



# INSTALAÇÃO E CONFIGURAÇÃO DO DOCKER E COMPOSE (LINUX)

Responsável: João Pedro Toledo Gonçalves.

Data: 26/01/2026

**Código:** ITGSUP 0007/26 | **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

Padronizar a instalação do Docker Engine e Docker Compose em servidores Linux (Ubuntu/Debian), garantindo suporte oficial e atualizações seguras.

## 3. PRÉ-REQUISITOS

### NOTA

Liste o que é necessário ANTES de começar.

- [ ] Acesso ao servidor via SSH.
- [ ] Usuário com privilégios de `sudo` ou `root`.
- [ ] Conexão com a internet liberada (Portas 80/443).
- [ ] Sistema Operacional compatível (Ubuntu 20.04+, Debian 11+).

## 4. PASSO A PASSO (EXECUÇÃO)

### Etapa 1: Preparação do Ambiente e Repositórios

1. [x] Atualize o índice de pacotes e instale as dependências essenciais para HTTPS.

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

2. [x] Adicione a chave GPG oficial do Docker para garantir a autenticidade dos pacotes.

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
```

3. [x] Configure o repositório estável.

> **NOTA:** O comando abaixo detecta automaticamente a versão do OS (`lsb\_release`).

```
echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] ht
tps://$(lsb_release -cs)&&echo ubuntu" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

## Etapa 2: Instalação dos Pacotes

1. [x] Atualize novamente o índice (agora com o repo do Docker) e instale o Engine, CLI e plugins.

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker
-compose-plugin
```

## Etapa 3: Pós-Instalação (Permissões)

### IMPORTANTE

**IMPORTANTE:** Execute isso para **não** precisar usar `sudo` em todo comando docker.

1. [x] Adicione seu usuário atual ao grupo `docker`.
2. [x] Aplique a mudança de grupo na sessão atual.

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

## Etapa 4: Validação (Hello World)

1. [x] Execute o container de teste para validar a instalação completa.

```
docker run hello-world
```

2. [x] O resultado esperado deve ser idêntico ao bloco abaixo:

### NOTA

Sucesso na execução do Docker.

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that prints the output string that you see now.

The Docker daemon then streamed the output to the Docker client, which sent it to your terminal.

## 5. SOLUÇÃO DE PROBLEMAS (TROUBLESHOOTING)

### Problema 1: "permission denied while trying to connect to the Docker daemon socket"

• **Causa:** O usuário não está no grupo `docker` ou a sessão não foi atualizada.

• **Solução:**

1. [x] Execute `sudo usermod -aG docker \$USER`.
2. [x] Faça **logout** e login novamente ou use `newgrp docker`.

### Problema 2: Erro de GPG ou Repositório não encontrado

• **Causa:** Chave expirada ou OS não suportado.

• **Solução:**

1. [x] Verifique se o codename do OS em `/etc/os-release` é compatível (focal, jammy, bullseye, bookworm).
2. [x] Remova o arquivo `/etc/apt/sources.list.d/docker.list` e refaça a **Etapa 1**.

## 6. DADOS TÉCNICOS

| Campo   | Valor                                | Descrição                         |
|---------|--------------------------------------|-----------------------------------|
| Serviço | <code>docker.service</code>          | Nome do serviço no Systemd        |
| Config  | <code>/etc/docker/daemon.json</code> | Arquivo principal de configuração |
| Dados   | <code>/var/lib/docker</code>         | Local padrão de imagens e volumes |
| Socket  | <code>/var/run/docker.sock</code>    | Socket de comunicação da API      |

## 7. VALIDAÇÃO FINAL (DEFINIÇÃO DE PRONTO)

- [ ] O comando `docker run hello-world` funciona sem sudo?
- [ ] O comando `docker compose version` retorna a versão (ex: v2.x.x)?
- [ ] O serviço está ativo? (`sudo systemctl status docker` mostra **active (running)**)