

# Introdução ao Docker

Rosicley Rosa

GRUMEC

19/08/2024



- 1 O que é Docker?
- 2 Docker vs Máquinas Virtuais
- 3 Por que usar Docker?
- 4 Primeiros Passos com Docker
- 5 Imagens e Contêiners
- 6 Docker no Fluxo de Trabalho da Pesquisa

# O que é Docker?

- **Docker** é uma plataforma para desenvolver, enviar e executar aplicativos em *containers*.
- Os *containers* permitem que você empacote um aplicativo com todas as suas dependências em uma única unidade executável.
- **Não é uma Máquina Virtual (VM).**



# Docker vs Máquinas Virtuais

- **Arquitetura:**

- **Docker:** Compartilha o kernel do sistema operacional do *host*.
- **VMs:** Cada VM tem seu próprio sistema operacional completo.

- **Desempenho:**

- **Docker:** Mais leve e rápido; inicia em segundos pois não há a sobrecarga de um SO completo.
- **VMs:** Mais pesado; inicia em minutos.

- **Uso de Recursos:**

- **Docker:** Menor uso de CPU, memória e armazenamento.
- **VMs:** Maior consumo de recursos devido ao SO completo.

- **Portabilidade:**

- **Docker:** Imagens Docker podem ser executadas em qualquer sistema com Docker instalado.
- **VMs:** Depende do *hypervisor* e da configuração do SO.

# Por que usar Docker?

- Consistência em diferentes ambientes de desenvolvimento.
- Reduz o tempo de configuração do ambiente.
- Isolamento dos aplicativos:
  - *Container* operam de forma independente.
  - Não interferem uns nos outros e no SO do *host*.
- *Container* são leves e podem ser rapidamente distribuídos e escalados em outros servidores.

# Instalando Docker

- Docker pode ser instalado em várias plataformas (Linux, Windows, Mac).
- Acesse <https://docs.docker.com/engine/install/> para mais informações.
- Instruções básicas para instalação no Ubuntu:

```
sudo apt-get update
sudo apt-get install docker.io

docker --version
docker run hello-world
```

# Imagens do Docker

- **Docker Hub:** Um repositório para armazenar e compartilhar imagens Docker.

```
docker pull nome-da-imagem
```

- Uma imagem pode ser criada a partir de um arquivo Dockerfile.
  - O Dockerfile contém instruções para construir a imagem.
  - Deve conter todas as dependências necessárias para executar o aplicativo.
- Exemplo de Dockerfile:

```
FROM ubuntu:22.04
WORKDIR /app
COPY . /app
RUN apt-get update
RUN apt-get install -y git make cmake vim
```

- Construir a imagem:

```
docker build -f /caminho/para/o/Dockerfile -t nome-da-imagem .
```

- Um *container* é uma instância em execução de uma imagem.
- Comandos úteis:
  - `docker ps`: Lista *containers* em execução.
  - `docker ps -a`: Lista todos os *containers*.
  - `docker run -it nome-da-imagem`: Cria e executa um *container* a partir de uma imagem.
  - `docker exec -it id-do-container bash`: Acessa o terminal do *container*.
  - `docker start id-do-container`: Inicia um *container* parado.
  - `docker stop id-do-container`: Para um *container* em execução.
  - `docker rm id-do-container`: Remove um *container* parado.
- Volumes são usados para persistir dados entre *containers* e o *host*.

```
docker run -v /caminho/no/host:/caminho/no/container nome-da-imagem
```

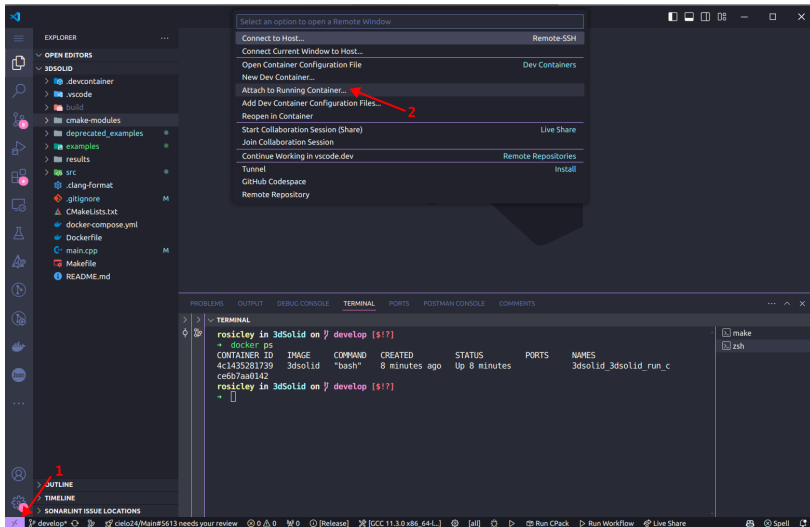


# Aplicações de Docker na Pesquisa

- Crie um Dockerfile contendo todas as dependências necessárias para executar o código (instalar [PETSc](#), [MPI](#), etc.).
- Construa a imagem do Docker.
- Inicialize um *container* definindo o volume para o diretório do código.
  - [Docker Compose](#) simplifica a configuração.
  - As alterações no código são refletidas no *container* e vice-versa.
- Acesse o terminal do *container* e execute o código.

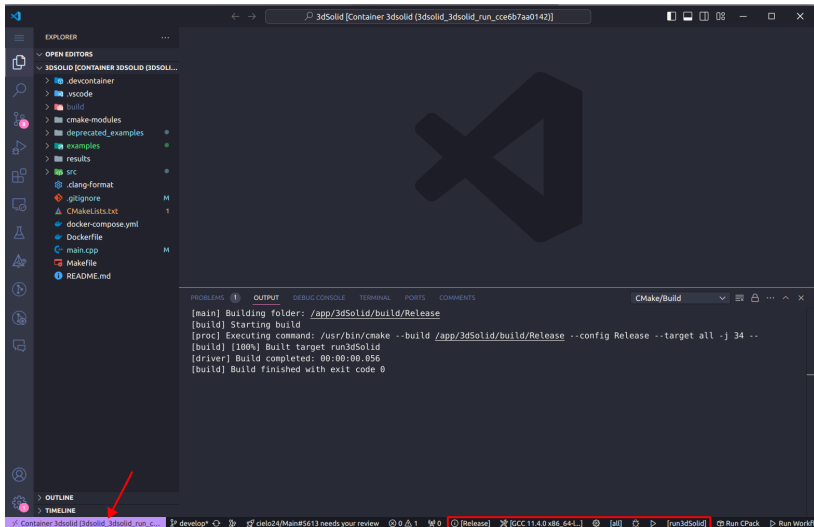
# Aplicações de Docker na Pesquisa com VS Code - Opção 1

- Instale a extensão **Dev Containers**.
- Use a opção *Attach to Running Container*.



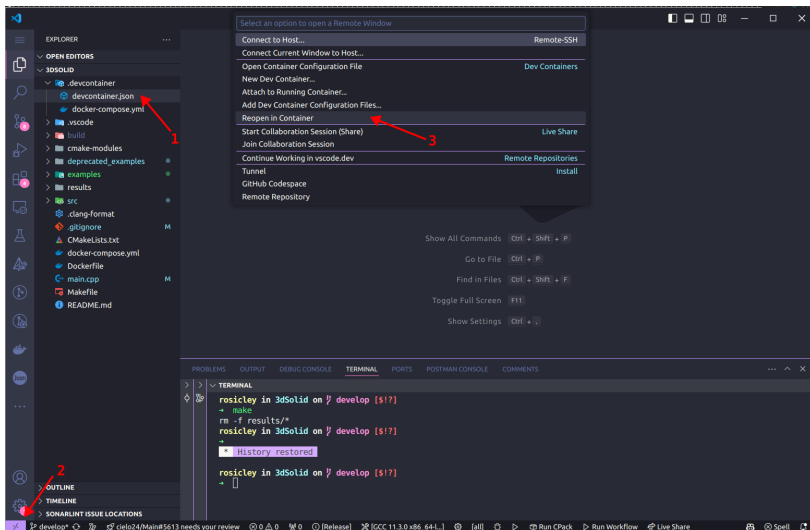
# Aplicações de Docker na Pesquisa com VS Code - Opção 1

- Instale as extensões necessárias no *container*, como o **CMake**.
- Execute o código diretamente no VS Code, porém dentro do *container*.



# Aplicações de Docker na Pesquisa com VS Code - Opção 2

- Crie um **Dev Container** do VS Code.
- Use a opção *Reopen in Container*.



- Docker é uma ferramenta poderosa para gerenciar ambientes de desenvolvimento e produção.
- Facilita a replicabilidade de experimentos e a configuração rápida de ambientes de simulação.
- Simplifica o compartilhamento de código e ambientes entre colaboradores.
- Encaminhe suas perguntas e vamos começar a usar Docker!