Club Informatique
& Télécom

Docker 101

Understanding Containerization

CRAFTED carefully

By IKBI Abdelilah

رمضان


NO SIGNAL

Club Informatique
& Télécom

Plan

Introduction
The Evolution of Application Deployment
Understanding Containers & Docker
How Docker Works
Building and Running Applications

The old days...

Most applications run on servers, and in the past, we were limited to running one application per server.

Hello VMware !

Which allows to run multiple business applications on a single server.

Buuuuuuuuuuuut,

The old days...

A feature of the VM model is every VM needing its own dedicated operating system (OS).

Club Informatique
& Télécom

Hello Containers!

oiiaiioiiiiaiii

oiiaiioiiiiaiii

oiiaiioiiiiaiii

oiiaiioiiiiaiii

Hello Containers!

A feature of the container model is that *every container shares the OS of the host it's running on.* *This means a single host can run more containers than VMs.*

For example, a host that can run 10 VMs might be able to run 50 containers, making containers far more efficient than VMs.

Hello Docker !

Docker made containers easy and brought them to the masses! because before Docker containers wer complicated in use...

The solution is the Blue Whale XD

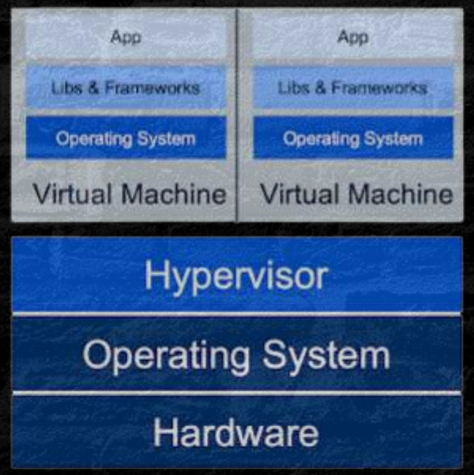
> *"Everything needed to run works everywhere"*

Apps Deployment

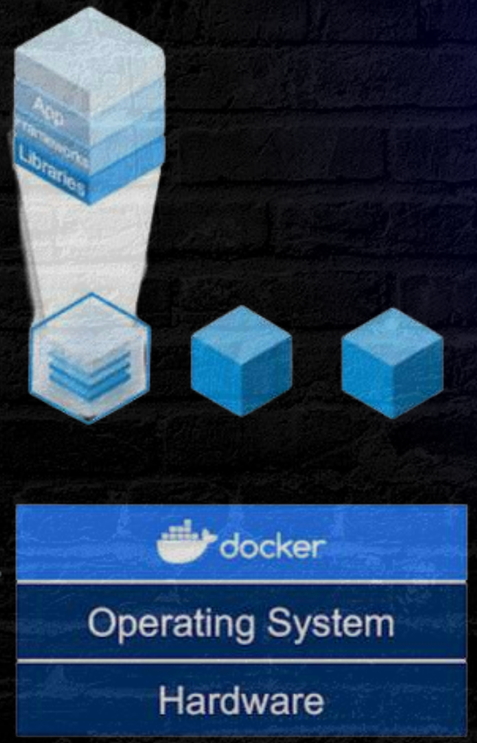
Single Server Era

Traditional Deployment

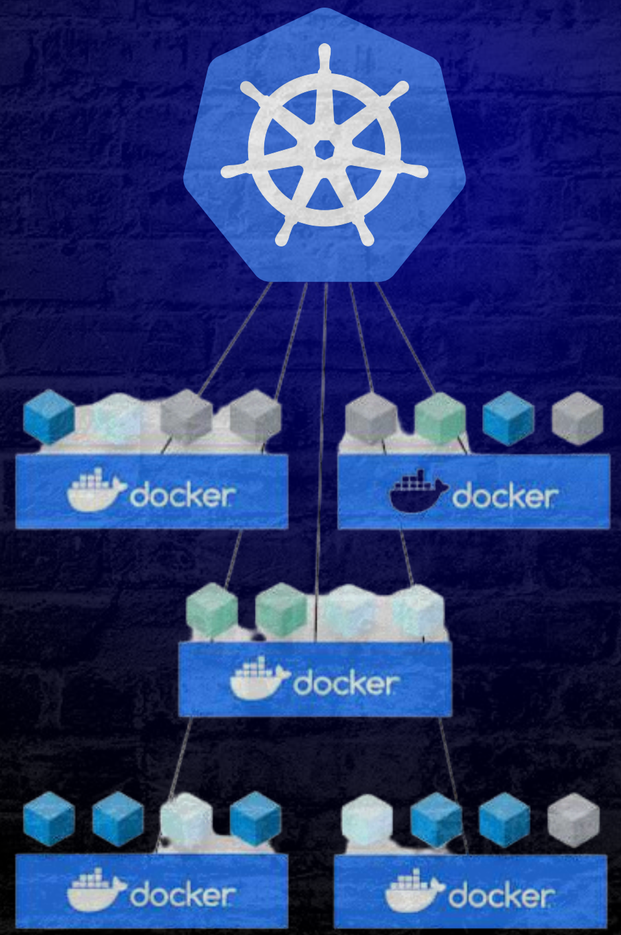
Virtual Machines



Virtualized Deployment

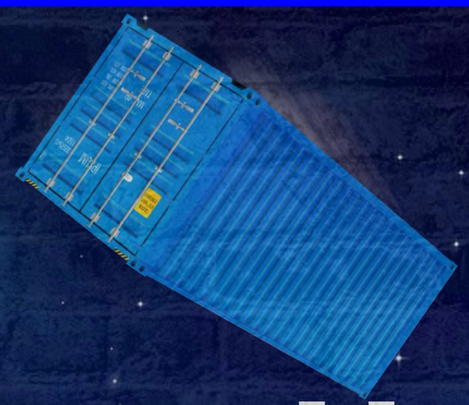


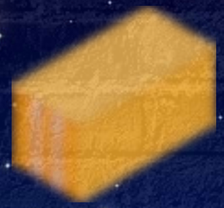
Container Deployment



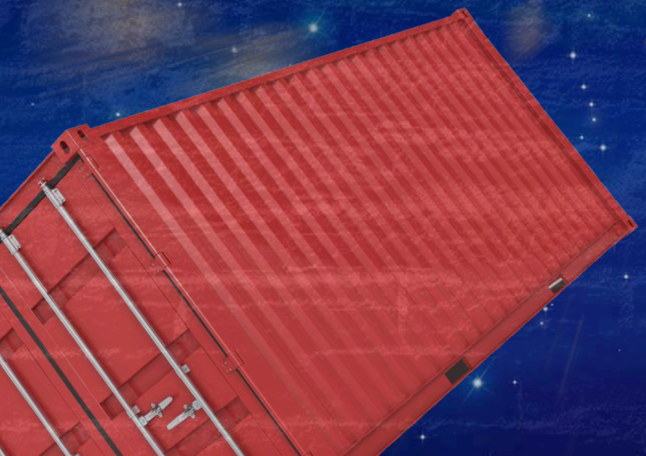
Kubernetes Deployment

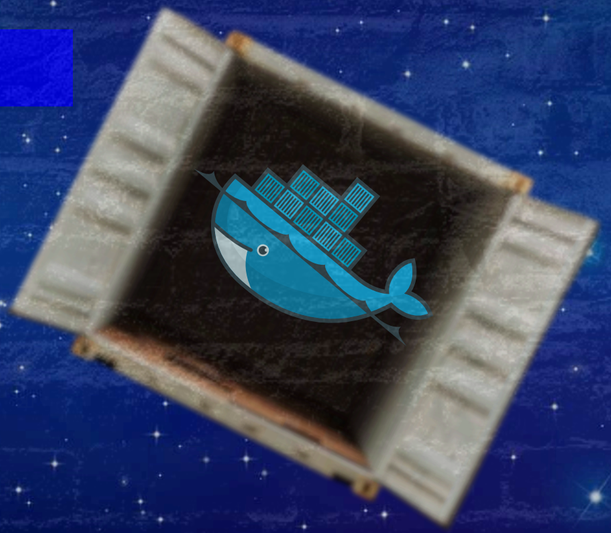
Club Informatique
& Télécom





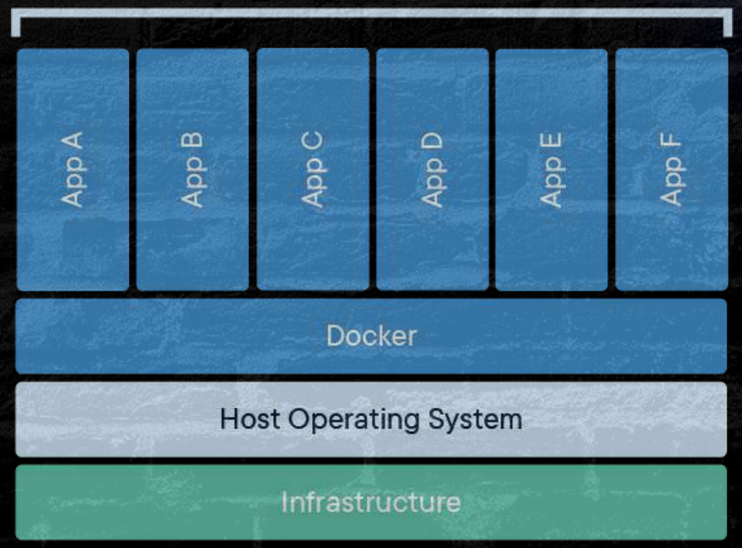
Understanding Containers & Docker





What is a Container?

A container is a lightweight, portable unit for running applications and their dependencies. Think of it as a "mini virtual machine" but without the heavy OS overhead.



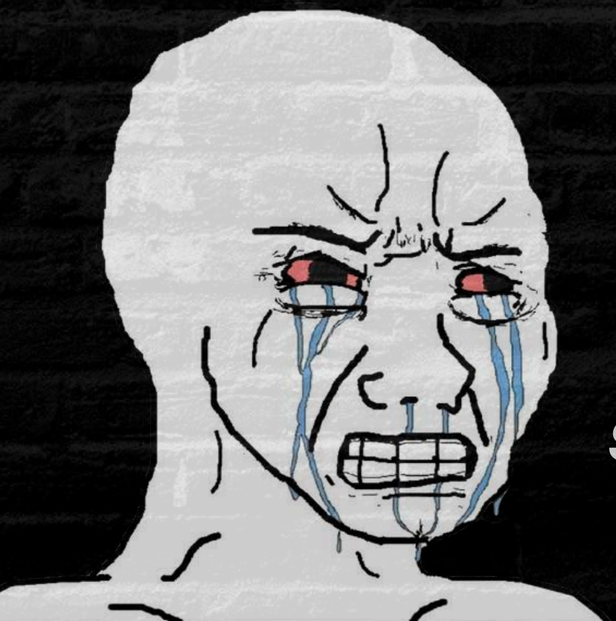
What is a Container?

Key Benefits:

- Runs consistently across different environments (no "works on my machine" issues).
- Fast startup and low resource consumption compared to Virtual Machines (VMs).
- Isolated environments for applications.

Configuration Challenges

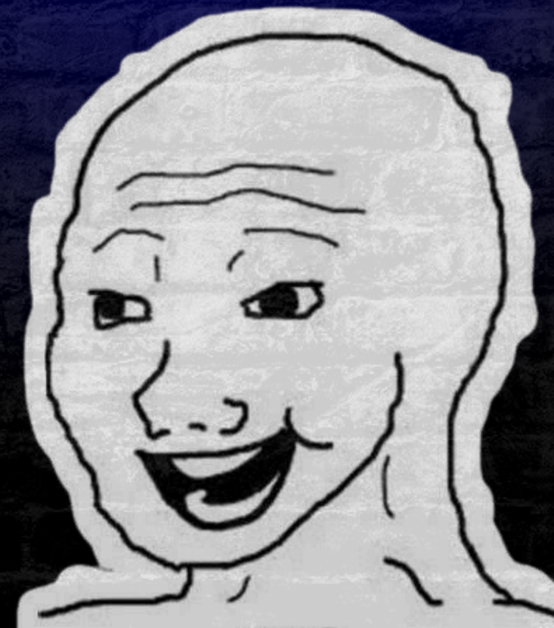
Tester



Sahbi hadchi makhdamch !!!

Developer

App khdama mzyan



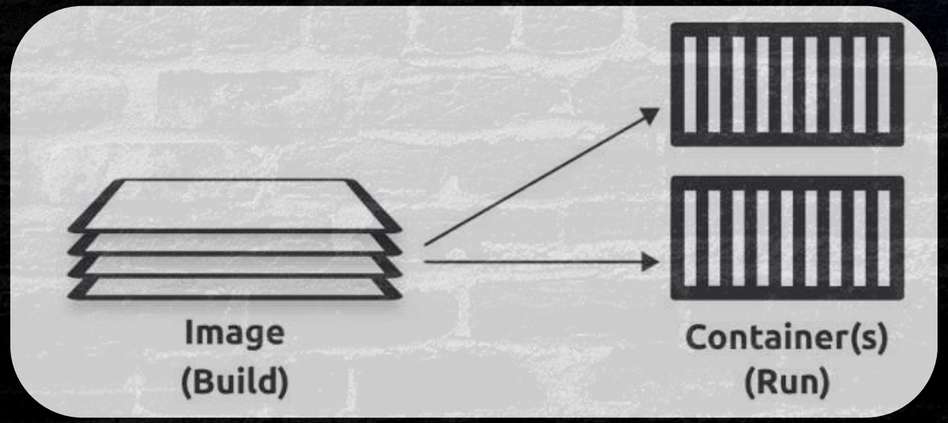
Docker vs. Containers

Feature	Containers (Concept)	Docker (Tool)
Isolation	Uses Linux namespaces & cgroups	Uses Docker Engine to manage them
Efficiency	Lightweight, no full OS	Simplifies container creation & management
Complexity	Requires manual setup	Provides user-friendly CLI commands

Docker simplifies container management just like Spotify makes music streaming easy, the tech existed before, but Docker made it accessible!

Images vs. Containers

An image is a read-only package containing everything you need to run an application. This means they include application code, dependencies, a minimal set of OS constructs, and metadata. You can start multiple containers from a single image.



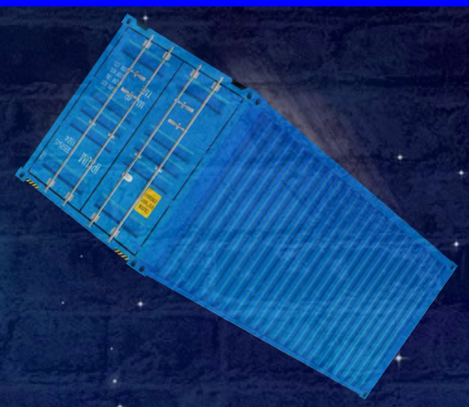
Images vs. Containers

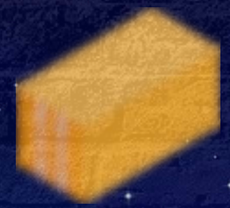
Docker Image: A blueprint (like an ISO file for VMs).
Docker Container: A running instance of that image.
💡 **Analogy:**

- **Image =** Recipe 📝 (defines what the app needs).
- **Container =** Prepared dish 🍽️ (running the app).
- You can make multiple dishes from the same recipe (multiple containers from one image).

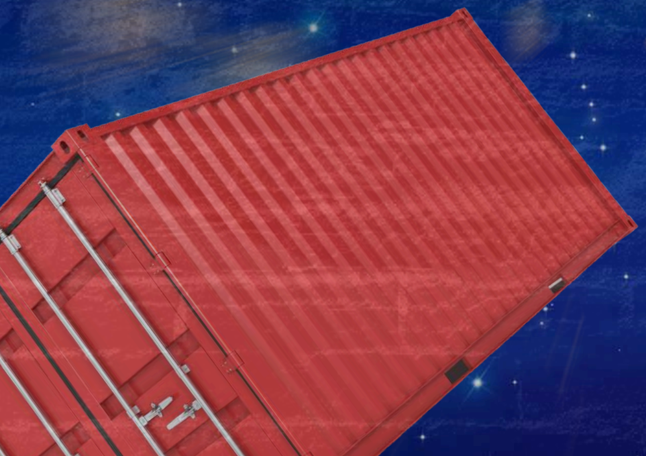
Live Demo 1: Getting started

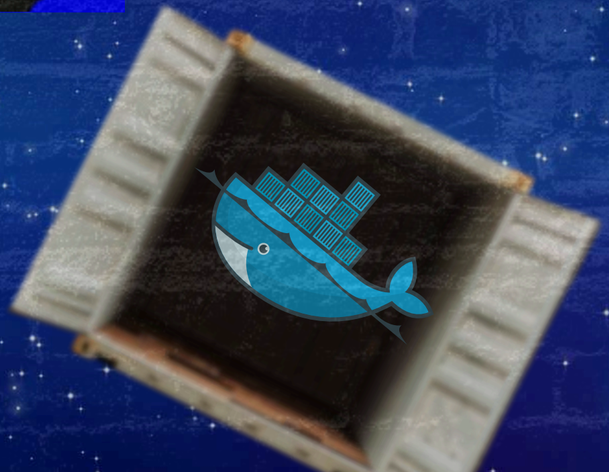
Club Informatique
& Télécom



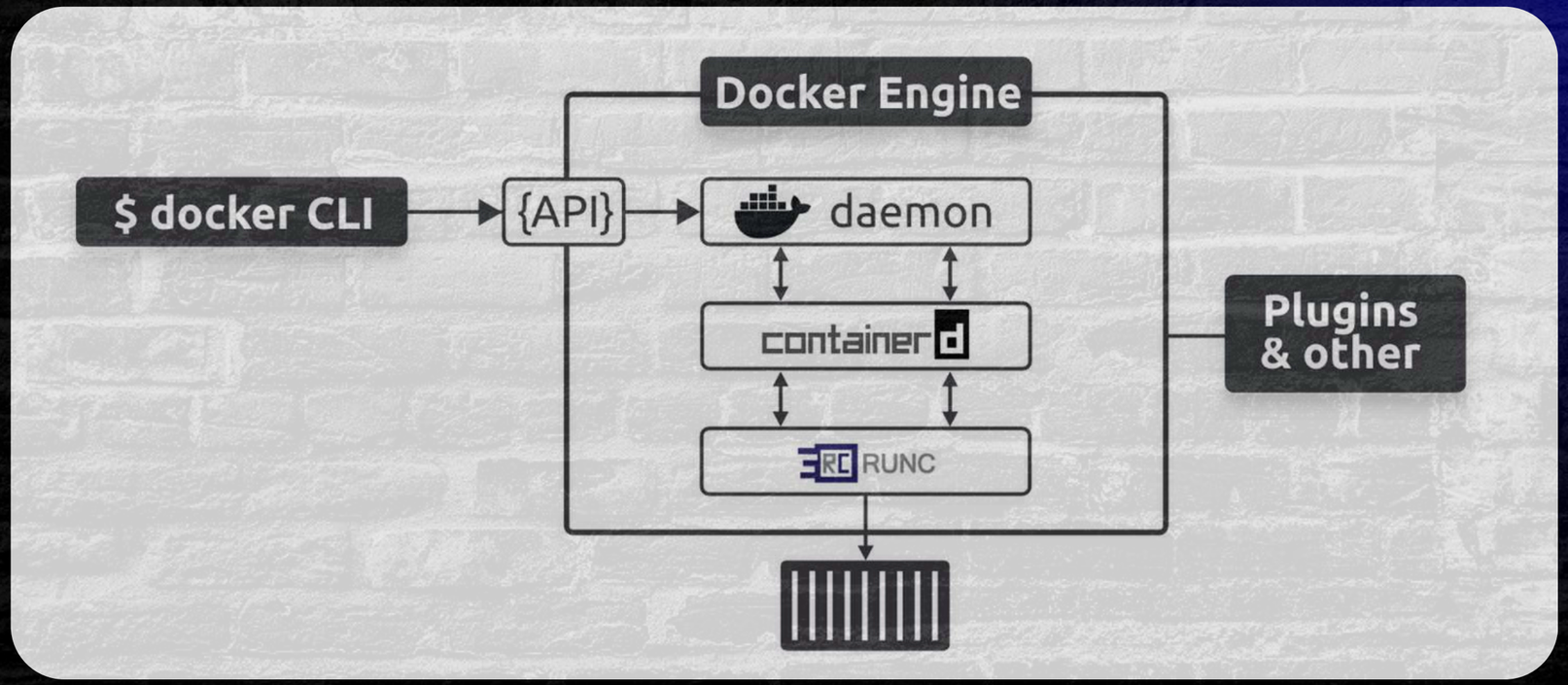


How Docker works?

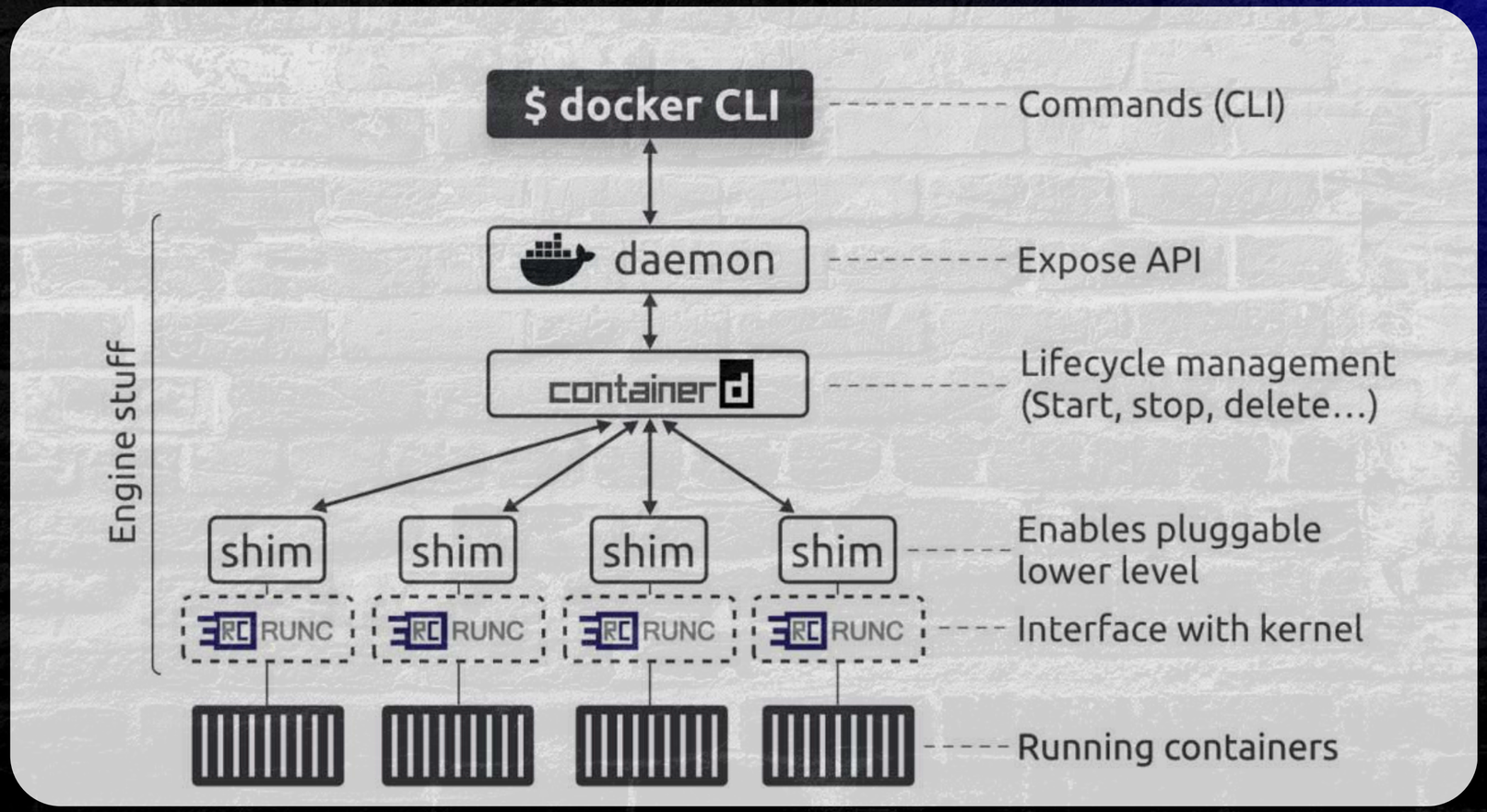




Docker Engine



Docker Engine

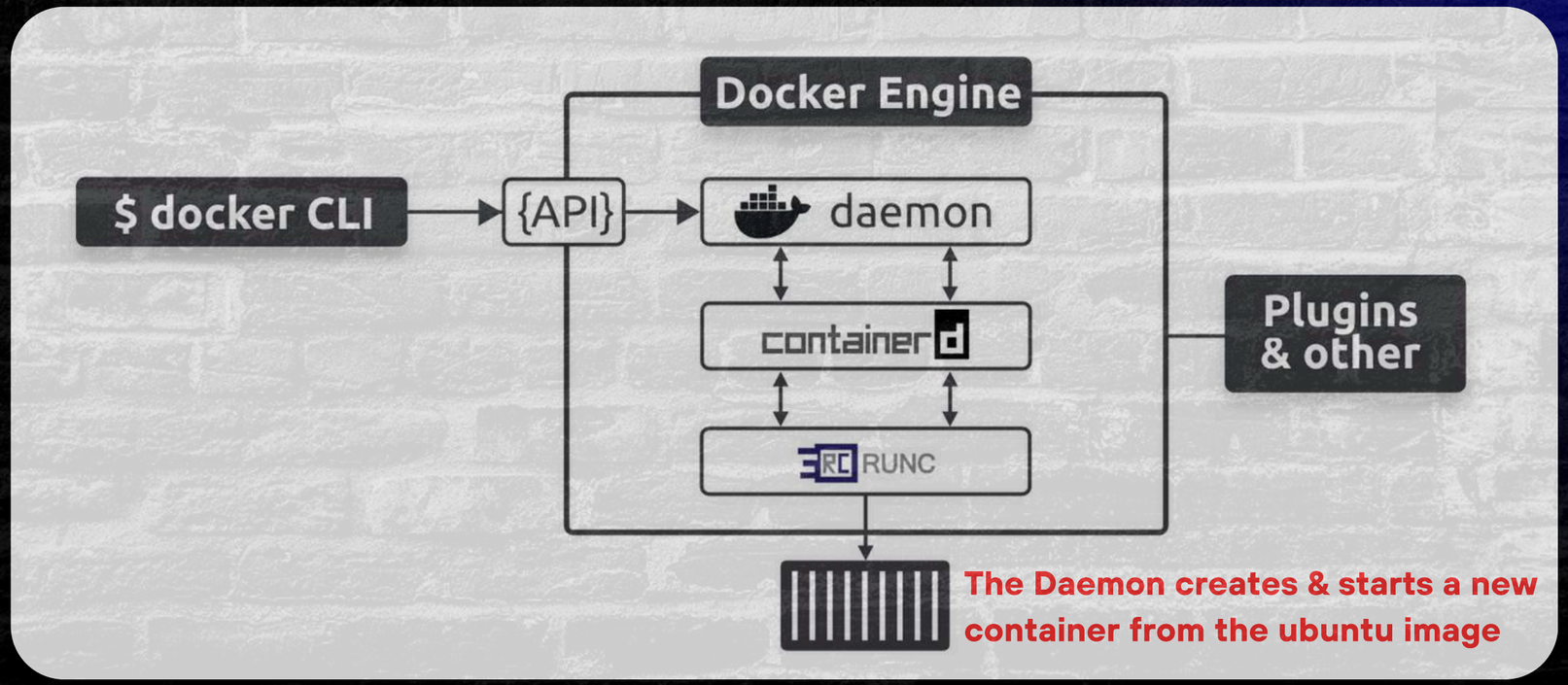


Docker CLI & Daemon

The Daemon checks if the ubuntu image exists locally

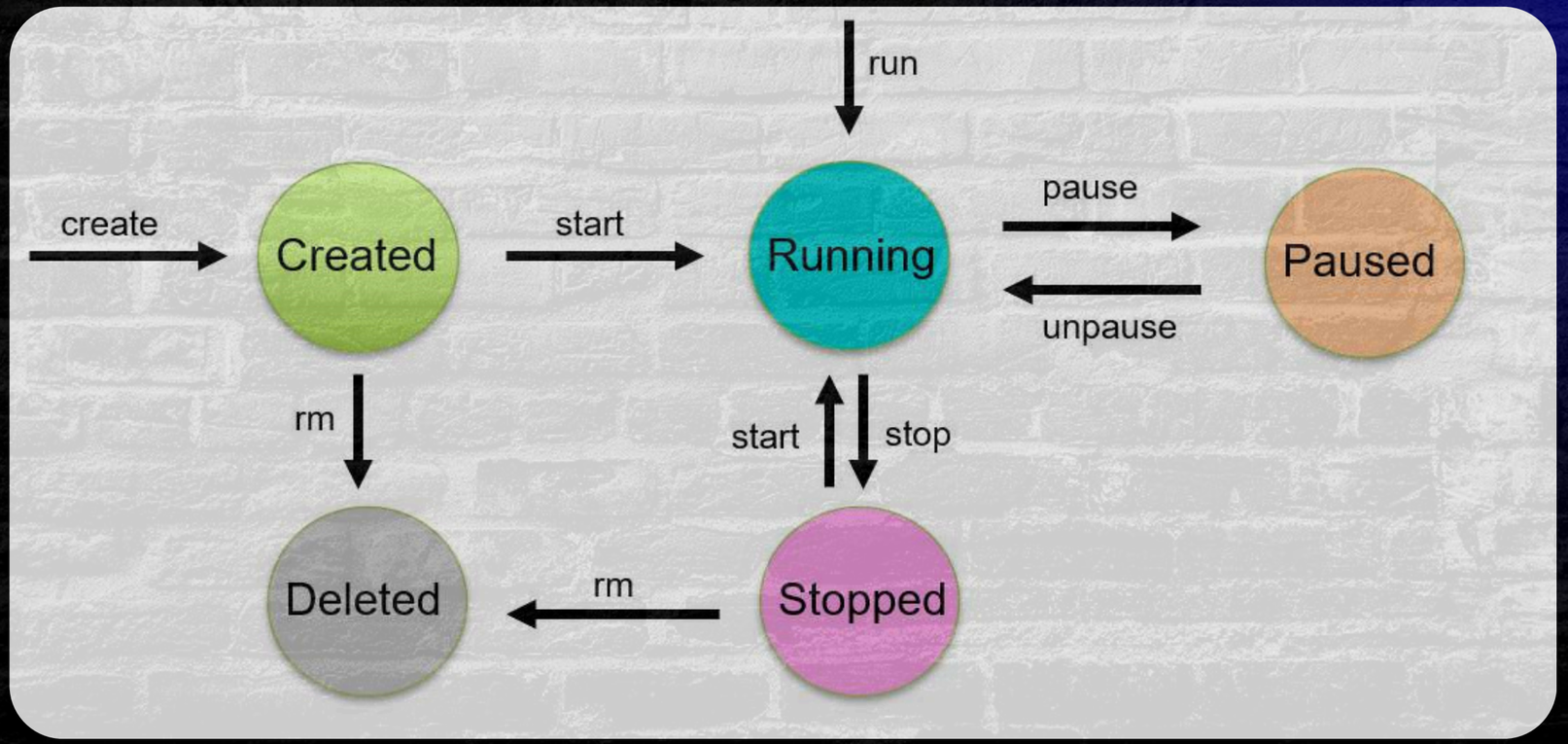
If not, it pulls it from Docker Hub

docker run ubuntu



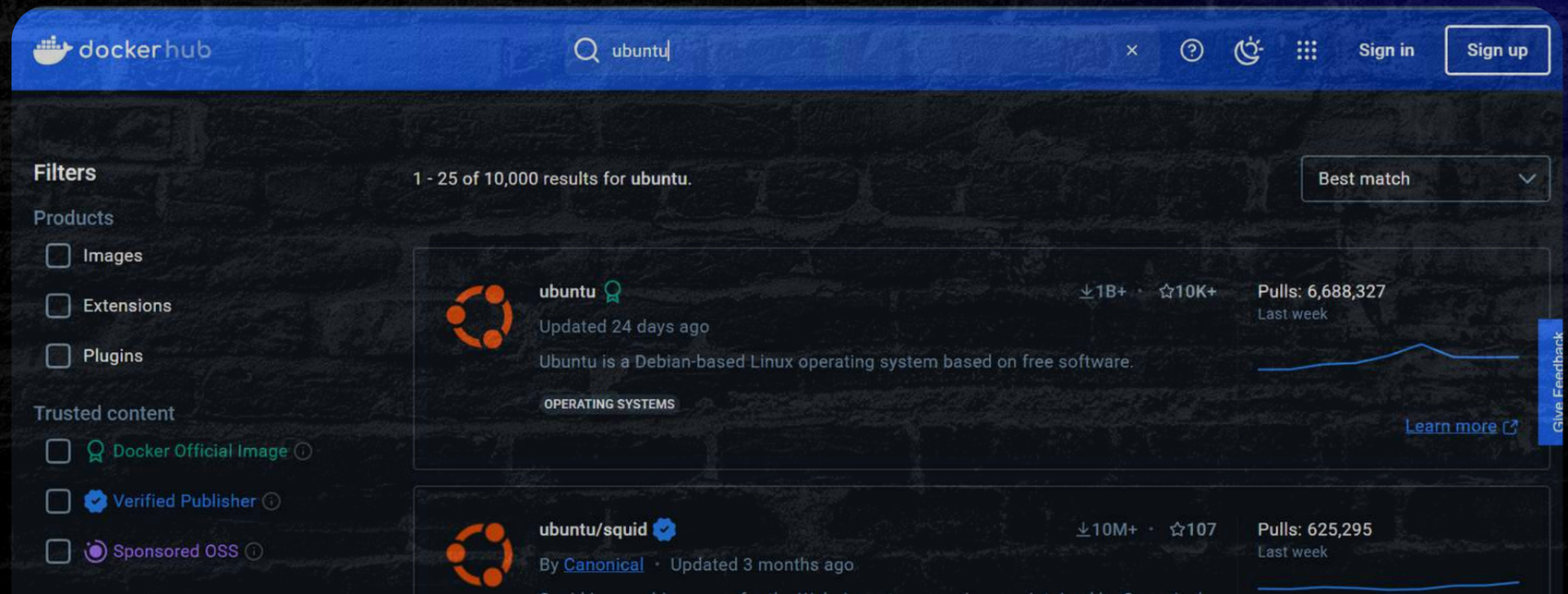
The Daemon creates & starts a new container from the ubuntu image

Container Lifecycle



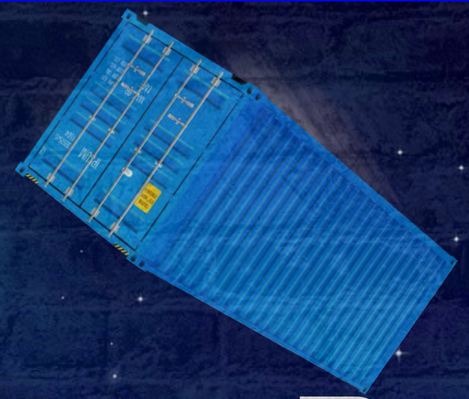
Docker Hub

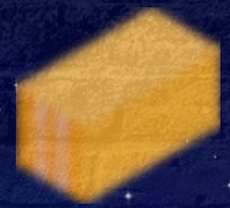
Docker Hub is a cloud-based repository for sharing container images.



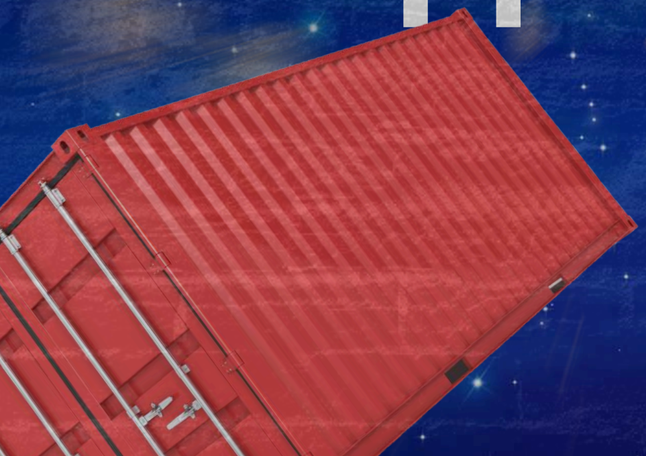
Live Demo 2: Managing Containers

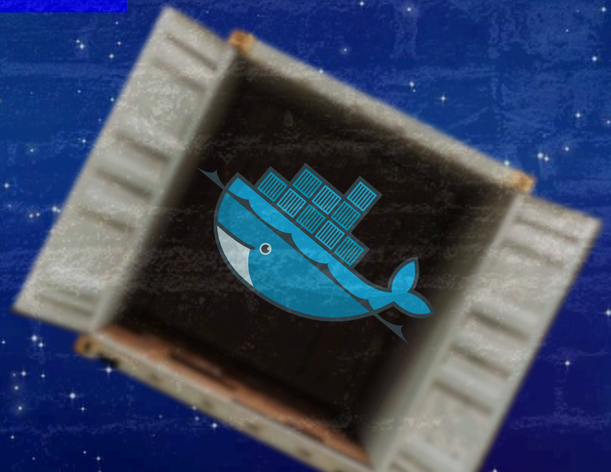
Club Informatique
& Télécom

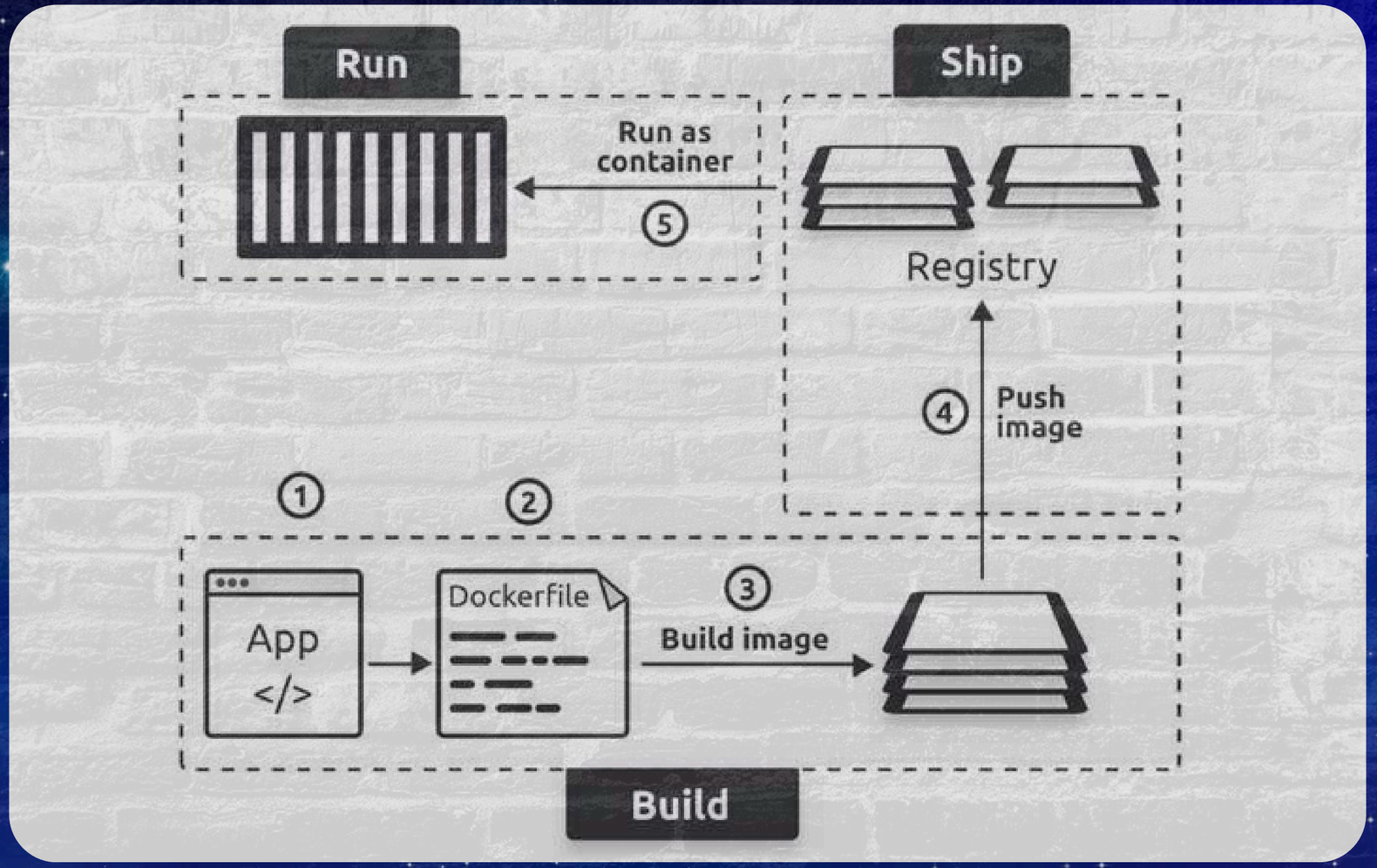




Building and Running Applications






Run
Ship
Run as
container
5
Registry
4
Push
image
1
2
App
</>
Dockerfile
3
Build image
Build

Creating a Docker Image

We use a Dockerfile to define how to build an image, it tells Docker how to prepare the application.

```
1. ARG NODE_VERSION=20.8.0
2. FROM node:${NODE_VERSION}-alpine
3. ENV NODE_ENV production
4. WORKDIR /usr/src/app
5. RUN --mount=type=bind,source=package.json,target=package.json \
    --mount=type=bind,source=package-lock.json,target=package-lock.json \
    --mount=type=cache,target=/root/.npm \
    npm ci --omit=dev
6. USER node
7. COPY . .
8. EXPOSE 8080
9. CMD node app.js
```

Building the Docker Image

🔍 What happens?

- Docker reads the Dockerfile, follows the steps, and creates an image.
- This image is now stored locally and can be used to create containers.

Live Demo 3: Containerize an app

Club Informatique
& Télécom

CRAFTED carefully

By IKBI Abdelilah