

WSL – Linux e Windows Trabalhando Juntos

Situação comum:

O Dilema do Desenvolvedor

Empresa usa Windows

O ambiente corporativo exige Windows para ferramentas e políticas de TI

Servidores rodam Linux

A infraestrutura de produção está toda em Linux

Ferramentas que você precisa são feitas para Linux

As melhores ferramentas de desenvolvimento foram criadas para Linux

As Velhas Soluções

Antes do WSL, os desenvolvedores tinham que lidar com alternativas problemáticas:

✗ Dual Boot

- Precisa reiniciar toda hora
- Fica alternando entre sistemas

✗ Máquina Virtual

- Lenta
- Consome muita memória
- Complicada de configurar

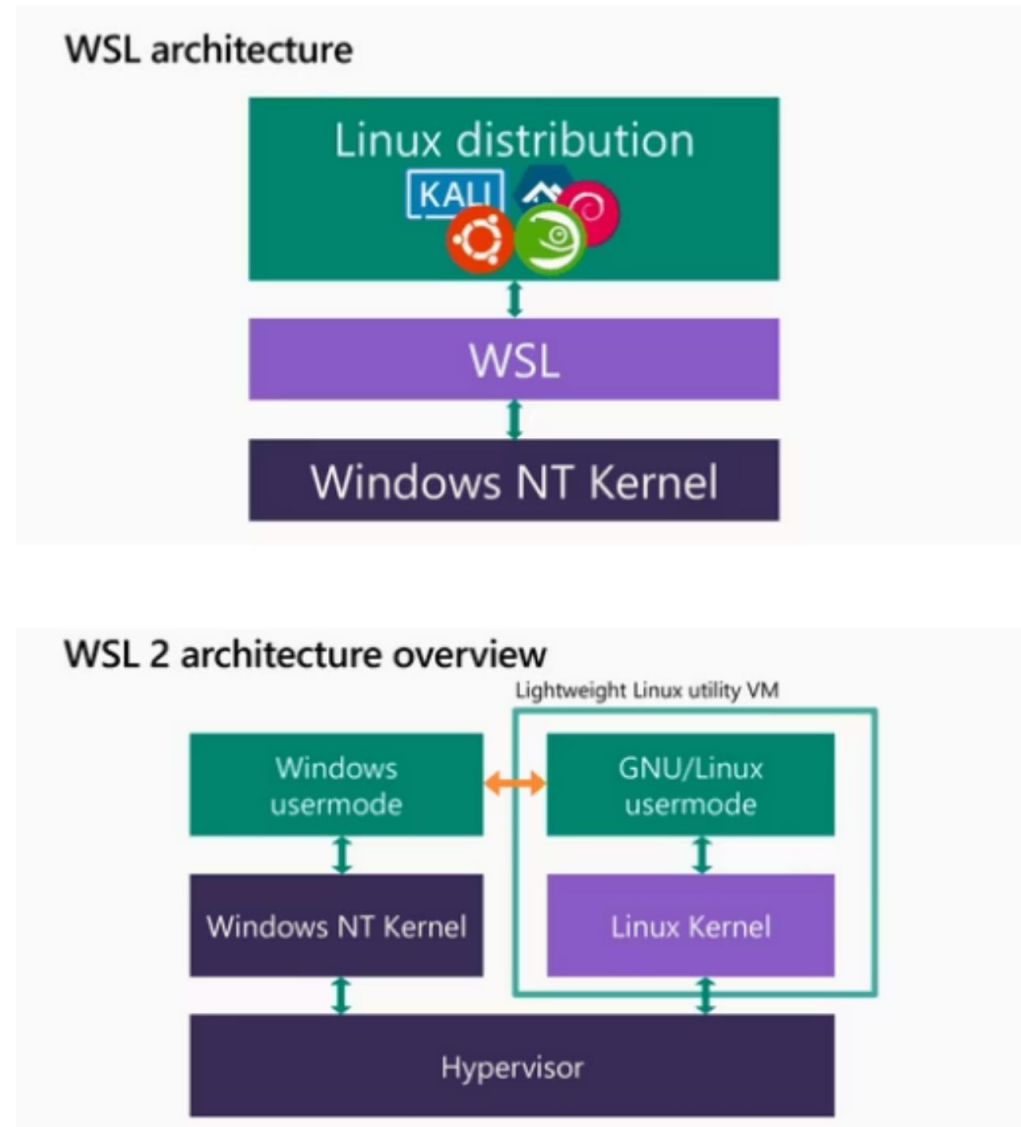
✗ "Adaptar" tudo pro Windows

- Ferramentas não funcionam direito
- Comandos diferentes
- Bugs estranhos

WSL – Windows Subsystem for Linux

WSL ([Windows Subsystem for Linux](#)) é um recurso do Windows que permite executar um ambiente Linux completo, incluindo ferramentas de linha de comando (como Bash) e aplicativos, diretamente no Windows sem a sobrecarga de uma máquina virtual tradicional, facilitando o desenvolvimento para desenvolvedores que precisam usar ambos os sistemas. Ele traduz as chamadas de sistema Linux para o kernel do Windows, permitindo rodar distribuições como Ubuntu ou Debian nativamente.

Versões – WSL1 vs WSL2



A arquitetura do WSL evoluiu significativamente entre as versões, trazendo melhorias substanciais de performance e compatibilidade.

Quão mais rápido é o WSL2?

As principais diferenças de performance e velocidade são:

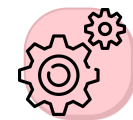


Operações de arquivos Linux: até 20x mais rápido

git clone, npm install, tar executam muito mais rapidamente

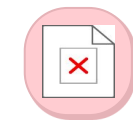


Docker: execução nativa, até 5x mais rápido que WSL1



Performance de CPU: ~95% da velocidade nativa

enquanto o WSL1 (uma camada de tradução) é menos eficiente em cargas de trabalho pesadas



Exceção: arquivos Windows (/mnt/c/) são mais lentos

use /home/ para projetos

Instalação e Uso Prático

```
Prompt de comando
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\Leonardo>wsl --list --online
A seguir está uma lista de distribuições válidas que podem ser instaladas.
Instale usando 'wsl.exe --install <Distro>'.

NAME                FRIENDLY NAME
Ubuntu              Ubuntu
Ubuntu-24.04        Ubuntu 24.04 LTS
openSUSE-Tumbleweed openSUSE Tumbleweed
openSUSE-Leap-16.0  openSUSE Leap 16.0
SUSE-Linux-Enterprise-15-SP7 SUSE Linux Enterprise 15 SP7
SUSE-Linux-Enterprise-16.0 SUSE Linux Enterprise 16.0
kali-linux          Kali Linux Rolling
Debian              Debian GNU/Linux
AlmaLinux-8         AlmaLinux OS 8
AlmaLinux-9         AlmaLinux OS 9
AlmaLinux-Kitten-10 AlmaLinux OS Kitten 10
AlmaLinux-10        AlmaLinux OS 10
archlinux           Arch Linux
FedoraLinux-43      Fedora Linux 43
FedoraLinux-42      Fedora Linux 42
eLxr                eLxr 12.12.0.0 GNU/Linux
Ubuntu-20.04        Ubuntu 20.04 LTS
Ubuntu-22.04        Ubuntu 22.04 LTS
OracleLinux_7_9     Oracle Linux 7.9
OracleLinux_8_10    Oracle Linux 8.10
OracleLinux_9_5     Oracle Linux 9.5
openSUSE-Leap-15.6  openSUSE Leap 15.6
SUSE-Linux-Enterprise-15-SP6 SUSE Linux Enterprise 15 SP6
```



Administrator: Command Promj



```
C:\Users\Craig>wsl --install
```

```
Installing: Virtual Machine Platform
```


```
Virtual Machine Platform has been installed.
```

```
Installing: Windows Subsystem for Linux
```

```
Windows Subsystem for Linux has been installed.
```

```
Downloading: Ubuntu
```

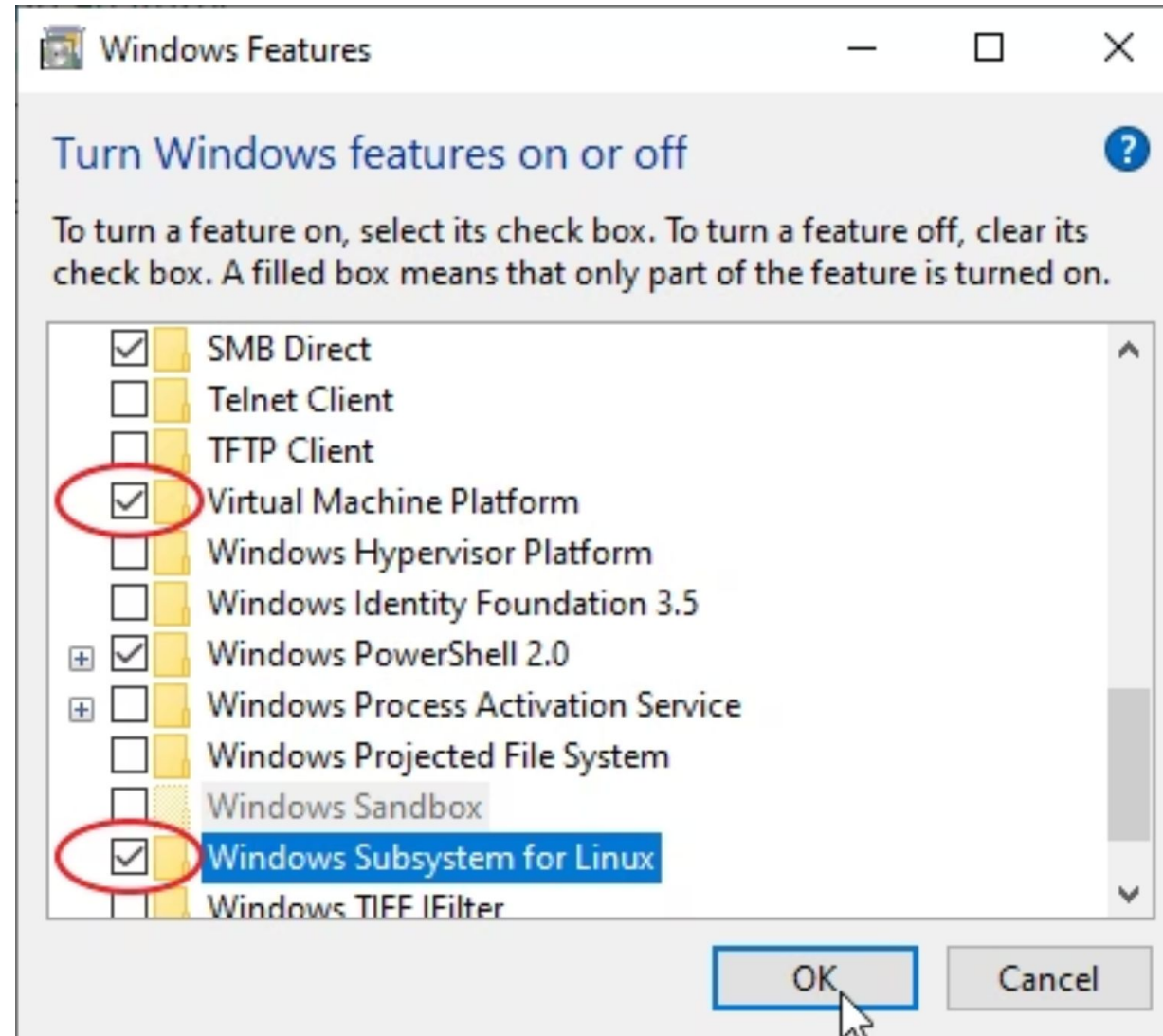
```
[=====93.4%=====]
```


 chrisdias@CHRISDIAS5020: ~


```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: chrisdias
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

chrisdias@CHRISDIAS5020:~$ uname
Linux
chrisdias@CHRISDIAS5020:~$ _
```

```
PS C:\Users\陈翔宇> ubuntu
Installing, this may take a few minutes...
WslRegisterDistribution failed with error: 0x80370102
Please enable the Virtual Machine Platform Windows feature and ensure virtualization is enabled in the BIOS.
For information please visit https://aka.ms/enablevirtualization
Press any key to continue...
PS C:\Users\陈翔宇>
```



▼  Este Computador

>  Disco Local (C:)

>  HDD (F:)

>  Rede

▼  Linux

>  docker-desktop

>  Ubuntu

Ubuntu

Linux > Ubuntu

Pesquisar em Ubuntu

Novo, Cortar, Copiar, Colar, Classificar, Visualizar, Detalhes

Início, Galeria, OneDrive

Área de Trabalho, Downloads, Documentos, Imagens, Músicas, Vídeos, projects, leomonte, System32, futura-diagrama-terraform, futura-diagrama-terraform, Capturas de Tela

Este Computador

26 itens | 1 item selecionado

bin.usr-is-merged	boot	dev
Docker	etc	home
lib.usr-is-merged	lost+found	media
mnt	opt	proc
root	run	sbin.usr-is-merged
snap	srv	sys
tmp	usr	var
bin Arquivo 7 bytes	init Arquivo 2,65 MB	lib Arquivo 7 bytes
lib64 Arquivo 9 bytes	sbin Arquivo 8 bytes	

home

< > ↑ ↻ > Linux > Ubuntu > home > Pesquisar em home

Novo ✕ Classificar ↕ Visualizar ☰ ... Detalhes

Nome	Data de modificação	Tipo	Tamanho
leomonte	21/01/2026 11:06	Pasta de arquivos	

Imagens
Músicas
Vídeos
projects
leomonte
System32
futura-diagrama-terraform
futura-diagrama-terraform
Capturas de Tela

Este Computador
Disco Local (C:)
HDD (F:)
Rede

Linux
docker-desktop
Ubuntu

1 item | 1 item selecionado



leomonte

Linux > Ubuntu > home > leomonte >

Pesquisar em leomonte

Novo

Classificar

Visualizar

Detalhes

Nome	Data de modificação	Tipo	Tamanho
.cache	12/01/2026 08:28	Pasta de arquivos	
.config	12/01/2026 14:25	Pasta de arquivos	
.docker	15/01/2026 12:08	Pasta de arquivos	
.dotnet	15/01/2026 08:45	Pasta de arquivos	
.landscape	12/01/2026 10:07	Pasta de arquivos	
.local	12/01/2026 10:54	Pasta de arquivos	
.npm	12/01/2026 10:22	Pasta de arquivos	
.nvm	12/01/2026 09:57	Pasta de arquivos	
.oh-my-zsh	08/01/2026 11:16	Pasta de arquivos	
.ssh	13/01/2026 09:47	Pasta de arquivos	
.terraform.d	19/01/2026 14:49	Pasta de arquivos	
.vscode-remote-containers	12/01/2026 08:24	Pasta de arquivos	
.vscode-server	12/01/2026 08:24	Pasta de arquivos	
projects	14/01/2026 15:42	Pasta de arquivos	
.aws	08/01/2026 10:44	Arquivo AWS	1 KB
.azure	08/01/2026 10:44	Arquivo AZURE	1 KB
.bash_history	12/01/2026 10:15	Arquivo BASH_HIS...	3 KB
.bash_logout	08/01/2026 09:18	Arquivo BASH_LO...	1 KB

33 itens

Terminal Personalizado - Oh My Zsh

Oh My Zsh é um framework de código aberto, orientado pela comunidade, para gerenciar sua configuração do Zsh.



```
ohmyz.sh demo  
→ projects omz update  
Updating Oh My Zsh  
  
Oh My Zsh is already at the latest version.  
  
To keep up with the latest news and updates, follow us on Twitter: @ohmyzsh  
Want to get involved in the community? Join our Discord: Discord server  
Get your Oh My Zsh swag at: Planet Argon Shop  
→ projects
```

Instalar Zsh:

```
sudo apt update  
sudo apt install zsh -y
```

Verificar instalação:

```
zsh --version
```

Definir Zsh como shell padrão:

```
chsh -s $(which zsh)
```

Instalar Oh My Zsh:

```
sh -c "$(curl -fsSL  
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

```
# Ou com wget
```

```
sh -c "$(wget  
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh -O -)"
```

Reiniciar terminal:


```
# Sair e entrar novamente, ou  
exec zsh
```

Trabalhando no Terminal

```
Ubuntu [x] + v - □ X
→ ~ pwd                               Windows
/home/leomonte                          \\wsl.localhost\Ubuntu\home\leomonte
→ ~ ls
LICENSE.txt  projects
→ ~ ls -a
.             .docker          .oh-my-zsh          .wget-hsts
..            .dotnet           .profile            .zcompdump
.aws          .gitconfig        .shell.pre-oh-my-zsh .zcompdump-fp-leobal-5.9
.azure        .landscape        .ssh                 .zcompdump-fp-leobal-5.9.zwc
.bash_history .lessht           .sudo_as_admin_successful .zsh_history
.bash_logout .local            .terraform.d        .zshrc
.bashrc       .motd_shown      .viminfo             LICENSE.txt
.cache        .npm              .vscode-remote-containers projects
.config       .nvm              .vscode-server
→ ~ cd projects/terraform-jenkins/terraformfuturajenkins
→ terraformfuturajenkins git:(treinamento) X █
```

● Diretórios (ocultos ou não)
● Symlinks ou tipos especiais
● Arquivos regulares

```
Ubuntu x Ubuntu x + v - □ ×
→ ~ cd /mnt
→ /mnt ls
c f wsl wslg
→ /mnt cd c/Users/Leonardo/Desktop
→ Desktop ls
'Docker Desktop.lnk'      leomonte-dev
'Visual Studio Code.lnk' portfolio-devops-main
apresentacao-wsl        'prefix-list-id = pl-04240c87d6da609.txt'
desktop.ini
→ Desktop cd portfolio-devops-main
→ portfolio-devops-main pwd
/mnt/c/Users/Leonardo/Desktop/portfolio-devops-main
→ portfolio-devops-main
```



```
Ubuntu x + v
→ cs16-server-devops git:(main) ls
Dockerfile  check_galileo.bat  galileo  requirements.txt  start_cs_server.bat  udp_test.py
README.md   docker-compose.override.yml  get_host_ip.ps1  restart_cs_server.bat  stop_cs_server.bat  users-adm-config
addons      docker-compose.yml  mostrar-ip.ps1  server.cfg        udp.py
→ cs16-server-devops git:(main) vim Dockerfile
→ cs16-server-devops git:(main) X █
```

```
Ubuntu [x] + v - □ ×
# Configurar ambiente Steam para SDK32
RUN mkdir -p /home/steam/.steam/sdk32 \
  && cp /home/steam/hlds/steamclient.so /home/steam/.steam/sdk32/ \
  && ln -sf /home/steam/hlds/steamclient.so /home/steam/hlds/cstrike/steamclient.so

# Configurações básicas do servidor
RUN mkdir -p /home/steam/hlds/cstrike \
  && printf "sv_lan 0\nmp_autokick 0\nmp_autoteambalance 0\nsv_region 255\n\
sv_visiblemaxplayers 12\nnet_public_adr 0.0.0.0\nlog on\n\
sv_logbans 1\nsv_logecho 1\nsv_logfile 1\n" > /home/steam/hlds/cstrike/server.cfg \
  && touch /home/steam/hlds/cstrike/{listip.cfg,banned.cfg} \
  && chown -R steam:steam /home/steam/hlds/cstrike

WORKDIR /home/steam/hlds

EXPOSE 27015/tcp 27015/udp 27020/udp 26900/udp

ENTRYPOINT ["/hlds_run"]
CMD ["/hlds_run", \
  "-game", "cstrike", \
  "+maxplayers", "16", \
  "+map", "de_dust2", \
  "+sv_lan", "0", \
  "+ip", "0.0.0.0", \
  "+port", "27015", \
  "+sys_ticrate", "1000", \
  "+fps_max", "1000", \
  "-pingboost", "3", \
  "-nomaster"]
```

125,1

Bot

```
Ubuntu x + v
# Configurar ambiente Steam para SDK32
RUN mkdir -p /home/steam/.steam/sdk32 \
  && cp /home/steam/hlds/steamclient.so /home/steam/.steam/sdk32/ \
  && ln -sf /home/steam/hlds/steamclient.so /home/steam/hlds/cstrike/steamclient.so

# Configurações básicas do servidor
RUN mkdir -p /home/steam/hlds/cstrike \
  && printf "sv_lan 0\nmp_autokick 0\nmp_autoteambalance 0\nsv_region 255\n\
sv_visiblemaxplayers 12\nnet_public_adr 0.0.0.0\nlog on\n\
sv_logbans 1\nsv_logecho 1\nsv_logfile 1\n" > /home/steam/hlds/cstrike/server.cfg \
  && touch /home/steam/hlds/cstrike/{listip.cfg,banned.cfg} \
  && chown -R steam:steam /home/steam/hlds/cstrike

WORKDIR /home/steam/hlds

EXPOSE 27015/tcp 27015/udp 27020/udp 26900/udp 28015/tcp 28015/udp

ENTRYPOINT ["/hlds_run"]
CMD ["/hlds_run", \
  "-game", "cstrike", \
  "+maxplayers", "16", \
  "+map", "de_dust2", \
  "+sv_lan", "0", \
  "+ip", "0.0.0.0", \
  "+port", "27015", \
  "+sys_ticrate", "1000", \
  "+fps_max", "1000", \
  "-pingboost", "3", \
  "-nomaster"]

-- INSERT --
```

125,67

Bot

```
Ubuntu x + v
# Configurar ambiente Steam para SDK32
RUN mkdir -p /home/steam/.steam/sdk32 \
  && cp /home/steam/hlds/steamclient.so /home/steam/.steam/sdk32/ \
  && ln -sf /home/steam/hlds/steamclient.so /home/steam/hlds/cstrike/steamclient.so

# Configurações básicas do servidor
RUN mkdir -p /home/steam/hlds/cstrike \
  && printf "sv_lan 0\nmp_autokick 0\nmp_autoteambalance 0\nsv_region 255\n\
sv_visiblemaxplayers 12\nnet_public_adr 0.0.0.0\nlog on\n\
sv_logbans 1\nsv_logecho 1\nsv_logfile 1\n" > /home/steam/hlds/cstrike/server.cfg \
  && touch /home/steam/hlds/cstrike/{listip.cfg,banned.cfg} \
  && chown -R steam:steam /home/steam/hlds/cstrike

WORKDIR /home/steam/hlds

EXPOSE 27015/tcp 27015/udp 27020/udp 26900/udp 28015/tcp 28015/udp

ENTRYPOINT ["/hlds_run"]
CMD ["/hlds_run", \
  "-game", "cstrike", \
  "+maxplayers", "16", \
  "+map", "de_dust2", \
  "+sv_lan", "0", \
  "+ip", "0.0.0.0", \
  "+port", "27015", \
  "+sys_ticrate", "1000", \
  "+fps_max", "1000", \
  "-pingboost", "3", \
  "-nomaster"]

:wq!
```

```
Ubuntu x + v
→ cs16-server-devops git:(main) X ls
Dockerfile  check_galileo.bat          galileo          requirements.txt  start_cs_server.bat  udp_test.py
README.md   docker-compose.override.yml  get_host_ip.ps1 restart_cs_server.bat stop_cs_server.bat  users-adm-config
addons      docker-compose.yml           mostrar-ip.ps1   server.cfg        udp.py

→ cs16-server-devops git:(main) X vim Dockerfile
→ cs16-server-devops git:(main) X
→ cs16-server-devops git:(main) X cat Dockerfile
FROM ubuntu:20.04

ENV DEBIAN_FRONTEND=noninteractive

# Instalar dependências necessárias para HLDS e AMX Mod X
RUN dpkg --add-architecture i386 \
  && apt-get update \
  && apt-get install -y \
    lib32gcc-s1 \
    lib32stdc++6 \
    wget \
    ca-certificates \
    lib32z1 \
    net-tools \
    iproute2 \
    libcurl4-gnutls-dev:i386 \
    unzip \
  && apt-get clean \
  && rm -rf /var/lib/apt/lists/*

# Criar usuário steam e diretórios
RUN useradd -m -s /bin/bash steam \
  && mkdir -p /home/steam/{steamcmd,hlds} \
```

```
Ubuntu x + v - □ ×
# Configurar ambiente Steam para SDK32
RUN mkdir -p /home/steam/.steam/sdk32 \
  && cp /home/steam/hlds/steamclient.so /home/steam/.steam/sdk32/ \
  && ln -sf /home/steam/hlds/steamclient.so /home/steam/hlds/cstrike/steamclient.so

# Configurações básicas do servidor
RUN mkdir -p /home/steam/hlds/cstrike \
  && printf "sv_lan 0\nmp_autokick 0\nmp_autoteambalance 0\nsv_region 255\n\
sv_visiblemaxplayers 12\nnet_public_adr 0.0.0.0\nlog on\n\
sv_logbans 1\nsv_logecho 1\nsv_logfile 1\n" > /home/steam/hlds/cstrike/server.cfg \
  && touch /home/steam/hlds/cstrike/{listip.cfg,banned.cfg} \
  && chown -R steam:steam /home/steam/hlds/cstrike

WORKDIR /home/steam/hlds

EXPOSE 27015/tcp 27015/udp 27020/udp 26900/udp 28015/tcp 28015/udp

ENTRYPOINT ["/hlds_run"]
CMD ["/hlds_run", \
  "-game", "cstrike", \
  "+maxplayers", "16", \
  "+map", "de_dust2", \
  "+sv_lan", "0", \
  "+ip", "0.0.0.0", \
  "+port", "27015", \
  "+sys_ticrate", "1000", \
  "+fps_max", "1000", \
  "-pingboost", "3", \
  "-nomaster"]
→ cs16-server-devops git:(main) X
```

```
Ubuntu x Prompt de comando x + v
(venv) C:\Users\Leonardo\Desktop\cs16-server-devops>stop_cs_server.bat
Encerrando servidor CS 1.6...
time="2026-01-22T11:16:53-03:00" level=warning msg="C:\\Users\\Leonardo\\Desktop\\cs16-server-devops\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
time="2026-01-22T11:16:53-03:00" level=warning msg="C:\\Users\\Leonardo\\Desktop\\cs16-server-devops\\docker-compose.override.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/1
  ✓ Container cs1.6_server-plugin Removed 1.7s
Servidor finalizado com sucesso.
Pressione qualquer tecla para continuar. . .

(venv) C:\Users\Leonardo\Desktop\cs16-server-devops>
```

```
Ubuntu x Prompt de comando x + v
(venv) → cs16-server-devops git:(main) X vim stop_cs_server_linux.sh
(venv) → cs16-server-devops git:(main) X cat stop_cs_server_linux.sh
#!/bin/bash

echo "Encerrando servidor CS 1.6..."
docker-compose down

if [ $? -ne 0 ]; then
    echo "ERRO ao finalizar servidor."
    read -p "Pressione Enter para continuar..."
    exit 1
fi

echo "Servidor finalizado com sucesso."
read -p "Pressione Enter para continuar..."
(venv) → cs16-server-devops git:(main) X chmod +x stop_cs_server_linux.sh
(venv) → cs16-server-devops git:(main) X ./stop_cs_server_linux.sh
Encerrando servidor CS 1.6...
WARN[0000] /mnt/c/Users/Leonardo/Desktop/cs16-server-devops/docker-compose.yml: the attribute `version` is obsolete, it
will be ignored, please remove it to avoid potential confusion
WARN[0000] /mnt/c/Users/Leonardo/Desktop/cs16-server-devops/docker-compose.override.yml: the attribute `version` is obso
lete, it will be ignored, please remove it to avoid potential confusion
[+] Running 1/1
  ✓ Container cs1.6_server-plugin Removed 1.7s
Servidor finalizado com sucesso.
Pressione Enter para continuar...
(venv) → cs16-server-devops git:(main) X
```

Docker Desktop: WSL2 vs Hyper-V

Docker Desktop (modo Windows)

I/O de disco: Mais lento devido à virtualização do Hyper-V

Operações de arquivo: Overhead significativo na tradução

Windows ↔ Linux

Build de imagens: Mais lento, especialmente com muitos layers

Uso de memória: Maior overhead do Hyper-V

WSL2

I/O de disco: 2–5x mais rápido

Operações de arquivo: 10–20x mais rápido em alguns casos

Build de imagens: 2x a 10x mais rápido

Uso de memória: Mais eficiente

Startup de containers: Mais rápido

Como Configurar o Docker?

Search settings

Resources

Docker Engine

Builders

AI

Kubernetes

Software updates

Extensions

Beta features

Notifications

Resources

Advanced

File sharing

Proxies

Network

WSL integration

Configure which WSL 2 distros you want to access Docker from.

Enable integration with my default WSL distro

Enable integration with additional distros:

Ubuntu

Refetch distros

Close

Apply



Search settings

General

Resources

Docker Engine

Builders

AI

Kubernetes

Software updates

Extensions

Beta features

Resources

Advanced

File sharing

Proxies

Network

WSL integration

You are using the WSL 2 backend, so resource limits are managed by Windows.

You can configure limits on the memory, CPU, and swap size allocated to WSL 2 in a [.wslconfig file](#).

Disk image location

C:\Users\Leonardo\AppData\Local\Docker\wsl

Browse

Resource Saver

Enable Resource Saver

Reduces CPU and memory utilization when no containers are running. Exit from Resource Saver mode happens automatically when containers are started.

Use the slider to set the duration of time between no containers running and Docker Desktop entering Resource Saver mode

Close

Apply



The image shows a terminal window with a dark background. At the top, there is a tab labeled "[wsl2]" and a plus sign for adding a new tab. Below the tab is a menu bar with the words "Arquivo", "Editar", and "Exibir" on the left, and a set of icons (font size, list, bold, italic, link, grid, and undo) on the right. The main area of the terminal contains the following text:

```
[wsl2]
memory=4GB      # Limits the VM memory to 4GB
processors=2    # Sets the number of processors to use
swap=12GB      # Sets amount of swap storage space to
dns=192.168.1.1 # Sets the DNS server
|
```

At the bottom left of the terminal, the file path `C:\Users\<UserName>\.wslconfig` is displayed in red text.

Search settings

Resources

Docker Engine

Builders

AI

Kubernetes

Software updates

Extensions

Beta features

Notifications

Docker Engine

v29.1.3

Configure the Docker daemon by typing a json Docker daemon [configuration file](#).

This can prevent Docker from starting. Use at your own risk.

```
{
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "experimental": false
}
```

Close

Apply

```
Ubuntu x Ubuntu x + v - □ ×
Cgroup Driver: cgroupfs
Cgroup Version: 2
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
CDI spec directories:
  /etc/cdi
  /var/run/cdi
Discovered Devices:
  cdi: docker.com/gpu=webgpu
Swarm: inactive
Runtimes: nvidia runc io.containerd.runc.v2
Default Runtime: runc
Init Binary: docker-init
containerd version: 1c4457e00facac03ce1d75f7b6777a7a851e5c41
runc version: v1.3.4-0-gd6d73eb8
init version: de40ad0
Security Options:
  seccomp
  Profile: builtin
  cgroupns
Kernel Version: 6.6.87.2-microsoft-standard-WSL2
Operating System: Docker Desktop
OSType: linux
Architecture: x86_64
CPUs: 4
```

Conclusão:

WSL = Linux + Windows trabalhando juntos

Ótima integração com Docker

Facilita desenvolvimento multiplataforma

Pronto para uso em minutos