



# SITEC 2024



# Docker Course: criando um mundo dos sonhos usando docker

Lucas Costa, Andrey Adailso and Heitor Anglada  
November 7, 2024

## Curso de Docker do Lasse



## Link para o Material da Aula no Github



# O que é Docker?

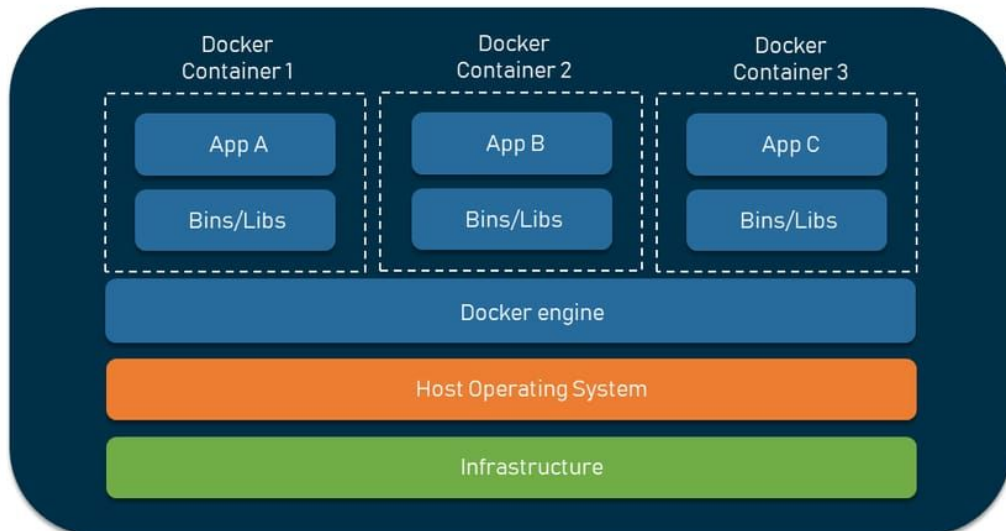
- Docker é uma plataforma de código aberto que permite aos desenvolvedores automatizar a implantação, escalonamento e gerenciamento de aplicações em contêineres.
- Docker simplifica o processo de criação, implantação e execução de aplicações ao utilizar contêineres, garantindo que elas funcionem de forma consistente em diferentes ambientes de computação.



# Docker vs Container

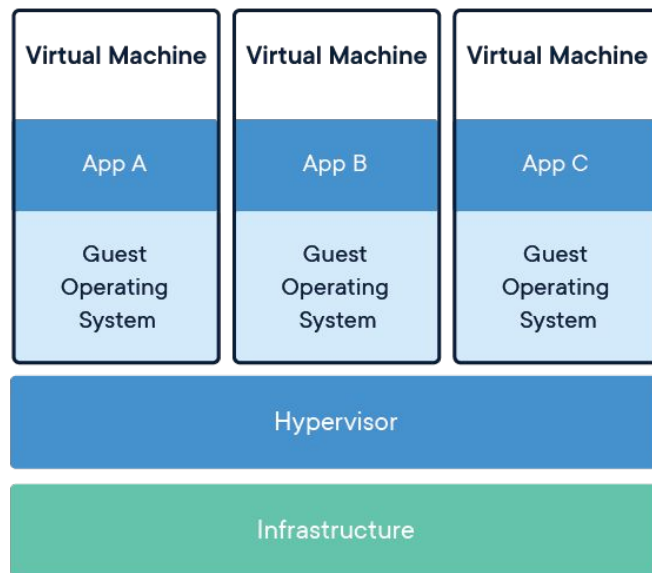
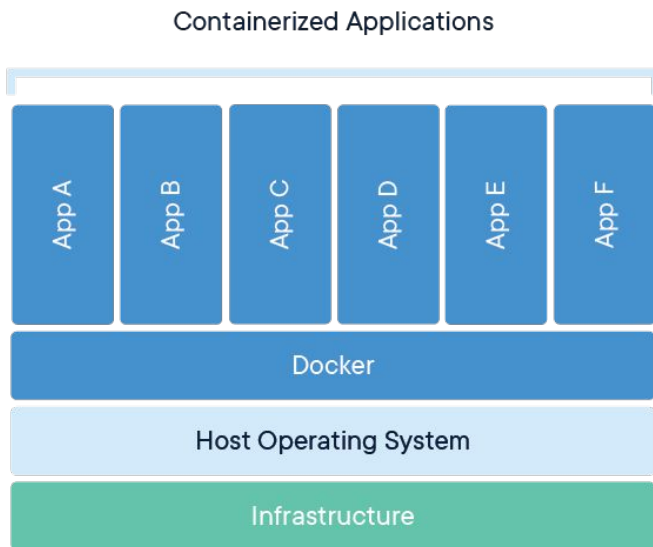
Um contêiner é uma unidade de software que empacota o código de uma aplicação e todas as suas dependências, como bibliotecas, ferramentas do sistema e configurações, para garantir que a aplicação seja executada de forma uniforme em qualquer ambiente de computação.

DOCKER CONTAINERS

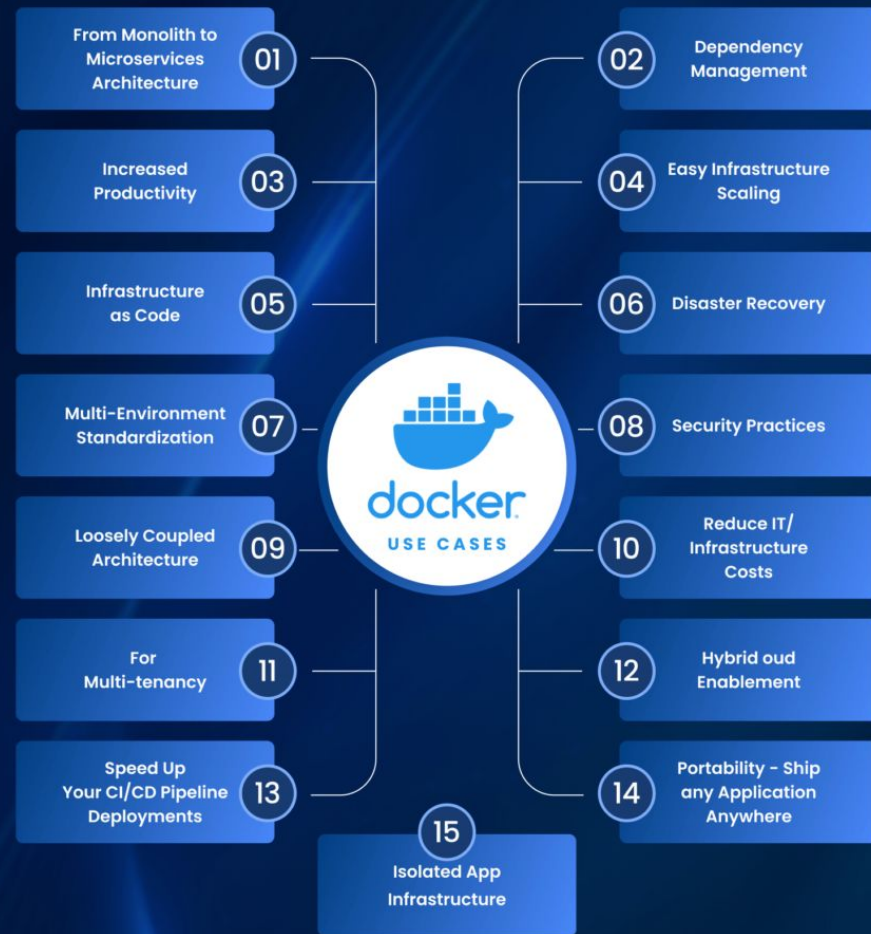


# Docker vs Virtual Machine

Uma máquina virtual (VM) é uma emulação de um sistema de computador que permite que múltiplos sistemas operacionais (OS) rodem em um único hardware físico. Cada VM inclui um sistema operacional completo, seus próprios drivers, bibliotecas e uma cópia completa do kernel.



# Principais Aplicações



# Vocês tem duas opções

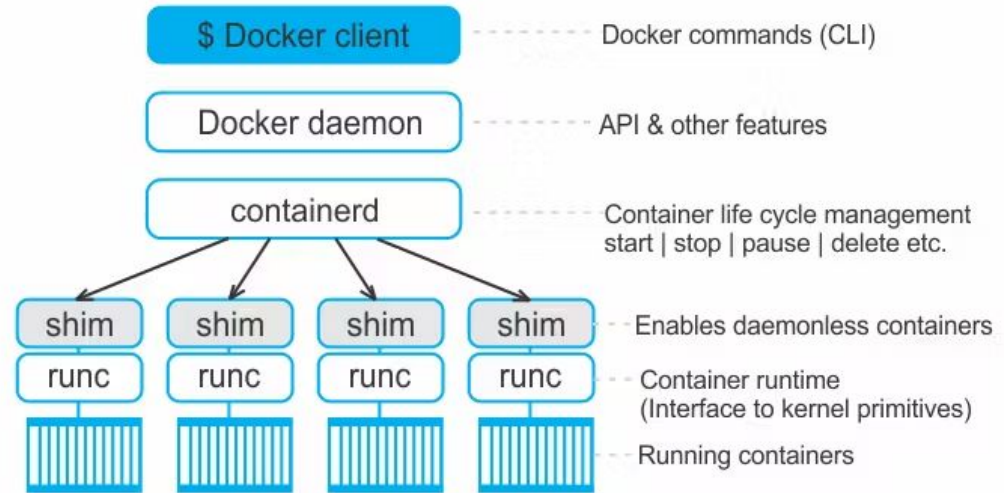
- **Docker Desktop**
- **Docker Engine**

# Docker Desktop Vs Docker Engine

Docker Engine e Docker Desktop são ferramentas de software desenvolvidas pela Docker Inc., uma empresa que fornece ferramentas de código aberto para trabalhar com contêineres.

O Docker Engine é o núcleo do Docker. O Docker Engine recebe comandos do cliente Docker e gerencia objetos Docker como imagens, contêineres, redes e volumes. Ele também é responsável por construir e executar contêineres Docker.

## Docker Engine Architecture

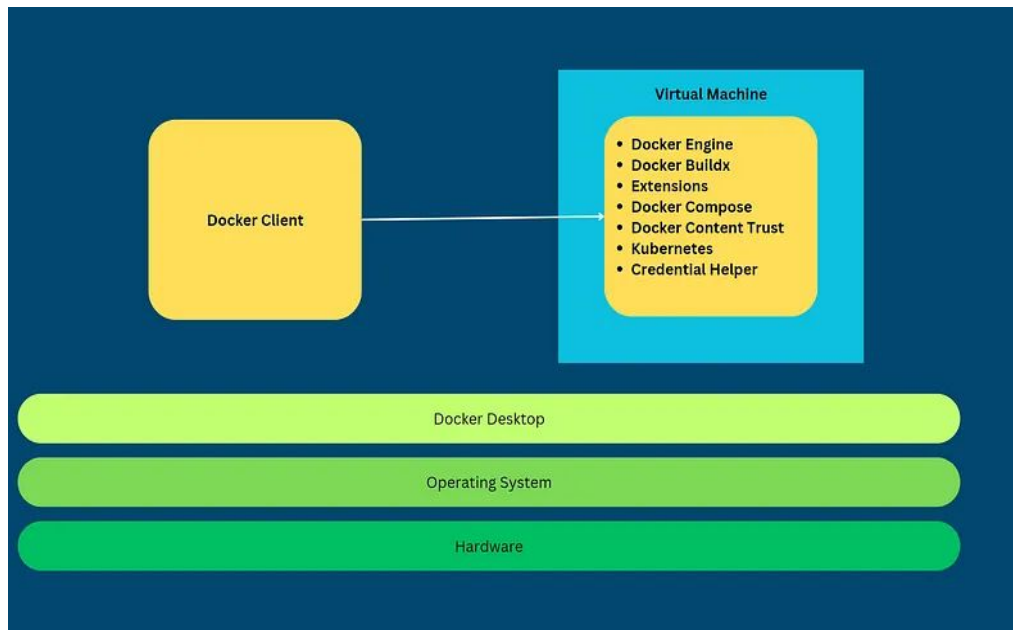




# Docker Desktop Vs Docker Engine

O Docker Desktop é um aplicativo para máquinas MacOS e Windows para a construção e compartilhamento de aplicativos em contêineres. Ele inclui o Docker Engine, o cliente Docker CLI, o Docker Compose e o Kubernetes.

Ele fornece uma interface gráfica de usuário (GUI) para trabalhar com o Docker, o que o torna mais amigável do que usar o Docker Engine sozinho. Com o Docker Desktop, você pode facilmente iniciar, parar e gerenciar contêineres Docker, e ele vem pré-configurado para trabalhar com seu sistema local.



# Como instalar docker desktop

Install docker on Windows



Install docker on MacOS



# Como instalar Docker Engine

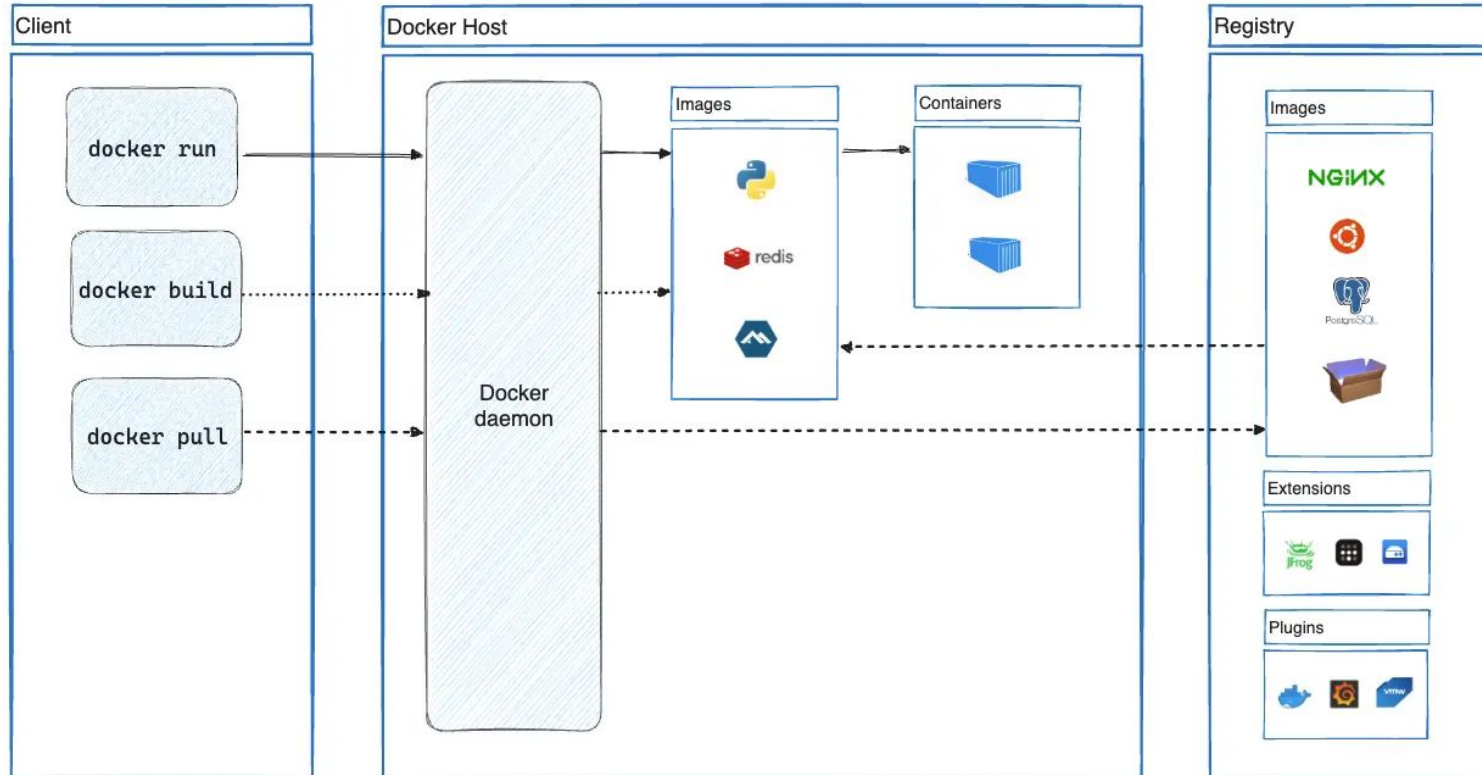
## [Install Docker in Linux](#)

### Supported platforms

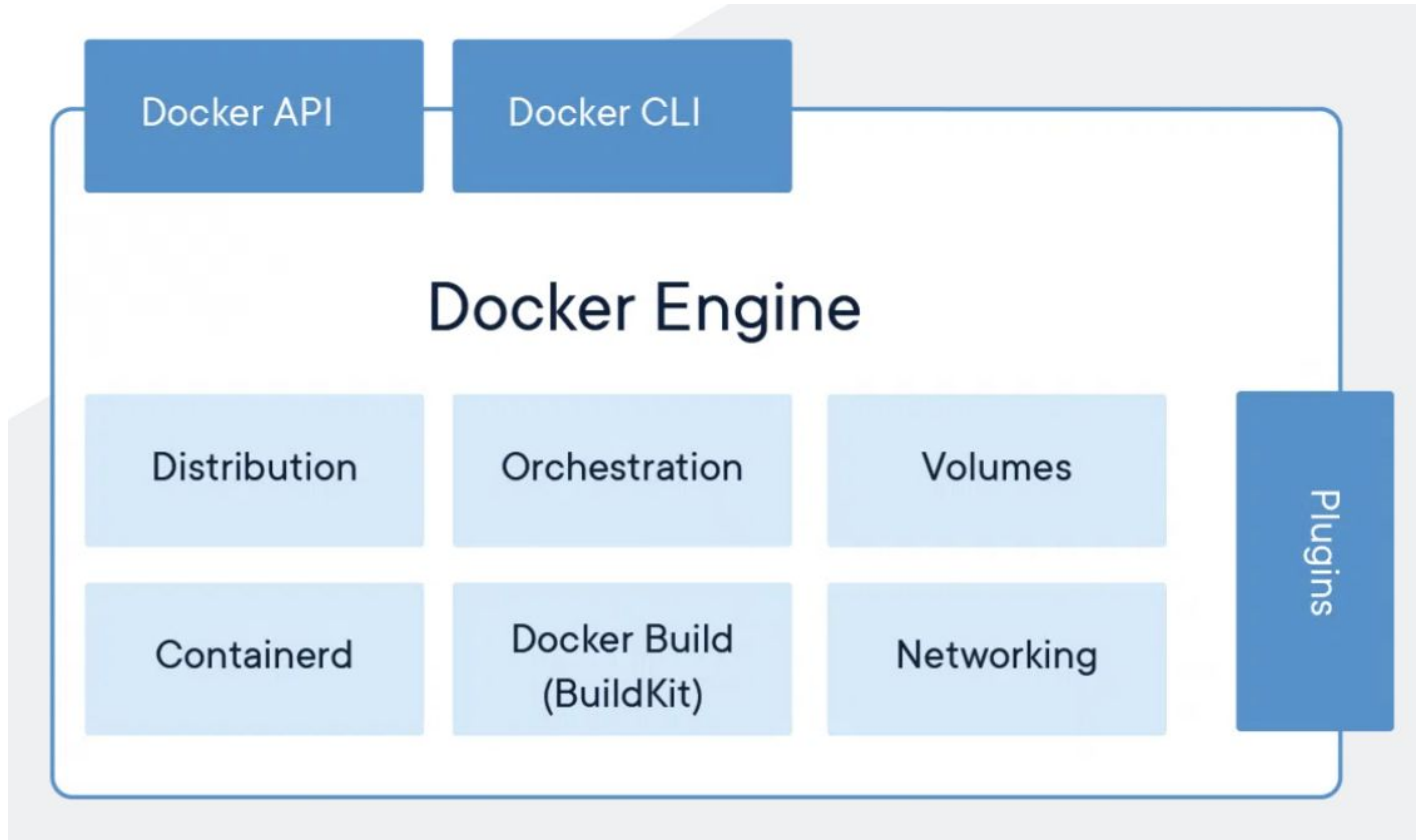
Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	ppc64le	s390x
<a href="#">CentOS</a>	✓	✓		✓	
<a href="#">Debian</a>	✓	✓	✓	✓	
<a href="#">Fedora</a>	✓	✓		✓	
<a href="#">Raspberry Pi OS (32-bit)</a>			✓		
<a href="#">RHEL</a>	⚠	⚠			✓
<a href="#">SLES</a>					✓
<a href="#">Ubuntu</a>	✓	✓	✓	✓	✓
<a href="#">Binaries</a>	✓	✓	✓		

⚠ = Experimental

# Docker architecture



# Docker Engine



# Docker Objects

**Image:** A container image is a snapshot of a filesystem that includes everything the application needs to run, such as application code, libraries, system tools, and settings. Images are immutable and serve as a template from which containers are created.

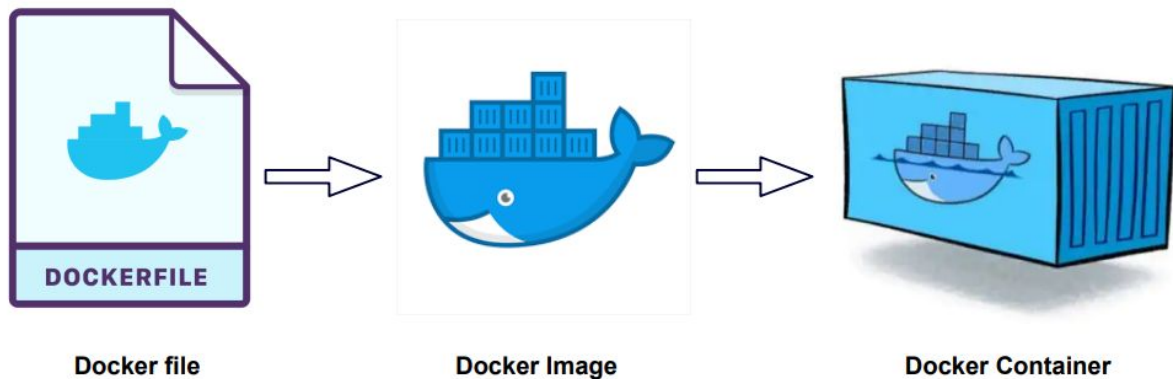
**Container:** A container is a running instance of an image. It adds a layer of abstraction over the host operating system, isolating the application and its dependencies from the rest of the system.

**Docker Hub:** It is a container registry built for developers and open source contributors to find, use, and share their container images.

# Conceitos importantes

## Dockerfile

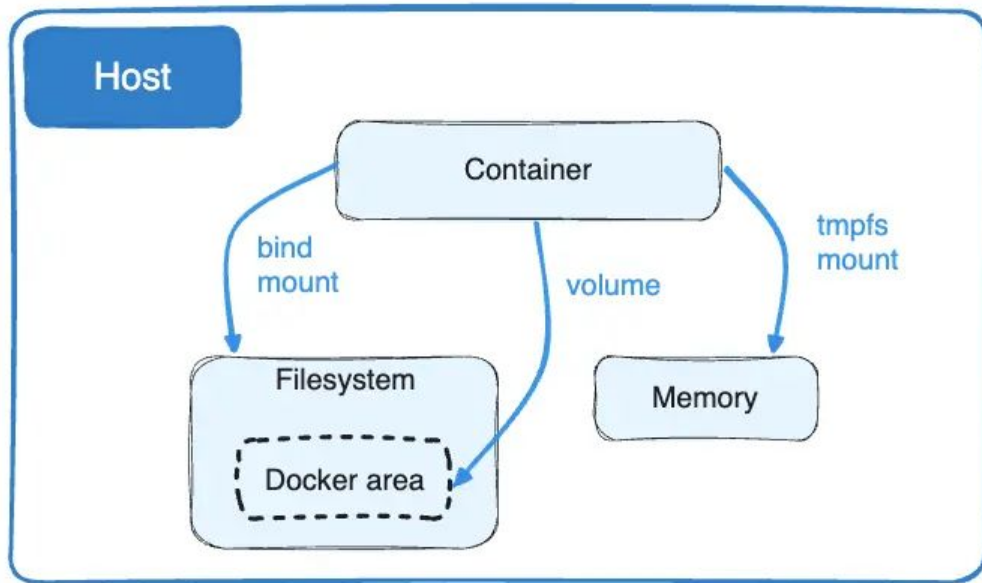
Um Dockerfile é um script que contém uma série de instruções sobre como construir uma imagem Docker. Cada instrução em um Dockerfile cria uma camada na imagem, tornando a imagem final uma pilha dessas camadas.



# Conceitos importantes

## Volumes

Volumes são usados no Docker para persistir dados gerados e usados por contêineres. Eles fornecem uma maneira de armazenar dados fora do sistema de arquivos do contêiner, o que significa que os dados podem ser preservados mesmo se o contêiner for removido. Volumes podem ser compartilhados entre vários contêineres e podem ser gerenciados usando comandos do Docker.

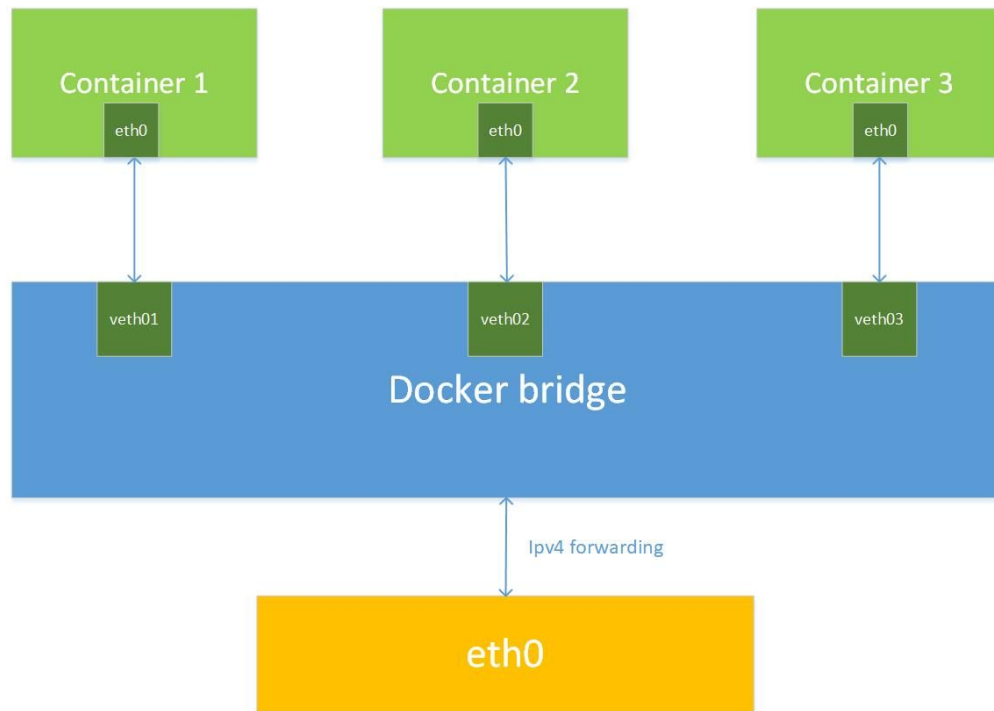




# Conceitos importantes

## Networks

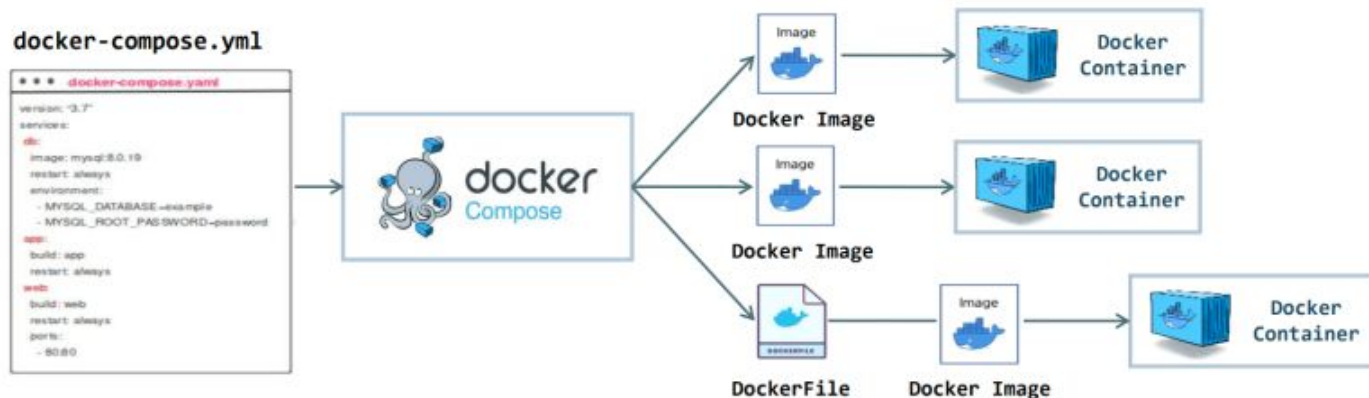
As redes Docker permitem a comunicação entre contêineres Docker. O Docker fornece vários drivers de rede para gerenciar a rede de contêineres



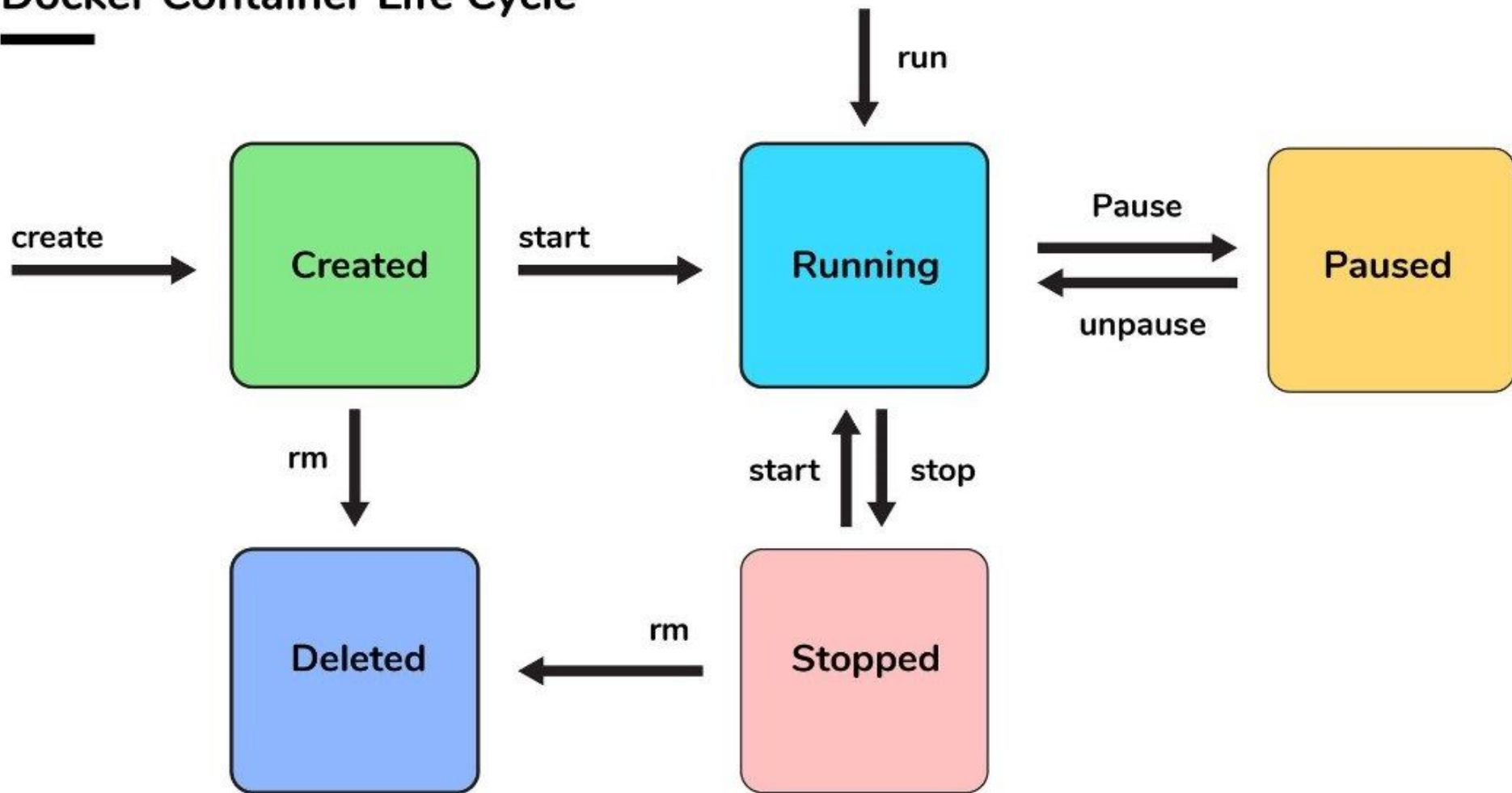
# Conceitos importantes

## Docker-compose

O Docker Compose é uma ferramenta para definir e executar aplicativos Docker multi-contêineres. Com um arquivo `docker-compose.yml`, você pode definir os serviços, redes e volumes necessários para seu aplicativo e gerenciá-los com comandos simples



# Docker Container Life Cycle



# Comandos Básicos

## Commands for Containers

**docker run:** criar e iniciar um container de uma imagem

**docker ps:** listar todos os containers

**docker ps -a:** listar todos os containers, inclusive os parados

**docker exec:** executar um container que esteja rodando em fundo

**docker stop:** parar um container que está rodando

**docker start:** iniciar um container que esteja parado

**docker restart:** reiniciar o container

**docker rm:** remover o container

**docker logs:** mostrar o log de um container

# Comandos Básicos

## Commands for Images:

**docker pull:** Baixar imagem de um registrador

**docker images:** listar todas as imagens

**docker rmi:** remover uma imagem localmente

## Commands for Volumes:

**docker volume create:** criar os volumes

**docker volume ls:** listar todos os volumes

**docker volume rm:** remover um volume

# Comandos básicos

## Commands for Networks

**docker network create:** criar redes

**docker network ls:** listar as redes

**docker network rm:** remover uma rede

**docker network connect:** conectar um container a uma rede

**docker network disconnect:** desconectar uma rede a um container

## 3 - Exemplos

Jupyter in Docker:

<https://docs.docker.com/guides/jupyter/>

web-service

LLM in docker

HTTP routing in Docker:

<https://docs.docker.com/guides/traefik/>

Pre-processing database in Docker:

<https://docs.docker.com/guides/pre-seeding/>



**Obrigado pela sua  
atenção!**