SELA

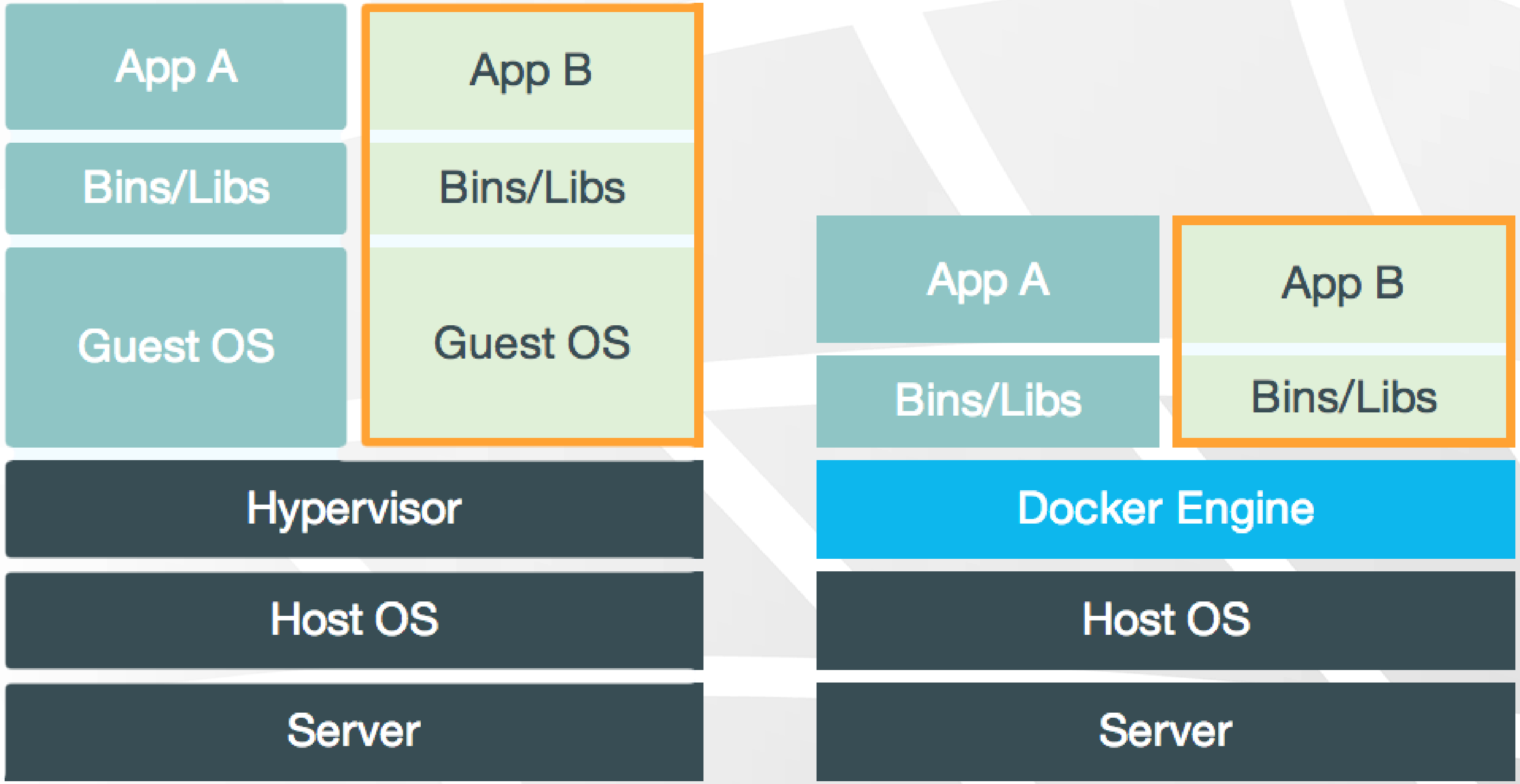
Module 10: Advanced Docker Overview

Docker Workshop

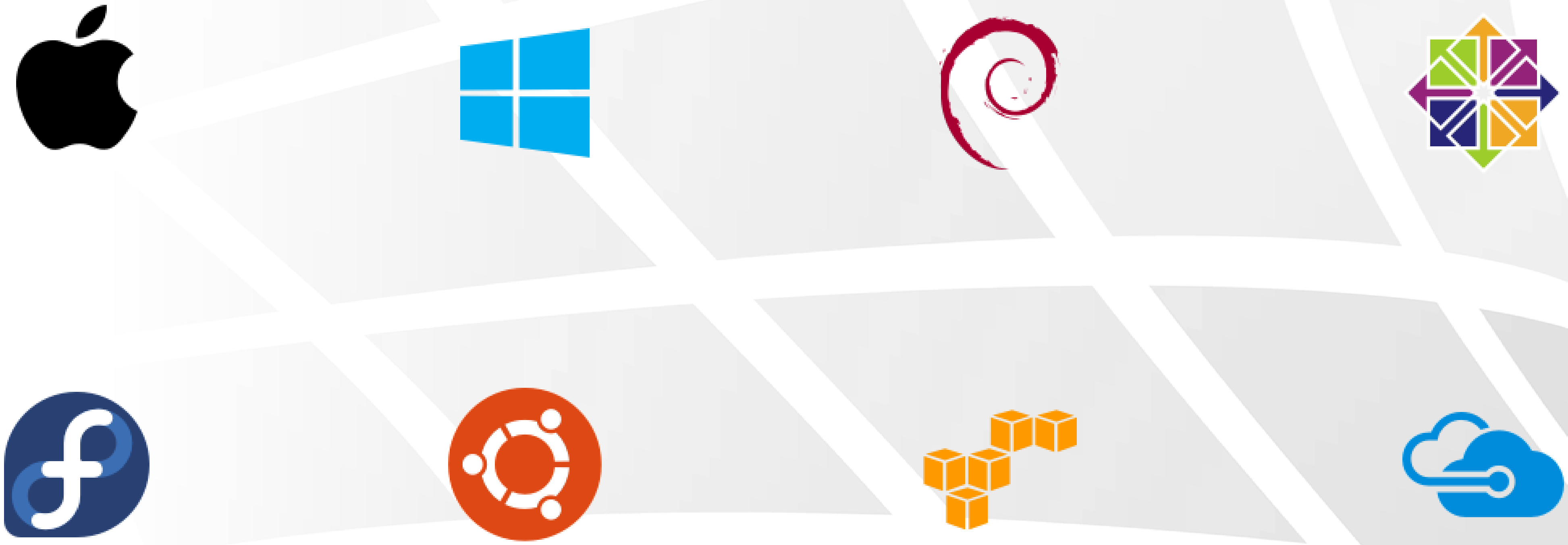
Agenda

- Basic Docker Overview
- Docker compose Introduction
- Container Orchestration Introduction

Why Docker?


App A
App B
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Hypervisor
Host OS
Server
App A
App B
Bins/Libs
Bins/Libs
Docker Engine
Host OS
Server

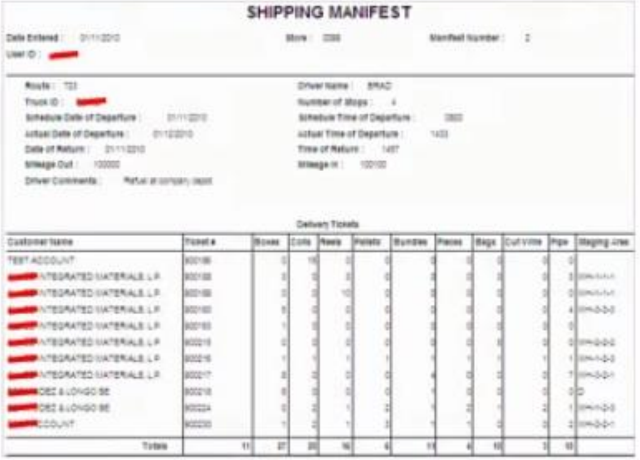
Docker Everywhere



Docker Architecture

(Shipping Yard)

(Manifests)



SHIPPING MANIFEST

Date Entered : 01/11/2010 Store : [illegible] Manifest Number : 2
User ID : [redacted]

Route : T03 Driver Name : BRAD
Truck ID : [redacted] Number of Stops : 4
Schedule Date of Departure : 01/11/2010 Schedule Time of Departure : 0800
Actual Date of Departure : 01/12/2010 Actual Time of Departure : 1403
Date of Return : 01/11/2010 Time of Return : 1457
Mileage Out : 100000 Mileage In : 100100
Driver Comments : Refuel at company depot

Delivery Tickets

Customer Name	Ticket #	Boxes	Coils	Reels	Pallets	Bundles	Pieces	Bags	Cut Wire	Pipe	Staging Area
TEST ACCOUNT	800198	0	10	0	0	0	0	0	0	0	
[redacted]NTEGRATED MATERIALS, L.P	800198	3	0	0	0	3	0	3	0	3	[illegible]
[redacted]NTEGRATED MATERIALS, L.P	800199	0	0	10	0	3	0	3	0	0	[illegible]
[redacted]NTEGRATED MATERIALS, L.P	800160	5	0	0	0	0	0	0	0	4	[illegible]
[redacted]NTEGRATED MATERIALS, L.P	800163	1	0	0	0	3	0	3	0	0	
[redacted]NTEGRATED MATERIALS, L.P	800215	0	0	0	0	0	0	6	0	0	[illegible]
[redacted]NTEGRATED MATERIALS, L.P	800216	1	1	1	1	1	1	1	1	1	[illegible]
[redacted]NTEGRATED MATERIALS, L.P	800217	8	2	0	0	4	0	2	0	7	[illegible]
[redacted]DEL & LONGO SE	800218	6	3	0	0	1	0	3	0	0	D
[redacted]DEL & LONGO SE	800224	0	2	1	2	1	2	1	2	1	[illegible]
[redacted]CCOUNT	800230	1	2	1	2	1	1	2	0	2	[illegible]
Totals	11	27	20	16	6	11	4	10	3	10	

(Shipping Containers)

- **Daemon:** The Docker environment to access the Docker services.
- **Images:** Read only templates used to create containers.
- **Containers:** Running instances of an image.

Building Containers

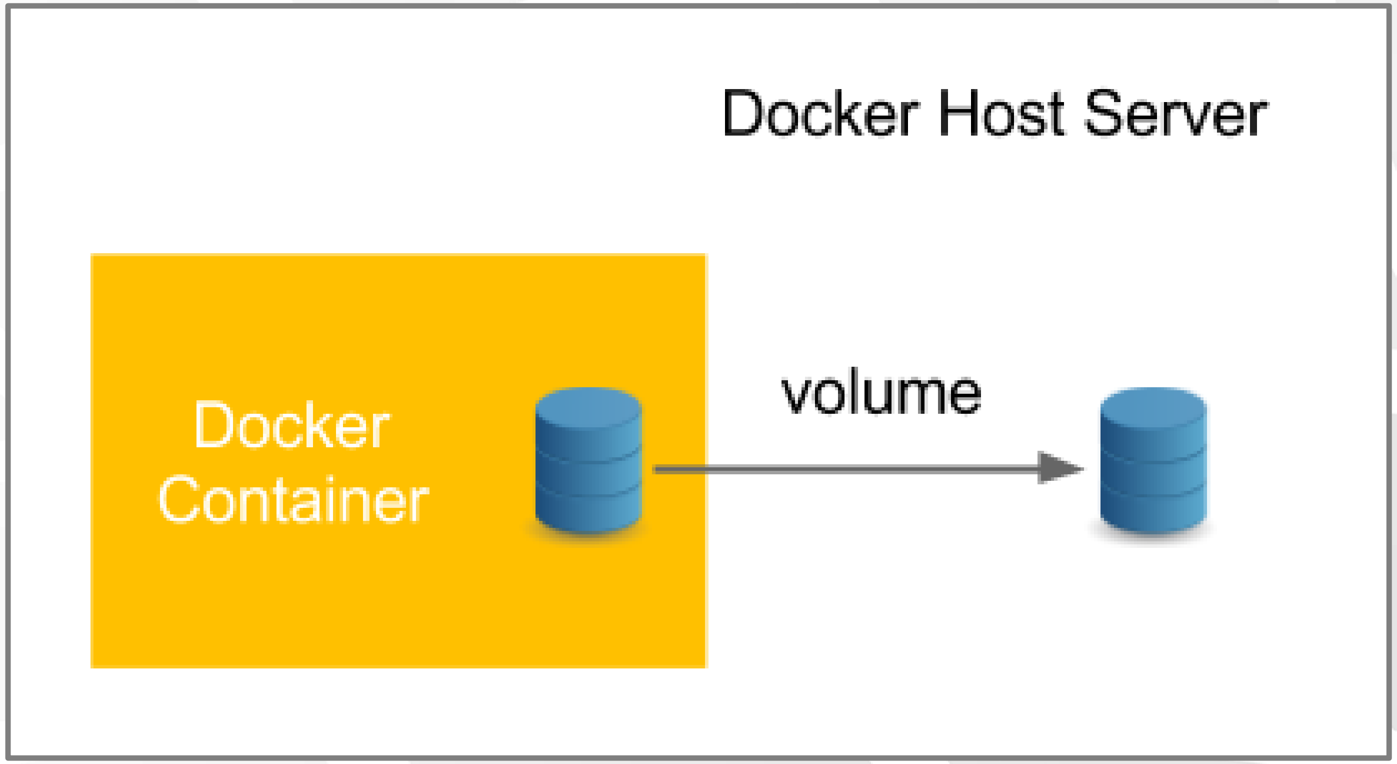
- FROM
- MANTAINER
- RUN
- CMD
- EXPOSE
- ENTRYPOINT
- ENV
- COPY
- ADD
- VOLUME

```
FROM ubuntu:14.04
RUN \
  apt-get update && \
  apt-get -y install apache2

VOLUME /myvol
ADD index.html /var/www/html/index.html

EXPOSE 80

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Managing Containers

- $ docker run
- $ docker ps
- $ docker images
- $ docker rm
- $ docker rmi
- $ docker attach
- $ docker exec
- $ docker save/load
- $ docker commit
- $ docker stop
- $docker kill
- $ docker start
- $ docker restart
- $ docker info
- $ docker top
- $ docker history
- $docker inspect
- $ docker logs

Docker Volumes



Docker Networks

- bridge
- host
- overlay
- macvlan
- none
- Network Plugins

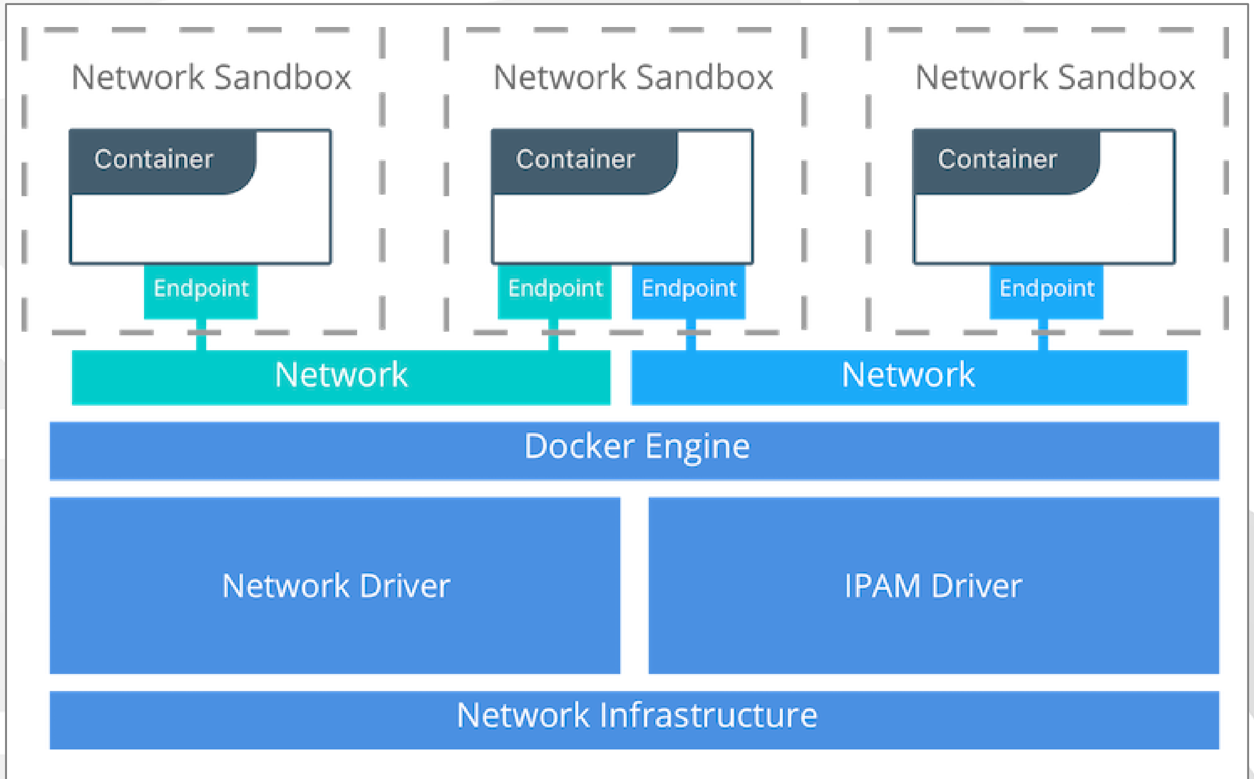


Fig 1: Architecture of Container Networking Model

Docker Registries

```
$ docker login -u <username> -p <password>
```

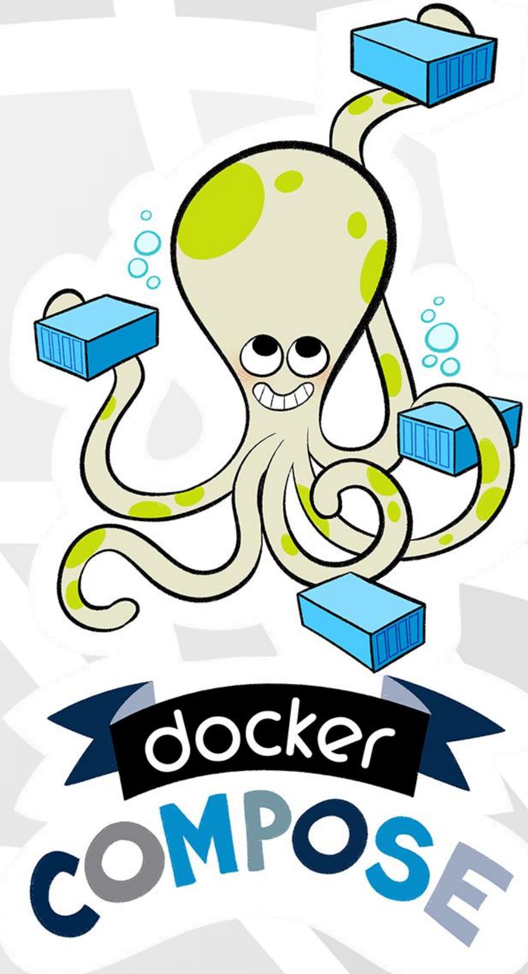
```
$ docker push <username>/<repo-name>:<tag>
```

```
$ docker pull <username>/<repo-name>:<tag>
```

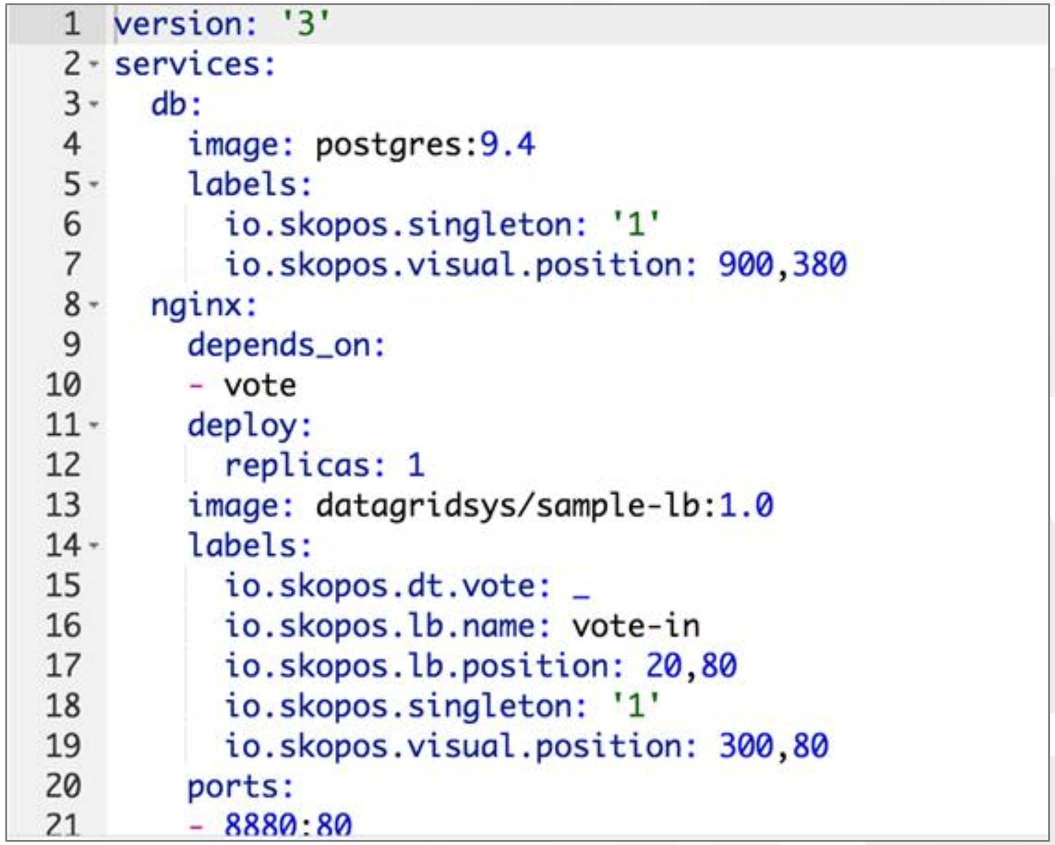
Containers Need Management

What is docker compose?

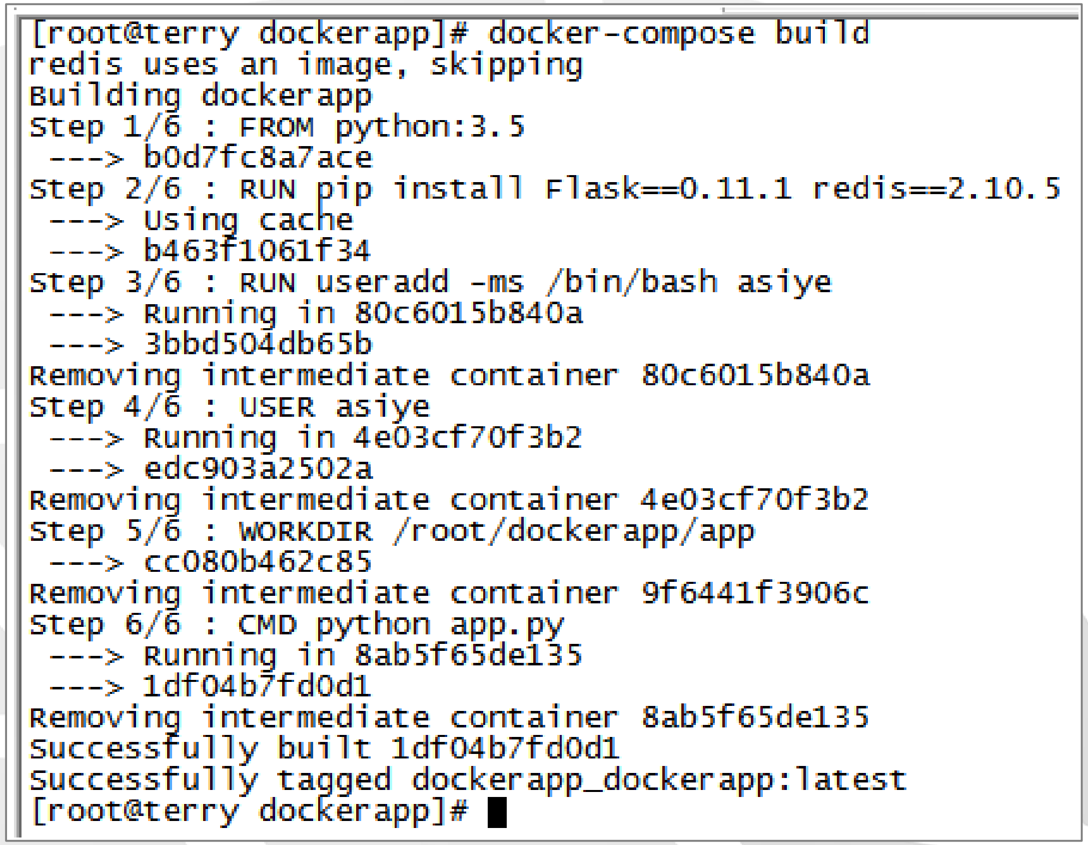
- Compose is a tool for defining and running complex applications with Docker.
- Define a multi-container application in a single file.
- Spin your application up in a single command.
- Docker-compose uses a YML file called “docker-compose.yml”



docker-compose.yml



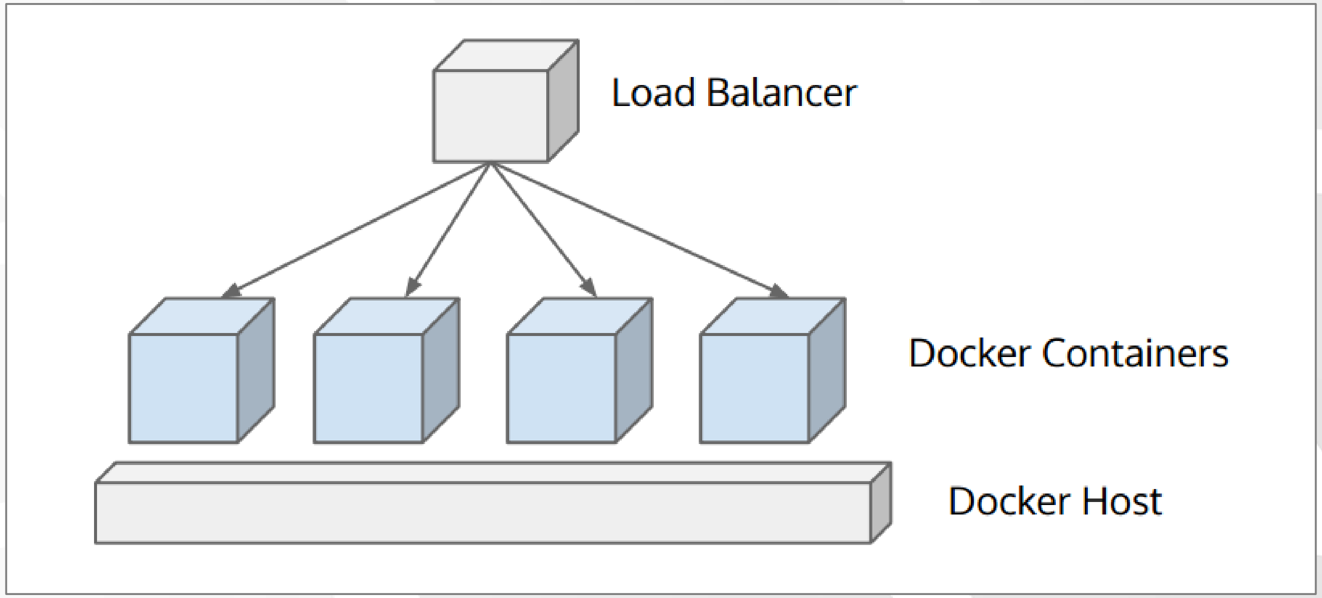
```yaml
version: '3'
services:
  db:
    image: postgres:9.4
    labels:
      io.skopos.singleton: '1'
      io.skopos.visual.position: 900,380
  nginx:
    depends_on:
    - vote
    deploy:
      replicas: 1
    image: datagridsys/sample-lb:1.0
    labels:
      io.skopos.dt.vote: _
      io.skopos.lb.name: vote-in
      io.skopos.lb.position: 20,80
      io.skopos.singleton: '1'
      io.skopos.visual.position: 300,80
    ports:
    - 8880:80
```



```
[root@terry dockerapp]# docker-compose build
redis uses an image, skipping
Building dockerapp
Step 1/6 : FROM python:3.5
 ---> b0d7fc8a7ace
Step 2/6 : RUN pip install Flask==0.11.1 redis==2.10.5
 ---> Using cache
 ---> b463f1061f34
Step 3/6 : RUN useradd -ms /bin/bash asiye
 ---> Running in 80c6015b840a
 ---> 3bbd504db65b
Removing intermediate container 80c6015b840a
Step 4/6 : USER asiye
 ---> Running in 4e03cf70f3b2
 ---> edc903a2502a
Removing intermediate container 4e03cf70f3b2
Step 5/6 : WORKDIR /root/dockerapp/app
 ---> cc080b462c85
Removing intermediate container 9f6441f3906c
Step 6/6 : CMD python app.py
 ---> Running in 8ab5f65de135
 ---> 1df04b7fd0d1
Removing intermediate container 8ab5f65de135
Successfully built 1df04b7fd0d1
Successfully tagged dockerapp_dockerapp:latest
[root@terry dockerapp]# █
```

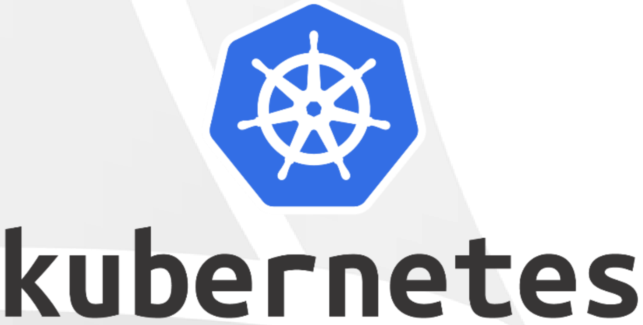
Problems with Standalone Docker

- Running a server cluster on a set of Docker containers, on a single Docker host is vulnerable to single point of failure!



Docker Orchestrators

- Scalability
- Service Discovery
- Networking
- Volume Management
- Monitoring and Logging
- High Availability
- Load Balancing
- Health Checks
- Rolling Upgrades



Questions

?