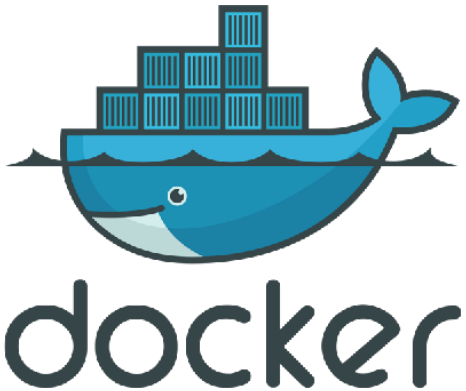
LAB-01

Introduction to Docker

Olarn Rojanapornpun

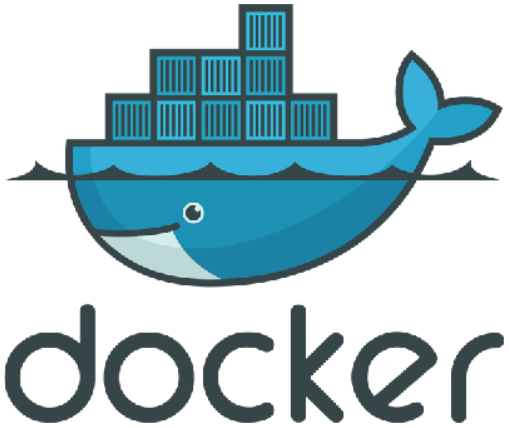
Docker is used to run **software packages** called "containers". **Containers are isolated** from each other and bundle their own application, tools, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and are thus more lightweight than virtual machines. **Containers are created from "images"** that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories.

https://en.wikipedia.org/wiki/Docker_(software)



Docker is an open platform for developing, shipping, and running applications. Docker enables you to **separate your applications from your infrastructure** so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can **significantly reduce the delay between writing code and running it in production**.

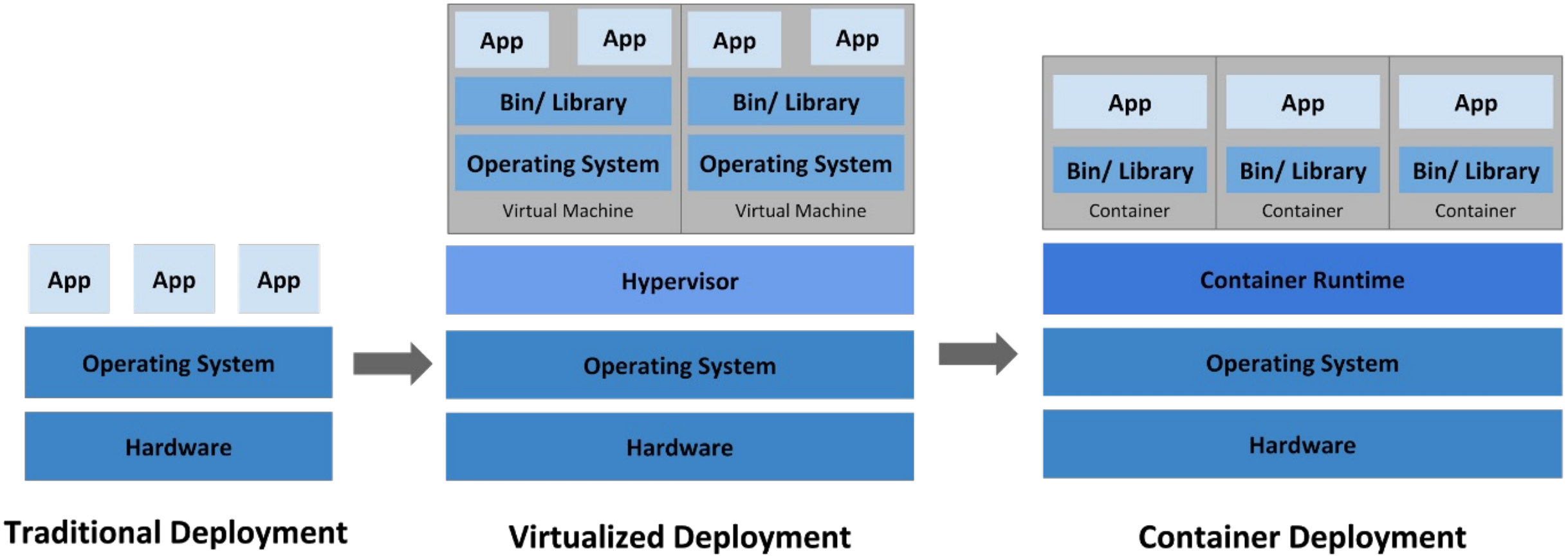
https://docs.docker.com/engine/docker-overview/



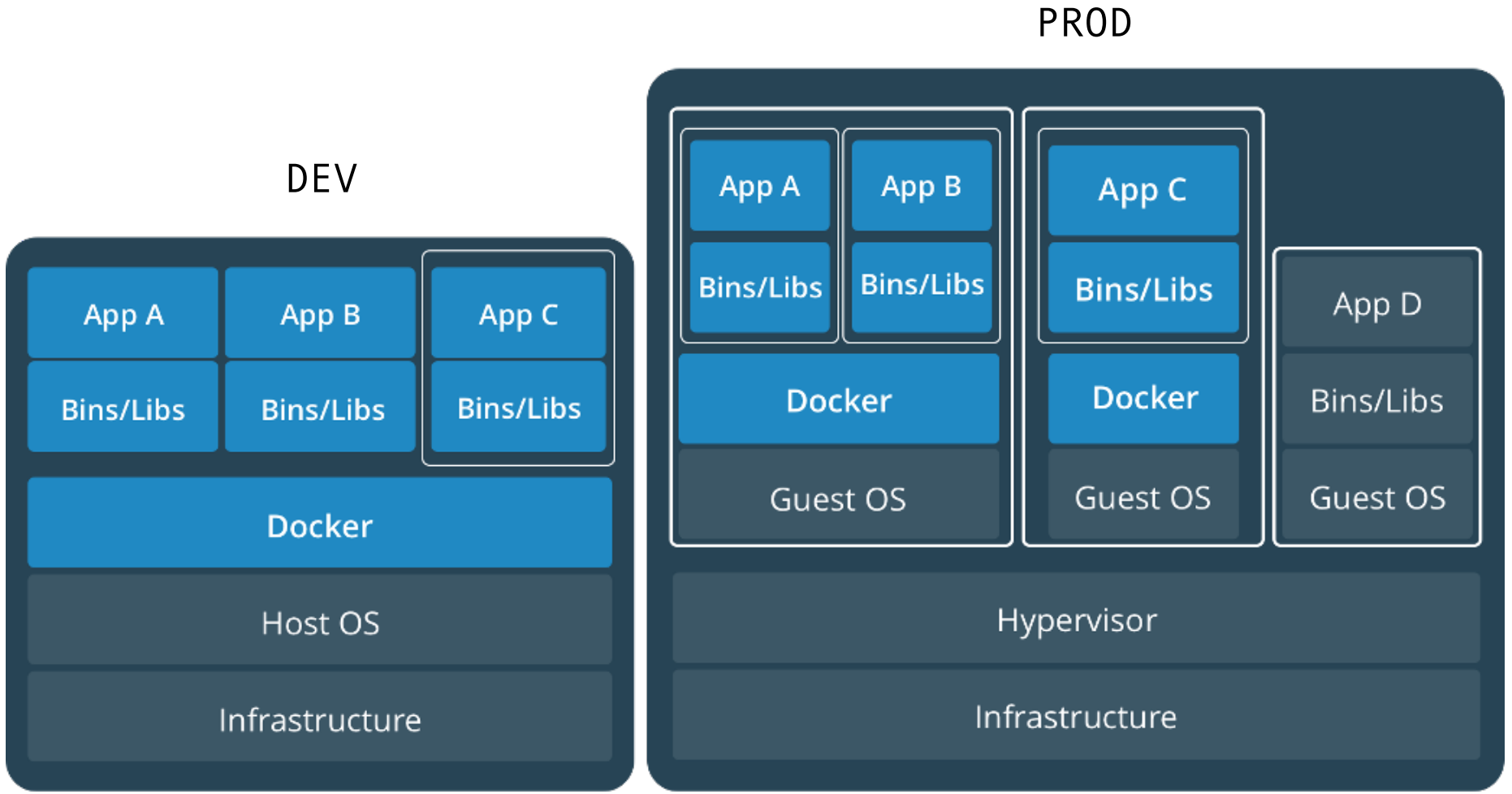
Docker is written in **Go** and takes advantage of **several features of the Linux kernel** to deliver its functionality.

Docker uses a technology called **namespaces** to provide the **isolated workspace** called the container. When you run a container, Docker creates a set of namespaces for that container.

Docker Container



VM and Container together



Docker Basic

Image

The basis of a Docker container. The content at rest.

Container

The image when it is ‘running.’ The standard unit for app service

Engine

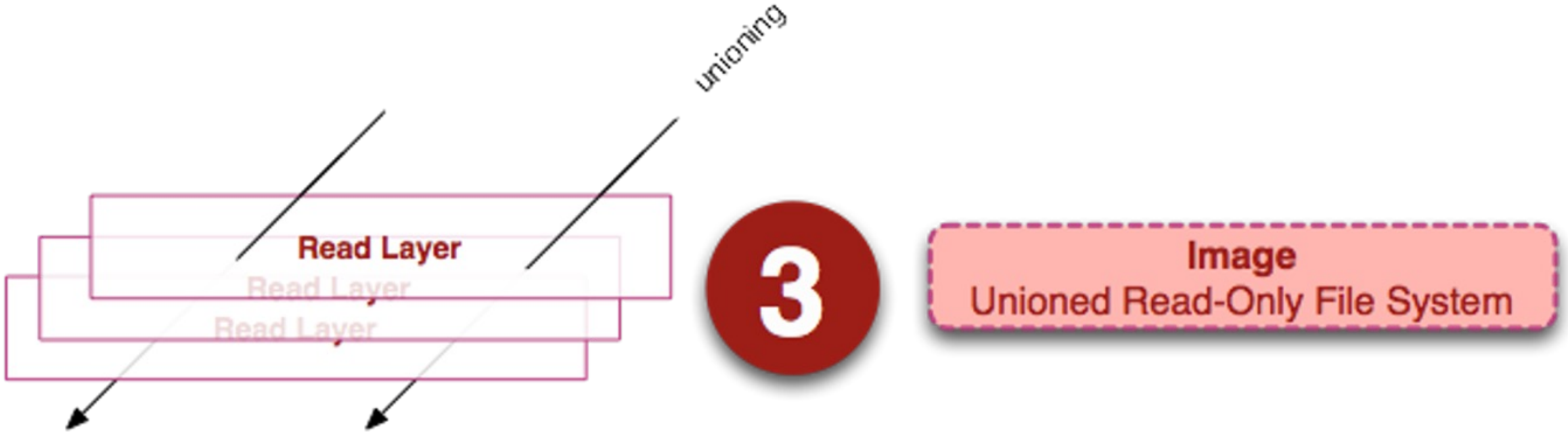
The software that executes commands for containers. Networking and volumes are part of Engine. Can be clustered together.

Registry

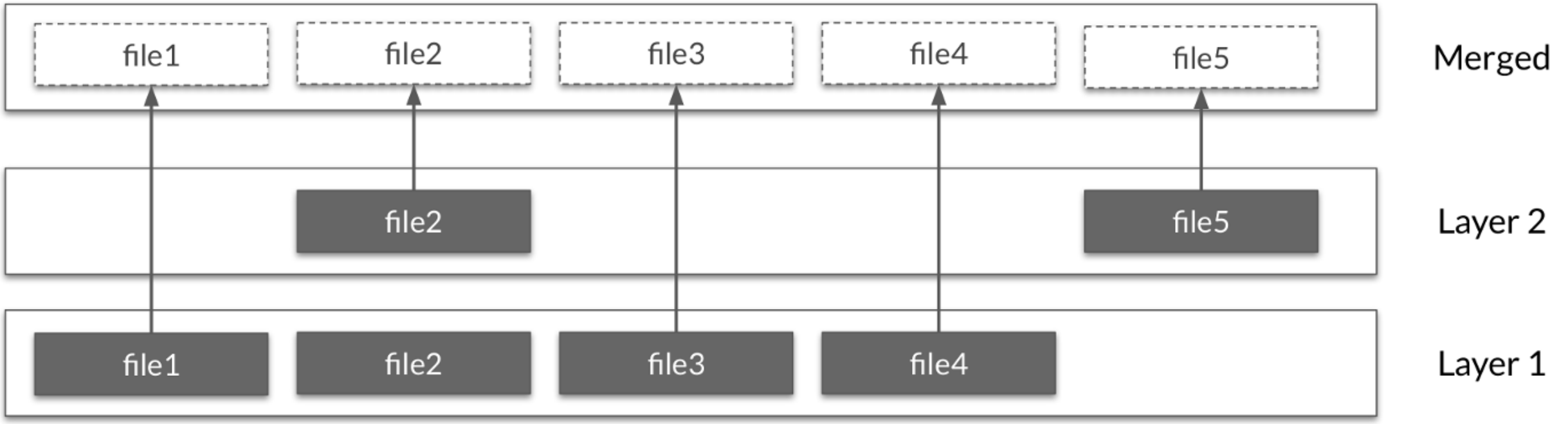
Stores, distributes and manages Docker images

Opsta Education

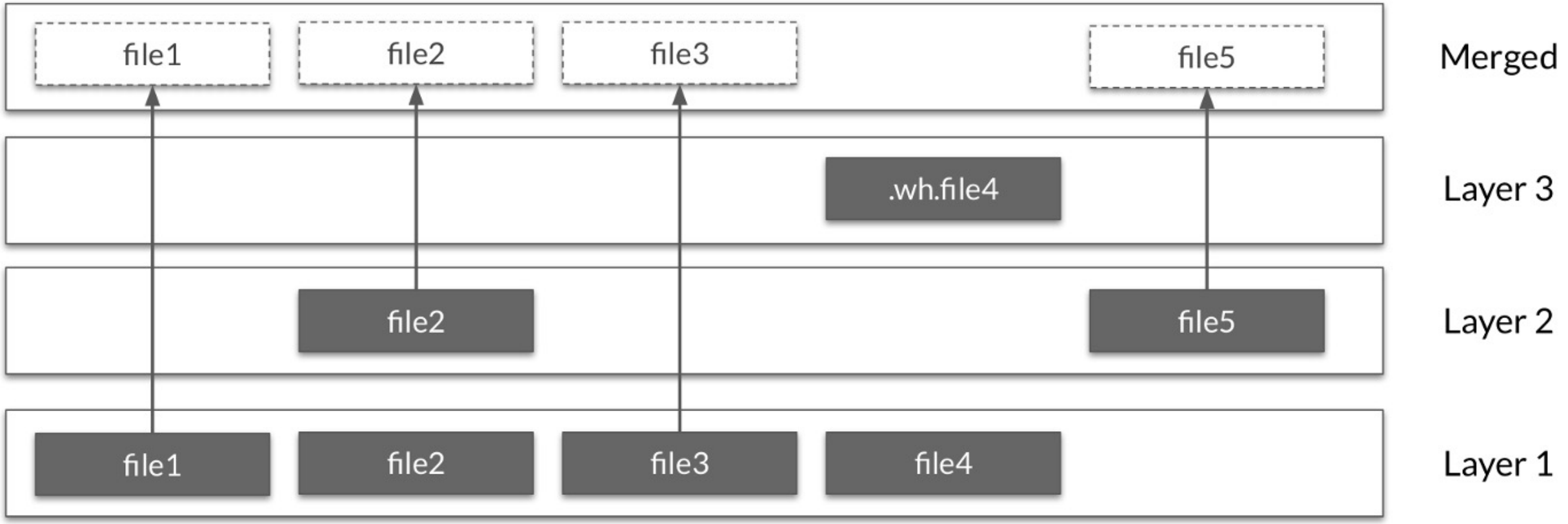
Docker Image



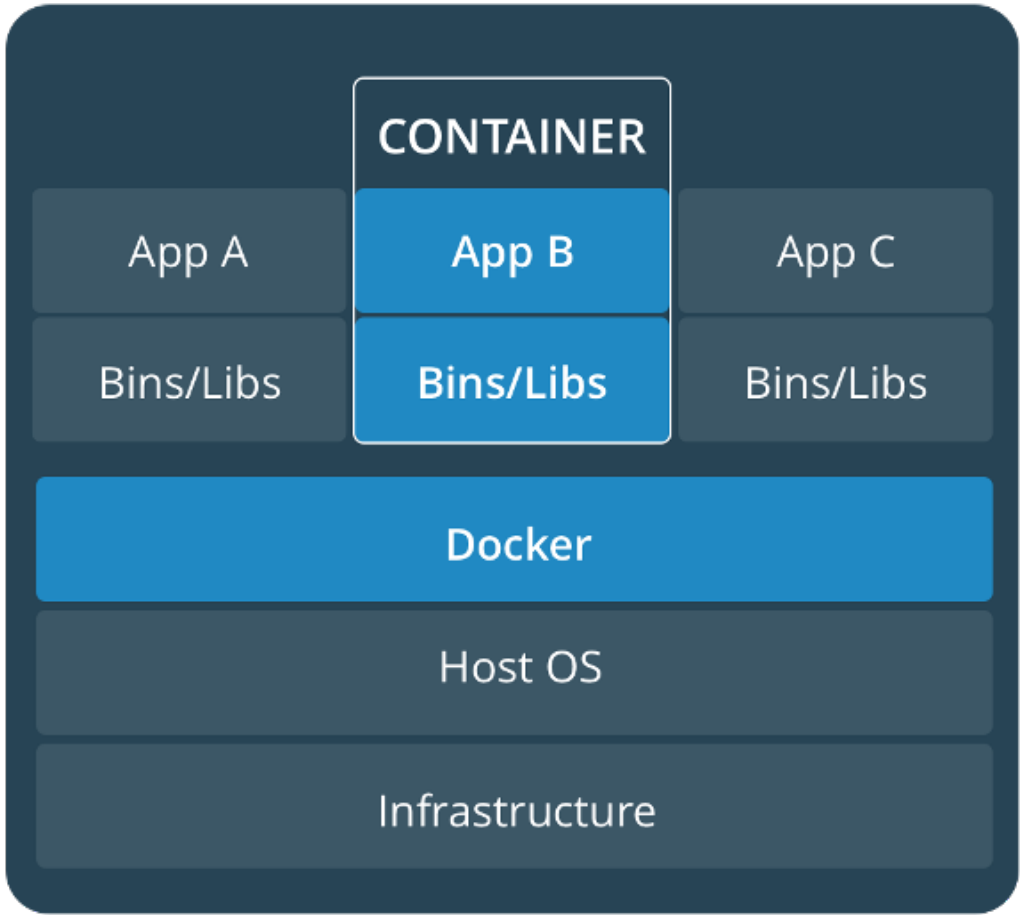
Docker Image Layer (1)



Docker Image Layer (2)

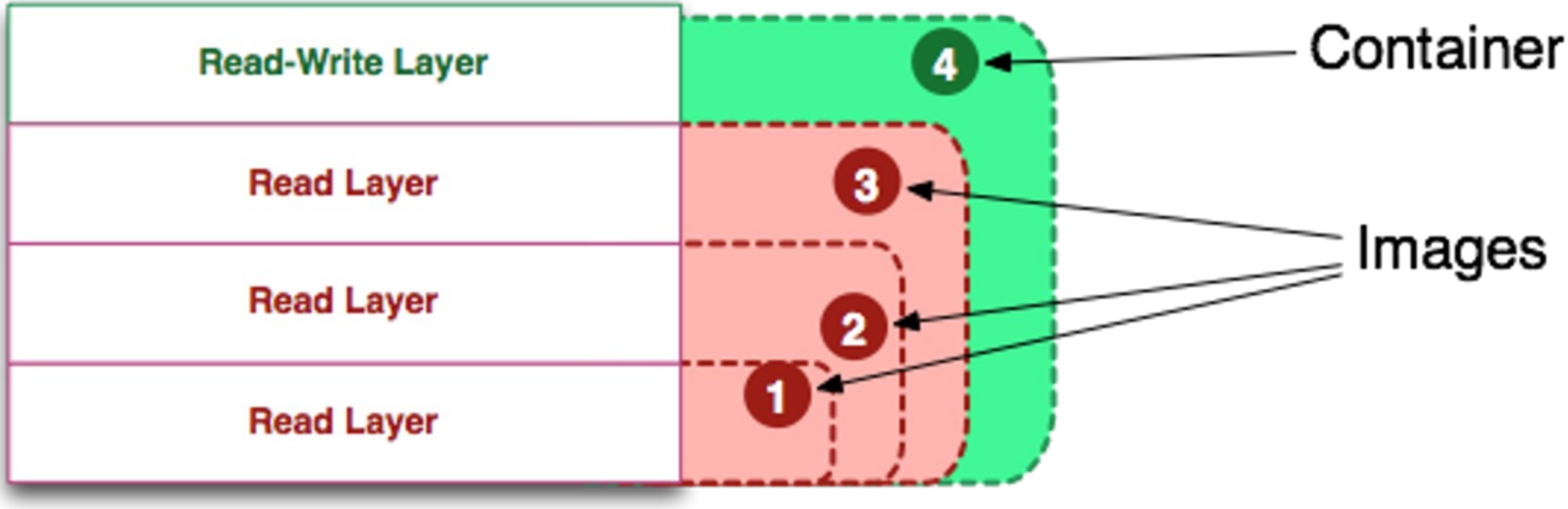


What is in Docker Image

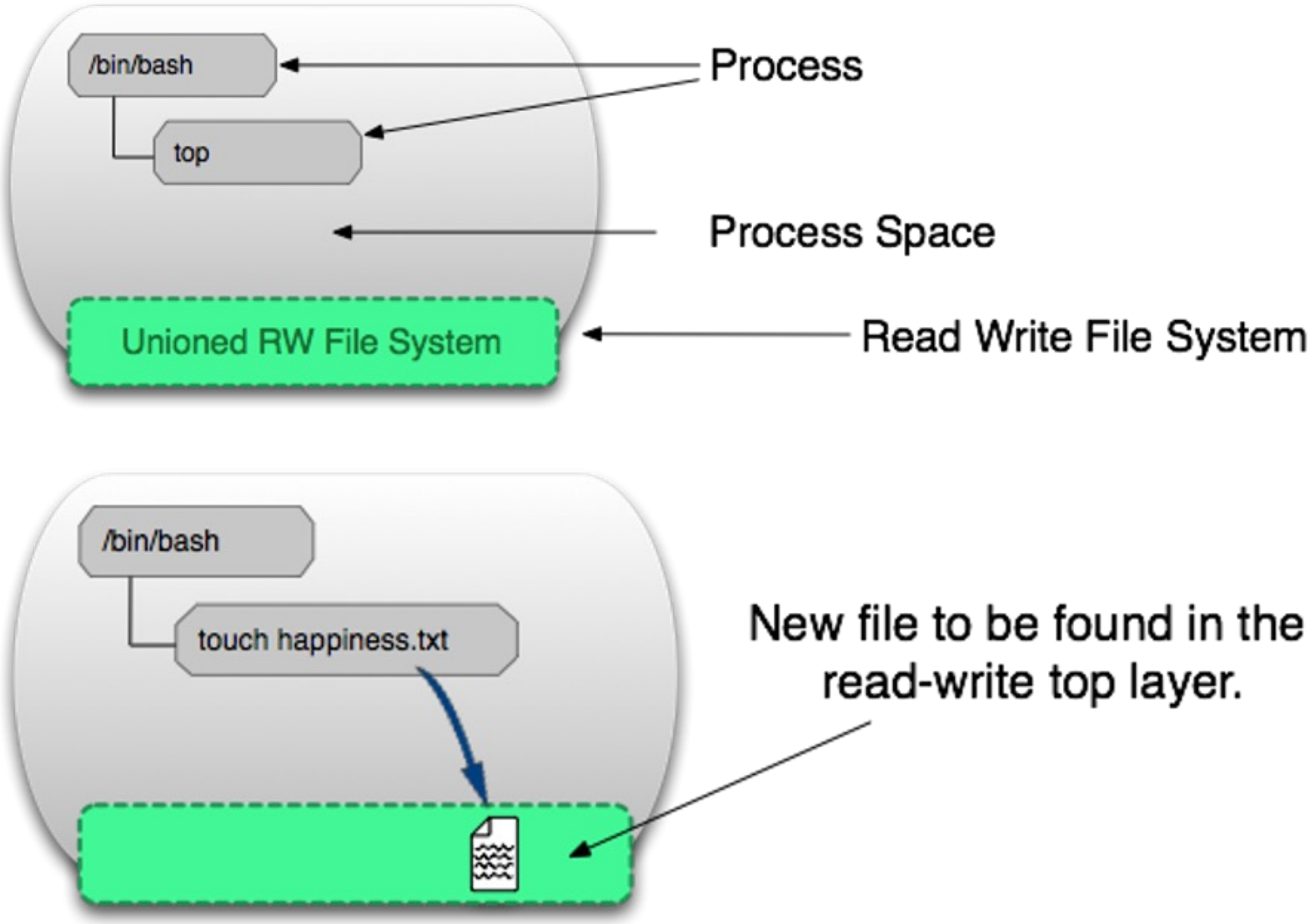


- **Base OS**
  (Ubuntu, CentOS, Alpine)
- **System Packages**
  (APT, YUM)
- **Library Dependencies**
  (PIP, NPM, Composer)
- **Configuration**
- **Source Code**

Docker Container



Running Container



Docker Image vs Container

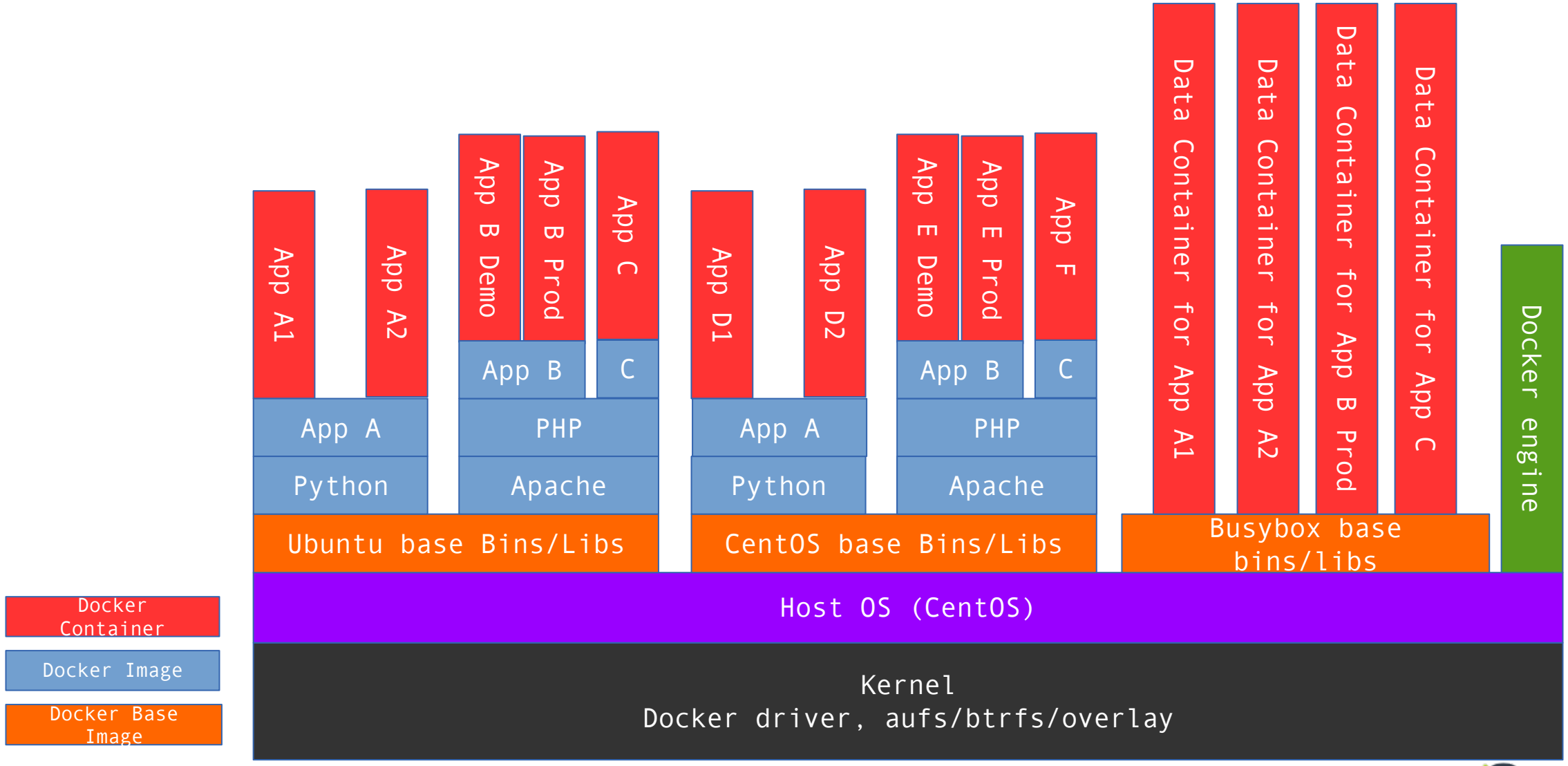
Docker Image is like a program

Docker Container is like a process

Docker Image is like a class

Docker Container is like a instance of class

Docker Layer



Dockerfile

- Dockerfile is instructions to build Docker Image
  - How to run commands
  - Add files or directories
  - Create environment variables
  - What process to run when launching container
- Result from building Dockerfile is Docker Image

Sample Dockerfile


```
FROM node:14.14.0-alpine3.12
COPY . /nodejs/.
WORKDIR /nodejs
RUN npm install
ENV VERSION 1.0
EXPOSE 8081
CMD ["node", "/nodejs/main.js"]
```

OS + System Packages

Source Code

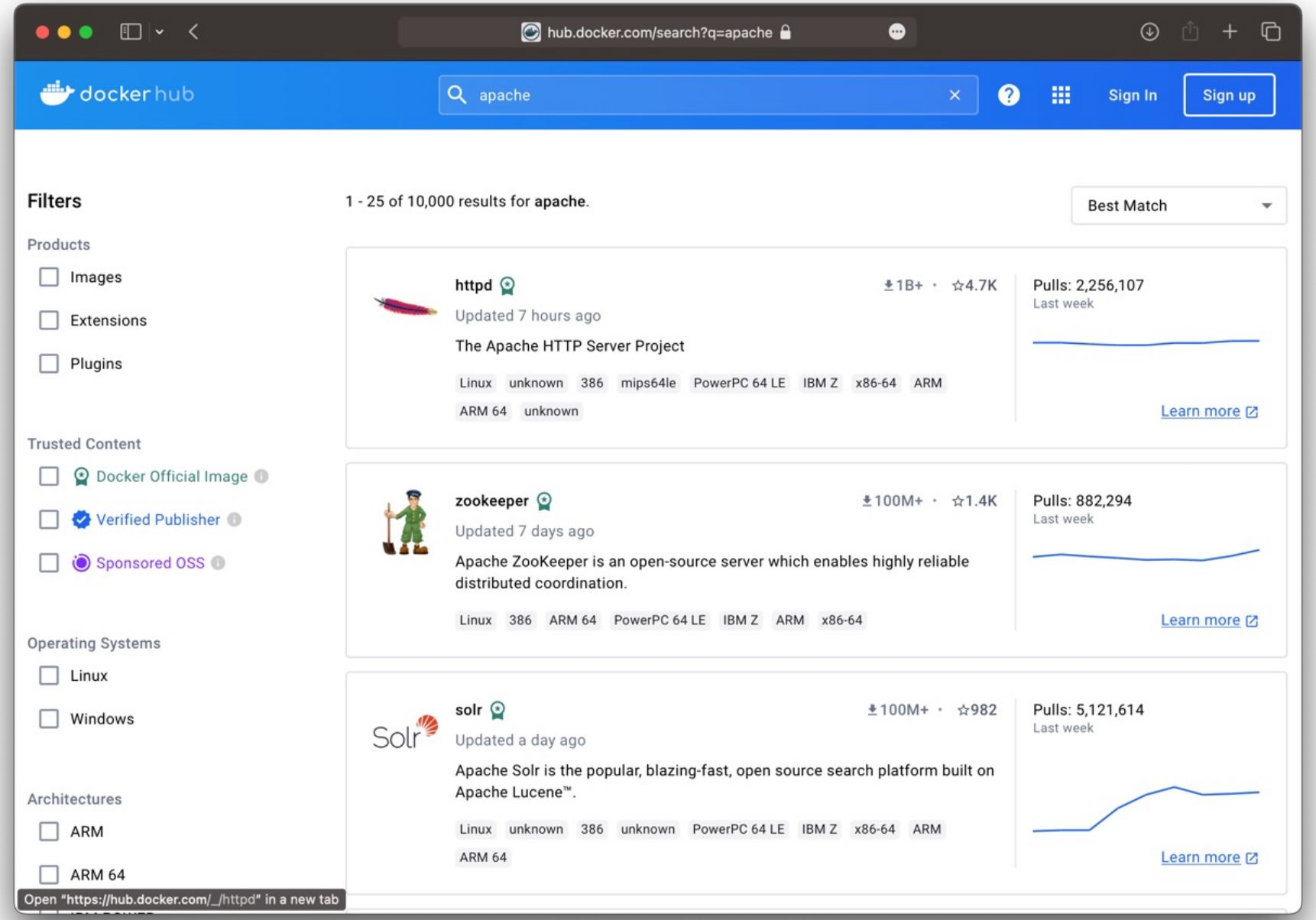
Library Dependencies

Configuration

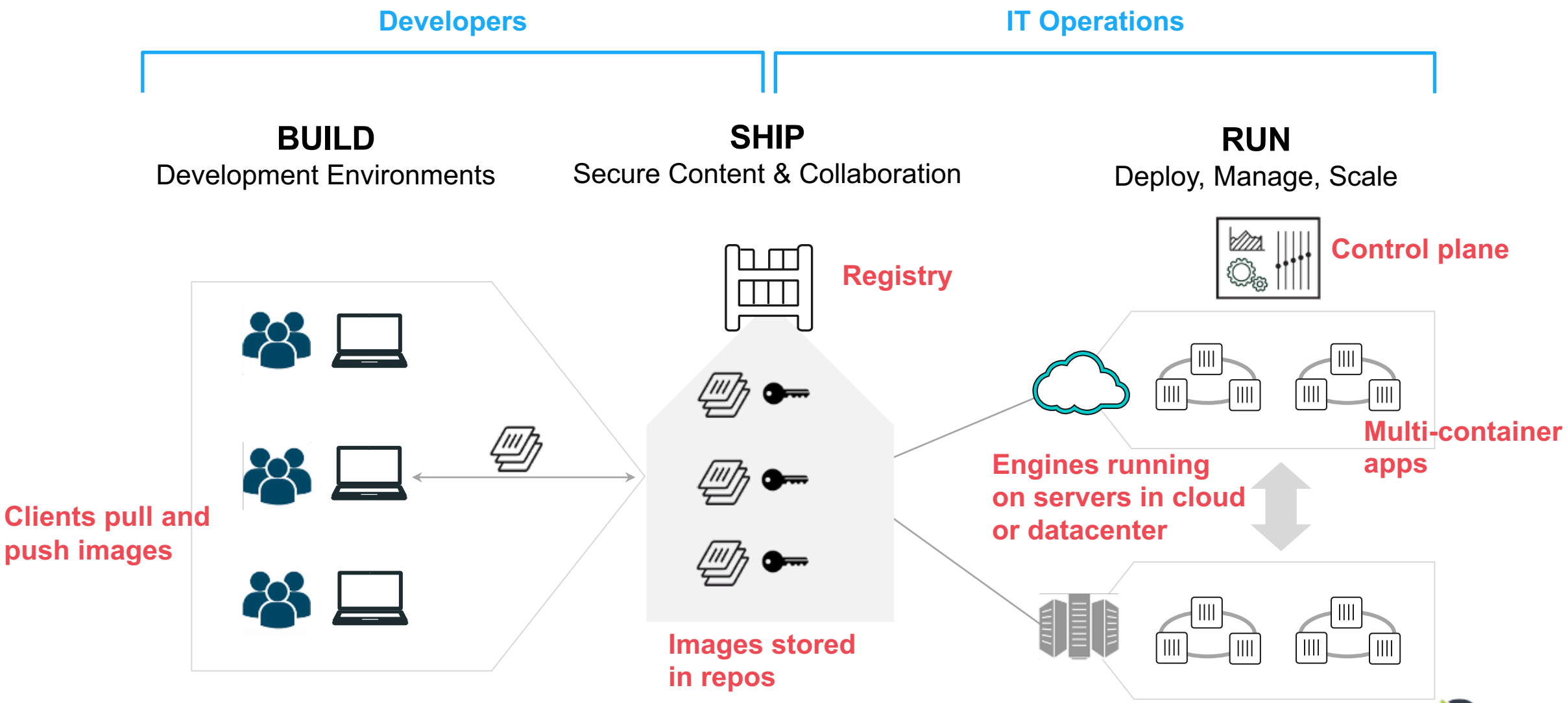
Docker Registry

- Docker Registry is the store for Docker Image
- Docker Hub is public Docker Registry like Github
- Using Docker client to push and pull Docker Image from Docker Registry
- You can create your own Private Docker Registry

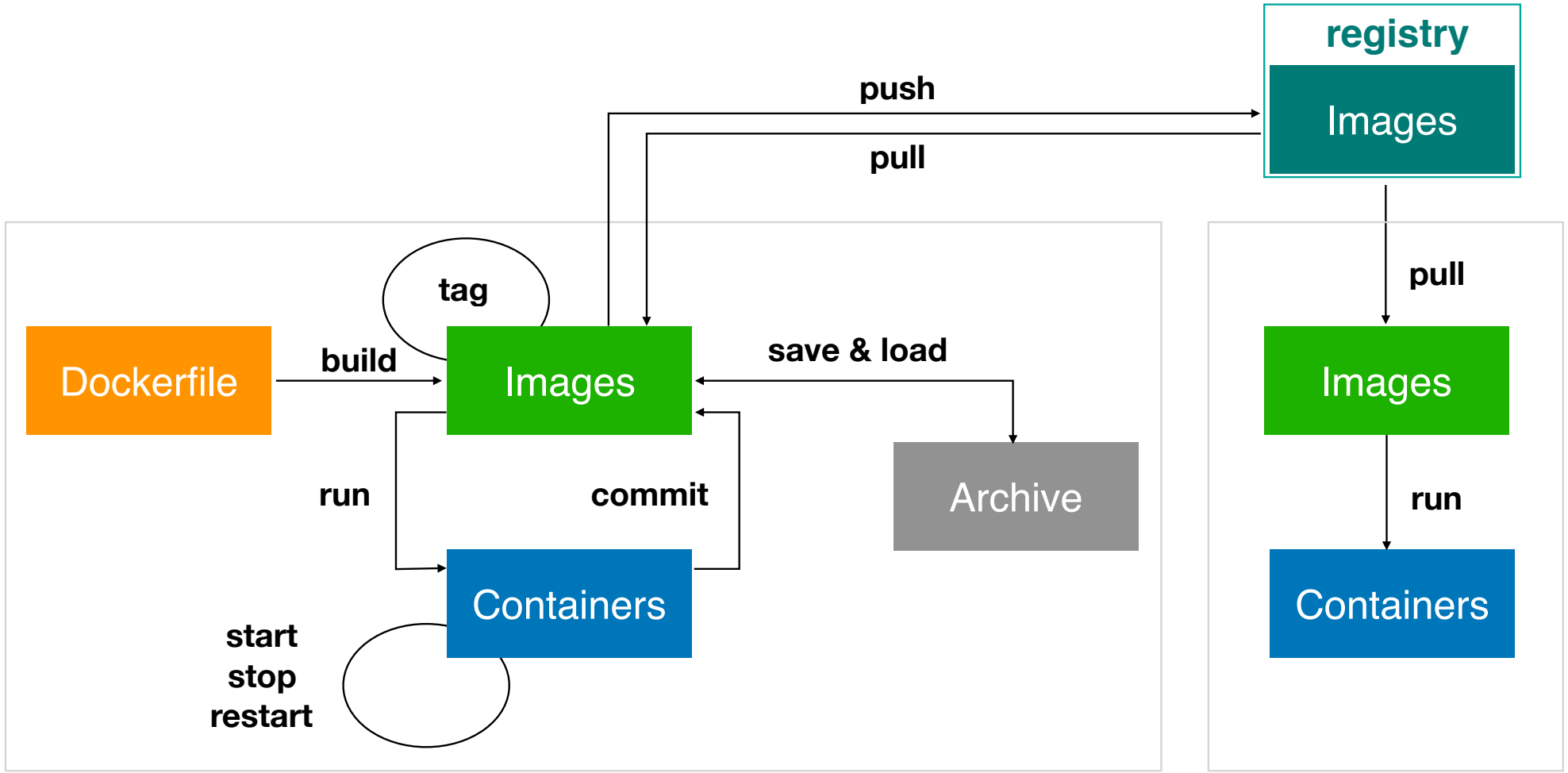
Docker Hub


hub.docker.com/search?q=apache
dockerhub
apache
Sign In
Sign up
Filters
1 - 25 of 10,000 results for apache.
Best Match
Products
Images
Extensions
Plugins
Trusted Content
Docker Official Image
Verified Publisher
Sponsored OSS
Operating Systems
Linux
Windows
Architectures
ARM
ARM 64
httpd
1B+ · ☆4.7K
Updated 7 hours ago
The Apache HTTP Server Project
Linux unknown 386 mips64le PowerPC 64 LE IBM Z x86-64 ARM
ARM 64 unknown
Pulls: 2,256,107
Last week
Learn more
zookeeper
100M+ · ☆1.4K
Updated 7 days ago
Apache ZooKeeper is an open-source server which enables highly reliable distributed coordination.
Linux 386 ARM 64 PowerPC 64 LE IBM Z ARM x86-64
Pulls: 882,294
Last week
Learn more
solr
100M+ · ☆982
Updated a day ago
Apache Solr is the popular, blazing-fast, open source search platform built on Apache Lucene™.
Linux unknown 386 unknown PowerPC 64 LE IBM Z x86-64 ARM
ARM 64
Pulls: 5,121,614
Last week
Learn more
Open "https://hub.docker.com/_/httpd" in a new tab

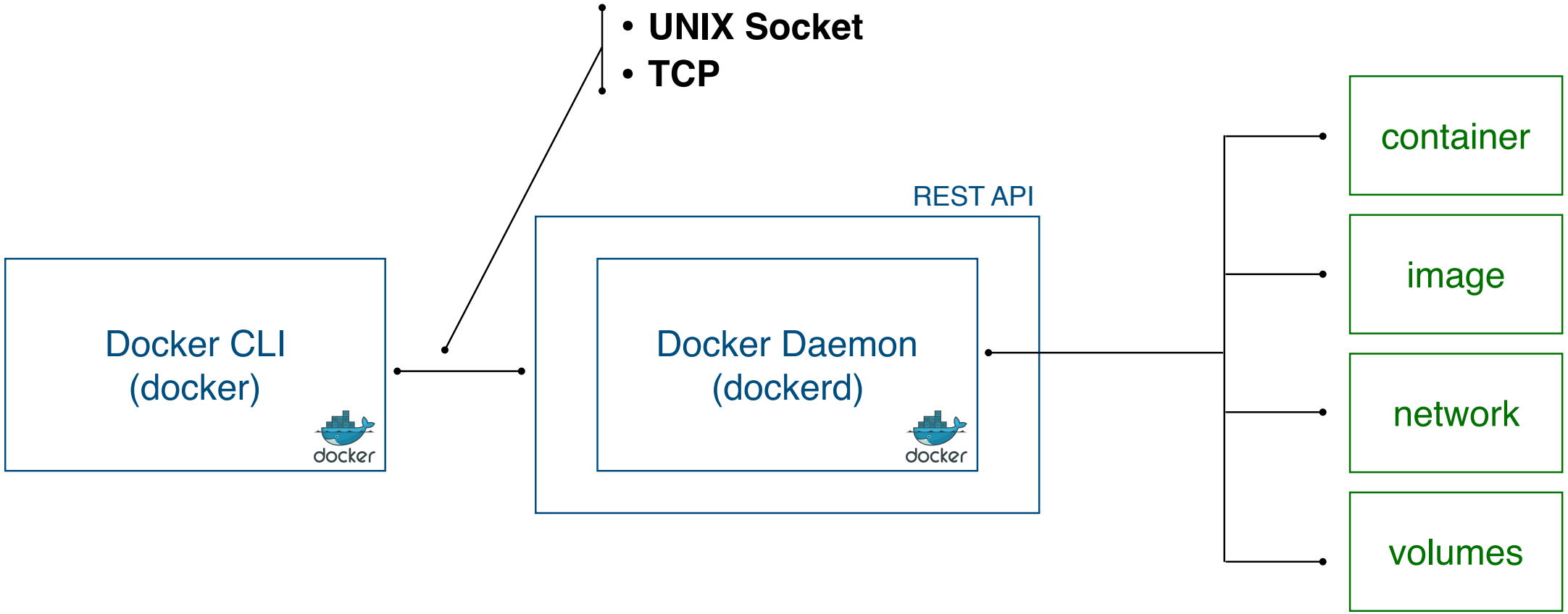
Docker and DevOps



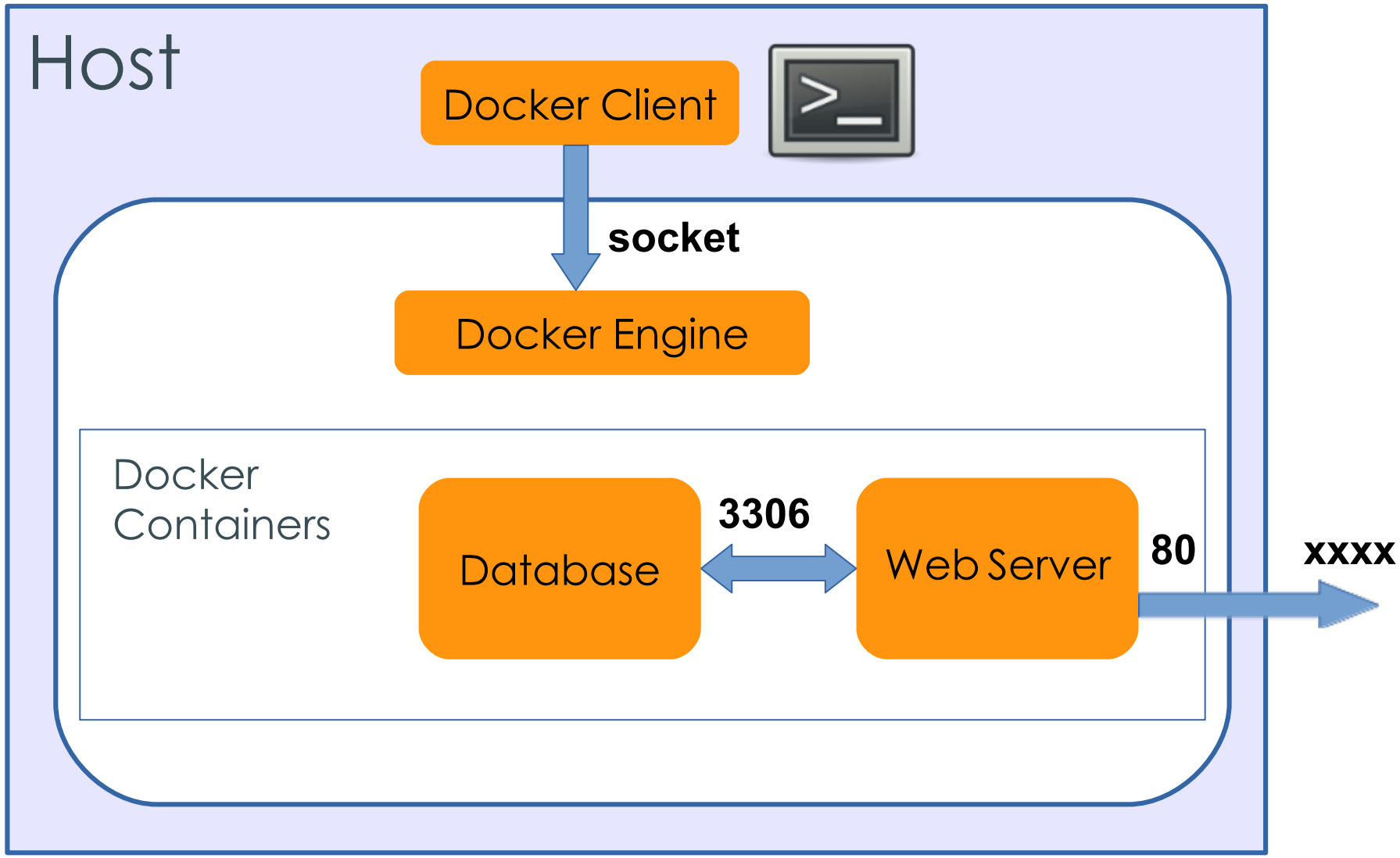
Containers and Images Lifecycle



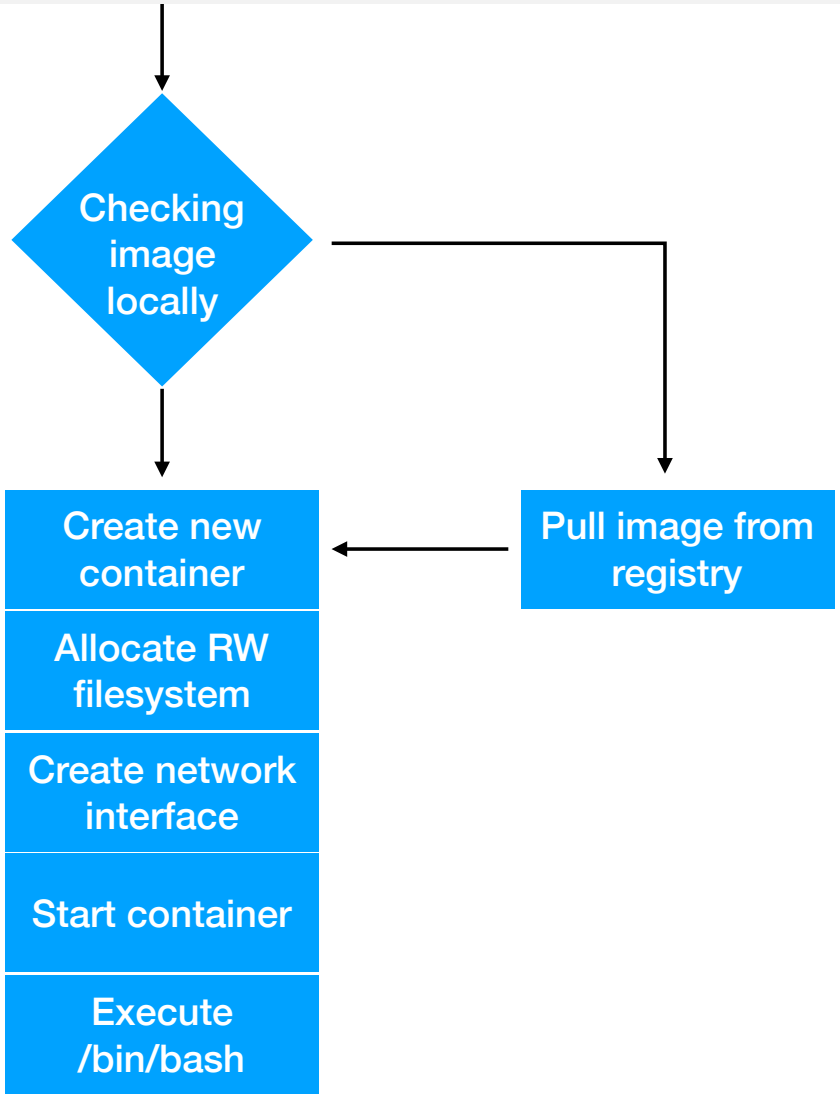
Docker Architecture



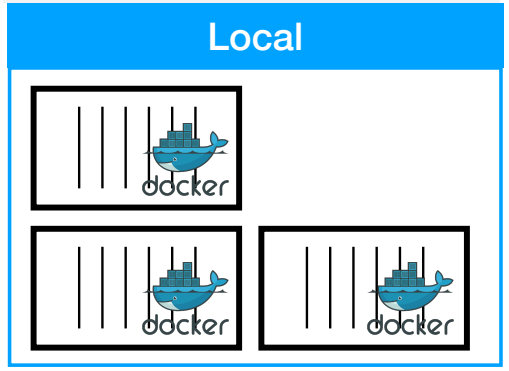
Docker Architecture

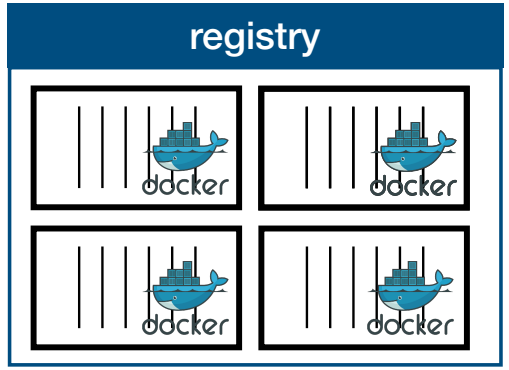


```
$ docker run -it ubuntu /bin/bash
```




```
$ docker images
```





https://docs.docker.com/get-started/overview/

Install Docker on Ubuntu

- Refer to https://docs.docker.com/engine/install/ubuntu/ for instructions
  - install the latest version using apt
- Perform post-installation https://docs.docker.com/engine/install/linux-postinstall/
  - add `sysadmin` to `docker` group

```
sudo usermod -aG docker sysadmin
```

  - configure 'local' as logging driver and add mirror

/etc/docker/daemon.json

```
{
  "log-driver": "local",
  "log-opts": {
     "max-size": "10m"
  },
  "registry-mirrors": ["https://lvmolarn.sit.kmutt.ac.th"]
}
```

Docker Help


```
# To show Docker online help
docker

# To show Docker run command help
# Note the alias is ‘docker container run’
docker run --help

# Show docker disk usage help
docker system df --help
```

Docker Image


```
# To show Docker Image on your machine
docker images

# To pull ubuntu image with tag latest from Docker Hub
docker pull ubuntu

# To show your newly pull ubuntu image on your machine
docker images

# Pull Ubuntu 20.04
docker pull ubuntu:20.04

# To show Docker Image on your machine
docker images
```

Docker Container


```
# Run first ubuntu container
docker run ubuntu echo "Hello World"

# Run container with bash command
docker run -i -t ubuntu bash

# Below command are run inside container
whoami
hostname
cat /etc/*release*
exit
```

Docker Container Basic Operation


```
# Show running containers
docker ps

# Show running and stopped containers
docker ps -a

# Run container with specify name
docker run --name ubuntu-universe ubuntu echo "Hello Universe"
docker ps -a

# Delete container by name
docker rm ubuntu-universe
docker ps -a

# Delete container by part of container id
docker rm 07f
docker ps -a
```

Run Docker as daemon and expose port


```
# Run Nginx
docker run nginx:alpine
docker ps -a

# Run Nginx in background
docker run -d nginx:alpine
docker ps

# Export 8080 port from outside forward to port 80 on container
docker run -d -p 8080:80 nginx:alpine

# What happen if try to expose same port again
docker run -d -p 8080:80 nginx:alpine

# What happen if we expose difference outside port
docker run -d -p 8888:80 nginx:alpine

# You can try view the web again
docker ps
```

Docker Exercise

- Try to run Apache Server 2.4.33 on Alpine Linux and expose to port 8083

Docker Utilities Commands


```
# Rename container name
docker rename vigorous_sammet nginx
# To go inside running container
docker exec -it nginx sh
ls -l
ps -ef
exit
# Show container processes
docker top nginx
# Show logs
docker logs nginx
# Follow logs
docker logs nginx -f
# Try Web preview to see log running


# Show container resource consumes
docker stats
# Show container all metadatas
docker inspect nginx
```