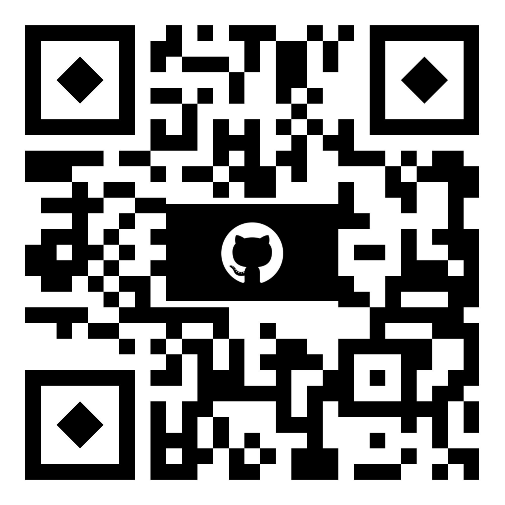
Docker Course: criando um mundo dos sonhos usando docker

Lucas Costa, Andrey Adailso and Heitor Anglada

November 7, 2024

Curso de Docker do Lasse

Link para o Material da Aula no Github

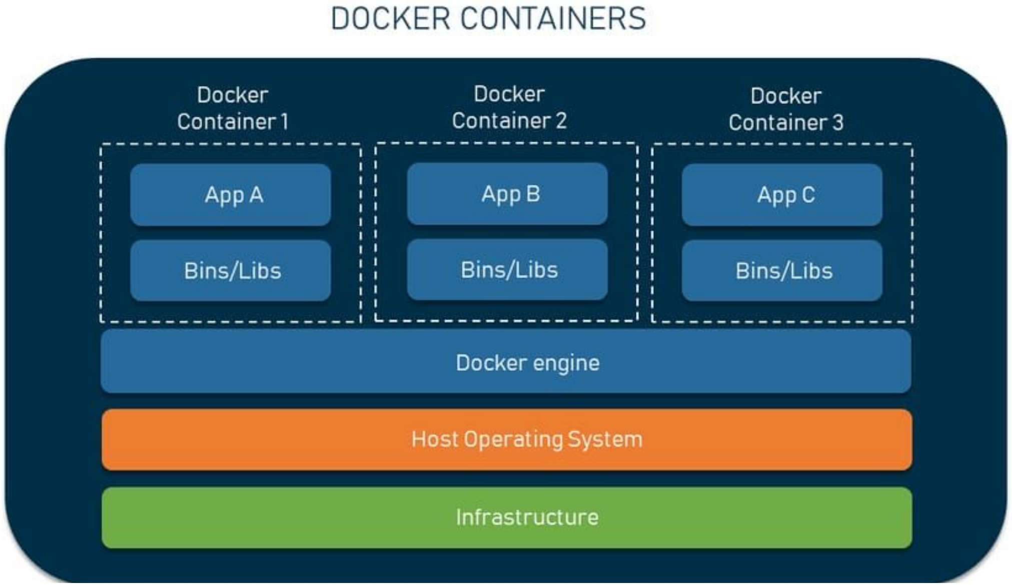


O que é Docker?

- 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.

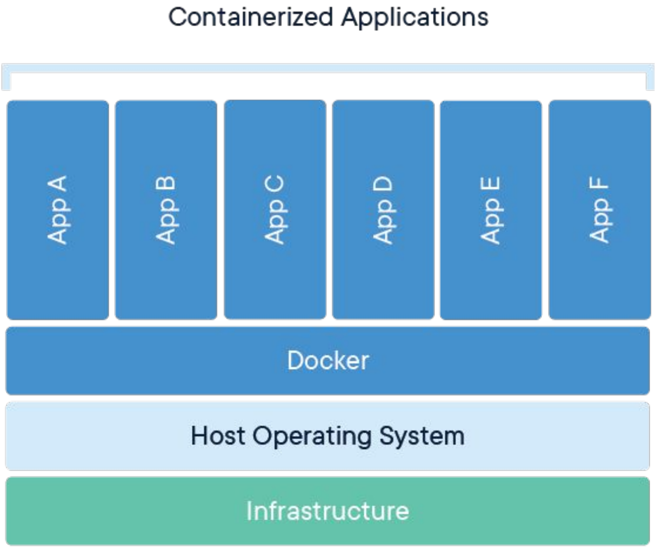
Docker vs Container

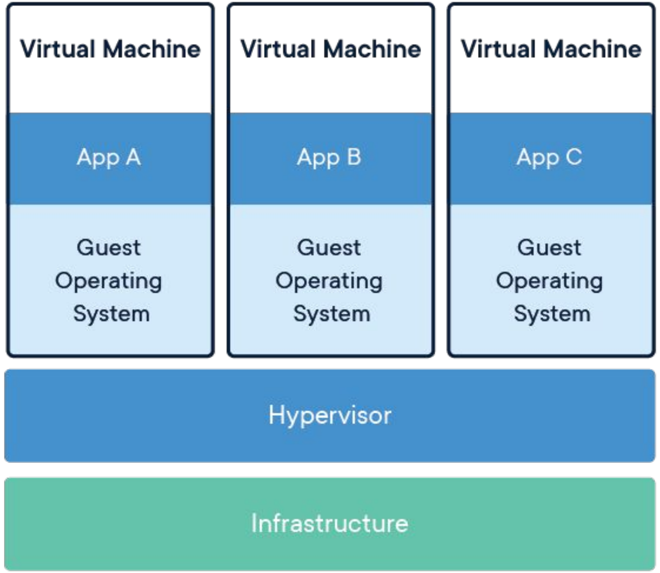
Um contêiner é uma unidade de software que empacota o código de uma aplicação e todas as suas dependências, como bibliotecas, ferramentas do sistema e configurações, para garantir que a aplicação seja executada de forma uniforme em qualquer ambiente de computação.



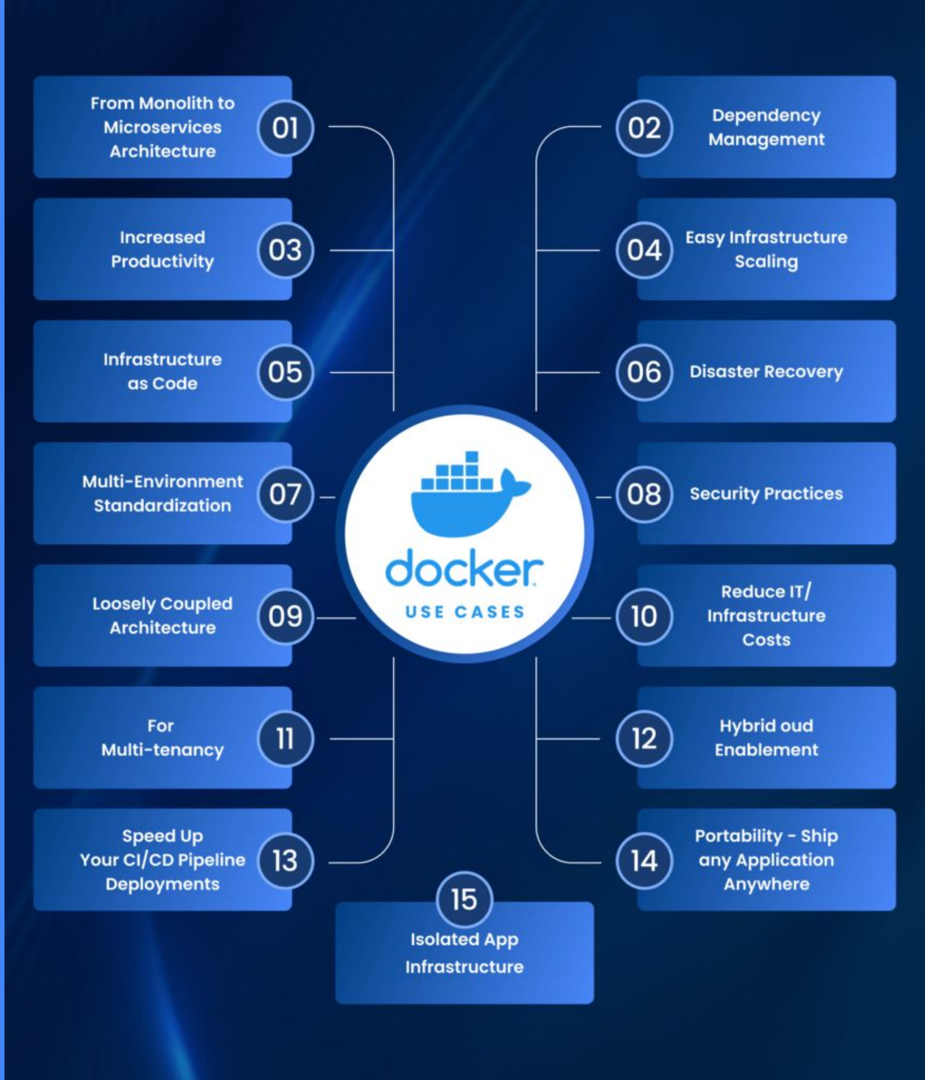
Docker vs Virtual Machine

A virtual machine (VM) is an emulation of a computer system that allows multiple operating systems (OS) to run on a single physical hardware. Each VM includes a complete operating system, its own drivers, libraries, and a full copy of the kernel.





MAIN APPLICATION



docker

USE CASES

01 From Monolith to Microservices Architecture

02 Dependency Management

03 Increased Productivity

04 Easy Infrastructure Scaling

05 Infrastructure as Code

06 Disaster Recovery

07 Multi-Environment Standardization

08 Security Practices

09 Loosely Coupled Architecture

10 Reduce IT/ Infrastructure Costs

11 For Multi-tenancy

12 Hybrid oud Enablement

13 Speed Up Your CI/CD Pipeline Deployments

14 Portability - Ship any Application Anywhere

15 Isolated App Infrastructure

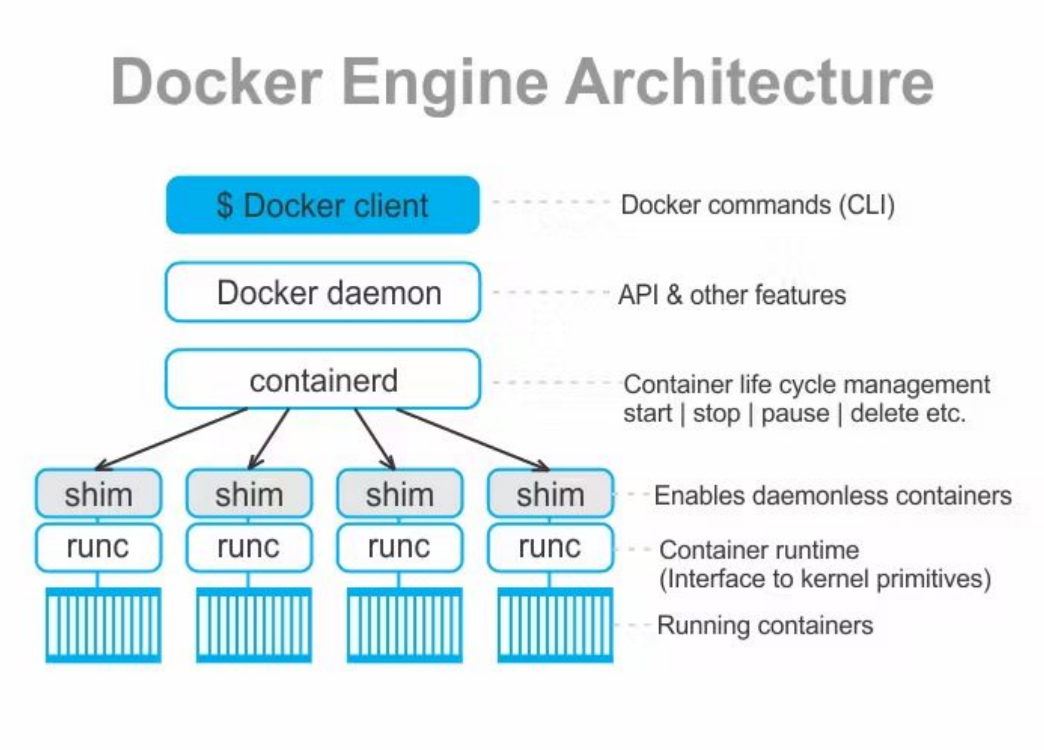
When installing docker you will find two options

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

Docker Desktop Vs Docker Engine

Docker Engine and Docker Desktop are both software tools developed by Docker Inc., a company that provides open-source tooling for working with containers.

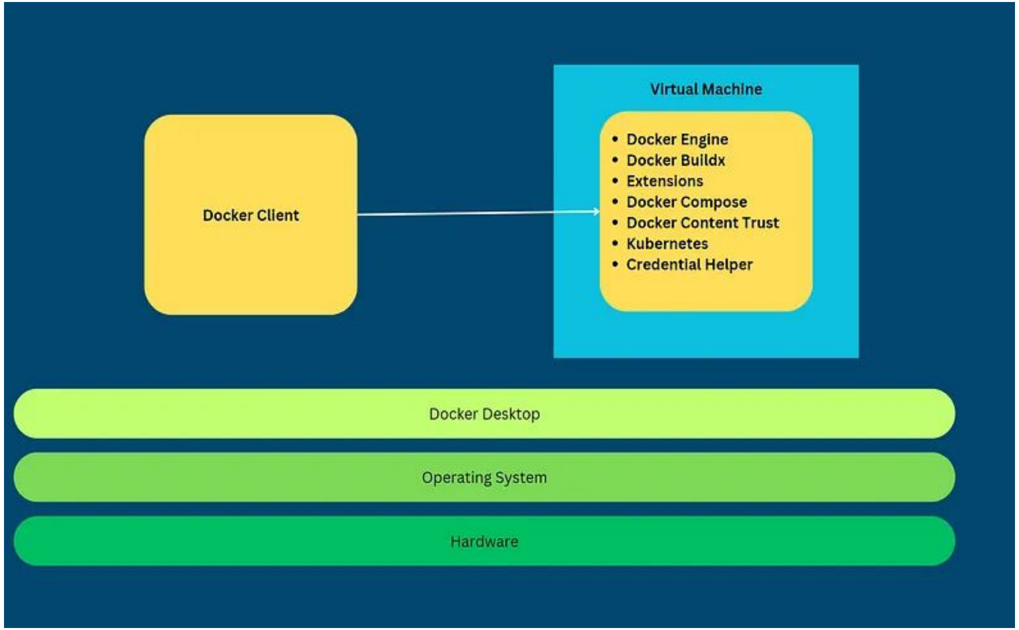
Docker Engine is the core of Docker. Docker Engine takes in commands from the Docker client, and manages Docker objects like images, containers, networks, and volumes. It’s also responsible for building and running Docker containers.


Docker Engine Architecture
$ Docker client
Docker commands (CLI)
Docker daemon
API & other features
containerd
Container life cycle management
start | stop | pause | delete etc.
shim
shim
shim
shim
Enables daemonless containers
runc
runc
runc
runc
Container runtime
(Interface to kernel primitives)
Running containers

Docker Desktop Vs Docker Engine

Docker Desktop is an application for MacOS and Windows machines for the building and sharing of containerized applications. It includes Docker Engine, Docker CLI client, Docker Compose, and Kubernetes.

It provides a graphical user interface (GUI) to work with Docker, which makes it more user-friendly than using the Docker Engine alone. With Docker Desktop, you can easily start, stop, and manage Docker containers, and it comes preconfigured to work with your local system.



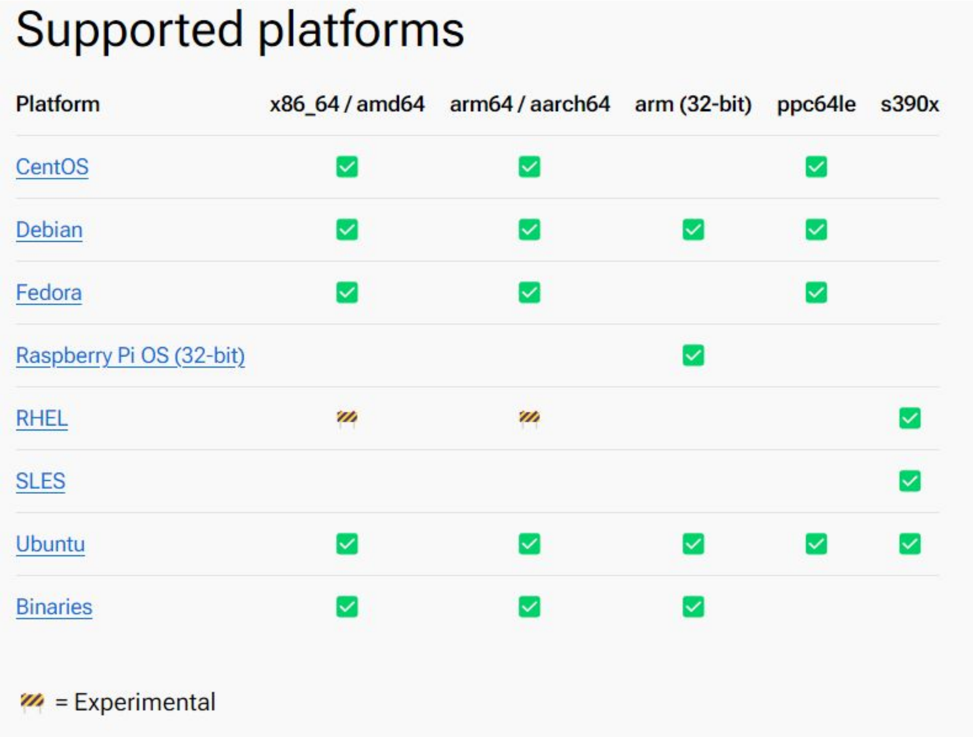
How to install Docker Desktop

Install docker on Windows

Install docker on MacOS

How to install Docker Engine

Install Docker in Linux

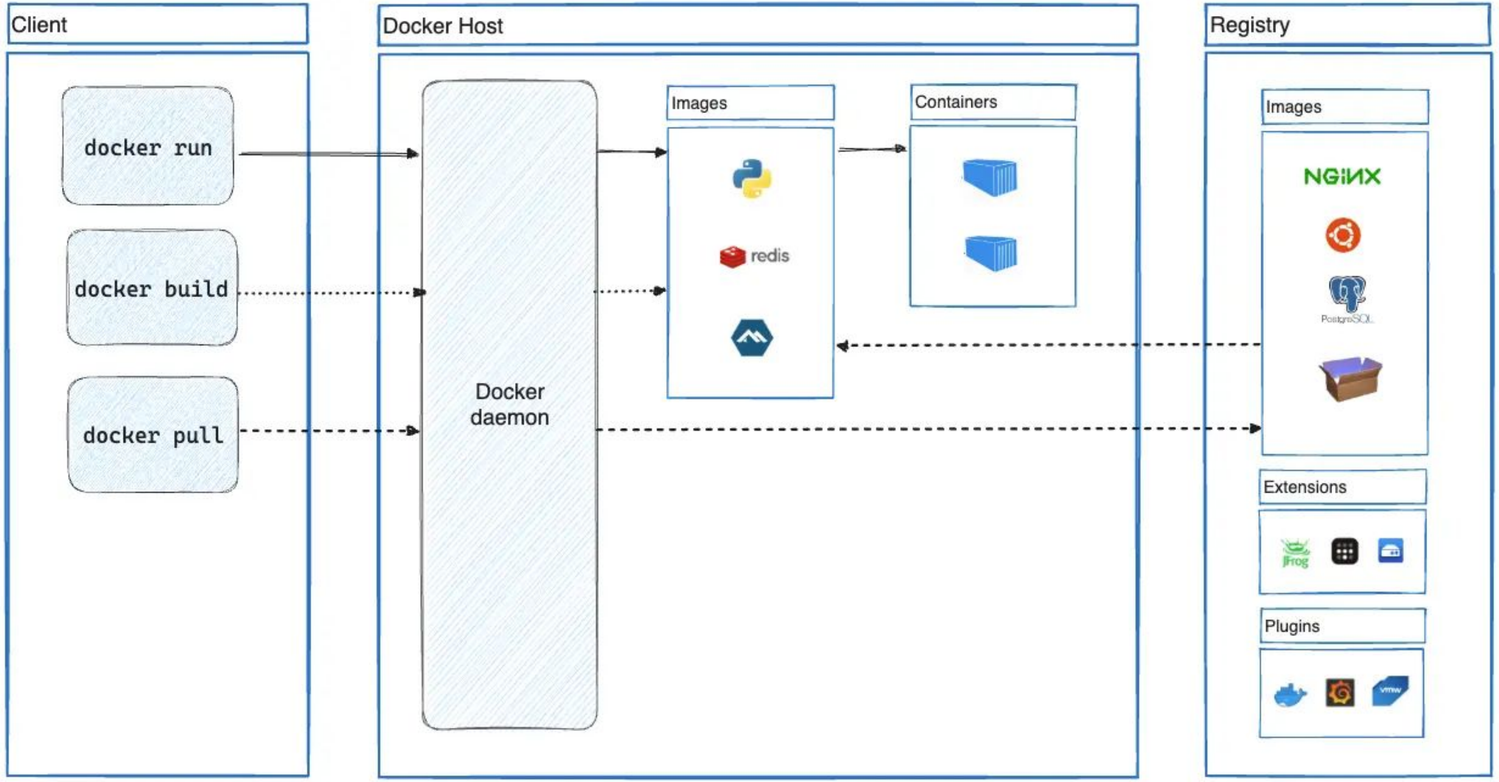


Supported platforms

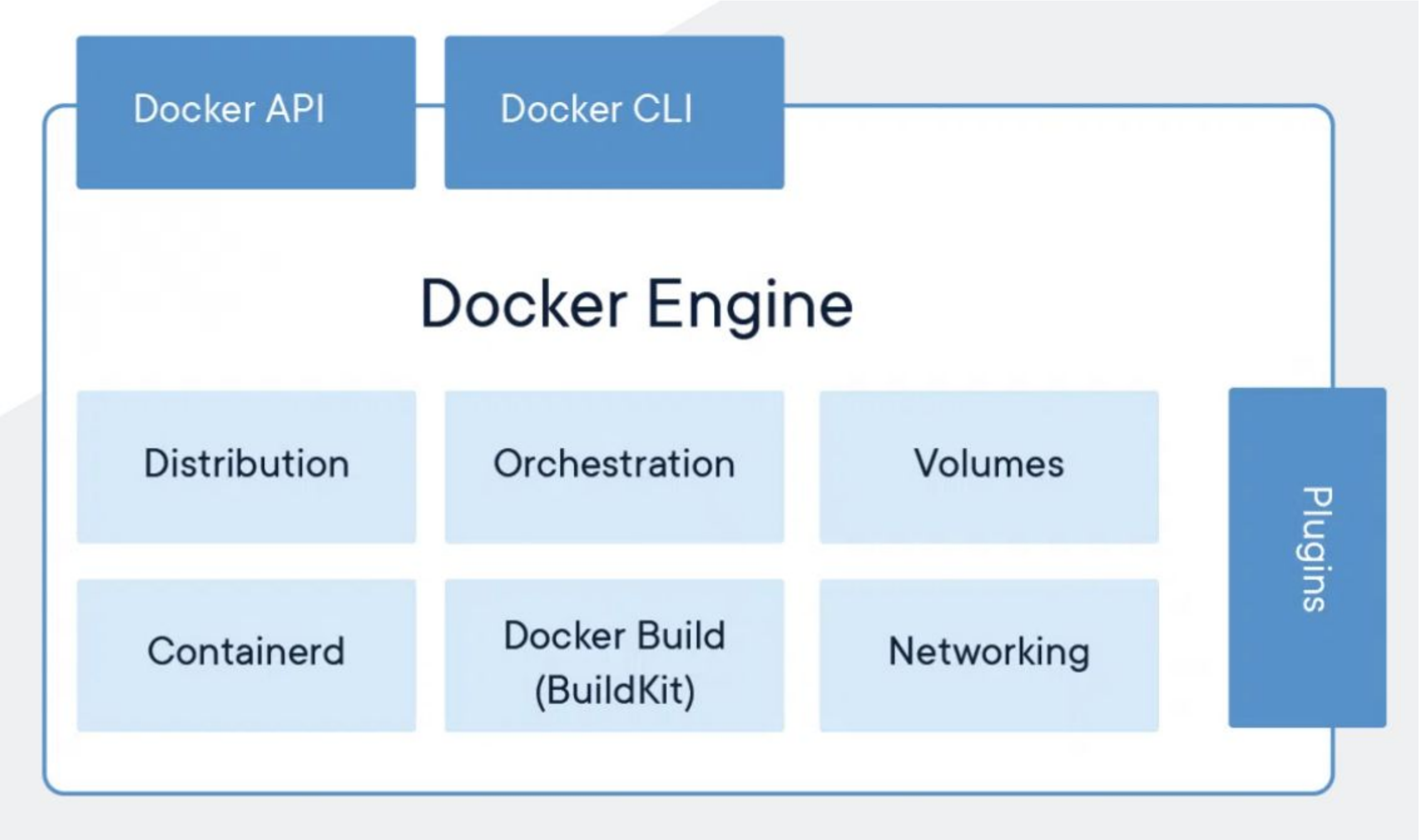
Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	ppc64le	s390x
CentOS	✅	✅		✅	
Debian	✅	✅	✅	✅	
Fedora	✅	✅		✅	
Raspberry Pi OS (32-bit)			✅		
RHEL	🚧	🚧			✅
SLES					✅
Ubuntu	✅	✅	✅	✅	✅
Binaries	✅	✅	✅		

🚧 = Experimental

Docker architecture



Docker Engine



Docker API

Docker CLI

Docker Engine

Distribution

Orchestration

Volumes

Containerd

Docker Build (BuildKit)

Networking

Plugins

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.

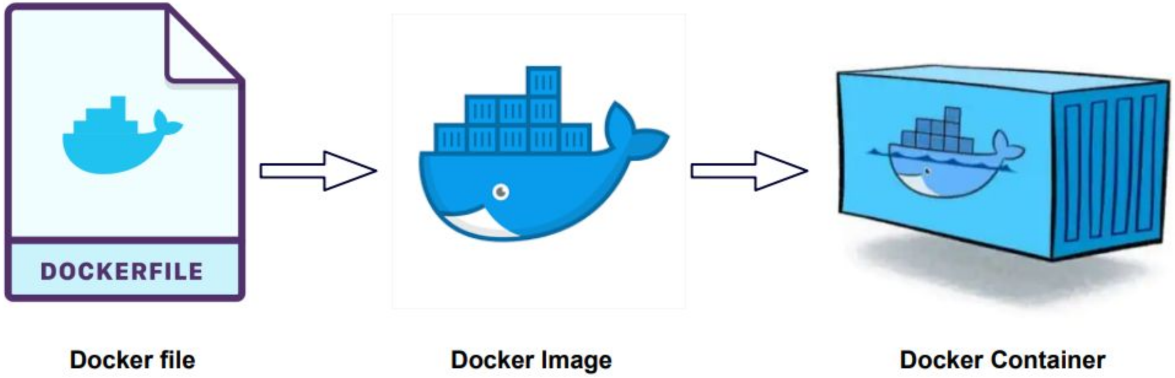
Container: A container is a running instance of an image. It adds a layer of abstraction over the host operating system, isolating the application and its dependencies from the rest of the system.

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

Important Docker concepts

Dockerfile

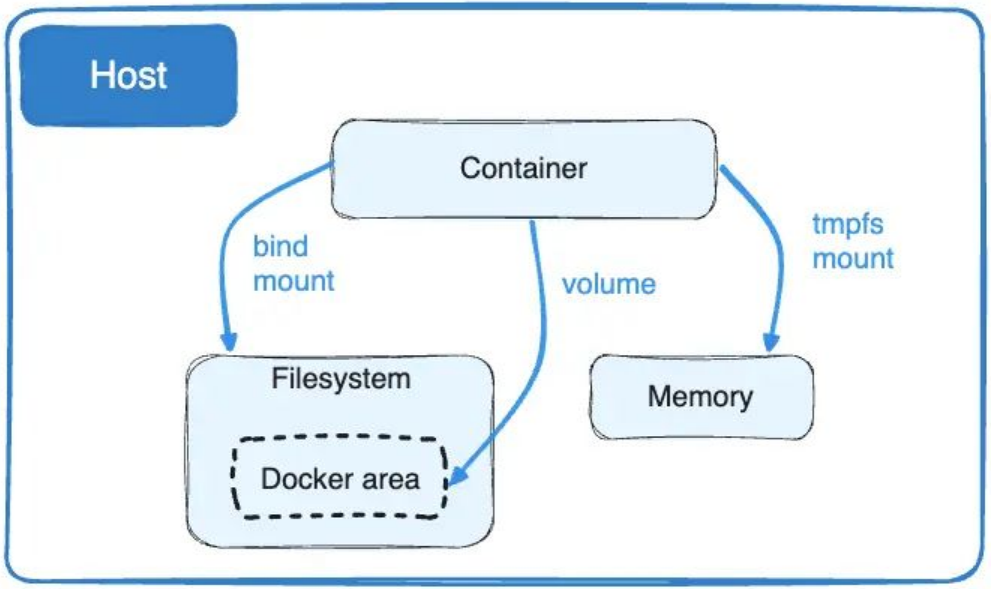
A Dockerfile is a script that contains a series of instructions on how to build a Docker image. Each instruction in a Dockerfile creates a layer in the image, making the final image a stack of these layers.



Important Docker concepts

Volumes

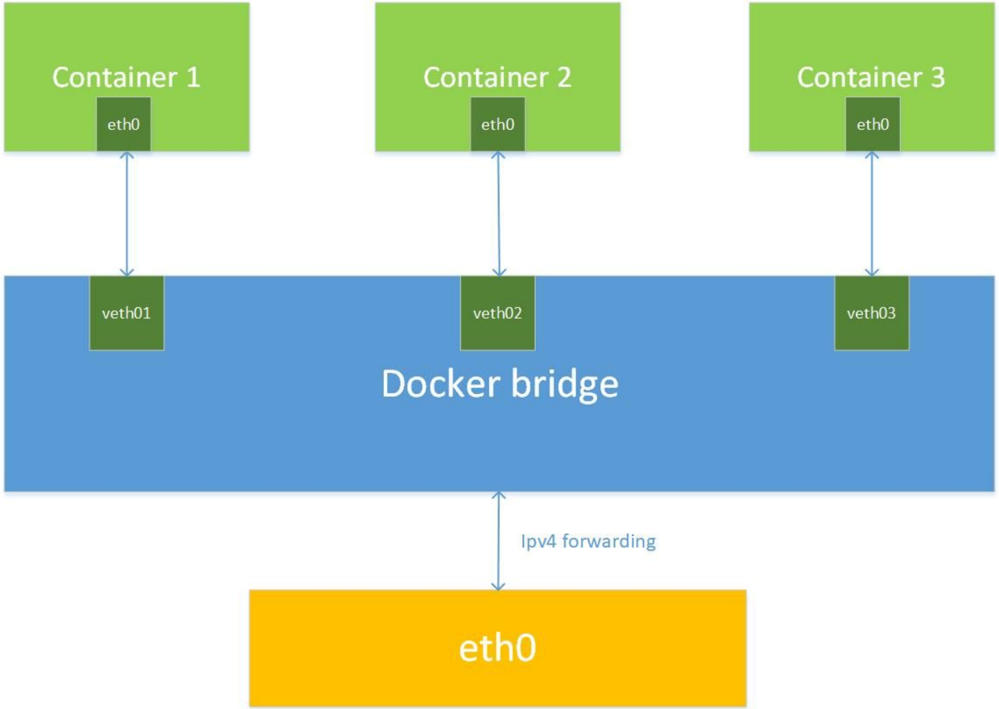
Volumes are used in Docker to persist data generated and used by containers. They provide a way to store data outside the container's filesystem, which means the data can be preserved even if the container is removed. Volumes can be shared between multiple containers and can be managed using Docker commands.



Important Docker concepts

Networks

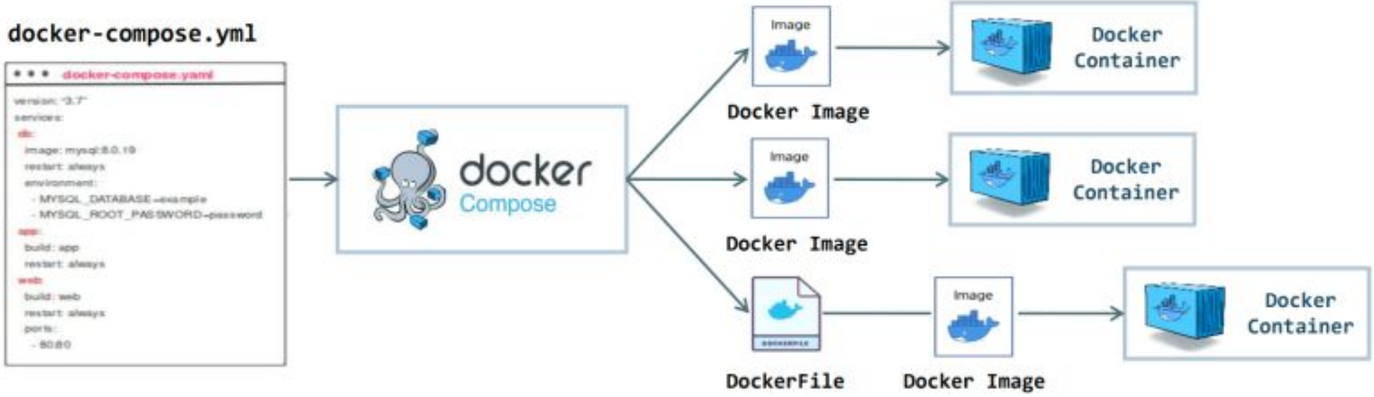
Docker networks enable communication between Docker containers. Docker provides several network drivers to manage container networking



Important Docker concepts

Docker-compose

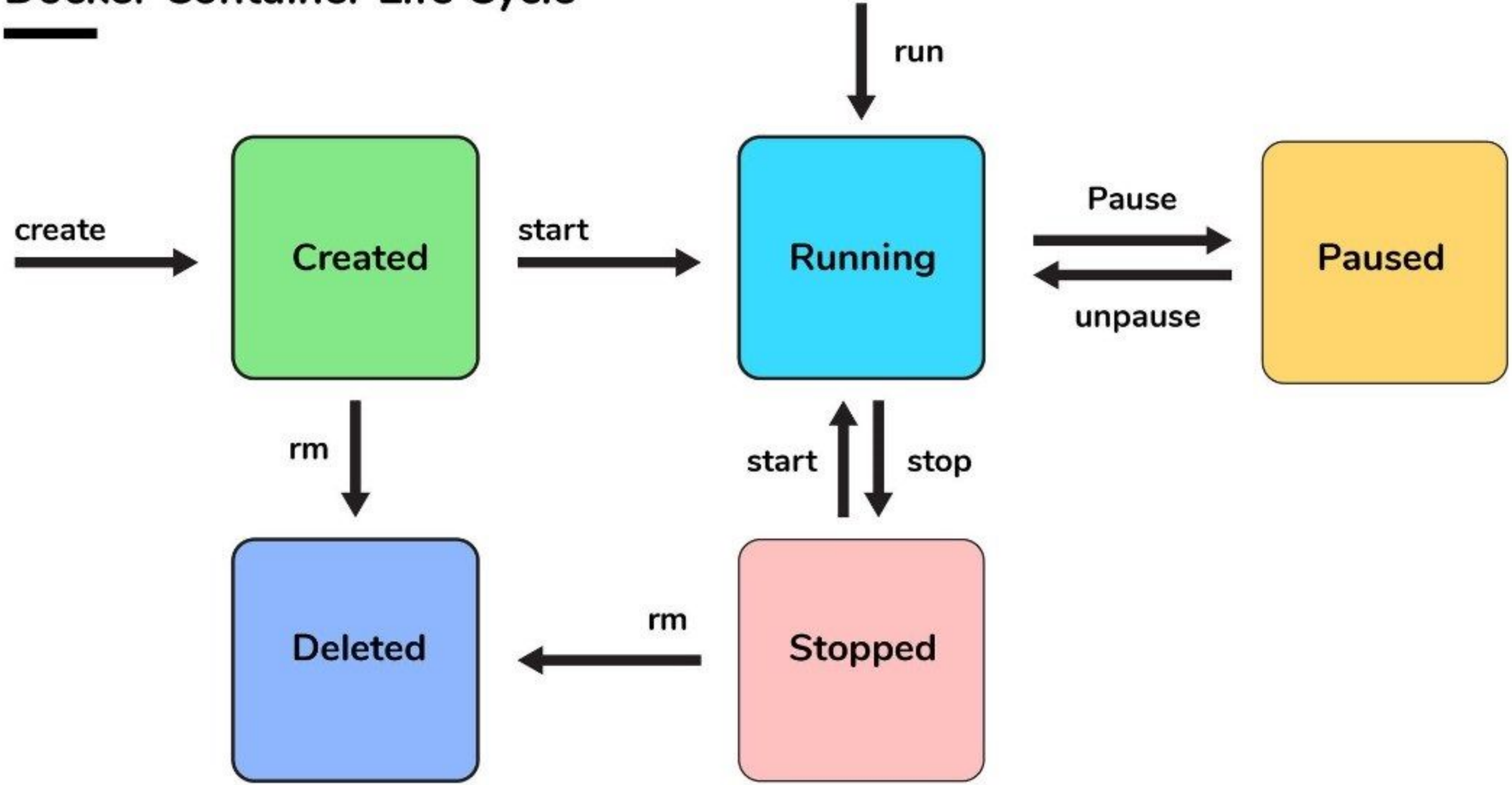
Docker Compose is a tool for defining and running multi-container Docker applications. With a docker-compose.yml file, you can define the services, networks, and volumes needed for your application and manage them with simple commands



docker-compose.yml

```
docker-compose.yaml
version: "3.7"
services:
 db:
  image: mysql:8.0.19
  restart: always
  environment:
   - MYSQL_DATABASE=example
   - MYSQL_ROOT_PASSWORD=password
 app:
  build: app
  restart: always
 web:
  build: web
  restart: always
  ports:
   - 80:80
```

Docker Container Life Cycle



Comandos Básicos

Commands for Containers

docker run: Creates and starts a new container from an image

docker ps: Lists all running containers

docker ps -a: Lists all containers, including stopped ones

docker exec: Runs a command inside a running container

docker stop: Stops a running container

docker start: Starts a stopped container

docker restart: Restarts a container

docker rm: Removes a container

docker logs: Displays logs from a container

Comandos Básicos

Commands for Images:

docker pull: Downloads an image from Docker Hub or another registry

docker images: Lists all locally available images

docker rmi: Removes a local Docker image

Commands for Volumes:

docker volume create: Creates a volume

docker volume ls: Lists all volumes

docker volume rm: Removes a volume

Comandos básicos

Commands for Networks

docker network create: Creates a network.

docker network ls: Lists all networks.

docker network rm: Removes a network.

docker network connect: Connects a container to a network.

docker network disconnect: Disconnects a container from a network.

3 - Exemplos

Jupyter in Docker:
https://docs.docker.com/guides/jupyter/

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!