Docker Introduction

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

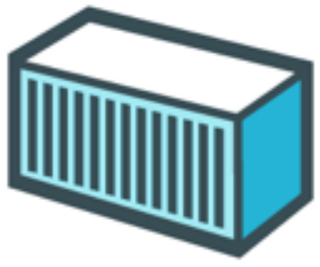
```
Top changeset contributors by employer
(Unknown)                       4520 (47.9%)
"Docker"                        3821 (40.5%)
"Red Hat"                        685 (7.3%)
"IBM"                            232 (2.5%)
"Google"                         119 (1.3%)
"Cisco"                           49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

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

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

Build

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

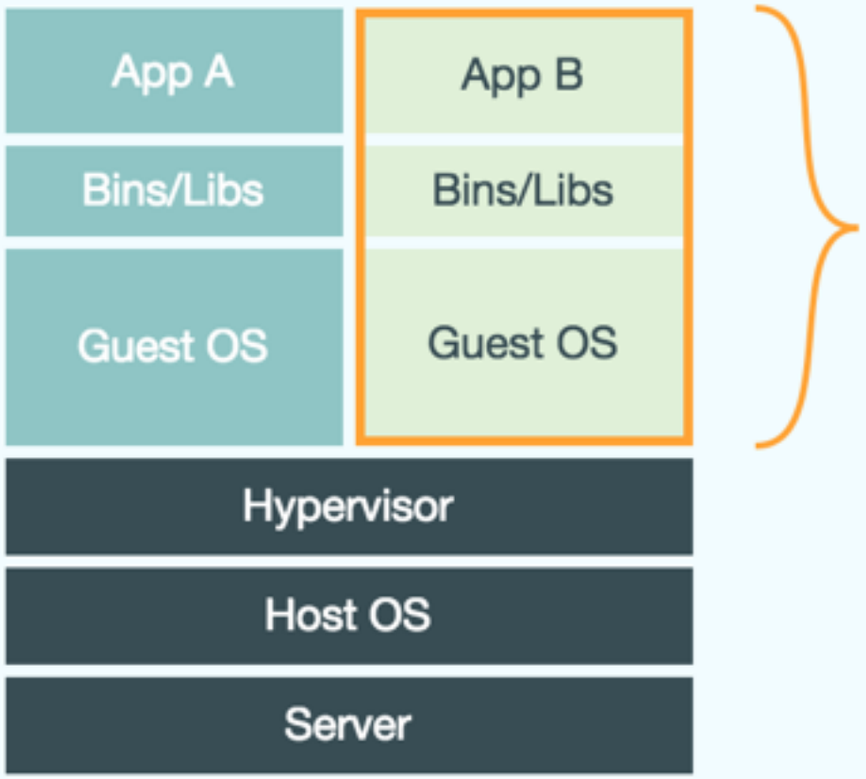


Ship

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

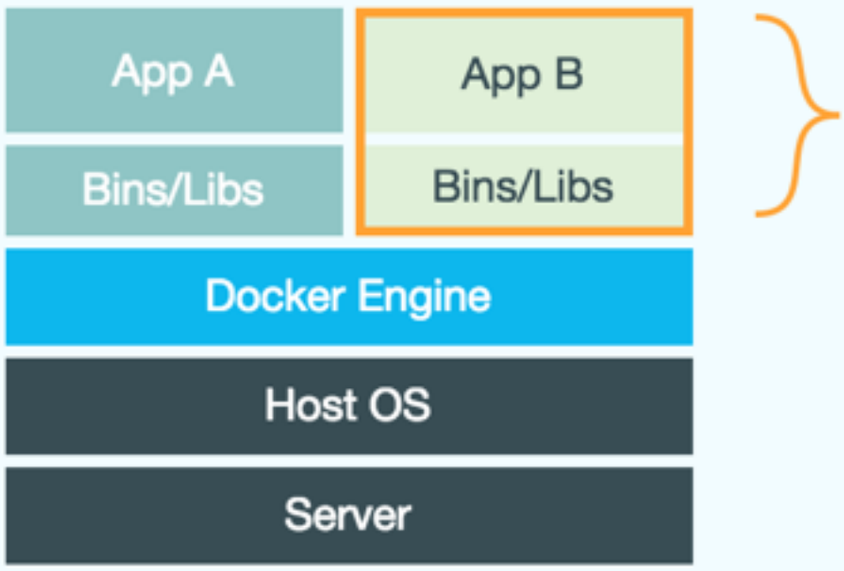
Run

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



Virtual Machines

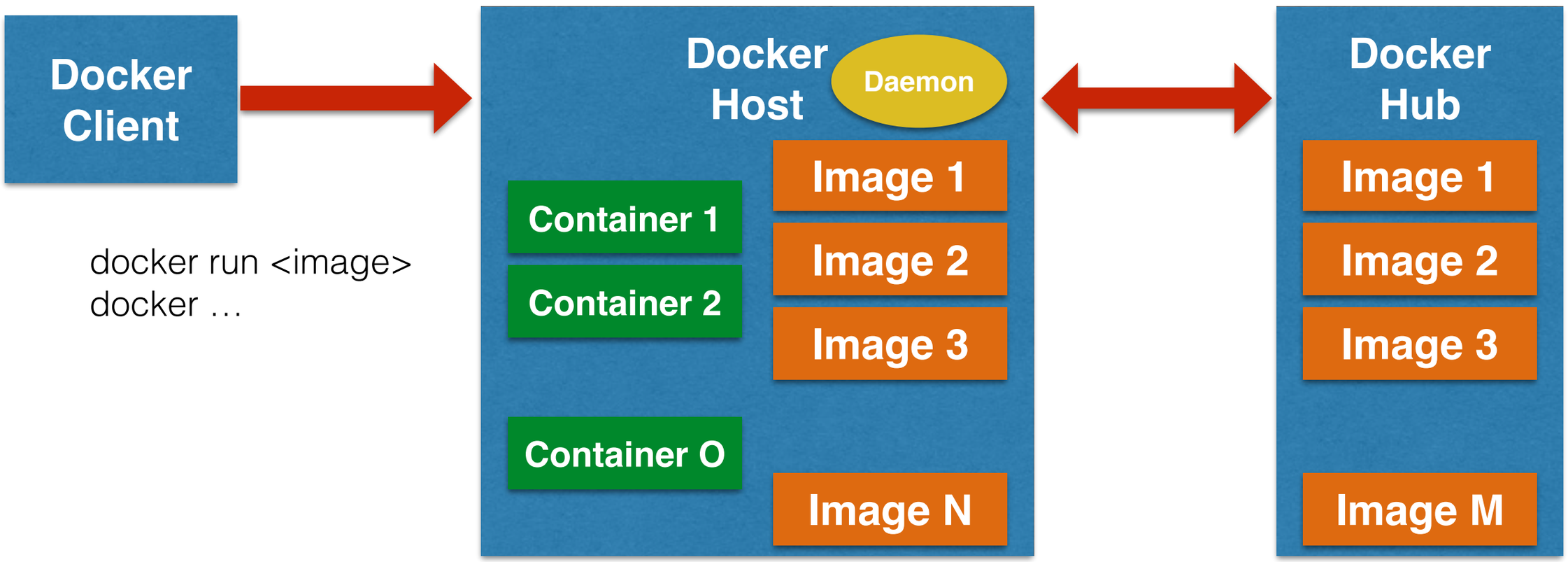
Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



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.

Docker Workflow



docker run <image>
docker …

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

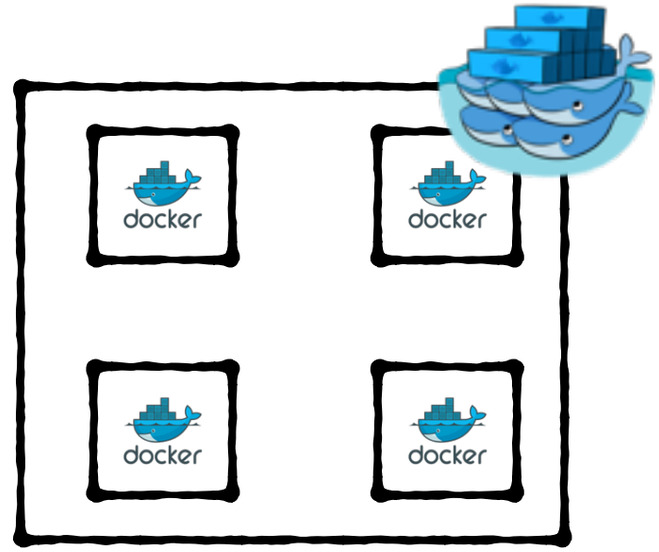
- Defining and running multi-container applications
- Configuration defined in a single file
- Great for dev, staging, and CI
- Not recommended for production yet

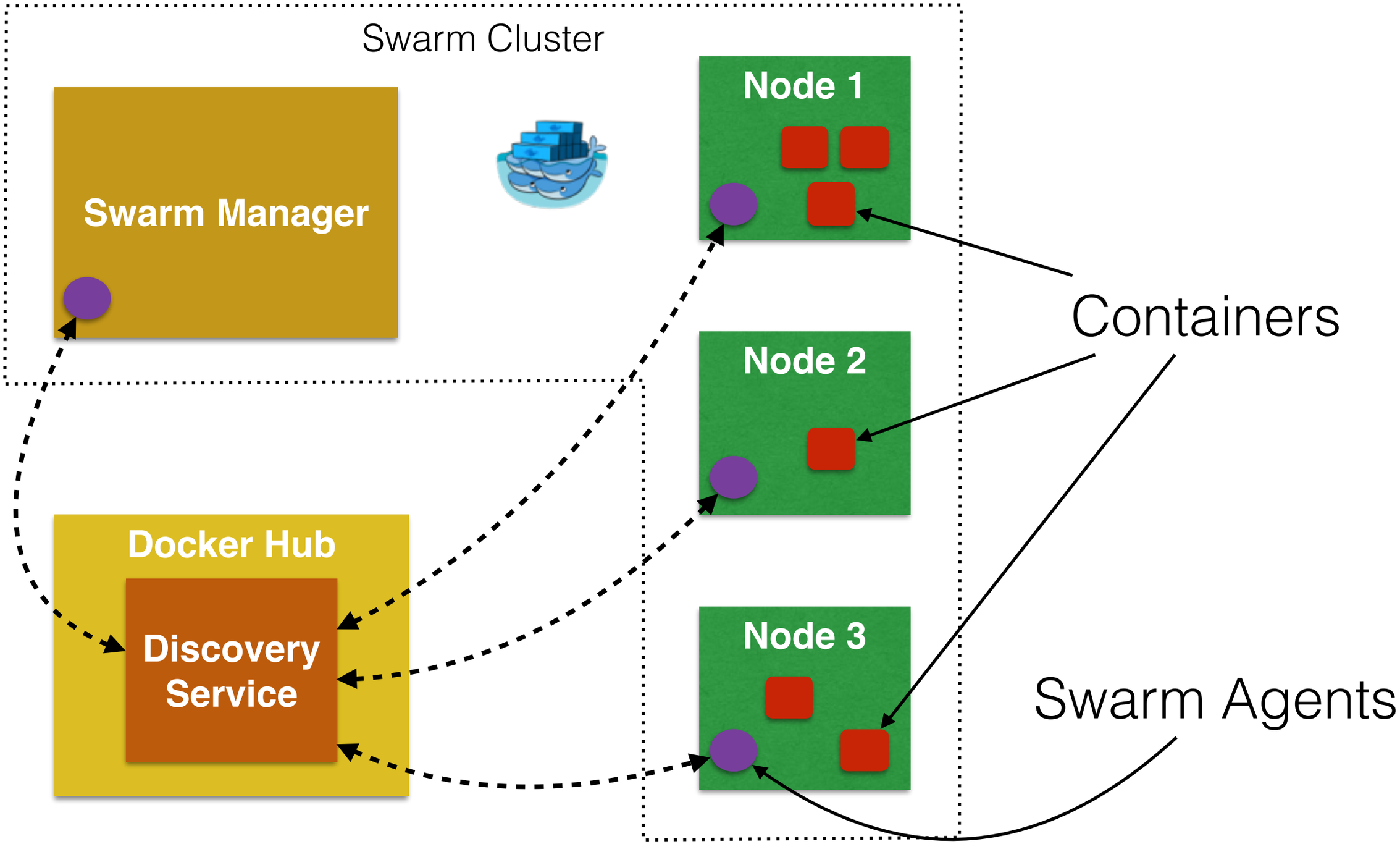
docker-compose.yml

```
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

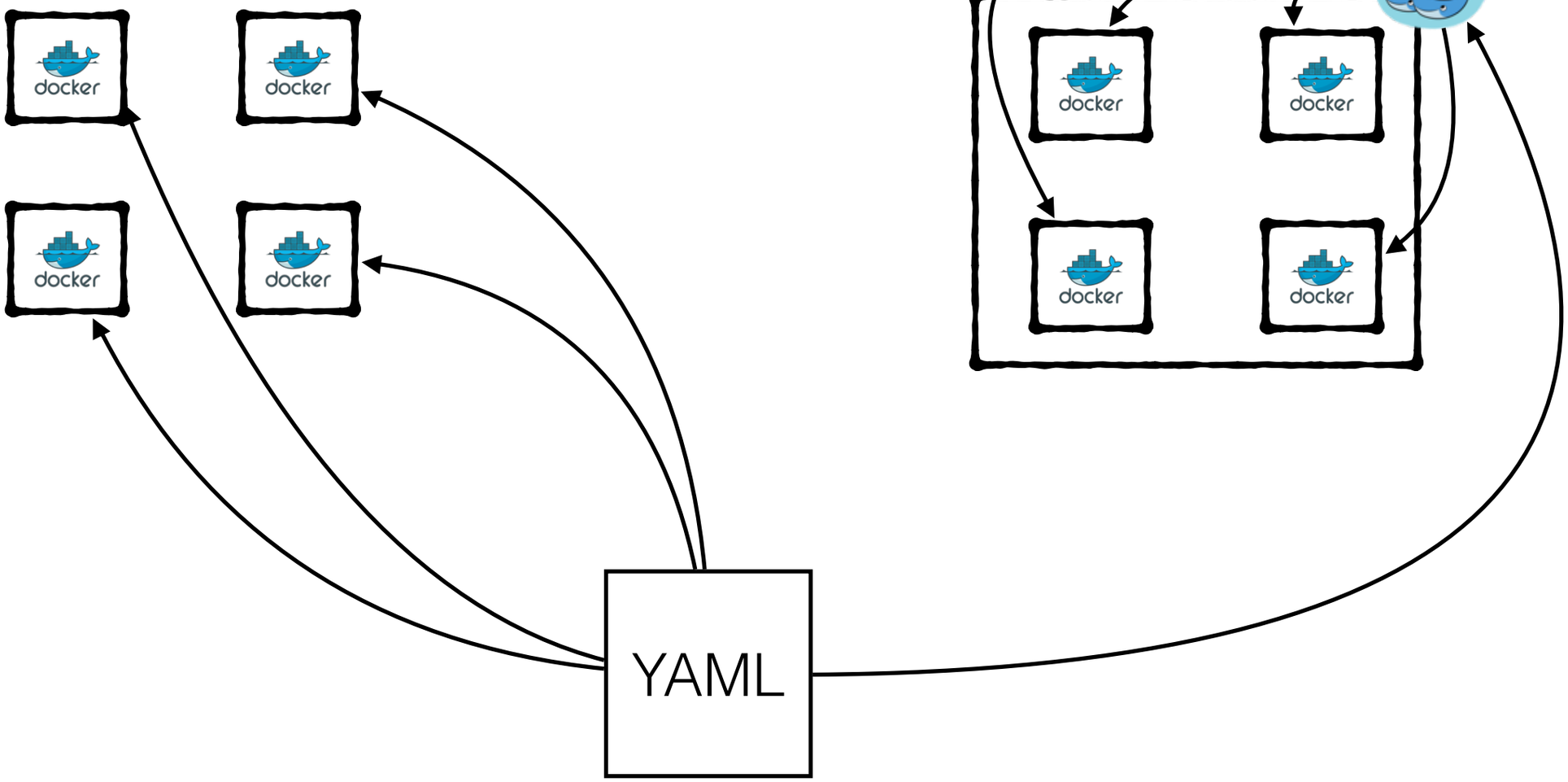
- Native clustering for Docker
- Provides a unified interface to a pool of Docker hosts
- Fully integrated with Machine
- Serves the standard Docker API
- Partially integrated with Compose
- Not recommended for production yet





http://blog.arungupta.me/clustering-docker-swarm-techtip85/

Machine



Swarm

Compose

Advantages of Containers

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