EvokeTechnologies

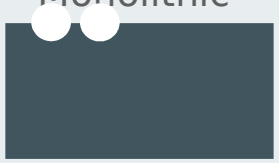
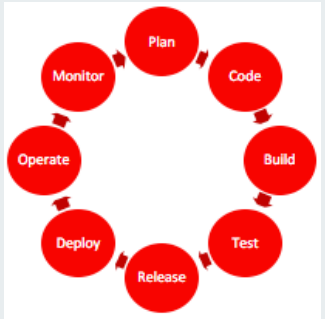
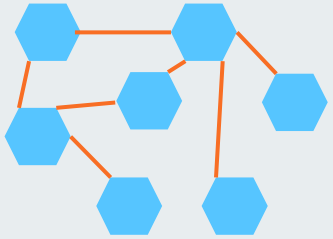
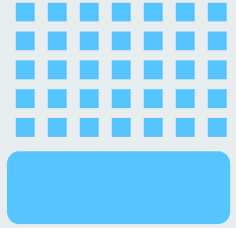
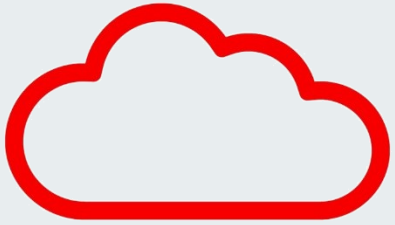
Docker Workshop – 1

By Zama

Agenda

- Evolution of Software dev process
- What is Docker
- History & Docker Community
- Market Adoption
- Features
- Docker ***vs*** Virtual Machine
- Core concepts
- Hands-on
  - Dockerize a simple runnable jar
  - Dockerize spring boot app
  - Setup docker hub (public docker registry) account
  - Upload docker image from local to docker hub

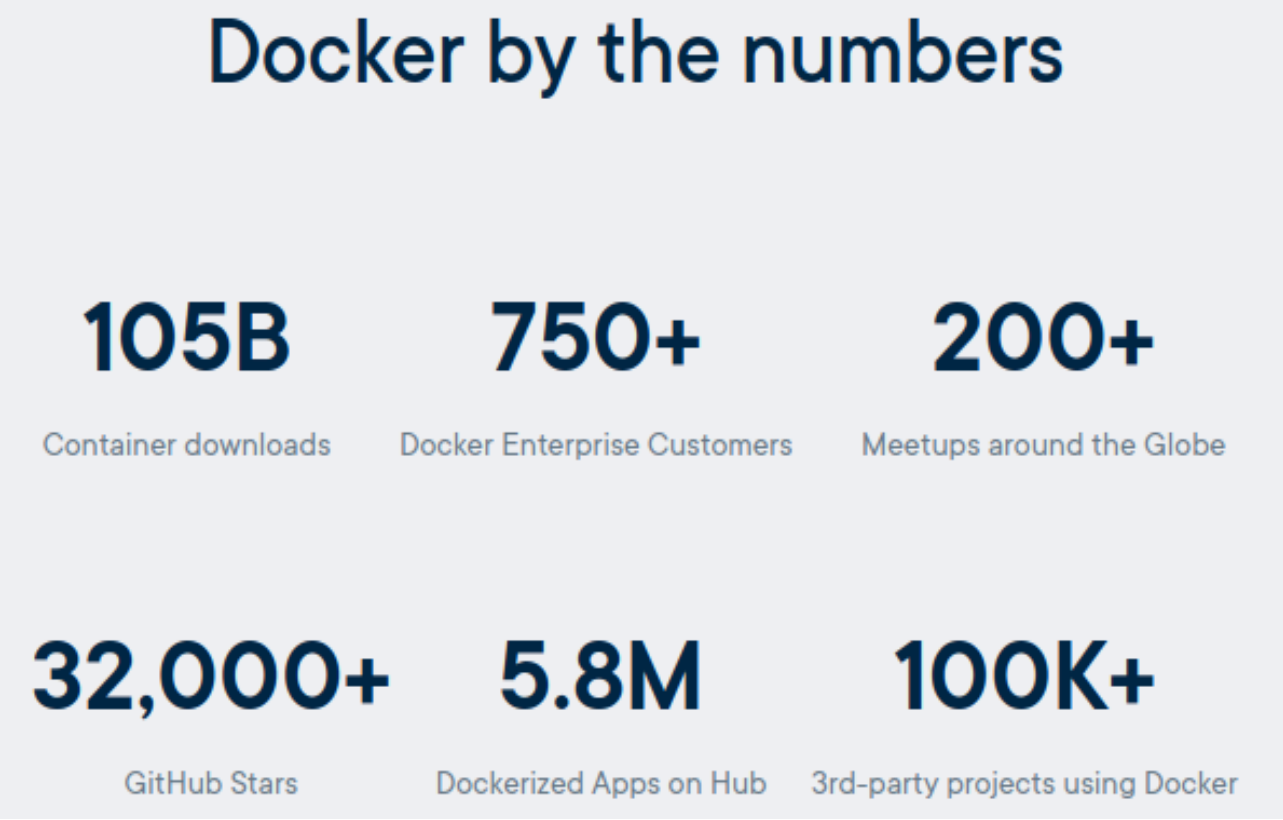
Evolution of Software Dev Process

Development Process	Application Architecture	Deployment and Packaging	Application Infrastructure
Waterfall 	Monolithic 	Physical Server 	Datacenter 
Agile 	N-Tier 	Virtual Servers 	Hosted 
Devops 	Microservices 	Containers 	Cloud 

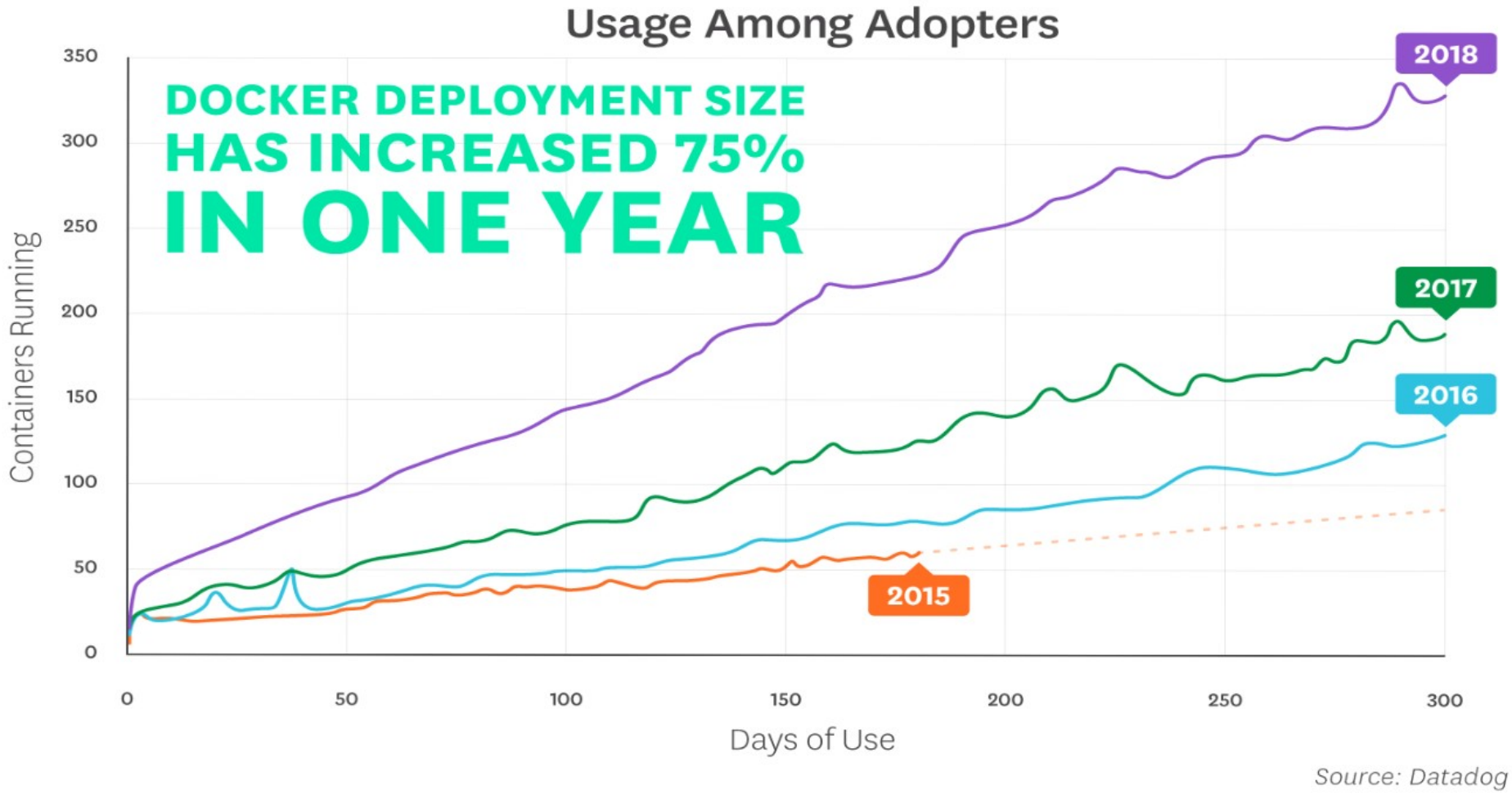
EvokeTechnologies

History & Docker Community

- March 2013: PyCon Lightning talk by *Solomon Hykes* introduces Docker
- Community



Market Adoption



EvokeTechnologies

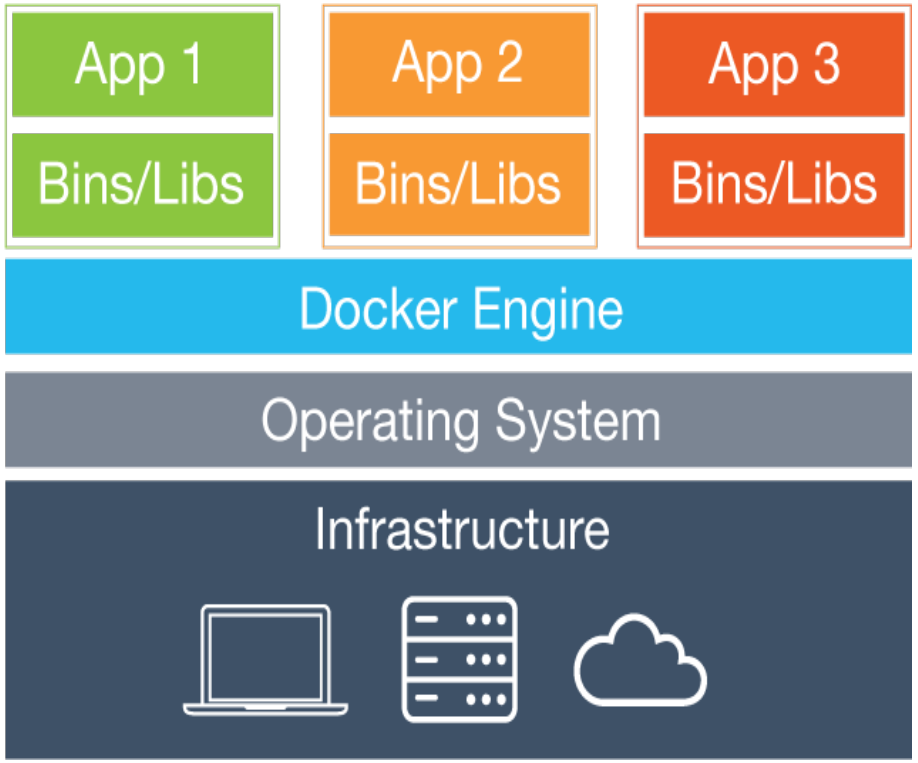
Definition

According to Wikipedia:

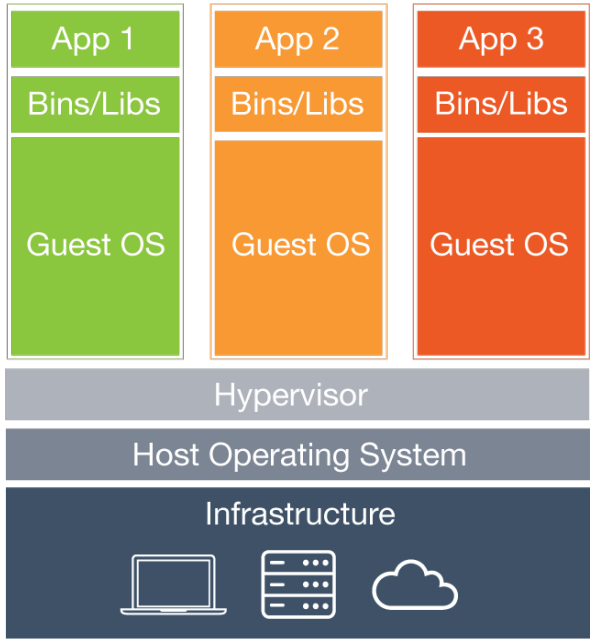
*an open-source project that automates the deployment of software applications inside **containers** by providing an additional layer of abstraction and automation of **OS-level virtualization** on Linux*

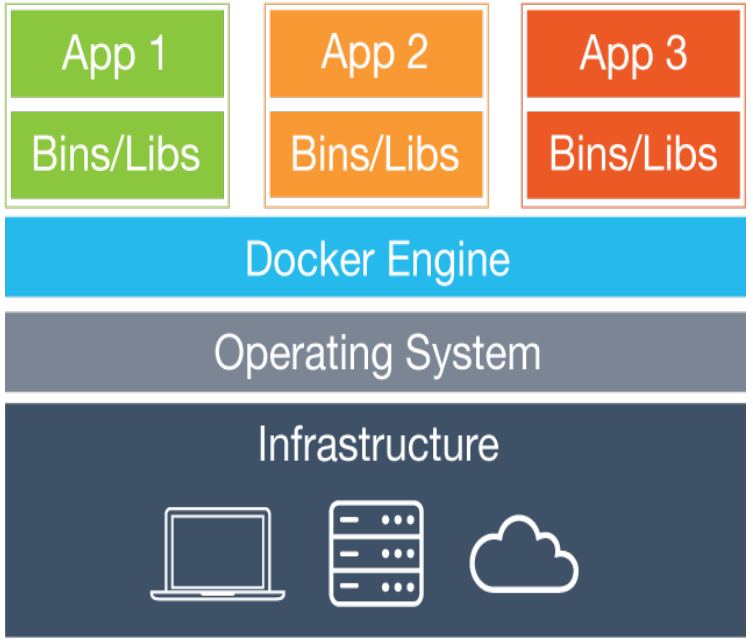
Simplified version:

Docker is a tool that allows developers, sys-admins etc. to easily deploy their applications in a sandbox (called containers) to run on the host operating system



Docker vs Virtual Machine





Virtual Machines

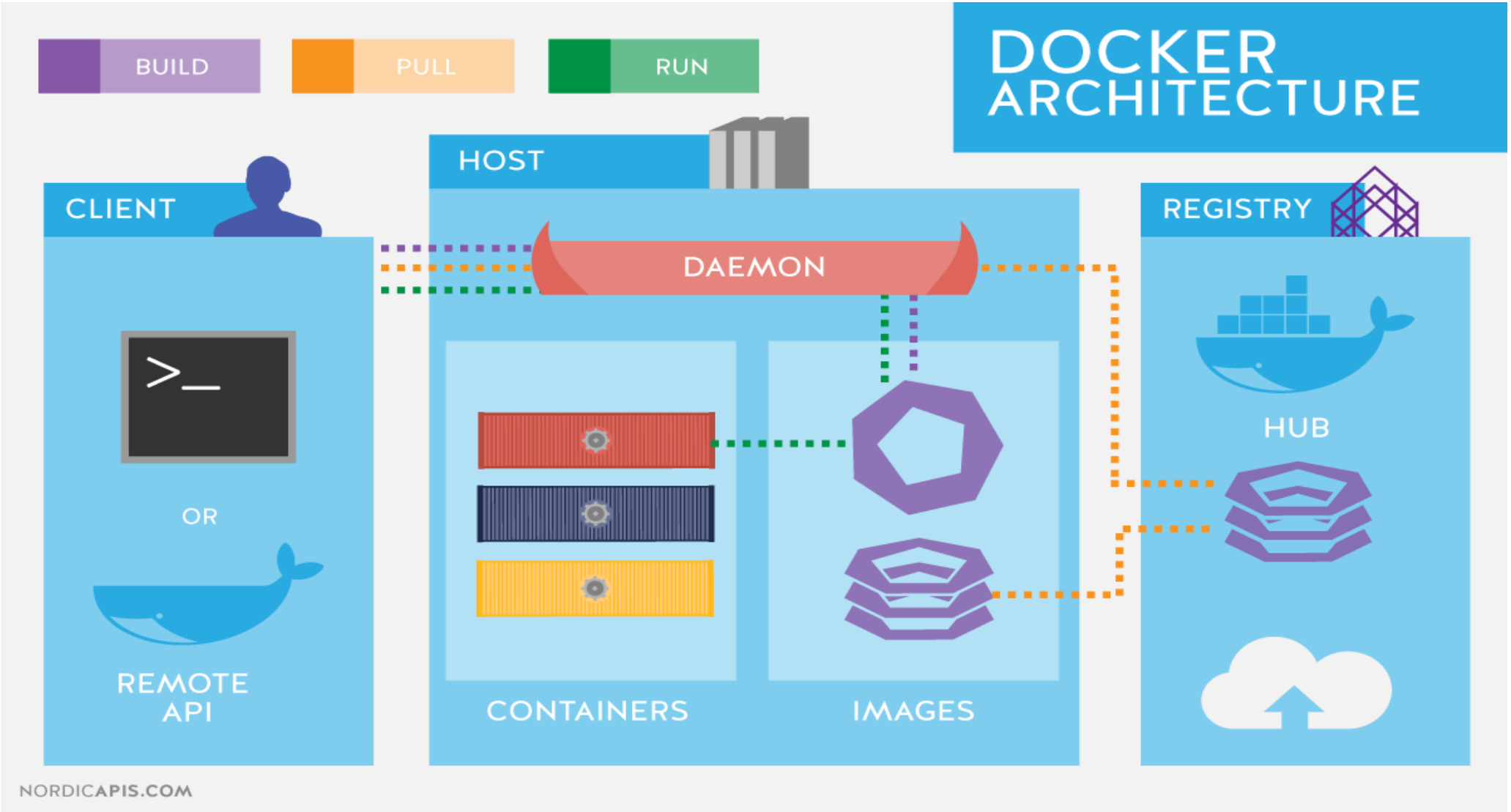
- Each virtual machine (VM) includes the app, the necessary binaries and libraries and an **entire guest operating system**

Containers

- Containers include the app & all of its dependencies, but **share the kernel** with other containers
- Run as an isolated process in userspace on the host OS
- Not tied to any specific infrastructure – containers run on any computer, infrastructure and cloud.

EvokeTechnologies

Core Concepts: Architecture

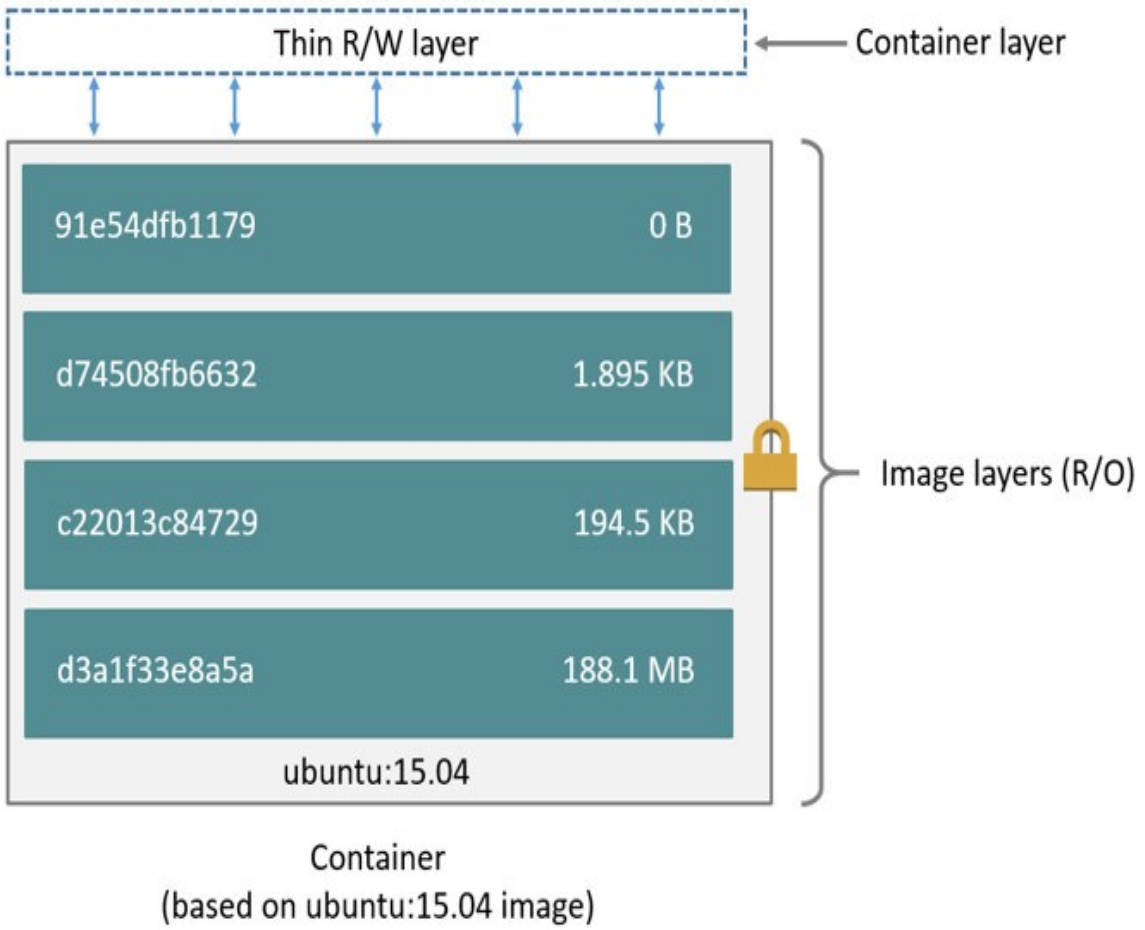


EvokeTechnologies

Core Concepts: Image

Dockerfile

```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```



EvokeTechnologies

Core concepts: Container Life-cycle

- Conception: BUILD an Image from a Dockerfile
- Birth: RUN (create and start) a container
- Sleep: STOP a running container
- Wake: START a stopped container
- Death: RM (delete) a stopped container

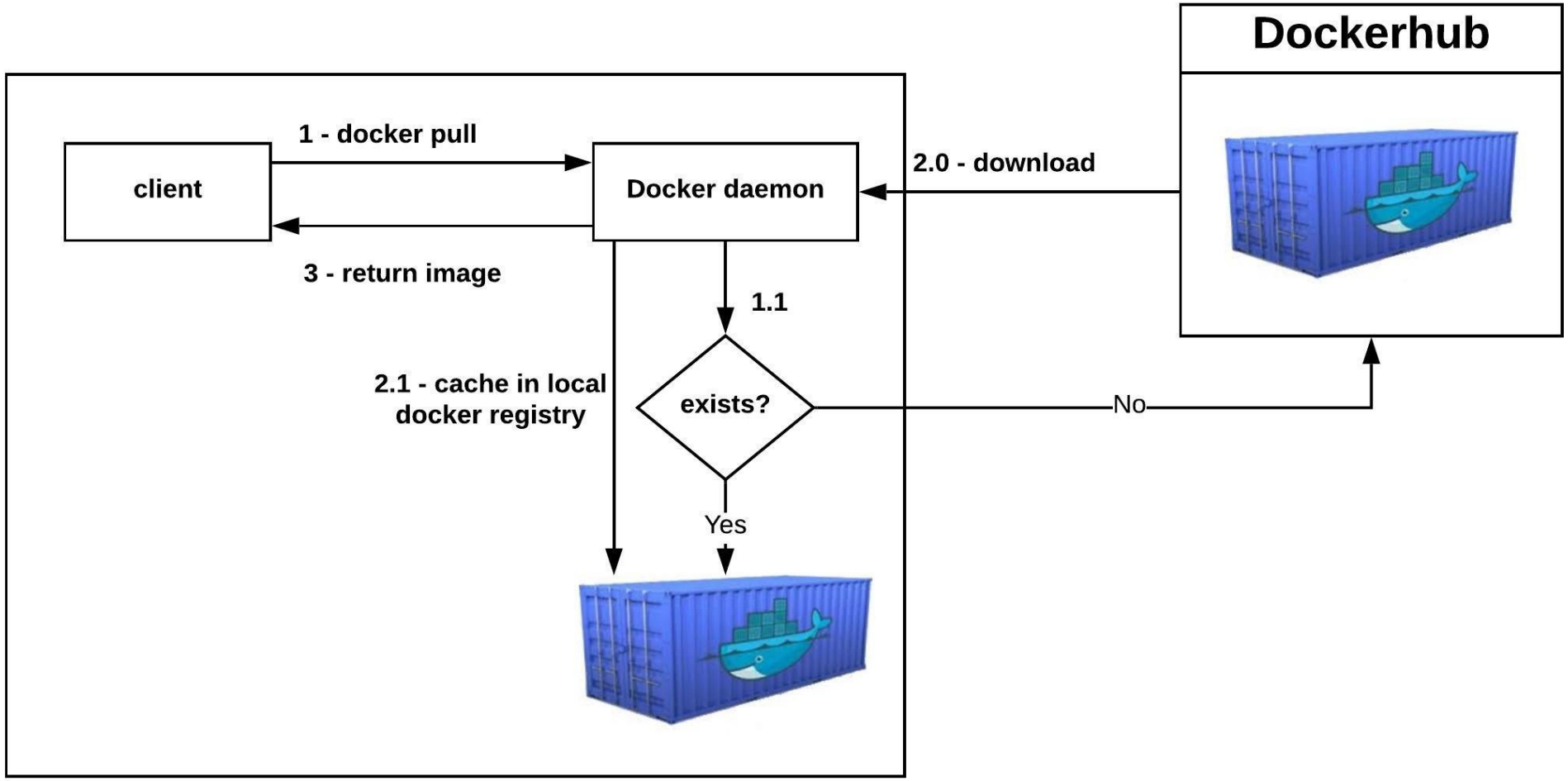
Hands-on: Assignment 1 – Runnable jar

- Run the app without docker
- Dockerize the runnable jar
- Build docker image
- Run the app with docker
- Push the image to Docker Hub (Public docker registry)
- Verify in another machine

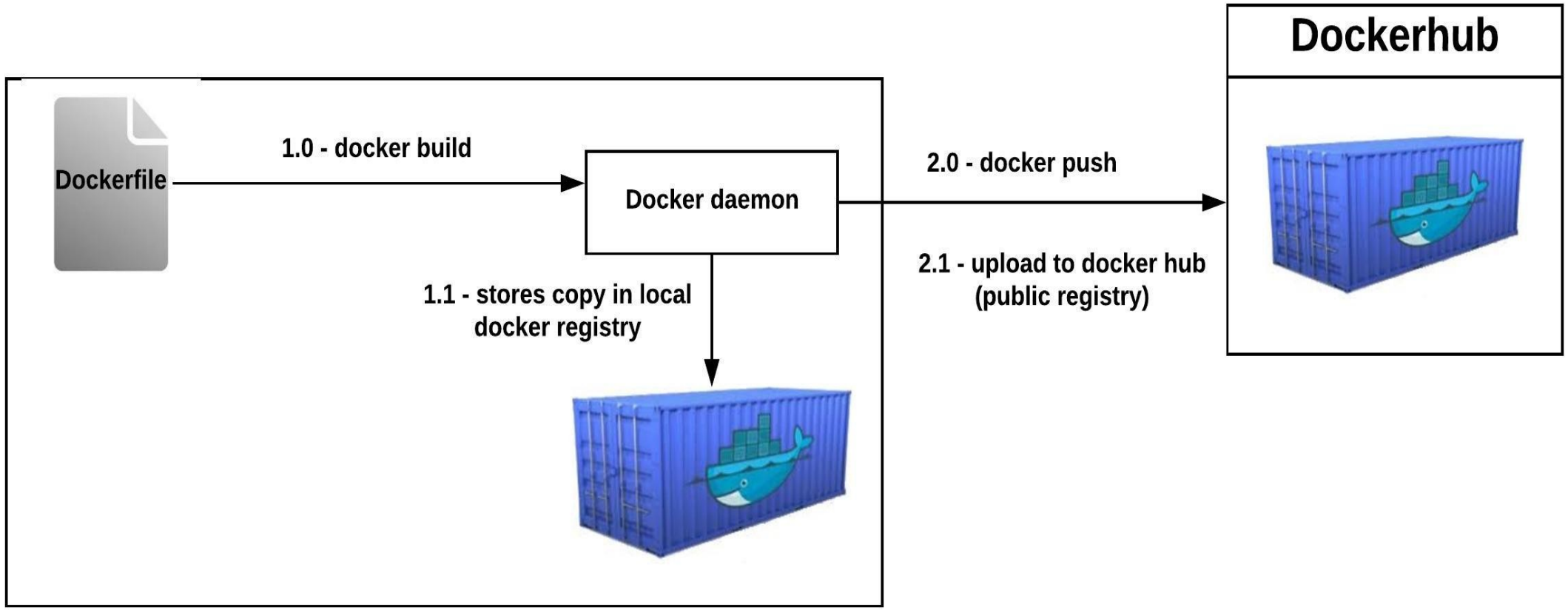
Hands-on: Assignment 2 – Spring boot

- Run the spring boot app without docker
- Dockerize the spring boot app
- Build docker image
- Run the app with docker
- Push the image to Docker Hub (Public docker registry)
- Verify in another machine

Hands-on: Download docker image



Hands-on: Upload docker image



Resources

- Instructions and source code for the hands-on is available at this location:

https://github.com/mbzama/docker-training/

Thank You

EvokeTechnologies