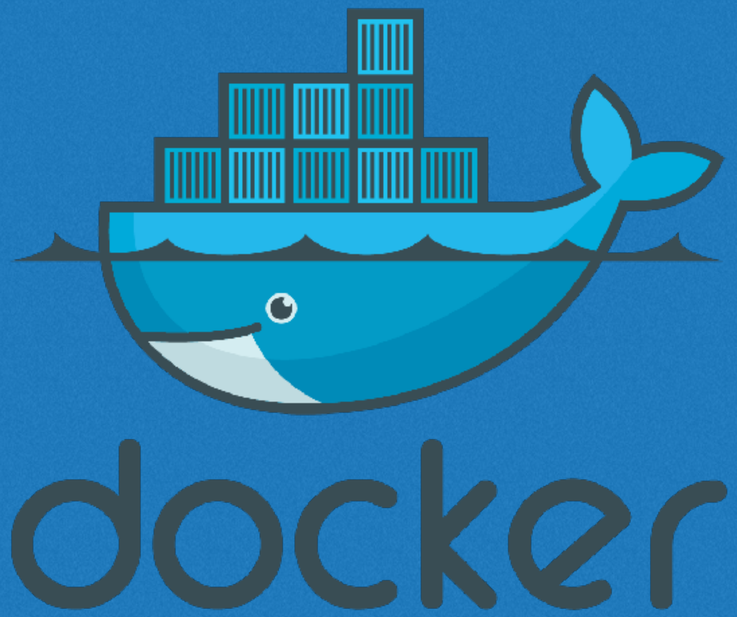
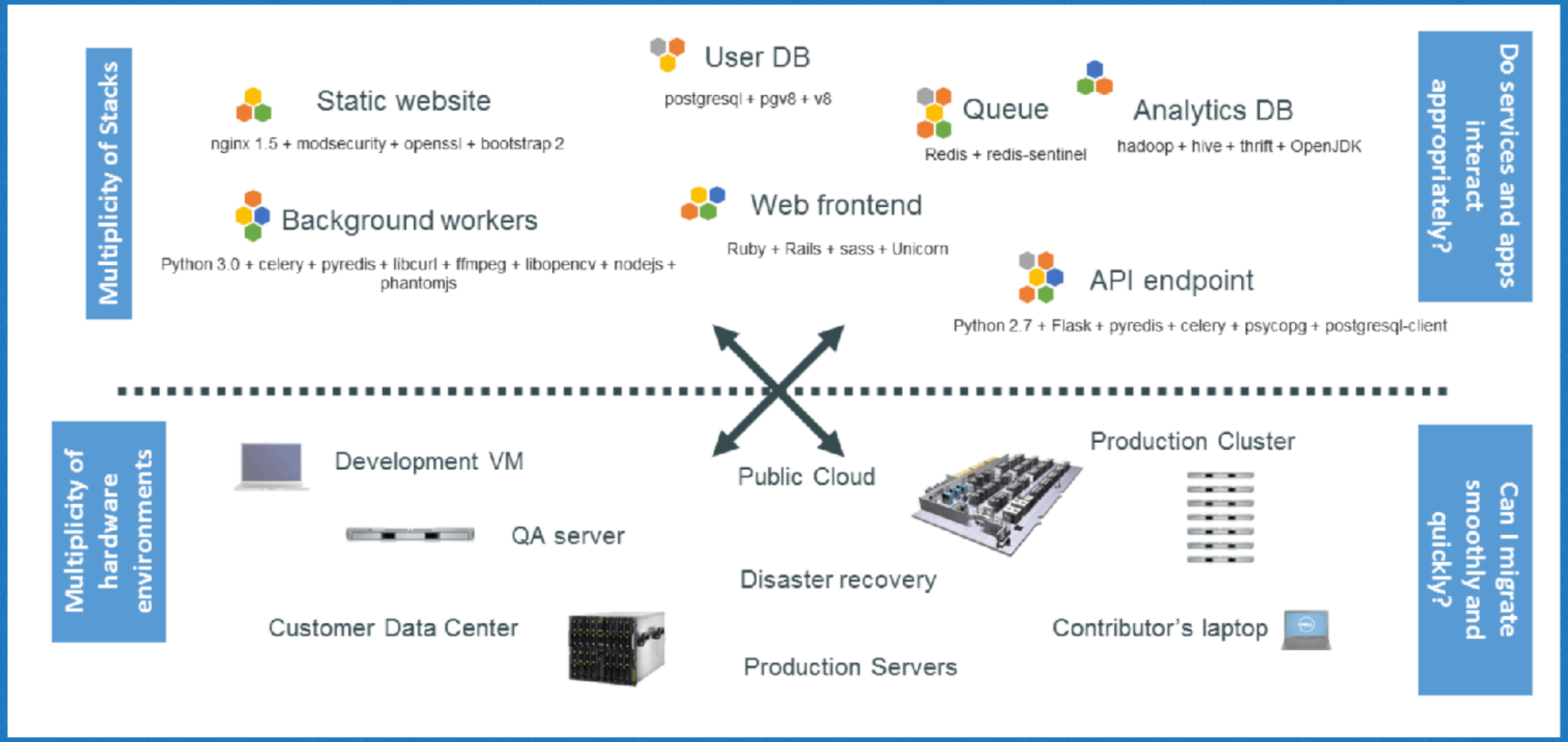
Welcome to Docker

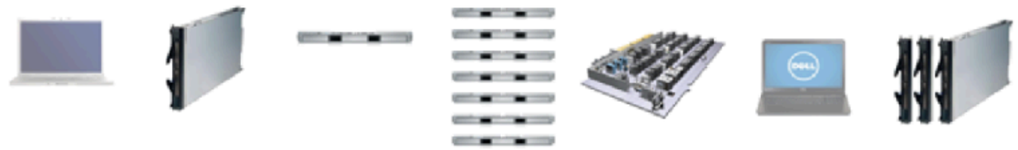


The Challenge

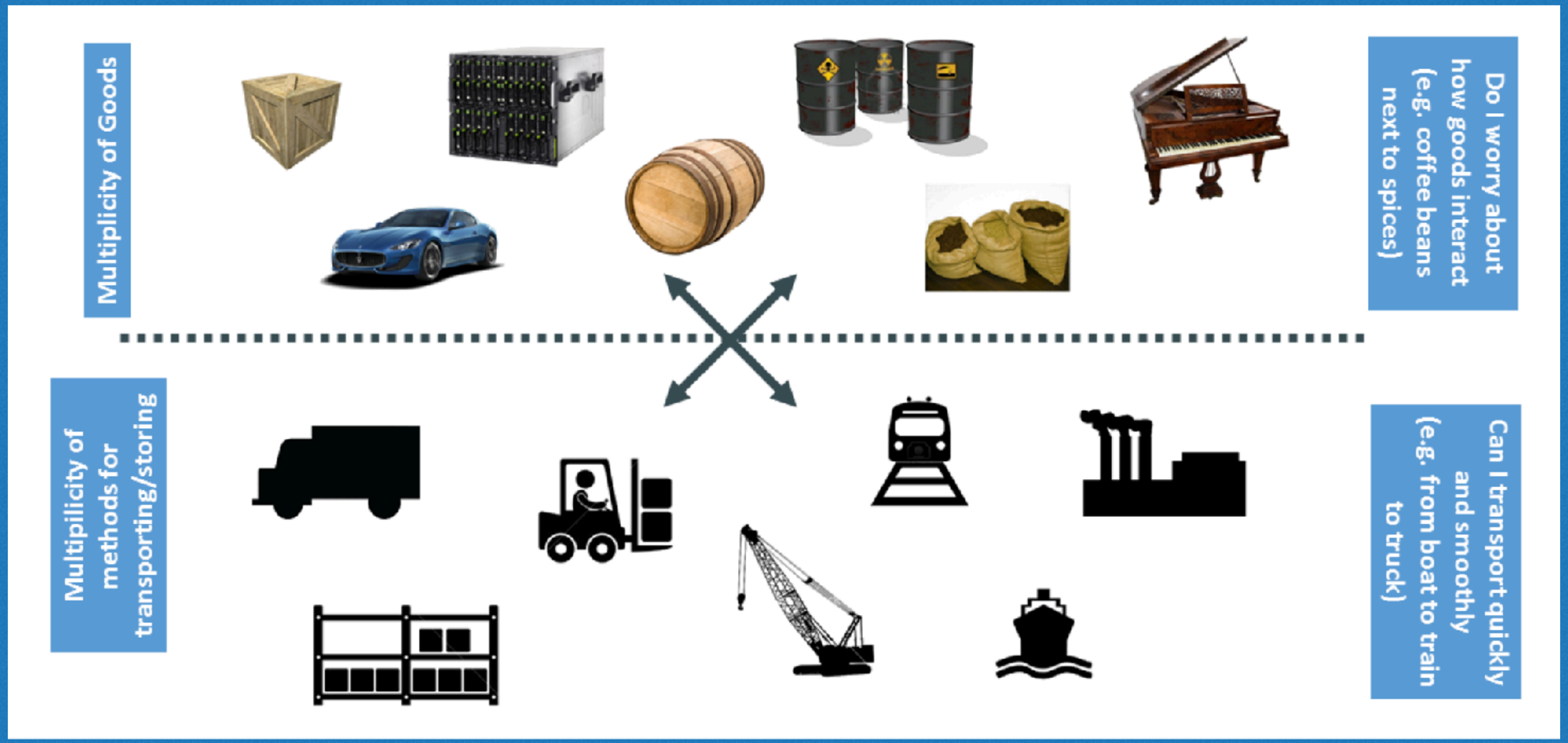


The Challenge

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

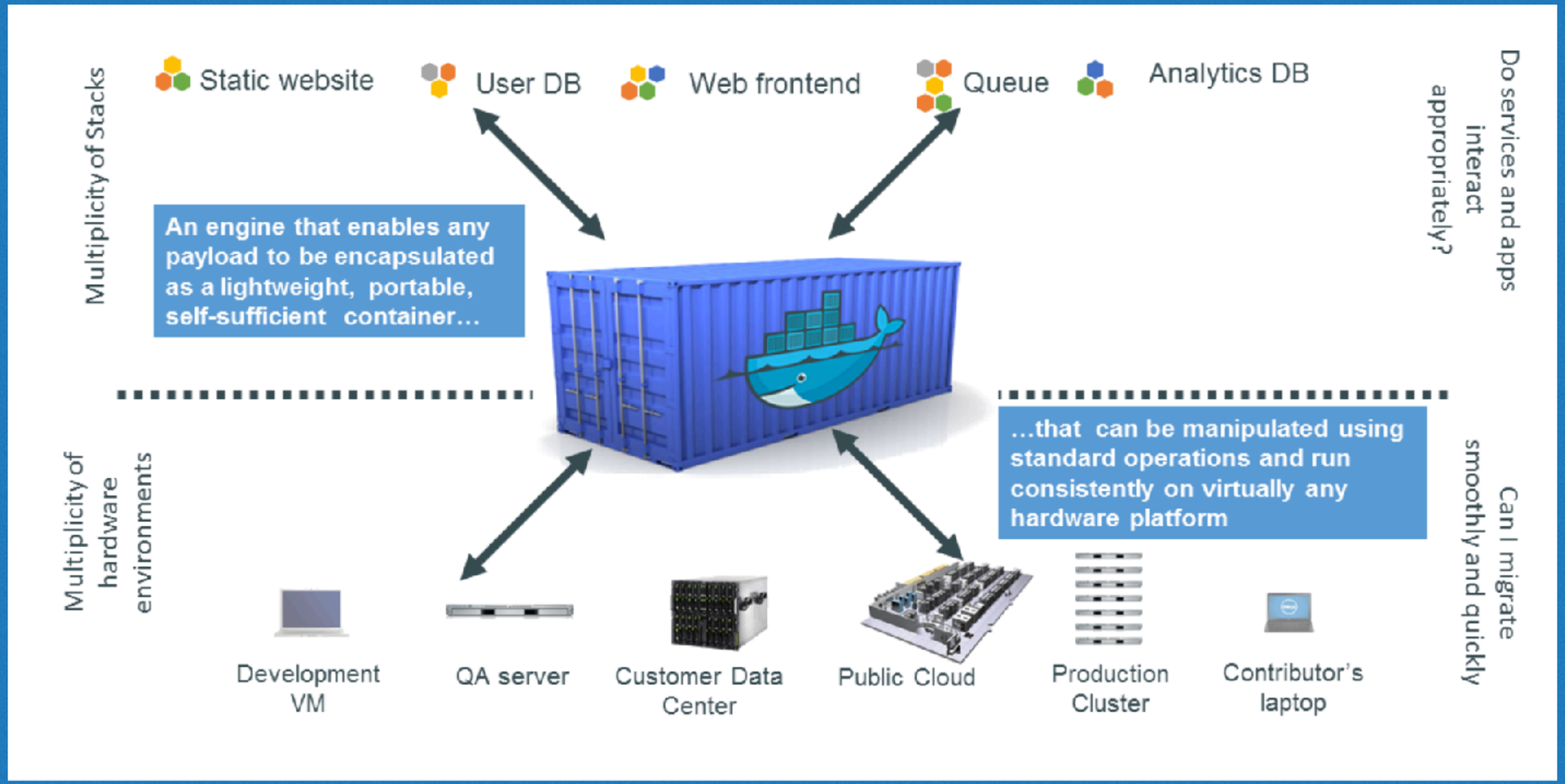


Cargo Transport Pre-1960



Solution: Intermodal Shipping Container

Docker is a Container System for Code



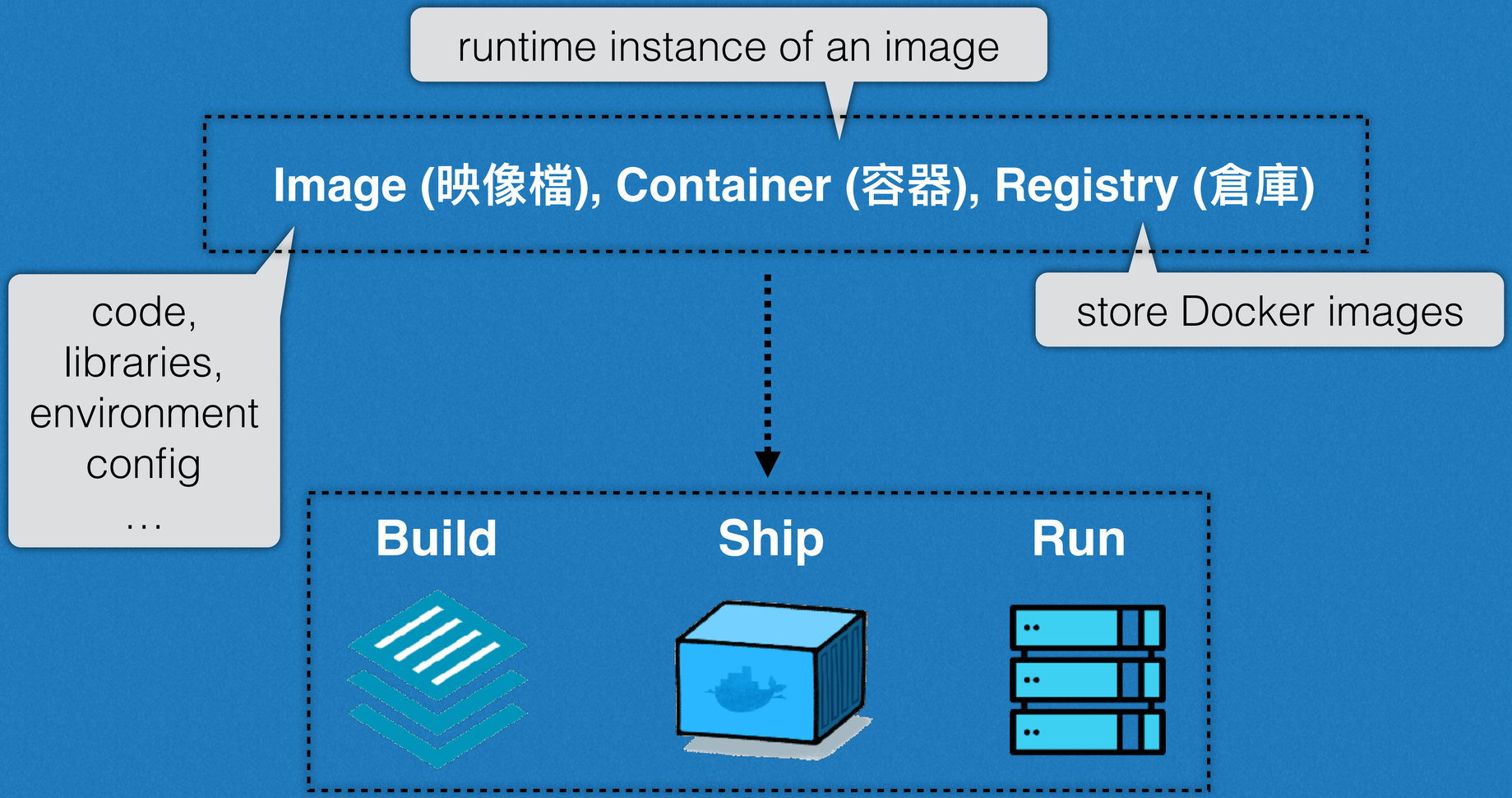
Cathy Hadoop Applications

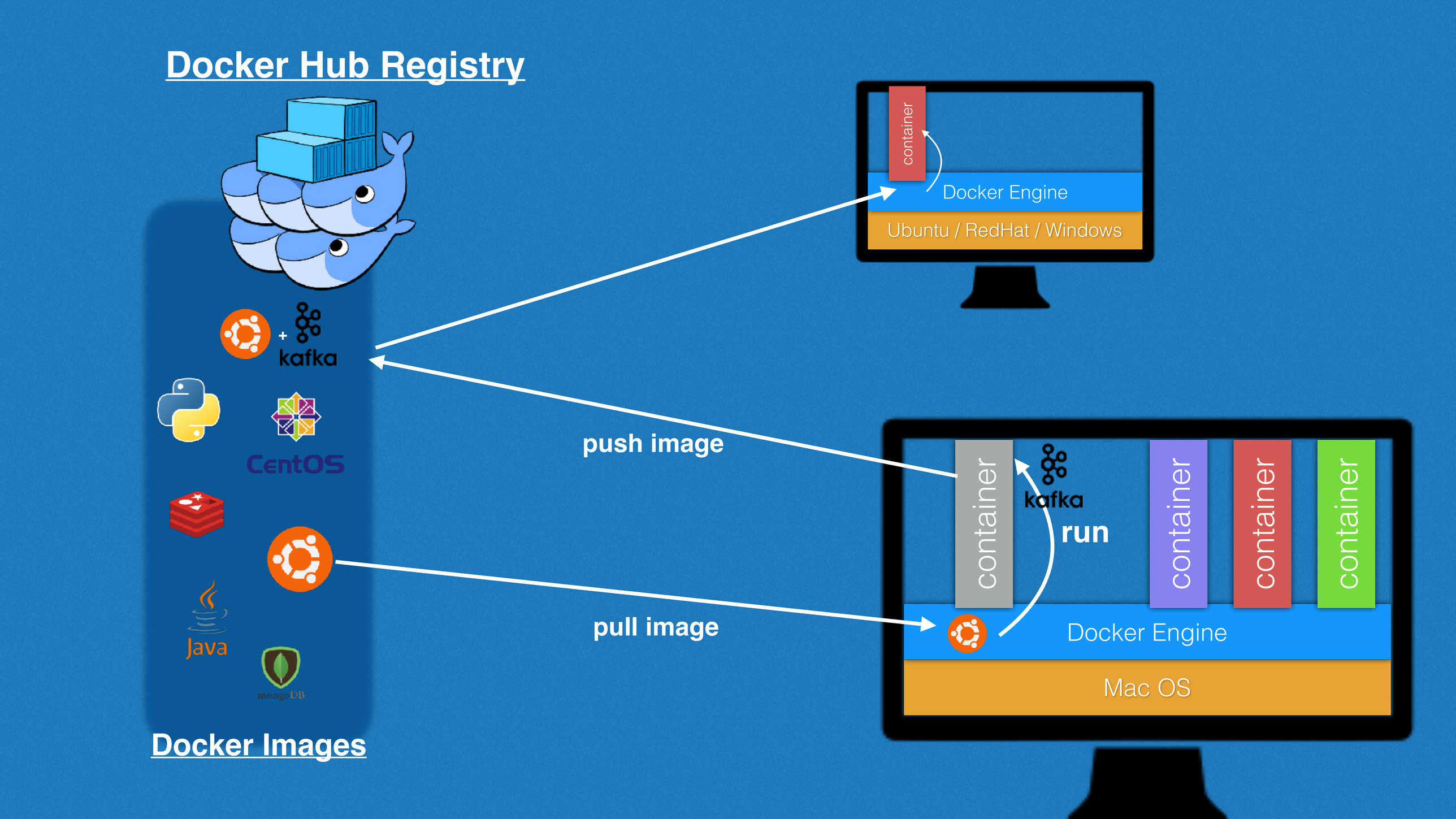
	Dev-Env.	Test-Env. (UAT.)	Prod-Env.
tagging			
Journey			
recommend			
frontier			

Docker，

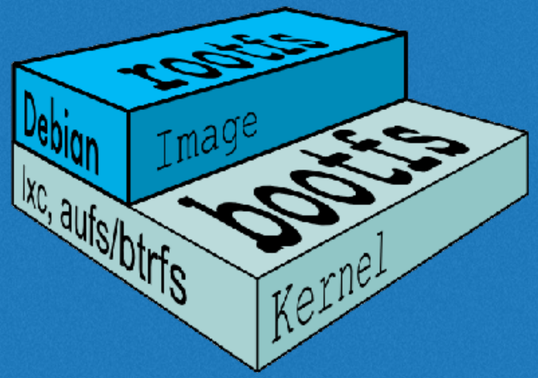
簡化作業流程，讓開發、測試、正式環境無縫接軌

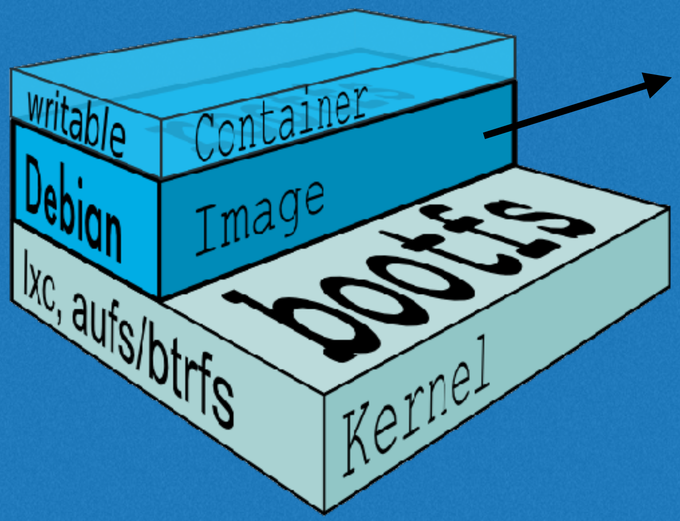
Docker 是 輕量級的作業系統虛擬化 快速部署工具



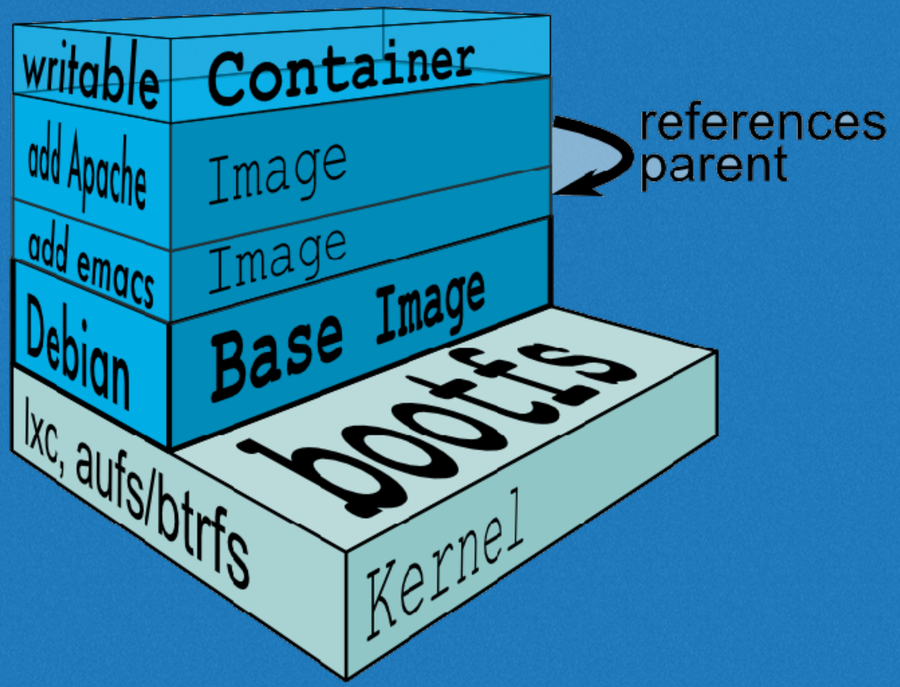

Docker Hub Registry
container
Docker Engine
Ubuntu / RedHat / Windows
kafka
CentOS
Java
mongoDB
push image
pull image
container
kafka
run
container
container
container
Docker Engine
Mac OS
Docker Images

About images, containers



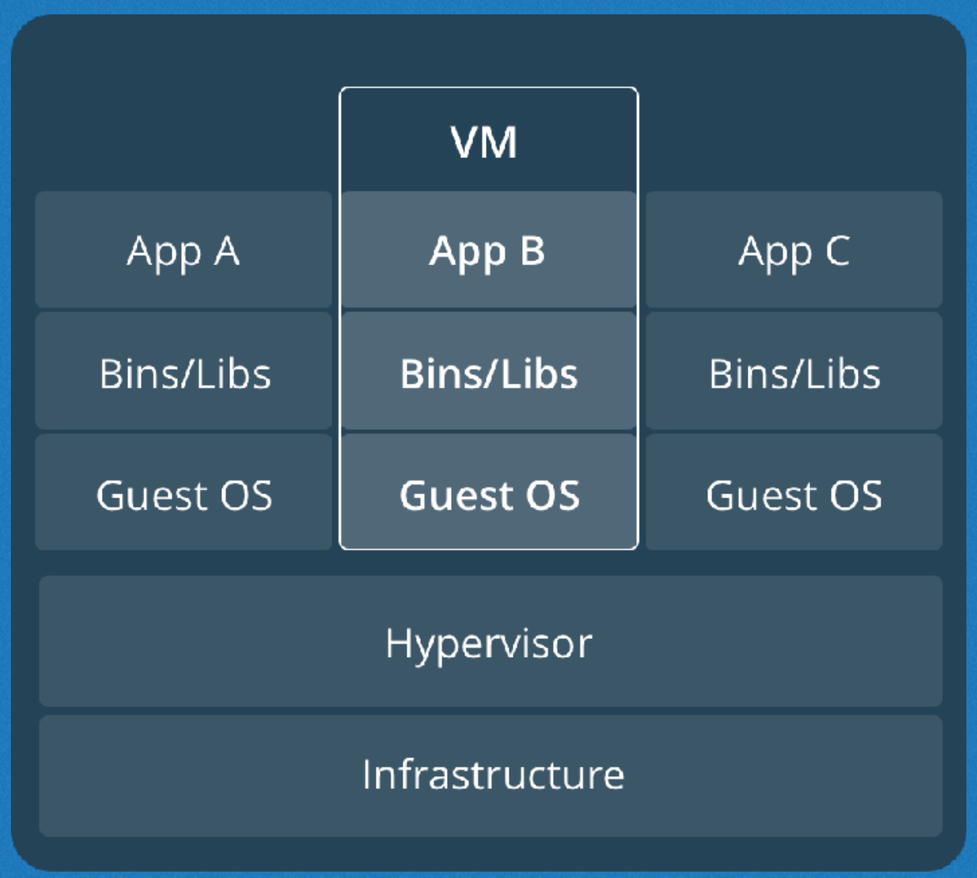


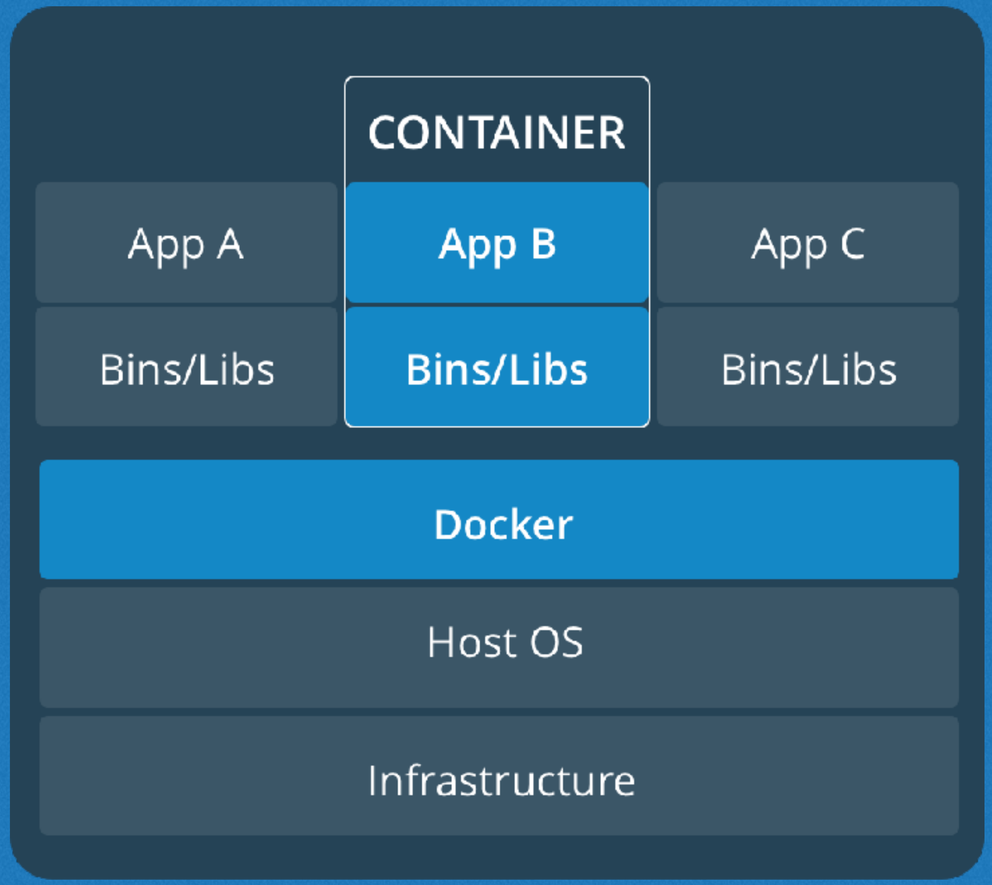
read-only



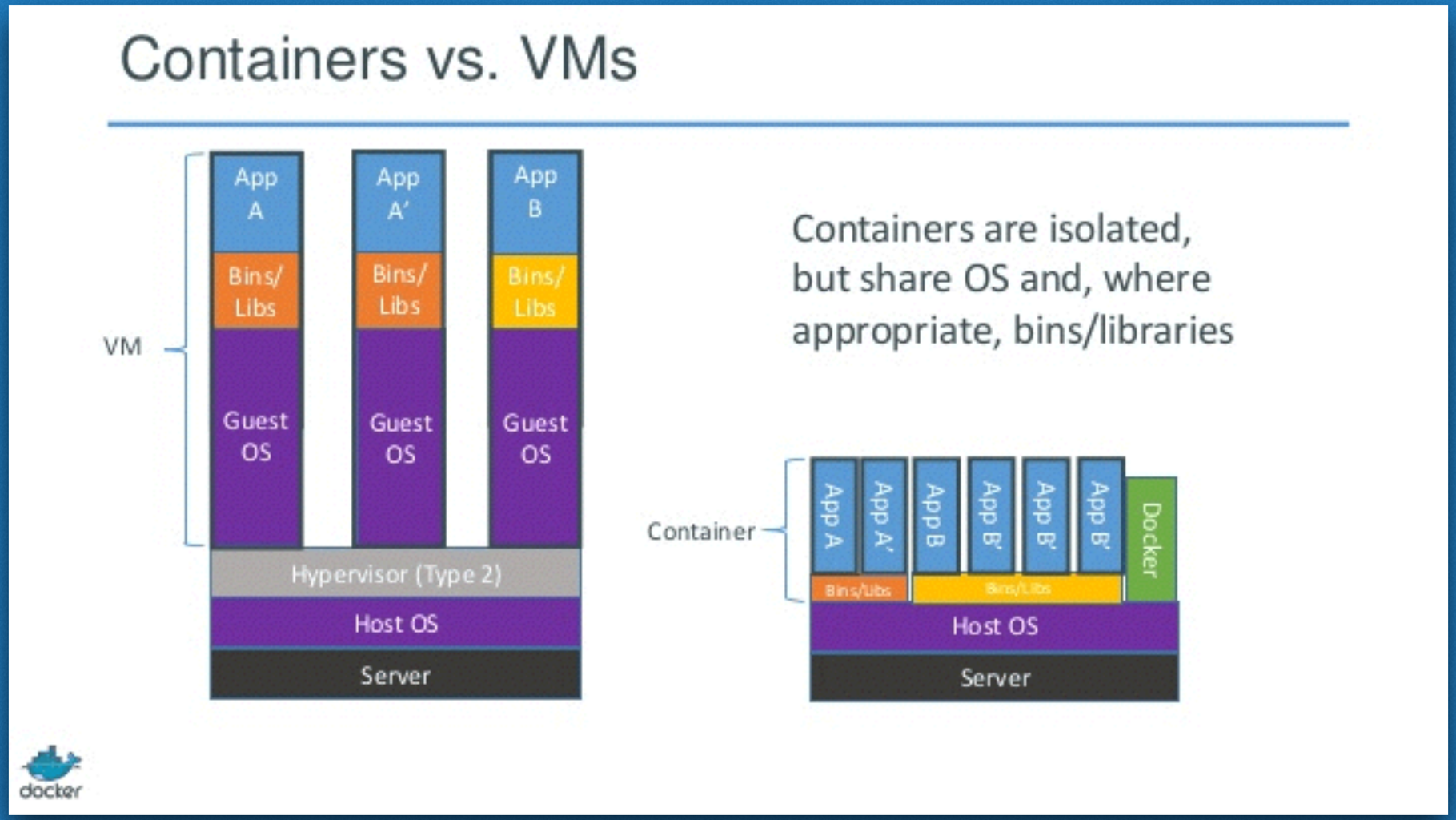
references parent

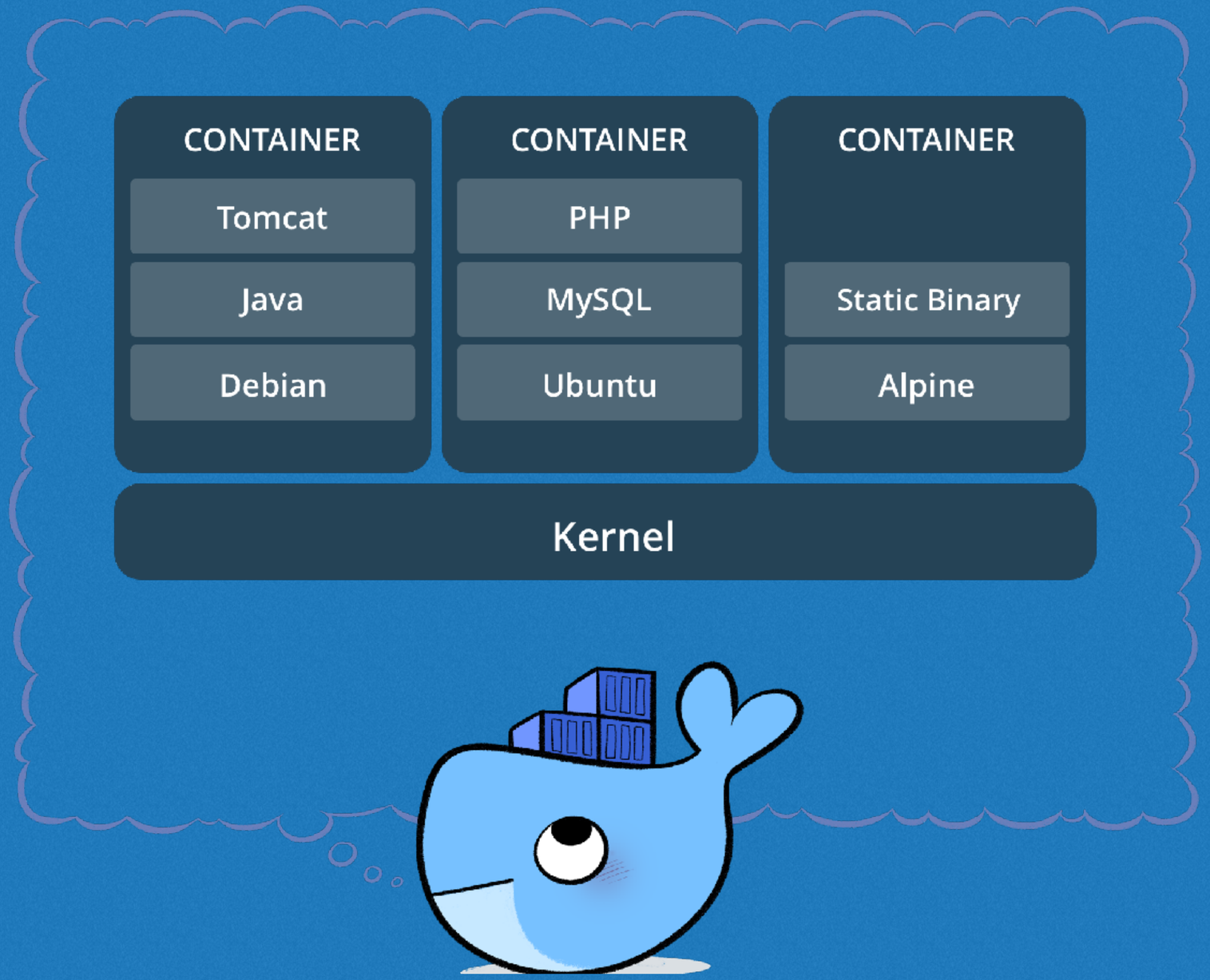
Containers vs. virtual machines





Containers vs. virtual machines




CONTAINER
Tomcat
Java
Debian
CONTAINER
PHP
MySQL
Ubuntu
CONTAINER
Static Binary
Alpine
Kernel

Ok, Let’s get our hands dirty

Part 1. Docker Basic Command

Part 2. Web app(Flask) with Docker

Part 1. Docker Basic Command

Installing Docker

https://docs.docker.com/engine/installation/

Sign up for Dockerhub

https://hub.docker.com/

Using the Docker Command

$ docker --version

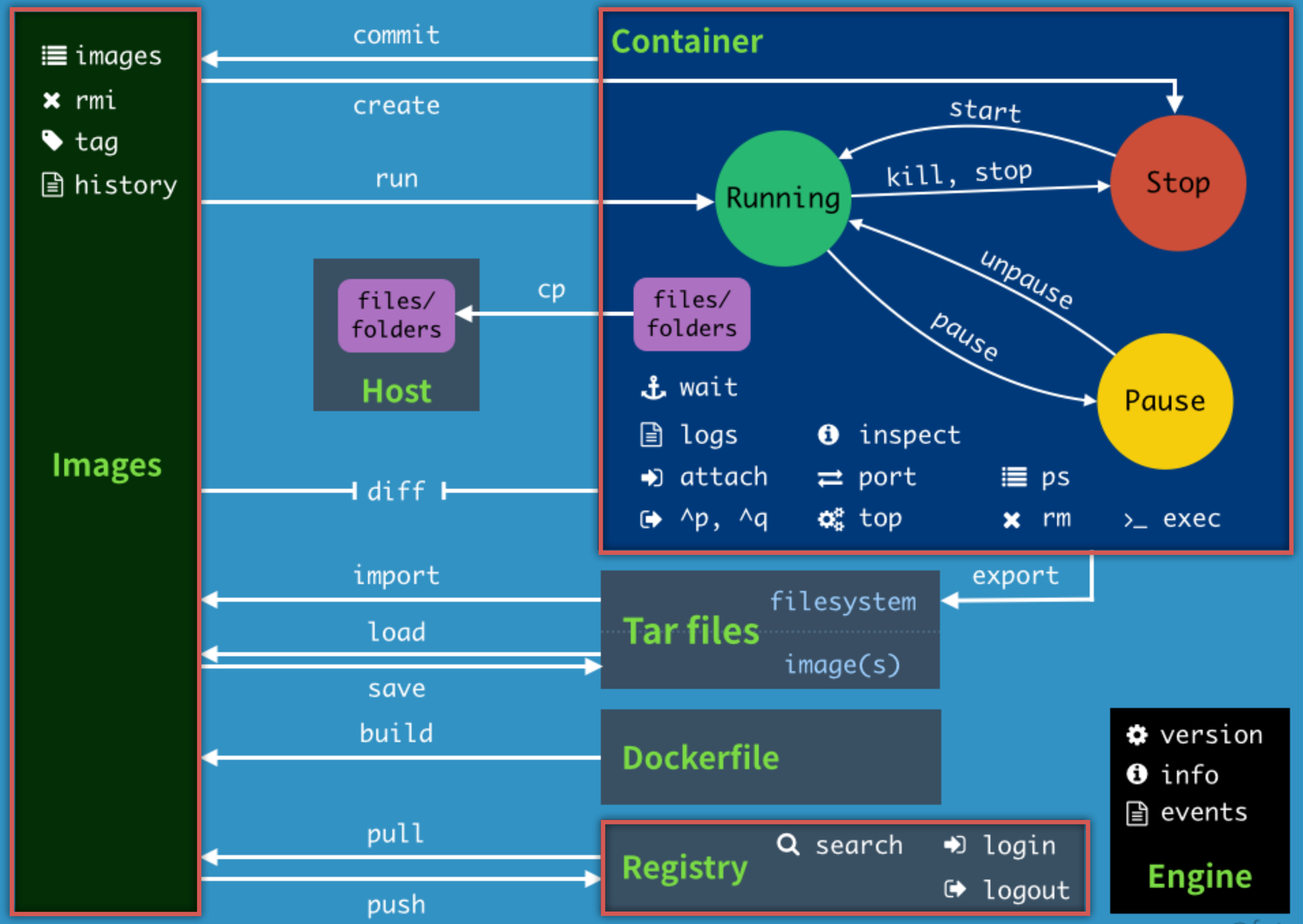
$ docker info

$ docker images

$ docker run hello-world

```
Tse-EndeMacBook-Pro:~ Tse-En$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker Hub account:
 https://hub.docker.com

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/
```

Docker Commands Diagram



// Search the Docker Hub for images

$ docker search tseenliu/ubuntu:sshd

// Pull an image from a registry

$ docker pull tseenliu/ubuntu:sshd

// List images

$ docker images

// Run a command in a new container

$ docker run -it tseenliu/ubuntu:sshd bash

// List containers

$ docker ps

// Stop one or more running containers

$ docker stop containerID / NAME

// Pull an image from a registry

$ docker start containerID / NAME

// Remove one or more containers

$ docker rm containerID / NAME

// Remove one or more images

$ docker rmi image:tag

// Run ubuntu container

$ docker run -d -P --name x tseenliu/ubuntu:sshd

// List port mappings for the container

$ docker port name 22

// ssh to docker ubuntu container

$ ssh root@localhost -p port

// Get a container's IP

$ docker inspect
--format '{{ .NetworkSettings.IPAddress }}'
containerID

// apt-get install …

……

// Create a new image from a container's changes

$ docker commit containerID …

// Push an image to a registry

$ docker push containerID / NAME

// Force the removal of a running container

$ docker rm -f containerID / NAME

The Dockerfile Instructions

// Sets the Base Image
FROM <image>[:<tag>] [AS <name>]

// Sets the Author field
MAINTAINER <name>

// Define an environment variable
ENV

// Stop one or more running containers
WORKDIR

// Remove one or more containers
COPY

// Execute a command and commit to create new image
RUN <command>

//Execute instruction
CMD <command>

// Pull an image from a registry
EXPOSE