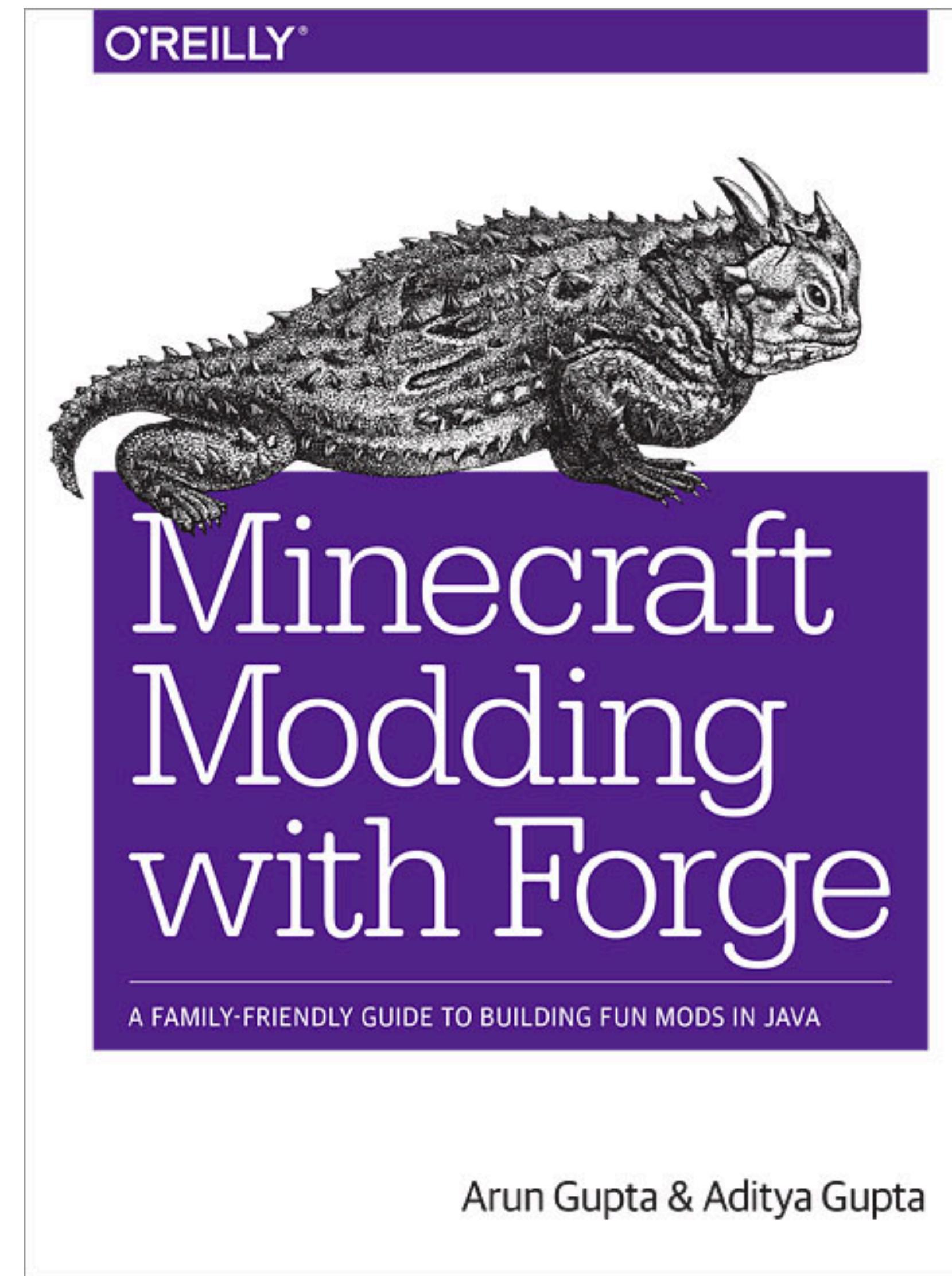


docker

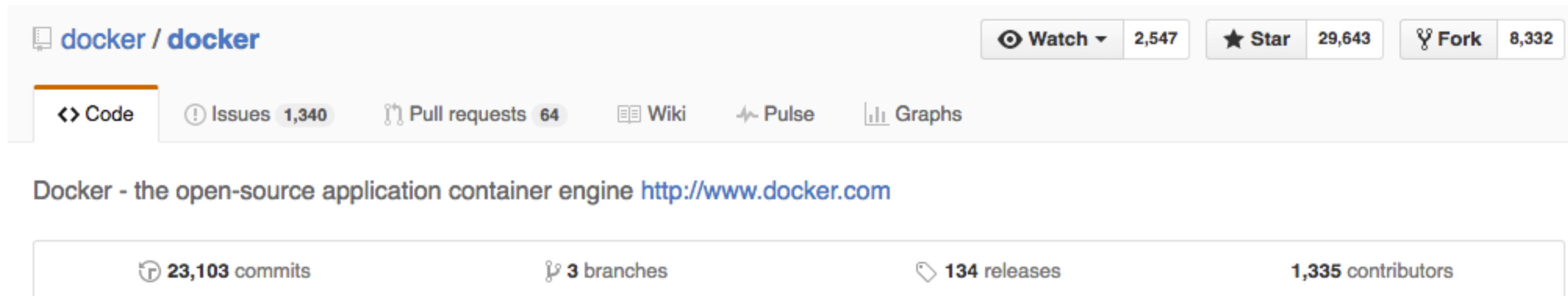
Docker for Java Developers

Arun Gupta, @arungupta
Java Champion, Docker Captain

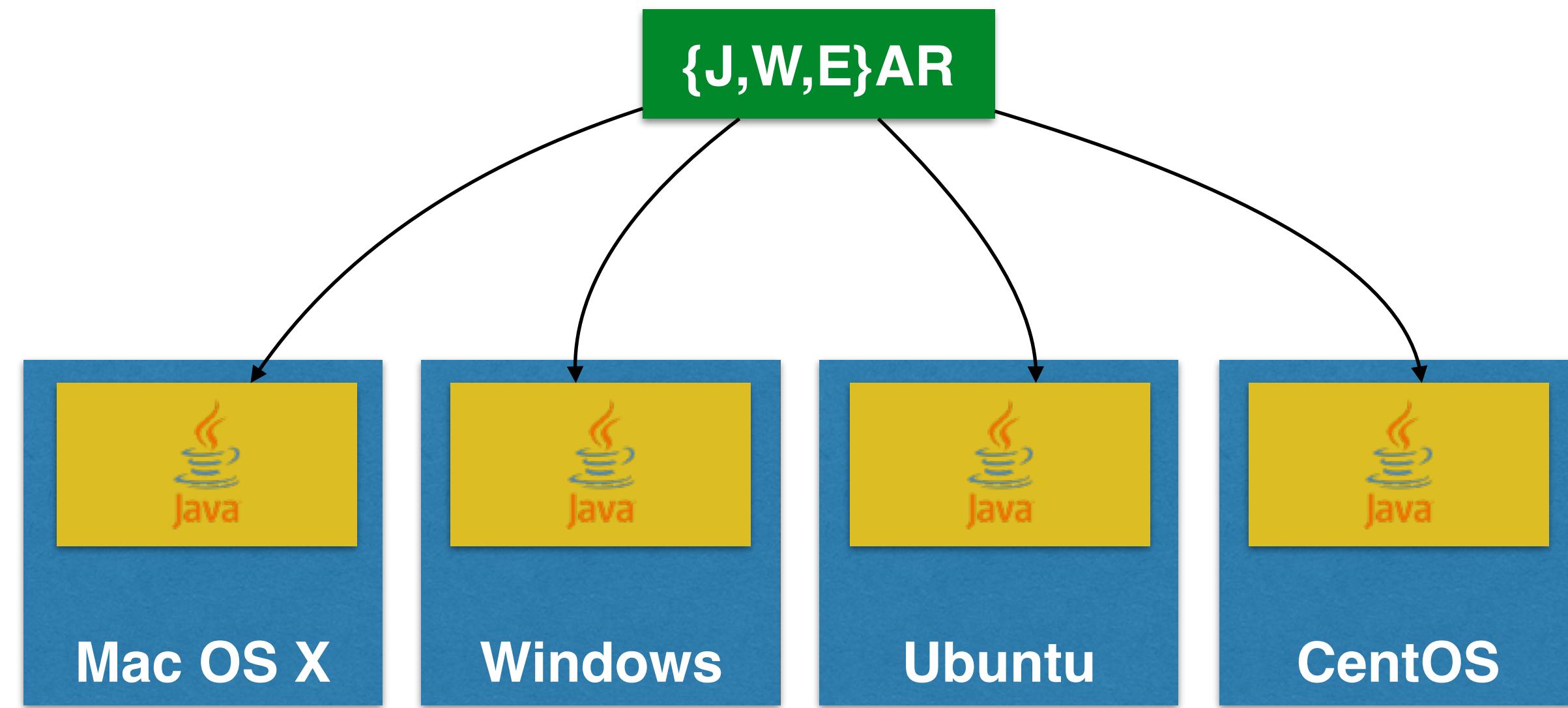


What is Docker?

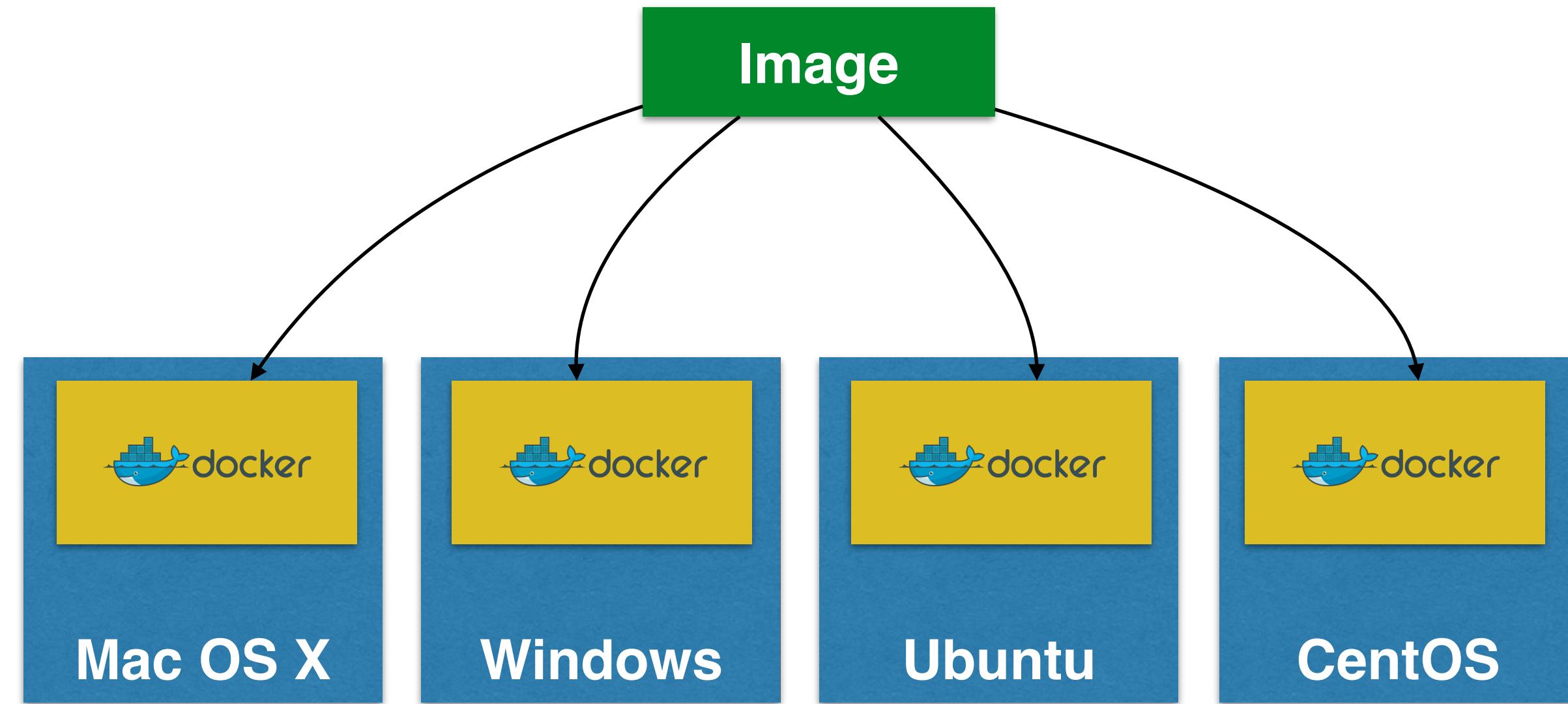
- Open source project and company



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

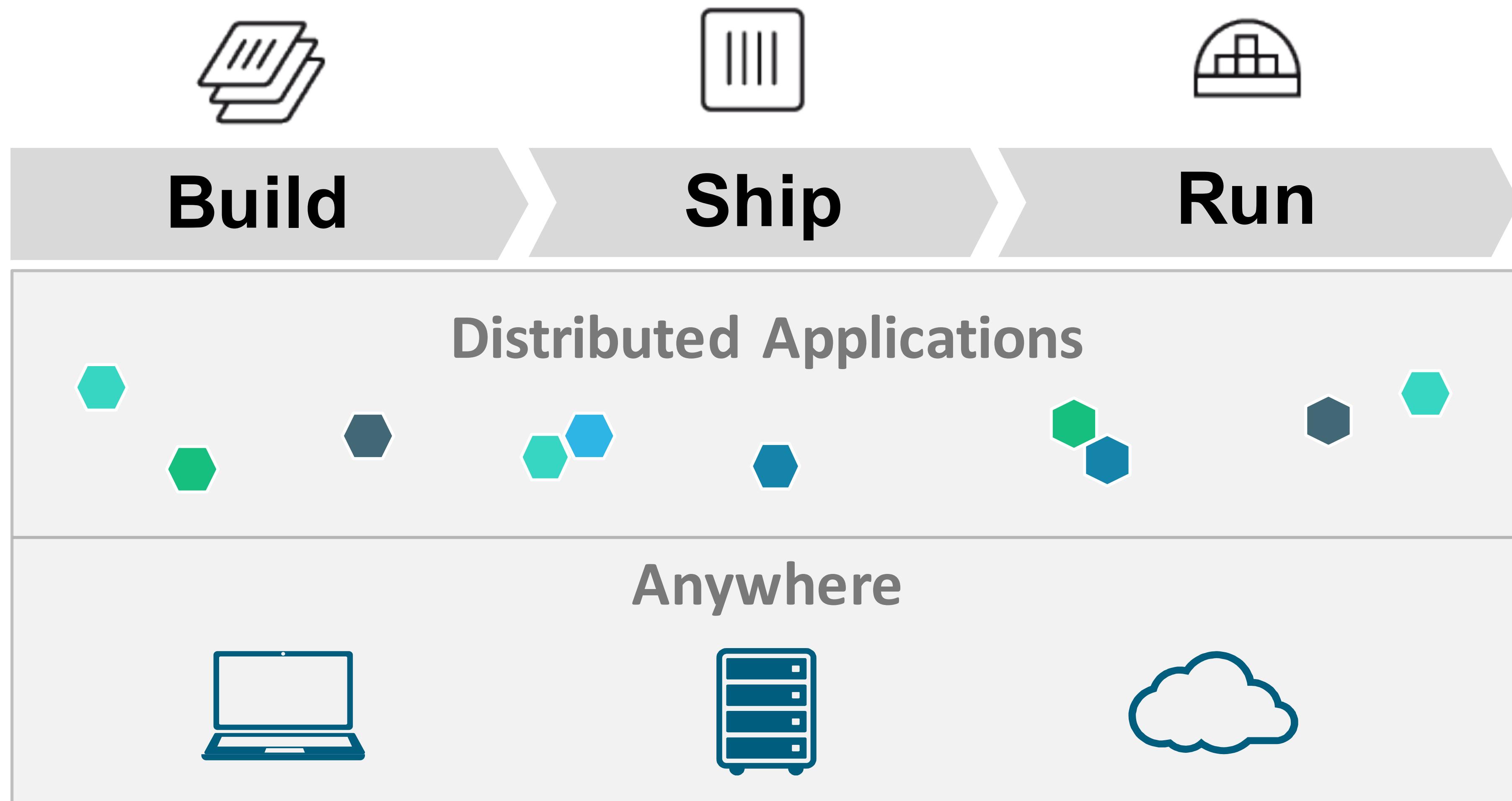


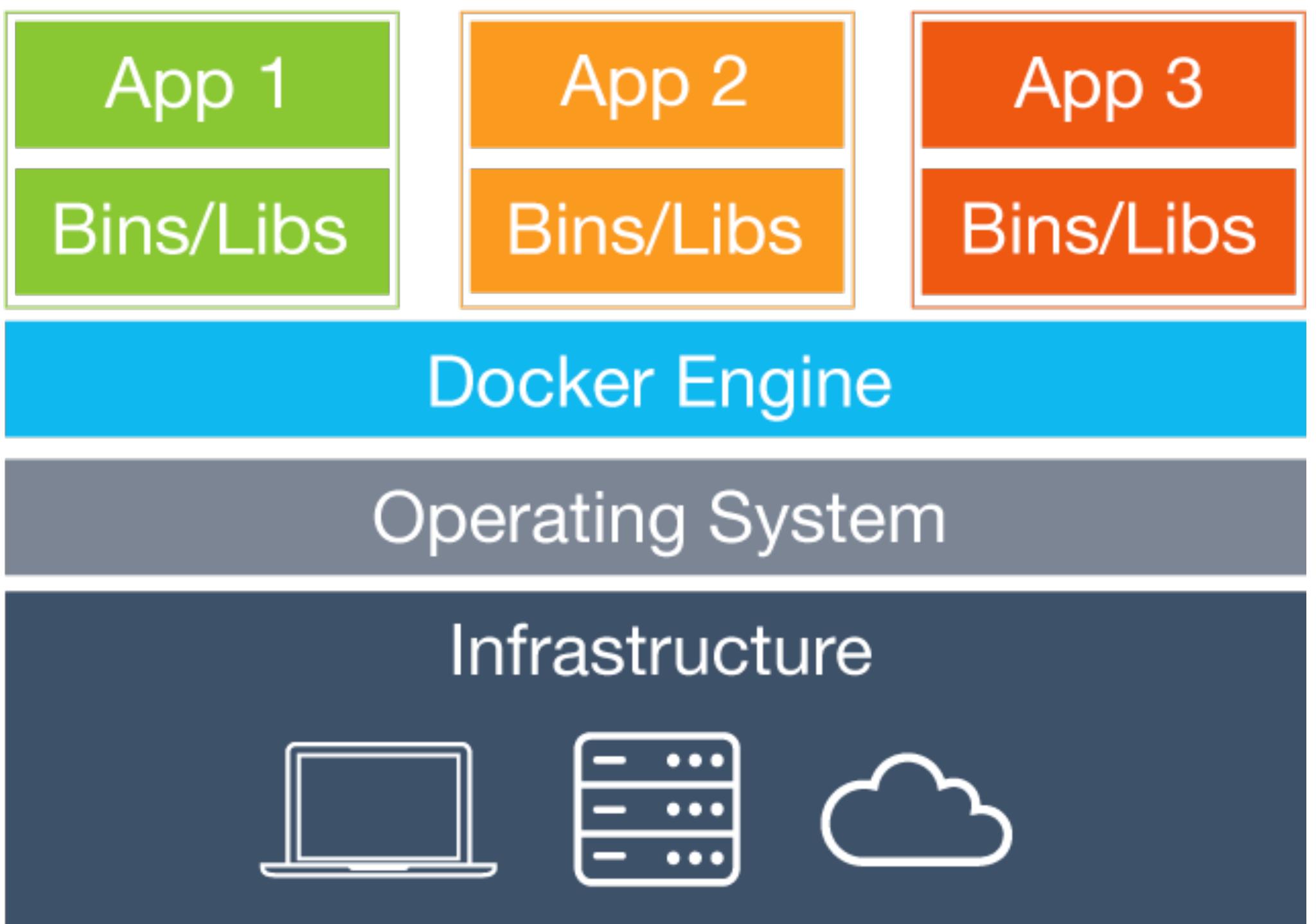
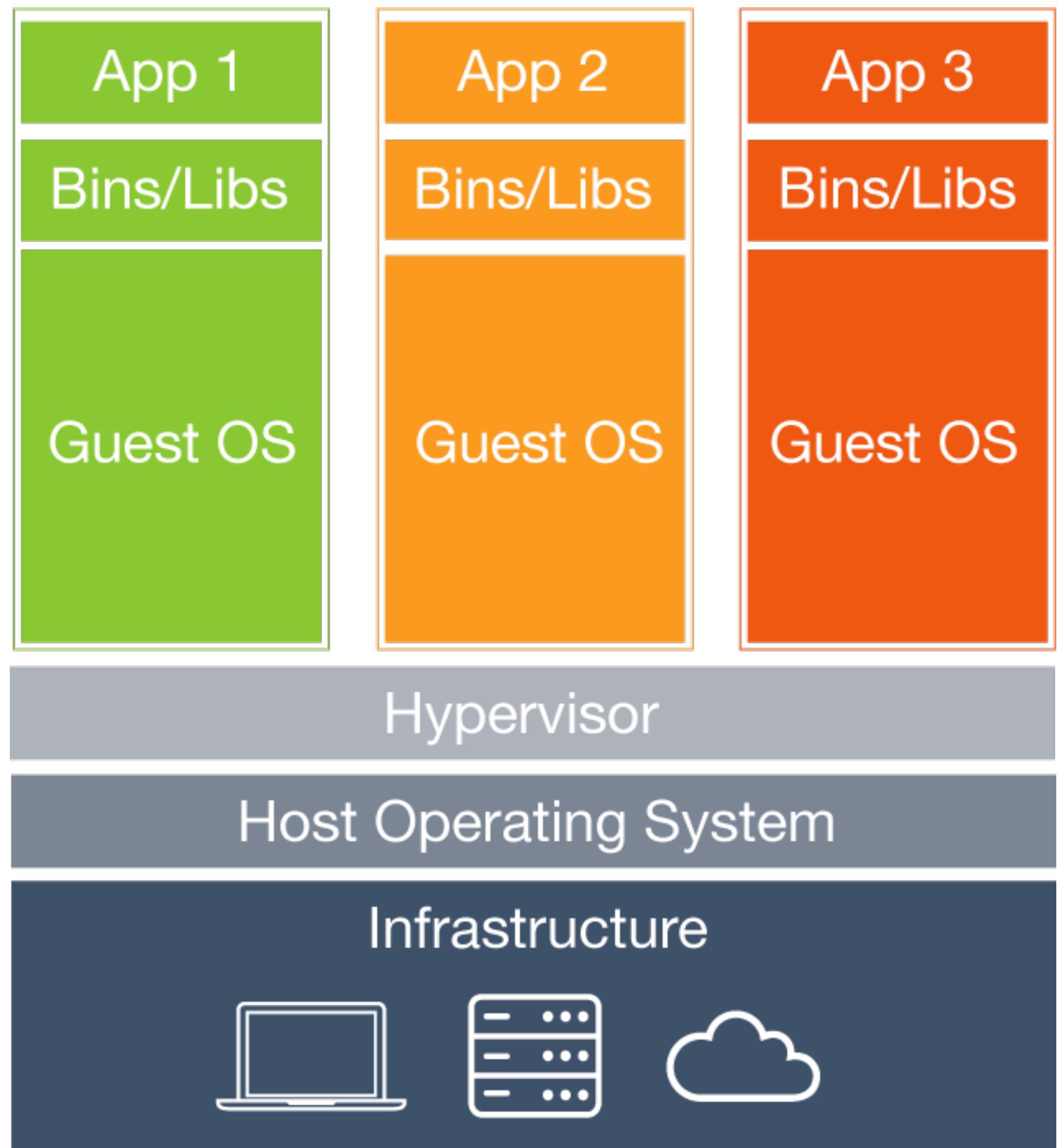
WORA = Write Once Run Anywhere



PODA = Package Once Deploy Anywhere

Docker Mission







Build

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

```
FROM ubuntu
```

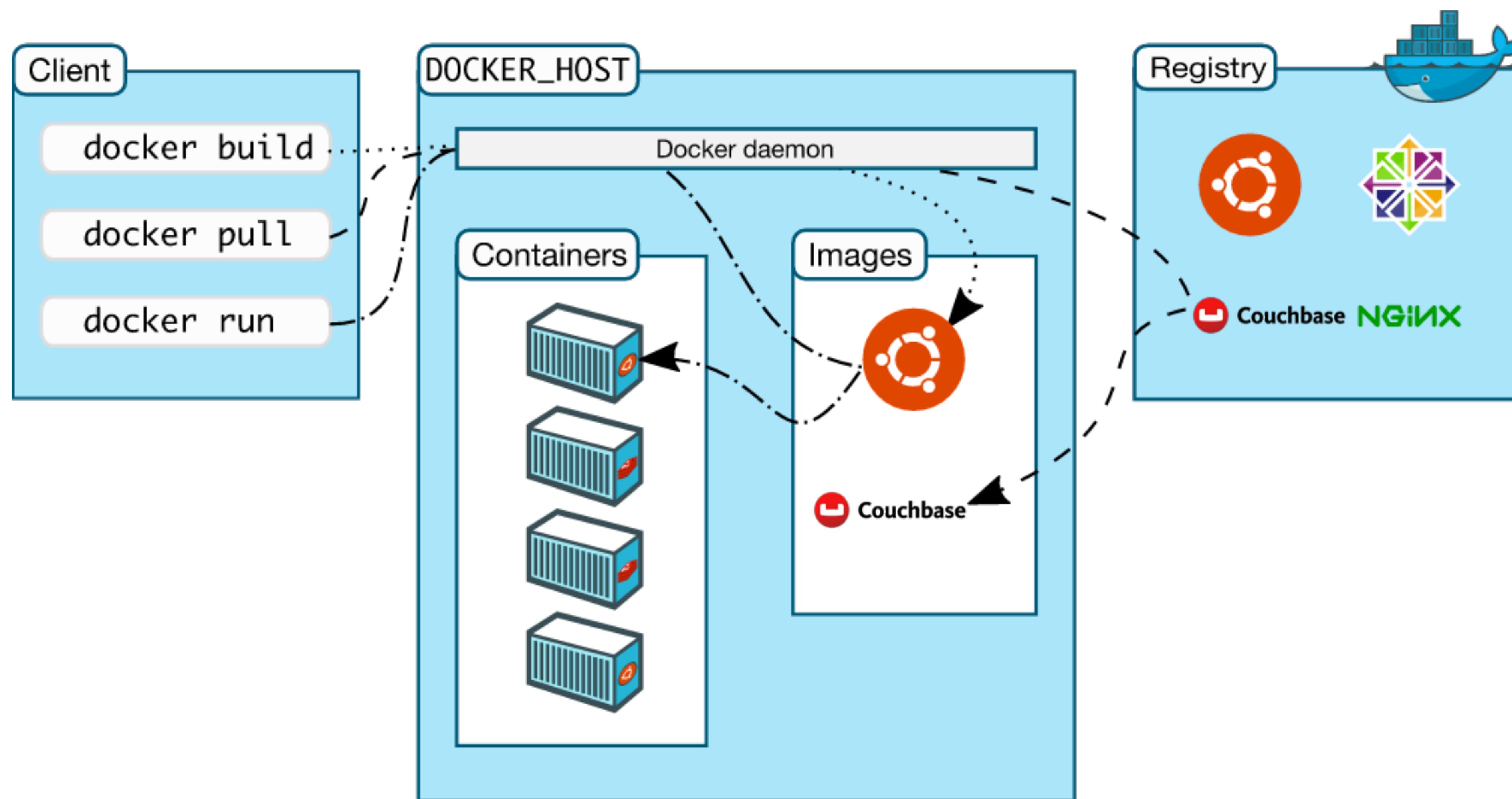
```
CMD echo "Hello world"
```

```
FROM java
```

```
COPY target/hello.jar /usr/src/hello.jar
```

```
CMD java -cp /usr/src/hello.jar org.example.App
```

Docker Workflow





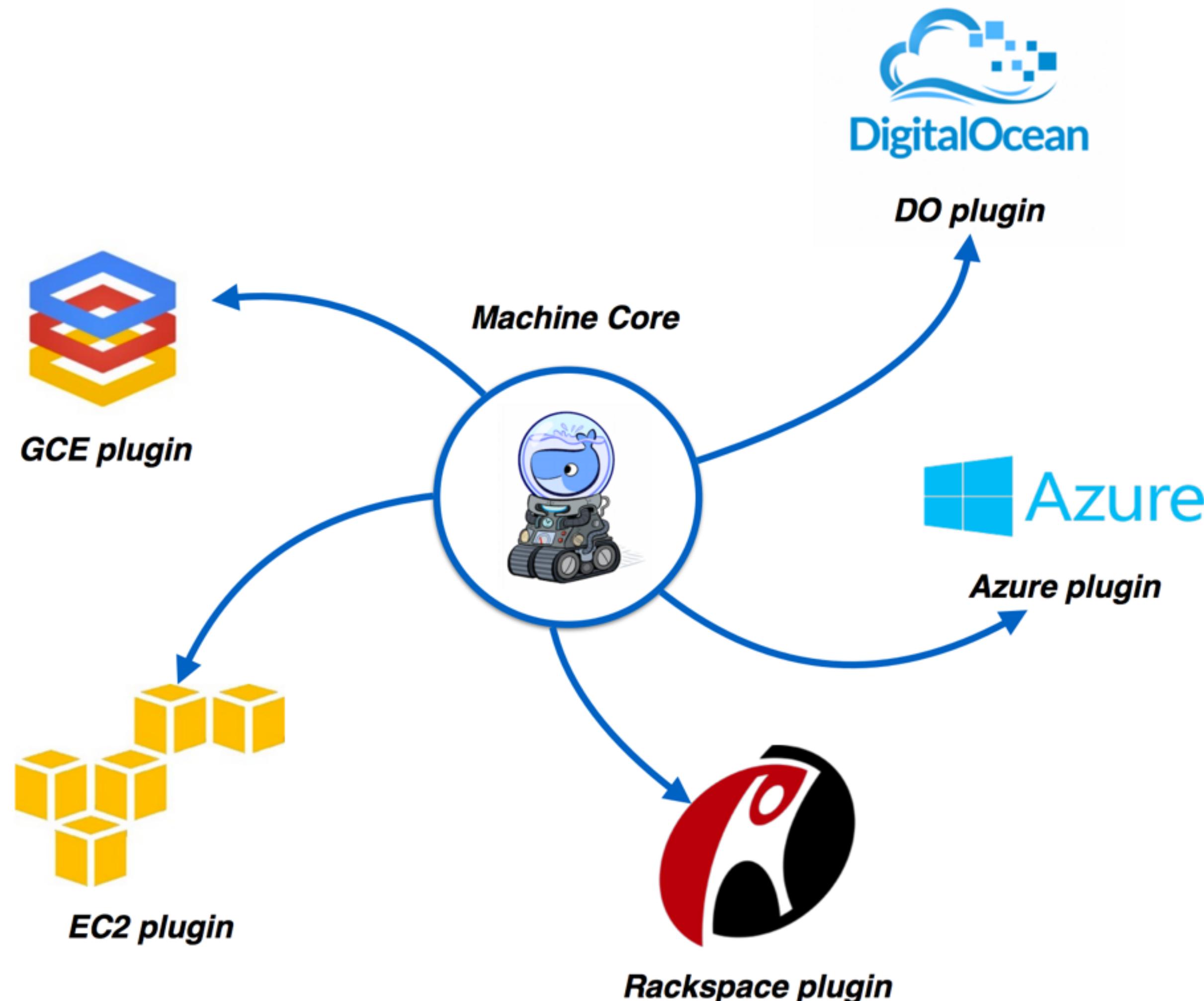
Docker Machine

- 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

Docker Machine Providers



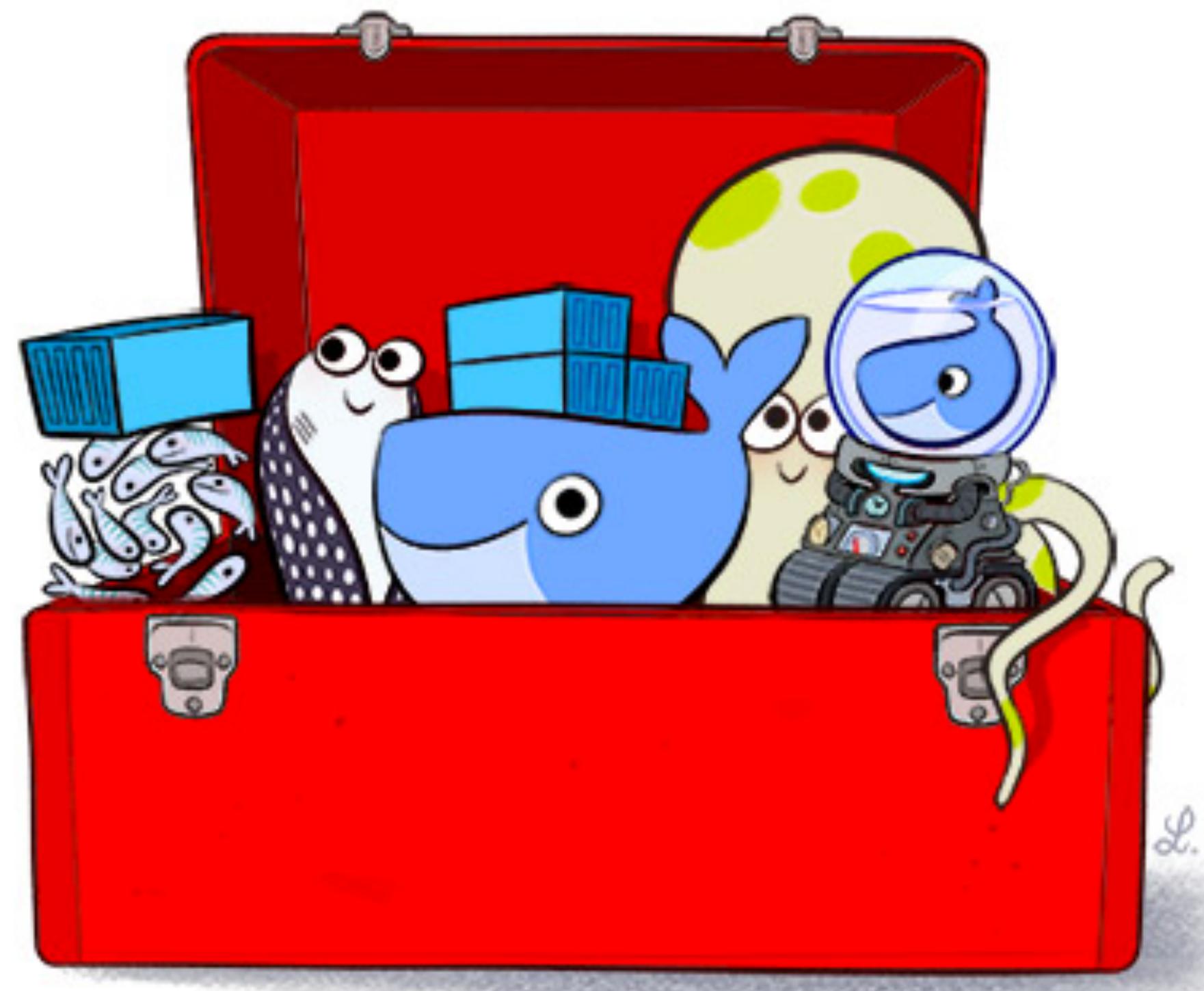


Docker for Mac/Windows

- beta.docker.com
- Native user interface and auto-update capability
- No VirtualBox!
 - **OSX**: xhyve VM using `Hypervisor.framework`
 - **Windows**: Hyper-V VM
- Better networking and filesystem mounting/notification

Docker Toolbox

- Docker Client
- Docker Machine
- Docker Compose
- Docker Kitematic
- Boot2Docker ISO
- Virtualbox



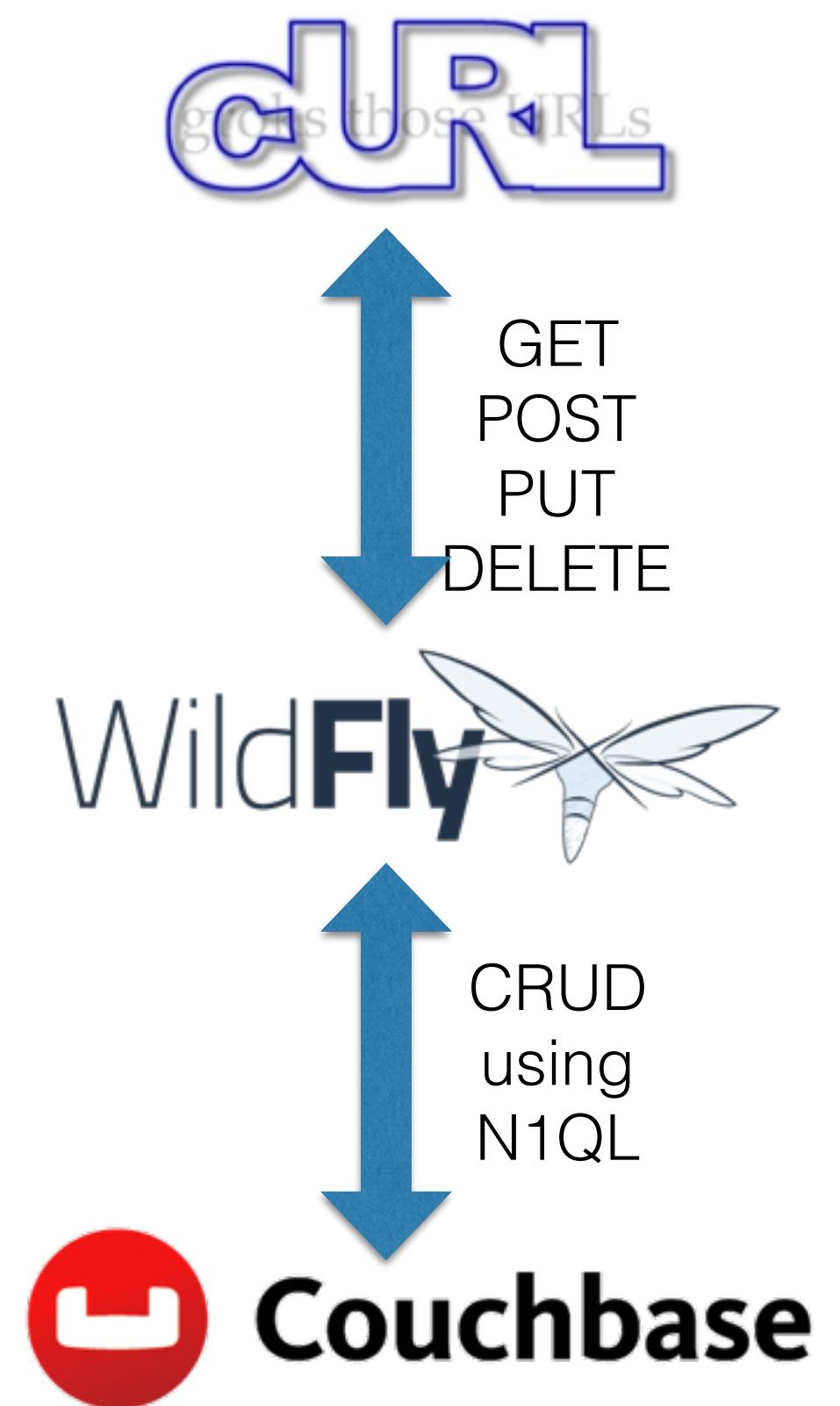


Docker Compose - One Service

```
version: "2"
services:
  db:
    image: couchbase
    volumes:
      - ~/couchbase:/opt/couchbase/var
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
```



Docker Compose - Two Services





Docker Compose

- Defining and running multi-container applications
- Configuration defined in one or more files
 - `docker-compose.yml` (default)
 - `docker-compose.override.yml` (default)
 - Multiple files specified using `-f`
 - All paths relative to base configuration file
- Great for dev, staging, and CI

Docker Compose - Two Services

```
version: "2"
services:
  db:
    container_name: "db"
    image: couchbase
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
  web:
    image: jboss/wildfly
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
      - 9990:9990
```



Overriding Services in Docker Compose

```
mywildfly:  
  image: jboss/wildfly  
  ports:  
    - 8080:8080
```

docker-compose.yml

```
mywildfly:  
  ports:  
    - 9080:8080
```

docker-compose.override.yml

docker-compose up -d

Dev/Prod with Compose

```
mycouchbase:  
  container_name: "db-dev"  
  image: arungupta/couchbase  
  ports:  
    - . . .  
mywildfly:  
  image: arungupta/wildfly  
  environment:  
    - COUCHBASE_URI=db-dev:8093  
  ports:  
    - 8080:8080
```

docker-compose.yml

docker-compose up -d

```
mywildfly:  
  environment:  
    - COUCHBASE_URI=db-prod:8093  
  ports:  
    - 8080:80  
mycouchbase:  
  container_name: "db-prod"
```

production.yml

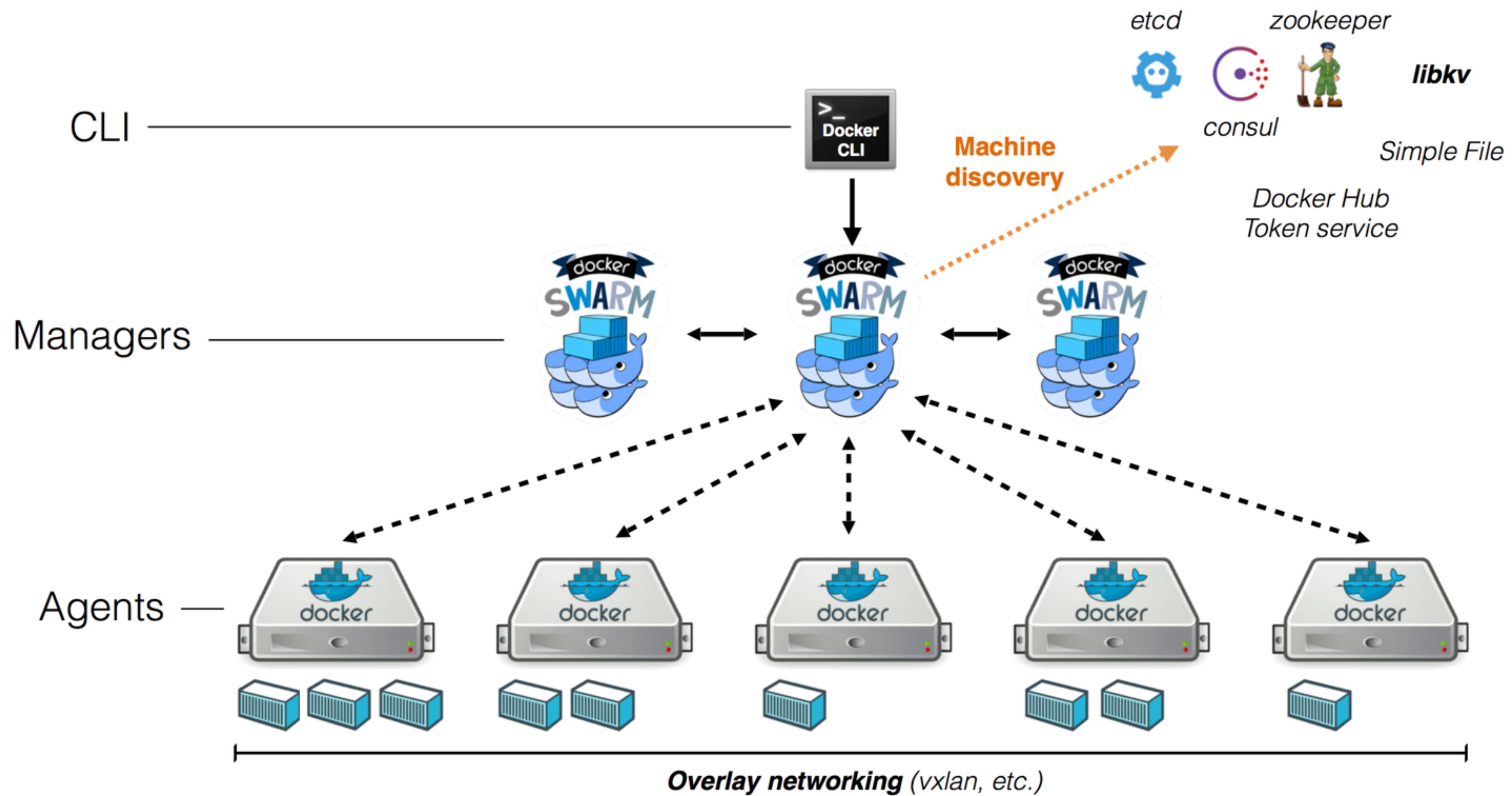
docker-compose up
-f docker-compose.yml
-f production.yml
-d



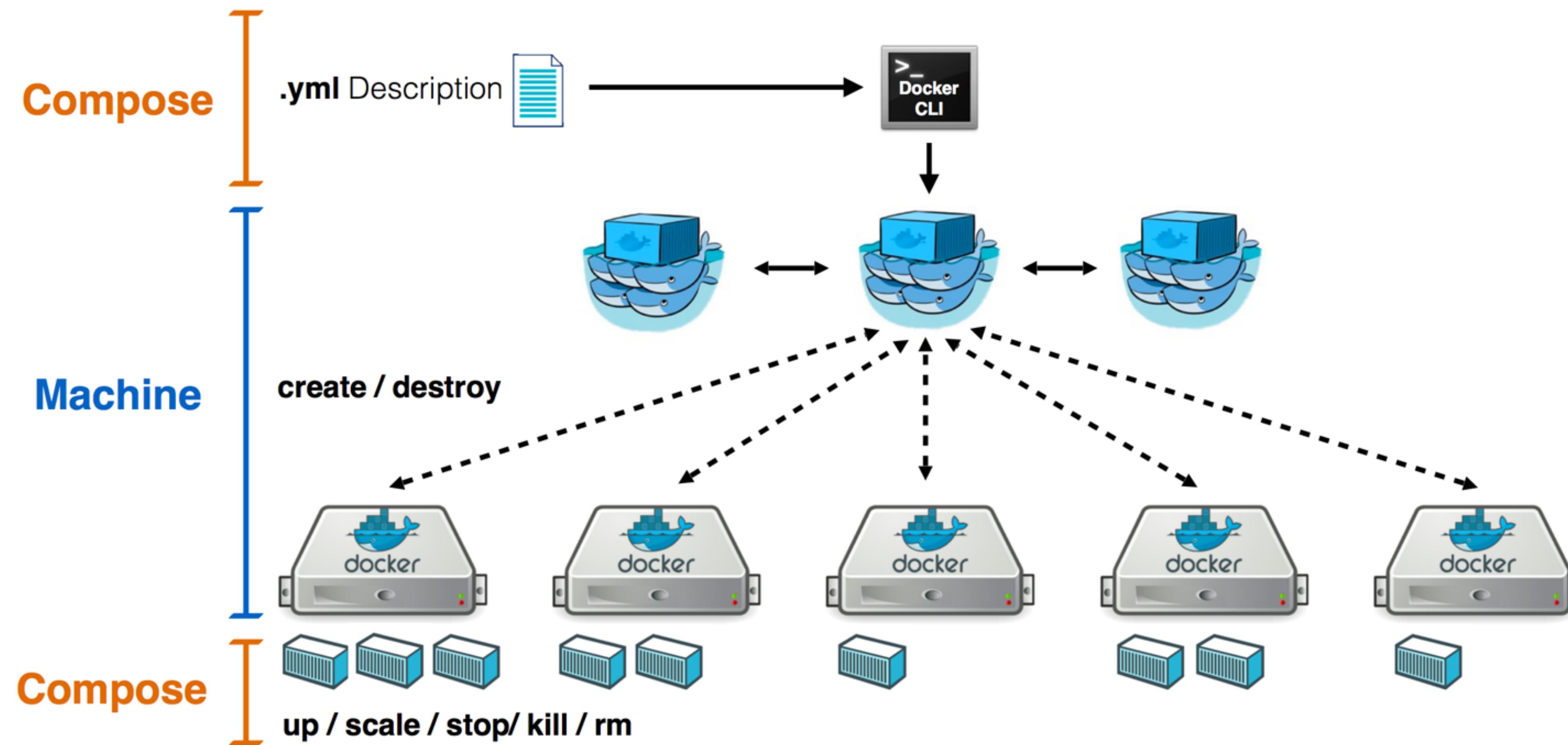
Docker Swarm

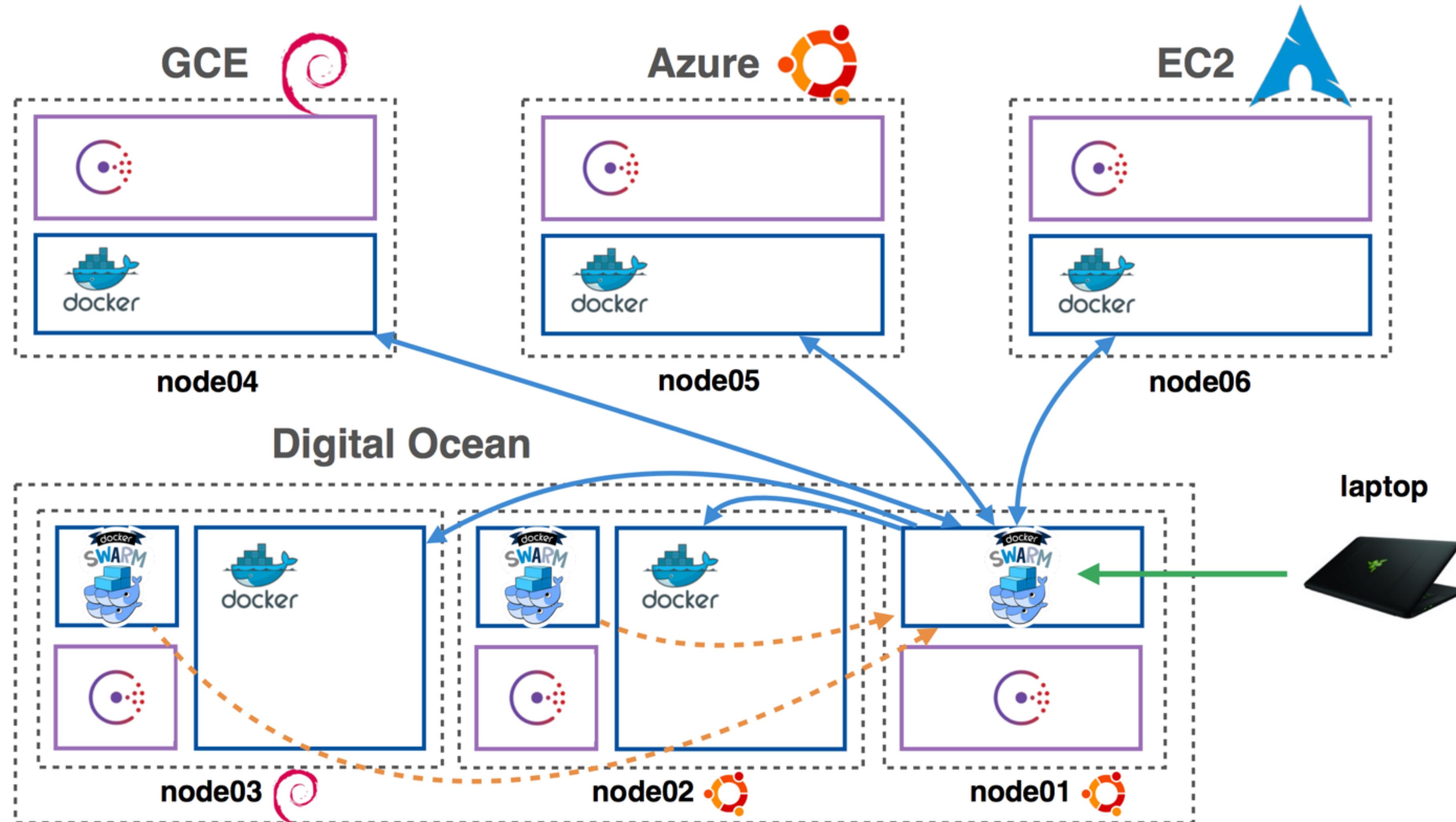
- Native clustering for Docker
- Provides a unified interface to a pool of Docker hosts
- Fully integrated with Machine and Compose
- Serves the standard Docker API
- 1.2 - Ready for production
 - Reschedule containers when a node fails
 - Better node management

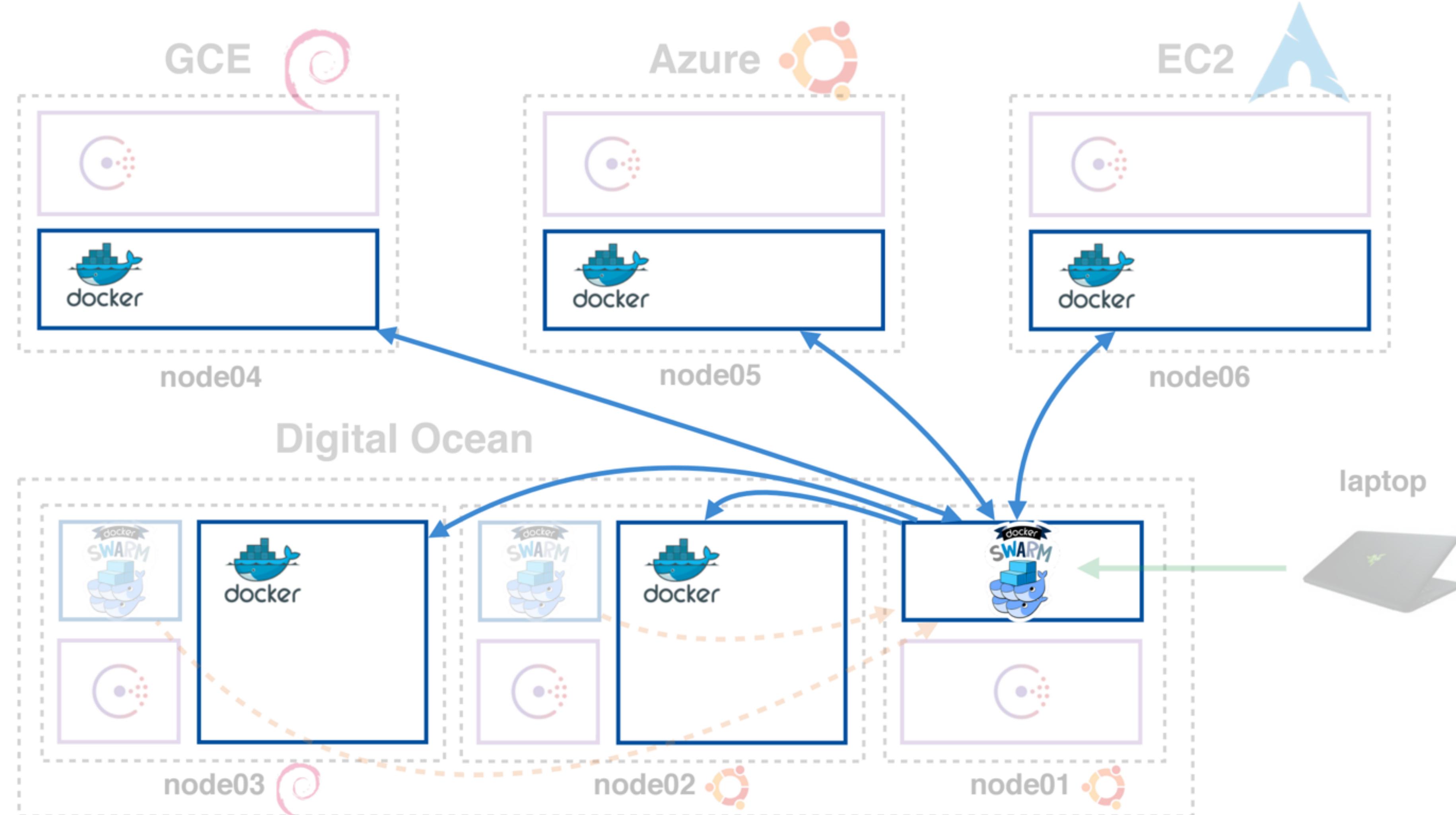
Stress tested on 1000 EC2 nodes, ~30k containers

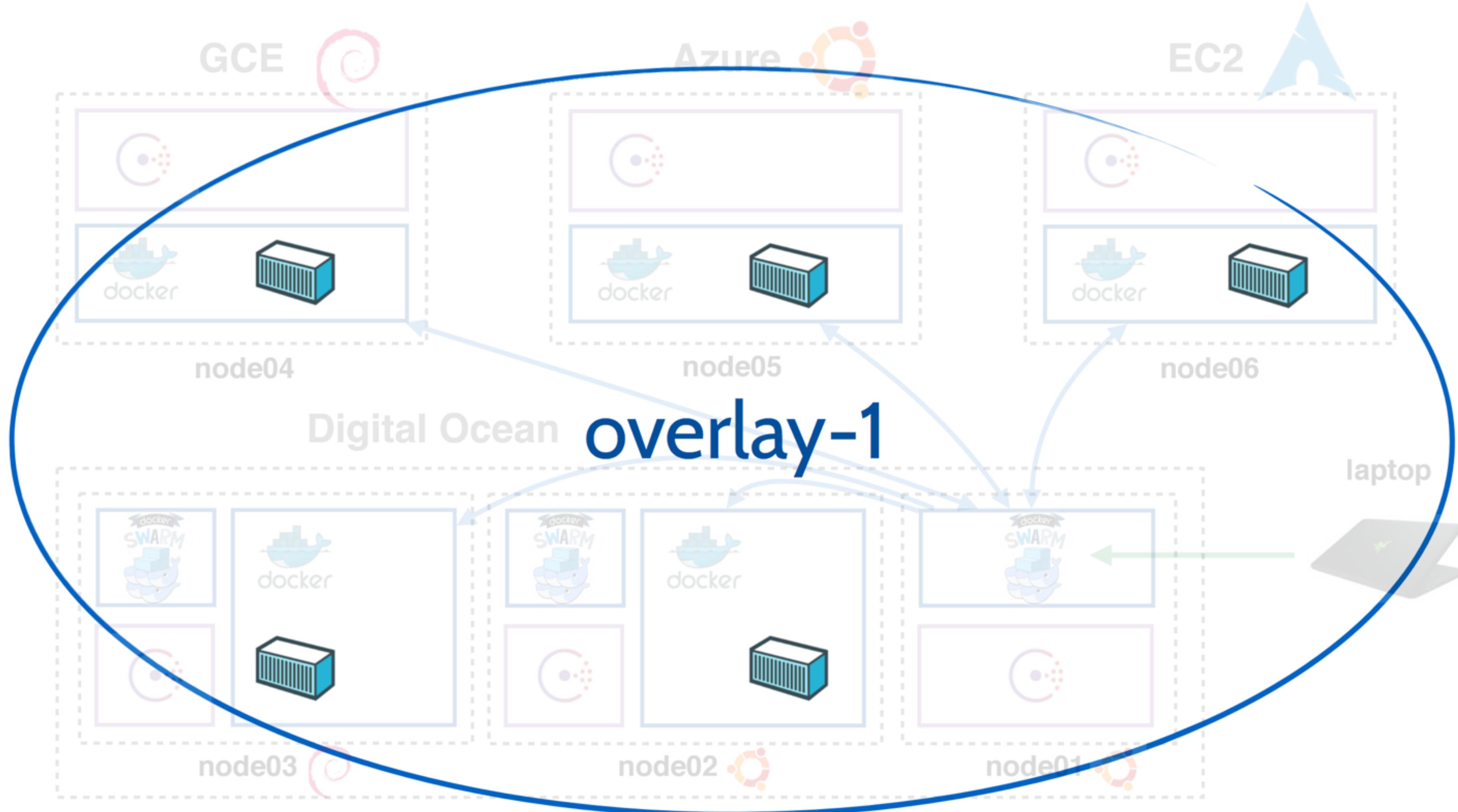


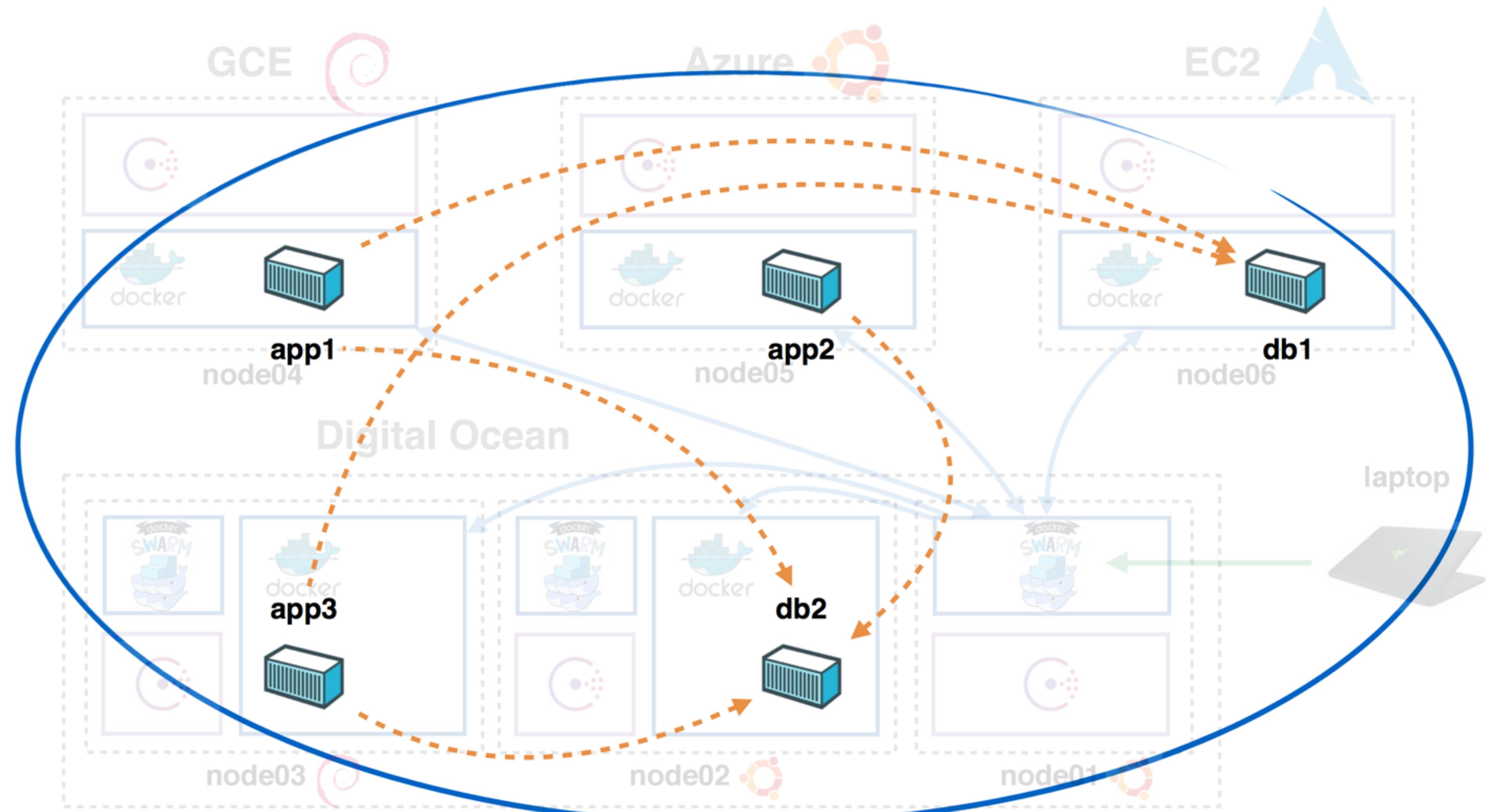
Machine + Swarm + Compose

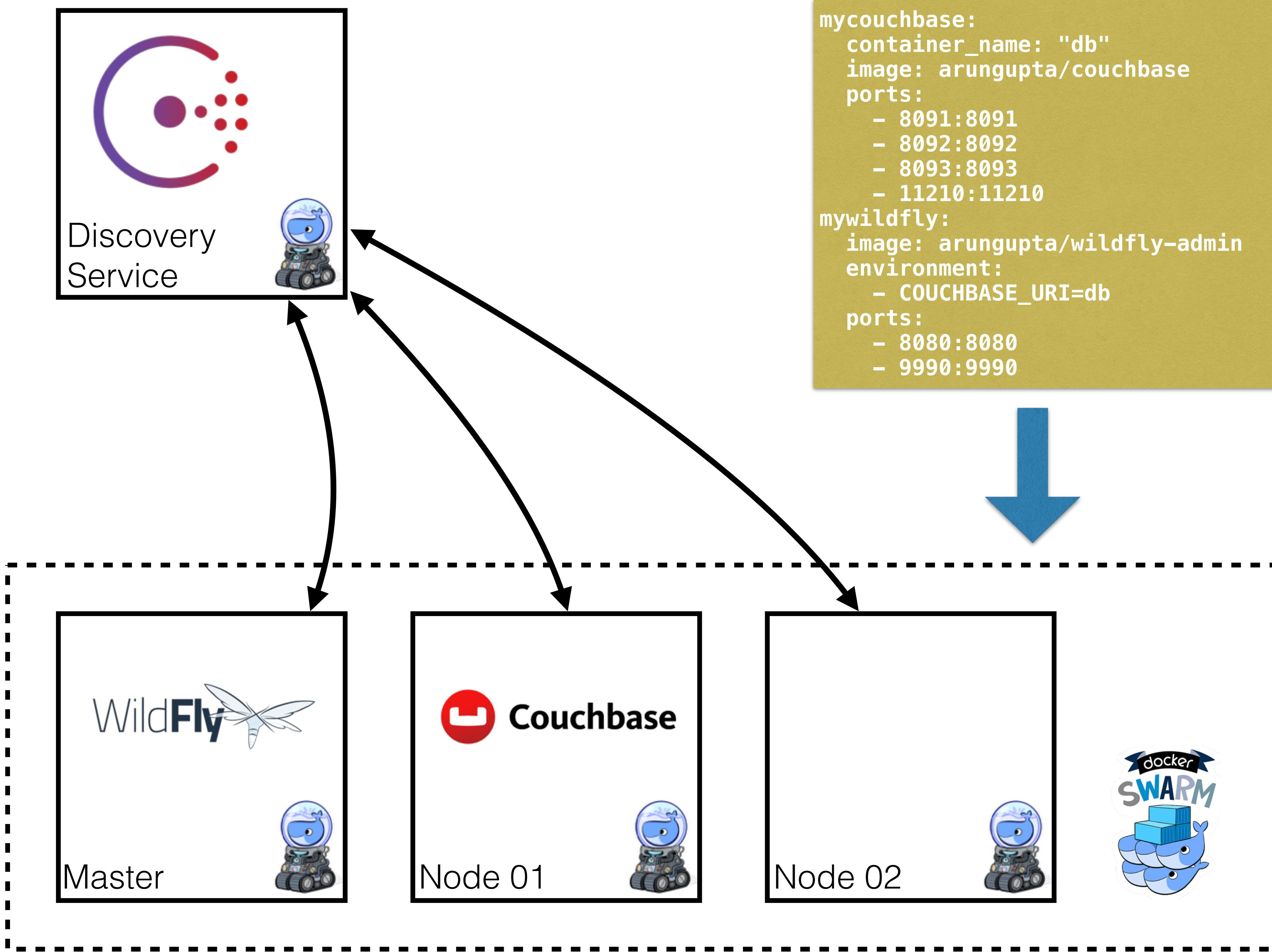




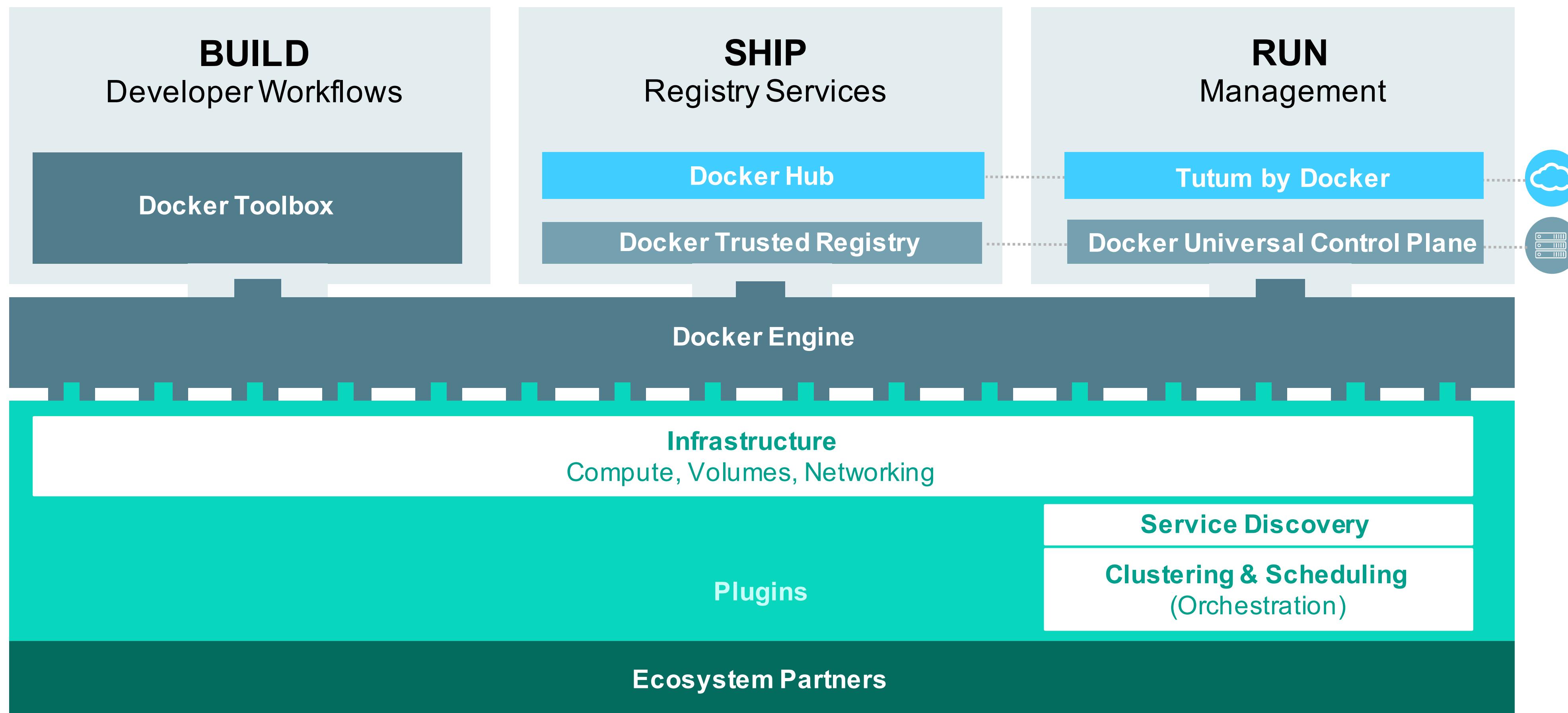








Docker Mission



References

- Docker for Java Tutorial: github.com/docker/labs/tree/master/java
- Docker Documentation: docs.docker.com
- IntelliJ Documentation: jetbrains.com/idea/docker/