

WOMEN WHO  
**CODE**®  
/medellín

# Containerization.

A guide to understanding and working with docker

# INTRO

APRIL 27, 2024



# Sebastian Pino

I am a proud Colombian software engineer and data scientist. I have worked in the last-mile shipping sector, where I learned different software development skills and started to love the software world. Furthermore, I have worked in the health sector in Colombia. Additionally, I worked on a clinical studies application ecosystem for a USA client. A little over a year ago, I decided to start my leadership journey working as a Scrum Master and Technical lead for a healthcare company named ABA Tech located in the USA.

I like sharing my knowledge, so during my university, I volunteered, teaching children and young people from different communities in my city. This work helped me understand the society we live in; it motivated me to keep working hard and focused on contributing to change

**01**

**What is docker?**

# GOALS

---

**02**

**What problems does docker solve?**

**03**

**A bit of history**

**04**

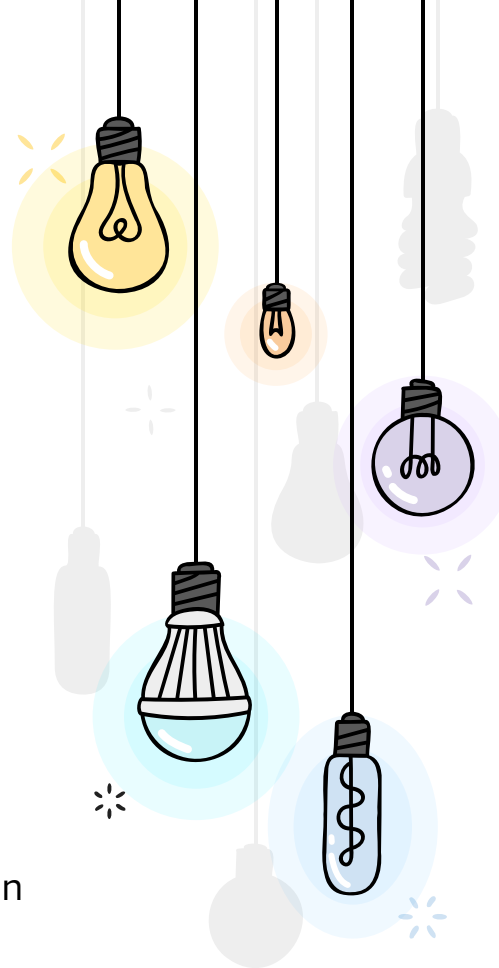
**Virtualization vs containerization**

# 1. What is docker?

What do you understand as Docker?

# What is docker?

- 1.** Docker is a platform for developing, shipping, and running applications
- 2.** It uses containerization to package software and its dependencies.
- 3.** Containers are isolated environments that ensure consistency across different systems.
- 4.** Docker enables easy deployment of applications in any environment, from development to production.



## 2. What problems does docker solve?

In my local environments works!



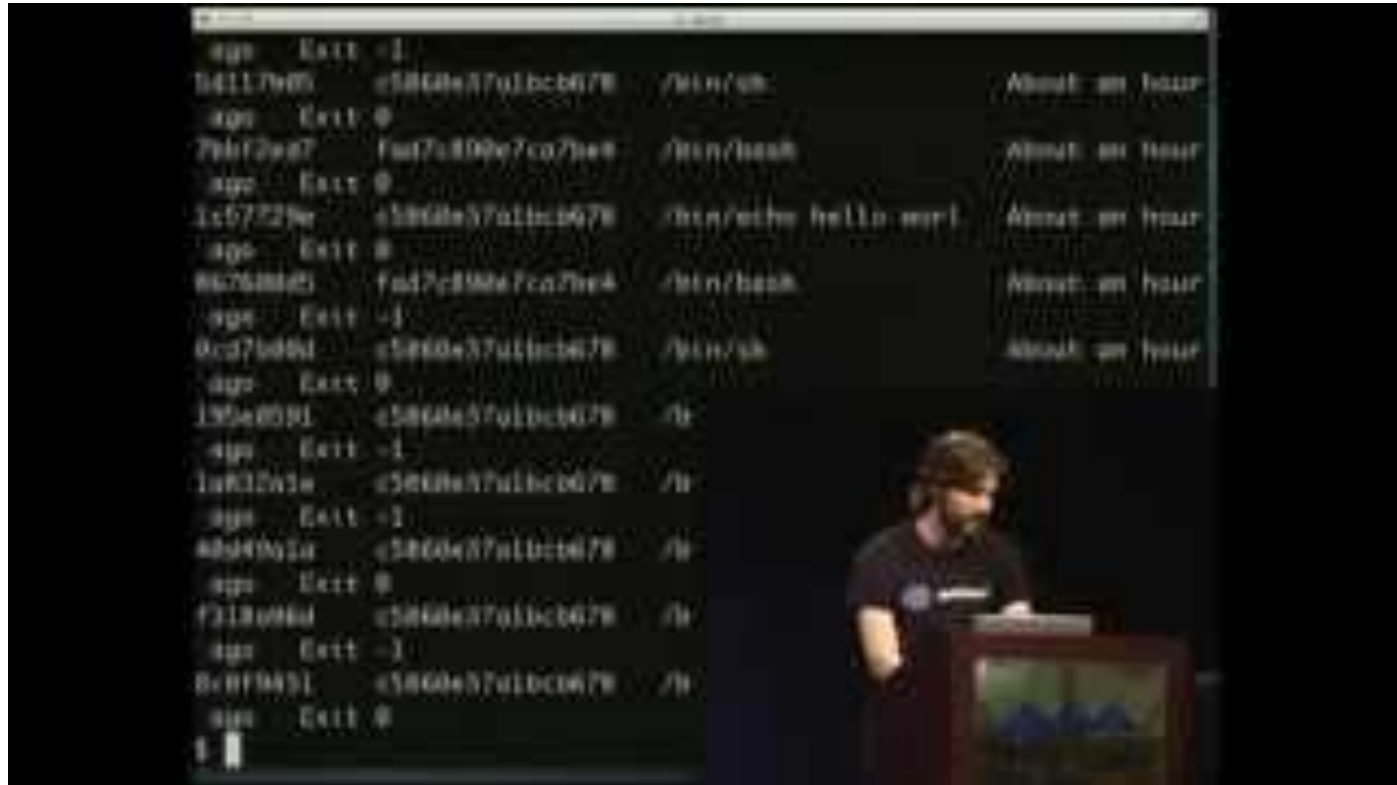
# What problems does docker solve?

1. **Dependency Management:** Docker resolves issues with conflicting software dependencies by encapsulating applications and their dependencies within containers.
2. **Consistency:** It ensures consistency across different environments, eliminating the "it works on my machine" problem.
3. **Portability:** Docker allows applications to run seamlessly on any infrastructure, from developer laptops to production servers.
4. **Isolation:** Containers provide isolation for applications, preventing conflicts and ensuring security.
5. **Resource Efficiency:** Docker optimizes resource utilization by sharing the host OS kernel among containers.
6. **Scalability:** It facilitates scaling applications horizontally by quickly spinning up multiple container instances.
7. **Continuous Integration/Continuous Deployment (CI/CD):** Docker streamlines the CI/CD pipeline, enabling rapid development cycles and deployment.

# 3. A bit of history

Where it all began

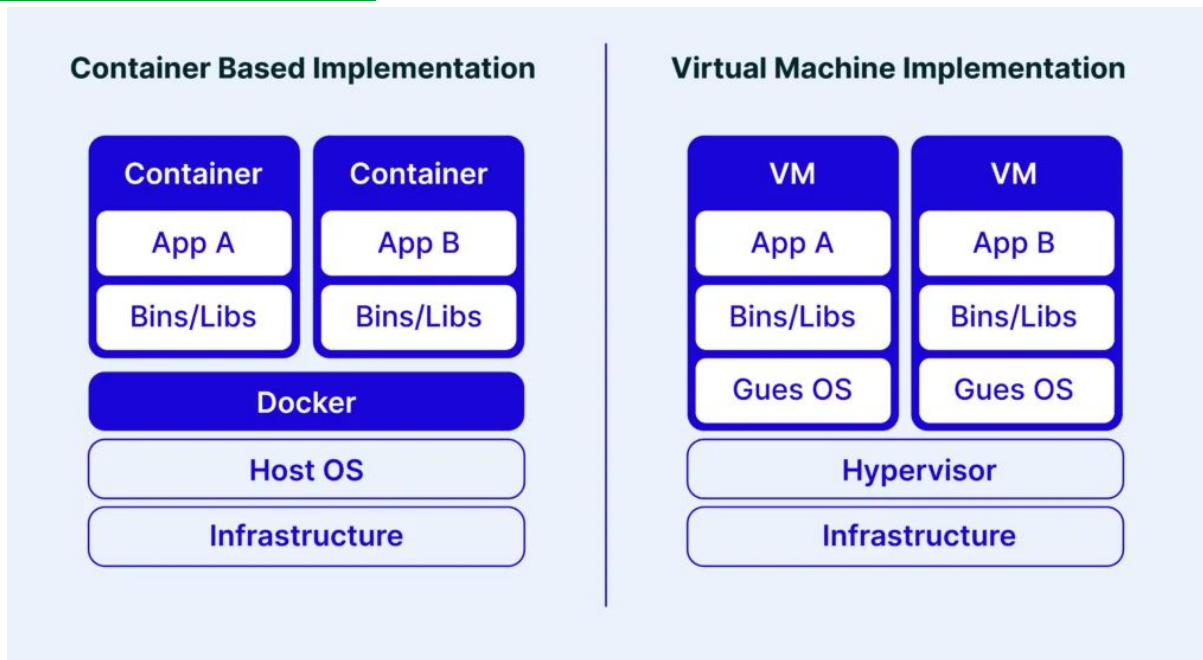
# A bit of history



## 4. Virtualization vs containerization

Is there any difference?

# Virtualization vs containerization



<https://www.aquasec.com/cloud-native-academy/docker-container/containerization-vs-virtualization/>