

PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA

Containers

Aula 03 — Containers, Docker e sua infraestrutura

Prof. Newton S. B. Miyoshi

newton.miyoshi@baraodemaua.br

Centro Universitário Barão de Mauá — Ribeirão Preto

TÓPICOS

O que veremos hoje

- Containers e Docker — e as diferenças com VMs
- Vantagens
- IaaS e PaaS
- Infraestrutura do Docker

PARTE 1

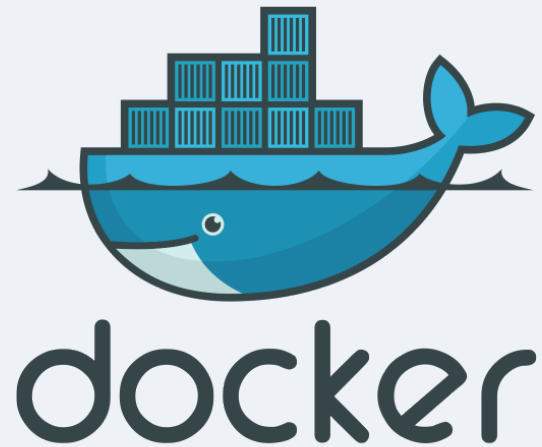
Containers e Docker

O que é um Container?

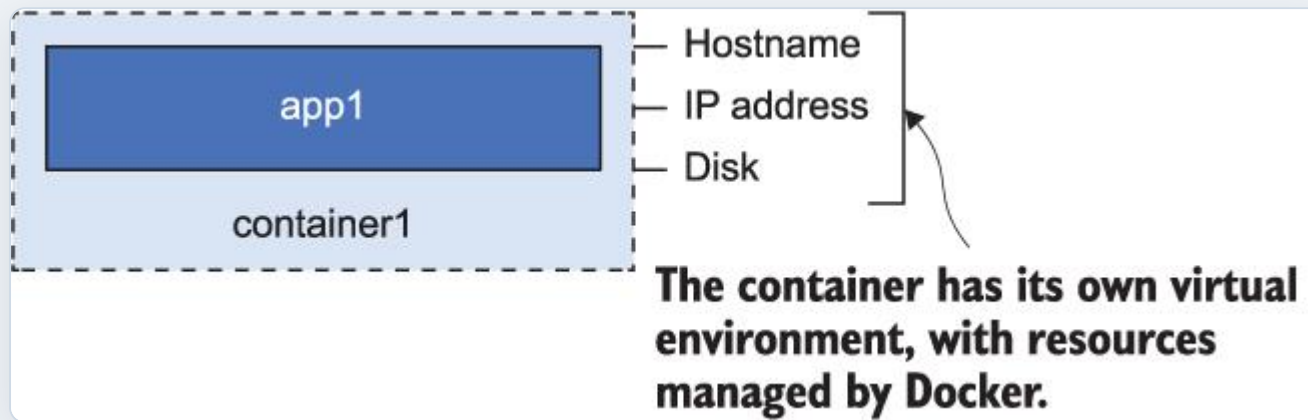
- Uma forma de isolar uma aplicação virtualizando alguns componentes.
- Uma “caixa” com os componentes mínimos para executar uma aplicação:
 - Próprio hostname, IP e disco
 - Aplicações dentro da caixa não acessam nada fora
- Diferente de uma VM, há compartilhamento de CPU, memória e do Sistema Operacional.

O que é o Docker?

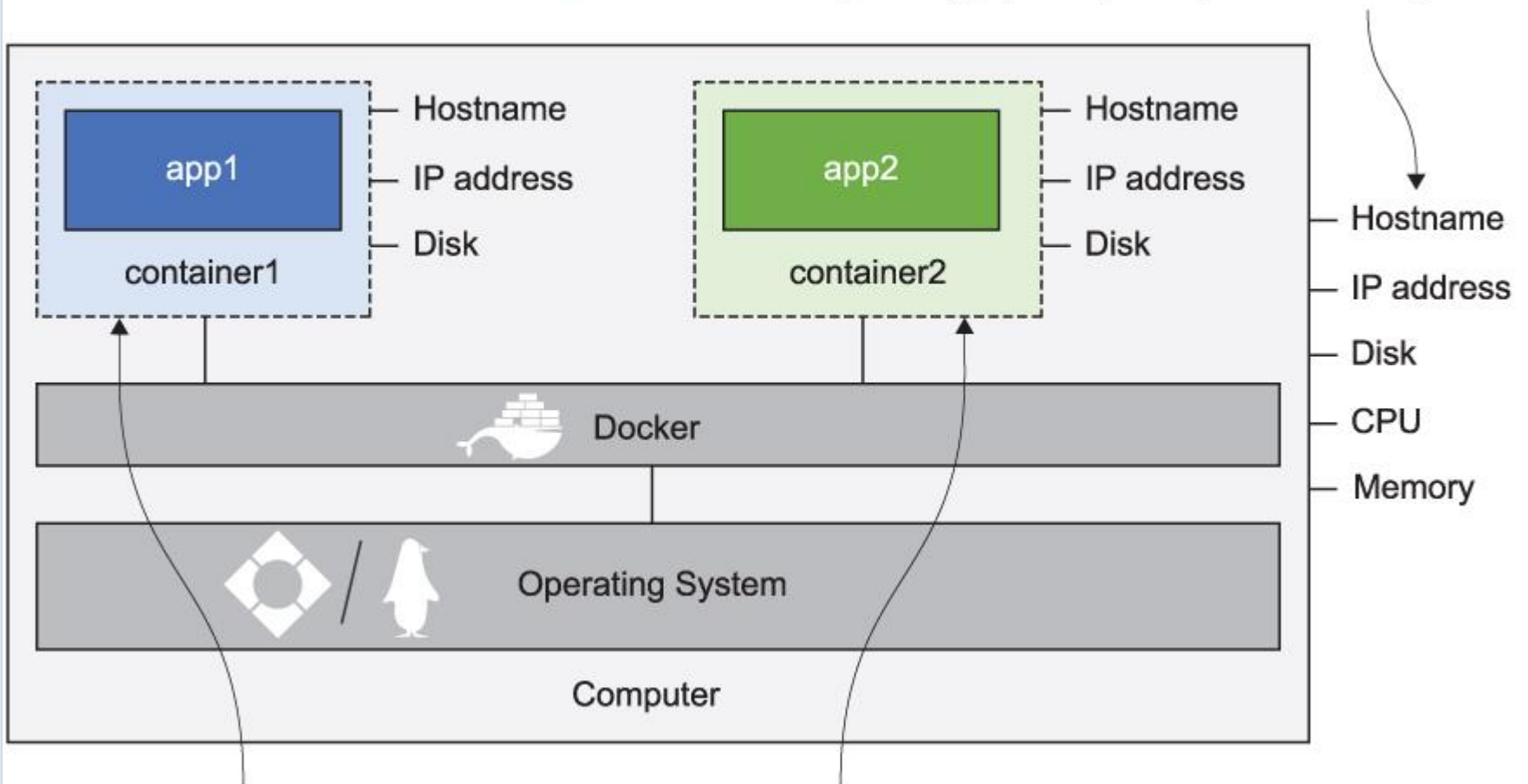
- Plataforma para **criar, gerenciar e compartilhar** containers.
- Oferece ferramentas para **todo o ciclo de vida** dos containers.



Visão geral de um container



The host computer has a separate computer name, IP address, and disk. Containers all share the host's operating system, CPU, and memory.



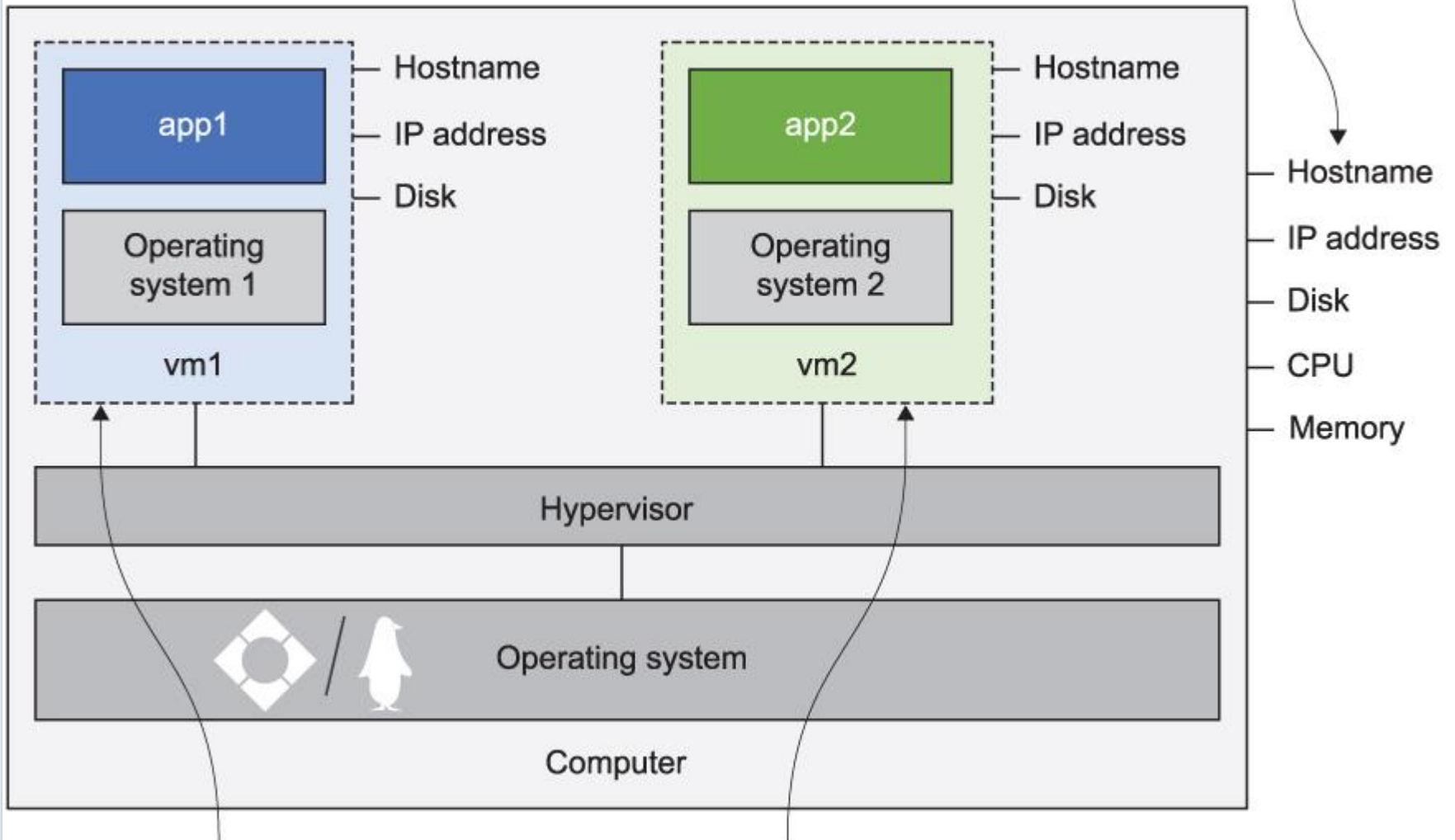
Each container has its own computer name, IP address, and disk.

Diferenças em relação às VMs

VMs têm o próprio SO e virtualizam todos os recursos computacionais.

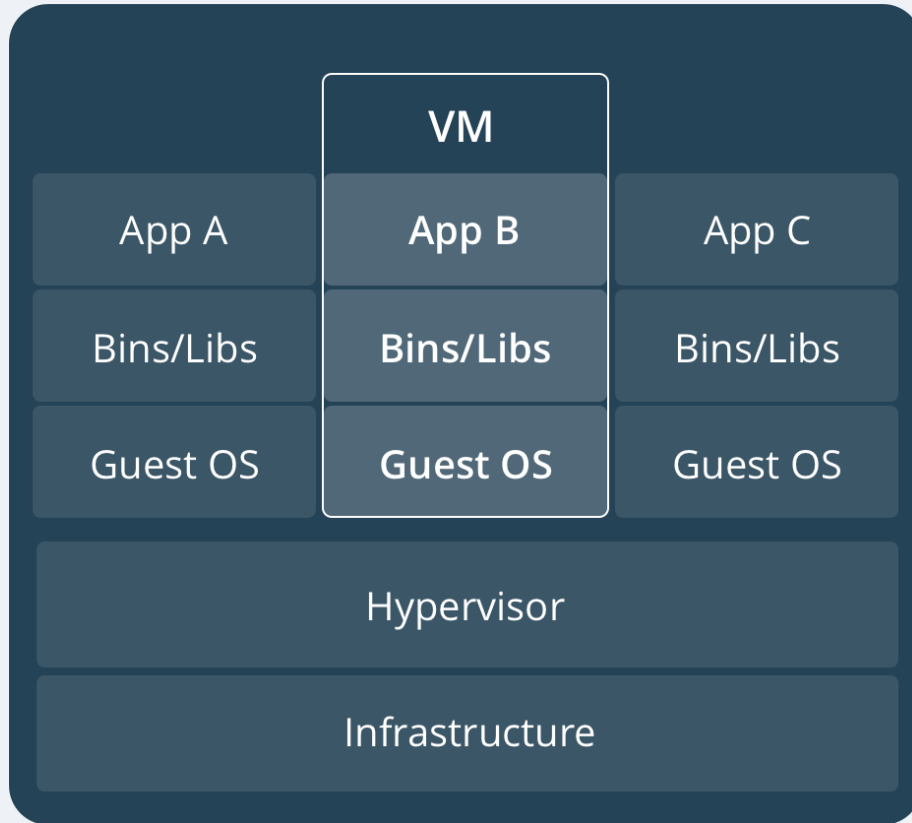
- SO host e guest
- Muitos GB de tamanho · tempo de CPU · atualização de SO
- Maior isolamento, porém menor densidade de computação e dados

The host computer has a separate computer name, IP address, and disk. Virtual machines all share the host's CPU and memory.

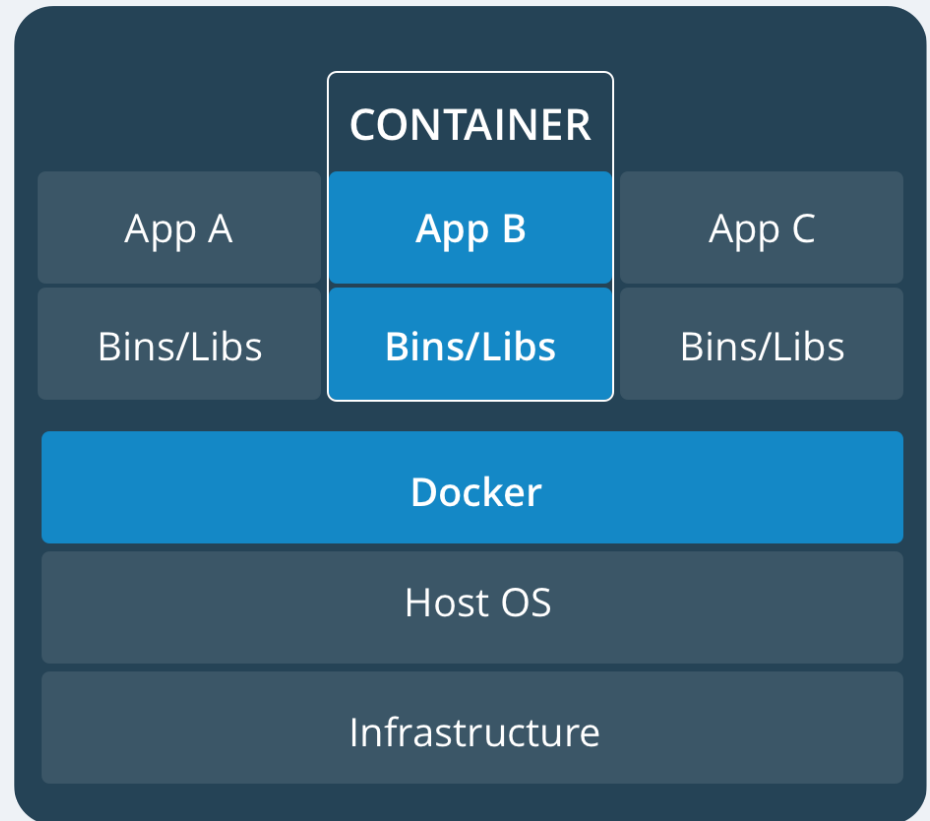


Each VM has its own OS, computer name, IP address, and disk.

VM × Container



Máquina Virtual — um SO convidado por aplicação

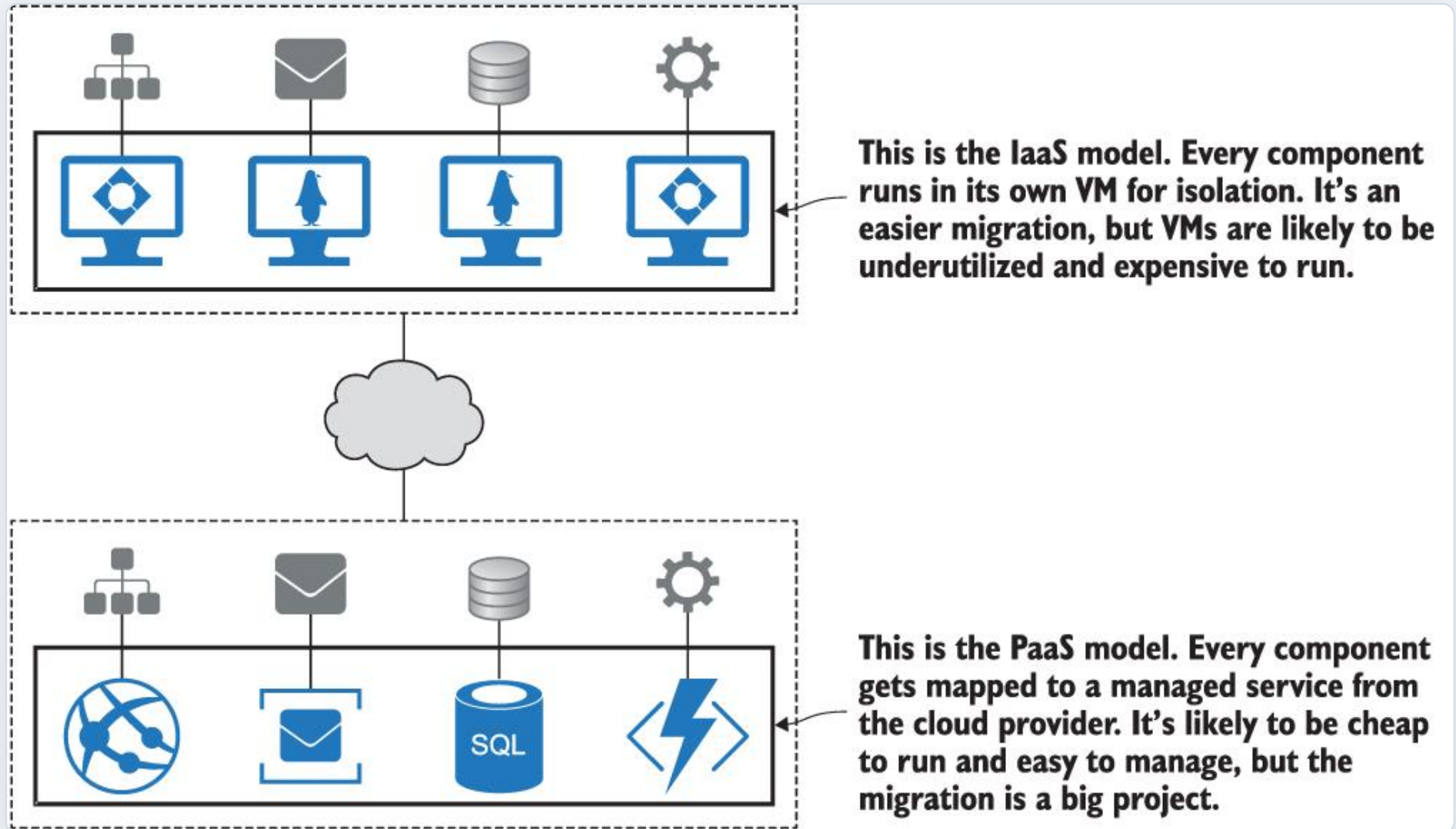


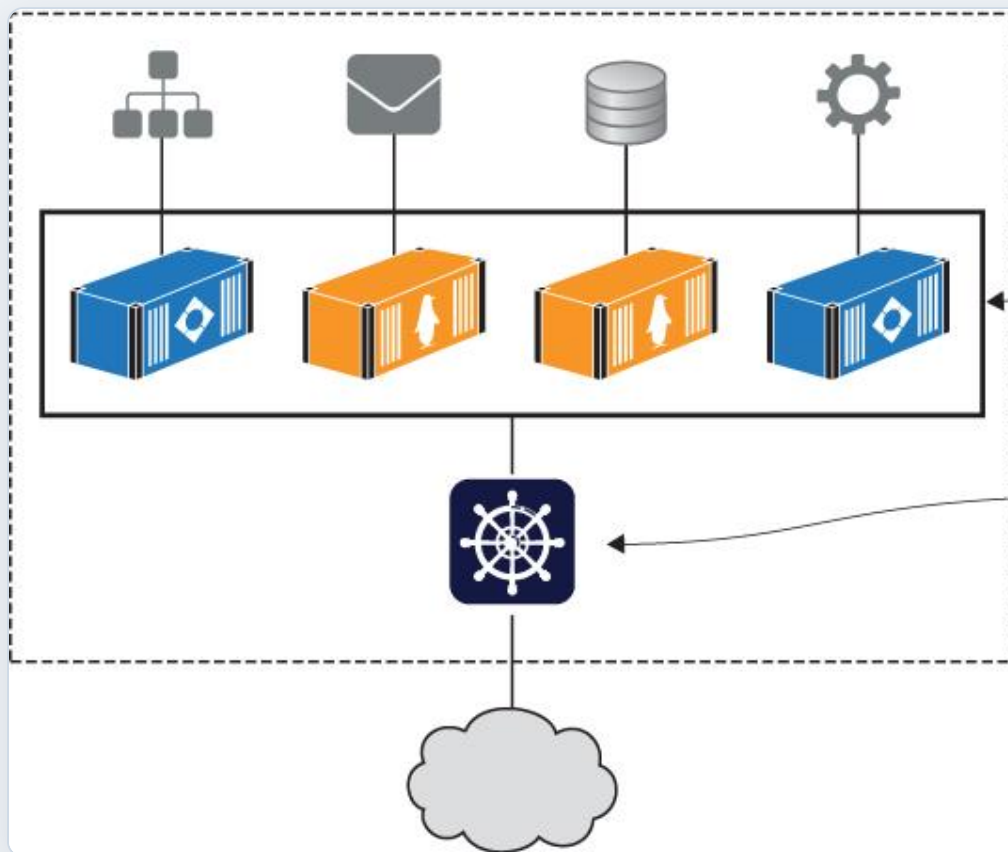
Container — SO compartilhado via Docker

PARTE 2

IaaS, PaaS e Containers

IaaS e PaaS





The application components all run in containers. They are isolated like VMs but lightweight and efficient like PaaS services.

Dockerized applications are portable. They can run in a managed service like Azure Kubernetes Service, and they can move to AWS or to the datacenter with no changes.

PARTE 3

Vantagens

ATIVIDADE EM GRUPO



**Quais as vantagens de usar
containers no projeto “Café
com Pão”?**

Comparem VM × Containers para o cenário.

Por que containers?

Flexível

Qualquer aplicação pode residir em um container.

Leve

Menos recursos que uma VM.

Portável

Build local, roda na nuvem.

Fracamente acoplado

Um container não interfere no outro.

Escalável

Replicável em um datacenter.

Seguro

Isolamento de processos.

Usos do Docker

- CI/CD — Continuous Integration & Continuous Delivery
- Desenvolvedores compartilham código por meio de containers
- O ambiente de produção é o mesmo do de desenvolvimento
- Criação de ambientes e rotinas de testes automatizados mais facilmente

PARTE 4

Infraestrutura do Docker

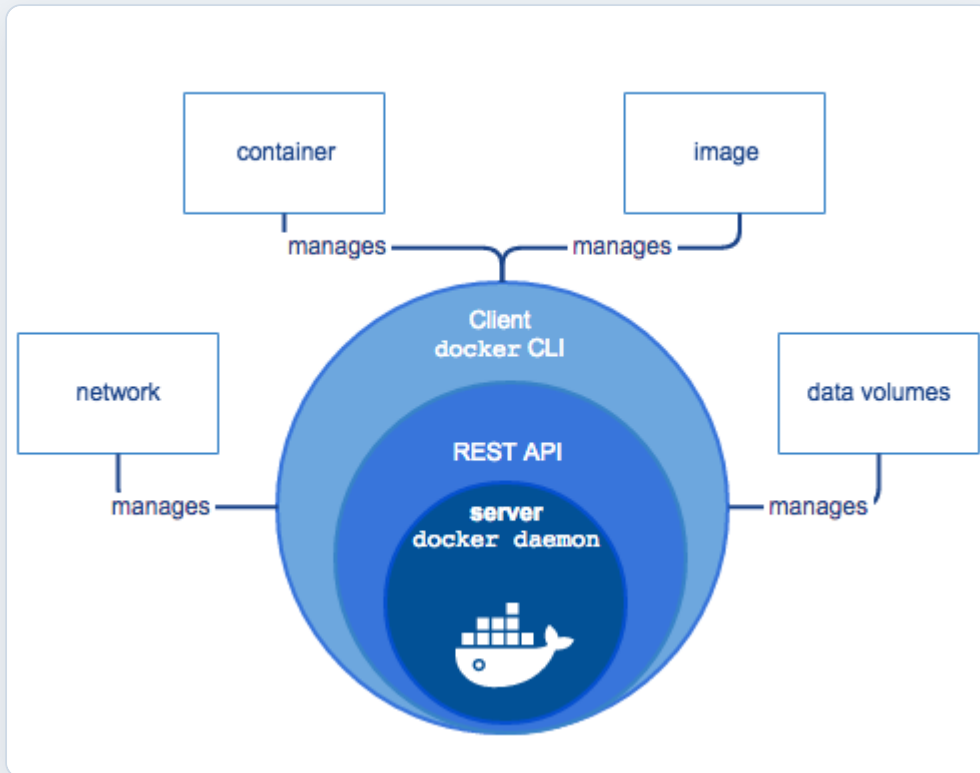
Docker Engine

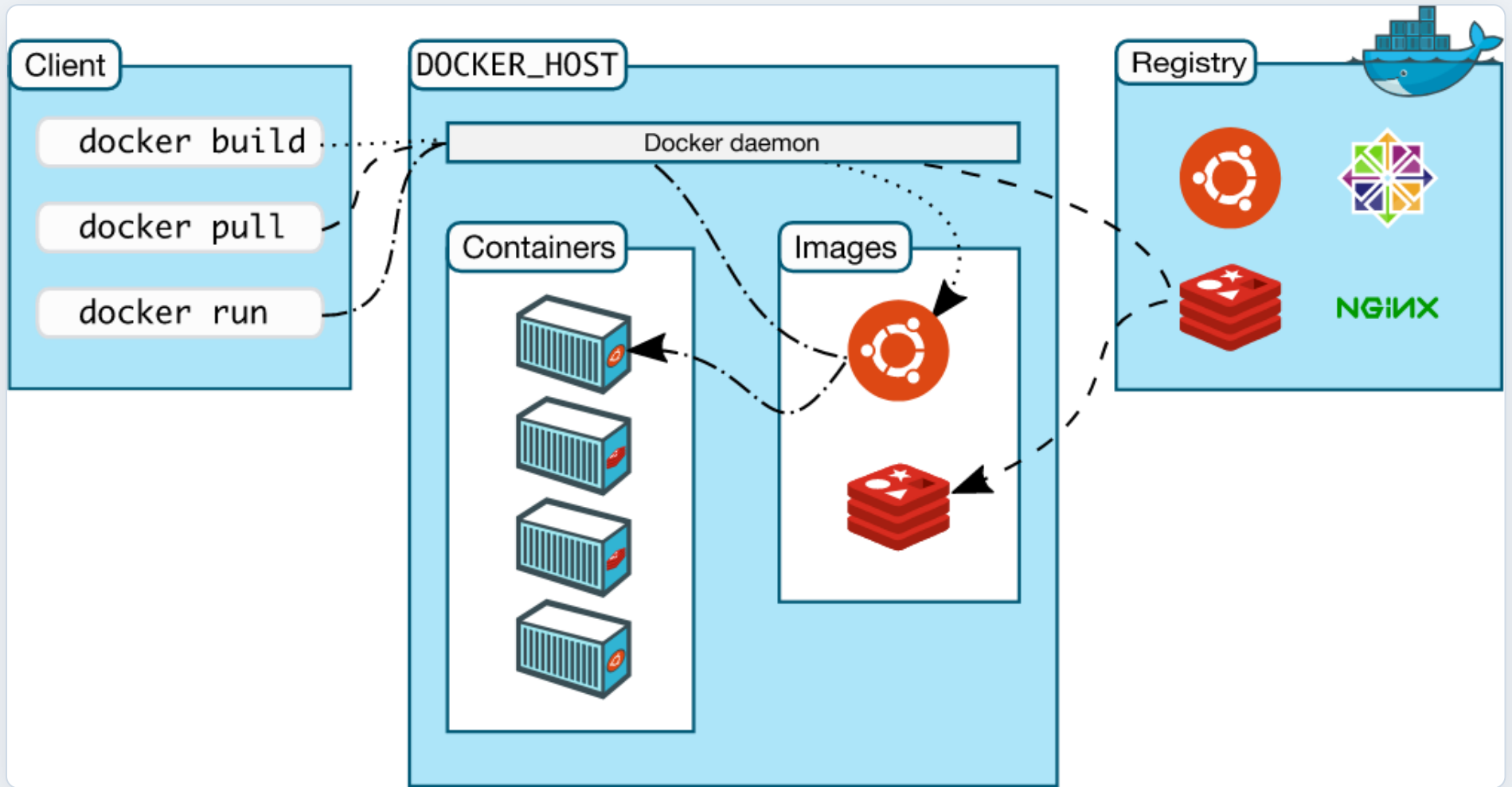
Aplicação cliente-servidor com os componentes:

- Servidor (daemon) — gerencia containers, acessado por `dockerd`
- API REST — interface para comunicar com o `dockerd`
- Cliente CLI — o comando `docker`
 - Usa a REST API para criar e gerenciar imagens, containers, redes e volumes

Arquitetura do Docker

- Arquitetura cliente-servidor.
- O cliente `docker` comunica-se com o `docker daemon` :
 - É possível conectar remotamente
 - Comunicação pela REST API
 - Via sockets UNIX ou pela rede (internet)





Instalando o Docker

docs.docker.com/get-docker



Docker Desktop for Mac

A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.



Docker Desktop for Windows

A native Windows application which delivers all Docker tools to your Windows computer.



Docker for Linux

Install Docker on a computer which already has a Linux distribution installed.

Listando containers

```
1 $ sudo docker ps --all
```

```
2 CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

`docker ps --all` lista todos os containers (inclusive os parados).

Primeiro container: hello-world

```
1 $ sudo docker run hello-world
2 Unable to find image 'hello-world:latest' locally
3 latest: Pulling from library/hello-world
4 0e03bdcc26d7: Pull complete
5 Status: Downloaded newer image for hello-world:latest
6
7 Hello from Docker!
8 This message shows that your installation appears to be working correctly.
```

O container executou e saiu

```
1 $ sudo docker ps --all
2 CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS          NAMES
3 5a8a47623ab3   hello-world    "/hello"       3 minutes ago Exited (0) 3 minutes ago jolly_keller
```

Ele rodou, imprimiu a mensagem e terminou — ficando com status Exited (0).

Instalação no Windows (WSL)

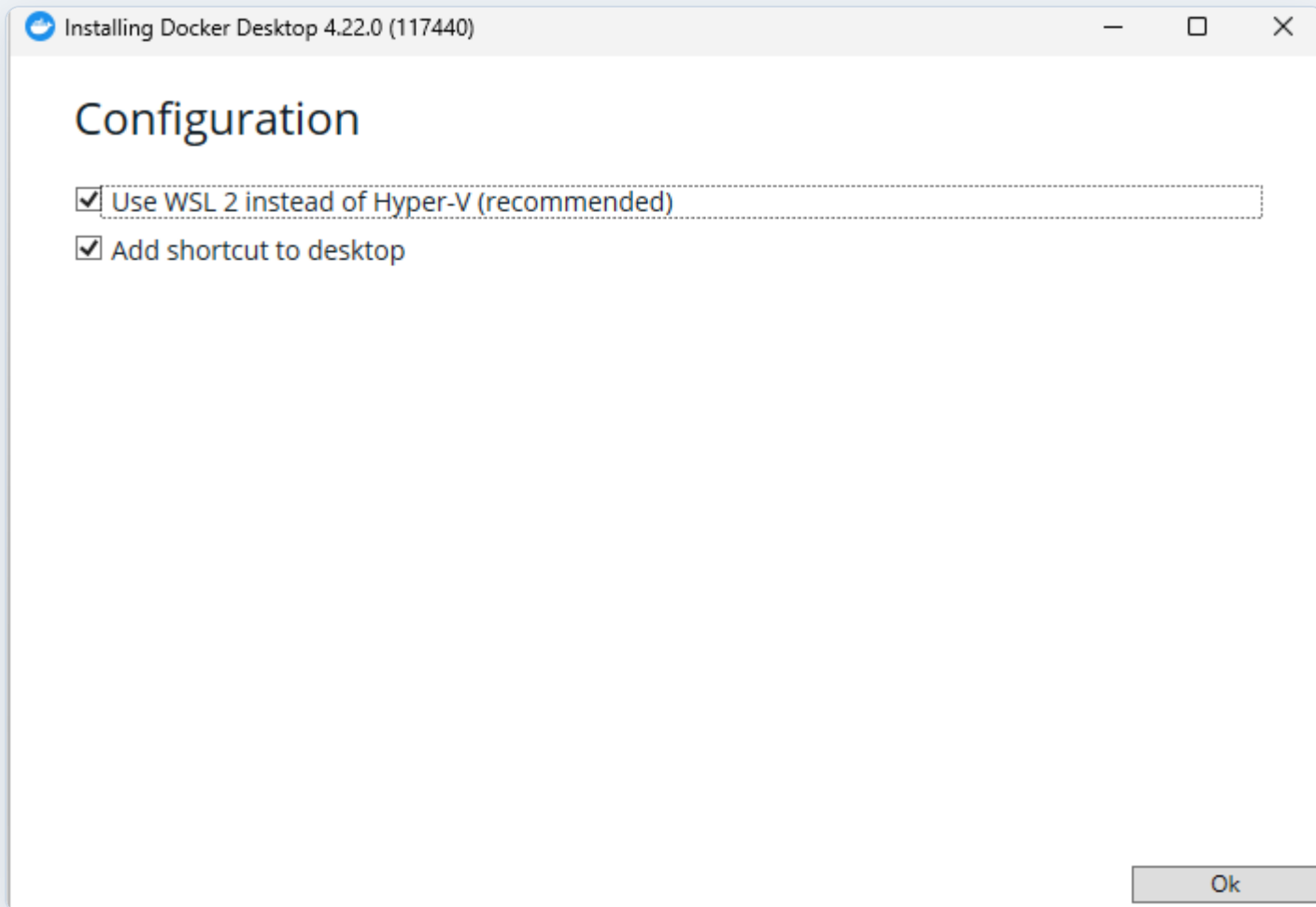
Abra o terminal como administrador e execute:

```
1 PS C:\Users> wsl --install
2 Instalando: Plataforma de Máquina Virtual
3 Instalando: Subsistema do Windows para Linux
4 Instalando: Ubuntu
5 Êxito na operação. As alterações só terão efeito após reiniciar o sistema.
```

IMPORTANTE

Reinicie o computador depois.

Configuração do Docker Desktop



ATIVIDADE

Instalar o Docker

- Instale o **Docker Desktop** (no Windows, instale o WSL).
- Rode o hello world visto em aula: `docker run hello-world`
- Explore os comandos para listar **containers**, **images**, **networks** e **volumes**.

Referências

- docs.docker.com/get-started/overview
- STONEMAN, Elton. Learn Docker in a Month of Lunches. Manning, 2019. manning.com
- Guia Full Cycle — github.com/codeedu/wsl2-docker-quickstart

MATERIAL EM VÍDEO



YOUTUBE O mínimo que você precisa saber sobre Docker! [youtube.com](https://www.youtube.com)

MATERIAL EM VÍDEO



YOUTUBE · FULL CYCLE Docker: o que todo dev precisa saber [youtube.com](https://www.youtube.com)

MATERIAL EM VÍDEO



YOUTUBE Entendendo o Funcionamento de Containers [Docker, Podman, OCI](#)