Curso: API RESTful com Spring Boot

https://lucasangelo.com

Lucas Ângelo

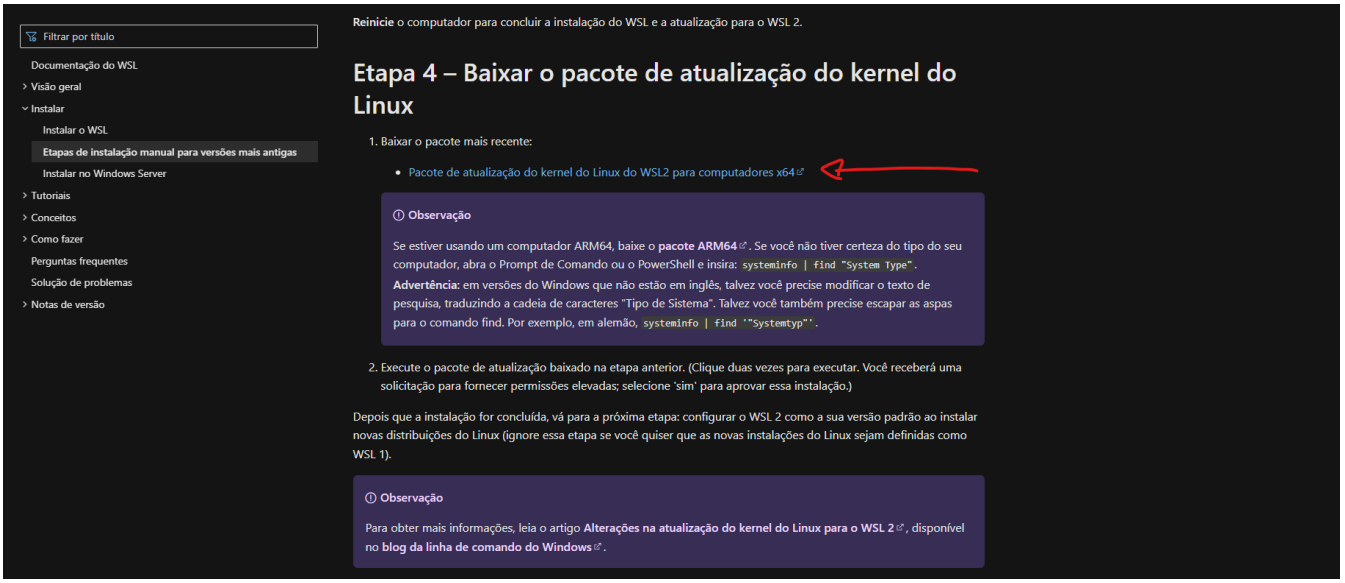
Instalação das ferramentas

Checklist:

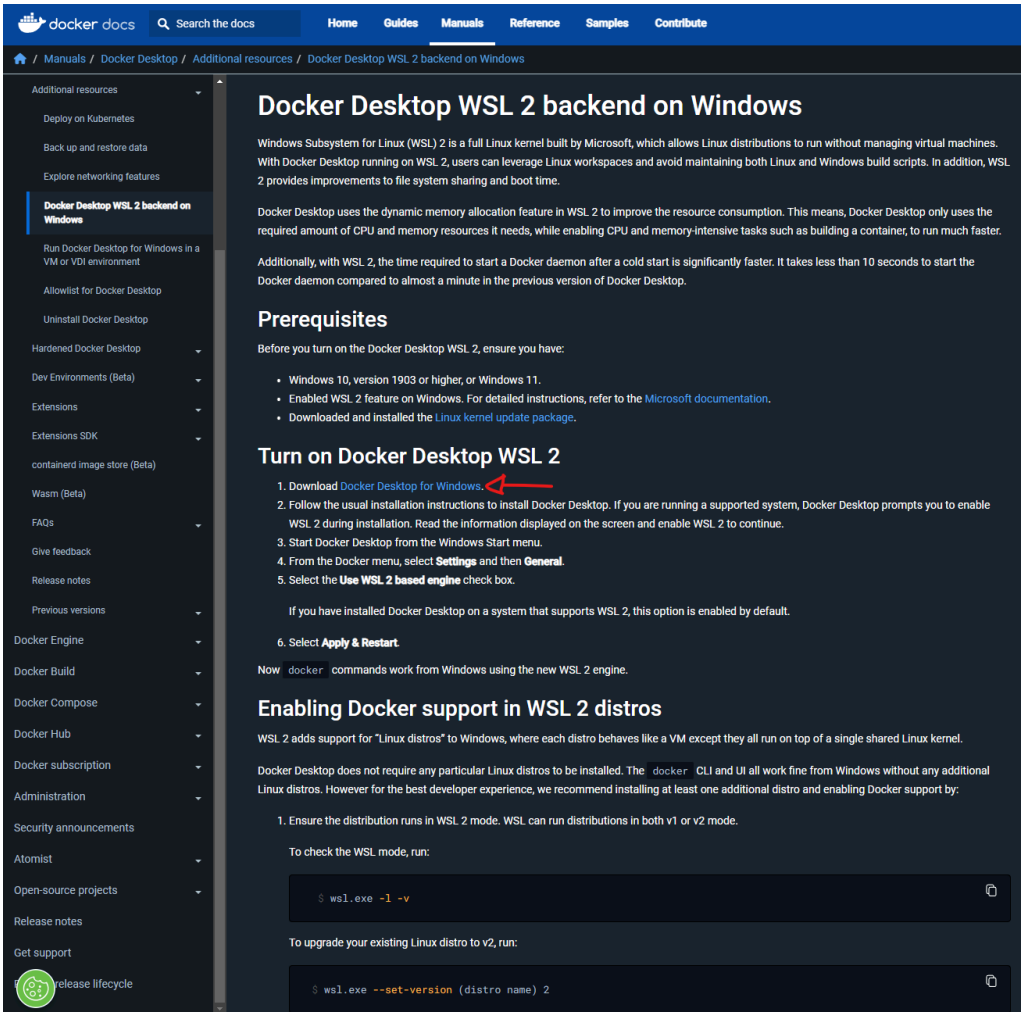
- Google Chrome / Microsoft Edge
- Conta no Github: https://github.com/signup
- Git Bash: https://git-scm.com/downloads
  - git --version
- Postman: https://www.postman.com/downloads/
- Java JDK (versão LTS 17)
  - https://www.oracle.com/java/technologies/javase-downloads.html
  - Configurar variáveis de ambiente do sistema
    - Painel de Controle -> Variáveis de Ambiente
    - JAVA_HOME:
      - C:\Program Files\Java\jdk-17.0.2
    - Path:
      - %JAVA_HOME%\bin
- IDE / Editor de Código
  - Visual Studio Code (VSCode): https://code.visualstudio.com/
  - Extensões para o VSCode:
    - Extension Pack for Java e Spring Boot Extension Pack: https://code.visualstudio.com/docs/java/extensions
  - Alternativas:
    - STS - Spring Tool Suit: https://spring.io/tools
    - IntelliJ IDEA: https://www.jetbrains.com/idea/ (Versão Ultimate grátis para estudantes)
    - Eclipse IDE https://www.eclipse.org/downloads/
- MySQL Community Server 8: https://dev.mysql.com/downloads/mysql/
  - Heidisql: https://www.heidisql.com/download.php
- Apache Maven (versão >= 3.8.6)
  - https://maven.apache.org/download.cgi
  - Configurar variáveis de ambiente do sistema
    - Painel de Controle -> Variáveis de Ambiente
    - MAVEN_HOME:
      - C:/Program Files/Maven/apache-maven-3.8.5
    - Path:
      - %MAVEN_HOME%\bin
- Gerador do projeto Spring Boot: https://start.spring.io/
  - Maven Project
  - Spring Boot >= 2.7.2 (Que não seja SNAPSHOT)
  - Packaging Jar
  - Java 17
  - Dependências:
    - Parametrizadas no link abaixo
  - https://start.spring.io/#!type=maven-project&language=java&platformVersion=2.7.2&packaging=jar&jvmVersion=17&groupId=com.lucasangelo&artifactId=todosimple&name=todosimple&description=Spring%20Boot%20API%20for%20To%20Do%20App&packageName=com.lucasangelo.todosimple&dependencies=devtools,web,data-jpa,h2,mysql,validation

- https://github.com/ICEI-PUC-Minas-PPLES-TI/PLF-ES-2022-2-MON-CursoAPIJava/commit/ab9a862a6c6fa0668bd2a6cbead71161d59a18b0?diff=unified&w=0#diff-9c5fb3d1b7e3b0f54bc5c4182965c4fe1f9023d449017cece3005d3f90e8e4d8 (ir para o pom.xml + Ctrl C + Ctrl V, corrigindo o pom.xml do spring.io)

- Docker Desktop
  - Abra o Windows PowerShell como administrador e execute o comando para instalar o WSL:
    - wsl --install
  - Baixe o pacote de atualização do kernel do Linux:
    - https://learn.microsoft.com/pt-br/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package



  - Instale o Ubuntu pela Microsoft Store:
    - https://apps.microsoft.com/store/detail/ubuntu-22042-lts/9PN20MSR04DW
  - Baixe o instalador do Docker Desktop, abra e siga os passos padrões de instalação:
    - https://docs.docker.com/desktop/windows/wsl/


docker docs
Search the docs
Home
Guides
Manuals
Reference
Samples
Contribute
/ Manuals / Docker Desktop / Additional resources / Docker Desktop WSL 2 backend on Windows
Additional resources
Deploy on Kubernetes
Back up and restore data
Explore networking features
Docker Desktop WSL 2 backend on Windows
Run Docker Desktop for Windows in a VM or VDI environment
Allowlist for Docker Desktop
Uninstall Docker Desktop
Hardened Docker Desktop
Dev Environments (Beta)
Extensions
Extensions SDK
containerd image store (Beta)
Wasm (Beta)
FAQs
Give feedback
Release notes
Previous versions
Docker Engine
Docker Build
Docker Compose
Docker Hub
Docker subscription
Administration
Security announcements
Atomist
Open-source projects
Release notes
Get support
release lifecycle
Docker Desktop WSL 2 backend on Windows
Windows Subsystem for Linux (WSL) 2 is a full Linux kernel built by Microsoft, which allows Linux distributions to run without managing virtual machines. With Docker Desktop running on WSL 2, users can leverage Linux workspaces and avoid maintaining both Linux and Windows build scripts. In addition, WSL 2 provides improvements to file system sharing and boot time.
Docker Desktop uses the dynamic memory allocation feature in WSL 2 to improve the resource consumption. This means, Docker Desktop only uses the required amount of CPU and memory resources it needs, while enabling CPU and memory-intensive tasks such as building a container, to run much faster.
Additionally, with WSL 2, the time required to start a Docker daemon after a cold start is significantly faster. It takes less than 10 seconds to start the Docker daemon compared to almost a minute in the previous version of Docker Desktop.
Prerequisites
Before you turn on the Docker Desktop WSL 2, ensure you have:
Windows 10, version 1903 or higher, or Windows 11.
Enabled WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
Downloaded and installed the Linux kernel update package.
Turn on Docker Desktop WSL 2
Download Docker Desktop for Windows.
Follow the usual installation instructions to install Docker Desktop. If you are running a supported system, Docker Desktop prompts you to enable WSL 2 during installation. Read the information displayed on the screen and enable WSL 2 to continue.
Start Docker Desktop from the Windows Start menu.
From the Docker menu, select Settings and then General.
Select the Use WSL 2 based engine check box.
If you have installed Docker Desktop on a system that supports WSL 2, this option is enabled by default.
Select Apply & Restart.
Now docker commands work from Windows using the new WSL 2 engine.
Enabling Docker support in WSL 2 distros
WSL 2 adds support for "Linux distros" to Windows, where each distro behaves like a VM except they all run on top of a single shared Linux kernel.
Docker Desktop does not require any particular Linux distros to be installed. The docker CLI and UI all work fine from Windows without any additional Linux distros. However for the best developer experience, we recommend installing at least one additional distro and enabling Docker support by:
Ensure the distribution runs in WSL 2 mode. WSL can run distributions in both v1 or v2 mode.
To check the WSL mode, run:
$ wsl.exe -l -v
To upgrade your existing Linux distro to v2, run:
$ wsl.exe --set-version (distro name) 2