

Instalação do Docker

O **Docker** é uma plataforma que permite rodar aplicações dentro de "contêineres" — ambientes isolados e padronizados que funcionam de forma independente do sistema operacional. É através dele que o sistema será executado no servidor.

“ **Pré-requisito:** Você deve estar logado como administrador (`sudo su`). Se ainda não fez isso, execute `sudo su` antes de continuar.

1. Abrindo o Terminal como Administrador

```
sudo su
```

2. Instalar o Docker e o Docker Compose

O comando abaixo instala dois componentes:

```
apt install docker.io -y && \  
apt install docker-compose
```

“ **O que cada um faz:**

- **docker.io** — o Docker em si, responsável por criar e executar os contêineres

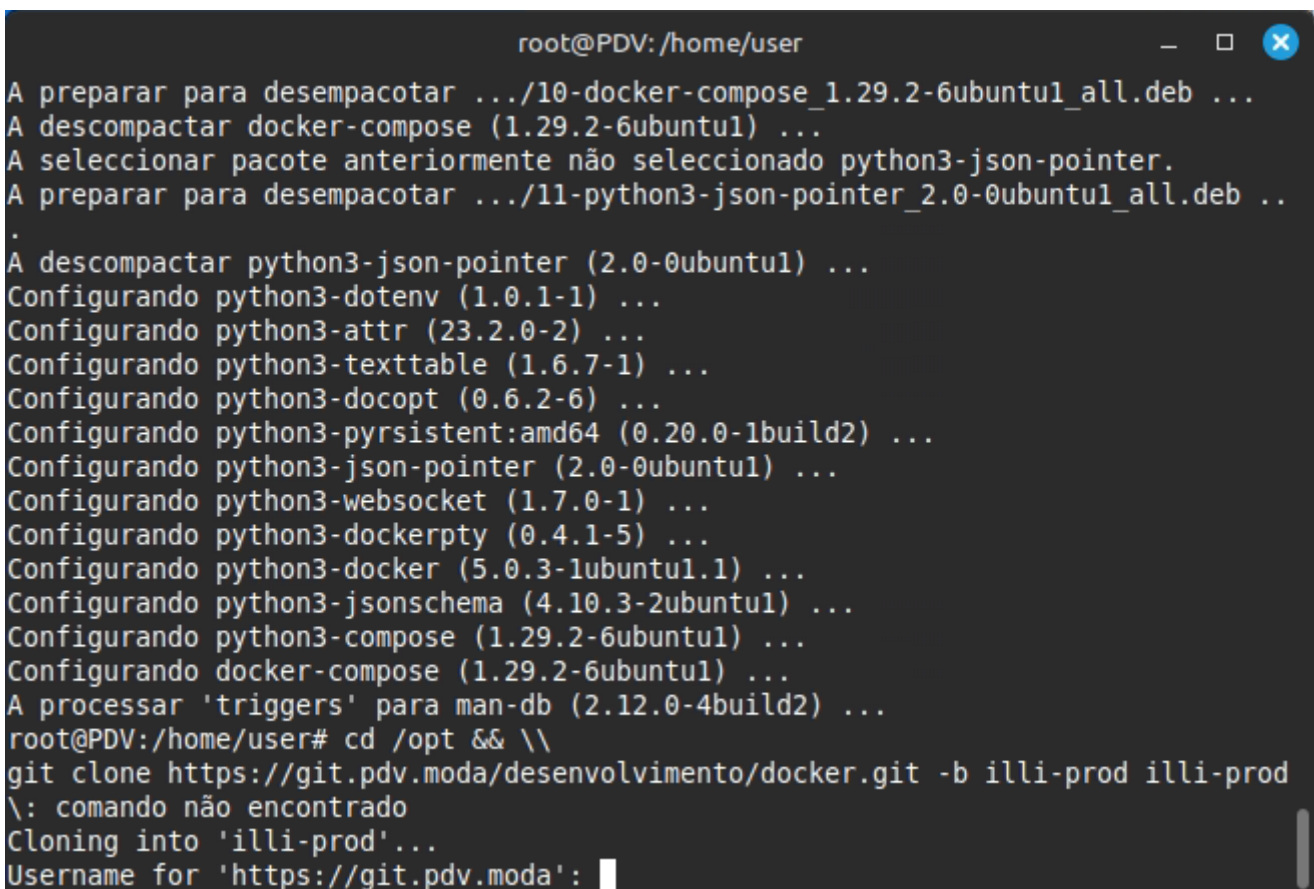
- **docker-compose** — ferramenta que permite iniciar e gerenciar vários contêineres de uma só vez usando um arquivo de configuração

3. Clonar o Repositório de Instalação

Agora vamos baixar os arquivos necessários para configurar o ambiente. Acesse a pasta `/opt` e faça o clone do repositório:

```
cd /opt && \  
git clone https://git.pdv.moda/desenvolvimento/docker.git -b illi-prod illi-prod
```

Quando solicitado, **digite seu usuário e senha** de acesso ao `git.pdv.moda` e pressione **Enter** após cada um.



```
root@PDV: /home/user  
A preparar para desempacotar .../10-docker-compose_1.29.2-6ubuntu1_all.deb ...  
A descompactar docker-compose (1.29.2-6ubuntu1) ...  
A seleccionar pacote anteriormente não seleccionado python3-json-pointer.  
A preparar para desempacotar .../11-python3-json-pointer_2.0-0ubuntu1_all.deb ..  
.  
A descompactar python3-json-pointer (2.0-0ubuntu1) ...  
Configurando python3-dotenv (1.0.1-1) ...  
Configurando python3-attr (23.2.0-2) ...  
Configurando python3-texttable (1.6.7-1) ...  
Configurando python3-docopt (0.6.2-6) ...  
Configurando python3-pyrsistent:amd64 (0.20.0-1build2) ...  
Configurando python3-json-pointer (2.0-0ubuntu1) ...  
Configurando python3-websocket (1.7.0-1) ...  
Configurando python3-dockerpty (0.4.1-5) ...  
Configurando python3-docker (5.0.3-1ubuntu1.1) ...  
Configurando python3-jjsonschema (4.10.3-2ubuntu1) ...  
Configurando python3-compose (1.29.2-6ubuntu1) ...  
Configurando docker-compose (1.29.2-6ubuntu1) ...  
A processar 'triggers' para man-db (2.12.0-4build2) ...  
root@PDV:/home/user# cd /opt && \  
git clone https://git.pdv.moda/desenvolvimento/docker.git -b illi-prod illi-prod  
\: comando não encontrado  
Cloning into 'illi-prod'...  
Username for 'https://git.pdv.moda':
```

O que esse comando faz: Baixa os arquivos de configuração do ambiente Docker a partir do repositório oficial da Illimitar, salvando-os na pasta `illi-prod` dentro de `/opt`.

4. Acessar a Pasta Clonada

Após o clone ser concluído, entre na pasta que foi baixada:

```
cd illi-prod
```

“ Dentro dessa pasta estará o arquivo `illi.sh`, que é o script responsável por instalar e configurar o ambiente.

5. Executar o Script de Instalação

Execute o script com o comando abaixo:

```
bash illi.sh
```

O script fará algumas perguntas para configurar a instância. Responda conforme indicado:

- **Nome da instância:** `illi`
- **Porta do host [8088]:** pressione **Enter** para manter a porta padrão `8088`

“ **O que é uma instância?** É o nome que identifica esta instalação do sistema no servidor. O nome `illi` é o padrão recomendado.

“ **O que é a porta 8088?** É o número pelo qual o sistema será acessado no navegador. Por exemplo: `http://IP-do-servidor:8088`. Mantenha o padrão a menos que essa porta já esteja em uso.

6. Confirmar a Inicialização do Contêiner

Após configurar a instância, o script fará o clone do repositório standalone automaticamente. Ao final, ele perguntará se deseja subir o contêiner:

Responda **S** e pressione **Enter** para iniciar o contêiner.

```
root@PDV: /opt/illi-prod
▶ Criar nova instância illi
  ▲ Será criado um novo diretório com container independente.
  _____
Nome da instância [illi-web]: illi
Porta do host [8088]: 8088
Clonando repositório standalone em www/illi...
Cloning into '/opt/illi-prod/illi/www/illi'...
remote: Enumerating objects: 77493, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 77493 (delta 0), reused 0 (delta 0), pack-reused 77470 (from 1)
Receiving objects: 100% (77493/77493), 444.94 MiB | 4.49 MiB/s, done.
Resolving deltas: 100% (40118/40118), done.
Updating files: 100% (12131/12131), done.
  ✓ Repositório clonado em illi/www/illi
  Definindo permissões no host...
  ✓ Host: chmod +x *.sh | chown www-data:www-data
  _____
  ✓ Instância criada em: /opt/illi-prod/illi
  Container : illi
  Porta     : 8088
  _____
Subir o container agora? [S/n]: █
```

“ O Docker irá baixar as imagens necessárias e iniciar todos os serviços. Esse processo pode levar alguns minutos dependendo da velocidade da conexão.

Digite 0 para sair do utilitário:

```
root@PDV: /opt/illi-prod

8) Saúde e uptime

ACESSO
9) Conectar (bash/sh)
10) Executar comando
11) MySQL (testar / acessar)

MANUTENÇÃO
12) Atualizar imagem
13) Reconstruir container
14) Backup (www + logs)
15) Alterar porta
16) Configurar

LIMPEZA
17) Limpar (dangling/órfãos)
18) Limpar tudo

SISTEMA
19) Instalar comando 'illi' no sistema

0) Sair (containers continuam rodando)

Escolha [0-19]:
```

☐ Instalação Concluída

O Docker está instalado e o contêiner do sistema está em execução. O ambiente estará acessível pelo navegador no endereço `http://IP-do-servidor:8088`.

Próximo Passo

Continue para: **Configuração da Interface**

Revisão #: contagem de revisões

Criado: duração de tempo por usuário

Atualizado: duração de tempo por usuário