



' Guia rápido do WSL2 + Docker

' O que é o WSL2

Em 2016, a Microsoft anunciou a possibilidade de rodar o Linux dentro do Windows 10 como um subsistema e o nome a isto foi dado de **WSL** ou **Windows Subsystem for Linux**.

O acesso ao sistema de arquivos no Windows 10 pelo Linux era simples e rápido, porém não tínhamos uma execução completa do kernel do Linux, além de outros artefatos nativos e isto impossibilitava a execução de várias tarefas no Linux, uma delas é o Docker.

Em 2019, a Microsoft anunciou o **WSL 2**, com uma dinâmica aprimorada em relação a 1ª versão:

- Execução do kernel completo do Linux.
- Melhor desempenho para acesso aos arquivos dentro do Linux.
- Compatibilidade completa de chamada do sistema.

O WSL 2 foi lançado oficialmente no dia 28 de maio de 2020.

Com WSL 2 é possível executar Docker no Linux usando o Windows 10/11.

Compare as versões: <https://docs.microsoft.com/pt-br/windows/wsl/compare-versions>

' O que é Docker

Docker é uma plataforma open source que possibilita o empacotamento de uma aplicação dentro de um container. Uma aplicação consegue se adequar e rodar em qualquer máquina que tenha essa tecnologia instalada.

' Porque usar WSL 2 + Docker para desenvolvimento

Configurar ambientes de desenvolvimento no Windows sempre foi burocrático e complexo, além do desempenho de algumas ferramentas não serem totalmente satisfatórias.

Com o nascimento do Docker este cenário melhorou bastante, pois podemos montar nosso ambiente de desenvolvimento baseado em Unix, de forma independente e rápida, e ainda unificada com outros sistemas operacionais.

Veja nossa live sobre **WSL 2 + Docker** no canal Full Cycle: <https://www.youtube.com/watch?v=usF0rYCcj-E>.

› Modos de usar Docker no Windows

- Docker Toolbox.
- Docker Desktop com Hyper-V.
- Docker Desktop com WSL2.
- Docker Engine (Docker Nativo) diretamente instalado no WSL2.

› Docker Toolbox

Roda em cima do programa de virtualização de sistemas da Oracle, chamado de **VirtualBox**. O desempenho do Docker Toolbox para muitas aplicações/ferramentas pode ser muito ruim, inviabilizando seu uso.

› Docker Desktop com Hyper-V

Roda em cima do **Hyper-V** da Microsoft em vez de usar o VirtualBox usando pelo Docker Toolbox. O Docker Desktop com Hyper-V necessita da versão **PRO** do Windows 10/11, portanto é necessário compra-la se você não a tem.

O Hyper-V costuma requerer muitos recursos da máquina e apesar do desempenho ser melhor que o Docker Toolbox, a máquina pode ficar lenta para se utilizar outras coisas no Windows.

A Docker já anunciou que vai remover o suporte ao Hyper-V futuramente.

› Docker Desktop com WSL2

Roda em cima do **Virtual Machine Platform** em vez de usar o VirtualBox ou Hyper-V. Se integra com o WSL2 permitindo rodar o Docker dentro do ambiente do Linux. Não é necessário adquirir licença PRO do Windows 10/11, tem um grande desempenho e consome menos recursos quando comparado ao Docker Toolbox ou Docker Desktop com Hyper-V.

Temos a grande vantagem de se trabalhar totalmente dentro do Linux para desenvolvimento, portanto, usar WSL2 + Docker é a melhor maneira de se desenvolver aplicações no Windows.

› Vantagens

- Simplifica a configuração do Docker tanto no Windows quanto no WSL 2.
- Permite rodar o Docker fora do WSL 2. É possível usar qualquer shell como PowerShell ou DOS.
- Suporta containers em modo Windows (Imagens que contém Windows por debaixo dos panos ao invés de Linux).
- Cria um ambiente centralizado para armazenamento de imagens, volumes e outros configurações Docker. Pode-se ter várias distribuições do WSL 2 rodando o mesmo Docker.
- Interface visual para administrar o Docker.

› Desvantagens

- Uso de memória inicial sem rodar nenhum container Docker pode chegar a 3GB.

- Adiciona infraestrutura complexa para executar Docker, quando se necessita apenas de rodar os containers Docker dentro de um WSL 2 apenas.

› Docker Engine (Docker Nativo) diretamente instalado no WSL2.

O Docker Engine é o Docker nativo que roda no ambiente Linux e completamente suportado para WSL 2. Sua instalação é idêntica a descrita para as próprias distribuições Linux disponibilizadas no site do Docker.

› Vantagens

- Consume o mínimo de memória necessário para rodar o Docker Daemon (servidor do Docker).
- É mais rápido ainda que com Docker Desktop, porque roda diretamente dentro da própria instância do WSL2 e não em uma instância separada de Linux.

› Desvantagens

- Necessário executar o comando `sudo service docker start` sempre que o WSL 2 foi reiniciado. Isto não é necessariamente uma desvantagem, mas é bom pontuar, mas isto é um pequeno detalhe e será resolvido futuramente com a inclusão do arquivo `/etc/wsl.conf` que permitirá incluir comandos para serem executados toda vez que o WSL for reiniciado.
- Se necessitar executar Docker em outra instância do WSL 2, é necessário instalar novamente o Docker nesta instância ou configurar o acesso ao socket do Docker desejado para compartilhar o Docker entre as instâncias.
- Não suporta containers no modo Windows.

› Requisitos mínimos

- Windows 10 Home ou Professional
 - Versão 1903 ou superior para sistemas x64, com Build 18362 ou superior.
 - Versão 2004 ou superior para sistemas ARM64, com Build 19041 ou superior.
- Windows 11 Home ou Professional
 - Versão 22000 ou superior.

Provavelmente seu Windows já está na versão suportada, mas verifique isto acessando o menu de notificações perto do relógio > Todas as configurações > Sistema > Sobre . Caso não esteja, use o Assistente do Windows Update para atualizar a sua versão do Windows.

- Uma máquina compatível com virtualização (verifique a disponibilidade de acordo com a marca do seu processador. Se sua máquina for mais antiga pode ser necessária habilitá-la na BIOS).
- Pelo menos 4GB de memória RAM.

Instalação do WSL 2

Windows 10

Habilitar o Virtual Machine Platform

Execute os seguintes comandos no PowerShell em modo administrador:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart  
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```



Abra o PowerShell e digite o comando `wsl`, se não funcionar reinicie sua máquina.

Instalar o executável do WSL

Baixe o Kernel do WSL 2 neste link: <https://docs.microsoft.com/pt-br/windows/wsl/wsl2-kernel> e instale o pacote.

Atribuir a versão default do WSL para a versão 2

A versão 1 do WSL é a padrão no momento, atribua a versão default para a versão 2, assim todas as distribuições Linux instaladas serão já por default da versão 2. Execute o comando com o PowerShell:

```
wsl --set-default-version 2
```

(Opcional) Alterar a versão do WSL 1 de uma distribuição para a versão 2

Se você já tiver o WSL 1 na máquina e acabou de instalar a versão 2, pode-se converter sua distribuição Linux WSL 1 para WSL 2, execute o comando com o PowerShell:

```
wsl --set-version <distribution name> 2
```

Windows 11

Execute o comando:

```
wsl --install
```

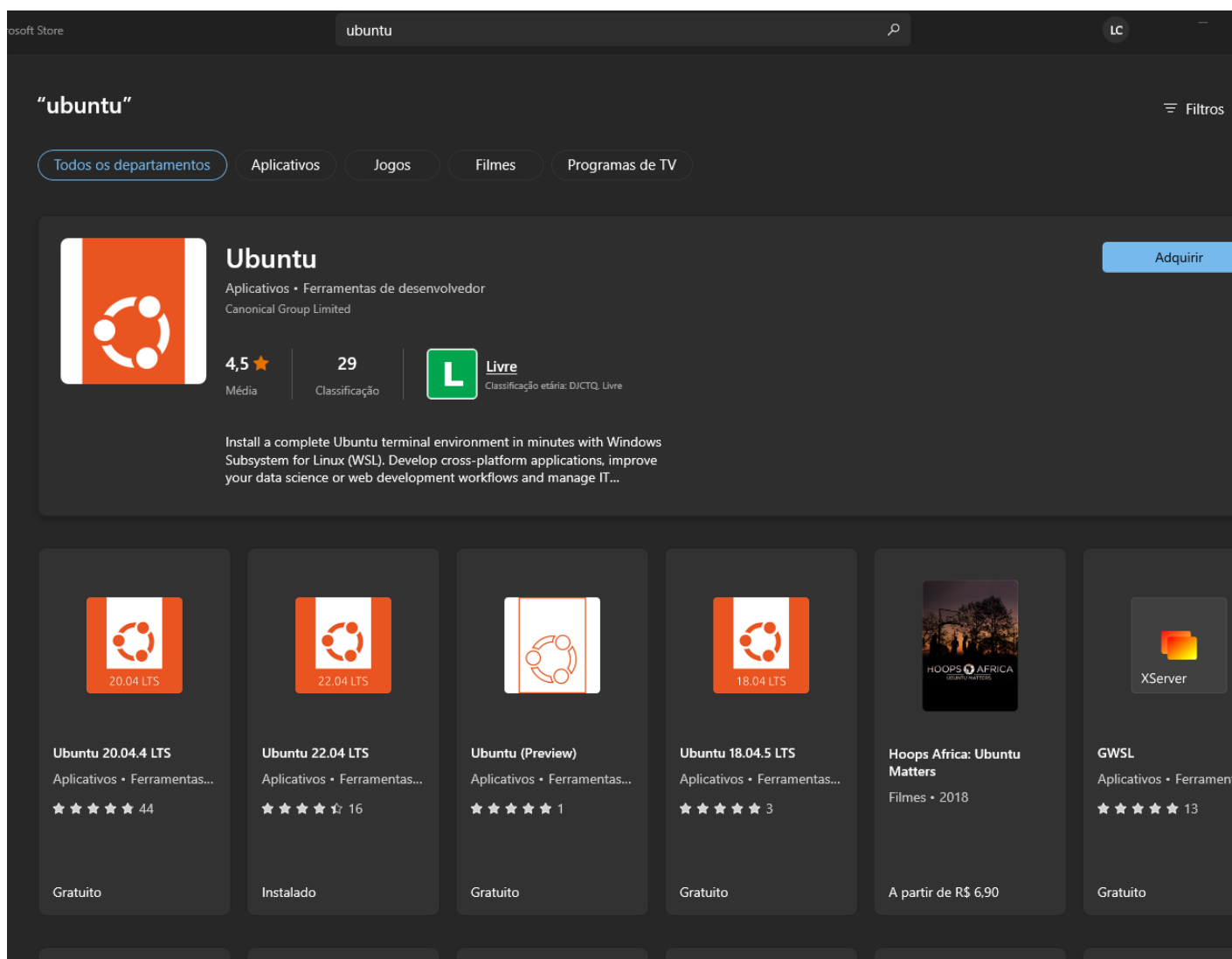
Este comando irá instalar todas as dependências do WSL instalando o `ubuntu` como o Linux padrão.

Se você quiser instalar uma versão diferente do Ubuntu, execute o comando `wsl -l -o`, será listado todas as versões de Linux disponíveis. Instale a versão escolhida com o comando `wsl --install -d nome-da-distribuicao`.

sugerimos o Ubuntu (sem versão) por ser uma distribuição popular e que já vem com várias ferramentas instaladas por padrão.

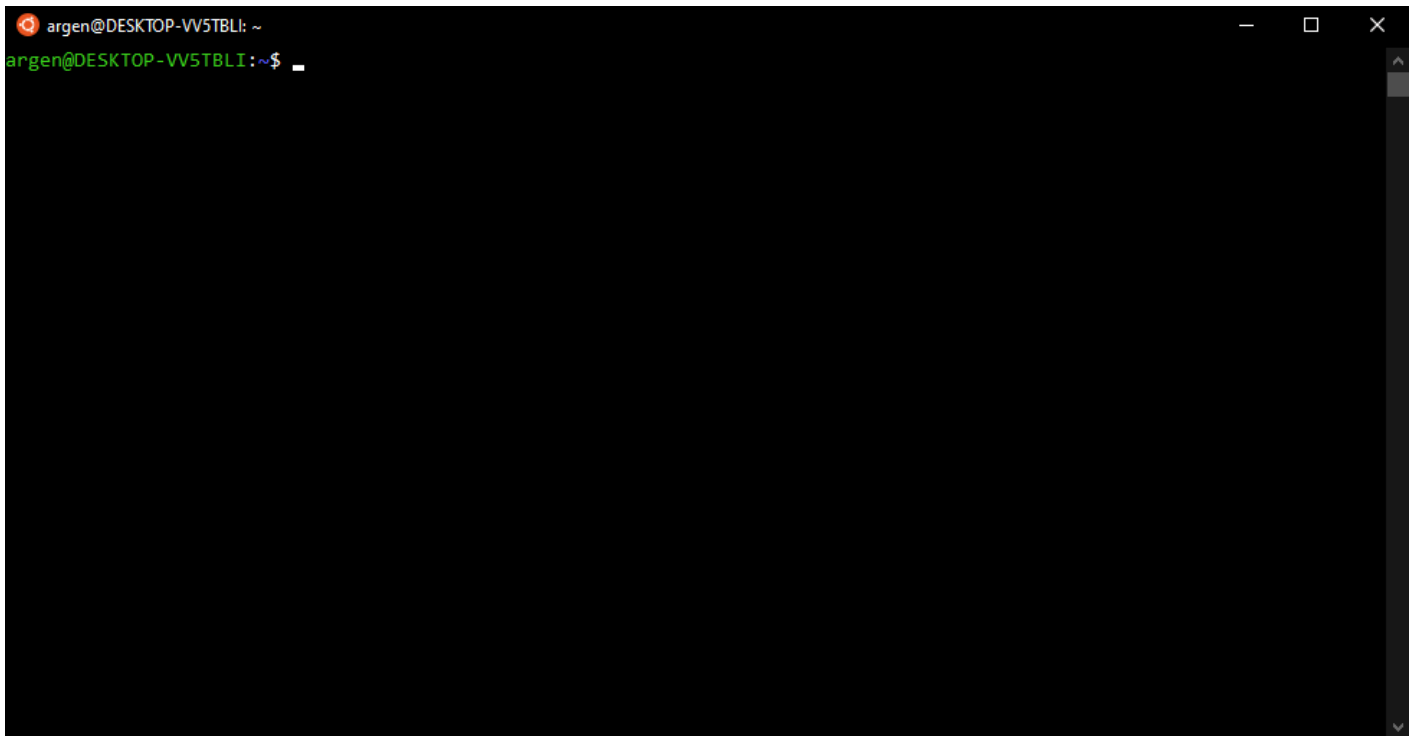
Escolha sua distribuição Linux no Windows Store

Também é possível instalar distribuições Linux pelo Windows Store. Escolha sua distribuição Linux preferida no aplicativo Windows Store, sugerimos o Ubuntu (sem versão) por ser uma distribuição popular e que já vem com várias ferramentas instaladas por padrão.



Ao iniciar o Linux instalado, você deverá criar um **nome de usuário** que poderá ser o mesmo da sua máquina e uma **senha**, este será o usuário **root** da sua instância **WSL**.

Parabéns, seu WSL2 já está funcionando:



’ (Opcional) Usar Windows Terminal como terminal padrão de desenvolvimento para Windows

Uma deficiência que o Windows sempre teve era prover um terminal adequado para desenvolvimento. Agora temos o **Windows Terminal** construído pela própria Microsoft que permite rodar terminais em abas, alterar cores e temas, configurar atalhos e muito mais.

Instale-o pelo Windows Store e use estas configurações padrões para habilitar WSL 2, Git Bash e o tema dracula e alguns atalhos.

Para sobrescrever as configurações **clique a seta para baixo do lado das abas e em configurações**, abrirá as configurações do Windows Terminal, apenas cole o conteúdo do arquivo JSON e salve, após isso clique em **Ubuntu** na seção **Perfis**, clique sobre **Diretório inicial** e altere o caminho para: `(\\wsl$\\Ubuntu\\home\\SEU_USUÁRIO_UBUNTU)`.

’ O que o WSL 2 pode usar de recursos da sua máquina

Podemos dizer que o WSL 2 tem acesso quase que total ao recursos de sua máquina. Ele tem acesso por padrão:

- A todo disco rígido.
- A usar completamente os recursos de processamento.
- A usar 80% da memória RAM disponível.
- A usar 25% da memória disponível para SWAP.

Isto pode não ser interessante, uma vez que o WSL 2 pode usar praticamente todos os recursos de sua máquina, mas podemos configurar limites.

Crie um arquivo chamado `.wslconfig` na raiz da sua pasta de usuário (`C:\\Users\\<seu_usuario>`) e defina estas configurações:

```
[ws12]  
memory=8GB  
processors=4  
swap=2GB
```

Estes são limites de exemplo e as configurações mais básicas a serem utilizadas, configure-os às suas disponibilidades. Para mais detalhes veja esta documentação da Microsoft: <https://docs.microsoft.com/pt-br/windows/wsl/wsl-config#wsl-2-settings>.

Para aplicar estas configurações é necessário reiniciar as distribuições Linux, então sugerimos executar no PowerShell o comando: `wsl --shutdown` (Este comando vai desligar todas as instâncias WSL 2 ativas e basta abrir o terminal novamente para usa-la já com as novas configurações).

Integrar Docker com WSL 2

No início deste tutorial vimos 4 modos de usar Docker no Windows, mas somente 2 que recomendamos:

- Docker Engine (Docker Nativo) diretamente instalado no WSL2.
- Docker Desktop com WSL2.

Recomendamos que escolha a 1ª opção pelos seus benefícios, já que a maioria das pessoas poderão usar o WSL 2 como ferramenta central para desenvolvimento. Mas, neste tutorial vamos mostrar as duas forma de instalação.

1 - Instalar o Docker com Docker Engine (Docker Nativo)

A instalação do Docker no WSL 2 é idêntica a instalação do Docker em sua própria distribuição Linux, portanto se você tem o Ubuntu é igual ao Ubuntu, se é Fedora é igual ao Fedora. A documentação de instalação do Docker no Linux por distribuição está aqui, mas vamos ver como instalar no Ubuntu.

Quem está migrando de Docker Desktop para Docker Engine, temos duas opções

1. Desinstalar o Docker Desktop.
2. Desativar o Docker Desktop Service nos serviços do Windows. Esta opção permite que você utilize o Docker Desktop, se necessário, para a maioria dos usuários a desinstalação do Docker Desktop é a mais recomendada. Se você escolheu a 2ª opção, precisará excluir o arquivo `~/.docker/config.json` e realizar a autenticação com Docker novamente através do comando "docker login"

Se necessitar integrar o Docker com outras IDEs que não sejam o VSCode

O VSCode já se integra com o Docker no WSL desta forma através da extensão Remote WSL ou Remote Container.

É necessário habilitar a conexão ao servidor do Docker via TCP. Vamos aos passos:

1. Crie o arquivo `/etc/docker/daemon.json`: `sudo echo '{"hosts": ["tcp://0.0.0.0:2375", "unix:///var/run/docker.sock"]}' > /etc/docker/daemon.json`
2. Reinicie o Docker: `sudo service docker restart`

Após este procedimento, vá na sua IDE e para conectar ao Docker escolha a opção TCP Socket e coloque a URL `http://IP-DO-WSL:2375`. Seu IP do WSL pode ser encontrado com o comando `cat /etc/resolv.conf`.

Se caso não funcionar, reinicie o WSL com o comando `wsl --shutdown` e inicie o serviço do Docker novamente.

Instale os pré-requisitos:

```
sudo apt update && sudo apt upgrade
sudo apt remove docker docker-engine docker.io containerd runc
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

Adicione o repositório do Docker na lista de sources do Ubuntu:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo \
    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Instale o Docker Engine

```
sudo apt-get update
```

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

Dê permissão para rodar o Docker com seu usuário corrente:

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

Inicie o serviço do Docker:

```
sudo service docker start
```


Este comando acima terá que ser executado toda vez que Linux for reiniciado. Se caso o serviço do Docker não estiver executando, mostrará esta mensagem de erro ao rodar comando `docker` :

```
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
```

O Docker Compose instalado agora estará na versão 2, para executa-lo em vez de `docker-compose` use `docker compose` .

’ Erro ao iniciar o Docker no Ubuntu 22.04

Se mesmo ao iniciar o serviço do Docker acontecer o seguinte erro ou similar:

```
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running? Rode o comando sudo update-alternatives --config iptables e escolha a opção 1 iptables-legacy
```

Rode novamente o `sudo service docker start` . Rode algum comando Docker como `docker ps` para verificar se está funcionando corretamente. Se não mostrar o erro acima, está ok.

’ Dica para Windows 11

No Windows 11 é possível especificar um comando padrão para ser executados sempre que o WSL for iniciado, isto permite que já coloquemos o serviço do docker para iniciar automaticamente. Edite o arquivo `/etc/wsl.conf` :

Rode o comando para editar:

```
sudo vim /etc/wsl.conf
```

Aperte a letra `i` (para entrar no modo de inserção de conteúdo) e cole o conteúdo:

```
[boot]
command="service docker start"
```

Quando terminar a edição, pressione `Esc` , em seguida tecle `:` para entrar com o comando `wq` (salvar e sair) e pressione `enter` . Pronto, para reiniciar o WSL com o comando `wsl -shutdown` no DOS ou PowerShell para testar. Após abrir o WSL novamente, digite o comando `docker ps` para avaliar se o comando não retorna a mensagem acima: `Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?`

’ 2 - Instalar o Docker com Docker Desktop

Baixe neste link: <https://hub.docker.com/editions/community/docker-ce-desktop-windows> e instale o Docker Desktop.

Clique no ícone do Docker perto do relógio -> Settings -> Settings no topo -> Resources -> WSL Integration .

Habilite `Enable integration with my default WSL distro` e habilite sua versão Linux.

```

argen@DESKTOP-VV5TBLI: ~
argen@DESKTOP-VV5TBLI:~$ docker version
Client: Docker Engine - Community
Version: 19.03.8
API version: 1.40
Go version: go1.12.17
Git commit: afacb8b7f0
Built: Wed Mar 11 01:25:46 2020
OS/Arch: linux/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version: 19.03.8
API version: 1.40 (minimum version 1.12)
Go version: go1.12.17
Git commit: afacb8b
Built: Wed Mar 11 01:29:16 2020
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: v1.2.13
GitCommit: 7ad184331fa3e55e52b890ea95e65ba581ae3429
runc:
Version: 1.0.0-rc10
GitCommit: dc9208a3303feef5b3839f4323d9beeb36df0a9dd
docker-init:
Version: 0.18.0
GitCommit: fec3683
argen@DESKTOP-VV5TBLI:~$

```

› Dicas e truques básicos com WSL 2

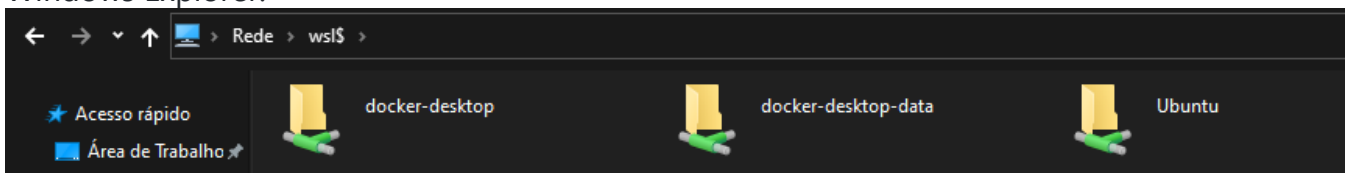
- A performance do WSL 2 está em se executar tudo dentro do Linux, por isso evite executar seus projetos com ou sem Docker do caminho `/mnt/c`, pois você perderá performance.
- Para abrir o terminal do WSL basta digitar o nome da distribuição no menu Iniciar ou executar `C:\Windows\System32\wsl.exe`.
- O sistema de arquivos do Windows 10/11 é acessível em `/mnt`.

```

argen@DESKTOP-VV5TBLI: /mnt
argen@DESKTOP-VV5TBLI:/mnt$ ls
WSL
argen@DESKTOP-VV5TBLI:/mnt$

```

- É possível acessar o sistema de arquivos do Linux pela rede do Windows, digite `\\ws1$` no Windows Explorer.



- É possível acessar uma pasta no Windows Explorer digitando o comando `explorer.exe .`
- É possível abrir uma pasta ou arquivo com o Visual Studio Code digitando o comando `code .` ou `code meu_arquivo.ext`.
- Incrivelmente é possível acessar executáveis do Windows no terminal do Linux executando-os com `.exe` no final (não significa que funcionarão corretamente).

```

argen@DESKTOP-VV5TBLI: ~
argen@DESKTOP-VV5TBLI:~$ php.exe -v
PHP 7.3.4 (cli) (built: Apr 2 2019 21:57:22) ( NTS MSVC15 (Visual C++ 2017) x64 )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.4, Copyright (c) 1998-2018 Zend Technologies
argen@DESKTOP-VV5TBLI:~$

```

- É possível executar algumas aplicações gráficas do Linux com WSL 2. Leia este tutorial: <https://medium.com/@dianaarnos/aplica%C3%A7%C3%B5es-gr%C3%A1ficas-no-wsl2-e0a481e9768c>.
- Execute o comando `wsl -l -v` com o PowerShell para ver as versões de Linux instaladas e seu status atual(parado ou rodando).

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\argen> wsl -l -v
  NAME                STATE      VERSION
*  Ubuntu              Running    2
  docker-desktop      Running    2
  docker-desktop-data Running    2
PS C:\Users\argen>

```

- Execute o comando `wsl --shutdown` com o PowerShell para desligar todas as distribuições Linux que estão rodando no momento (ao executar o comando, as distribuições do Docker também serão desligadas e o Docker Desktop mostrará uma notificação ao lado do relógio perguntando se você quer iniciar as distribuições dele novamente, se você não aceitar terá que iniciar o Docker novamente com o ícone perto do relógio do Windows).
- Execute com o PowerShell o comando `wsl --t <distribution name>` para desligar somente uma distribuição Linux específica.
- Se verificar que o WSL 2 está consumindo muitos recursos da máquina, execute os seguintes comandos dentro do terminal WSL 2 para liberar memória RAM:

```
echo 1 | sudo tee /proc/sys/vm/drop_caches
```

- Acrescente `export DOCKER_BUILDKIT=1` no final do arquivo `.profile` do seu usuário do Linux para ganhar mais performance ao realizar builds com Docker. Execute o comando `source ~/.profile` para carregar esta variável de ambiente no ambiente do seu WSL 2.
- Caso queira automatizar a inicialização do docker(pode-se iniciar outros serviços também), descrito na seção de desvantagens da utilização do docker nativo, enquanto não é incluído o `wsl.conf`, você pode configurar uma tarefa no windows para executar comandos no wsl, veja o tutorial a seguir, WSL 2: Run Ubuntu Services on Windows 10 Startup, substituindo o conteúdo do arquivo `/etc/init-wsl`, por um dos exemplos abaixo:

Exemplo de inicialização do serviço do docker

```

#!/bin/sh
#Mensagem de Inicialização dos serviços
echo initializing services

```

```
#Inicializa o serviço do docker  
service docker start
```

Exemplo gerando um log na Área de trabalho

```
#!/bin/sh  
#Grava a data/hora de início da execução no log  
date +%d/%m/%Y%t%H:%M:%S >> /mnt/c/Users/seu-usuario-do-windows/Desktop/wsl_start.log  
#Grava mensagem de Inicialização dos serviços no log  
echo initializing services >> /mnt/c/Users/seu-usuario-do-windows/Desktop/wsl_start.log  
#Inicializa o serviço do docker e grava o resultado do comando no log  
service docker start >> /mnt/c/Users/seu-usuario-do-windows/Desktop/wsl_start.log  
#Verifica o status do serviço do docker e grava o resultado do comando no log  
service docker status >> /mnt/c/Users/seu-usuario-do-windows/Desktop/wsl_start.log
```

- Infelizmente, caso reinicie o WSL manualmente, será necessário reiniciar os serviços manualmente, ou executar o `init-wsl` manualmente.
- No Windows 11 é possível iniciar o Docker automaticamente, veja a seção: Dica para Windows 11

’ Dúvidas

- O WSL 2 funciona junto com outras máquinas virtuais como **VirtualBox** ou **VMWare**? Siga a referência