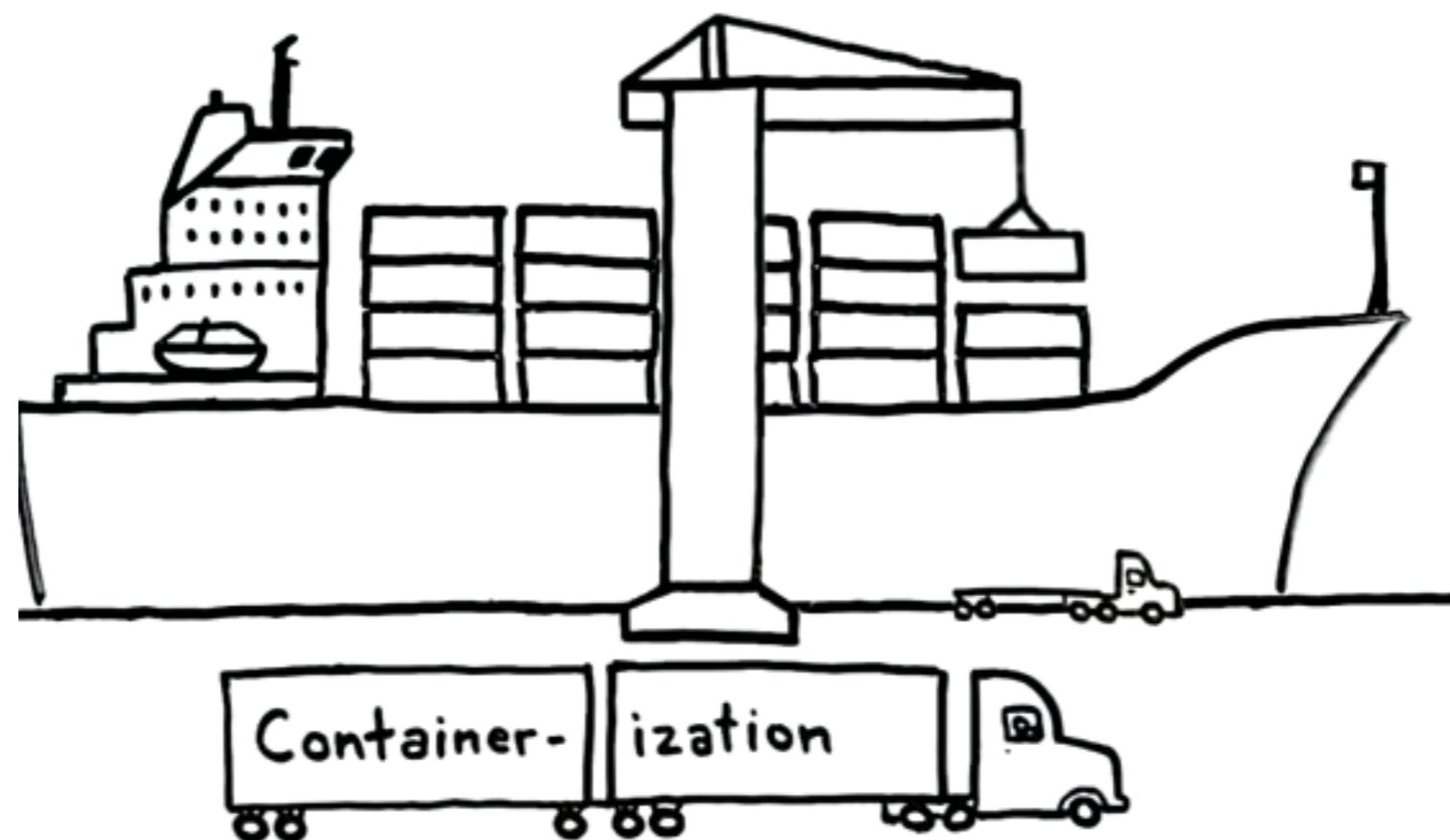


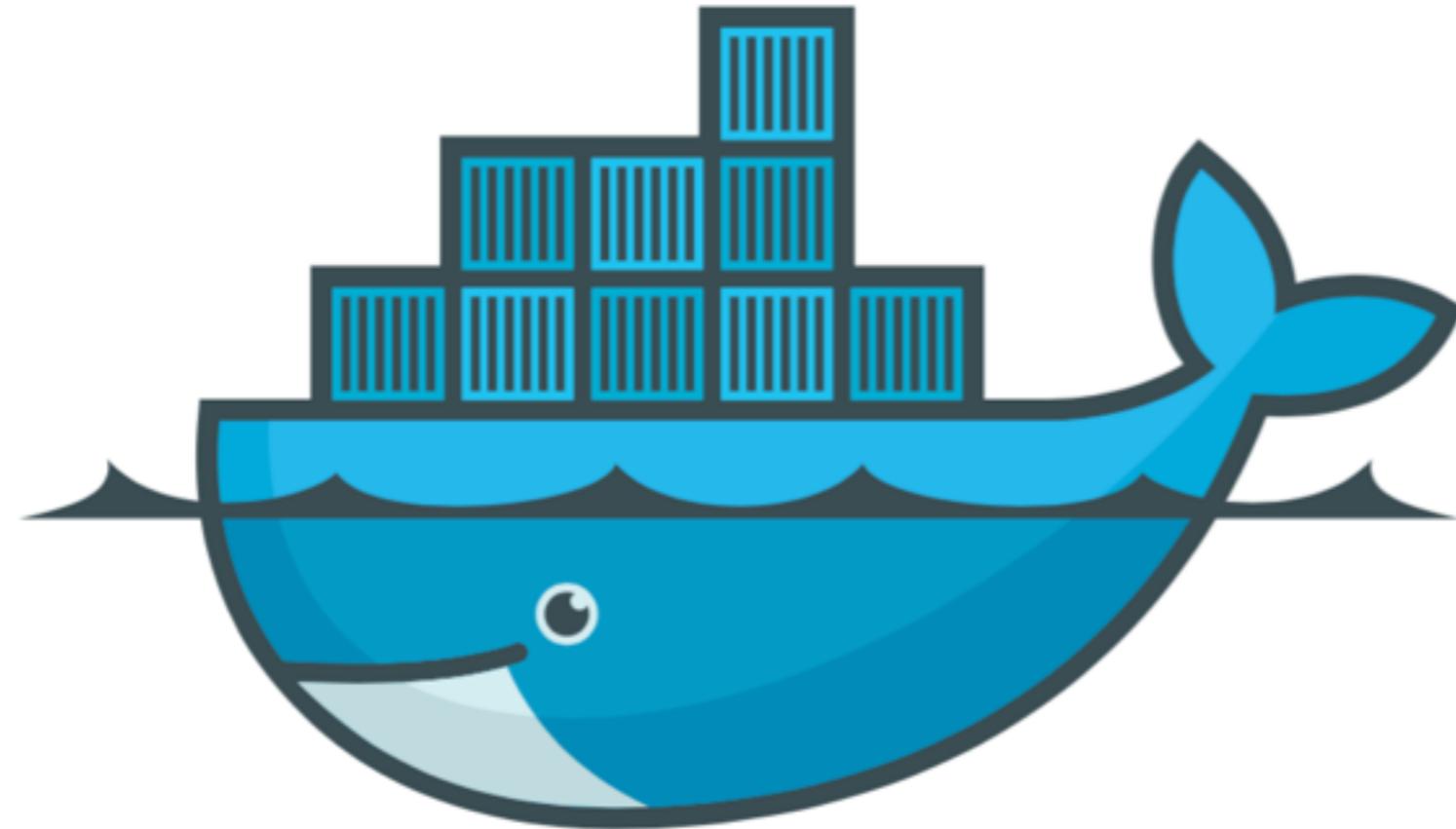


# docker 101

From Developer View  
[somkiat.cc](http://somkiat.cc)

# Containerization





docker

# **Docker != Containers**

# Docker **manages** Containers

# **Docker manages Containers**

**Build images to run as containers**

**Manage applications with docker compose**

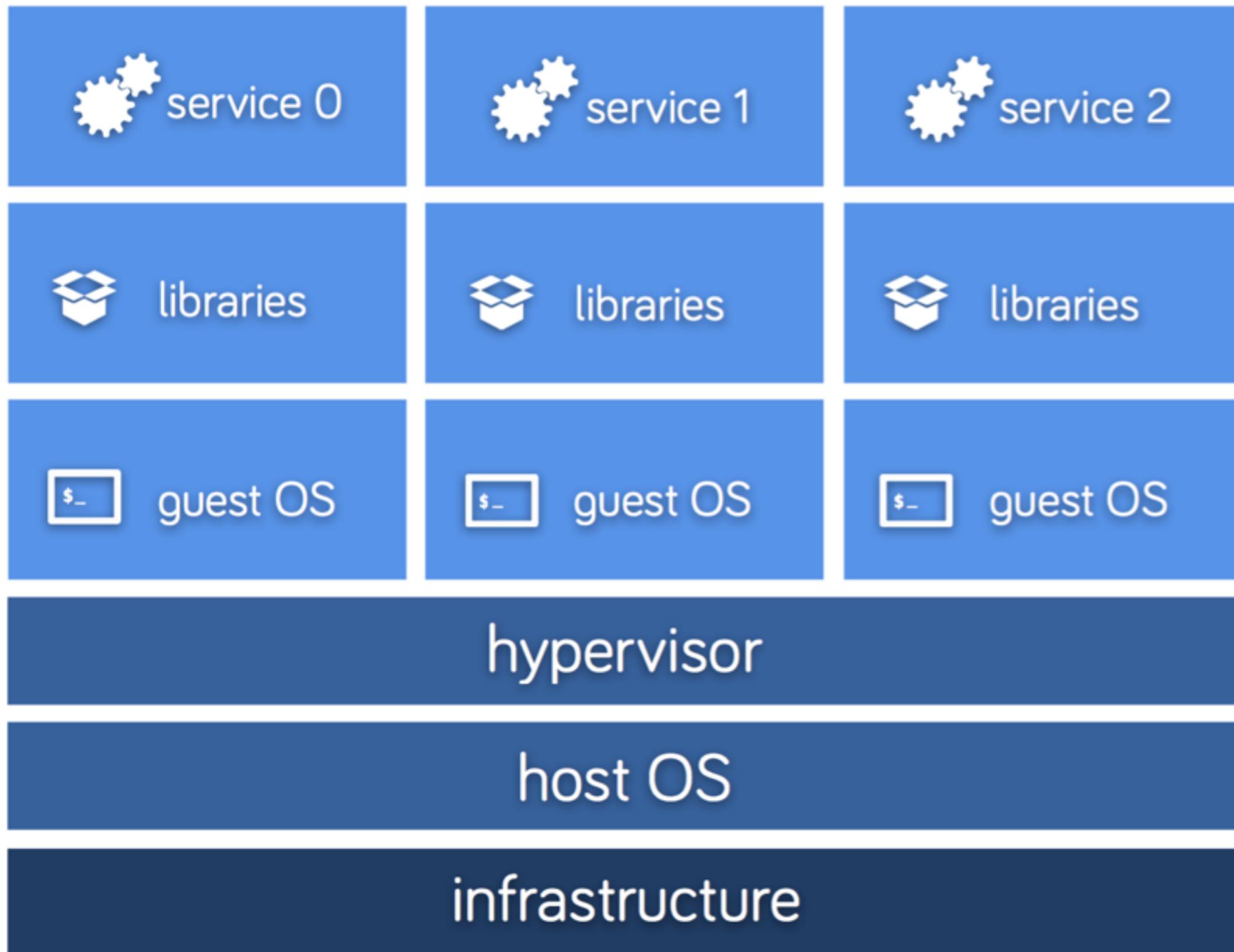
**Provision machines with docker machine**

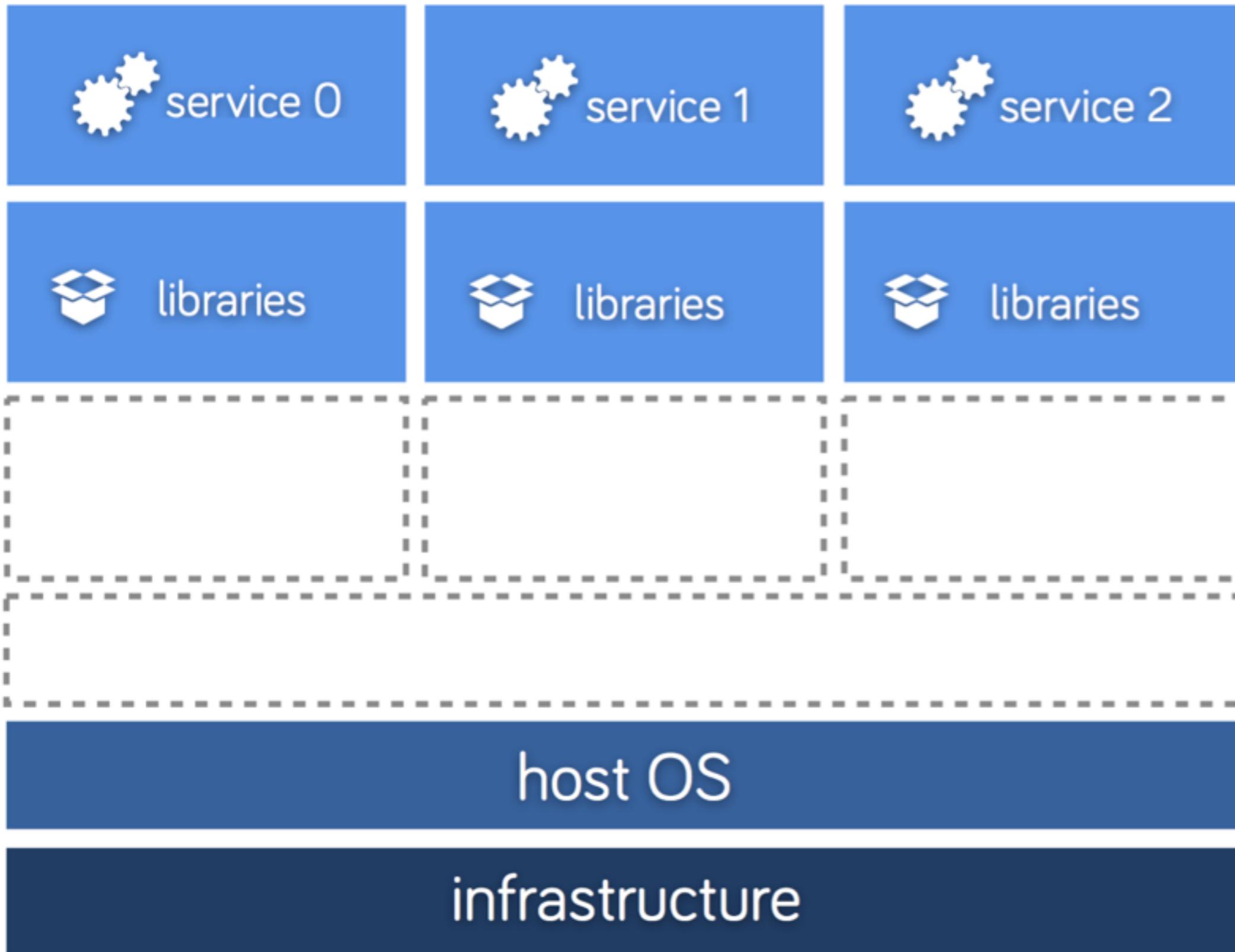
# Containers

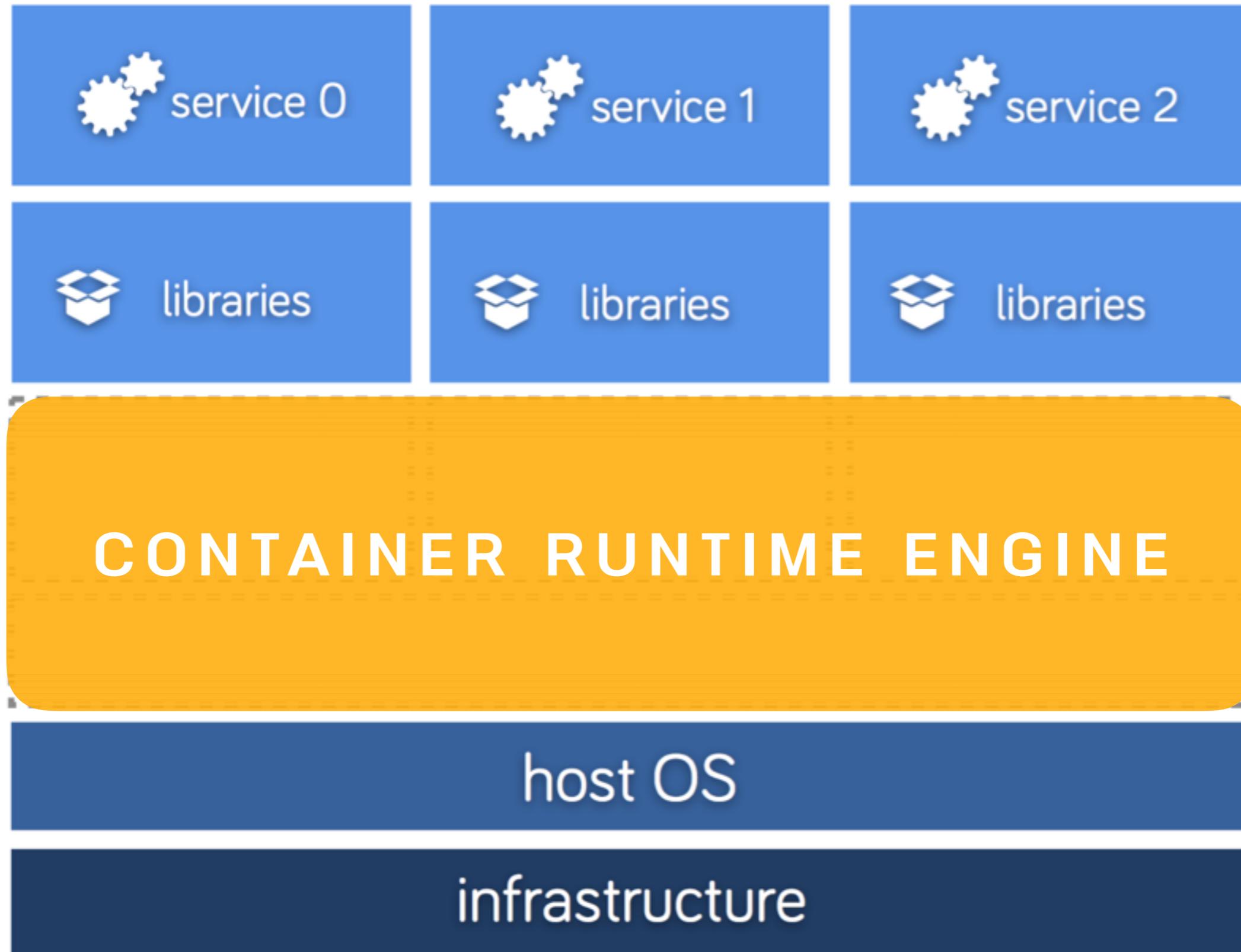
**Share** the kernel of host system

**Isolated** from other containers

**Fast** boot time and **Low** overhead







**Start with containers  
seem more complex !!**

**BUT**  
reduce time to run your app

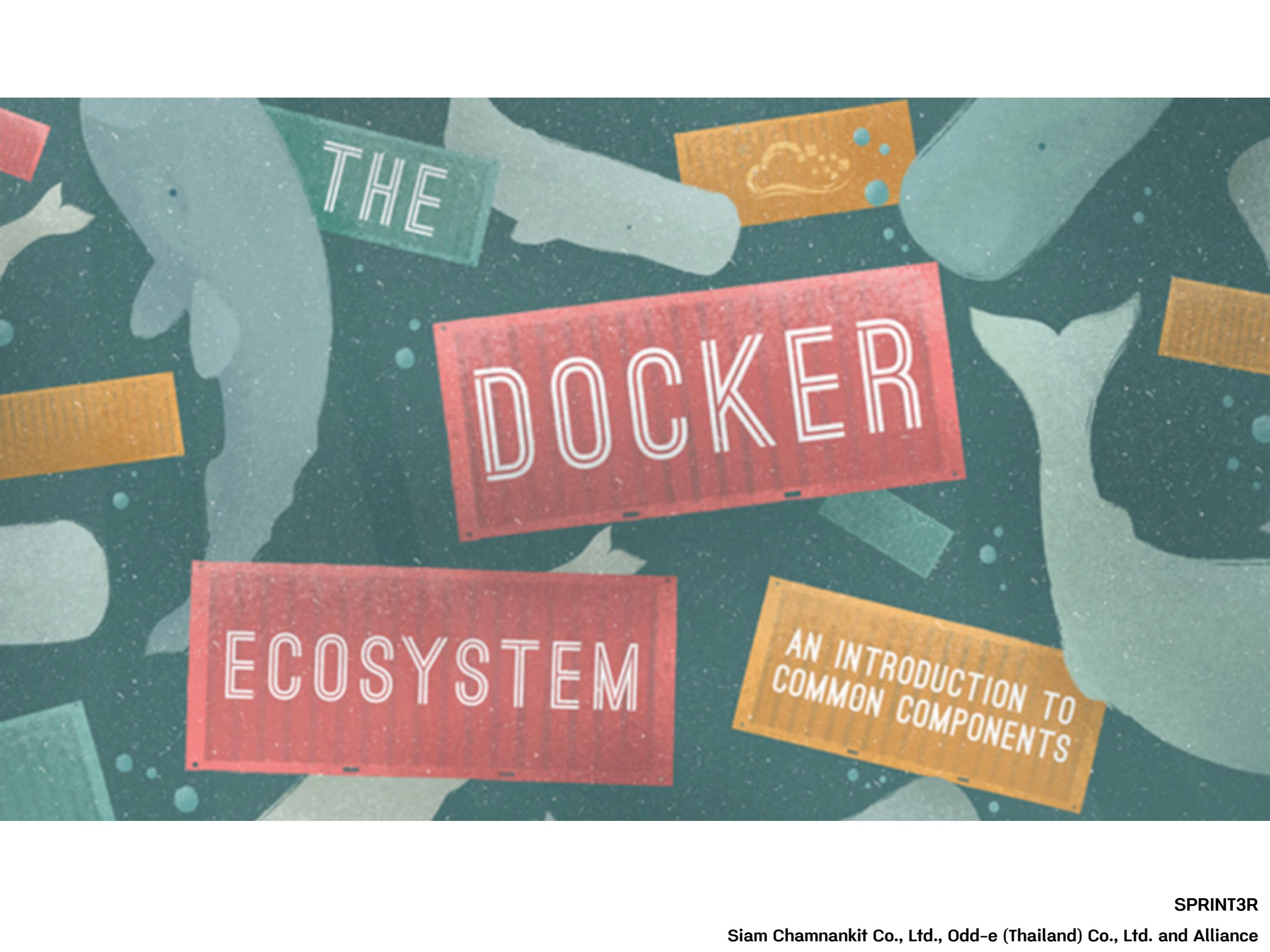
**BUT**  
less time to provisioning  
and rebooting

**BUT**  
easy to manage dependencies

**BUT**  
use multiple language/version  
without hacking or additional tools

**BUT**  
use the same OS in production

**BUT**  
**easy to deploy**



THE  
ECOSYSTEM

DOCKER

AN INTRODUCTION TO  
COMMON COMPONENTS



Build



Ship



Run

# **Docker Images**

# **Image vs Container**

**Image like a class**

**Container is a instance of class**

# Create my image

## Create a Dockerfile

```
$docker build -t <image name> . \  
$docker images  
$docker commit  
$docker push  
$docker pull
```

# Docker Hub



## Explore Official Repositories

 <b>busybox</b> official	442 STARS	59.4 M PULLS	<a href="#">DETAILS</a>
 <b>ubuntu</b> official	3.0 K STARS	38.1 M PULLS	<a href="#">DETAILS</a>
 <b>swarm</b> official	122 STARS	21.5 M PULLS	<a href="#">DETAILS</a>
 <b>nginx</b> official	2.0 K STARS	18.9 M PULLS	<a href="#">DETAILS</a>
 <b>redis</b> official	1.5 K STARS	16.8 M PULLS	<a href="#">DETAILS</a>

# Docker Hub

The screenshot shows the Docker Hub homepage with a search bar, sign-up, and log-in buttons at the top. Below the header, a link to "Explore Official Repositories" is visible. A large red annotation ">> Many images" is overlaid on the repository list. The list includes the following entries:

Repository	Stars	Pulls	Details
busybox/official	442	59.4 M	> DETAILS
ubuntu/official	3.0 K	31.1 M	> DETAILS
swarm/official	122	21.5 M	> DETAILS
nginx/official	2.0 K	18.9 M	> DETAILS
redis/official	1.5 K	16.8 M	> DETAILS

Use  
**Docker Compose**  
to build

# Docker Compose

Use a **Dockerfile** to define your app's env

Define services of your app in **docker-compose.yml**

**Start and run with**  
**\$docker-compose up**

# Dockerfile

```
1 FROM ruby:2.2.3
2 RUN apt-get update -qq && apt-get i
3 WORKDIR /var/app
4 COPY Gemfile /var/app/Gemfile
5 RUN bundle install
```

# docker-compose.ml

```
1 db:
2   image: postgres
3
4 web:
5   build: .
6   ports:
7     - "3000:3000"
8   volumes:
9     - "./:/var/app"
10  working_dir: /var/app
11  command: rails s -b '0.0.0.0'
12  links:
13    - db:db
```

# Web Architecture



# \$docker-compose up

```
[MacBook-Pro-2:rails-app somkiat$ docker-compose up
Pulling db (postgres:latest)...
latest: Pulling from library/postgres
8b57c6bf5c36: Pull complete
ada8aa55613a: Pull complete
b3de40c730c6: Pull complete
7b8f7ba65711: Pull complete
1bb8b8636476: Pull complete
eca1007b842c: Pull complete
189c3f75d39c: Pull complete
ad45579db688: Pull complete
7c5cb7f4a1a8: Pull complete
c6770af17534: Pull complete
6b6e9868d88a: Downloading [=====>                                         ] 21.23 MB/42.82 MB
3b7eb1ffa9e2: Download complete
c8d89116b183: Download complete
6d9d81c13df4: Download complete
af366c0e9aed: Pulling fs layer
6246cbebdfe0: Download complete
b731665a3278: Download complete
7f586af960a2: Download complete
9aae83d4127f: Download complete
```

# \$docker-compose up

```
---> 72336174e973
Step 2 : RUN apt-get update -qq && apt-get install -y build-essential nodejs libpq-dev && mkdir -p /var/app
    ---> Running in 76efa3075ba3
Reading package lists...
Building dependency tree...
Reading state information...
libpq-dev is already the newest version.
The following extra packages will be installed:
  dpkg-dev fakeroot libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libc-ares2 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libtimedate-perl libv8-3.14.5
Suggested packages:
  debian-keyring
The following NEW packages will be installed:
  build-essential dpkg-dev fakeroot libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libc-ares2 libdpkg-perl
  libfakeroot libfile-fcntllock-perl libtimedate-perl libv8-3.14.5 nodejs
0 upgraded, 13 newly installed, 0 to remove and 16 not upgraded.
Need to get 4903 kB of archives.
After this operation, 12.5 MB of additional disk space will be used.
Get:1 http://security.debian.org/ jessie/updates/main libdpkg-perl all 1.17.26 [1072 kB]
Get:2 http://security.debian.org/ jessie/updates/main dpkg-dev all 1.17.26 [1545 kB]
Get:3 http://httpredir.debian.org/debian/ jessie/main libc-ares2 amd64 1.10.0-2 [76.7 kB]
Get:4 http://httpredir.debian.org/debian/ jessie/main libtimedate-perl all 2.3000-2 [42.2 kB]
```

# \$docker-compose up

```
Removing intermediate container 98005244c64c
Step 5 : RUN bundle install
--> Running in c4bf91c8e778
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/..
Resolving dependencies.....
Installing rake 10.5.0
Installing i18n 0.7.0
Installing json 1.8.3 with native extensions
Installing minitest 5.8.3
Installing thread_safe 0.3.5
Installing tzinfo 1.2.2
Installing activesupport 4.2.3
Installing builder 3.2.2
Installing erubis 2.7.0
Installing mini_portile2 2.0.0
Installing nokogiri 1.6.7.1 with native extensions
Installing rails-deprecated_sanitizer 1.0.3
Installing rails-dom-testing 1.0.7
Installing loofah 2.0.3
Installing rails-html-sanitizer 1.0.2
Installing actionview 4.2.3
----- 1 / 1
```

# \$docker-compose up

```
Successfully built 326594d9cf2c
Creating railsapp_web_1
Attaching to railsapp_db_1, railsapp_web_1
db_1  | The files belonging to this database system will be owned by user "postgres".
db_1  | This user must also own the server process.
db_1
db_1  | The database cluster will be initialized with locale "en_US.utf8".
db_1  | The default database encoding has accordingly been set to "UTF8".
db_1  | The default text search configuration will be set to "english".
db_1
db_1  Data page checksums are disabled.
db_1
db_1  fixing permissions on existing directory /var/lib/postgresql/data ... ok
db_1  creating subdirectories ... ok
db_1  selecting default max_connections ... 100
db_1  selecting default shared_buffers ... 128MB
db_1  selecting dynamic shared memory implementation ... posix
db_1  creating configuration files ... ok
db_1  creating template1 database in /var/lib/postgresql/data/base/1 ... ok
db_1  initializing pg_authid ... ok
db_1  initializing dependencies ... ok
db_1  creating system views ... ok
db_1  loading system objects' descriptions ... ok
db_1  creating collations ... ok
db_1  creating conversions ... ok
db_1  creating dictionaries ... ok
db_1  setting privileges on built-in objects ... ok
db_1  creating information schema ... ok
```

# \$docker-compose up

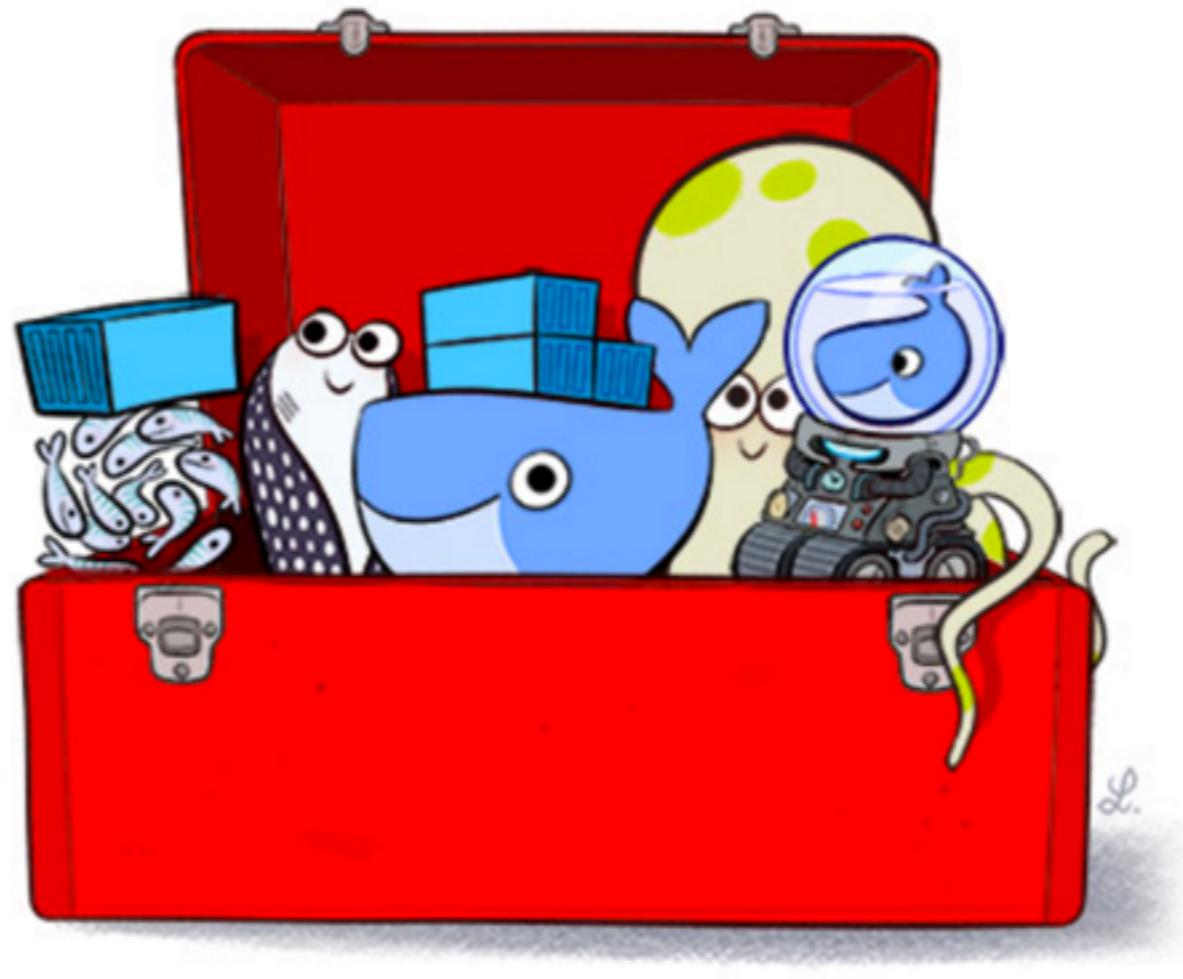
```
Starting railsapp_db_1
Starting railsapp_web_1
Attaching to railsapp_db_1, railsapp_web_1
db_1  | LOG:  database system was interrupted; last known up at 2016-01-14 06:15:28 UTC
db_1  | LOG:  database system was not properly shut down; automatic recovery in progress
db_1  | LOG:  invalid record length at 0/17096B0
db_1  | LOG:  redo is not required
db_1  | LOG:  MultiXact member wraparound protections are now enabled
db_1  | LOG:  database system is ready to accept connections
db_1  | LOG:  autovacuum launcher started
web_1 | [2016-01-14 06:15:45] INFO  WEBrick 1.3.1
web_1 | [2016-01-14 06:15:45] INFO  ruby 2.2.3 (2015-08-18) [x86_64-linux]
web_1 | [2016-01-14 06:15:45] INFO  WEBrick::HTTPServer#start: pid=1 port=3000
```

# **Building web app with Docker**

# Install docker

## Docker Toolbox

[Getting Started Guide \(Mac\)](#) | [Getting Started Guide \(Windows\)](#) | [Contribute to Toolbox](#)



# Docker Machine



# Docker Machine

Easy way to create dev & testing machine  
from your local command line

**\$docker-machine create <name> --driver virtualbox**

# Demo

Running multi-Container  
with Docker compose



# Demo

Running multi-Container  
with Docker compose



# Demo

Running multi-Container  
with Docker compose



# My Goals

Run web app in a container

**Developer can ::**

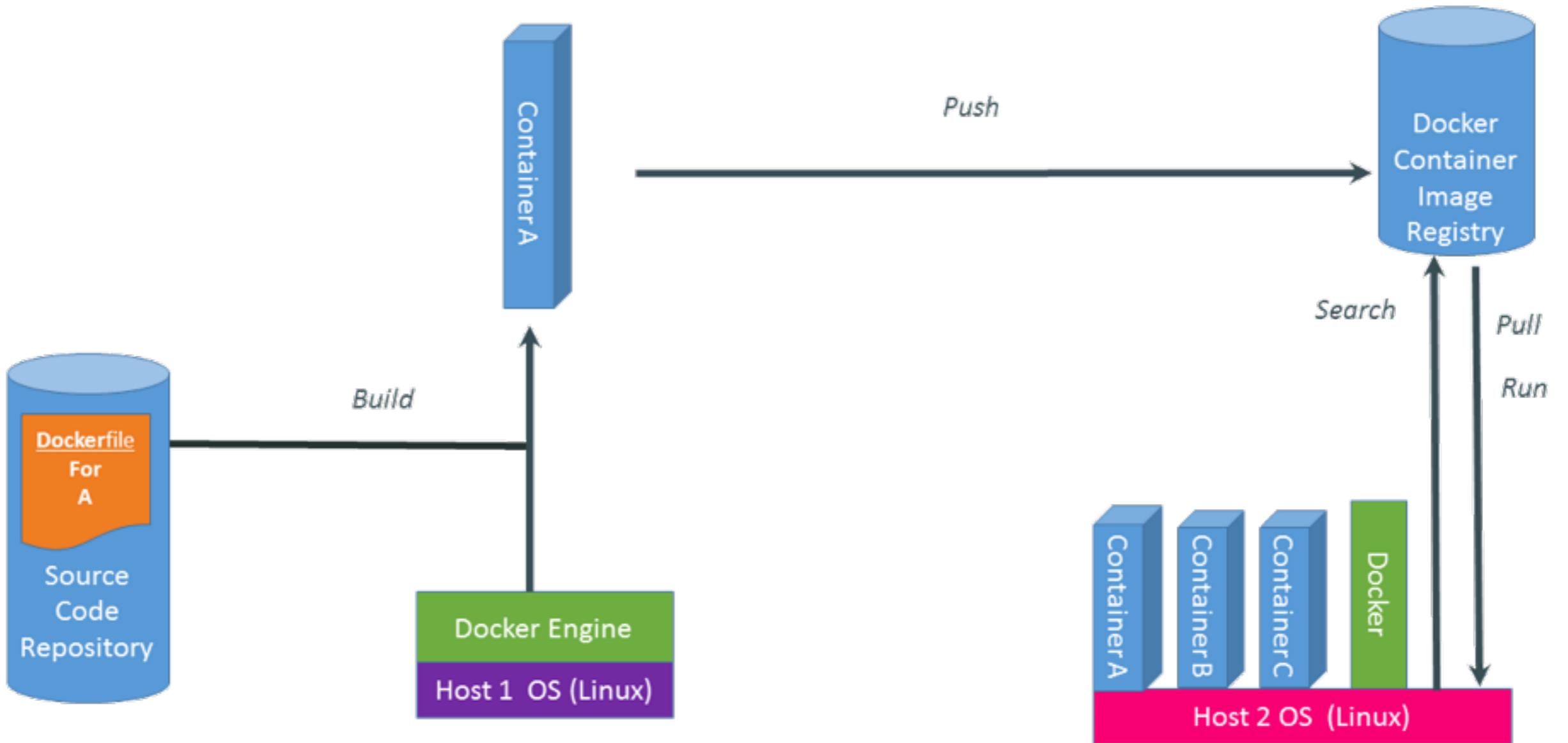
- open app in browser
- modify in a local and see the changes
- focus on software development

# Docker Compose

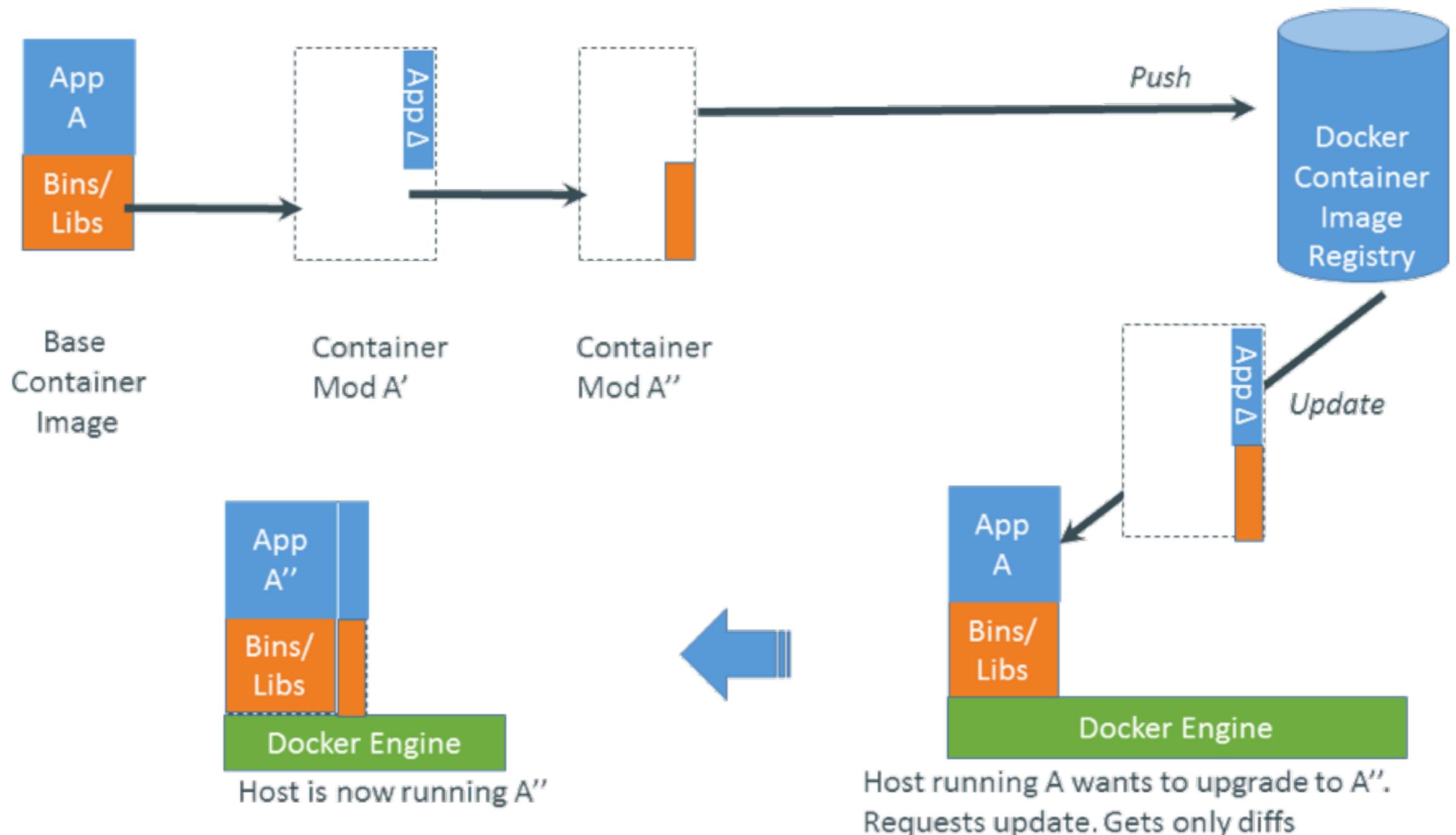
Makes it easy

# **Demo time**

# Docker System



# Change/Update



# Incremental Development





SPRINT3R

Siam Chamnankit Co., Ltd., Odd-e (Thailand) Co., Ltd. and Alliance



**Docker expert**  
**exp 3-5 years**



**Docker expert**  
**exp 3-5 years**

**Docker released 2013 !!**

# Resources

[https://github.com/up1/docker\\_101](https://github.com/up1/docker_101)