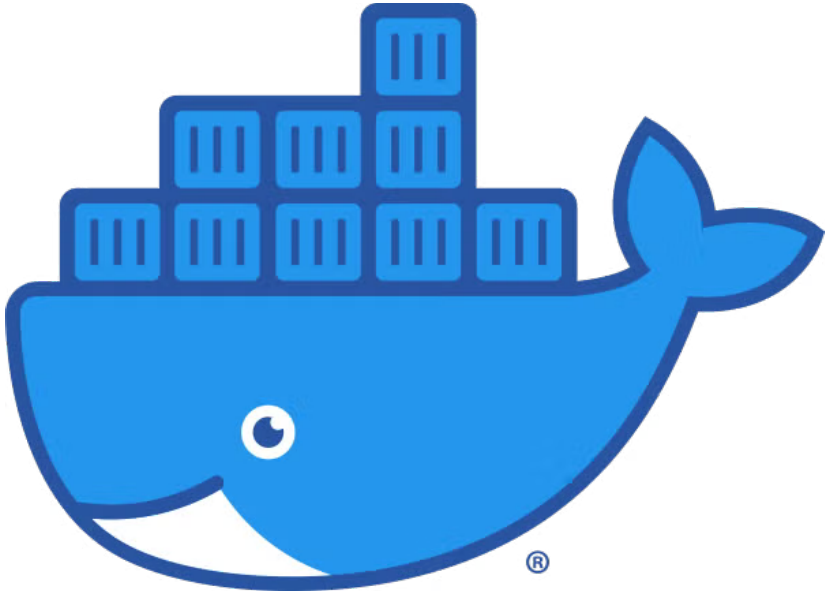
Ambientes Reprodutíveis com Docker

Minicurso Introdutório - Dia 1

Instrutor: Sérgio Fontes
E-mail: fontes.sergio@graduacao.uerj.br

> **Objetivo:** entender os conceitos fundamentais de containers, imagens e isolamento de ambiente, além de executar na prática o primeiro container com Jupyter Notebook.



Por que o mesmo software pode se comportar de forma diferente em ambientes distintos?

- O comportamento de um programa depende do **ambiente** onde é executado (sistema operacional, bibliotecas, versões e configurações)
- Pequenas diferenças nesses elementos podem gerar **resultados diferentes**
- Sem padronização, é difícil **reproduzir o mesmo comportamento** em outra máquina
- Surge a necessidade de **isolar e empacotar o ambiente**

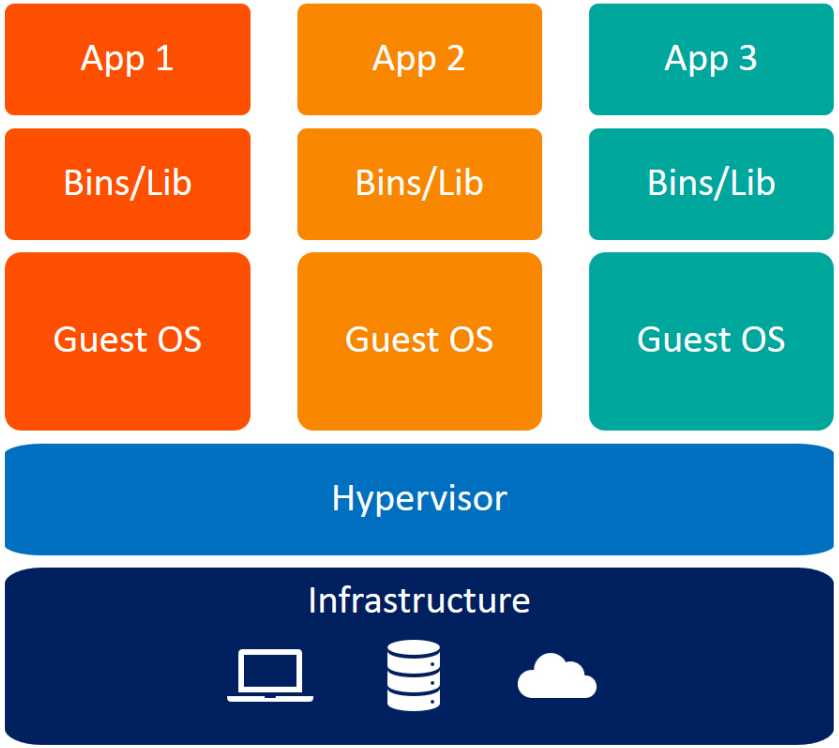
O que é um Container?

- Um **container** é um ambiente isolado que executa processos com suas próprias dependências e configurações
- **Compartilha o kernel** do sistema hospedeiro (em Linux); em **Windows/macOS**, containers rodam em uma **VM com kernel Linux**
- Criado a partir de uma **imagem imutável**, que contém tudo o que o container precisa para executar
- O isolamento é implementado pelo kernel por meio de **namespaces** (separação lógica) e **cgroups** (controle de recursos)
- É **leve, portátil** e inicia em **segundos**, pois não carrega um sistema operacional completo

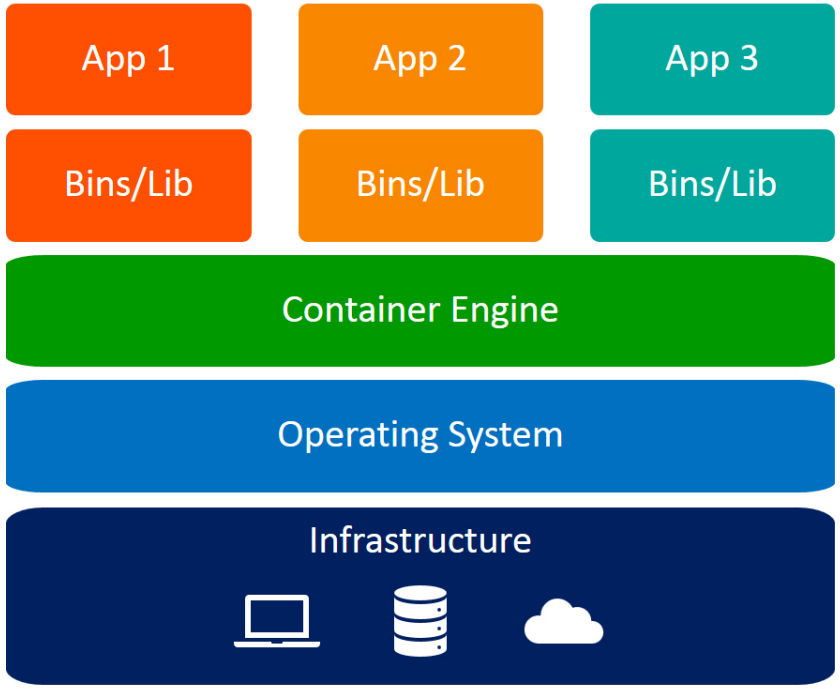
Container × Máquina Virtual

Container	Máquina Virtual
Compartilha o **kernel** do sistema hospedeiro (ou de uma VM Linux)	Executa um **sistema operacional completo**, incluindo kernel e espaço de usuário (*userspace*)
Isola **processos**, **rede** e **sistema de arquivos** via kernel	Isola todo o **hardware virtualizado** (CPU, memória, disco)
Usa **namespaces** e **cgroups** para isolamento	Usa um **hipervisor** para virtualizar hardware
Leve, inicia em segundos	**Pesada**, inicia em minutos
Ideal para **aplicações isoladas**	Ideal para **ambientes multiusuário** ou **de alta segurança**

Container × Máquina Virtual



Virtual Machines



Containers

O que é o Docker?

- Plataforma que **automatiza a criação, execução e gerenciamento de containers**
- Fornece ferramentas para **construir, versionar e distribuir imagens**
- Inclui componentes principais:
  - **Docker Engine:** executa containers no sistema hospedeiro
  - **Docker CLI:** fornece interface de linha de comando
  - **Docker Hub:** repositório público de imagens
- Em **Windows e macOS**, o Docker utiliza uma **VM com kernel Linux** (via **WSL2** ou **HyperKit**)
- Base do conceito: *“Build once, run anywhere”*

Principais Vantagens do Docker

- **Reprodutibilidade:** empacota código, dependências e configurações
  → garante resultados idênticos em qualquer ambiente
- **Portabilidade:** executa em **Linux**, **Windows (WSL2)** e **macOS**
  → sempre sobre um **kernel Linux**, nativo ou virtualizado
- **Leveza:** compartilha o kernel do sistema hospedeiro
  → inicia rapidamente e consome poucos recursos
- **Isolamento:** separa processos, rede e sistema de arquivos
  → evita conflitos entre projetos e facilita testes paralelos
- **Escalabilidade:** replica containers idênticos em múltiplos hosts
  → ideal para **clusters**, **pipelines** e **ambientes em nuvem**