

# Docker Introduction

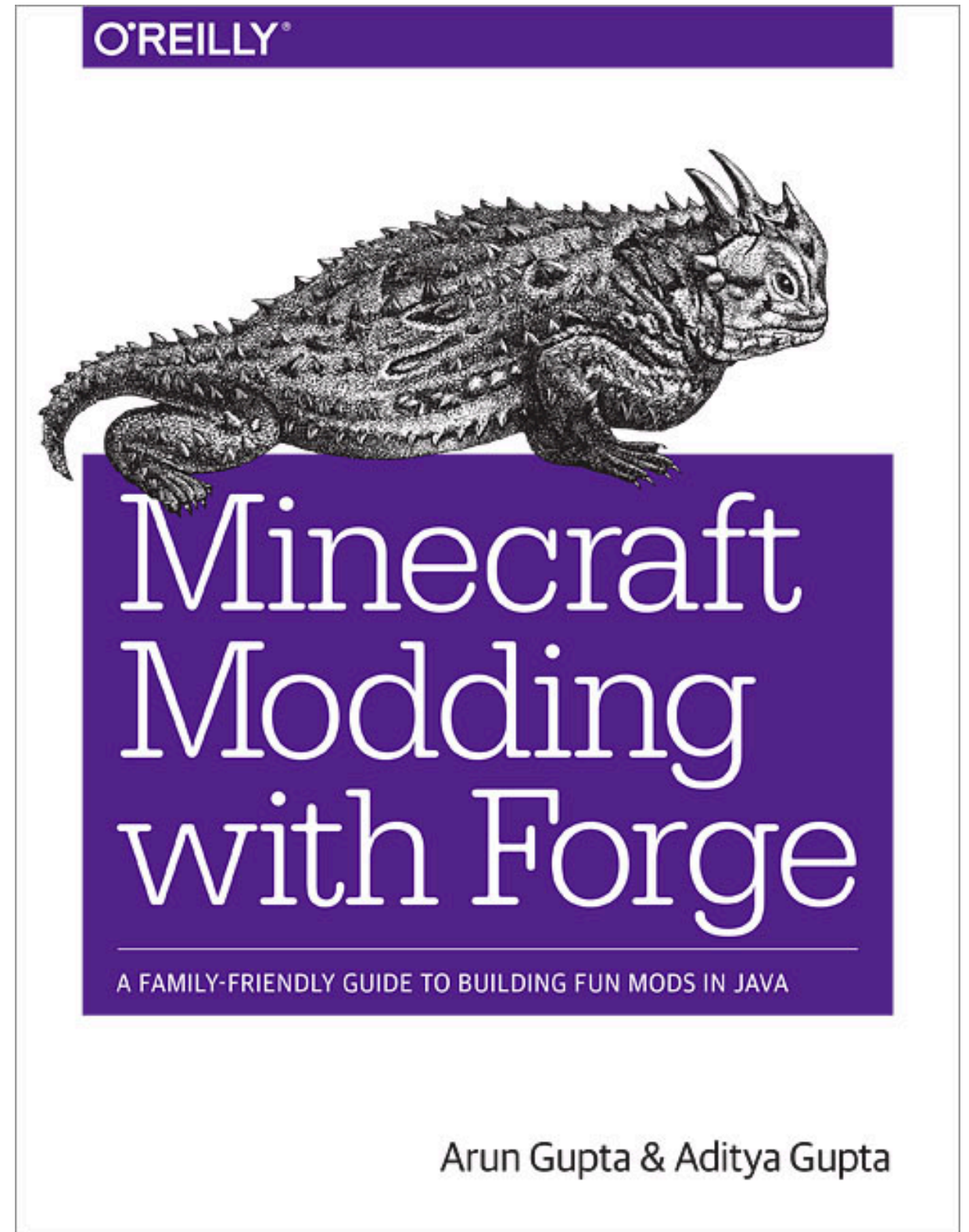


Arun Gupta

Vice President, Developer Advocacy

@arungupta, [blog.arungupta.me](http://blog.arungupta.me)

[arun@couchbase.com](mailto:arun@couchbase.com)



# What is Docker?

- Open source project and company



- Used to create containers for software applications
- Package Once Deploy Anywhere (PODA)

# Advantages of Containers

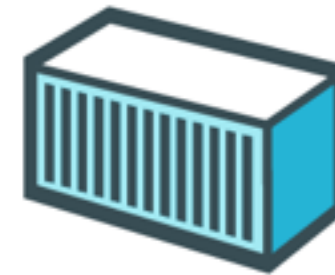
- Immutability
- Reproducibility
- Isolation
- Faster deployments
- Portability - “it works on my machine”
- Snapshotting
- Security sandbox
- Limit resource usage
- Simplified dependency
- Sharing





## Build

Develop an app using Docker containers with any language and any toolchain.



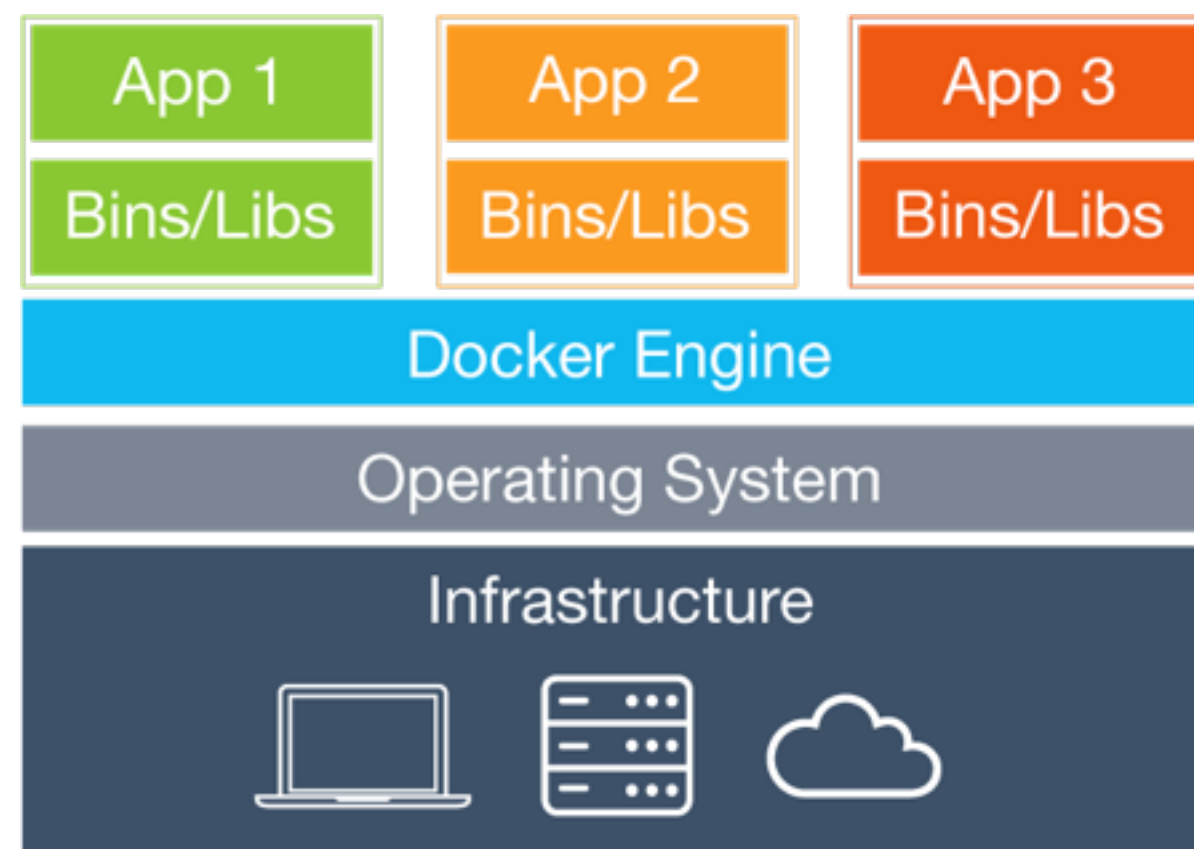
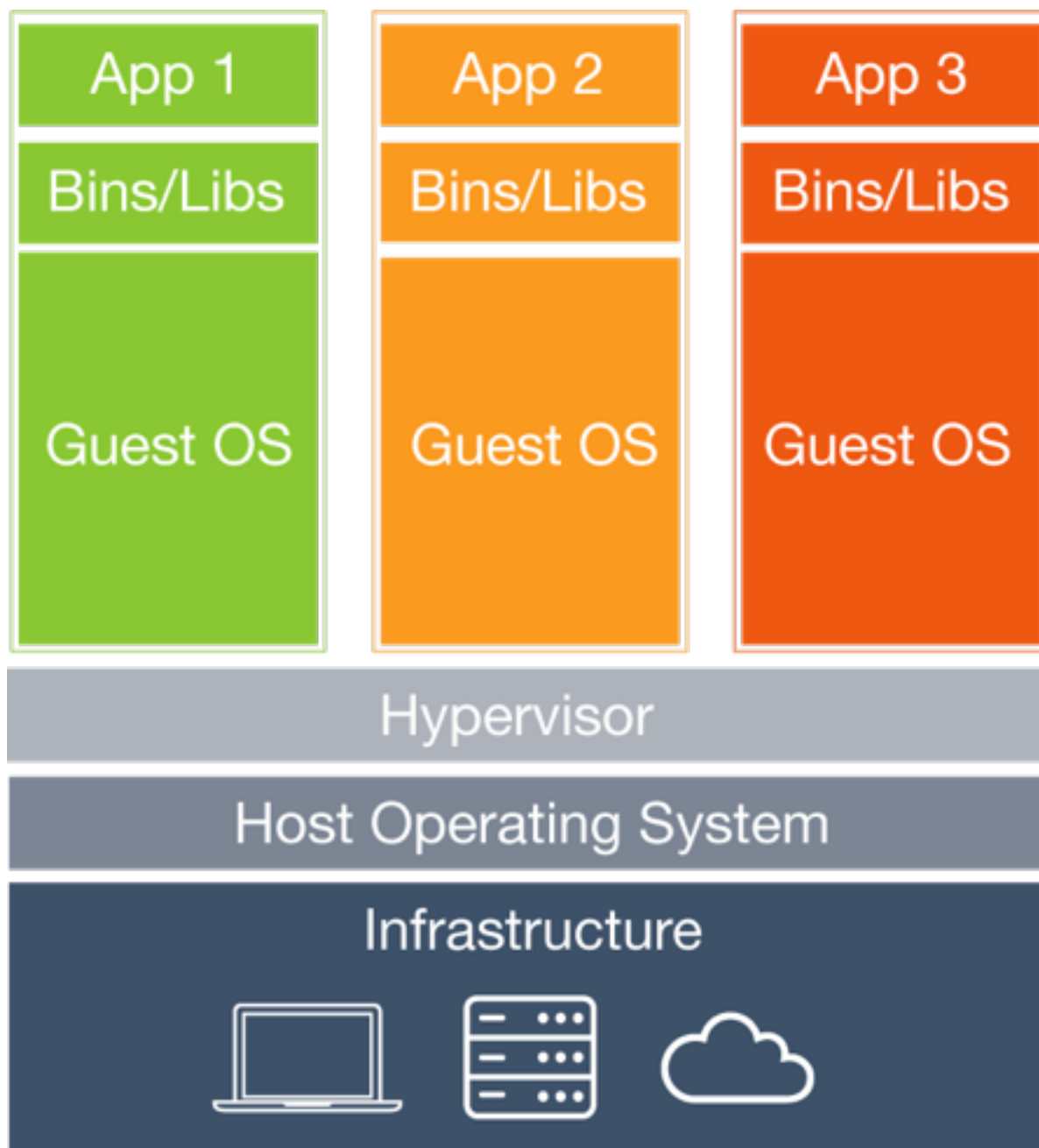
## Ship

Ship the “Dockerized” app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.



## Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.





Build

Develop an app using Docker containers with  
any language and any toolchain.

- Image defined in text-based **Dockerfile**
- List of commands to build the image

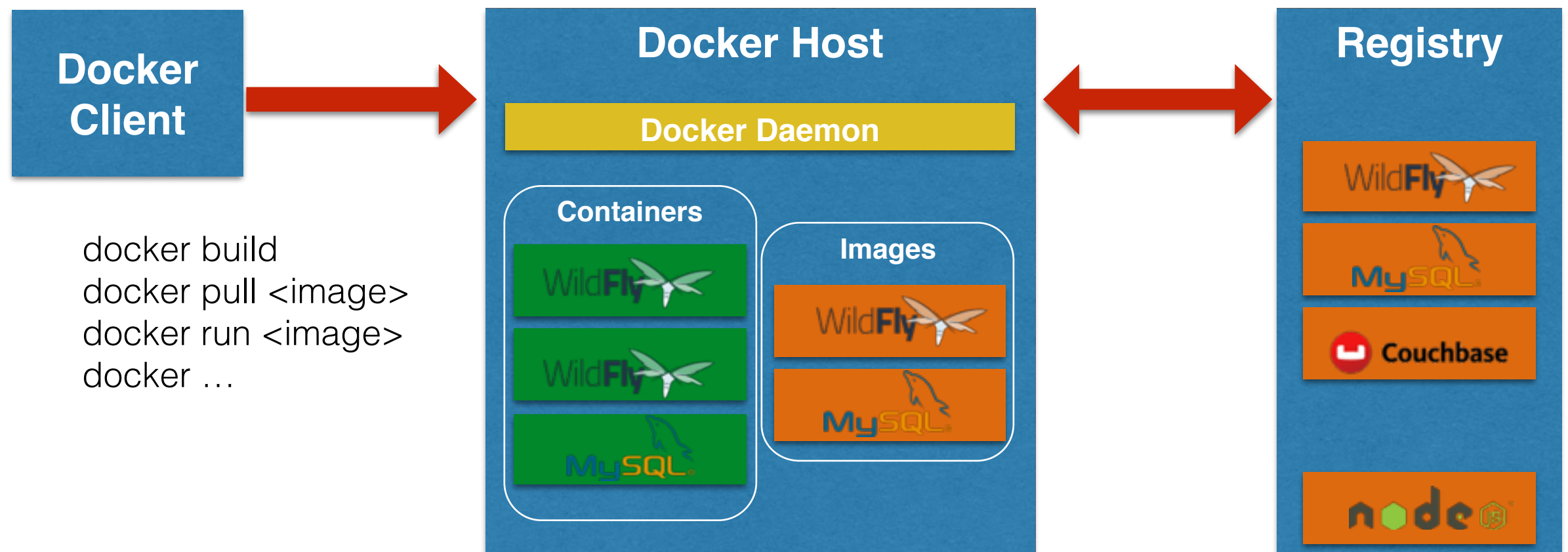
```
FROM fedora:latest
```

```
CMD echo "Hello world"
```

```
FROM jboss/wildfly
```

```
RUN curl -L https://github.com/javaee-samples/javaee7-hol/raw/master/solution/  
movieplex7-1.0-SNAPSHOT.war -o /opt/jboss/wildfly/standalone/deployments/  
movieplex7-1.0-SNAPSHOT.war
```

# Docker Workflow





Image

Image

Image

Base Image

Bootfs/Kernel

jboss/wildfly

jboss/base-jdk:8

jboss/base

centos:7

# Docker Machine



- Create Docker Host on computer or cloud provider

```
docker-machine create --driver=virtualbox  
myhost
```

- Configure Docker client to talk to host
  - Create and pull images
  - Start, stop, restart containers
  - Upgrade Docker
- Not recommended for production yet

# Docker Machine Providers

Microsoft Azure



SOFTLAYER®



# Docker Compose

- Defining and running multi-container applications
- Configuration defined in a single file
- Great for dev, staging, and CI
- Not recommended for production yet

# docker-compose.yml

**mysql**db:

image: mysql

environment:

MYSQL\_DATABASE: sample

MYSQL\_USER: mysql

MYSQL\_PASSWORD: mysql

MYSQL\_ROOT\_PASSWORD: supersecret

**mywildfly**:

image: arungupta/wildfly-mysql-javaee7

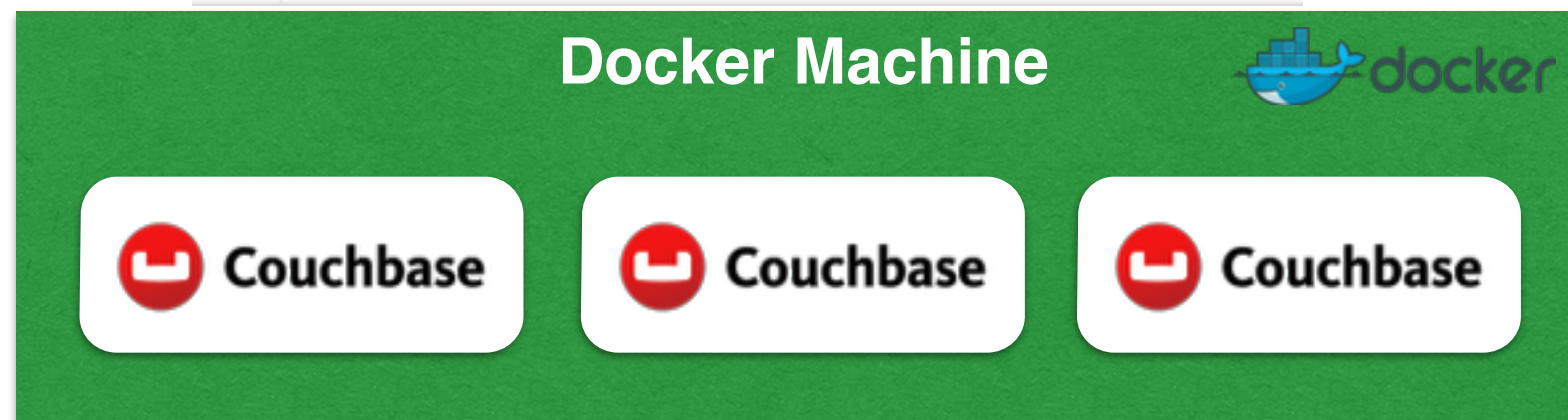
links:

- mysql



# Couchbase Cluster using Docker Compose

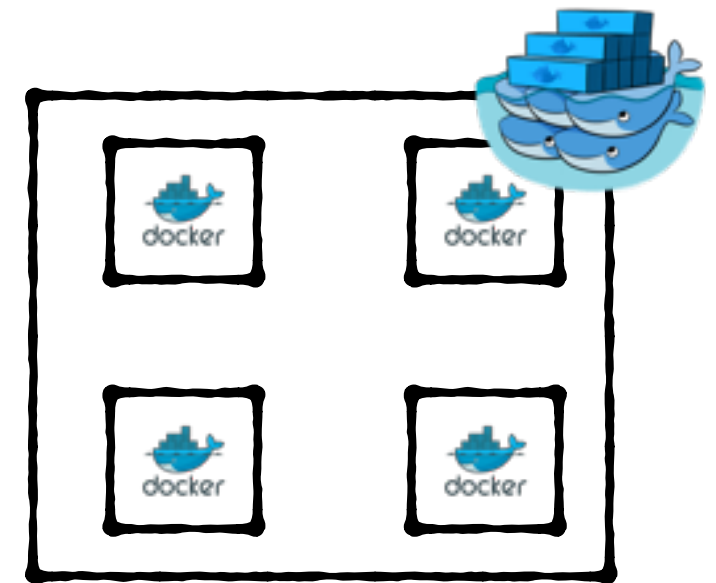
```
1 couchbase1:
2   image: couchbase/server
3   volumes:
4     - ~/couchbase/node1:/opt/couchbase/var
5 couchbase2:
6   image: couchbase/server
7   volumes:
8     - ~/couchbase/node2:/opt/couchbase/var
9 couchbase3:
10  image: couchbase/server
11  volumes:
12    - ~/couchbase/node3:/opt/couchbase/var
13  ports:
14    - 8091:8091
15    - 8092:8092
16    - 8093:8093
17    - 11210:11210
```



# Docker Swarm



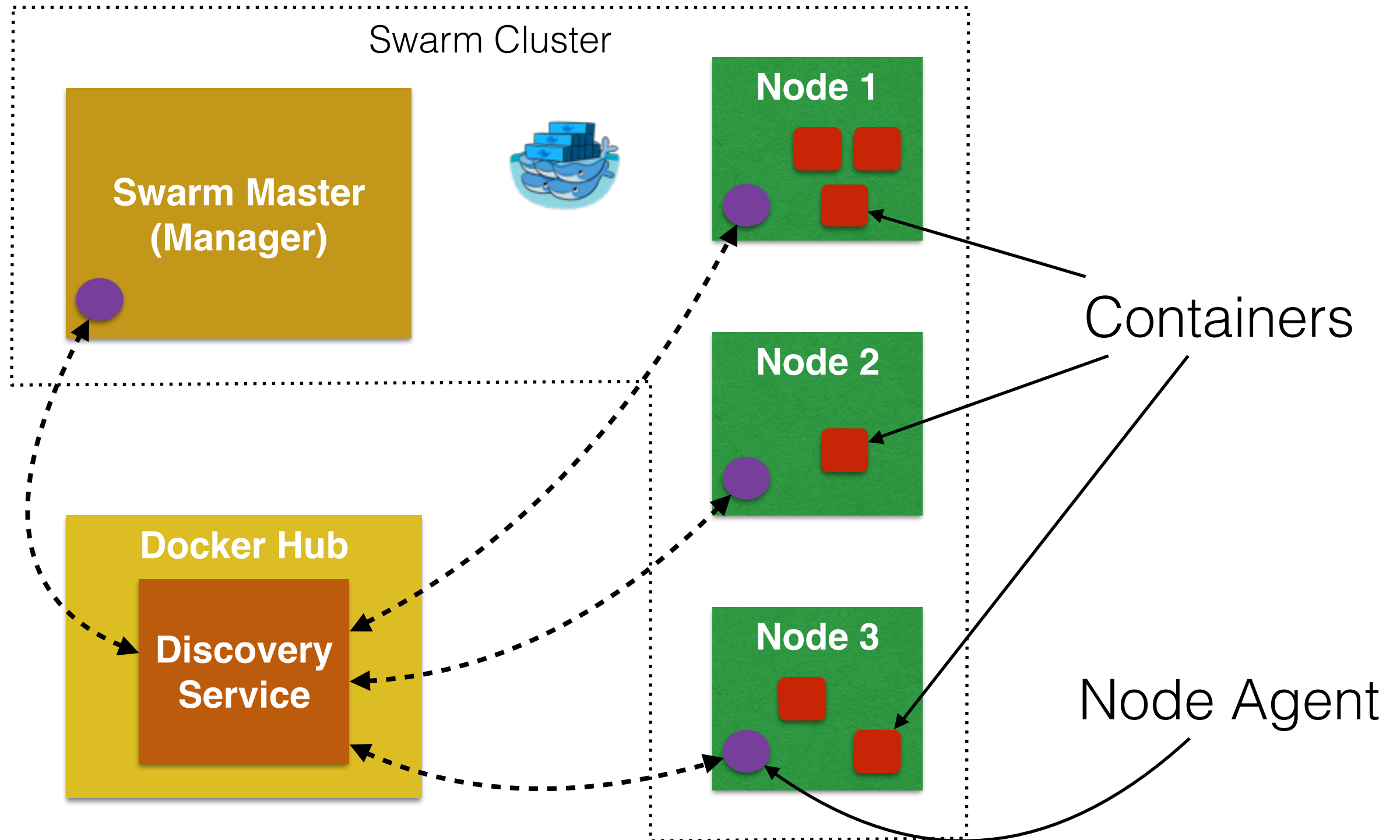
- Native clustering for Docker
- Provides a unified interface to a pool of Docker hosts
- Fully integrated with Machine
- Serves the standard Docker API
- Partially integrated with Compose
- 1.0.0 - Ready for production



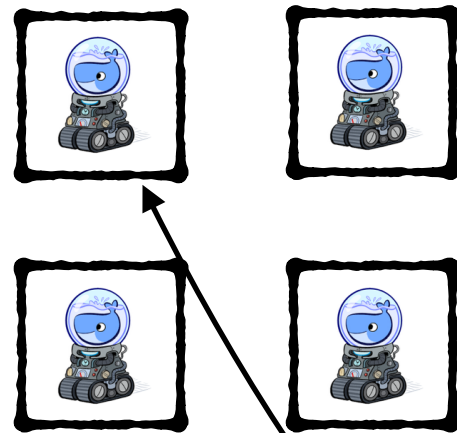
# Docker Swarm

- Simple scheduling backends included, API for pluggable backends (e.g. Mesos) coming
  - Based on CPU (`-c`), RAM (`-m`), number of containers
  - `spread` (default): node with least number of running containers
  - `binpack`: node with most number of running containers
  - `random`: mostly for debugging
  - Specified using `--strategy`

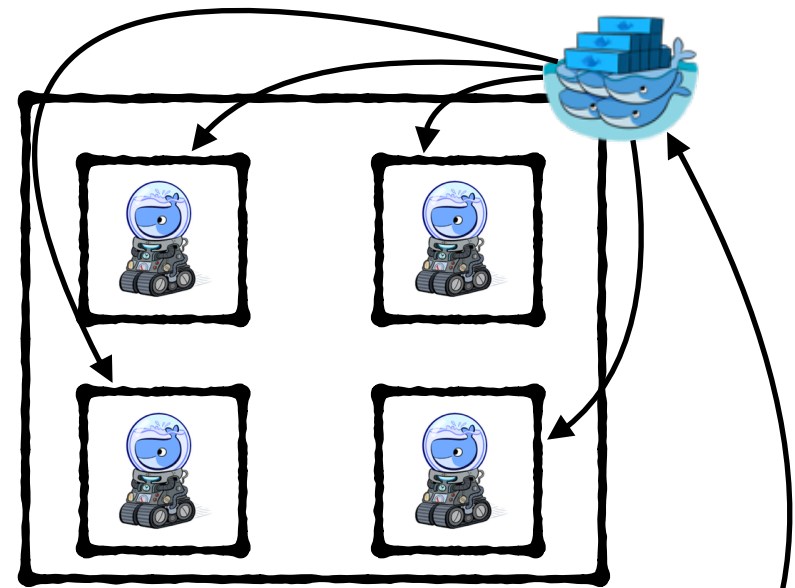
# Docker Swarm



# Machine

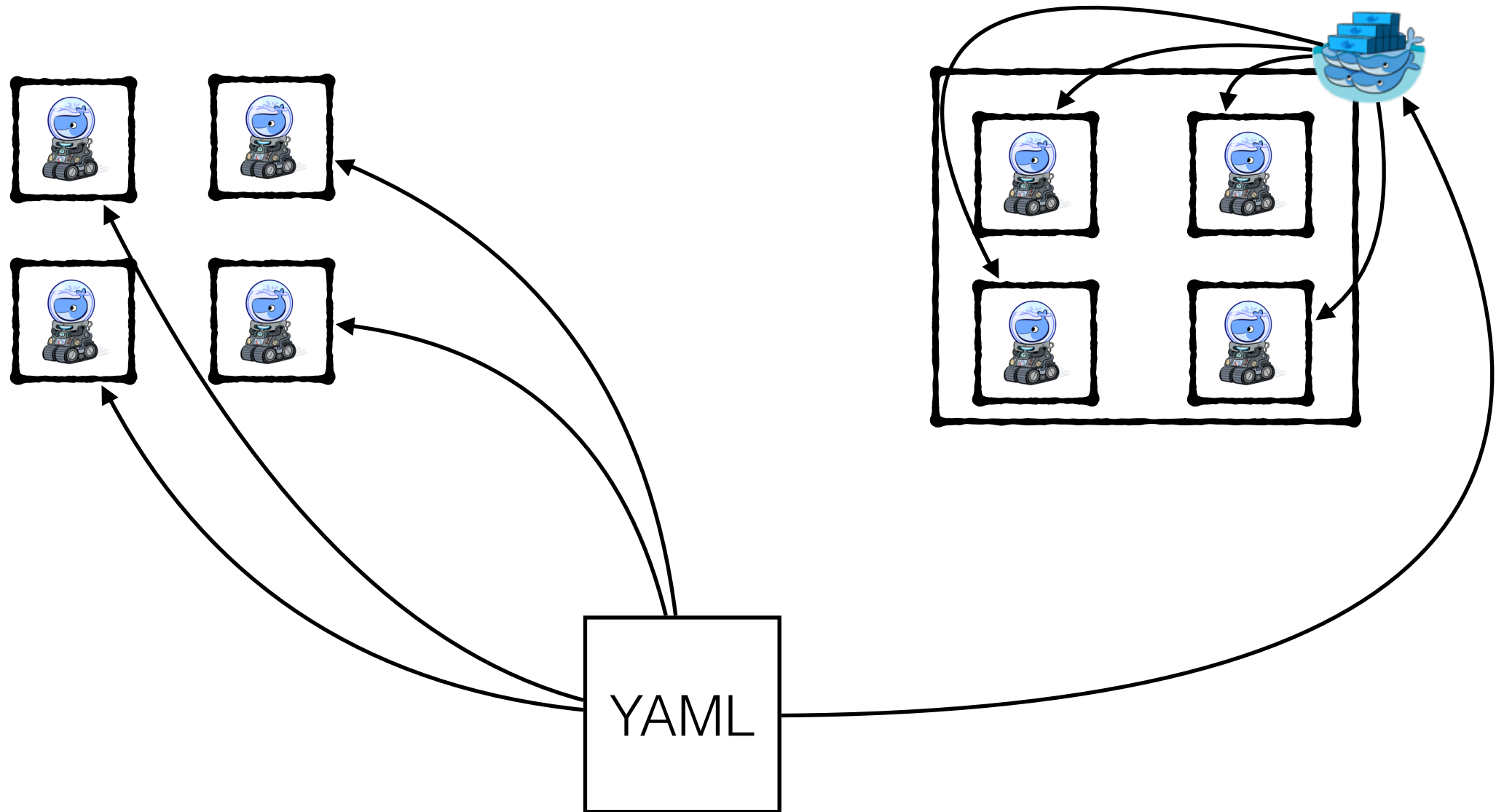


# Swarm



YAML

# Compose





# Docker Toolbox

- Docker Client 1.9.0
- Docker Machine 0.5.0
- Docker Compose 1.5.0 (~~Mac only~~)
- Docker Kitematic 0.9.3
- Boot2Docker ISO 1.9.0
- Virtualbox 5.0.8



# Persistent Storage

- Data volumes - used to persist data independent of container's lifecycle
- Multiple plugins: Flocker, Ceph, . . .
- `docker run -d -P -v /opt/couchbase/var couchbase/server`
- `docker run -d -P -v ~/couchbase:/opt/couchbase/var couchbase/server`

# Multi-host Networking

- Create virtual networks and attach containers
  - Software defined networking
- Works with Swarm and Compose
- Pluggable: Calico, Cisco, Weave, . . .

# Networking vs Links

- Connect containers to each other across different physical or virtual hosts
- Containers can be easily stopped, started and restarted w/o disrupting connection to other containers
- Can be created in any order

# docker network

`docker network --help`

Usage: `docker network [OPTIONS] COMMAND [OPTIONS]`

## Commands:

<code>disconnect</code>	Disconnect container from a network
<code>inspect</code>	Display detailed network information
<code>ls</code>	List all networks
<code>rm</code>	Remove a network
<code>create</code>	Create a network
<code>connect</code>	Connect container to a network

Run '`docker network COMMAND --help`' for more information on a command.

<code>--help=false</code>	Print usage
---------------------------	-------------



# Continuous Delivery using Docker and Jenkins Workflow

