

docker

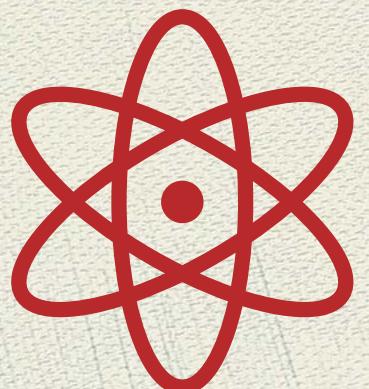
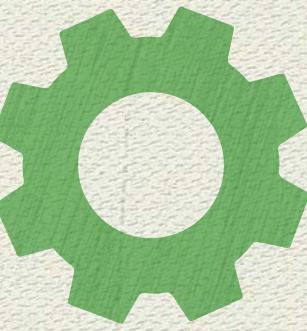
Quality Excellence
Ekapol W.

Key take away



Get to know Docker Anatomy

How to create Docker Image and Container



Build up Container Network

Daily Life Scenarios



It is working on my machine.

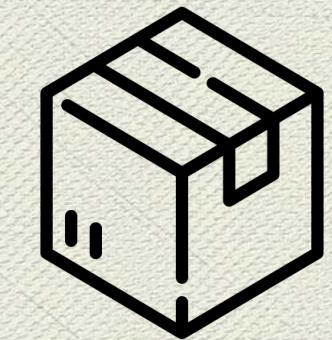
Develop, SIT, UAT and Production are not the same.

Software is broken every time change environment.

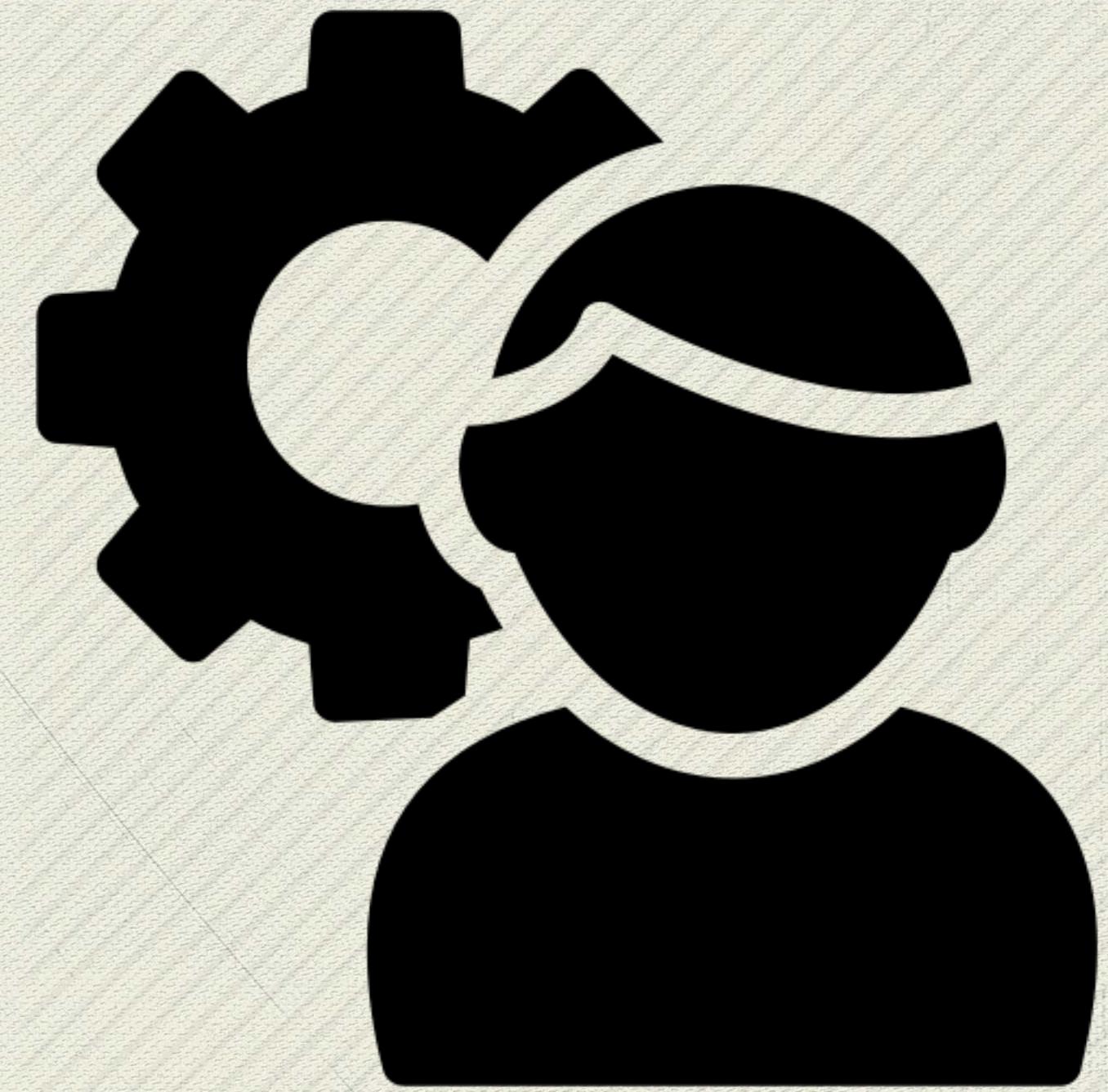
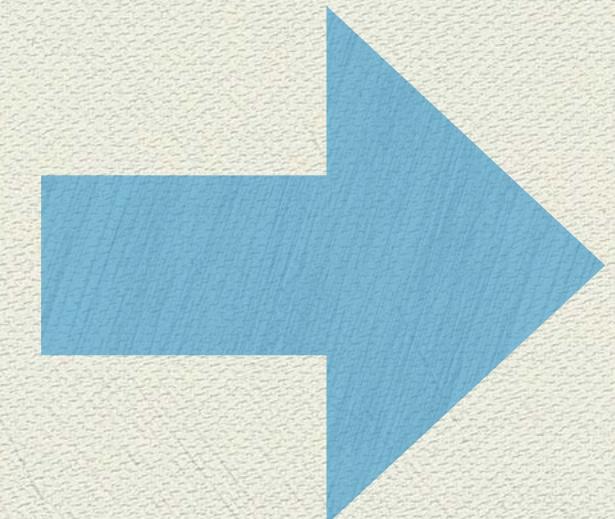
Traditional Software Transition



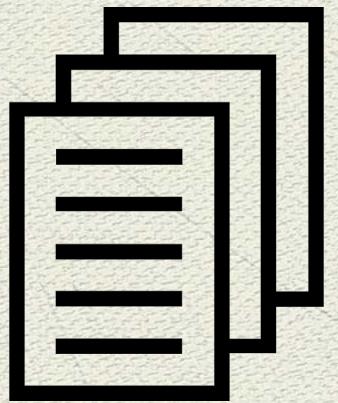
Software .ear .war .jar



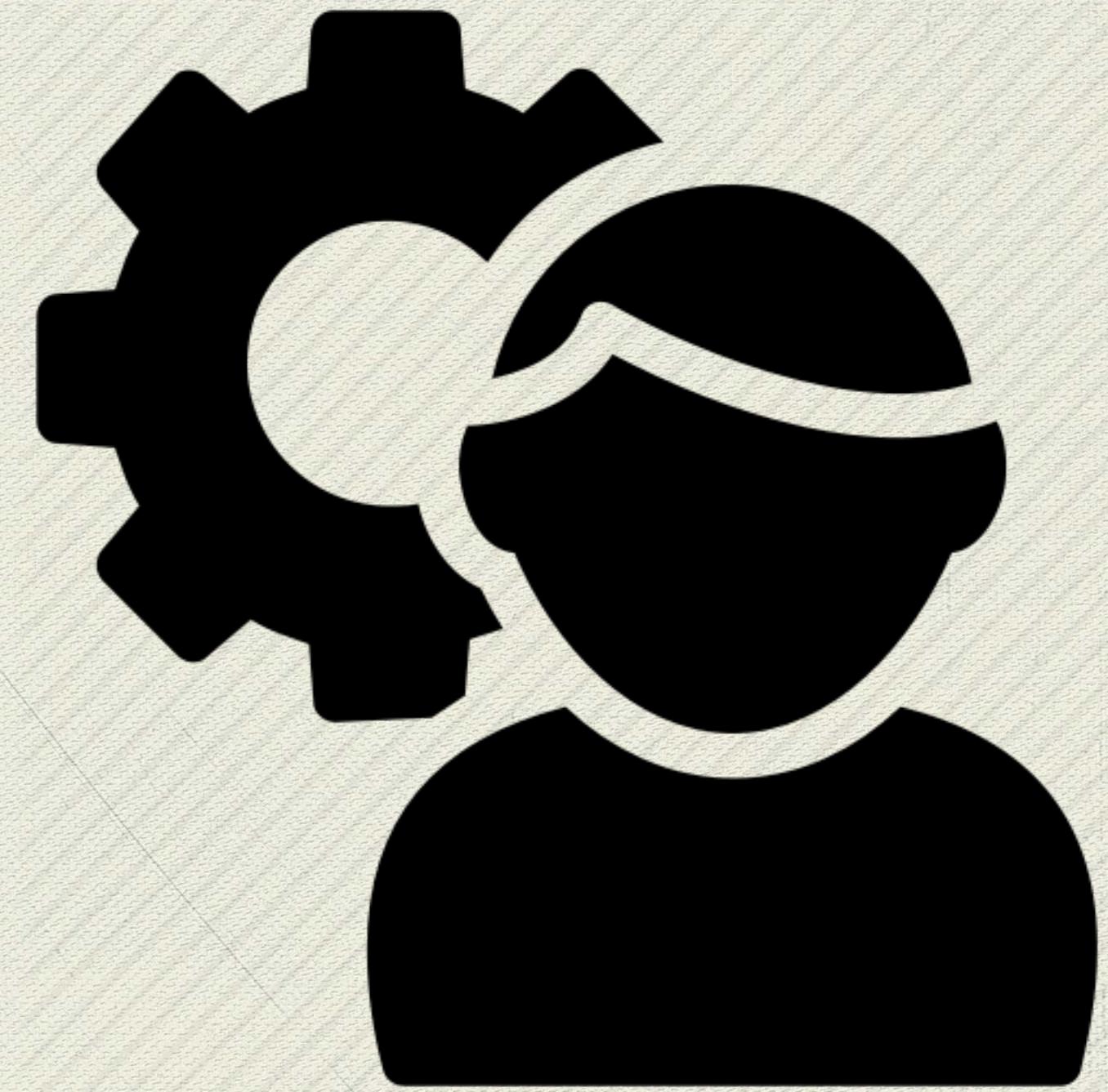
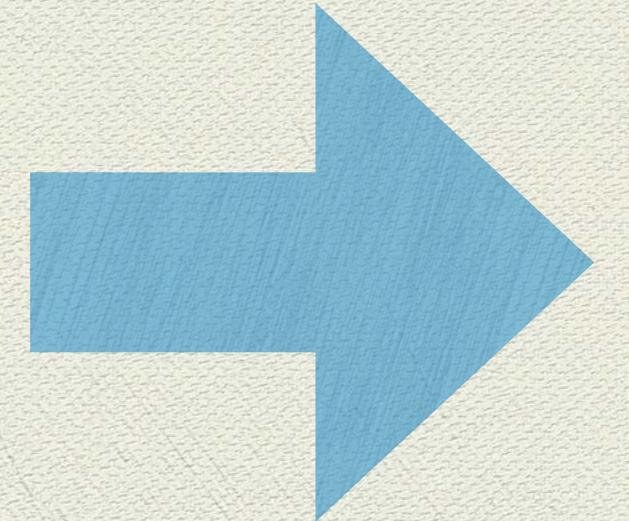
Installation Instruction



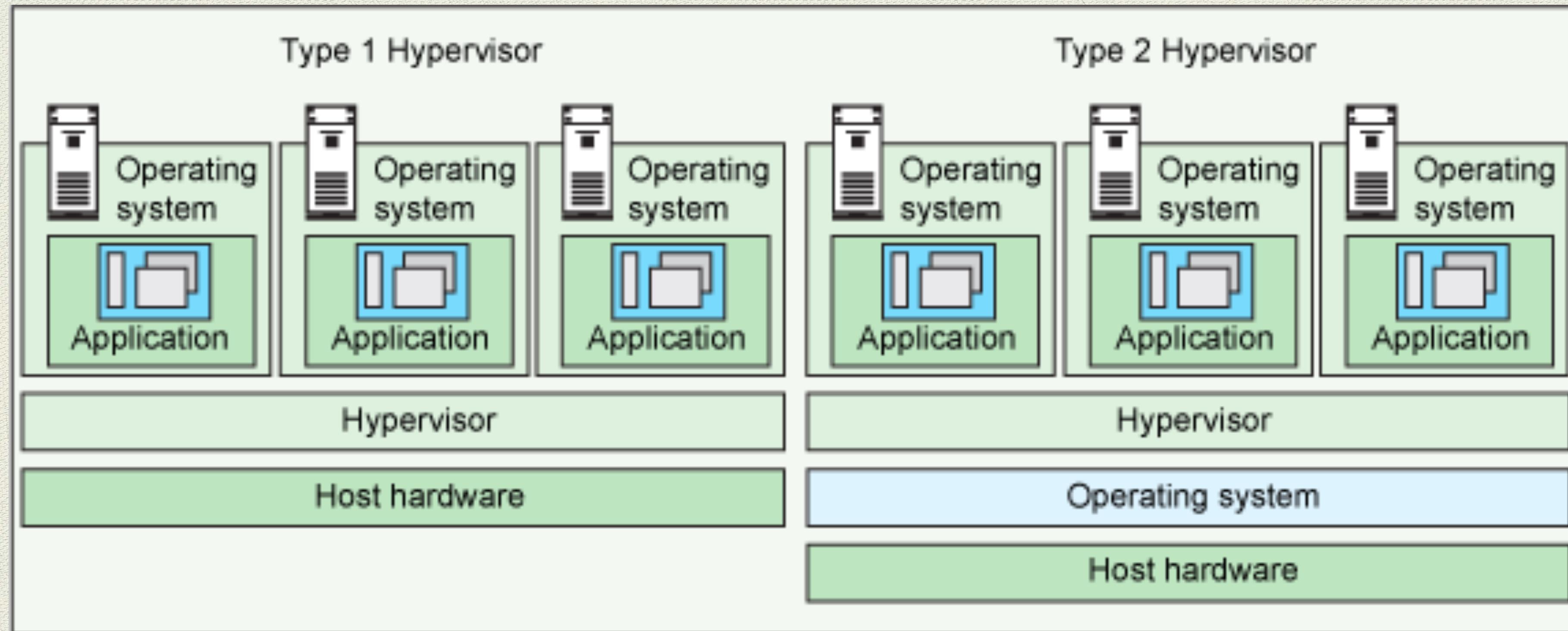
Modern Software Transition



Images



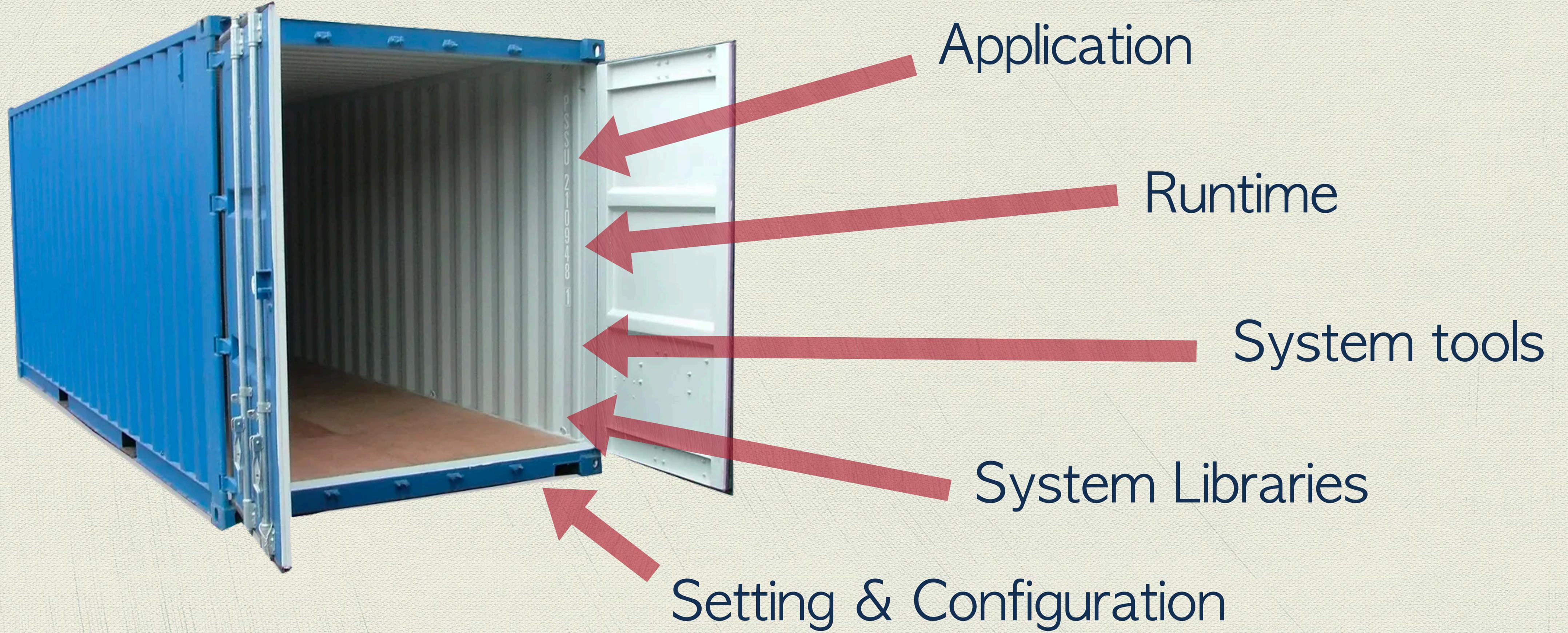
Virtual Machine



Type 1 Hypervisors: *Xen, VMware ESXi, and Microsoft Hyper-*

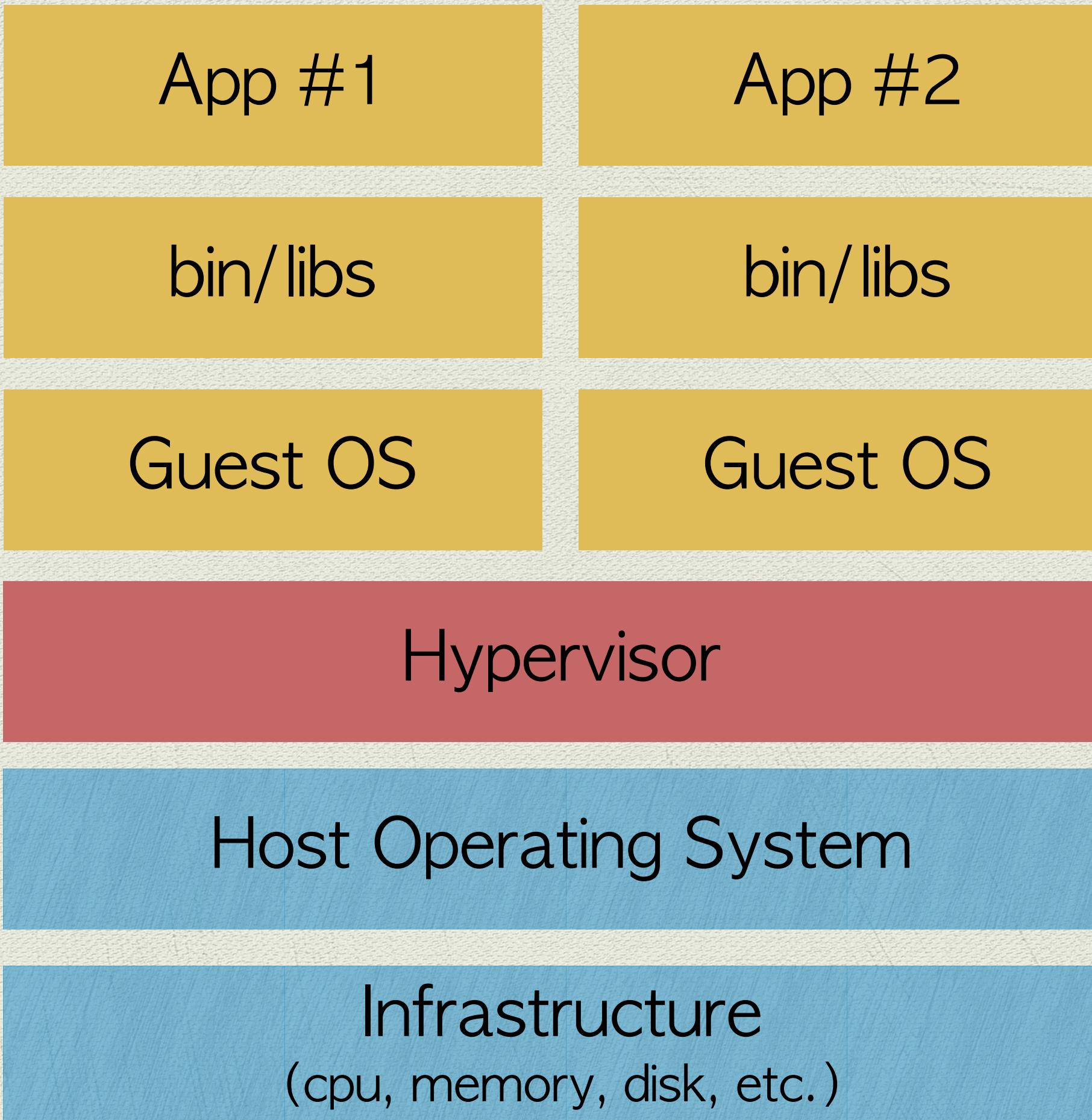
Type 2 Hypervisors: *VMware Workstation, VMware Player, Oracle VirtualBox*

What is Container ?

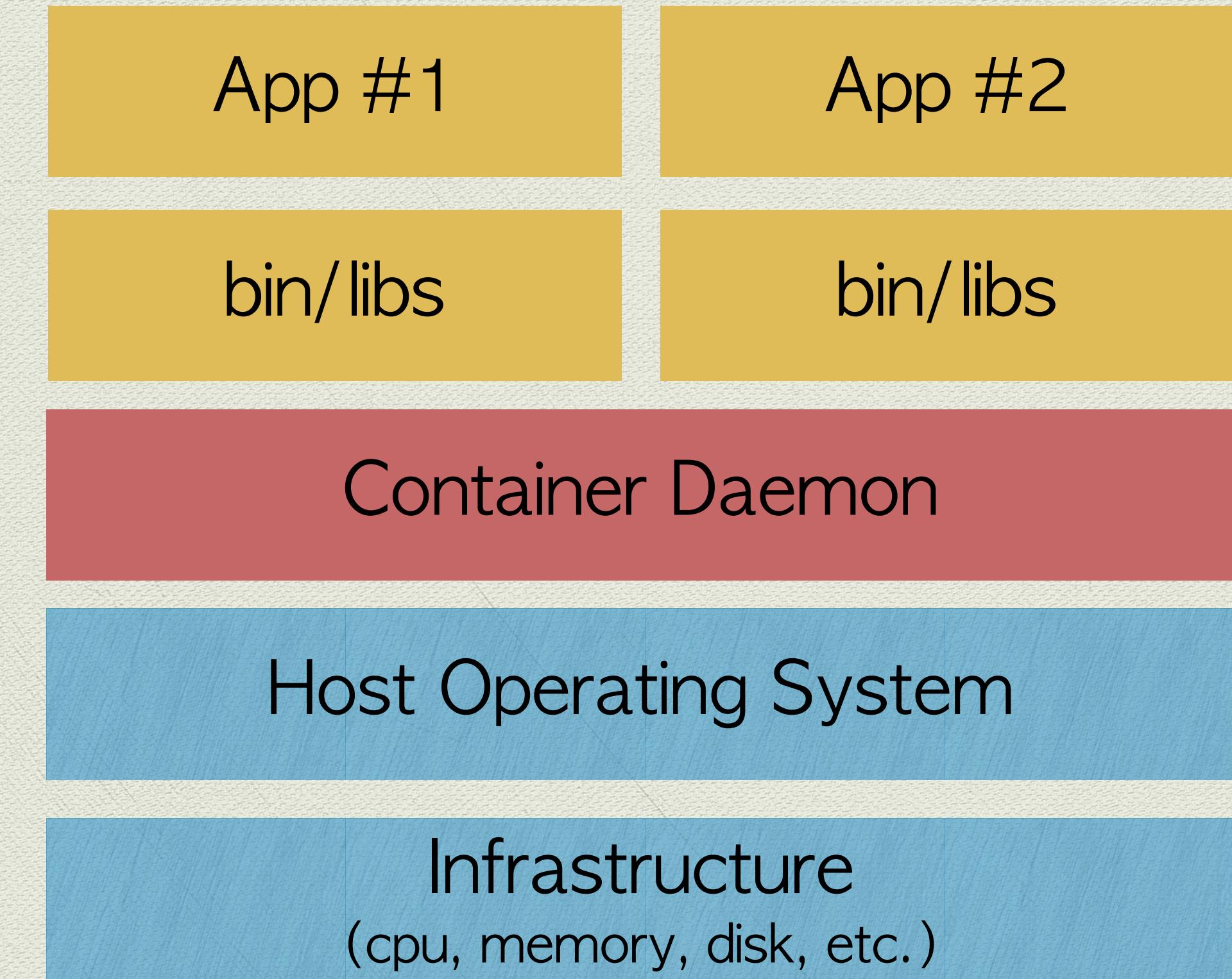


Package Software into Standardized Units for Development, Shipment and Deployment

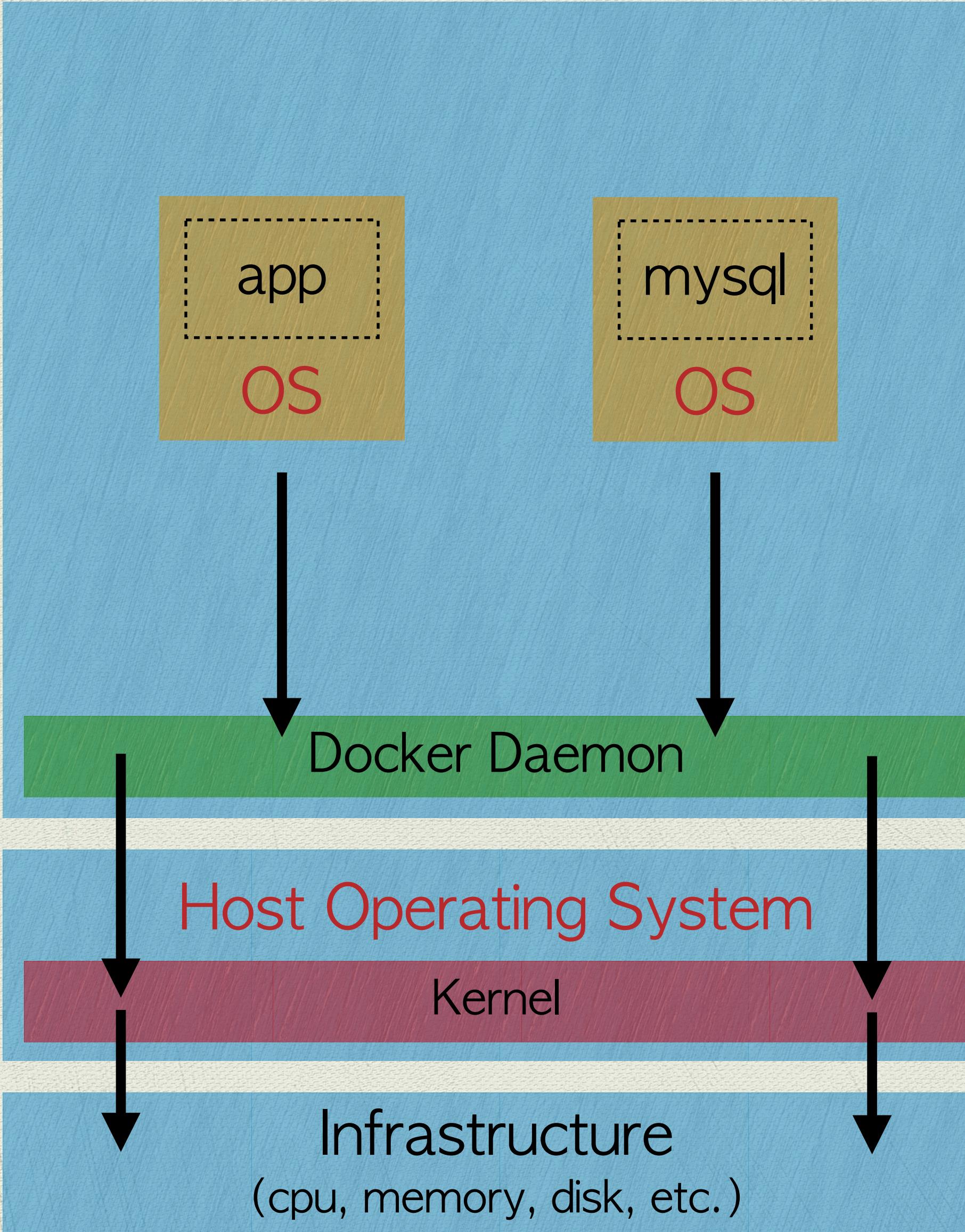
Virtual Machines



Containers



Containers virtualise the operating system instead of hardware.

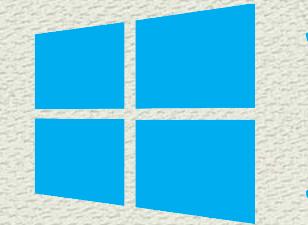


Every OS must use the same Kernel.



Linux

- Debian / Ubuntu
- Redhat
- Alpine

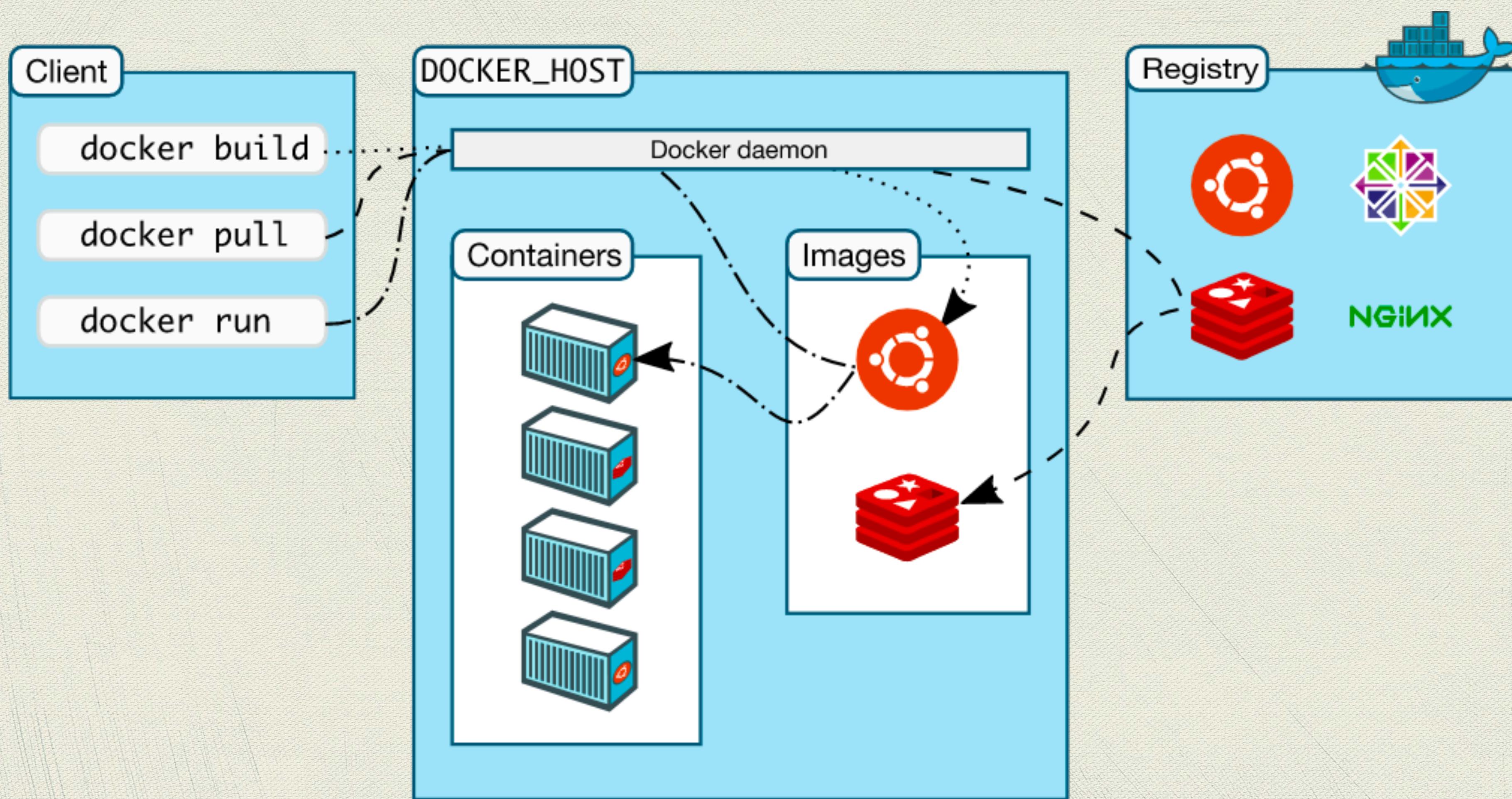


Windows Server

- Microsoft Server Core
- Microsoft Nano Server

How about Linux container on Windows ?

Docker Architecture



Installing Docker

Windows 10 pro & macOS 10.13 up

<https://www.docker.com/products/docker-desktop>



Windows 10 home, 8, 8.1, 7 & old mac

windows
https://docs.docker.com/toolbox/toolbox_install_windows/

Mac
https://docs.docker.com/toolbox/toolbox_install_mac/



```
Ekapols-MBP:~ ekapolwongnapapan$ docker --version
Docker version 19.03.2, build 6a30dfc
Ekapols-MBP:~ ekapolwongnapapan$ █
```

Validate Docker Installation

```
[Ekapol-MBP:~ ekapolwongnapapan$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:4df8ca8a7e309c256d60d7971ea14c27672fc0d10c5f303856d7bc48f8cc17ff
Status: Downloaded newer image for hello-world:latest
```

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

To generate this message, Docker took the following steps:

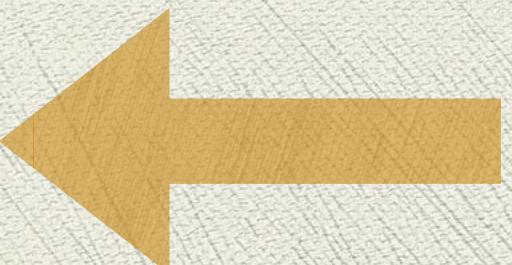
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that 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 ID:
<https://hub.docker.com/>

For more examples and ideas, visit:
<https://docs.docker.com/get-started/>

```
[Ekapol-MBP:~ ekapolwongnapapan$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world    latest    fce289e99eb9   10 months ago  1.84kB
Ekapol-MBP:~ ekapolwongnapapan$
```

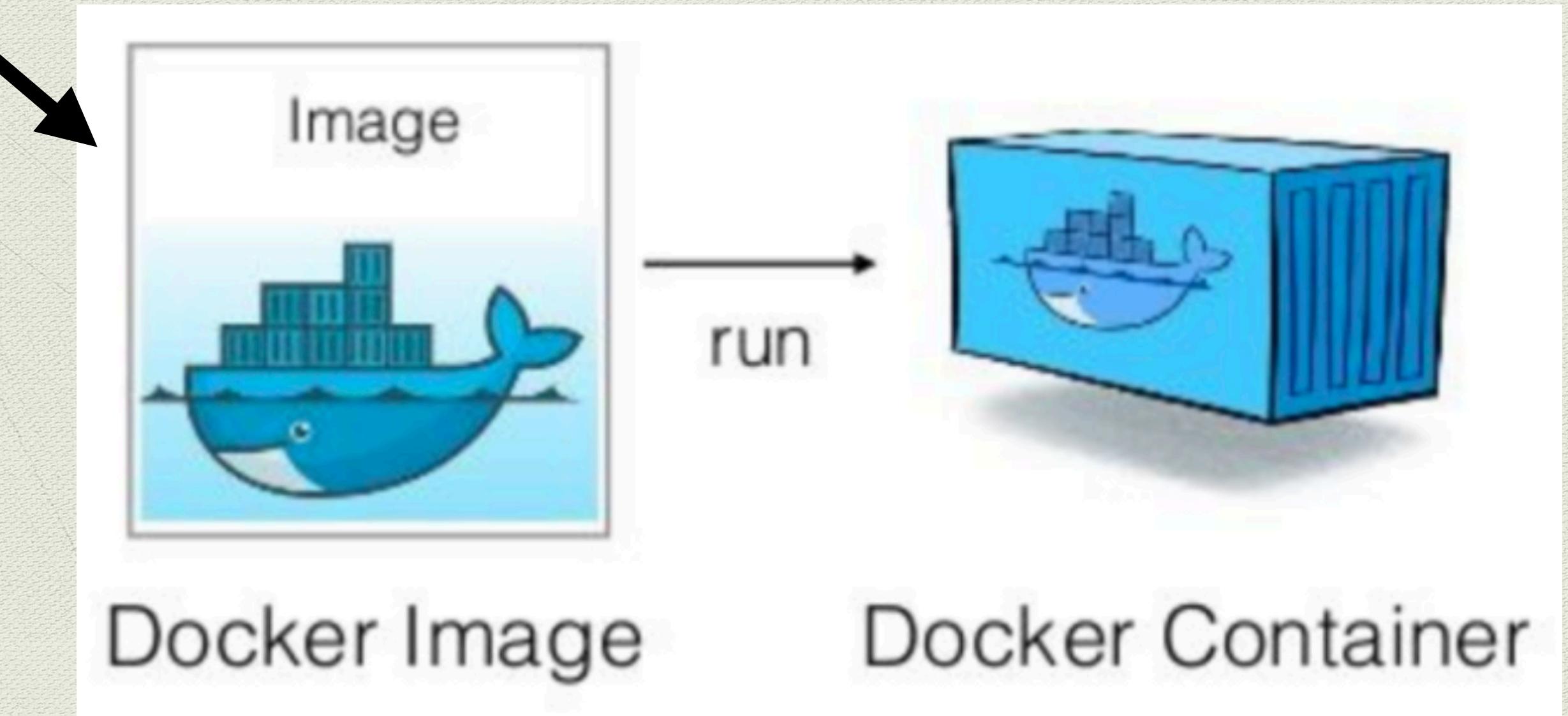


The image for hello-world

1st workshop



pull



Docker Hub

The screenshot shows the Docker Hub interface for the 'busybox' image. On the left, there's a 'BuildRoot' logo with the tagline 'Making Embedded Linux Easy'. The main title is 'busybox ☆' with a star icon. Below it, it says 'Docker Official Images' and 'Busybox base image.' A download count of '10M+' is displayed next to a download icon. Below the download count are several filter buttons: 'Container', 'Linux', 'ARM', 'ARM 64', '386', 'IBM Z', 'PowerPC 64 LE', 'x86-64', 'Base Images', and 'Official Image'. On the right, there's a dropdown menu set to 'Linux - PowerPC 64 LE (latest)'. Below it is a text field containing the command 'docker pull busybox', which is highlighted with a red rectangle and has a black arrow pointing to it from below. To the right of the command is a copy icon. Below the command is a link 'View Available Tags'.

<https://hub.docker.com/>

Run this command to pull the image.

Pull images into local repository

```
[Ekapol-MBP:~ ekapolwongnapapan$ docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
0f8c40e1270f: Pull complete
Digest: sha256:1303dbf110c57f3edf68d9f5a16c082ec06c4cf7604831669faf2c712260b5a0
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
Ekapol-MBP:~ ekapolwongnapapan$ ]
```

```
[Ekapol-MBP:~ ekapolwongnapapan$ docker images
REPOSITORY          TAG           IMAGE ID        CREATED       SIZE
busybox              latest        020584afccce   3 weeks ago   1.22MB
hello-world          latest        fce289e99eb9   10 months ago 1.84kB
Ekapol-MBP:~ ekapolwongnapapan$ ]
```

Run container based on the image

```
[Ekapols-MBP:~ ekapolwongnapapan$ docker run busybox  
Ekapols-MBP:~ ekapolwongnapapan$ ]
```

```
[Ekapols-MBP:~ ekapolwongnapapan$ docker ps -a  
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS           PORTS     NAMES  
906df8c13a36        busybox            "sh"         3 minutes ago   Exited (0) 3 minutes ago  
Ekapols-MBP:~ ekapolwongnapapan$ ]
```

- ❖ Create a container from image.
- ❖ Run the command in the container.
- ❖ Exit the container.

Run container with command

```
[Ekaps-MBP:~ ekapolwongnapapan$ docker run busybox echo "hello from busybox"
hello from busybox]
```

```
[Ekaps-MBP:~ ekapolwongnapapan$ docker ps -a
CONTAINER ID        IMAGE               COMMAND            CREATED             STATUS              PORTS               NAMES
5f1ba64694d8        busybox            "echo 'hello from bu..."   20 minutes ago   Exited (0) 20 minutes ago   inspiring_haslett
906df8c13a36        busybox            "sh"               28 minutes ago   Exited (0) 28 minutes ago   ecstatic_nightingale
```

- ❖ Create a container from image.
- ❖ Run the command in the container.
- ❖ Exit the container.

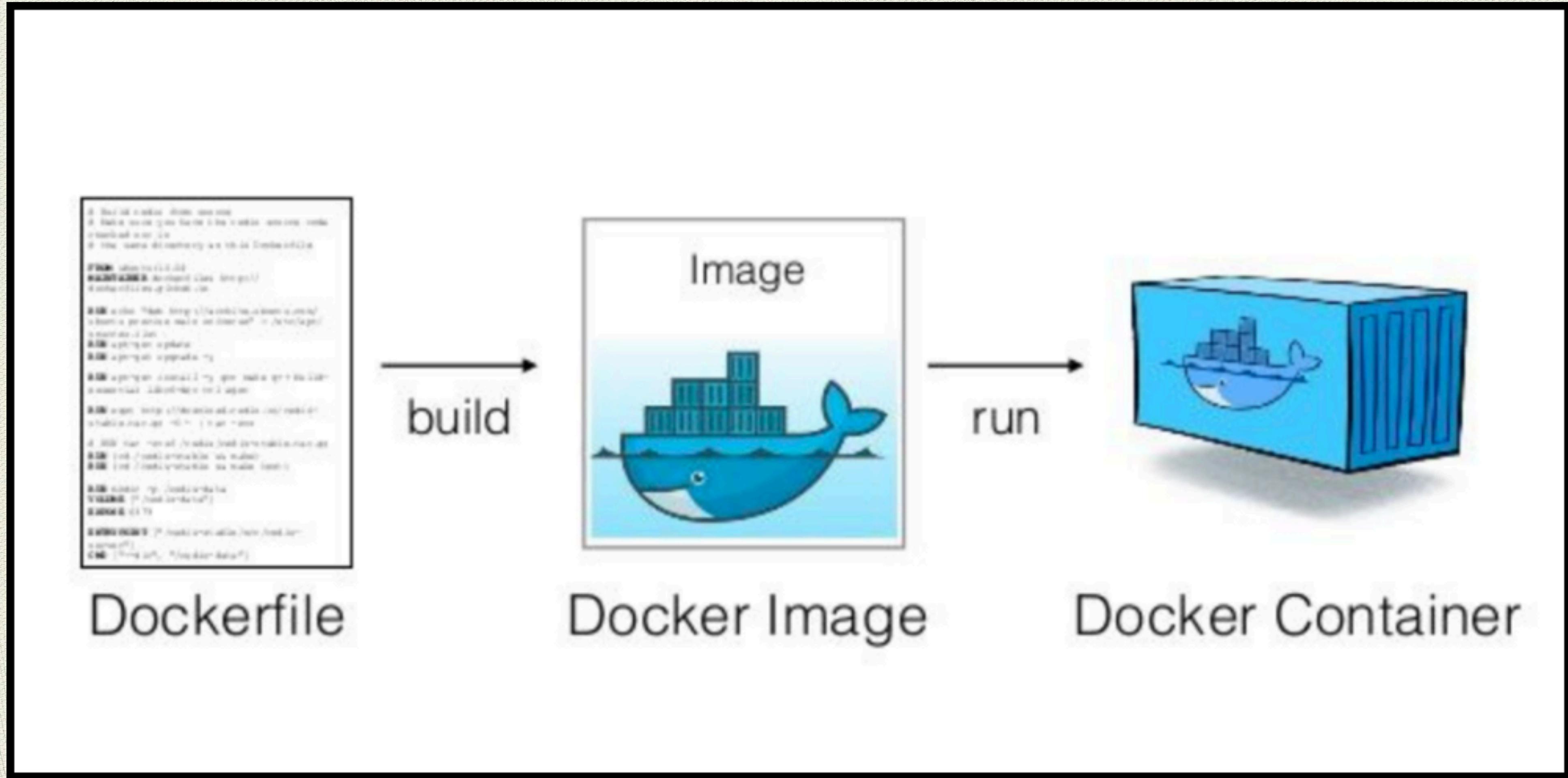
Run container with interactive tty

Enter Bourne shell



```
[Ekapol's-MBP:~ ekapolwongnapapan$ docker run -it busybox sh
/ # ls
bin  dev  etc  home  proc  root  sys  tmp  usr  var
/ # uptime
10:21:00 up 1:22, 0 users, load average: 0.00, 0.02, 0.00
/ # ]
```

Create image and container



Dockerfile

1. Write Dockerfile

```
FROM busybox:latest
CMD ["echo", "hello from busybox"]
```

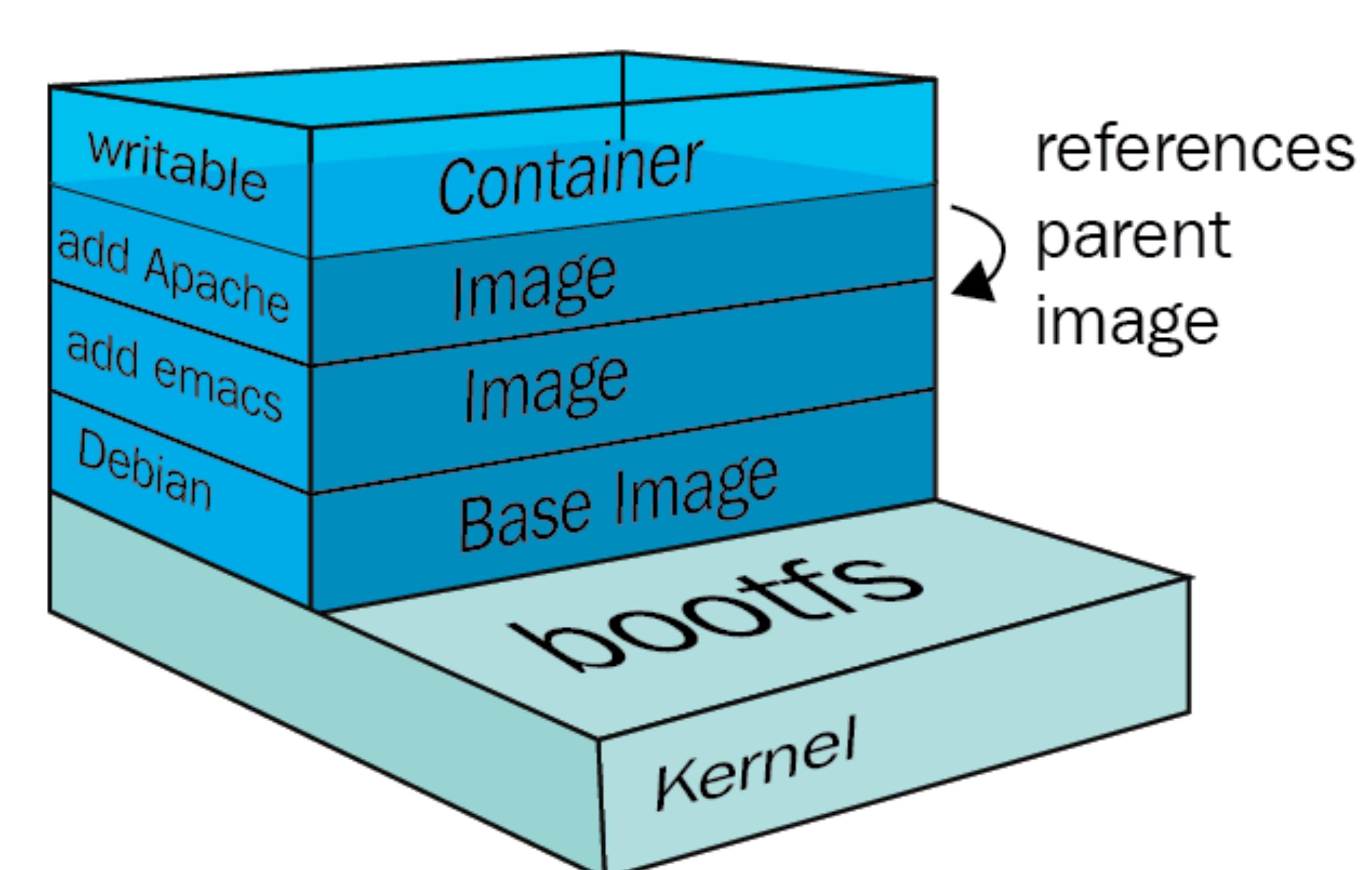
3. List image

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ekapolw/hellobusy	latest	4687063799ec	2 minutes ago	1.22MB

4. Run Container

```
ekapolwongnapapan@Ekaps-MacBook-Pro online_workshop % docker run ekapolw/hellobusy
hello from busybox
```

Image Structure



Images

An image is a read-only template with instructions for creating a Docker container.

Base Image (example)

scratch ☆

[Docker Official Images](#)

an explicitly empty image, especially for building images "FROM scratch"

debian ☆

[Docker Official Images](#)

Debian is a Linux distribution that's composed entirely of free and open-source software.

busybox ☆

[Docker Official Images](#)

Busybox base image.

alpine ☆

[Docker Official Images](#)

A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

ubuntu ☆

[Docker Official Images](#)

Ubuntu is a Debian-based Linux operating system based on free software.

centos ☆

[Docker Official Images](#)

The official build of CentOS.

openjdk ☆

[Docker Official Images](#)

OpenJDK is an open-source implementation of the Java Platform, Standard Edition

python ☆

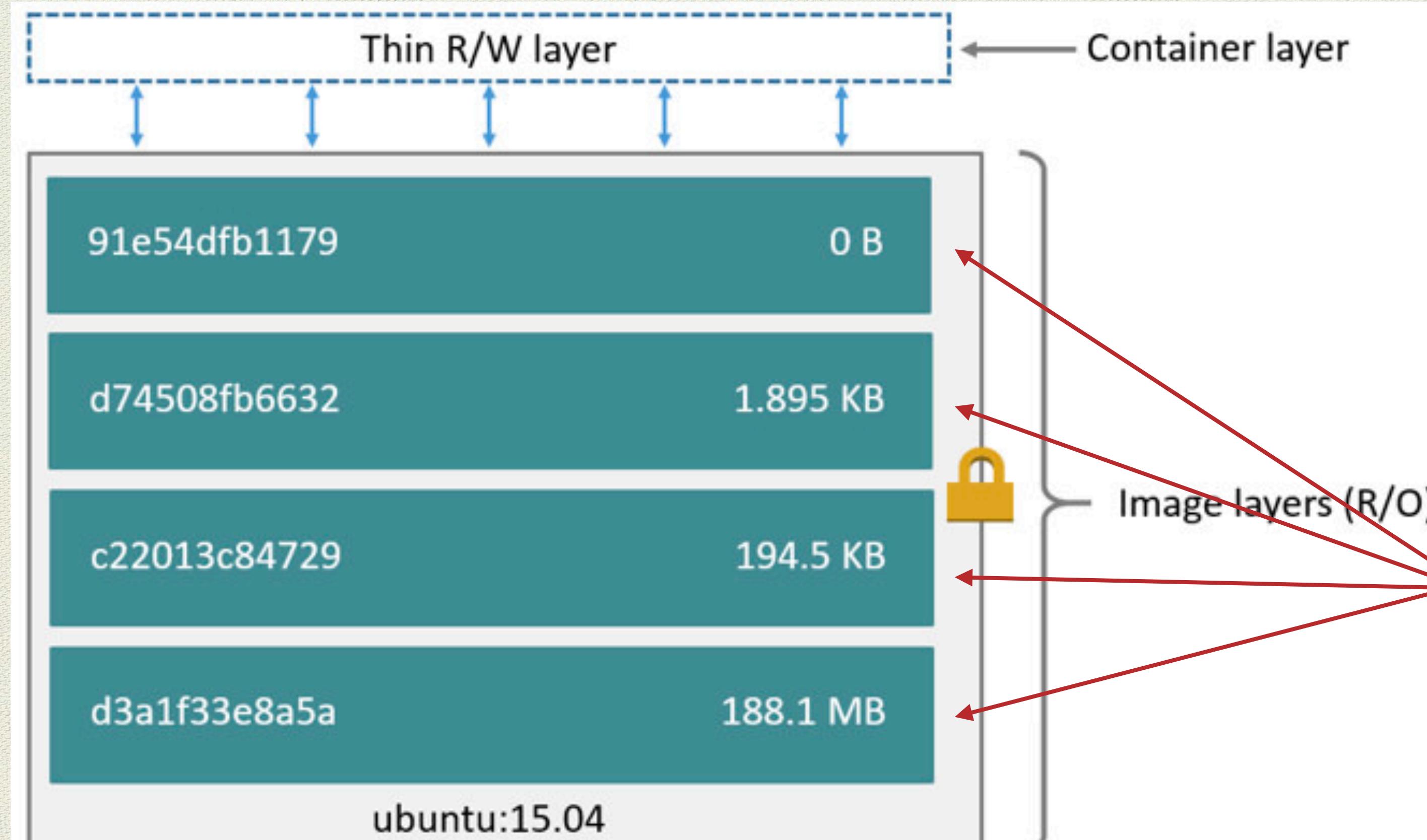
[Docker Official Images](#)

Python is an interpreted, interactive, object-oriented, open-source programming language.

Base Image Size

<u>ubuntu:latest</u> 188 mb Layers: 4	<u>busybox:latest</u> 2 mb Layers: 3	<u>centos:latest</u> 172 mb Layers: 3	<u>opensuse:latest</u> 82 mb Layers: 2	<u>alpine:latest</u> 5 mb Layers: 1
ADD file:c8f078961a543cd... 188 mb	MAINTAINER Jérôme Peta... 0 bytes	MAINTAINER The CentOS ... 0 bytes	MAINTAINER Flavio Castelli 0 bytes	ADD file:98d5decf83ee59e... 5 mb
RUN echo '#!/bin/sh' > /usr... 195 kb	ADD file:8cf517d90fe79547... 2 mb	ADD file:82835f82606420c... 172 mb	ADD file:30a527143b57cd1... 82 mb	
RUN sed -i 's/^#\s*(deb.*u... 2 kb	CMD "/bin/sh" 0 bytes	CMD "/bin/bash" 0 bytes		
CMD "/bin/bash" 0 bytes				

Container



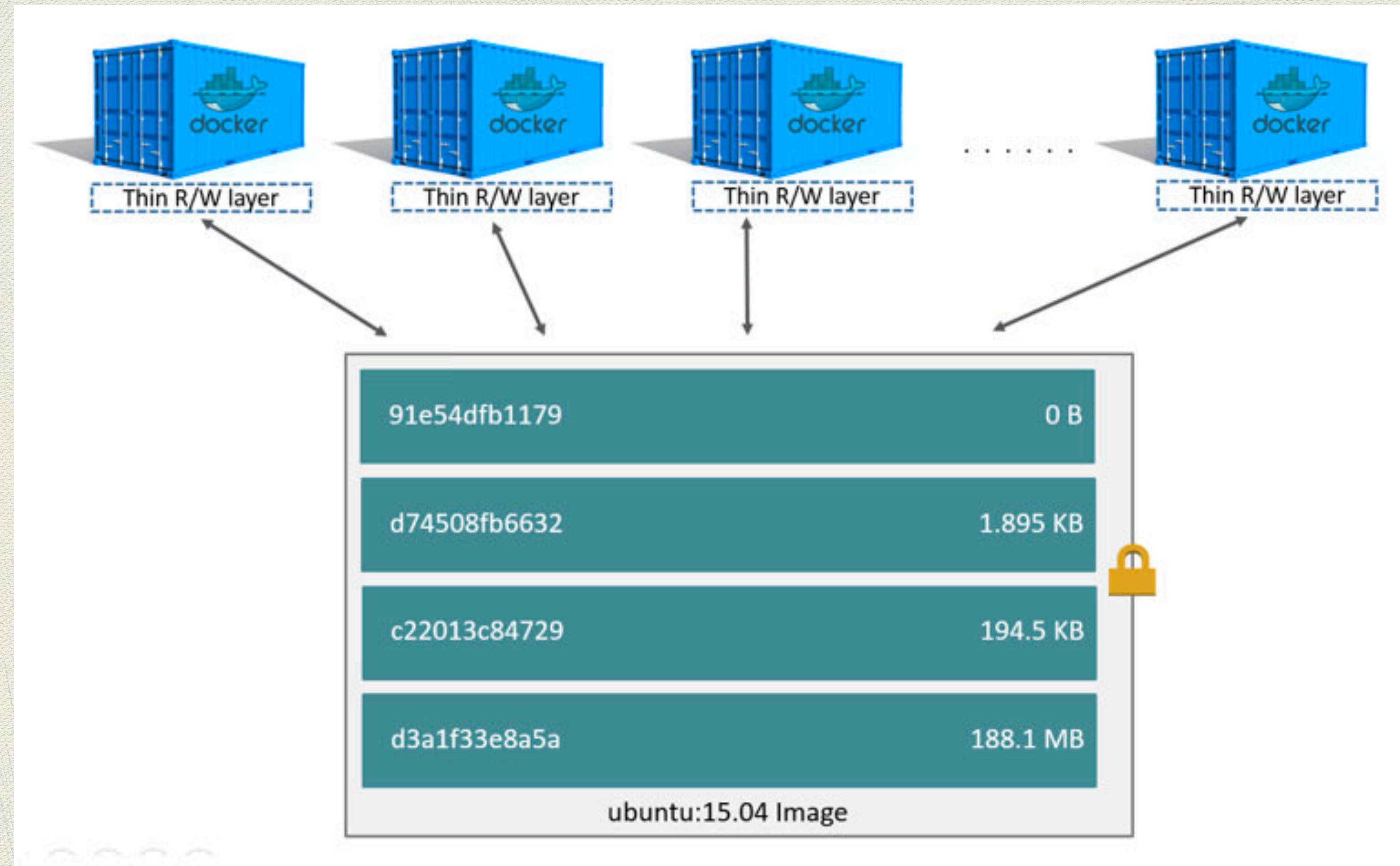
Containers

A container is a runnable instance of an image.

Dockerfile

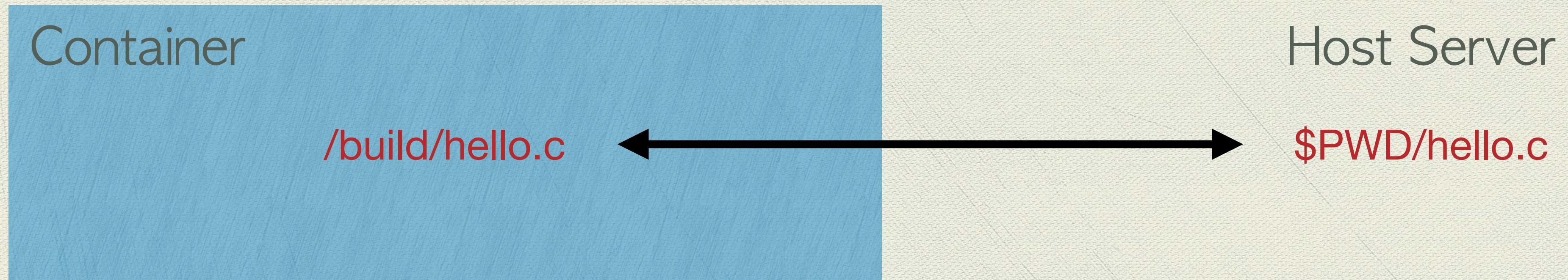
```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Multiple Containers



Hello my edition (2nd workshop)

- Copy hello.c from <https://github.com/docker-library/hello-world/> into local directory.
- Docker Desktop for Windows or Mac is using Linux VM, so we need to build hello binary in Linux, not Windows or Mac.
- In local directory where place hello.c, run “`docker run --rm -it -v $PWD:/build ubuntu:16.04`”
- In container, `apt-get update && apt-get install build-essential`
- In container, `cd /build`
- In container, `gcc -o hello -static -nostartfiles hello.c`



* for Docker Toolbox : `docker run --rm -it -v //c/Users/docker/workshop_2:/build ubuntu:16.04`

Dockerfile

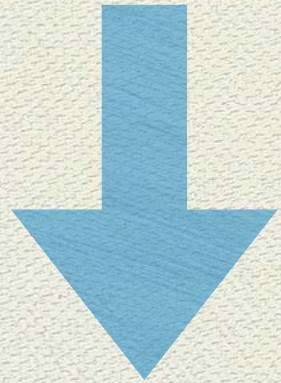
A Dockerfile is a simple text file that contains a list of commands that Docker client calls while creating an image.

Dockerfile

```
FROM scratch
ADD hello /
CMD ["/hello"]
```



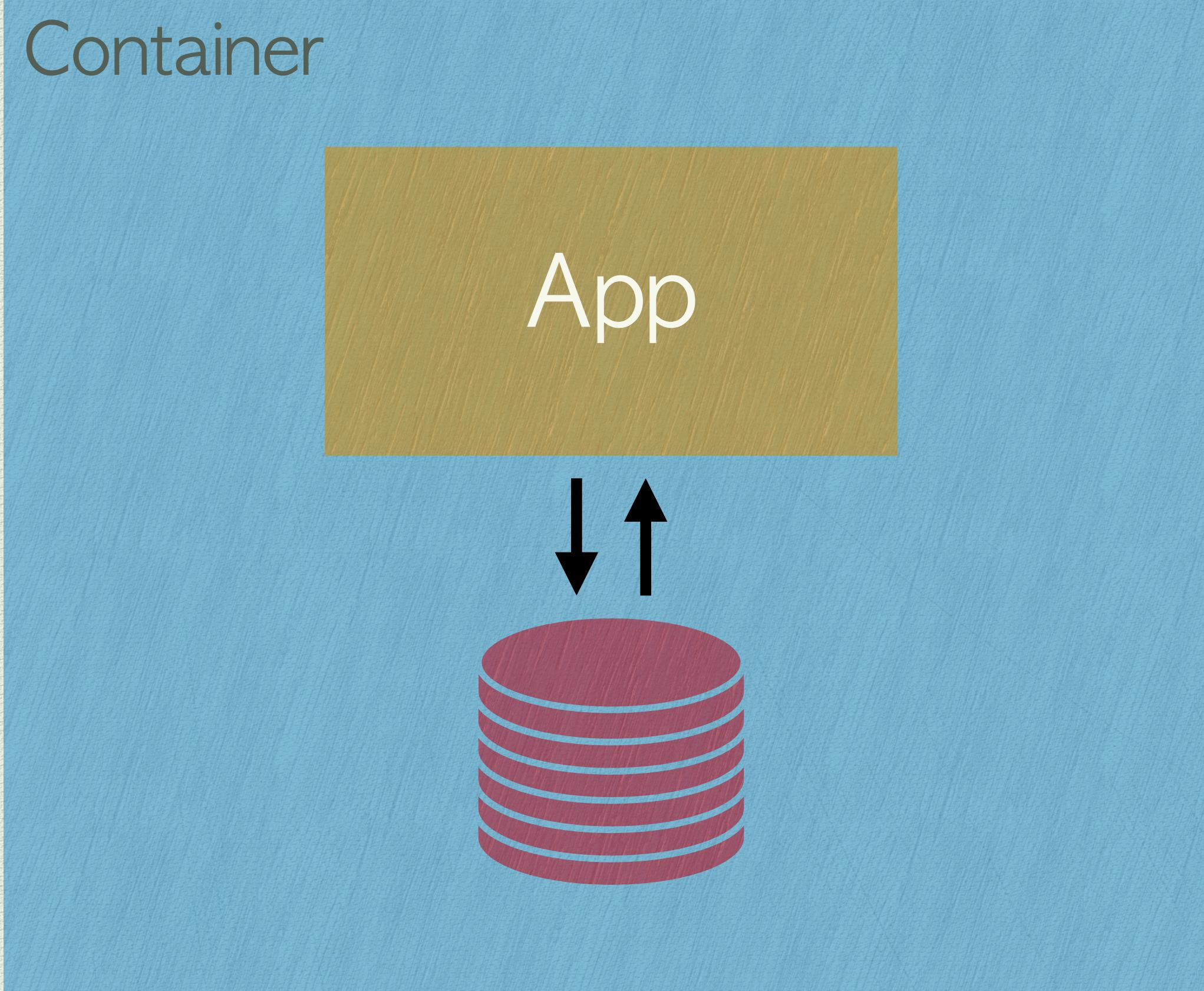
docker build --tag hello .



Image

* See how to build hello in next page

Persistence ?

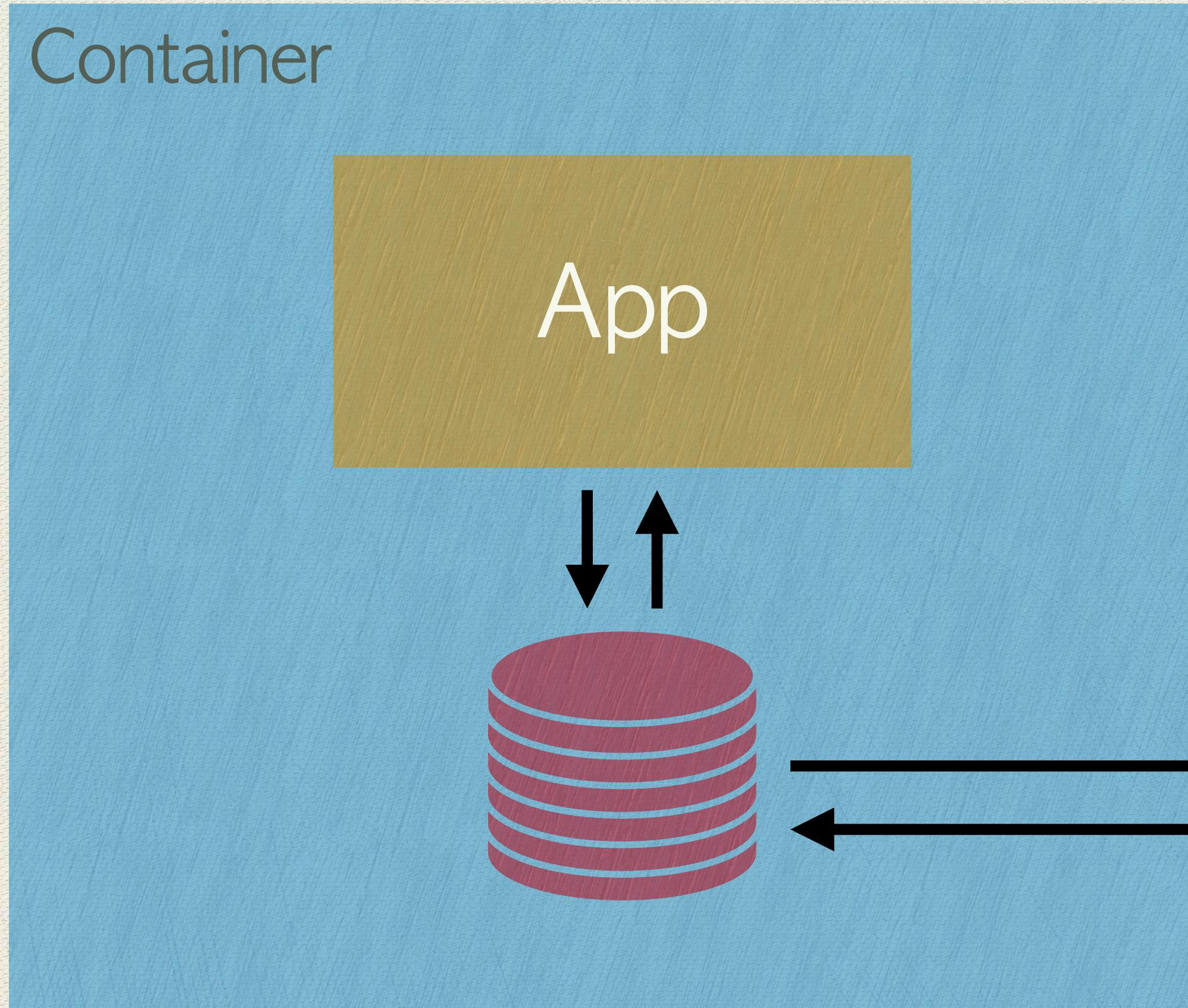


When stop container, data could not be accessed.

When delete container, data are deleted.

Volume

To keep data around when container is removed.
To share data between the host filesystem and the Docker container.
To share data among Docker containers.



```
docker run --rm -it -v $PWD:/build ubuntu:16.04
```

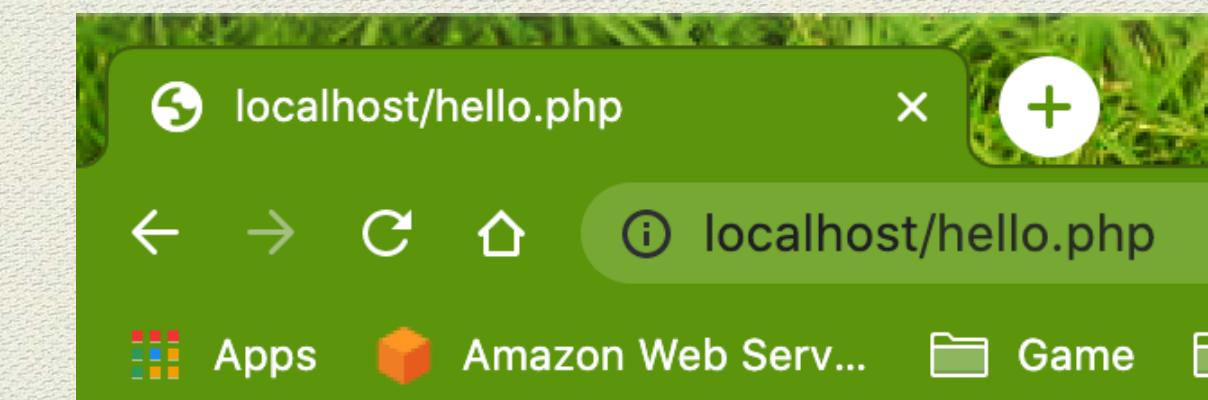
Storage in Host Server
Permanent

Develop Mode (3rd workshop)

Create and Run PHP Web Page

```
php hello.php ×  
  
workshop_3 > src > php hello.php  
1  <!DOCTYPE html>  
2  <html>  
3  <body>  
4  <h1>Hello php</h1>  
5  <?php echo "Hello from php"; ?>  
6  </body>  
7  </html>
```

```
Dockerfile ×  
  
workshop_3 > Dockerfile > ...  
1  FROM php:7.2-apache  
2  COPY src/ /var/www/html/
```



Hello php

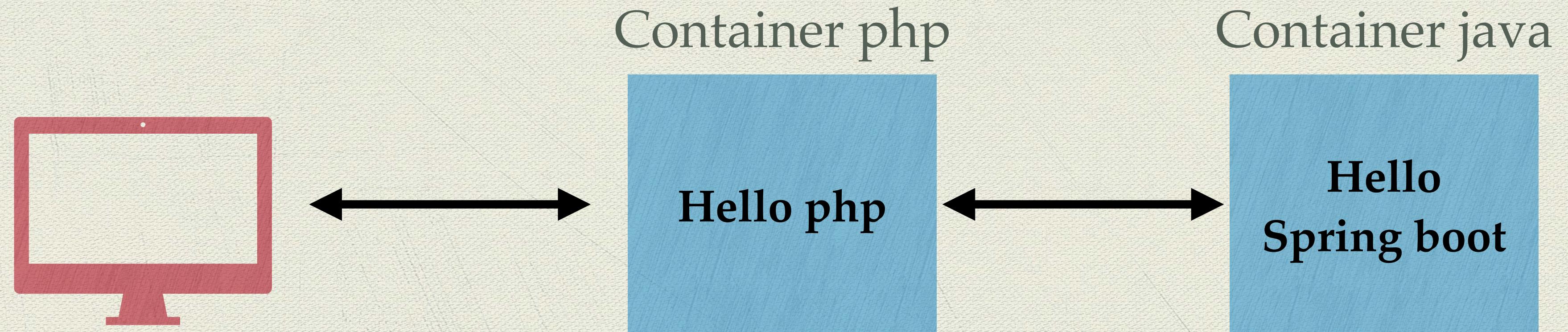
Hello from php

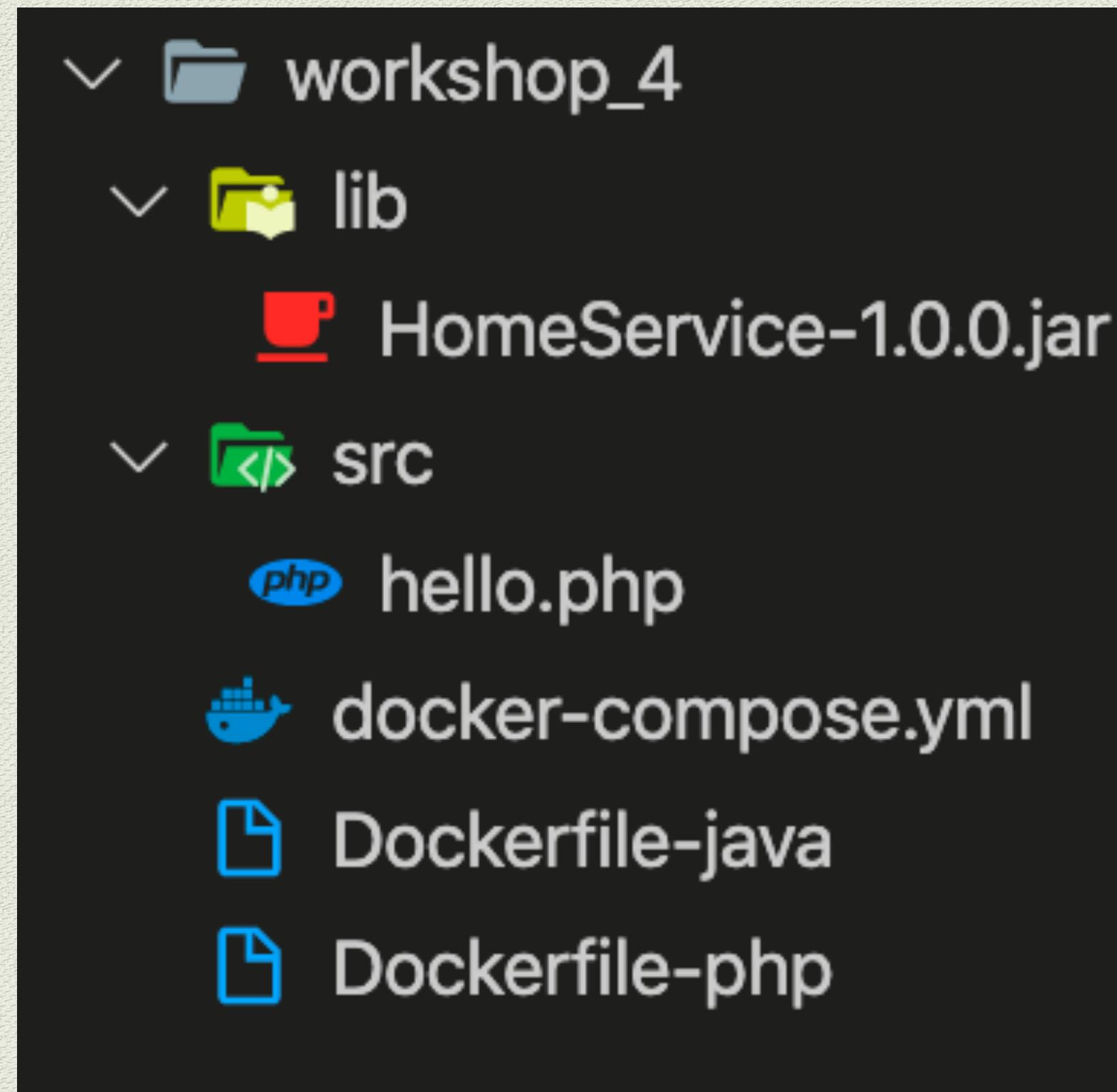
```
docker build -t ekapolw/hellp-php .
```

```
docker run -d -p 80:80 --name my-apache-php-app -v "$PWD":/var/www/html ekapolw/hellp-php
```

```
docker run -d -p 80:80 --name my-apache-php-app -v //c/Users/docker/workshop_3/src:/var/www/html ekapolw/hello-php
```

Hello Network (4th workshop)





```
workshop_4 > ⌂ Dockerfile-java
1  FROM openjdk:8-jdk-alpine
2  VOLUME /tmp
3  ARG JAR_FILE=*.jar
4  COPY lib/${JAR_FILE} HomeService.jar
5  ENTRYPOINT [ "java", "-jar", "/HomeService.jar" ]
```

```
workshop_4 > ⌂ Dockerfile-php
1  FROM php:7.2-apache
2  COPY src/ /var/www/html/|
```

```
workshop_4 > ⚓ docker-compose.yml
 1  version: "3"
 2  services:
 3
 4    java-container:
 5      build:
 6        context: .
 7        dockerfile: Dockerfile-java
 8        container_name: java-container
 9      ports:
10        - 8080:8080
11      networks:
12        - workshop-network
13
14    php-container:
15      build:
16        context: .
17        dockerfile: Dockerfile-php
18        container_name: php-container
19      ports:
20        - 80:80
21      networks:
22        - workshop-network
23      depends_on:
24        - java-container
25
26  networks:
27    workshop-network:
28      driver: bridge
```

Build container network

docker-compose.yml

docker-compose up -d

docker-compose down

Network Drivers

Bridge

Default network driver.
Multiple containers are communicating with the same docker host.

Host

This driver removes the network isolation between docker containers and docker host.

Overlay

This driver enables swarm services to communicate with each other.
Containers on different Docker Hosts could communicate.

None

This driver disables all the networking.

macvlan

This driver assigns mac address to containers to make them look like physical devices.

Save image into tar

```
[Ekapol-MacBook-Pro>CreateImage ekapolwongnapapan$ docker images
REPOSITORY          TAG           IMAGE ID        CREATED       SIZE
hello               latest        60cac4cae542   47 minutes ago  2.75kB
ubuntu              16.04         56bab49eef2e   14 hours ago   123MB
busybox             latest        020584afccce   3 weeks ago    1.22MB
[Ekapol-MacBook-Pro>CreateImage ekapolwongnapapan$ docker save -o hello.tar hello
[Ekapol-MacBook-Pro>CreateImage ekapolwongnapapan$ ls
Dockerfile          hello         hello.c        hello.tar
Ekapol-MacBook-Pro>CreateImage ekapolwongnapapan$
```

Load tar into image

```
[Ekaps-MacBook-Pro>CreateImage ekapolwongnapapan$ ls
Dockerfile      hello          hello.c          hello.tar
[Ekaps-MacBook-Pro>CreateImage ekapolwongnapapan$ docker images
REPOSITORY      TAG            IMAGE ID         CREATED        SIZE
ubuntu          16.04          56bab49eef2e    14 hours ago   123MB
busybox         latest         020584afccce  3 weeks ago   1.22MB
[Ekaps-MacBook-Pro>CreateImage ekapolwongnapapan$ docker load -i hello.tar
c93b9ce75bcd: Loading layer [=====>] 4.608kB/4.608kB
Loaded image: hello:latest
[Ekaps-MacBook-Pro>CreateImage ekapolwongnapapan$ docker images
REPOSITORY      TAG            IMAGE ID         CREATED        SIZE
hello           latest         60cac4cae542    52 minutes ago 2.75kB
ubuntu          16.04          56bab49eef2e    14 hours ago   123MB
busybox         latest         020584afccce  3 weeks ago   1.22MB
Ekaps-MacBook-Pro>CreateImage ekapolwongnapapan$
```

Clean Up Containers

```
[Ekapolwongnapapan:~ ekapolwongnapapan$ docker ps -a
CONTAINER ID        IMAGE               COMMAND
0cbeb8e612b4        busybox            "sh"
58621879e96a        busybox            "echo 'Hello from bu..."
e46f8958a6d9        busybox            "sh"
```

```
[Ekapolwongnapapan:~ ekapolwongnapapan$ docker container rm 0cbeb8e612b4
0cbeb8e612b4
```

Container id

```
[Ekapolwongnapapan:~ ekapolwongnapapan$ docker ps -a
CONTAINER ID        IMAGE               COMMAND
58621879e96a        busybox            "echo 'Hello from bu..."
e46f8958a6d9        busybox            "sh"
```

```
[Ekapolwongnapapan:~ ekapolwongnapapan$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
58621879e96a5ad452dd71f3bb53cf33a64fc54e94e9836dc7d5a004a6dd424e
e46f8958a6d979647da12313e1c50db67e04d158591be05e70ce47f713f5c791

Total reclaimed space: 0B
```

Remove all
not running containers

```
Ekapols-MBP:~ ekapolwongnapapan$ docker ps -a
CONTAINER ID        IMAGE               COMMAND
Ekapols-MBP:~ ekapolwongnapapan$
```

Thanks
Enjoy Docker



docker

Create TechnicalLabService image

Create image for TechnicalLabService.jar (Spring Boot)

```
FROM openjdk:8-jdk-alpine
VOLUME /tmp
ARG JAR_FILE=*.jar
COPY ${JARFILE} TechnicalLabService.jar
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/TechnicalLabService.jar"]
```



docker build -t springio/technical-lab-service .



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springio/technical-lab-service	latest	e5eff8a3e7a6	8 minutes ago	122MB

Run TechnicalLabService Container

```
docker run -p 8090:8090 springio/technical-lab-service
```

