

## Configuração Inicial

### Remover a necessidade de usar sudo com Docker

Por padrão, é necessário usar sudo para executar comandos Docker. Para evitar isso e facilitar o uso no dia a dia, siga os passos abaixo:

#### 1. Adicione seu usuário ao grupo docker:

```
sudo usermod -aG docker $USER
```

#### 2. Aplique a nova permissão:

Reinicie a sessão (faça logout/login) **ou** execute o comando abaixo para aplicar as permissões imediatamente:

```
newgrp docker
```

#### 3. Teste:

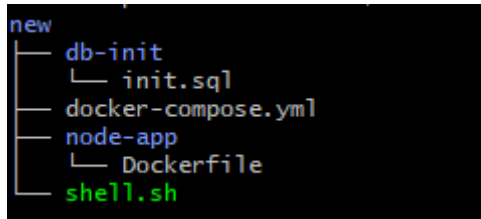
Verifique se o Docker pode ser executado sem sudo:

```
docker ps
```

Se não houver erro, a configuração foi aplicada com sucesso.

EC2

Abrir a porta 3333



```
new
├── db-init
│   └── init.sql
├── docker-compose.yml
├── node-app
│   └── Dockerfile
└── shell.sh
```

### 1. Parar todos os containers em execução

Para parar todos os containers em execução:

```
bash
```

CopiarEditar

```
docker stop $(docker ps -aq)
```

### 2. Remover todos os containers

Depois de parar os containers, você pode removê-los:

bash

CopiarEditar

```
docker rm $(docker ps -aq)
```

### **3. Remover todas as imagens**

Para remover todas as imagens:

bash

CopiarEditar

```
docker rmi $(docker images -q)
```

### **4. Remover todos os volumes**

Se você também deseja remover todos os volumes (que armazenam dados persistentes dos containers):

bash

CopiarEditar

```
docker volume rm $(docker volume ls -q)
```

### **5. Remover todos os networks não utilizados**

Você pode remover redes que não estão mais em uso:

bash

CopiarEditar

```
docker network prune -f
```

### **6. Remover cache de construção**

Para remover o cache de construção (quando você executa o comando `docker build`), você pode usar o comando:

bash

CopiarEditar

```
docker builder prune -a -f
```

### **7. Limpar tudo de uma vez**

Se você deseja remover todas as imagens, containers, volumes e redes não utilizados em um único comando, pode usar o seguinte comando:

bash

CopiarEditar

```
docker system prune -a -f --volumes
```