

Instalação do Servidor Standalone via DOCKER - Linux Mint 22.1 "Xia" / Ubuntu 24.04 LTS "noble"

- [Configuração Inicial do Servidor](#)
- [Instalação do DWService](#)
- [Instalação do Percona MySQL](#)
- [Instalação do Docker](#)
- [Configuração da Interface](#)

Configuração Inicial do Servidor

Esta etapa prepara o sistema operacional antes de instalar qualquer programa. Siga os passos na ordem indicada.

1. Abrindo o Terminal como Administrador

Para executar os comandos abaixo, você precisa ter permissão de administrador (também chamado de *root*). Digite o comando abaixo e pressione **Enter**. Quando o sistema pedir a senha, digite-a e pressione **Enter** novamente (os caracteres não aparecem na tela — isso é normal).

```
sudo su
```

“ **O que esse comando faz:** Eleva seus privilégios para administrador do sistema, permitindo fazer alterações importantes.

2. Atualizando o Sistema e Instalando Ferramentas Básicas

2.1 – Atualizar os programas instalados e remover o PHP

Execute o bloco de comandos abaixo. Ele vai atualizar todos os programas do sistema, remover o PHP (que não será usado) e instalar o agendador de tarefas `cron`.

```
apt update && apt upgrade && \  
apt remove php* && \  
apt install cron
```

“ Quando aparecer uma pergunta de confirmação, digite `S` e pressione **Enter** para continuar.

2.2 – Instalar ferramentas de monitoramento e utilitários

O comando abaixo instala um conjunto de ferramentas úteis para gerenciar o servidor:

```
apt install ntp ntpdate crudini htop software-properties-common iperf logrotate bmon netcat-traditional mtr -y
```

Para que serve cada ferramenta:

Ferramenta	O que faz
<code>ntp</code> / <code>ntpdate</code>	Mantém o horário do servidor sempre sincronizado com a internet
<code>crudini</code>	Facilita a edição de arquivos de configuração
<code>htop</code>	Mostra em tempo real o uso de CPU e memória do servidor
<code>software-properties-common</code>	Permite adicionar novas fontes de instalação de programas
<code>iperf</code>	Testa a velocidade da rede
<code>logrotate</code>	Gerencia e rotaciona os arquivos de log automaticamente
<code>bmon</code>	Monitora o tráfego de rede em tempo real
<code>netcat-traditional</code>	Ferramenta para testar conexões de rede
<code>mtr</code>	Diagnostica problemas de rota na rede

3. Configurando o Horário do Servidor

O servidor precisa ter o horário correto para funcionar bem. Os comandos abaixo definem o fuso horário como **Brasília (America/Sao_Paulo)** e sincronizam o relógio com servidores de tempo oficiais do Brasil.

```
echo "America/Sao_Paulo" > /etc/timezone && \  
dpkg-reconfigure --frontend noninteractive tzdata
```

```
service ntp stop && \  
ntpdate a.ntp.br
```

```
service ntp start
```

“ O que acontece aqui:

1. O fuso horário é definido para Brasília
2. O serviço de tempo é parado temporariamente
3. O relógio é sincronizado com o servidor `a.ntp.br` (servidor oficial do NIC.br)
4. O serviço de tempo é reiniciado para manter a sincronização contínua

4. Configurando o Idioma do Sistema

Os comandos abaixo definem o idioma do sistema como **Português do Brasil (pt_BR.UTF-8)**:

```
echo "LANG=\"pt_BR.UTF-8\"" >> /etc/environment && \  
echo "LANGUAGE=\"pt_BR.UTF-8:en\"" >> /etc/environment && \  
echo "" > /etc/default/locale
```

```
echo "LANG=\"pt_BR.UTF-8\"" >> /etc/default/locale && \  
echo "LANGUAGE=\"pt_BR.UTF-8:en\"" >> /etc/default/locale && \  

```

```
echo "pt_BR.UTF-8 pt_BR.UTF-8" >> /etc/locale.alias && \  
dpkg-reconfigure --frontend noninteractive locales && \  
locale-gen --no-purge --lang pt_BR
```

“ **O que esse bloco faz:** Configura o sistema para exibir mensagens, datas, formatos numéricos e outros textos em português do Brasil. O inglês (`en`) fica como idioma secundário caso alguma tradução não esteja disponível.

5. Configurando a Segurança

Nesta etapa, dois recursos de segurança padrão do Linux são desativados: o **AppArmor** e o **UFW (firewall)**. Isso é necessário porque o ambiente Docker que será instalado posteriormente possui seu próprio sistema de controle de acesso, e manter esses dois ativos pode causar conflitos.

```
systemctl stop apparmor && \  
systemctl disable apparmor
```

```
/usr/sbin/update-rc.d -f apparmor remove && \  
ufw disable && \  
reboot
```

“ **⚠ Atenção:** O servidor será **reiniciado automaticamente** ao final deste passo (`reboot`). Após a reinicialização, você poderá continuar para a próxima etapa da instalação.

“ **O que é cada recurso desativado:**

- **AppArmor:** Sistema de controle de acesso que restringe o que cada programa pode fazer. Será substituído pelo controle do Docker.
- **UFW (Uncomplicated Firewall):** Gerenciador de firewall padrão do Ubuntu/Linux Mint. Também será gerenciado pelo Docker.

Próximo Passo

Após a reinicialização do servidor, continue para: **Instalação do DWService**

Instalação do DWService

O **DWService** é uma ferramenta de acesso remoto que permite controlar o servidor a distância pelo navegador, sem precisar estar fisicamente na máquina. Pense nele como um "TeamViewer" para servidores Linux.

“ **Pré-requisito:** Você precisará de um **código de instalação**, disponível na sua licença do DWService. Tenha-o em mãos antes de começar.

1. Baixar e Iniciar o Instalador

O comando abaixo faz três coisas em sequência: baixa o instalador, dá permissão para executá-lo e o inicia.

```
wget https://www.dwservice.net/download/dwagent_x86.sh && \  
chmod +x dwagent_x86.sh && \  
./dwagent_x86.sh
```

“ **O que cada parte faz:**

- `wget ...` — Faz o download do instalador direto do site oficial do DWService
- `chmod +x ...` — Dá permissão de execução ao arquivo baixado
- `./dwagent_x86.sh` — Inicia o instalador

2. Respondendo ao Instalador (Passo a Passo)

Após rodar o comando acima, o instalador fará uma série de perguntas no terminal. Siga as respostas indicadas abaixo:

Pergunta 1 — Aceitar os termos de uso

1. Instalar / 2. Executar / 3. Eu não aceito

Opção (3):

☐ **Digite** `1` e pressione **Enter** para aceitar e instalar.

Pergunta 2 — Escolher o local de instalação

Caminho (/usr/share/dwagent):

☐ **Apenas pressione Enter** para usar o caminho padrão sugerido (`/usr/share/dwagent`).

“ O caminho padrão é o local recomendado. Não é necessário alterar.

Pergunta 3 — Confirmar o local de instalação

Você deseja instalar o DWAgent em '/usr/share/dwagent'?

1. Sim / 2. Não

Opção (2):

☐ **Digite** `1` e pressione **Enter** para confirmar.

Pergunta 4 — Forma de configuração do agente

Configurar o agente?

1. Insira o código de instalação / 2. Criar um novo agente

Opção (1):

☐ **Digite** **1** e pressione **Enter** para usar o código da sua licença.

Pergunta 5 — Inserir o código de instalação

Inserir o código de instalação

Código:

☐ **Digite o código de instalação** que está disponível na sua licença e pressione **Enter**.

“ **⚠ Atenção:** O código é único para cada licença. Não compartilhe esse código com outras pessoas.

☐ Instalação Concluída

Após inserir o código corretamente, o DWService será configurado e iniciado automaticamente. O servidor estará disponível para acesso remoto pelo painel do DWService.

Próximo Passo

Continue para: [Instalação do Percona MySQL](#)

Instalação do Percona MySQL

O **Percona MySQL** é um banco de dados — o programa responsável por armazenar e organizar todos os dados do sistema (cadastros, configurações, registros, etc.). É uma versão aprimorada do MySQL tradicional, com melhor desempenho e confiabilidade.

“ **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. Preparando o Sistema para a Instalação

O Percona MySQL exige uma dependência chamada `percona-telemetry-agent`. Como ela não está disponível diretamente nos repositórios do Ubuntu/Linux Mint, vamos criar um "pacote fictício" para satisfazer esse requisito sem instalar o agente de telemetria real.

2.1 – Instalar o utilitário `equivs`

O `equivs` é uma ferramenta que permite criar pacotes fictícios para resolver dependências.

```
apt install -y equivs && \  
equivs-control percona-telemetry-agent
```

2.2 – Criar o pacote fictício

O bloco abaixo cria um arquivo de configuração descrevendo o pacote fictício:

```
cat << 'EOF' > percona-telemetry-agent  
Section: misc  
Priority: optional  
Standards-Version: 3.9.2  
  
Package: percona-telemetry-agent  
Version: 1.0  
Maintainer: local <root@localhost>  
Description: Dummy package to satisfy Percona dependency  
EOF
```

“ **Por que fazer isso?** O instalador do Percona exige esse componente de telemetria, mas ele não é necessário para o funcionamento do banco de dados. Criando um pacote fictício, enganamos o instalador de forma segura e evitamos instalar software desnecessário.

3. Instalar o Percona MySQL e Configurar a Segurança

O comando abaixo faz tudo em sequência: constrói o pacote fictício, o instala, adiciona o repositório oficial do Percona, atualiza a lista de pacotes, instala o banco de dados e inicia o assistente de segurança.

“ **Atenção:** Antes de exibir o assistente de segurança, o instalador mostrará uma tela pedindo uma senha padrão para o MySQL. Digite a senha desejada e confirme-a quando solicitado.

```
equivs-build percona-telemetry-agent && \  
dpkg -i percona-telemetry-agent_1.0_all.deb && \  
mkdir -p /etc/apt/keyrings && \  
curl -fsSL https://github.com/percona/percona-repositories/raw/main/deb/percona-keyring.gpg | gpg --dearmor -o  
/etc/apt/keyrings/percona.gpg && \  
echo "deb [signed-by=/etc/apt/keyrings/percona.gpg] http://repo.percona.com/ps-84-lts/apt noble main" | tee  
/etc/apt/sources.list.d/percona-ps-84-lts.list && \  
apt update && apt install -y percona-server-server && \  
mysql_secure_installation
```

“ O que esse bloco faz, passo a passo:

1. Monta e instala o pacote fictício criado anteriormente
2. Cria a pasta para guardar a chave de segurança do repositório Percona
3. Baixa e registra a chave oficial do Percona (garante que os pacotes são autênticos)
4. Adiciona o repositório oficial do Percona ao sistema
5. Atualiza a lista de pacotes disponíveis
6. Instala o Percona MySQL
7. Abre o assistente de segurança `mysql_secure_installation`

4. Respondendo ao Assistente de Segurança (`mysql_secure_installation`)

Após a instalação, o terminal exibirá uma série de perguntas para proteger o banco de dados. Siga as respostas indicadas abaixo:

- Enter password for user root: **digite a senha temporária gerada durante a instalação**
- Press y|Y for Yes, any other key for No: **Y**
- There are three levels of password validation policy: **0**
- Change the password for root? (Press y|Y for Yes, any other key for No): **Y**
- New password: **digite a nova senha**
- Re-enter new password: **repita a nova senha**
- Do you wish to continue with the password provided? (Press y|Y for Yes, any other key for No): **Y**
- Remove anonymous users? (Press y|Y for Yes, any other key for No): **Y**
- Disallow root login remotely? (Press y|Y for Yes, any other key for No): **No**

- Remove test database and access to it? (Press y|Y for Yes, any other key for No): **Y**
- Reload privilege tables now? (Press y|Y for Yes, any other key for No): **Y**

“ **⚠ Importante:** Escolha uma senha forte para o root e guarde-a em local seguro. Você precisará dela nas próximas etapas.

5. Instalar o `crudini` (se ainda não estiver instalado)

```
apt install crudini
```

“ O `crudini` é uma ferramenta para editar arquivos de configuração de forma simples e segura, sem precisar abrir o arquivo manualmente.

6. Ajustar as Configurações do MySQL

Os comandos abaixo ajustam o comportamento do MySQL para funcionar corretamente com o sistema. Essas configurações controlam coisas como o tamanho máximo de dados que podem ser enviados de uma vez, quais endereços o banco aceita conexões e como os backups são feitos.

```
crudini --set /etc/mysql/mysql.conf.d/mysqld.cnf mysqld sql_mode  
STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTI  
TUTION && \  
crudini --set /etc/mysql/mysql.conf.d/mysqld.cnf mysqld max_allowed_packet 64M && \  
crudini --set /etc/mysql/mysql.conf.d/mysqld.cnf mysqld bind-address 0.0.0.0 && \  
crudini --set /etc/mysql/mysql.conf.d/mysqld.cnf mysqld skip-external-locking 1 && \  
crudini --set /etc/mysql/mysql.conf.d/mysqld.cnf mysqld performance_schema 1 && \  
crudini --set /etc/mysql/mysql.conf.d/mysqldump.cnf mysqldump quick 1 && \  
crudini --set /etc/mysql/mysql.conf.d/mysqldump.cnf mysqldump quote-names 1 && \  

```

```
crudini --set /etc/mysql/mysql.conf.d/mysqldump.cnf mysqldump max-allowed-packet 128M
```

O que cada configuração faz:

Configuração	O que define
<code>sql_mode</code>	Regras de validação de dados — impede que dados inválidos (como datas zeradas) sejam salvos
<code>max_allowed_packet 64M</code>	Tamanho máximo de uma requisição ao banco (64 MB)
<code>bind-address 0.0.0.0</code>	Permite conexões de qualquer endereço de rede (necessário para o Docker)
<code>skip-external-locking</code>	Desativa bloqueio externo de arquivos (melhora desempenho)
<code>performance_schema</code>	Ativa o monitoramento interno de desempenho do MySQL
<code>quick</code> (dump)	Exporta os dados linha a linha, economizando memória em backups
<code>quote-names</code> (dump)	Coloca aspas nos nomes de tabelas e colunas nos backups, evitando erros
<code>max-allowed-packet 128M</code> (dump)	Tamanho máximo de dados em backups (128 MB)

7. Corrigir Formatação das Configurações

Este comando garante que as configurações inseridas fiquem no formato correto dentro dos arquivos:

```
sed -i -e "s/.*skip-external-locking.*/skip-external-locking/" /etc/mysql/mysql.conf.d/mysqld.cnf && \  
sed -i -e "s/.*performance_schema.*/performance_schema/" /etc/mysql/mysql.conf.d/mysqld.cnf && \  
sed -i -e "s/.*quick.*/quick/" /etc/mysql/mysql.conf.d/mysqldump.cnf && \  
sed -i -e "s/.*quote-names.*/quote-names/" /etc/mysql/mysql.conf.d/mysqldump.cnf
```

8. Permitir Acesso Remoto ao Usuário Root

Por padrão, o usuário `root` do MySQL só pode se conectar localmente. Os passos abaixo alteram essa configuração para permitir conexões de qualquer lugar — necessário para que o Docker acesse o banco.

8.1 – Ativar o método de autenticação nativo

Abra o arquivo de configuração do MySQL:

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Localize a seção `[mysqld]` e adicione a linha abaixo ao final dela:

```
mysql_native_password=ON
```

Salve o arquivo pressionando **Ctrl + O**, depois **Enter**, e saia com **Ctrl + X**.

“ **Por que isso é necessário?** O Percona MySQL 8.4 usa por padrão um método de autenticação mais novo (`caching_sha2_password`) que pode ser incompatível com alguns clientes. Ativar o método nativo garante compatibilidade com o sistema.

8.2 – Liberar o acesso remoto do root via linha de comando

Execute o comando abaixo. Quando solicitado, **digite a senha do MySQL** (a senha nova que você definiu no passo 4):

```
mysql -h localhost -u root -p mysql -e "UPDATE user SET Host = '%' WHERE user.Host = 'localhost' AND user.User = 'root' LIMIT 1;"
```

8.3 – Atualizar o método de autenticação do root dentro do banco

Acesse o MySQL interativamente:

```
mysql -u root -p
```

Quando solicitado, **digite a senha do MySQL** e execute os comandos abaixo, substituindo `SENHA` pela senha que você definiu para o banco:

```
ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'SENHA';  
FLUSH PRIVILEGES;
```

“ O que esses comandos fazem:

- `ALTER USER` — altera o usuário root para usar o método de autenticação nativo com a senha definida
- `FLUSH PRIVILEGES` — aplica as alterações de permissão imediatamente, sem precisar reiniciar o serviço

Digite `exit` e pressione **Enter** para sair do MySQL.

9. Reiniciar o MySQL para Aplicar as Configurações

```
service mysql restart
```

“ Após reiniciar, todas as configurações entram em vigor e o banco de dados estará pronto para uso.

☐ Instalação Concluída

O Percona MySQL está instalado, configurado e em execução.

Próximo Passo

Continue para: **Instalação do Docker**

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:**

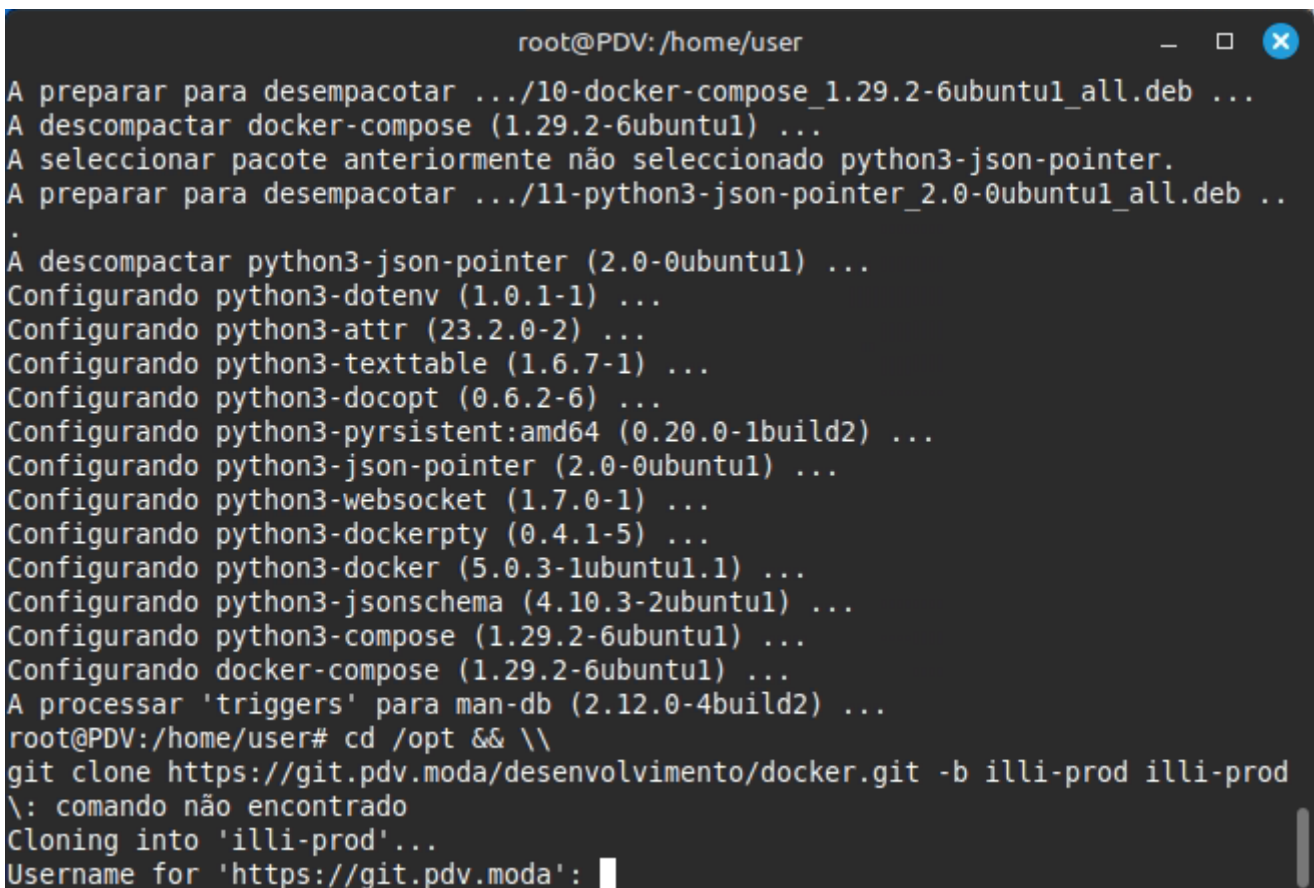
- **docker.io** — o Docker em si, responsável por criar e executar os contêineres
- **docker-compose** — ferramenta que permite iniciar e gerenciar vários contêineres de uma só vez usando um arquivo de configuração

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**

Configuração da Interface

Com o Docker em execução, a próxima etapa é acessar a interface do sistema pelo navegador e informar os dados de conexão com o banco de dados. É nessa tela que o sistema "aprende" onde está o MySQL e como se conectar a ele.

1. Acessar a Interface no Navegador

Abra o navegador e acesse o endereço abaixo, substituindo `IP-DO-SERVIDOR` pelo IP da máquina onde o Docker foi instalado:

```
http://IP-DO-SERVIDOR:8088
```

“ Se estiver acessando direto na máquina do servidor, use `http://localhost:8088` .

Você verá uma tela de configuração inicial como esta:



CONFIGURAÇÃO DE ACESSO

MÚLTIPLOS
DOMÍNIOS

DOMÍNIO

CONFIGURAÇÃO DE BANCO DE DADOS (MYSQL)

SERVIDOR

PORTA

USUÁRIO

SENHA

NOME DA BASE DE DADOS

BINÁRIO

CERTIFICADO SSL

Configurar



2. Preencher os Campos de Configuração

Preencha cada campo conforme indicado abaixo:

Campo	O que preencher
Múltiplos domínios	Deixar desmarcado
Domínio	O nome da instância criada no Docker (ex: <input type="text" value="illi"/>) ou <input type="text" value="localhost:8088"/>
Servidor MySQL	<input type="text" value="host.docker.internal"/>
Porta	<input type="text" value="3306"/>
Usuário	<input type="text" value="root"/>
Senha	A senha do usuário root definida durante a instalação do Percona MySQL

Campo	O que preencher
Nome da base de dados	illi
Binário	/usr/bin
Certificado SSL	Deixar em branco

“ O que é o `host.docker.internal` ? É um endereço especial que o Docker usa para se comunicar com o sistema operacional da máquina onde ele está rodando. Ao informar esse endereço como servidor MySQL, o contêiner Docker consegue se conectar ao Percona MySQL instalado diretamente no servidor.

3. Configurar a Execução Automática de Tarefas (Cron)

O sistema precisa executar tarefas programadas automaticamente em segundo plano (como processar filas, enviar notificações, etc.). Isso é feito através do **cron**, um agendador de tarefas do Linux.

Abra o editor do cron com o comando:

```
crontab -e
```

“ Se for a primeira vez que você abre o `crontab -e`, o sistema perguntará qual editor de texto usar. Escolha a opção **nano** (geralmente a opção `1`) e pressione **Enter** — é o mais simples de usar.

Cole a linha abaixo **no final do arquivo**:

```
***** docker exec -u apache illi /usr/bin/php /var/www/illi/index.php cron executar
```

“ O que essa linha faz? Os cinco asteriscos (`*****`) significam "executar a cada minuto". O restante do comando acessa o contêiner Docker chamado `illi` e roda o processo de tarefas agendadas do sistema.

Após colar, salve e feche o arquivo:

- **CTRL + O** → salva o arquivo
 - **Enter** → confirma o nome do arquivo
 - **CTRL + X** → fecha o editor
-

☐ Configuração Concluída

A interface está configurada e as tarefas automáticas estão ativas. O sistema está pronto para uso.