberto na pasta do projeto.

Parte 1: Executando os contêineres com Docker Compose

1. Execute o Docker Compose

Execute o seguinte comando para subir os contêineres (db, backend e frontend):

```
docker-compose up -d
```

O parâmetro `-d` faz com que os contêineres rodem em segundo plano.

O Docker Compose vai baixar as imagens necessárias e iniciar os contêineres. Isso pode demorar alguns minutos, dependendo da sua conexão de internet.

2. Verifique se os contêineres estão rodando

Para garantir que os contêineres estão rodando corretamente, use o comando:

```
docker ps
```

Você verá uma lista de contêineres em execução.

Procure o contêiner do MySQL (geralmente com o nome `mysql:8.0`).



Guarde o **CONTAINER ID** do contêiner `mysql:8.0`, pois você vai usá-lo depois.

Parte 2: Importando o arquivo de backup

O que é isso?

O arquivo `vendinha_dump.sql` é um backup do seu banco de dados. Ele contém todas as informações e a estrutura do banco, e agora você precisa restaurar essas informações no banco de dados que será criado pelos contêineres gerenciados pelo Docker Compose.

Passo a passo

Agora que o MySQL está rodando em um contêiner, você pode restaurar os dados usando o arquivo `vendinha_dump.sql`.

1. Certifique-se de que o arquivo `vendinha_dump.sql` está na pasta certa

Verifique se o arquivo `vendinha_dump.sql` está na pasta `db` dentro do seu projeto.

2. Copie o arquivo `vendinha_dump.sql` para dentro do contêiner MySQL

Use o comando abaixo para copiar o arquivo de backup para dentro do contêiner MySQL:

```
docker cp ./db/vendinha_dump.sql <CONTAINER_ID>:/tmp
```



Substitua `<CONTAINER_ID>` pelo ID do contêiner MySQL que você pegou anteriormente.

Esse comando vai copiar o arquivo `vendinha_dump.sql` para a pasta `/tmp` dentro do contêiner.

3. Acesse o contêiner MySQL

Agora, você precisa entrar no contêiner MySQL para rodar o comando que vai importar o banco de dados. Execute o seguinte comando:

```
docker exec -it <CONTAINER_ID> /bin/bash
```



Substitua `<CONTAINER_ID>` pelo ID do contêiner MySQL.

Você estará dentro do contêiner, com acesso ao terminal do MySQL.

4. Importe o arquivo de backup

Agora, execute o seguinte comando dentro do contêiner para importar o arquivo de backup:

```
mysql -u root -p vendinha_db < /tmp/vendinha_dump.sql
```

- `u root` : Você está se conectando como o usuário `root`.
- `p` : O MySQL vai pedir a senha. Use a senha configurada no arquivo `docker-compose.yml` :
 - `(yourpassword)`.
- `vendinha_db` : Esse é o banco de dados que já foi criado automaticamente quando o contêiner foi iniciado.

Após rodar o comando, o backup será importado para o banco de dados.

5. Saia do contêiner

Quando o processo de importação terminar, você pode sair do contêiner digitando:

```
exit
```

Parte 3: Testando a aplicação

Agora que o banco de dados está restaurado, você pode testar a aplicação normalmente. Os dados importados estarão disponíveis para uso no sistema.

1. Acesse a aplicação no navegador

Abra seu navegador e acesse o frontend da aplicação no endereço:

```
http://localhost:3000/
```

A aplicação deve estar rodando com os dados restaurados.

Observação

Após importação dos dados pode realizar o login com o usuário abaixo ou criar seu próprio usuário com email diferente.



Email: teste@gmail.com

Senha: 123456789