Developer Student Clubs

Docker 101

An Introduction to Docker and Containers

Parshant Sharma
@parshnt_

Google Developers

The Problem

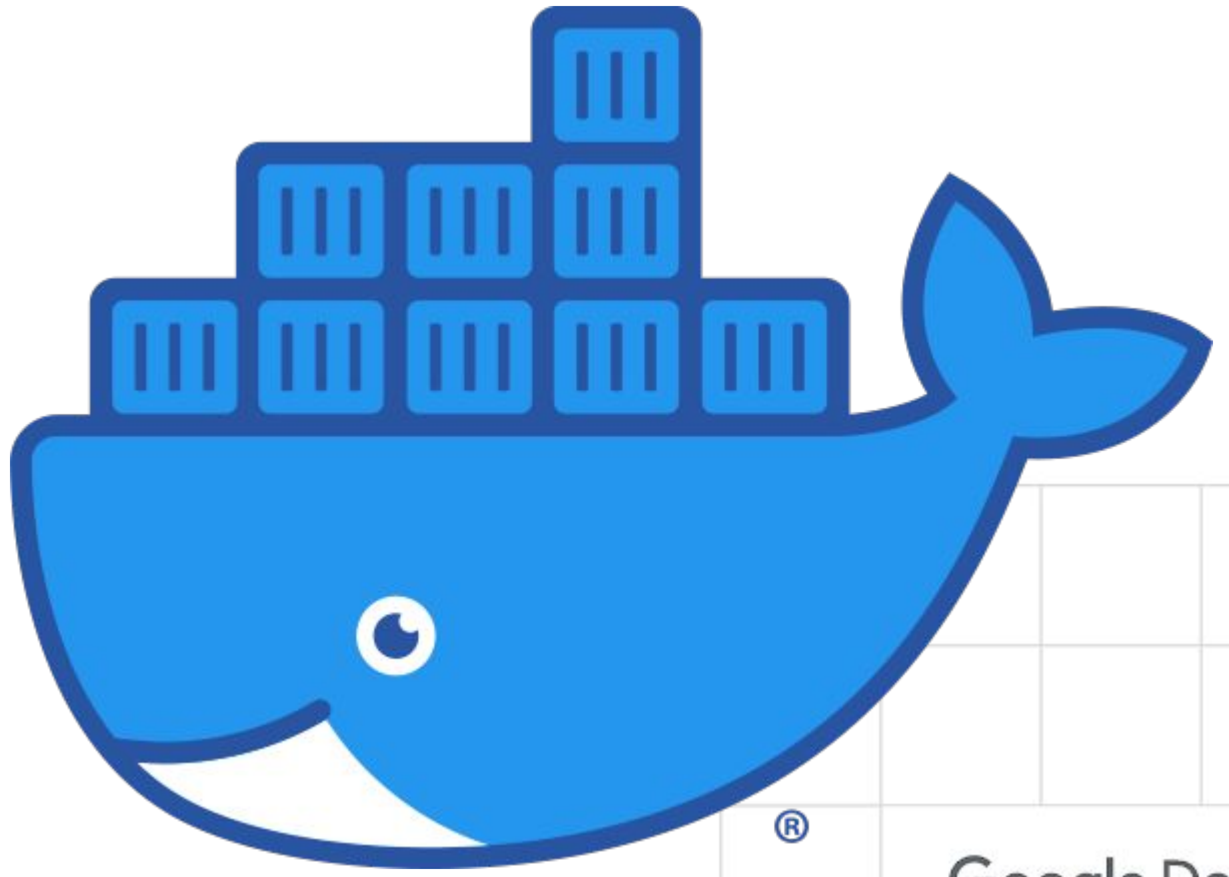
The Solution

What is Docker ?

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

Docker containers are not Virtual Machines



Virtual Machines Vs Containers

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware.

Containers are more portable and efficient.

Hypervisor

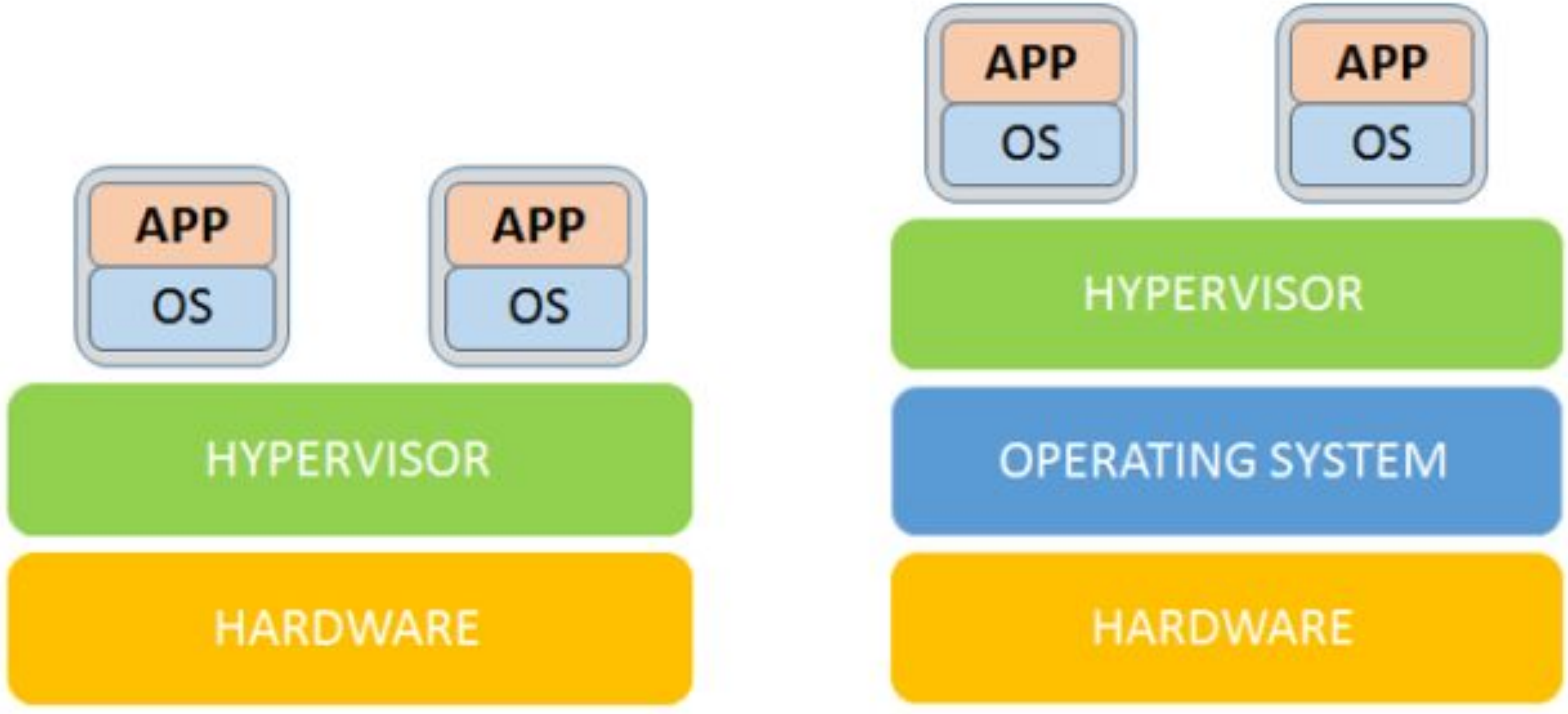
A hypervisor, is a process that creates and runs virtual machines (VMs)

A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, like memory and processing.

Generally, there are two types of hypervisors.

Type 1 hypervisors, run directly on the host’s hardware.
Type 2 hypervisors,run as a software layer on an operating system, like programs.

Overview : T1 Vs T2



Type 1 Hypervisor

Type 2 Hypervisor

Virtual Machines

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.

The hypervisor allows multiple VMs to run on a single machine.

Each VM includes a full copy of an operating system, the application, necessary binaries and libraries - taking up tens of GBs.

VMs can also be slow to boot.

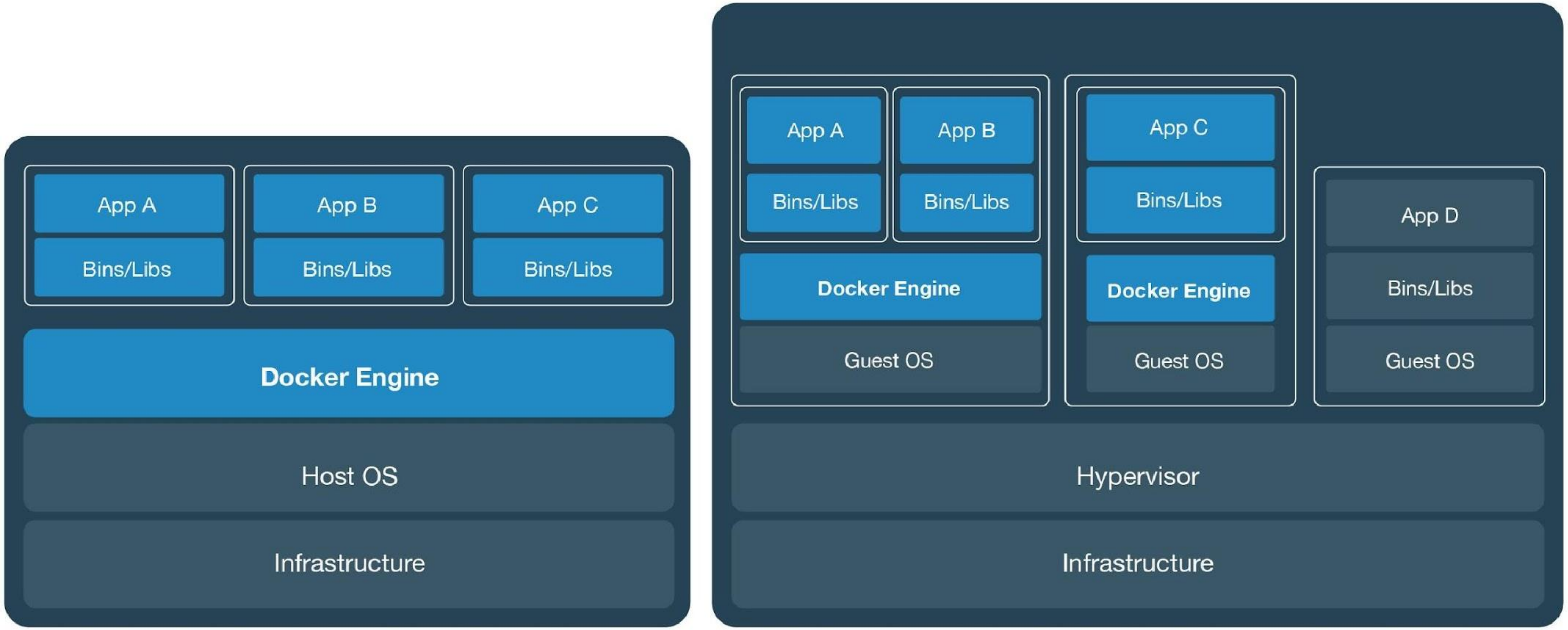
Containers

Containers are an abstraction at the app layer that packages code and dependencies together.

Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space.

Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

An Overview

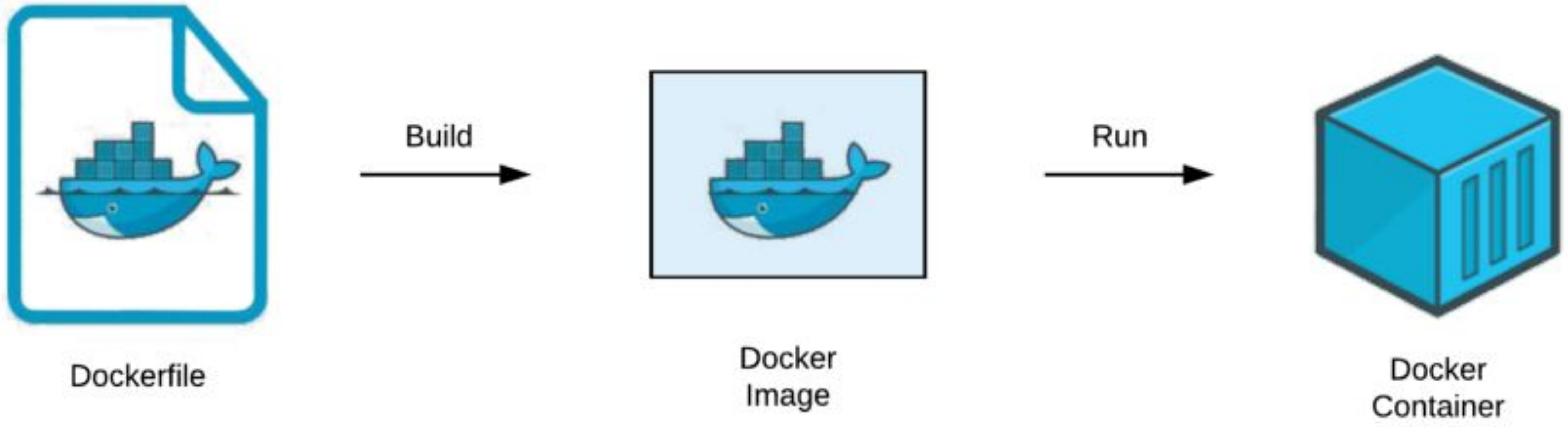

App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker Engine
Host OS
Infrastructure
App A
App B
Bins/Libs
Bins/Libs
Docker Engine
Guest OS
App C
Bins/Libs
Docker Engine
Guest OS
App D
Bins/Libs
Guest OS
Hypervisor
Infrastructure

Some Docker vocabulary

Image	a package defining a working environment
Container	an instance of an image / a running image
Dockerfile	a file with a series of commands to build an image
Docker Hub	a website/repository for sharing Docker images

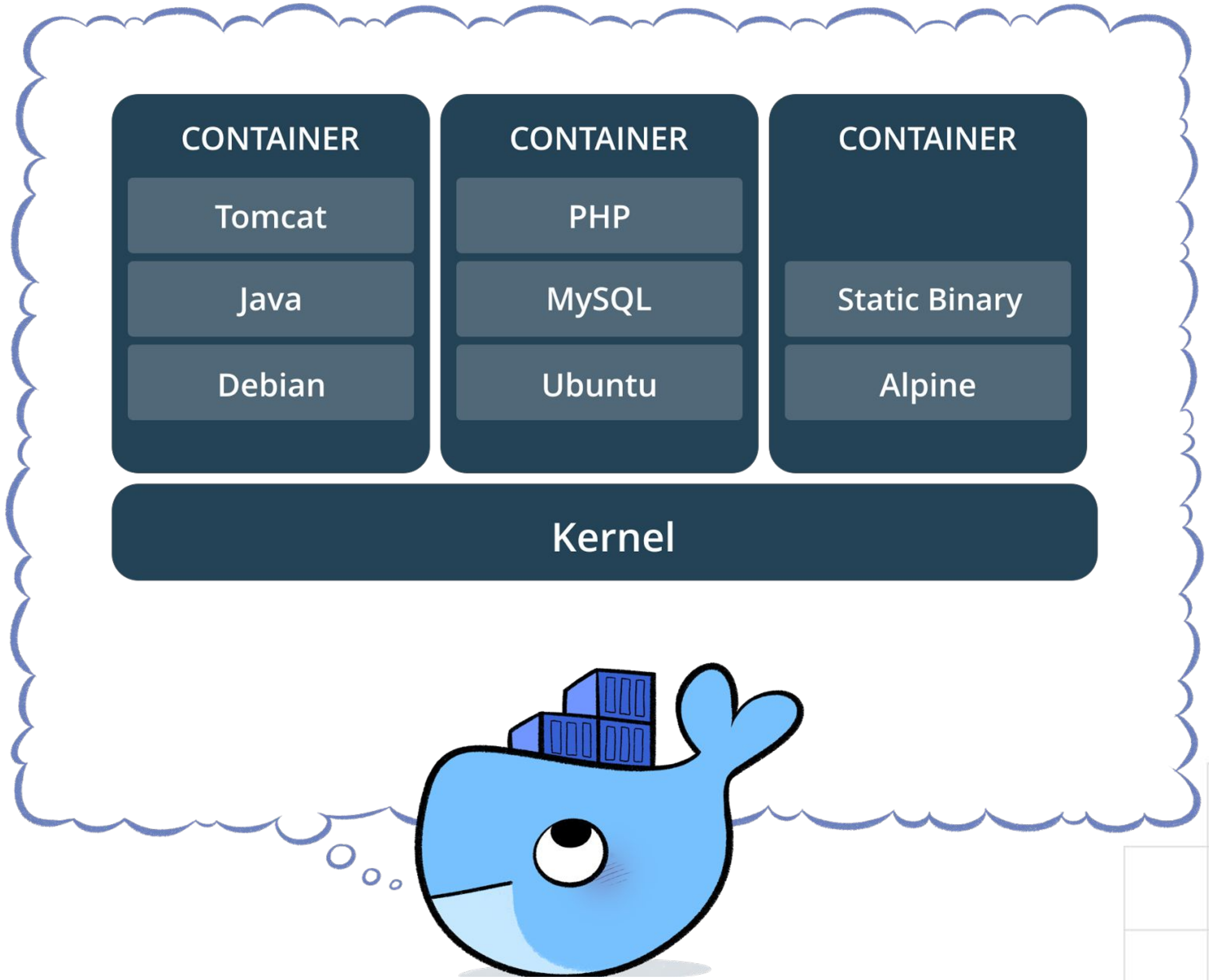
Image

An image becomes container at runtime



Container

A standardized unit of software



CONTAINER

Tomcat

Java

Debian

CONTAINER

PHP

MySQL

Ubuntu

CONTAINER

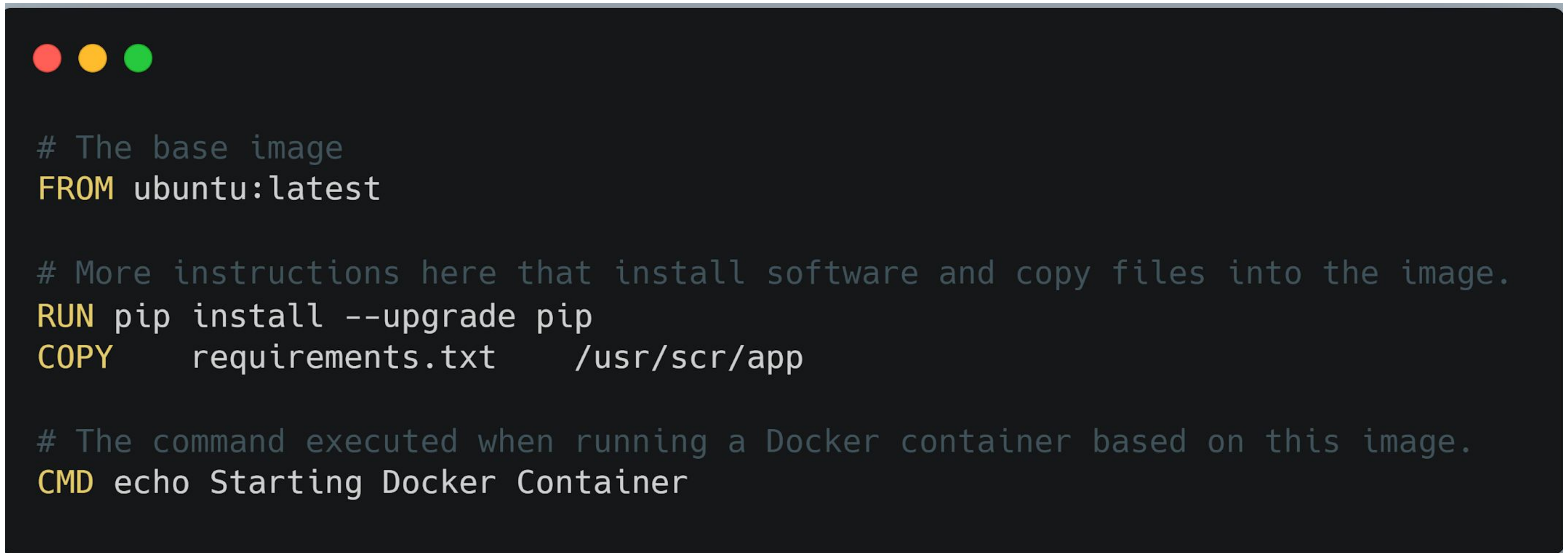
Static Binary

Alpine

Kernel

Dockerfile

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.



```
# The base image
FROM ubuntu:latest

# More instructions here that install software and copy files into the image.
RUN pip install --upgrade pip
COPY    requirements.txt    /usr/scr/app

# The command executed when running a Docker container based on this image.
CMD echo Starting Docker Container
```

Docker Hub

Docker Hub is a service provided by Docker for finding and sharing container images with your team.

https://hub.docker.com/

Basic Docker Commands

docker build

builds an image from a Dockerfile

Stores it to your local machine (computer)

docker images

displays all Docker images on that machine

docker run

starts container and runs any commands in that container

additional options:

- -p — allows you to specify ports in host and Docker container
- -d — starts the container in daemon mode (it runs in a background process)

docker rmi

removes one or more images

docker rm

removes one or more containers

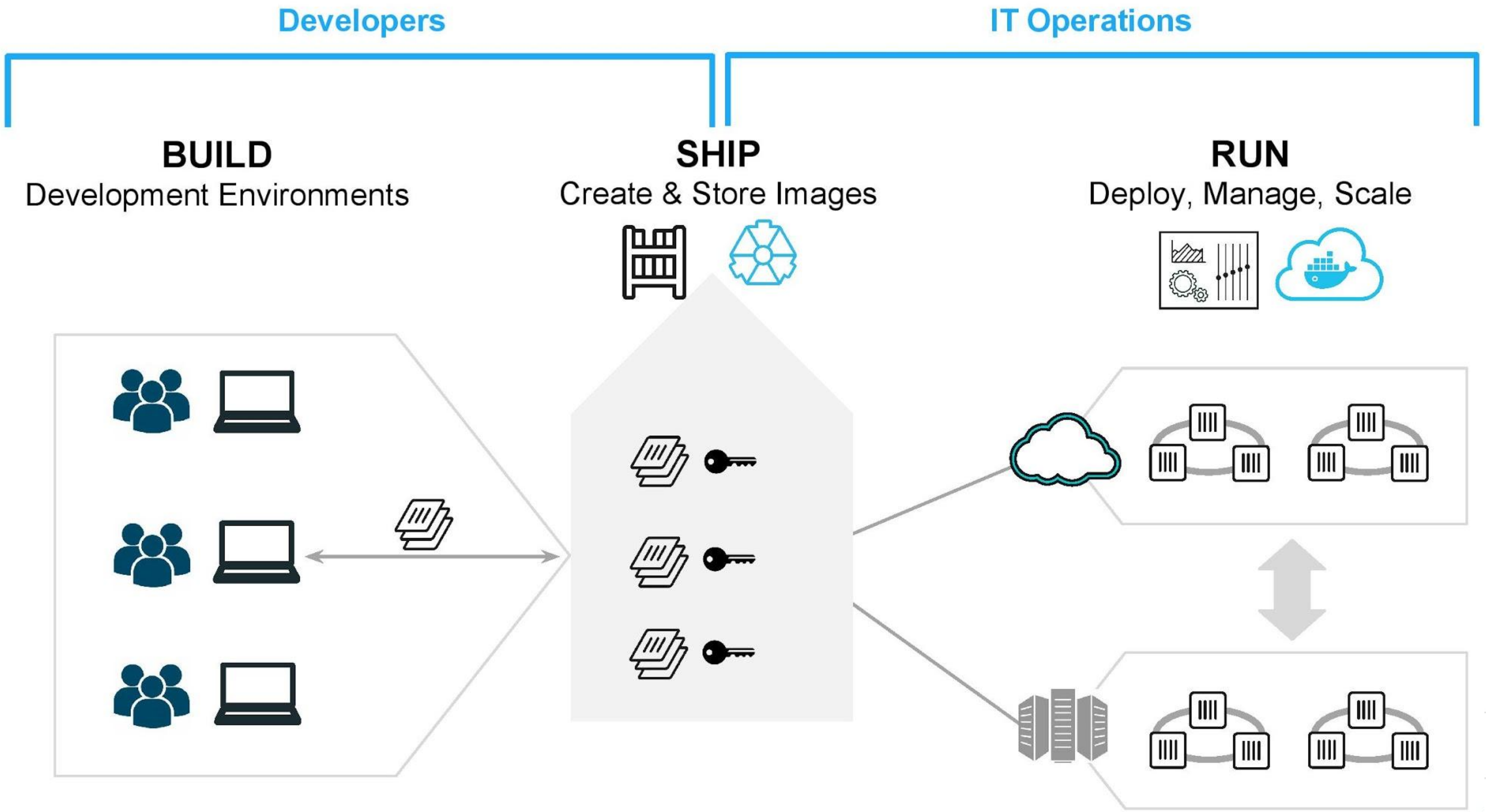
docker kill

kills one or more running containers

docker ps

displays a list of running containers

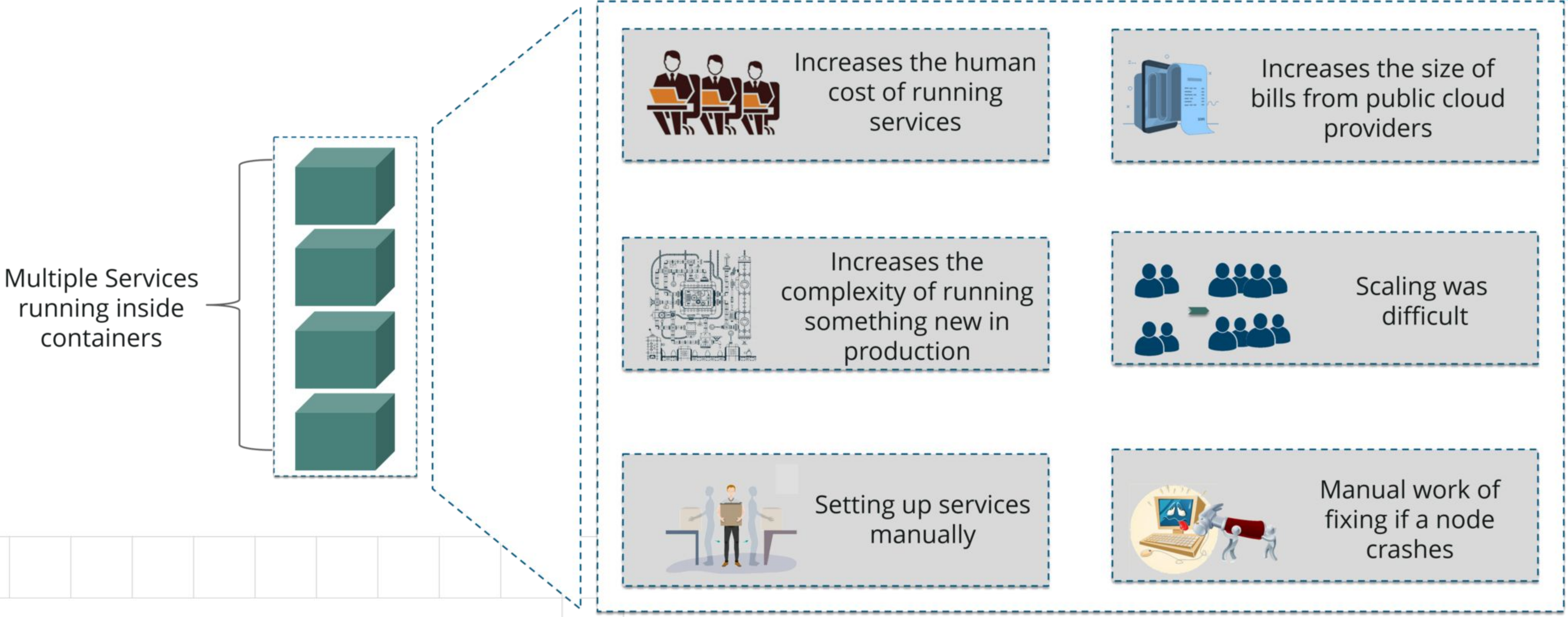
Using Docker



Your First Container 🐳

[Learn Docker & Containers using Interactive Browser-Based Labs]

Problems with Containers



Multiple Services running inside containers

Increases the human cost of running services

Increases the size of bills from public cloud providers

Increases the complexity of running something new in production

Scaling was difficult

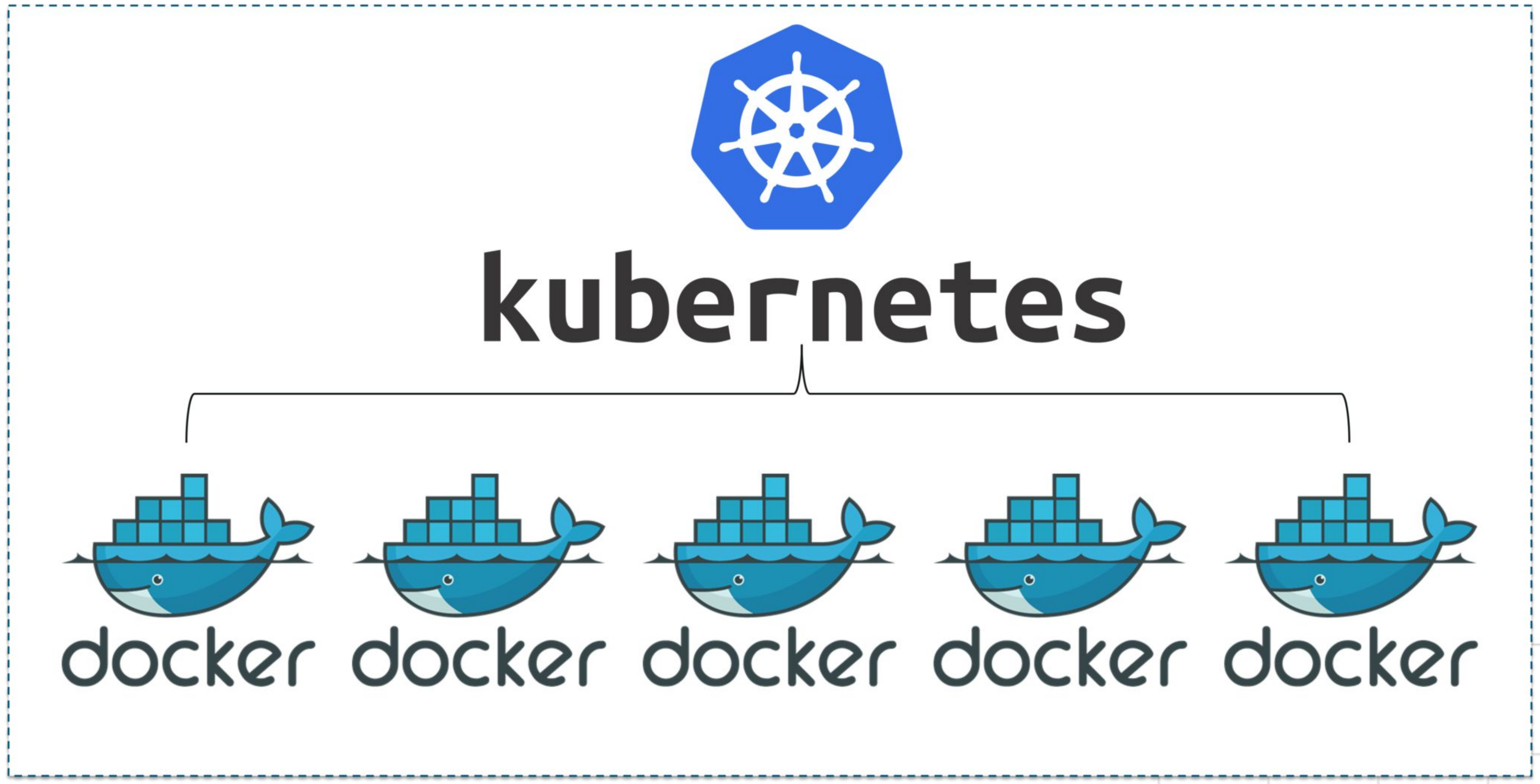
Setting up services manually

Manual work of fixing if a node crashes

Solution a.k.a Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

It was originally designed by Google.

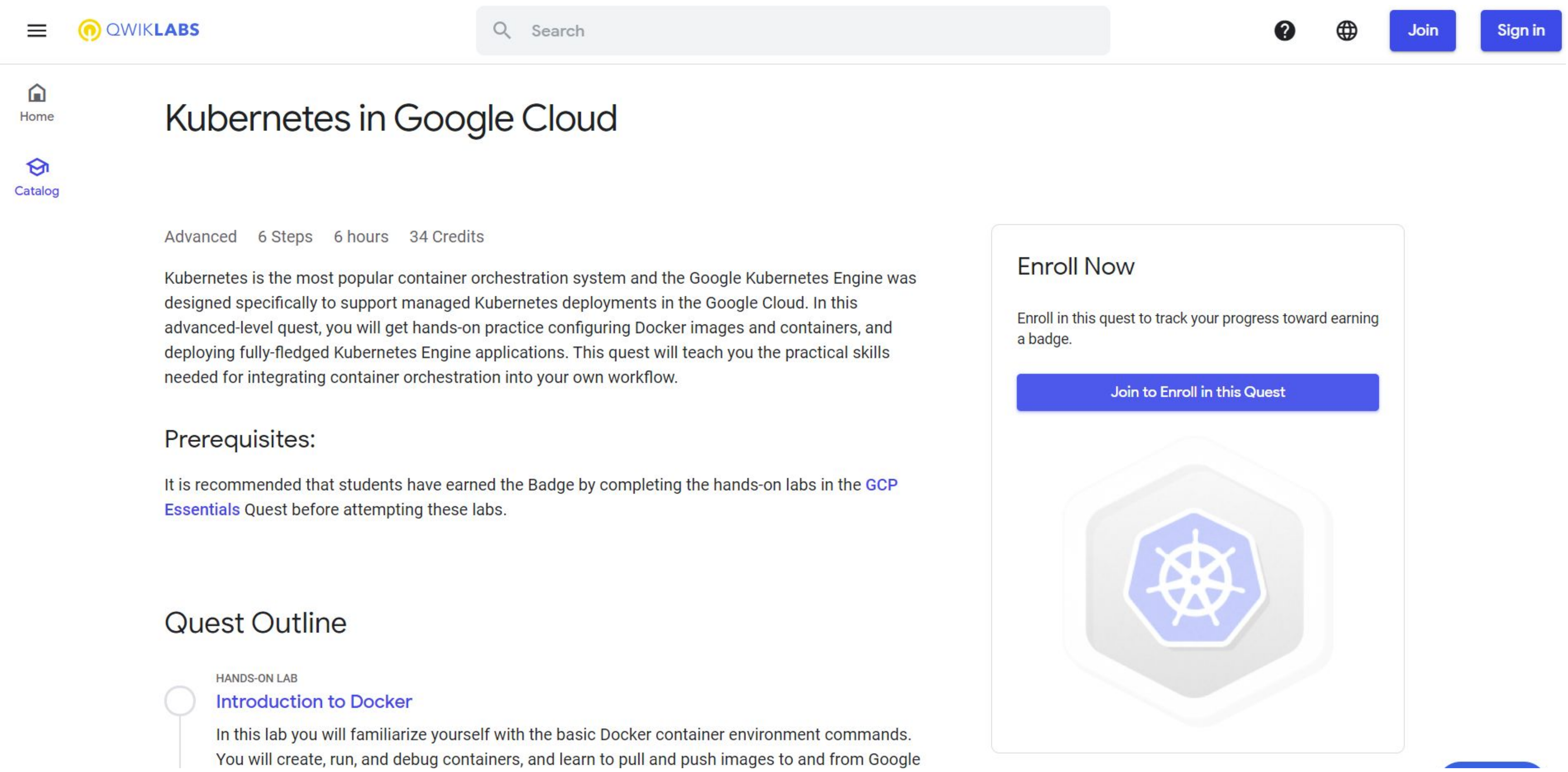


Keep learning!

It’s only the tip of the iceberg.

Practice for free on GCP
https://go.qwiklabs.com/qwiklabs-free

And earn Badges!
https://www.qwiklabs.com/quests/29



Kubernetes in Google Cloud

Advanced 6 Steps 6 hours 34 Credits

Kubernetes is the most popular container orchestration system and the Google Kubernetes Engine was designed specifically to support managed Kubernetes deployments in the Google Cloud. In this advanced-level quest, you will get hands-on practice configuring Docker images and containers, and deploying fully-fledged Kubernetes Engine applications. This quest will teach you the practical skills needed for integrating container orchestration into your own workflow.

Prerequisites:

It is recommended that students have earned the Badge by completing the hands-on labs in the GCP Essentials Quest before attempting these labs.

Quest Outline

HANDS-ON LAB
Introduction to Docker

In this lab you will familiarize yourself with the basic Docker container environment commands. You will create, run, and debug containers, and learn to pull and push images to and from Google

Enroll Now

Enroll in this quest to track your progress toward earning a badge.

Join to Enroll in this Quest

Developer Student Clubs

Any Questions

Developer Student Clubs

Thank You!

Parshant Sharma
@parshnt_