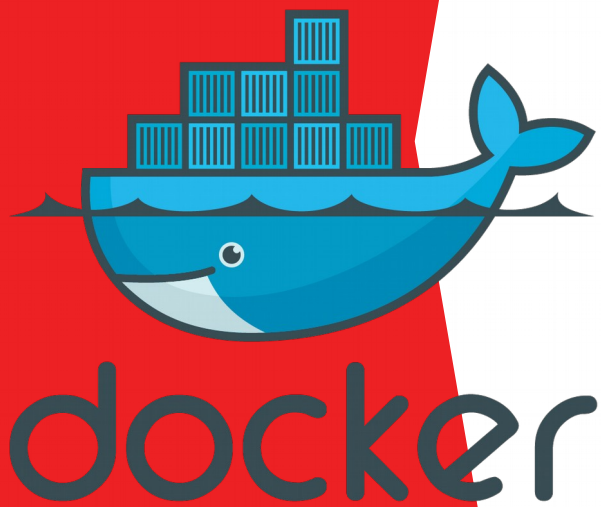
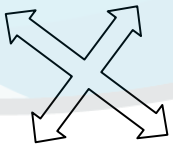
Docker Introduction

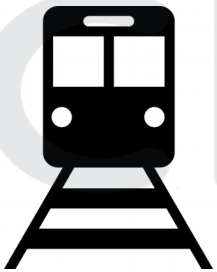


Yusuf Hadiwinata Sutandar

The Problem

Cargo Transport 1960s





The Solution?

Solution?

Intermodal Shipping Container

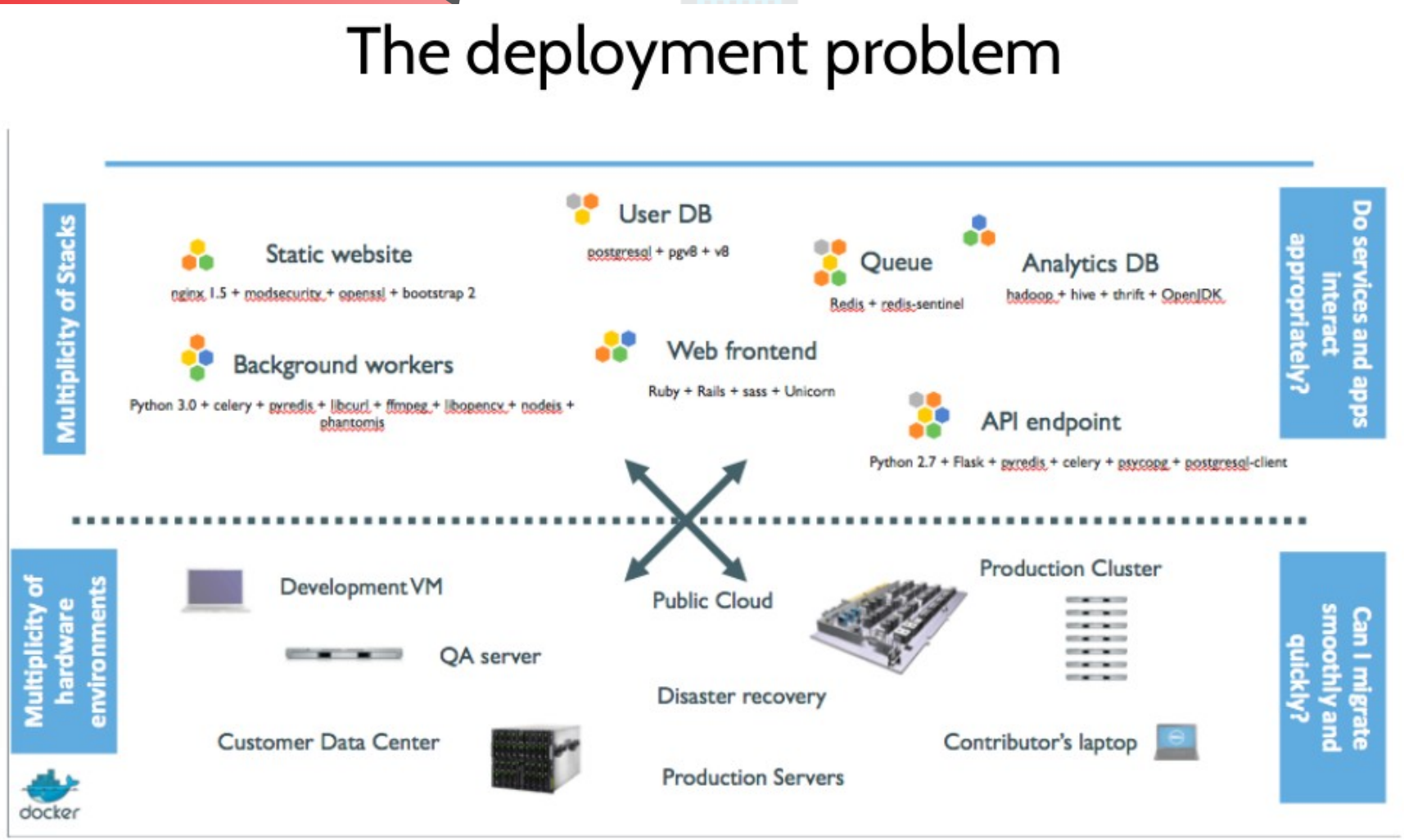
The Solution?

90% of all cargo now shipped in a standard container

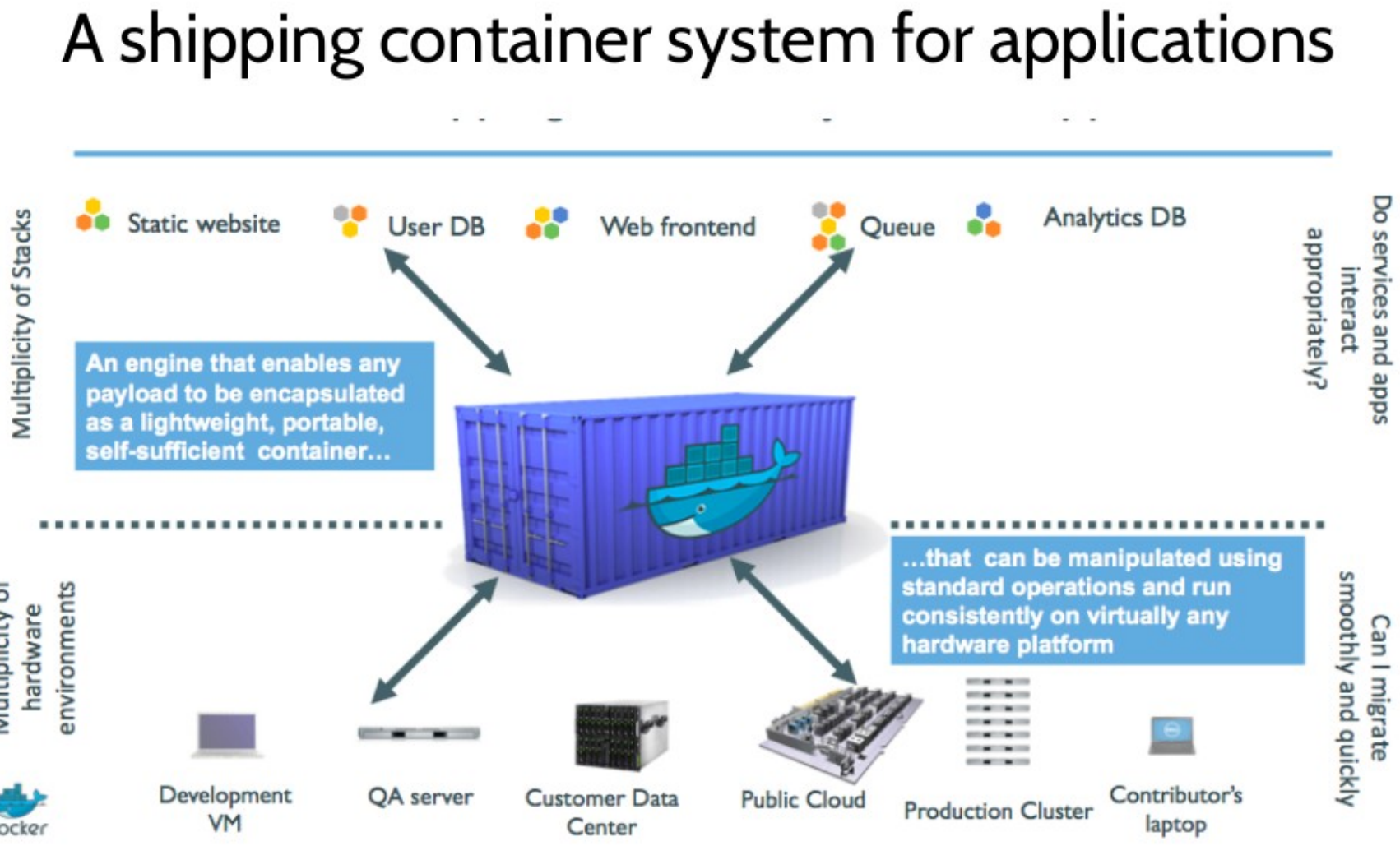
Order of magnitude reduction in cost and time to load and unload ships, trains, trucks

The Solution?

The App Problem?



The Solution?



The Docker Solution

So now what's Docker?

About Docker Inc.

- Founded in 2009.
- Formerly dotCloud Inc.
- Primary sponsor of the Docker Project.
- Hires maintainers and contributors.
- Provides infrastructure for the project.
- Runs the Docker Hub.
- HQ in San Francisco.
- Backed by more than 100M in venture capital.

The Docker Solution

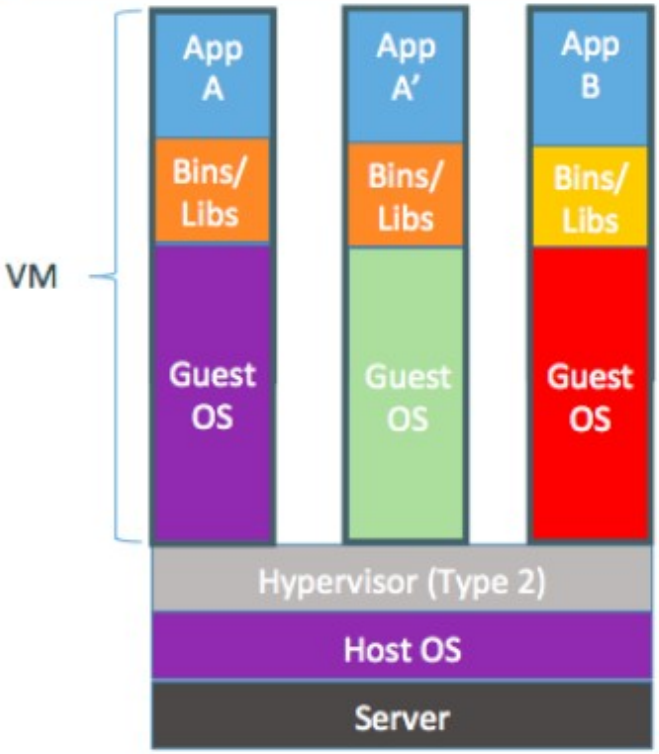
So now what's Docker?

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.
Consisting of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows, Docker enables apps to be quickly assembled from components and eliminates the friction between development, QA, and production environments. As a result, IT can ship faster and run the same app, unchanged, on laptops, data center VMs, and any cloud."
source: https://www.docker.com/whatisdocker/

The Docker Solution

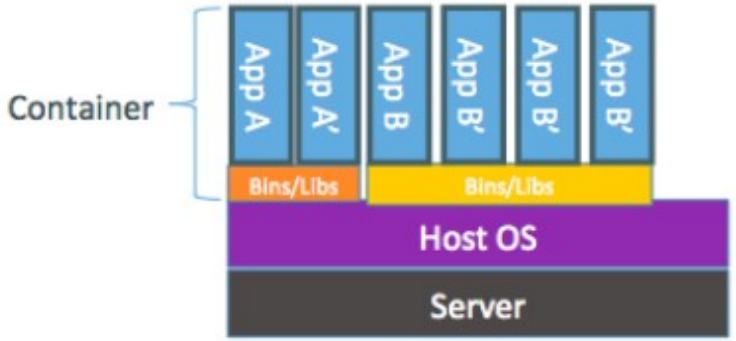
containers as lightweight VMs

Less overhead!



Containers are isolated, but share OS kernel and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



The Docker Solution

Virtual Machine and Container Complement each other

Virtual Machine

- Virtual machines include the application, the necessary binaries and libraries, and an entire guest operating system
- Each Guest OS has its own Kernel and user space

Containers

- Containers run as isolated processes in user space of host OS
- They share the kernel with other container (container-processes)
- Containers include the application and all of its dependencies
- Not tied to specific infrastructure

The Docker Solution

Containers before Docker

- No standardized exchange format.

(No, a rootfs tarball is not a format!)

- Containers are hard to use for developers.

(Where's the equivalent of docker run debian?)

- No re-usable components, APIs, tools.

(At best: VM abstractions, e.g. libvirt.)

Analogy:

- Shipping containers are not just steel boxes.
- They are steel boxes that are a standard size, with the same hooks and holes

The Docker Solution

Containers after Docker

- Standardize the container format, because containers were not portable.
- Make containers easy to use for developers.
- Emphasis on re-usable components, APIs, ecosystem of standard tools.
- Improvement over ad-hoc, in-house, specific tools.

The Docker Solution

Docker Benefit

Make the entire lifecycle more efficient, consistent, and repeatable

Increase the quality of code produced by developers

Eliminate inconsistencies between development, test, production, and customer environments

Support segregation of duties

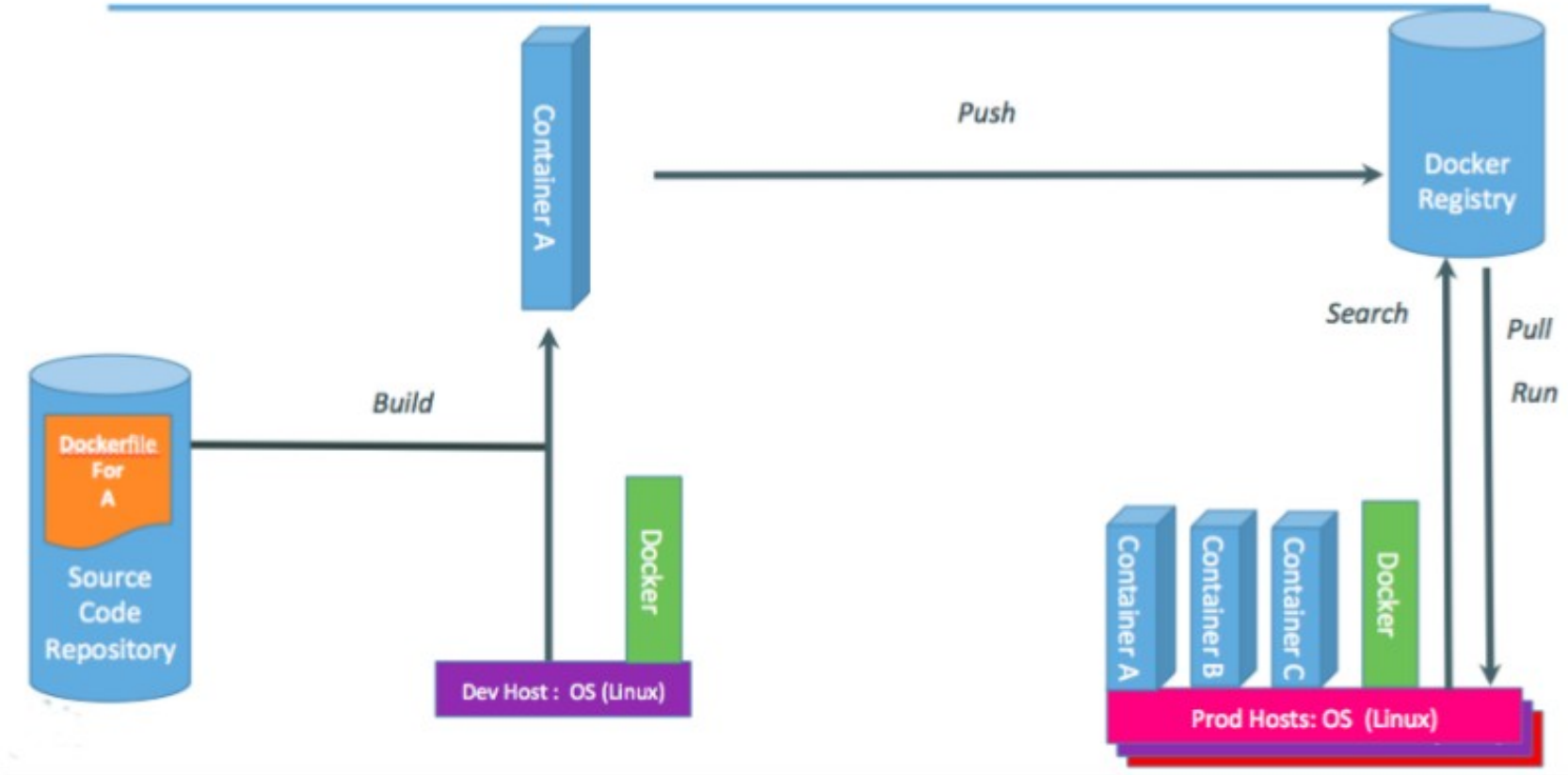
Significantly improves the speed and reliability of continuous deployment and continuous integration systems

Because the containers are so lightweight, address significant performance, costs, deployment, and portability issues normally associated with VMs

The Docker Solution

Container image as build artifact

The same container can go from dev, to test, to QA, to prod.



The Docker Solution

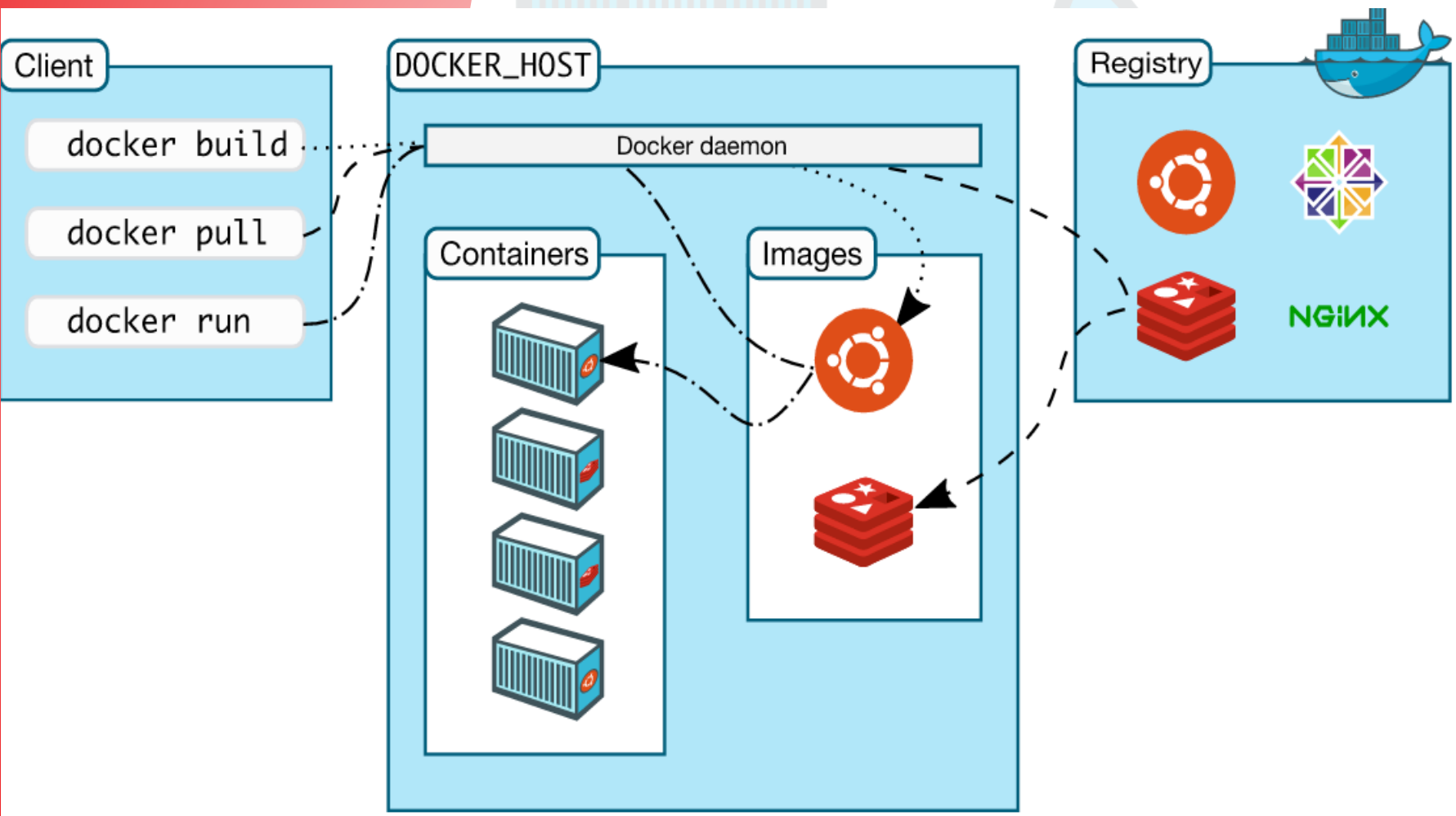
Developer Say:

Build Once, Run Anywhere

Operator: Configure Once, Run Anything

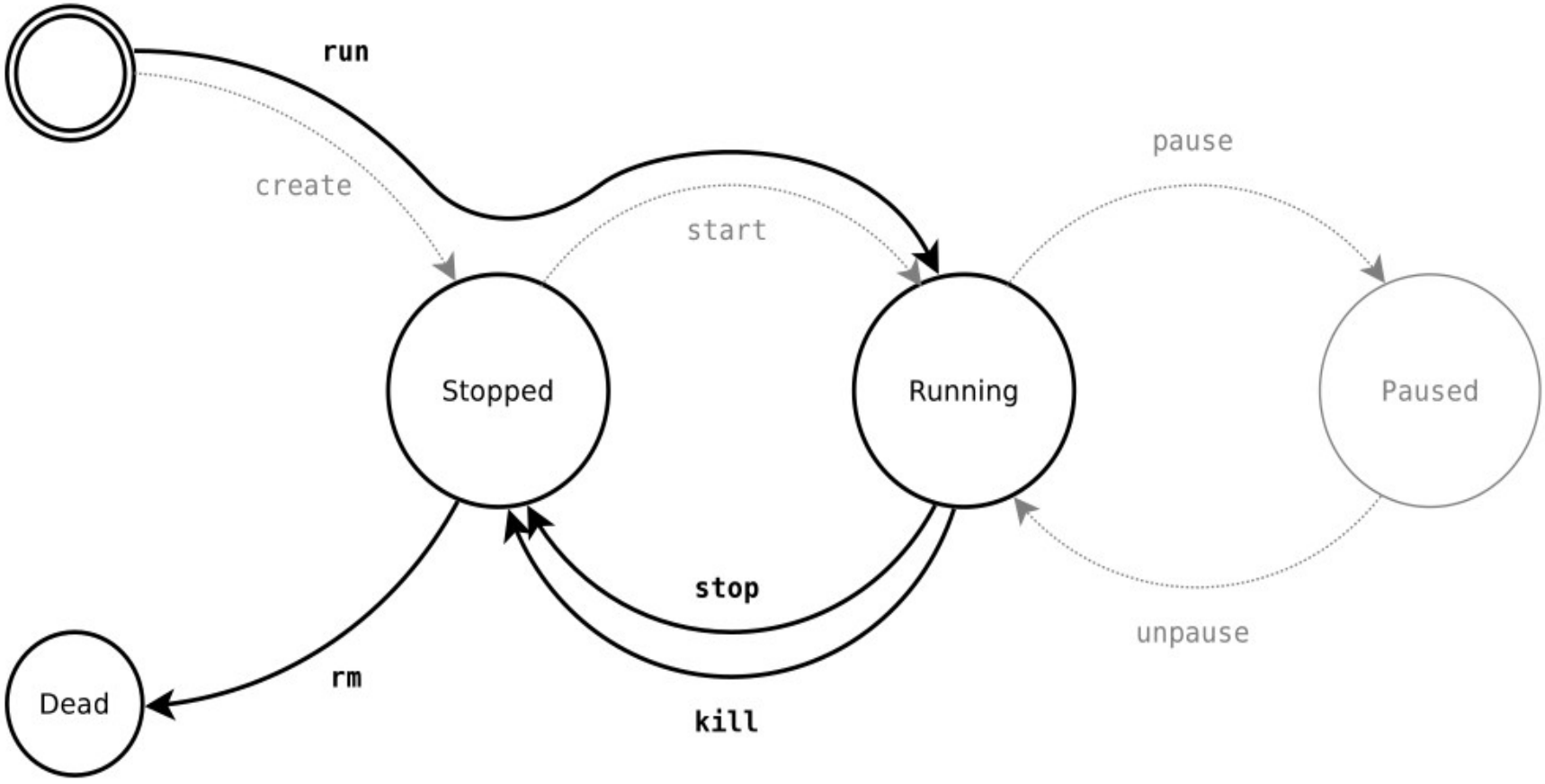
The Docker Solution

Docker Architecture

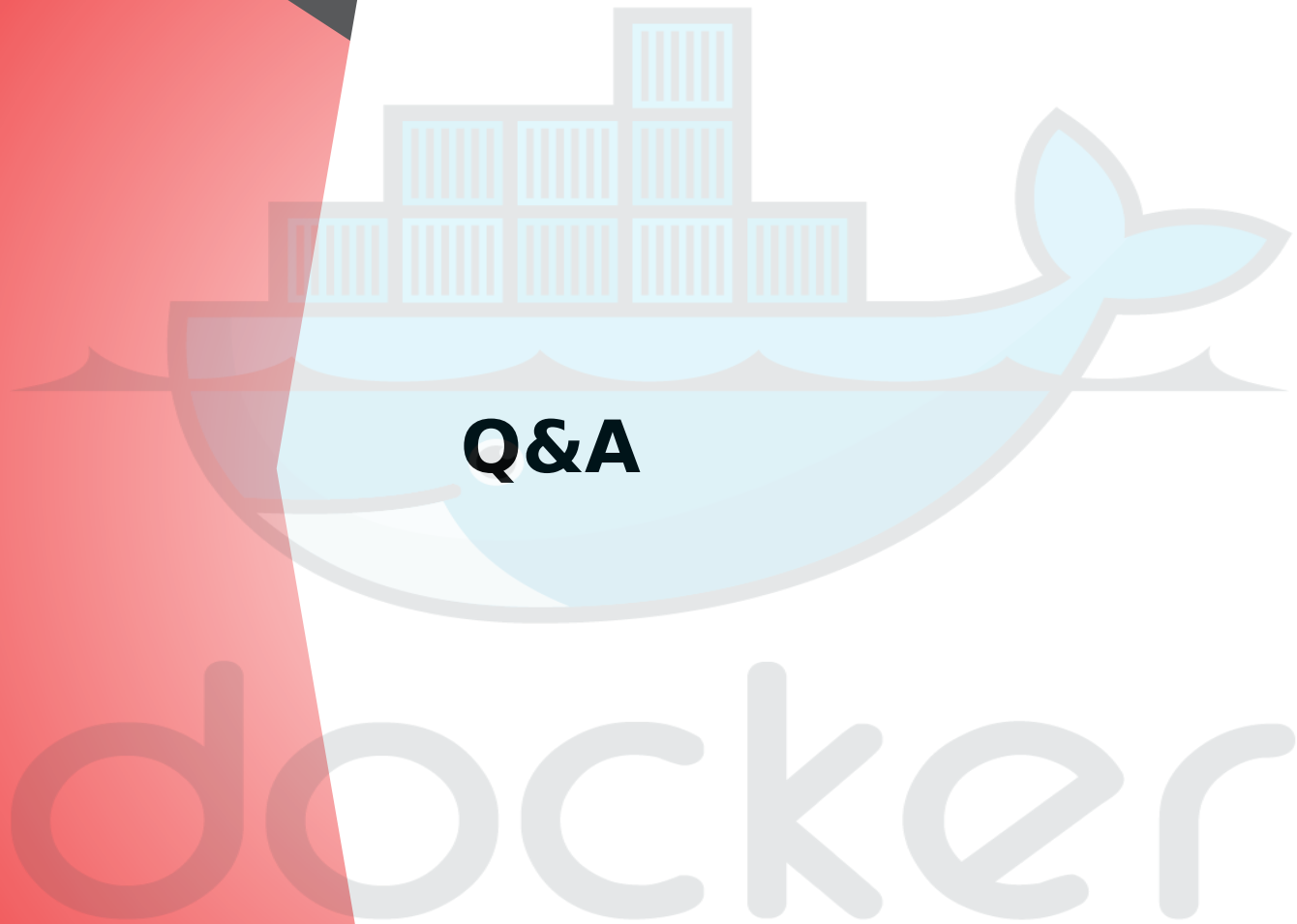


The Docker Solution

Lifecycle of a docker container



The Docker Question



Q&A

INOVASI
INFORMATIKA
INDONESIA

Graha BIP 6th Floor
Jl. Jend. Gatot Subroto Kav. 23
Jakarta 12930, Indonesia
Phone : (62) 21 290 23393
Fax : (62) 21 525 8065
info@i-3.co.id
www.i-3.co.id

Inovasi Informatika Indonesia
I3_ID