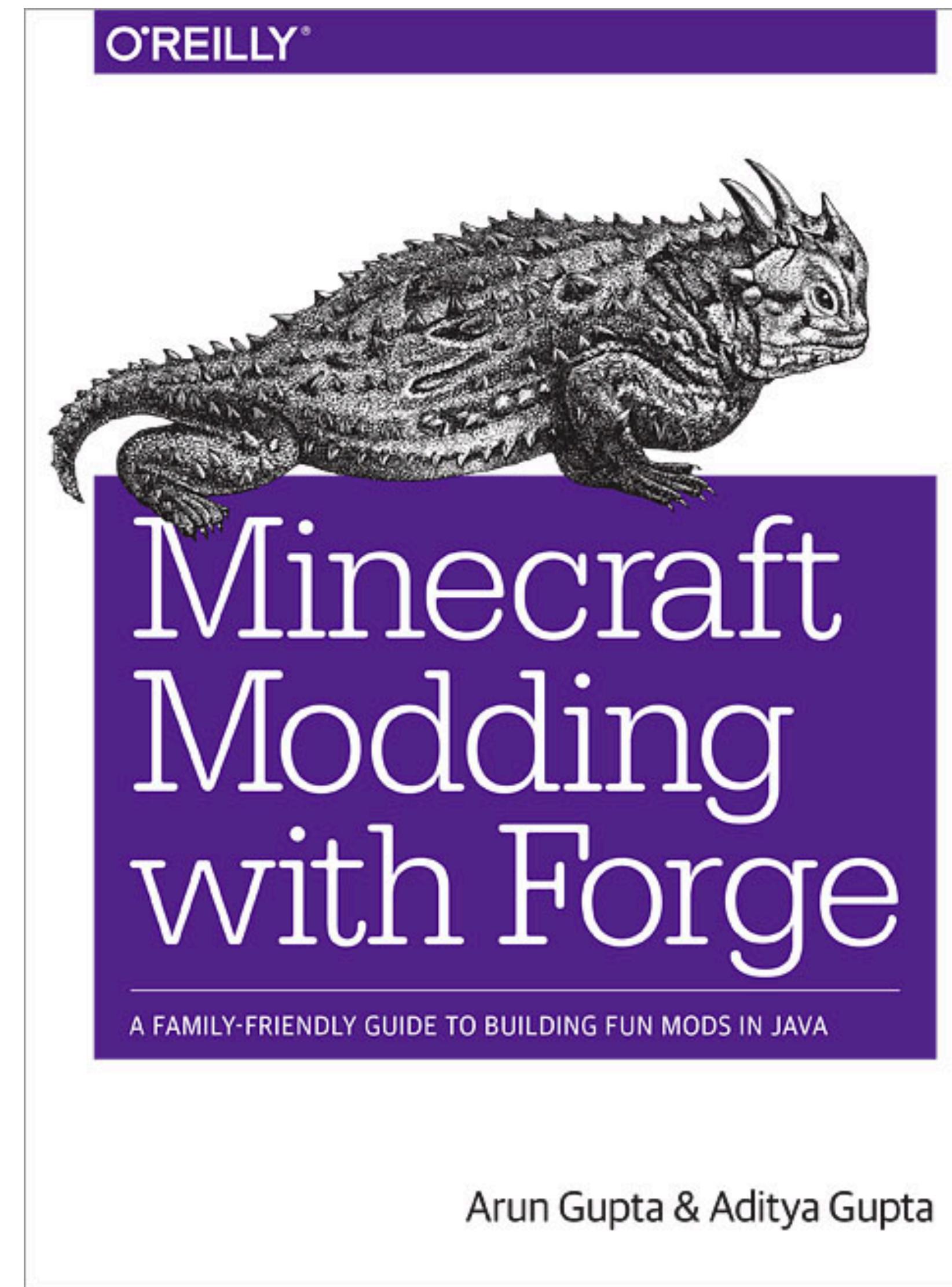


docker

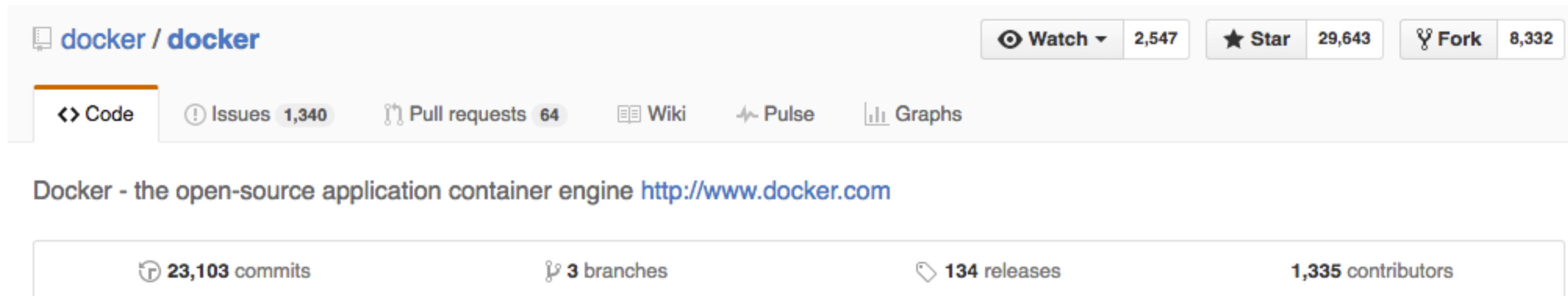
# Docker for Java Developers

Arun Gupta, @arungupta  
Docker Captain, Java Champion

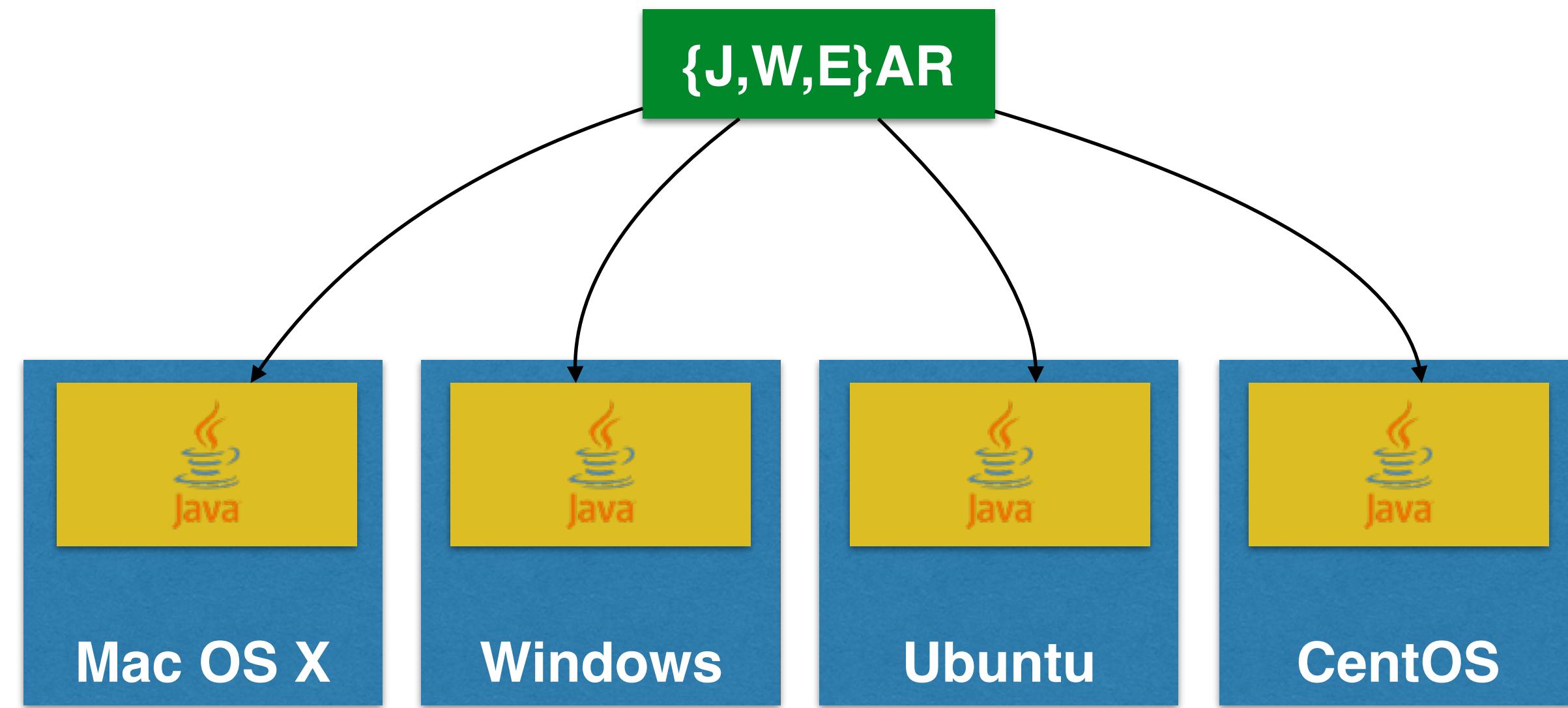


# What is Docker?

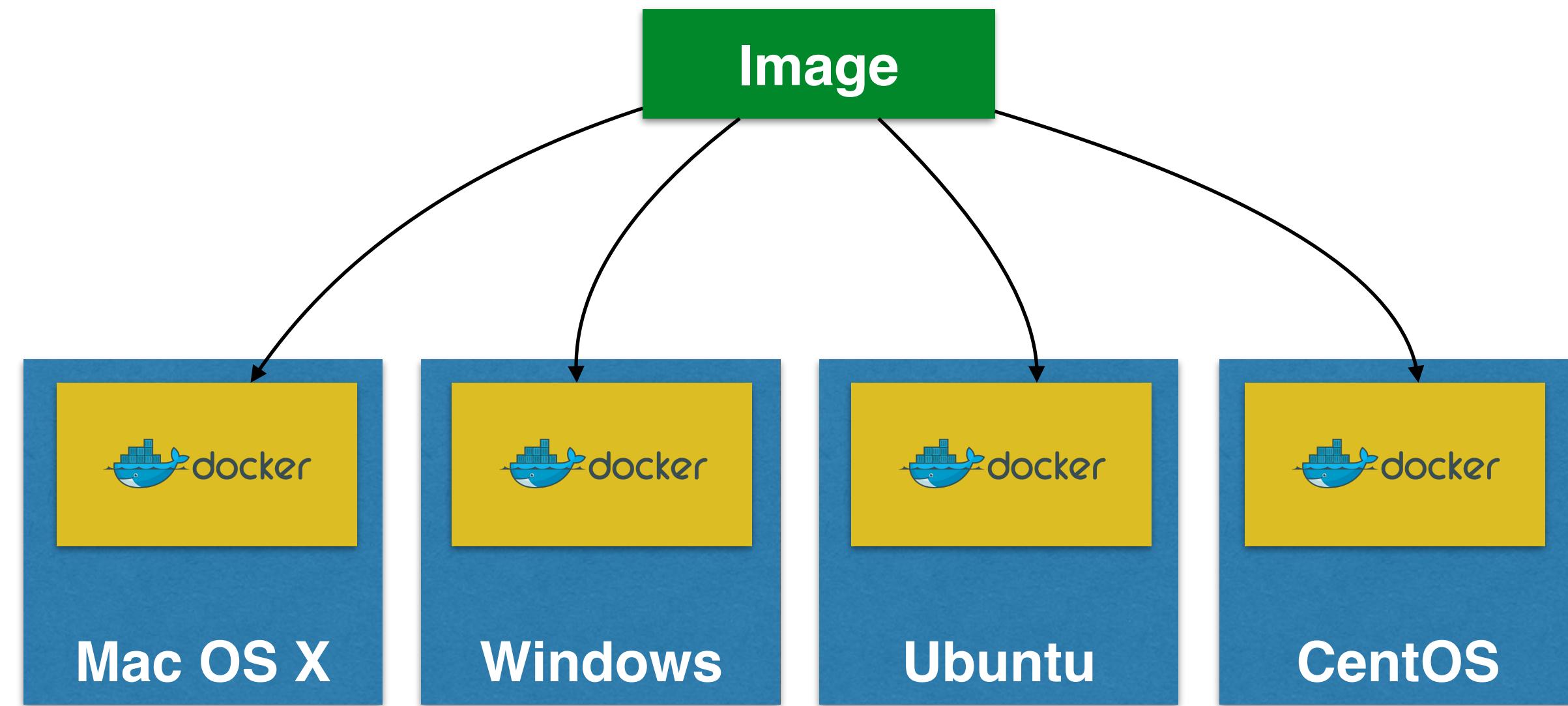
- Open source project and company



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

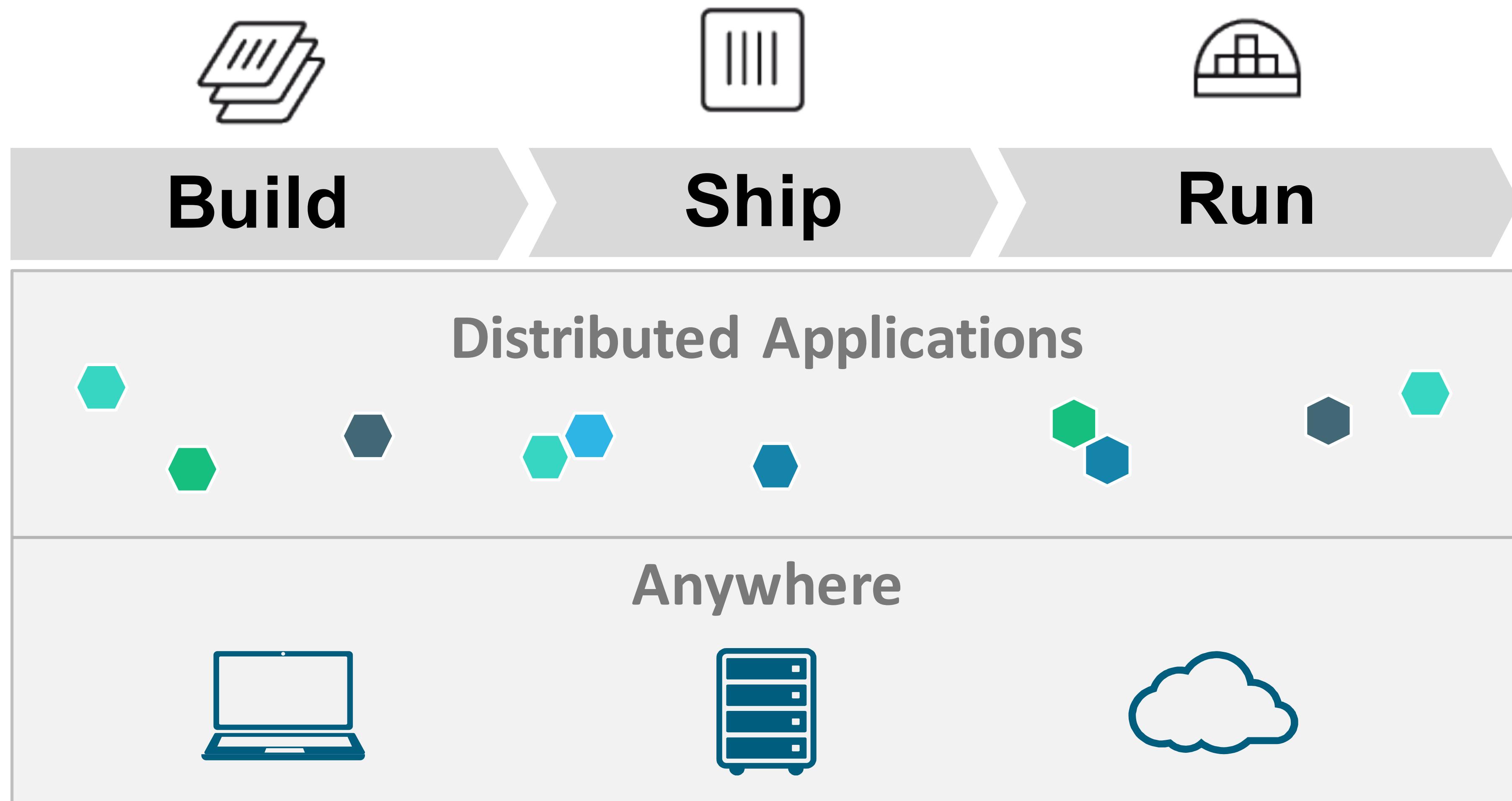


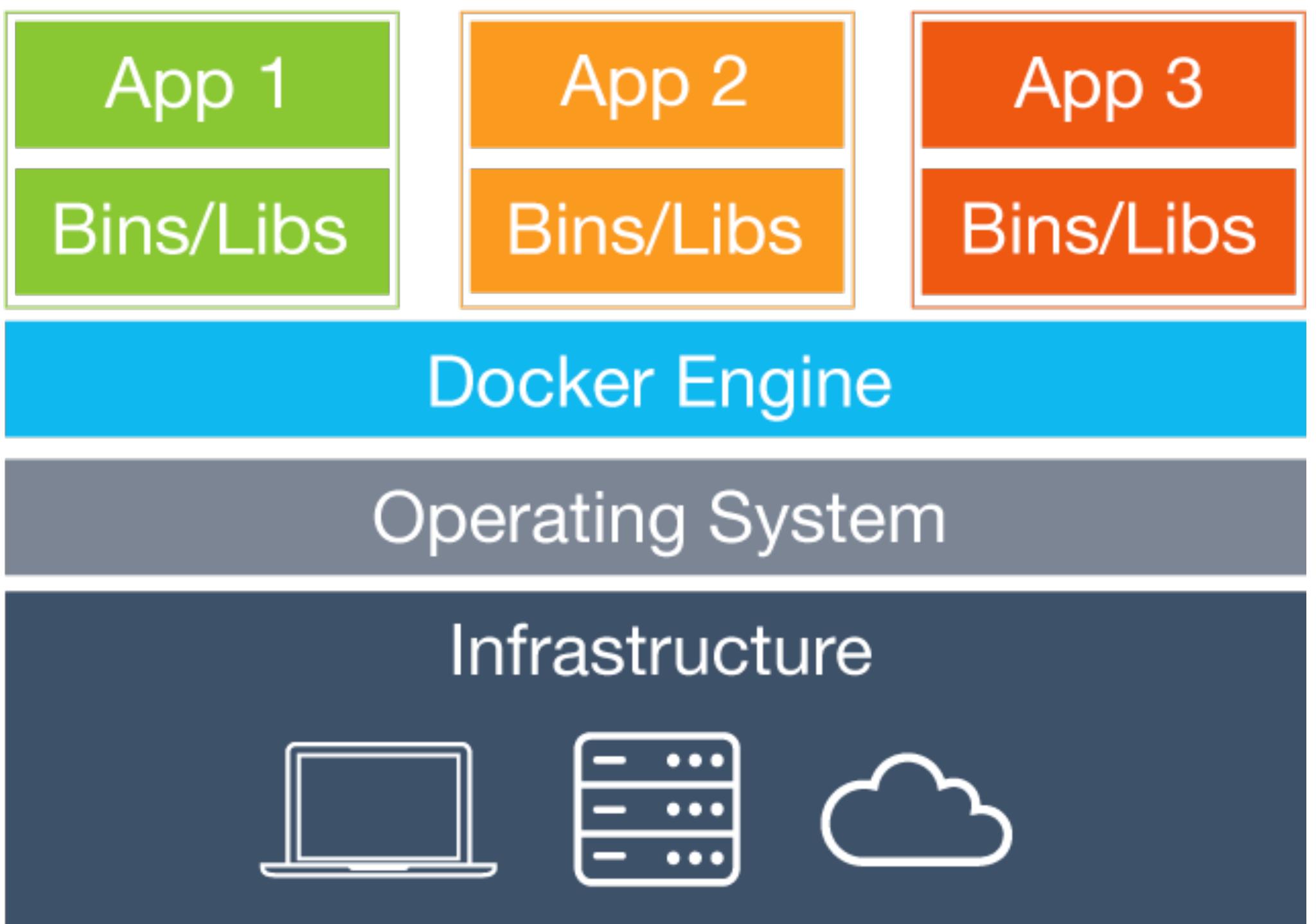
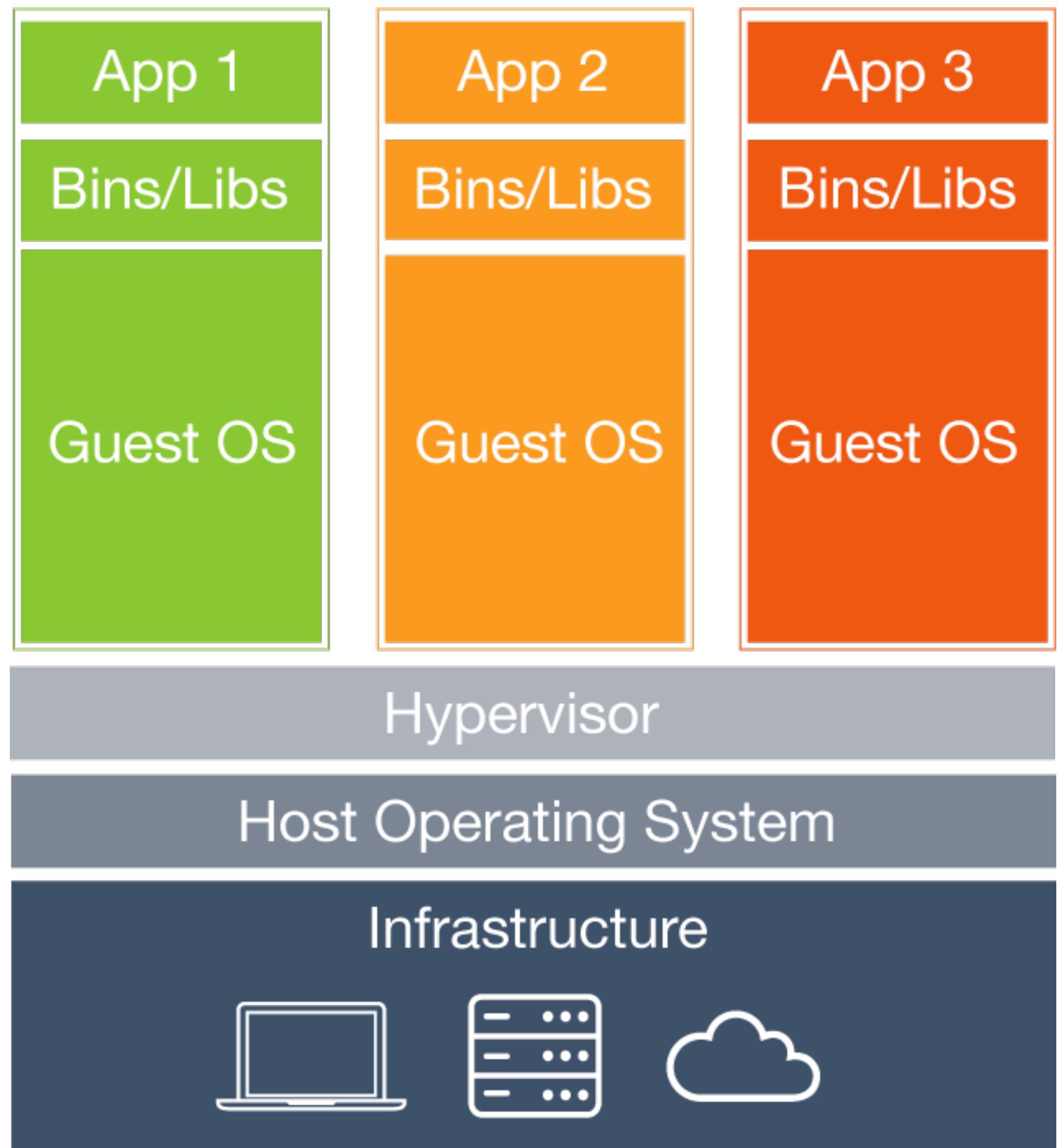
**WORA = Write Once Run Anywhere**



**PODA = Package Once Deploy Anywhere**

# Docker Mission







## Build

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

```
FROM ubuntu
```

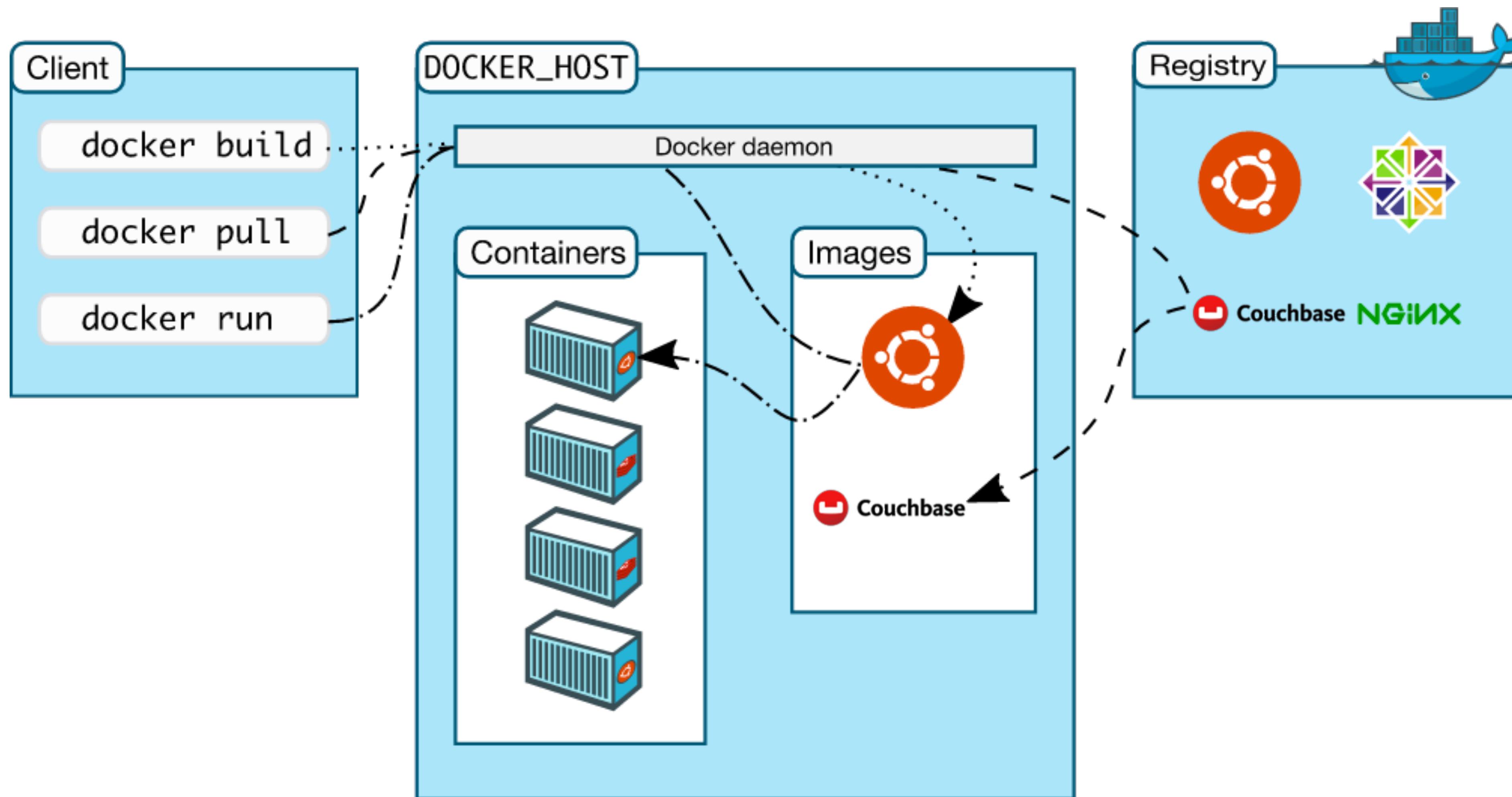
```
CMD echo "Hello world"
```

```
FROM java
```

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

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

# Docker Workflow





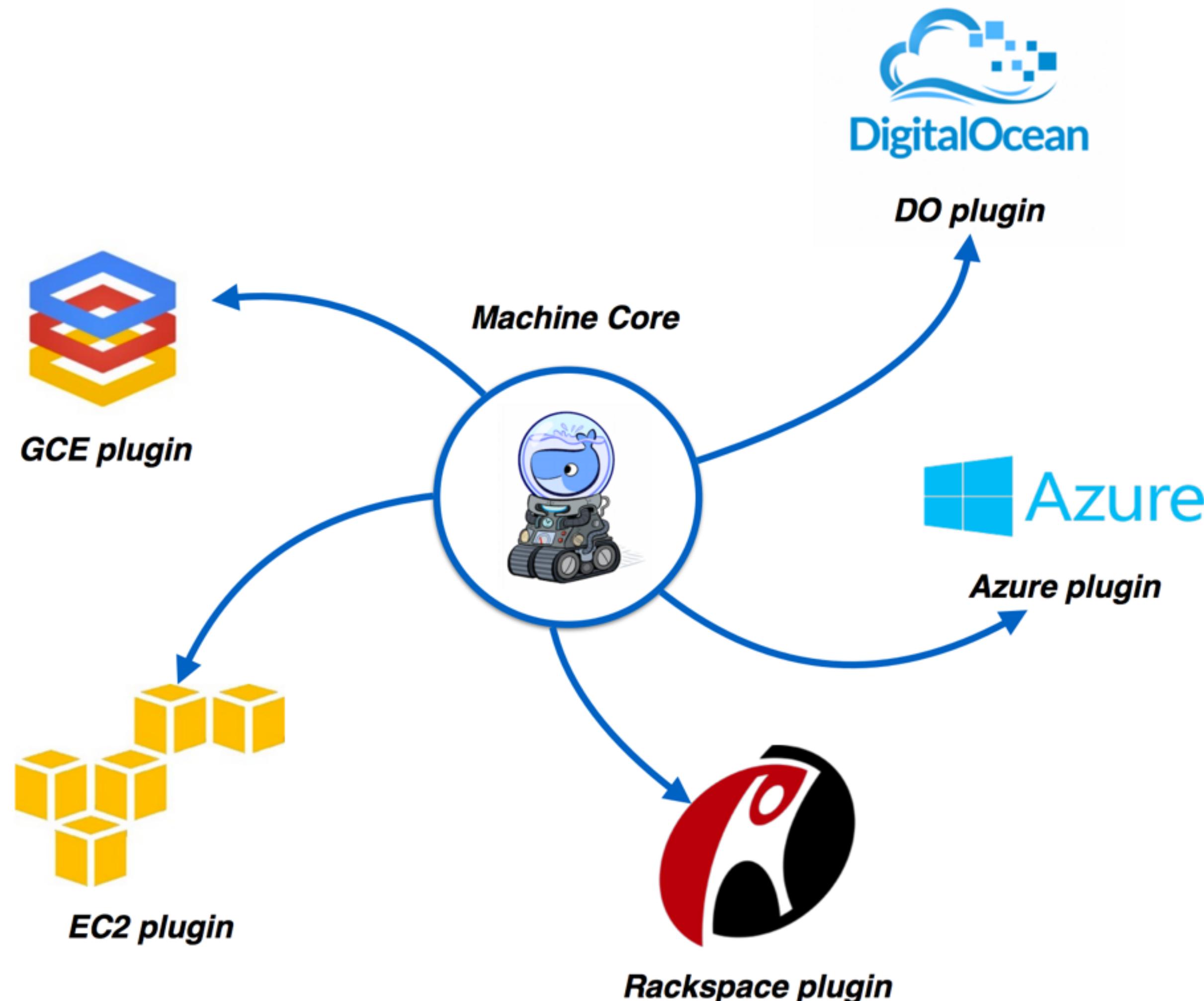
# Docker Machine

- 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



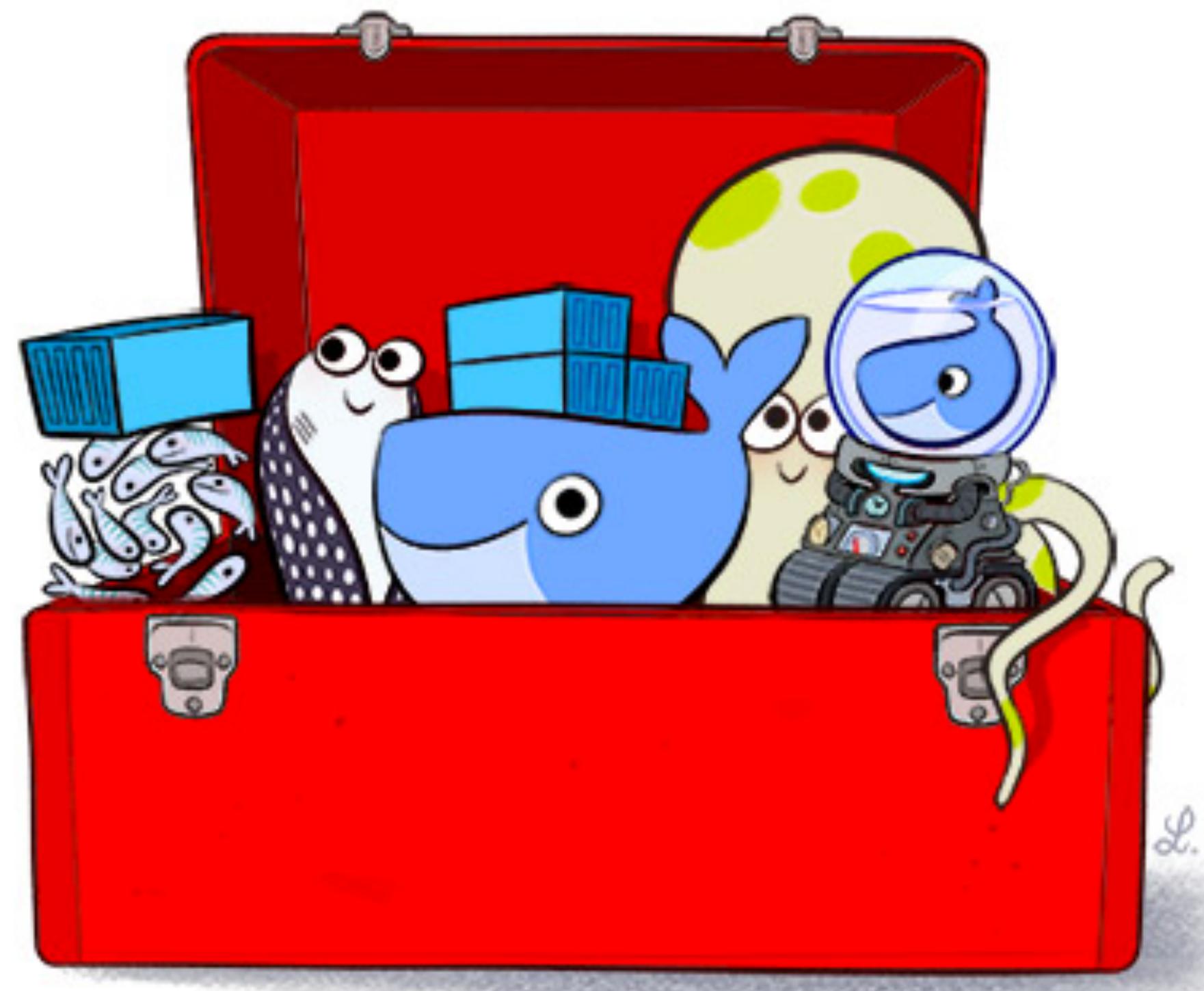


# Docker for Mac/Windows

- [beta.docker.com](http://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



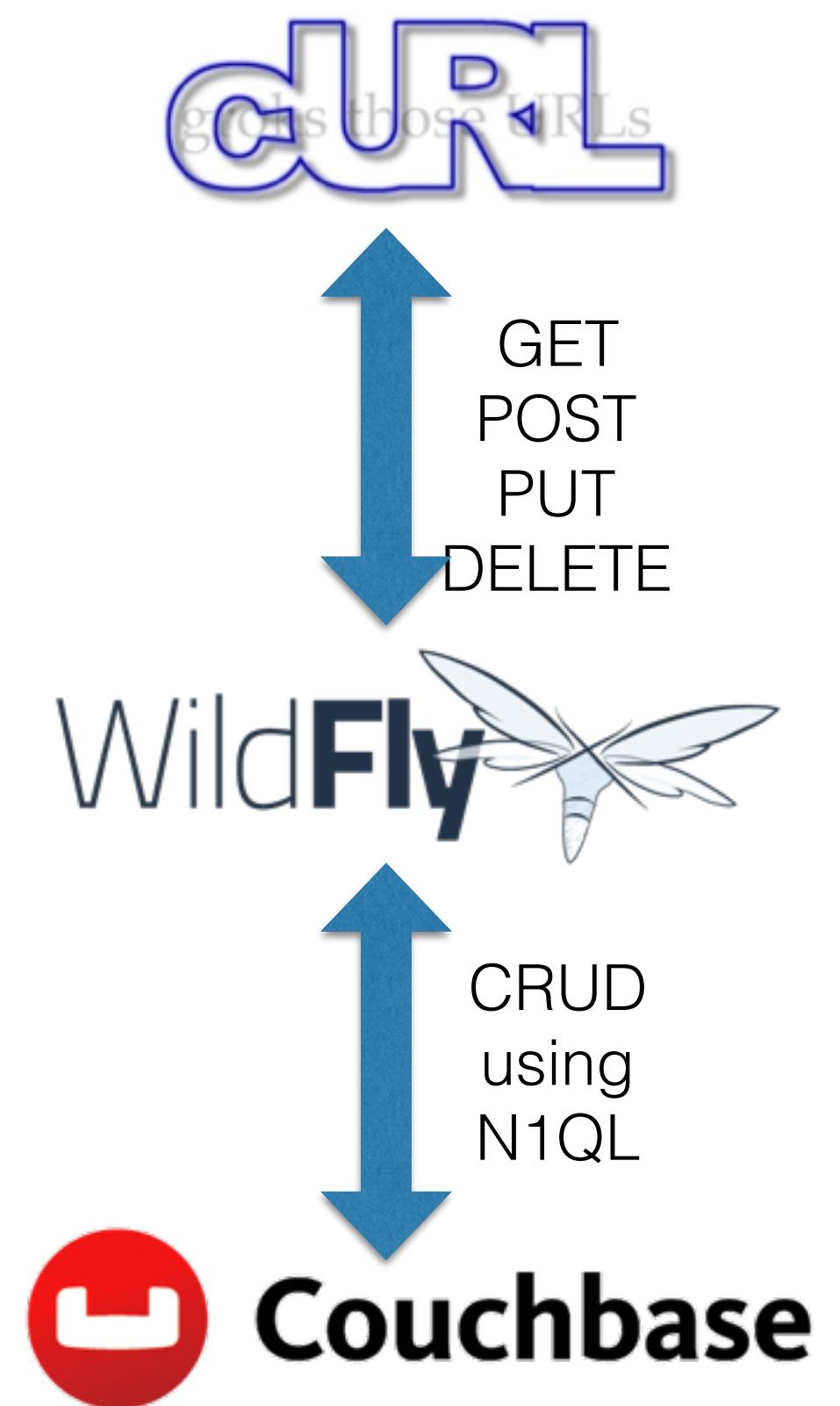


# Docker Compose - One Service

```
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

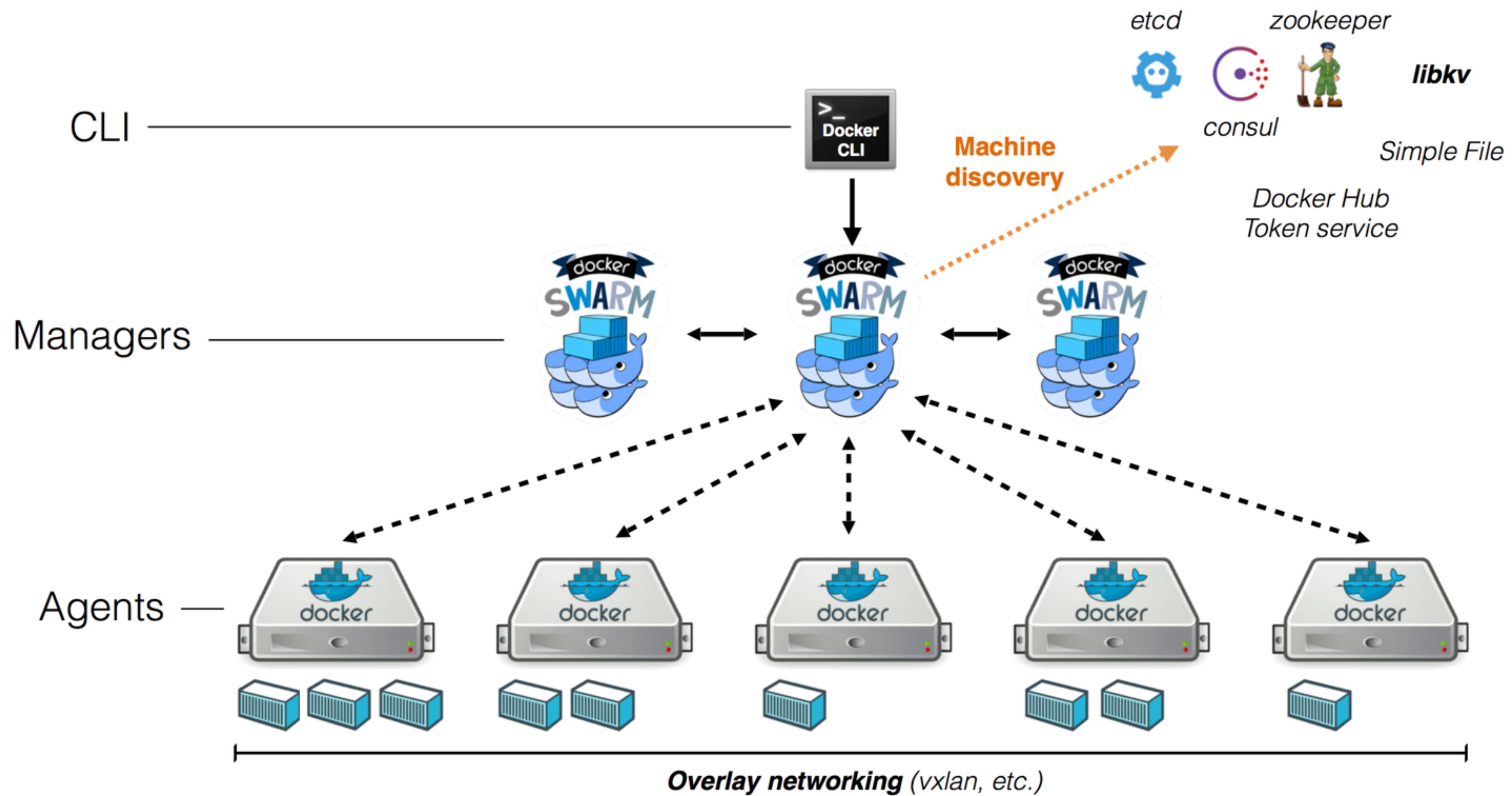
docker-compose up  
-f docker-compose.yml  
-f production.yml  
-d



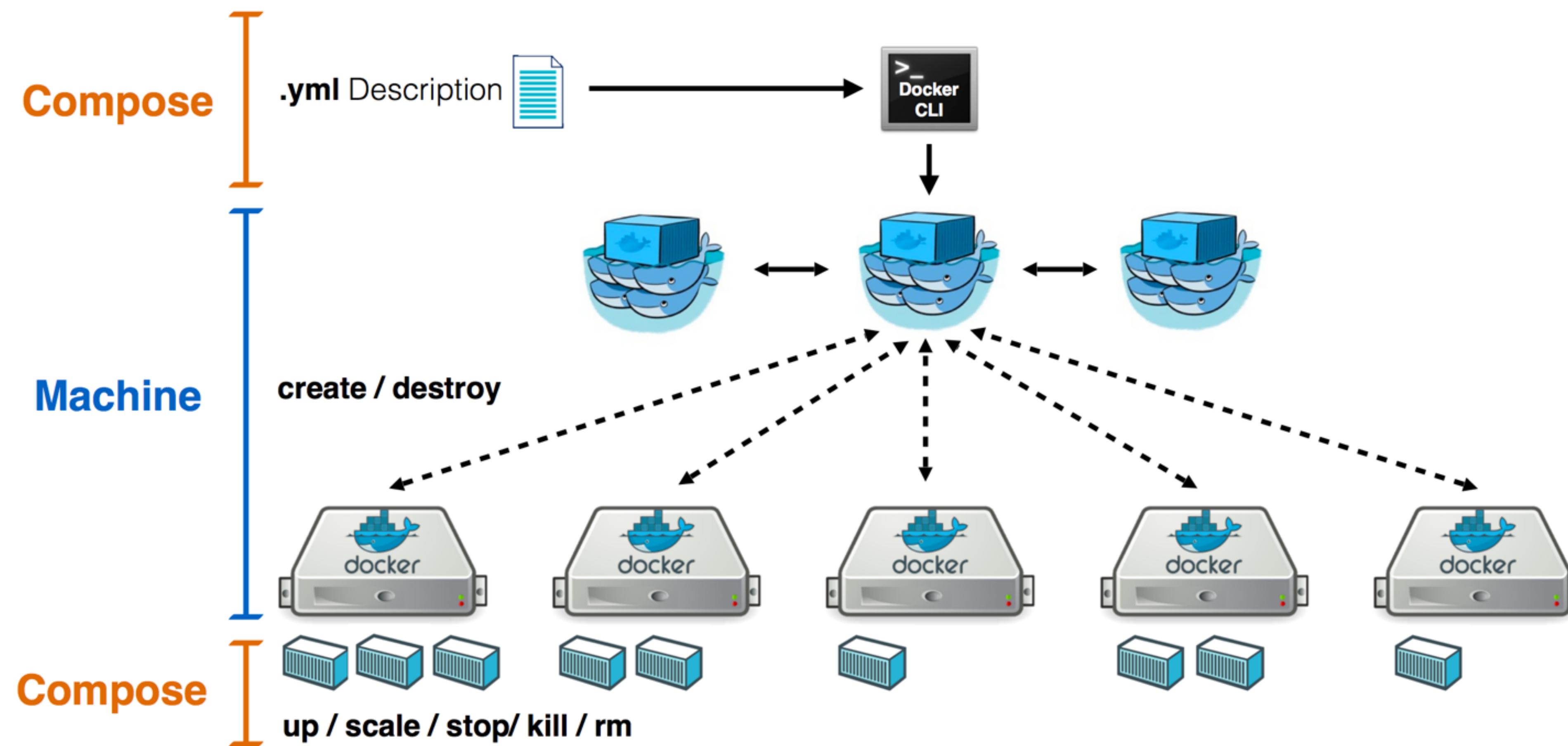
# Docker Swarm

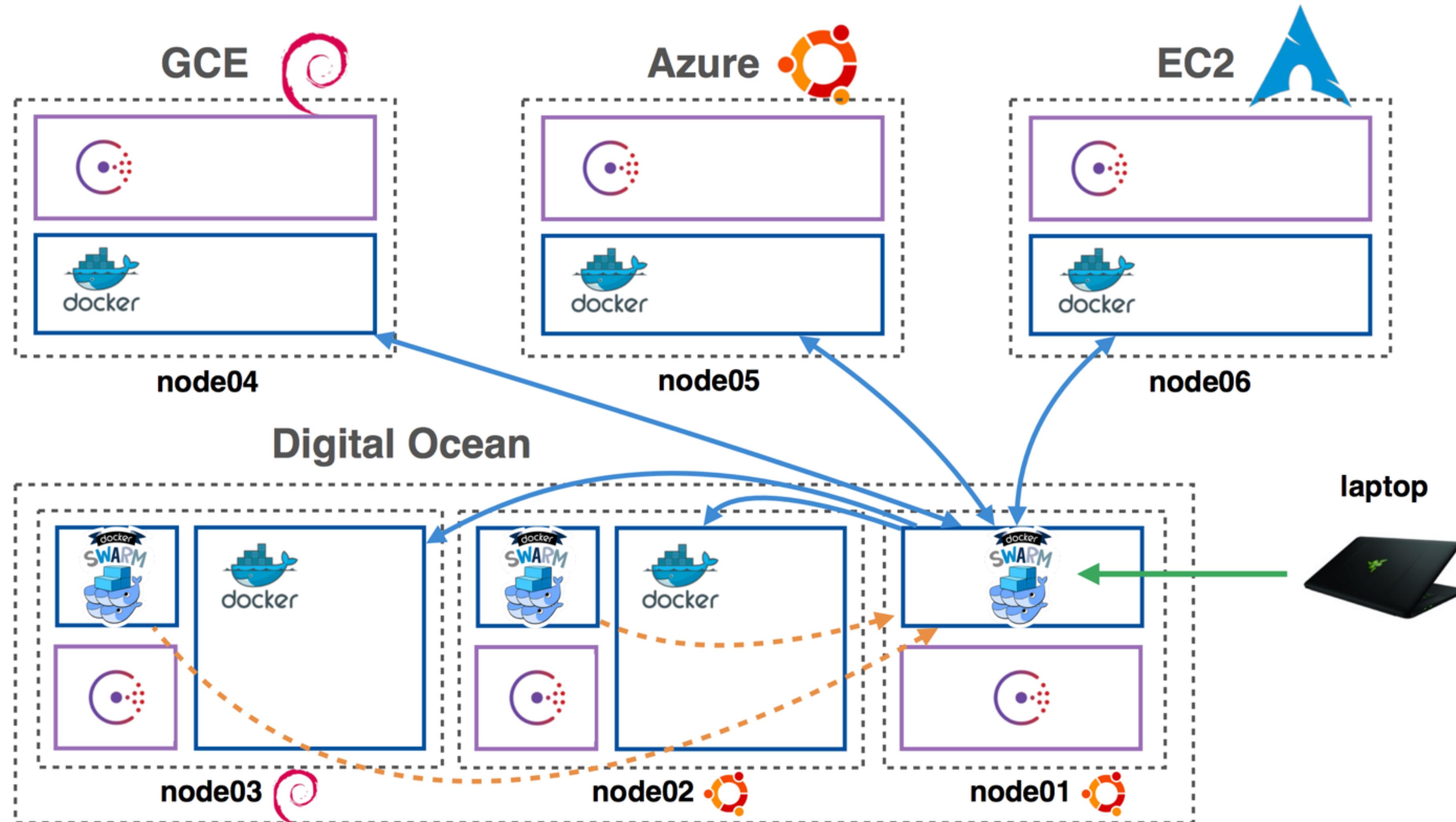
- Native clustering for Docker
- Provides a unified interface to a pool of Docker hosts
- Fully integrated with Machine and Compose
- Serves the standard Docker API
- 1.2 - Ready for production
  - Reschedule containers when a node fails
  - Better node management

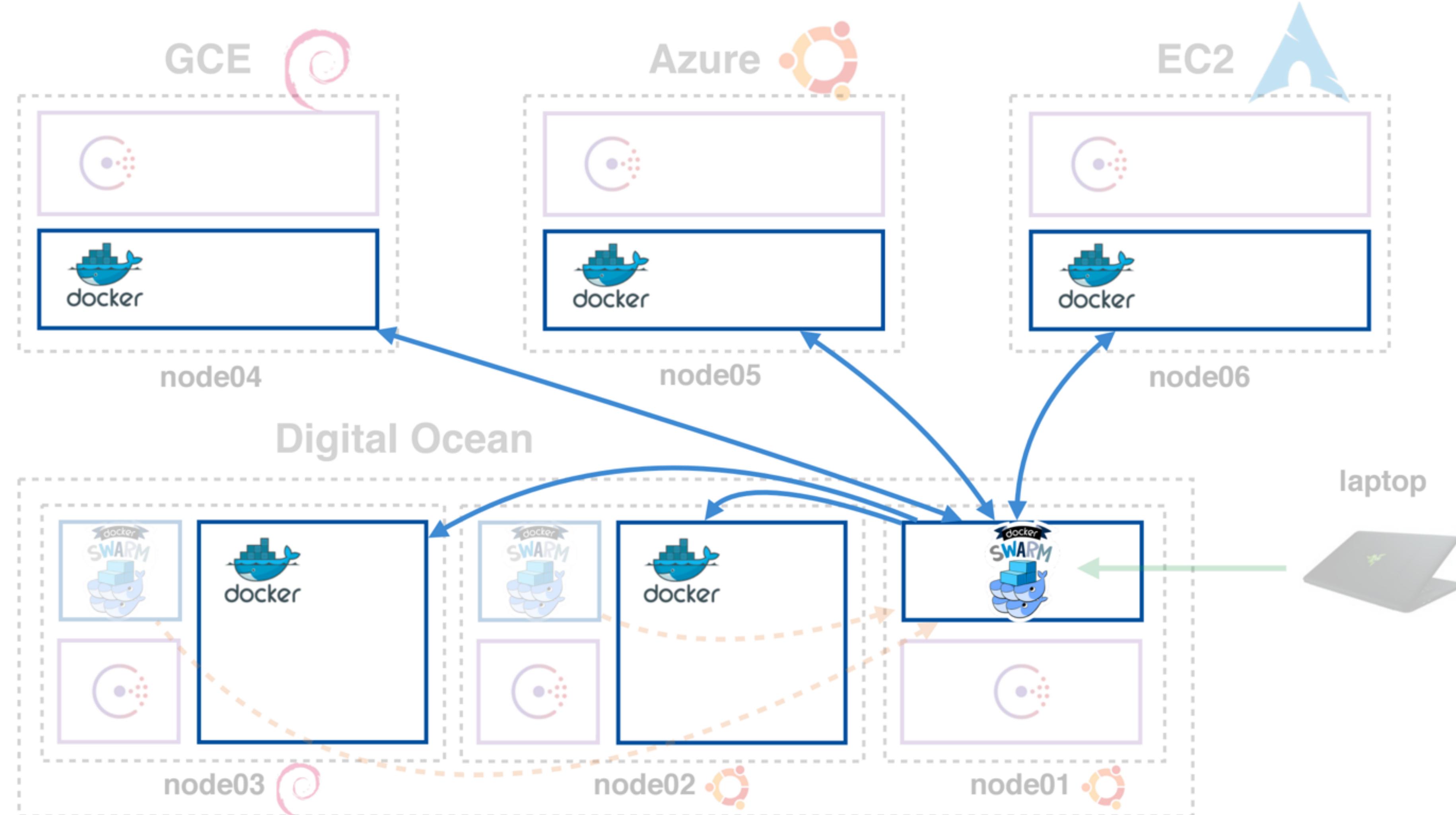
Stress tested on 1000 EC2 nodes, ~30k containers

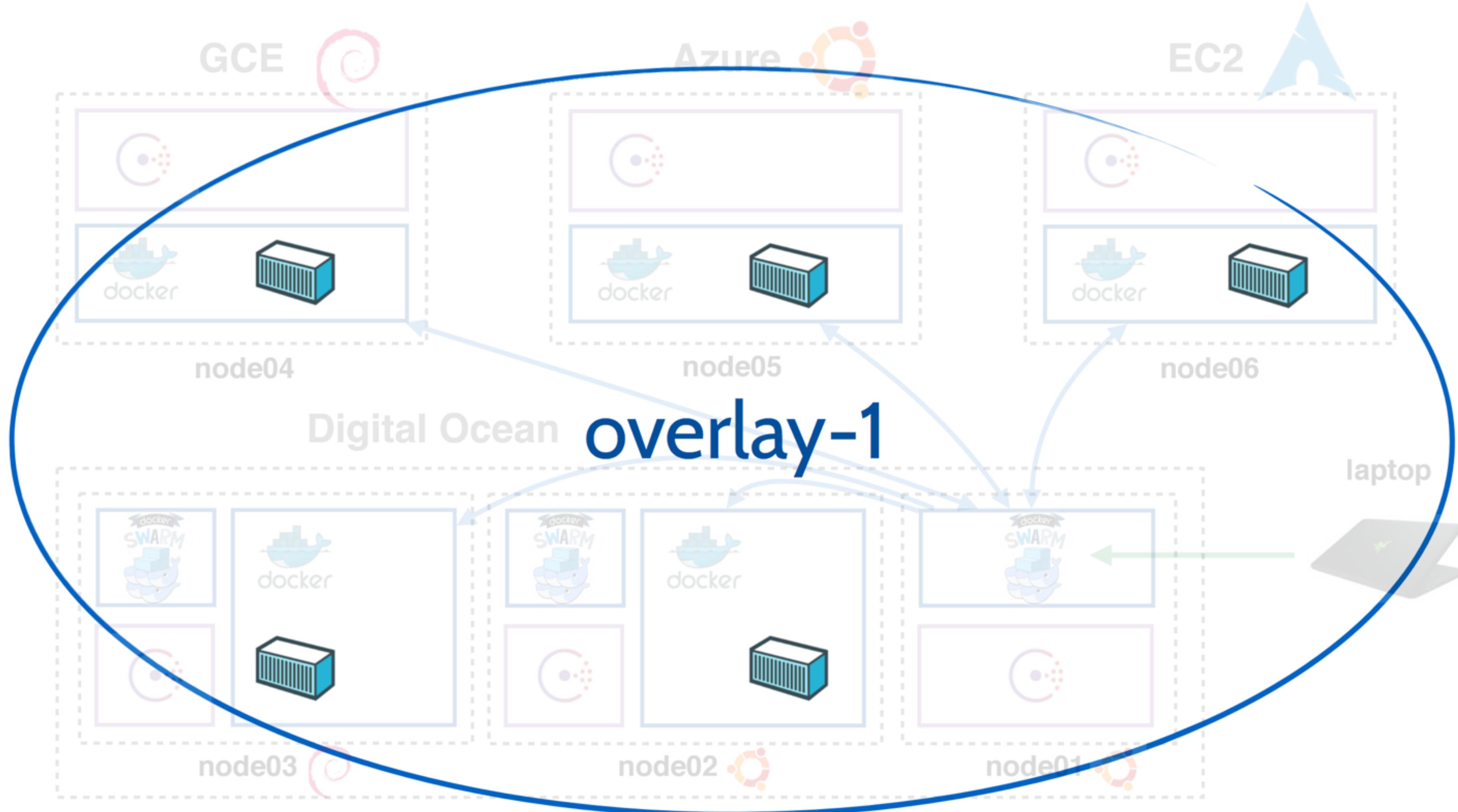


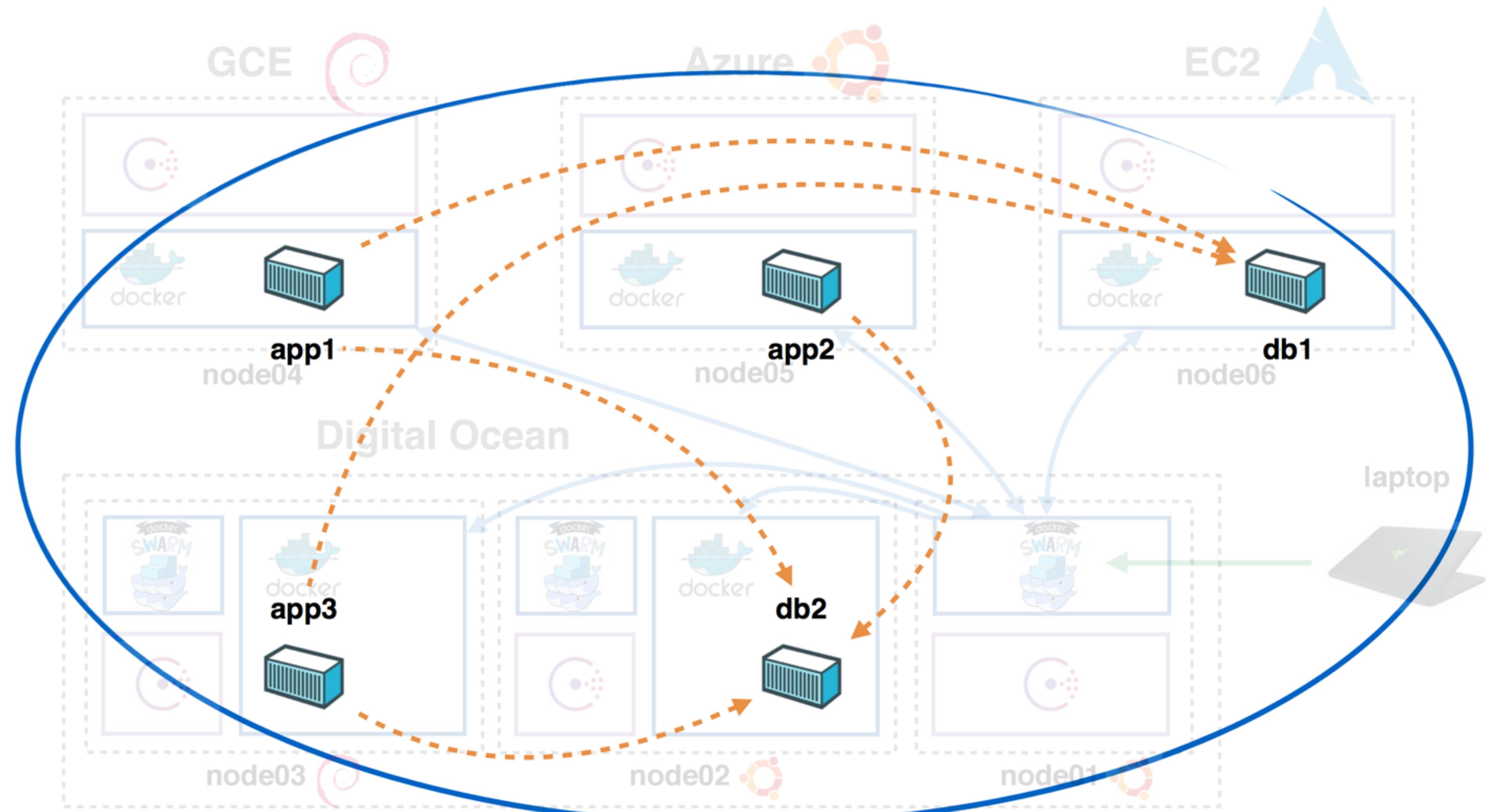
# Machine + Swarm + Compose

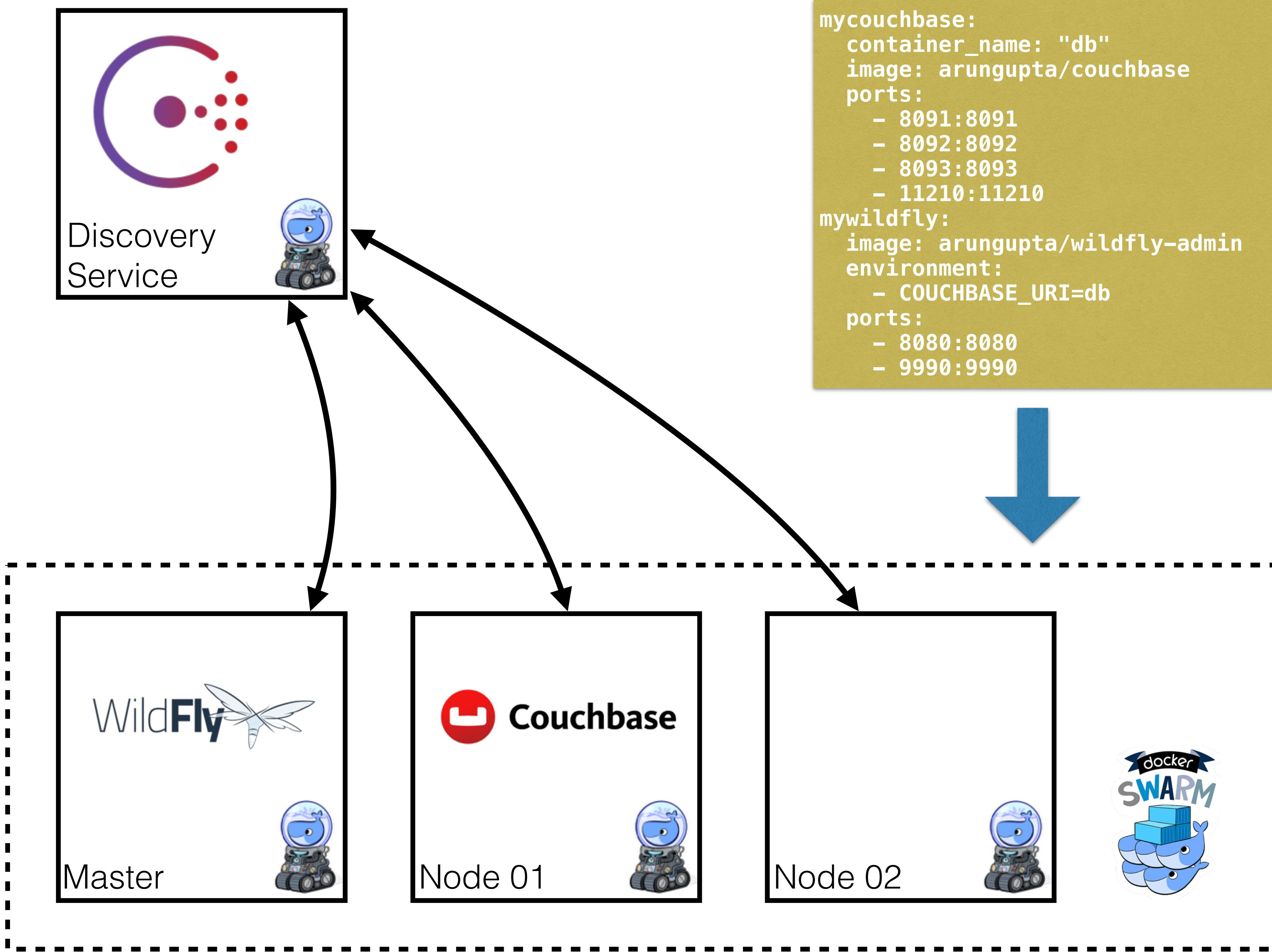














# Docker Cloud

- SaaS
- Build your images
- Deploy and manage across different clouds
  - Amazon, Digital Ocean, Microsoft, Azure, IBM SoftLayer
  - BYON



# Create a node cluster



Image arungupta/couchbase

Service name couchbase-1924aaf6

Image tag latest

Stack (No stack)

Deployment strategy Emptiest node

Number of containers 1

Deploy tags

Provider Amazon Web Services

Region us-west-1

VPC Auto

Subnet Auto

Ports

Container port	Protocol	Published	Node port
11207	tcp	<input type="checkbox"/>	--
11210	tcp	<input checked="" type="checkbox"/>	dynamic
11211	tcp	<input type="checkbox"/>	--
tcp		<input type="checkbox"/>	--
tcp		<input type="checkbox"/>	--
tcp		<input checked="" type="checkbox"/>	dynamic
tcp		<input checked="" type="checkbox"/>	dynamic
tcp		<input checked="" type="checkbox"/>	dynamic
tcp		<input checked="" type="checkbox"/>	dynamic

couchbase-1924aaf6 / COUC...

Stop Terminate Redeploy

▶ Running

34 minutes ago

Endpoints Logs Environment variables Volumes Terminal Timeline

Automatically refreshing

arungupta/couchbase:latest

/entrypoint.sh /opt/couchbase/...

32775->8091/tcp 32774->8092/tcp  
32773->8093/tcp 32772->11210/tcp

11207/tcp 11211/tcp 18091/tcp  
18092/tcp

10.7.0.2

off

None

Bridge

```
2016-03-21T22:07:13.351371182Z * upload completely sent off: 26 out of 26 bytes
2016-03-21T22:07:13.353756594Z < HTTP/1.1 200 OK
2016-03-21T22:07:13.353780275Z < Server: Couchbase Server
2016-03-21T22:07:13.353789344Z < Pragma: no-cache
2016-03-21T22:07:13.353797575Z < Date: Mon, 21 Mar 2016 22:07:13 GMT
2016-03-21T22:07:13.353805473Z < Content-Length: 0
2016-03-21T22:07:13.353816319Z < Cache-Control: no-cache
2016-03-21T22:07:13.353824677Z <
2016-03-21T22:07:13.354209316Z 100 26 0 0 100 26 0 6772 --:--:-- --:--:-- 8666
2016-03-21T22:07:13.354226834Z * Connection #0 to host 127.0.0.1 left intact
2016-03-21T22:07:13.357390925Z * Trying 127.0.0.1...
2016-03-21T22:07:13.357713200Z * Total % Received % Xferd Average Speed Time Time Current
2016-03-21T22:07:13.357765231Z Dload Upload Total Spent Left Speed
2016-03-21T22:07:13.357940902Z 0 0 0 0 0 0 0 0 --:--:-- --:--:-- 0* Conn
2016-03-21T22:07:13.358003091Z > POST /settings/web HTTP/1.1
2016-03-21T22:07:13.358057746Z > User-Agent: curl/7.40.0-DEV
2016-03-21T22:07:13.358110137Z > Host: 127.0.0.1:8091
2016-03-21T22:07:13.358162421Z > Accept: */*
2016-03-21T22:07:13.358217086Z > Content-Length: 50
2016-03-21T22:07:13.358268561Z > Content-Type: application/x-www-form-urlencoded
2016-03-21T22:07:13.358317827Z >
2016-03-21T22:07:13.358677410Z } [50 bytes data]
2016-03-21T22:07:13.359053352Z * upload completely sent off: 50 out of 50 bytes
2016-03-21T22:07:13.939725813Z < HTTP/1.1 200 OK
2016-03-21T22:07:13.939801035Z < Server: Couchbase Server
2016-03-21T22:07:13.939854726Z < Pragma: no-cache
2016-03-21T22:07:13.939901212Z < Date: Mon, 21 Mar 2016 22:07:13 GMT
2016-03-21T22:07:13.939945663Z < Content-Type: application/json
2016-03-21T22:07:13.939991254Z < Content-Length: 39
2016-03-21T22:07:13.940043843Z < Cache-Control: no-cache
2016-03-21T22:07:13.940088813Z <
2016-03-21T22:07:13.940306545Z { [39 bytes data]
2016-03-21T22:07:13.941109018Z 100 89 100 39 100 50 66 85 --:--:-- --:--:-- 85
2016-03-21T22:07:13.941177878Z * Connection #0 to host 127.0.0.1 left intact
2016-03-21T22:07:13.942283607Z {"newBaseUri":"http://127.0.0.1:8091/"}/entrypoint.sh couchbase-server
```

# Docker Cloud CLI

- brew install docker-cloud
- docker-cloud nodecluster create -t 1 --tag couchbase couchbase-node aws us-west-1 m3.large
- docker-cloud service create --tag couchbase -p 8091:8091 -p 8092:8092 -p 8093:8093 -p 11210:11210 arungupta/couchbase
- docker-cloud service start {SERVICE\_ID}
- docker-cloud service inspect {SERVICE\_ID} | jq ".container\_ports[0].endpoint\_uri" | sed 's/tcp/http/g'

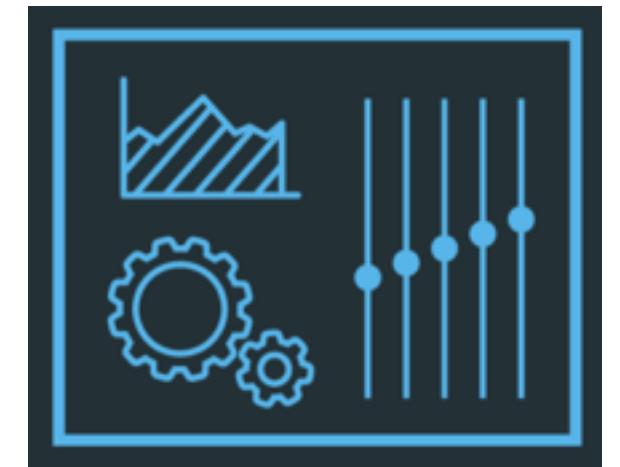


# Docker Registry

- Store and distribute Docker images
  - Control where images are stored
  - Own image distribution pipeline
  - Integrate image storage/distribution in dev workflow
- Docker Hub
  - Free-to-use and hosted
- Docker Trusted Registry
  - Commercially supported
  - RBAC, LDAP/AD integration, updates, etc

# Monitoring Docker Containers

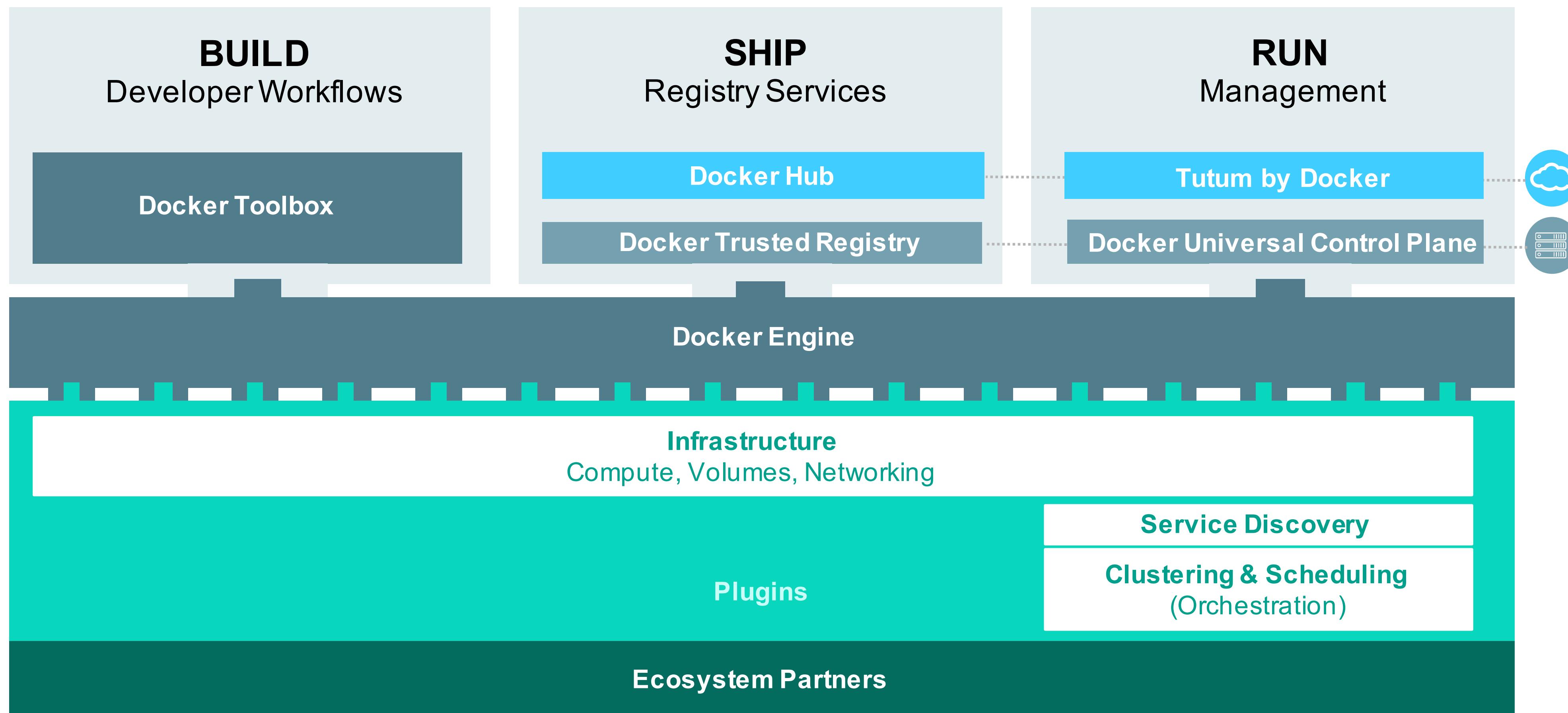
- `docker stats` command
  - LogEntries
- Docker Remote API: `/container/{container-name|cid}/stats`
- Docker Universal Control Plane
- cAdvisor
  - Prometheus
  - InfluxDB



cAdvisor



# Docker Mission



# IntelliJ IDEA Roadmap

- Docker
  - Coding assistance within Dockerfile files
  - Actions to start/stop/restart Docker machines
- Docker Compose
  - Coding assistance within Compose file
  - Displaying services when they are down
  - Displaying service configuration options in the Docker tool window
  - Gutter icons to start/stop/restart/build services within Compose file
  - Navigation between the Docker tool window and Compose file

# References

- Docker for Java Tutorial: [github.com/docker/labs/tree/master/java](https://github.com/docker/labs/tree/master/java)
- Docker Documentation: [docs.docker.com](https://docs.docker.com)
- IntelliJ Documentation: [jetbrains.com/intellij/docker](https://jetbrains.com/intellij/docker)