# Docker Recipes for Java Developers

Arun Gupta, @arungupta

# What is Docker?

- Open source project and company
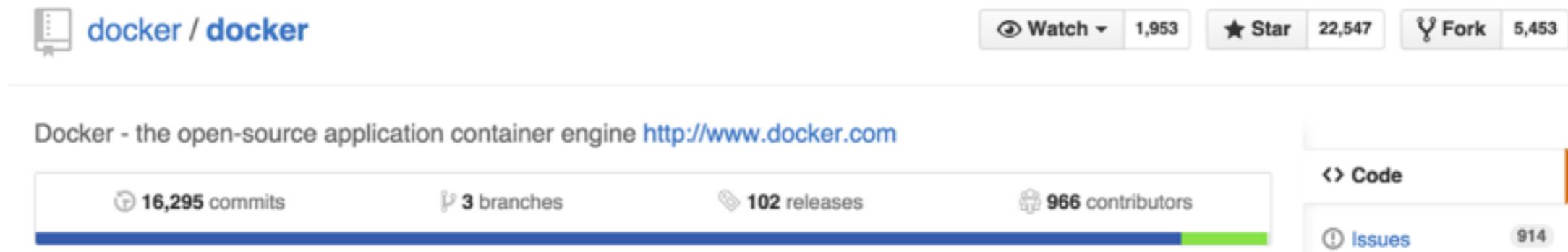


- Used to create containers for software applications

- Package Once Deploy Anywhere (PODA)

# Docker Contributors by employers/hackers

<> **gistfile1.txt**

```
 1    Top changeset contributors by employer
 2    (Unknown)                 4520 (47.9%)
 3    "Docker"                  3821 (40.5%)
 4    "Red Hat"                  685 (7.3%)
 5    "IBM"                      232 (2.5%)
 6    "Google"                   119 (1.3%)
 7    "Cisco"                     49 (0.5%)
 8    "Amadeus"                    4 (0.0%)
 9    "VMWare"                     2 (0.0%)
10    "CoreOS"                     1 (0.0%)
11
```

https://gist.github.com/arun-gupta/7d5a373099ff831d7213

http://www.infoworld.com/article/2925484/application-virtualization/look-whos-helping-build-docker-besides-docker-itself.html

**red**hat

## Build

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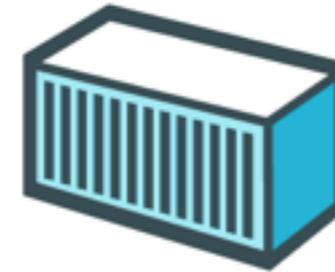
## Ship

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.

redhat
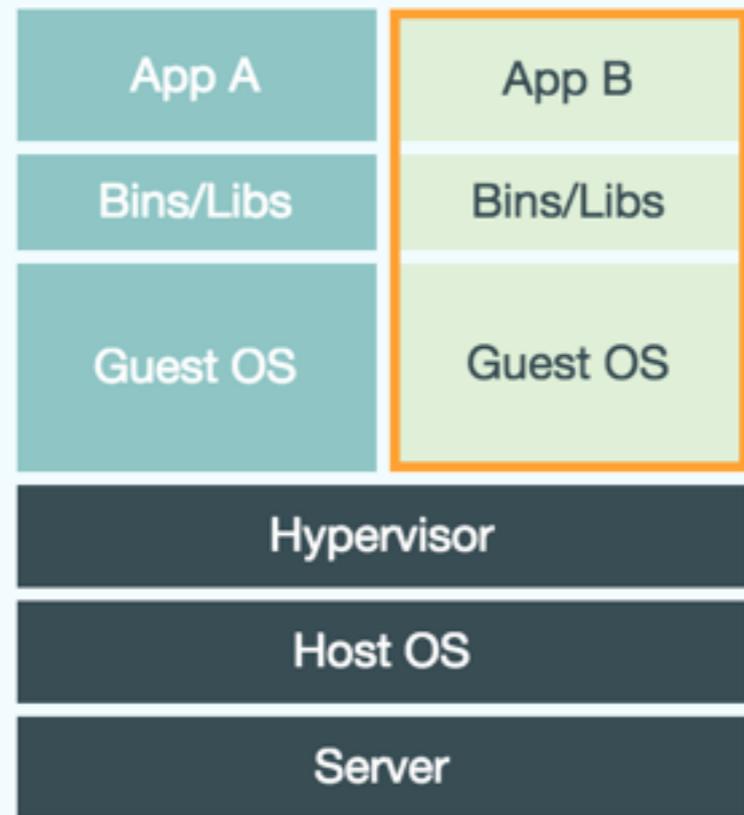
## Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.
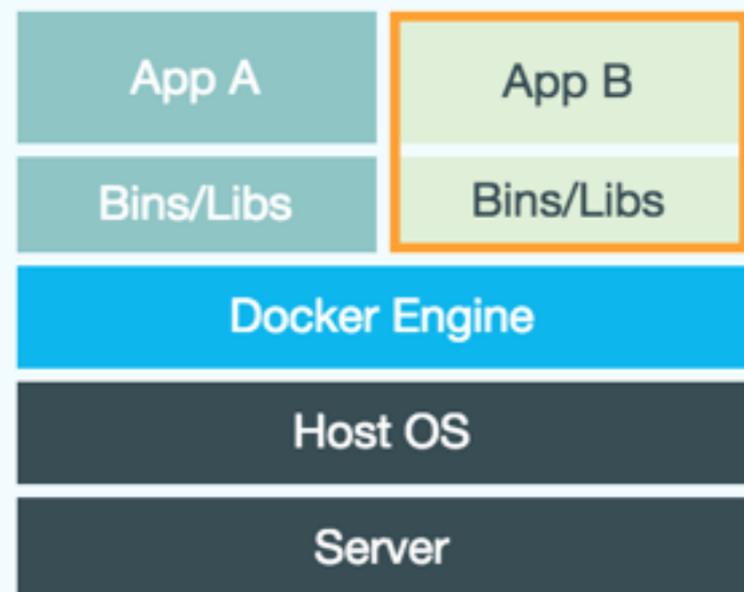
## Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

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

redhat

# Docker Workflow

**Docker Client**

docker run <image>
docker ...

**Docker Host**

**Daemon**

**Container 1**

**Container 2**

**Container O**

**Image 1**

**Image 2**

**Image 3**

**Image N**

**Docker Hub**

**Image 1**

**Image 2**
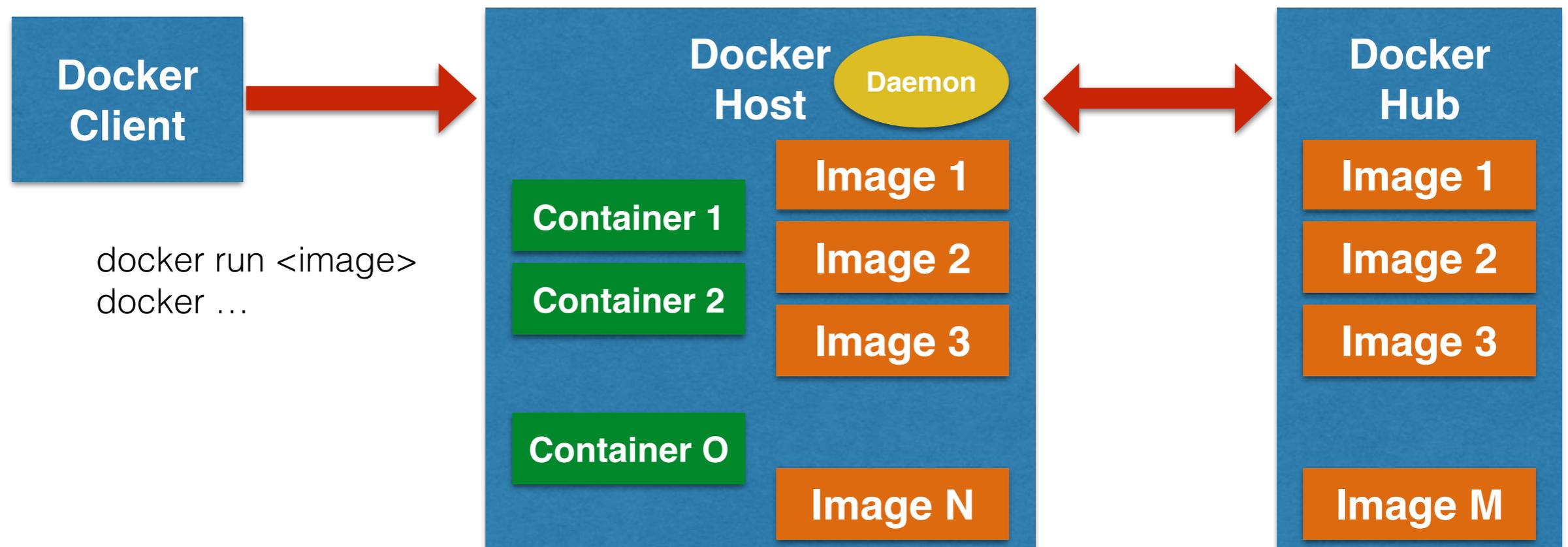
**Image 3**

**Image M**

# 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
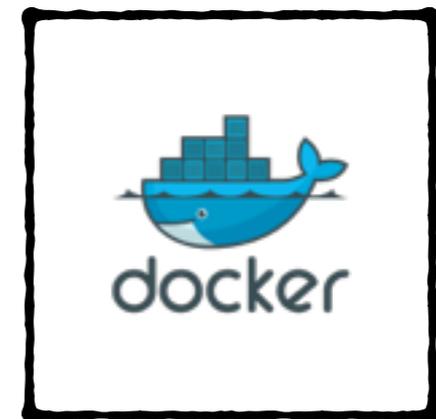
# Docker Compose

- 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

```yaml
mysqldb:
  image: mysql
  environment:
    MYSQL_DATABASE: sample
    MYSQL_USER: mysql
    MYSQL_PASSWORD: mysql
    MYSQL_ROOT_PASSWORD: supersecret
mywildfly:
  image: arungupta/wildfly-mysql-javaee7
  links:
    - mysqldb:db
```

# 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

- Not recommended for production yet

Swarm Cluster

Swarm Master (Manager)

Docker Hub

Discovery Service

Node 1

Node 2

Node 3

Containers

Node Agent

**Machine**

**Swarm**

YAML
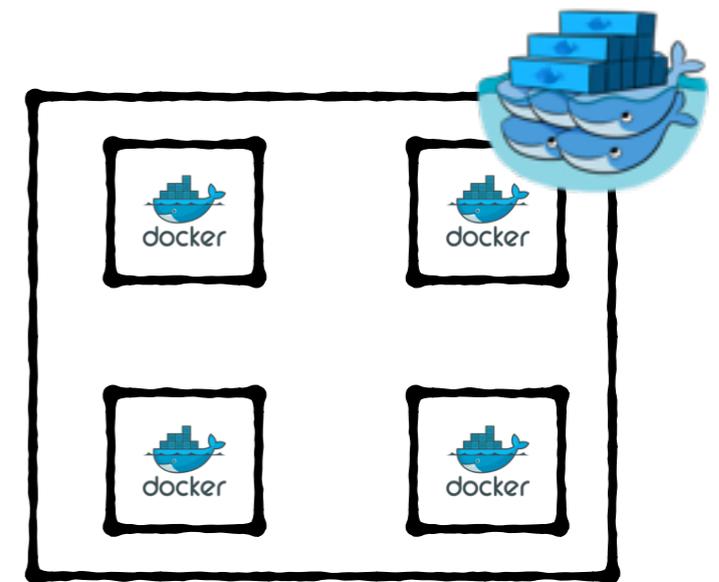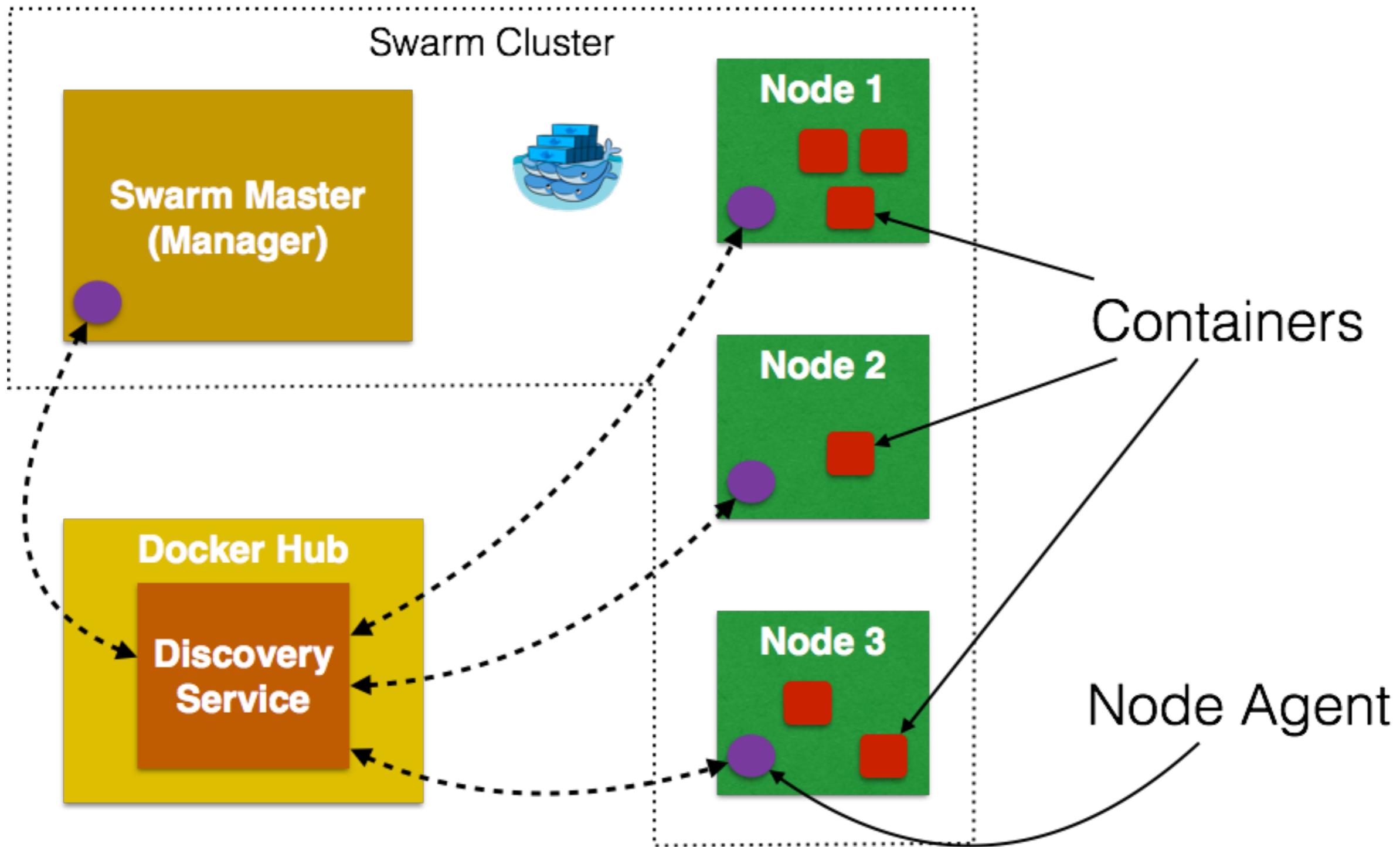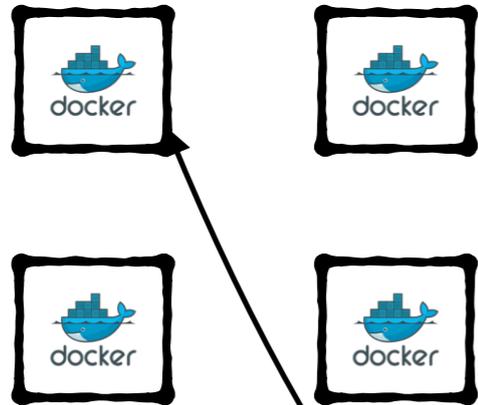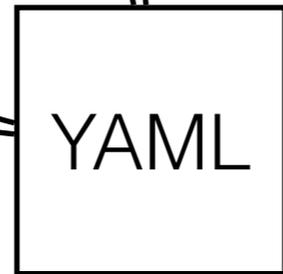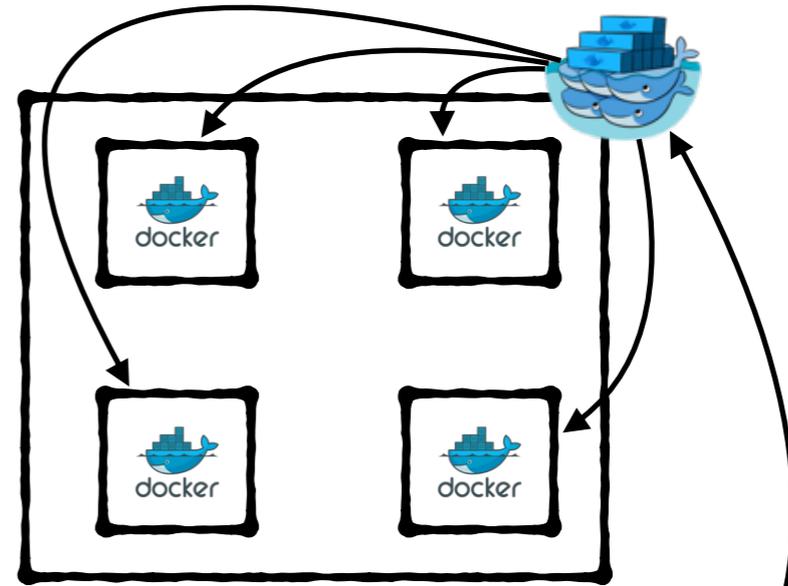
**Compose**
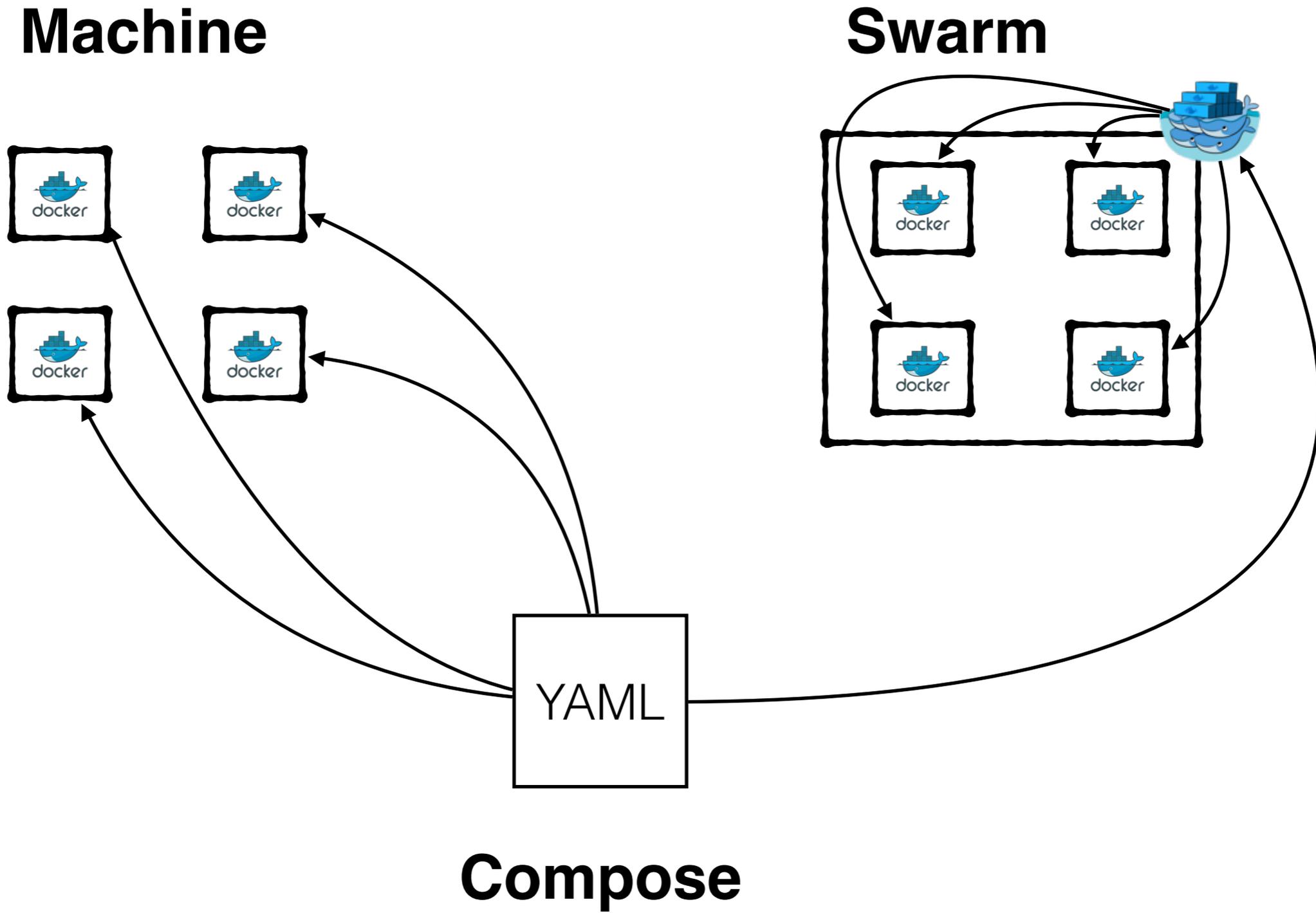
# Advantages of Containers

- Immutability

- Reproducibility

- Isolation

- Faster deployments

- Portability - "it works on my machine"

- Snapshotting

- Security sandbox

- Limit resource usage

- Simplified dependency

- Sharing

redhat

# Kubernetes

- Open source orchestration system for Docker containers

- Provide declarative primitives for the "desired state"

  - Self-healing

  - Auto-restarting

  - Schedule across hosts

  - Replicating

# Concepts

- **Pods**: collocated group of Docker containers that share an IP and storage volume

- **Service**: Single, stable name for a set of pods, also acts as LB

- **Replication Controller**: manages the lifecycle of pods and ensures specified number are running

- **Label**: used to organize and select group of objects



C1   C2    C3
Pod 1      Pod 2
Docker

Service "web"

port 8080      port 8080

JBoss      JBoss
Pod 1      Pod 2
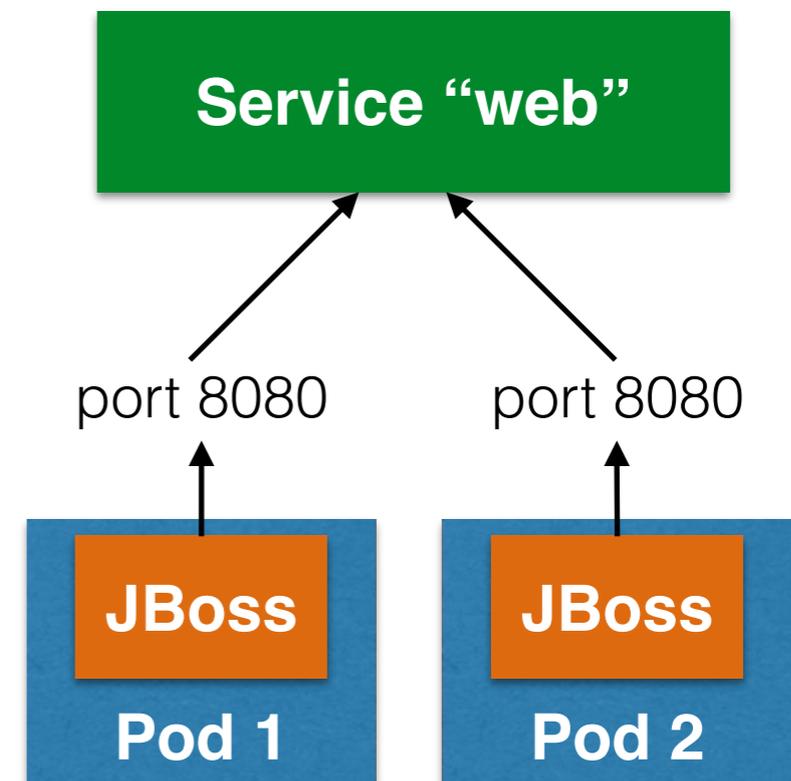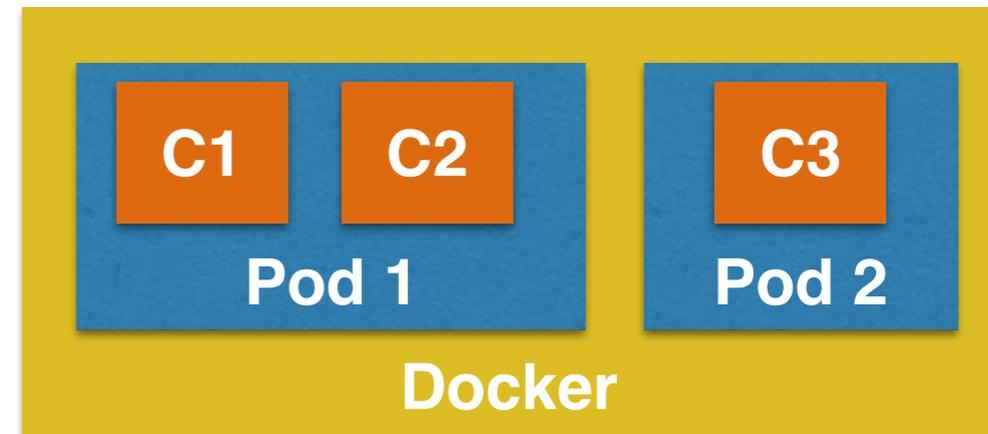
## Collect statistics about who wrote kubernetes

gitdm (git data mine) is a tool written by Jonathan Corbet at LWN which he uses to do his 'who wrote the kernel' articles. I spent a couple of minutes to run it against kube. Some intersting results...

```
...

Developers with the most changesets
Brendan Burns            643 (10.3%)
Daniel Smith             485 (7.8%)
Clayton Coleman          453 (7.3%)
Tim Hockin               312 (5.0%)
derekwaynecarr           209 (3.4%)

Developers with the most changed lines
Brendan Burns            455888 (18.0%)
Nan Monnand Deng         379610 (15.0%)
Yifan Gu                 219191 (8.7%)
Patrick                  169265 (6.7%)
nikhiljindal             124915 (4.9%)

Top changeset contributors by employer
"Google"                 3563 (58.3%)
"Red Hat"                1252 (20.5%)
(Unknown)                899 (14.7%)
"CoreOS"                 144 (2.4%)
"FathomDB"               142 (2.3%)
```



16