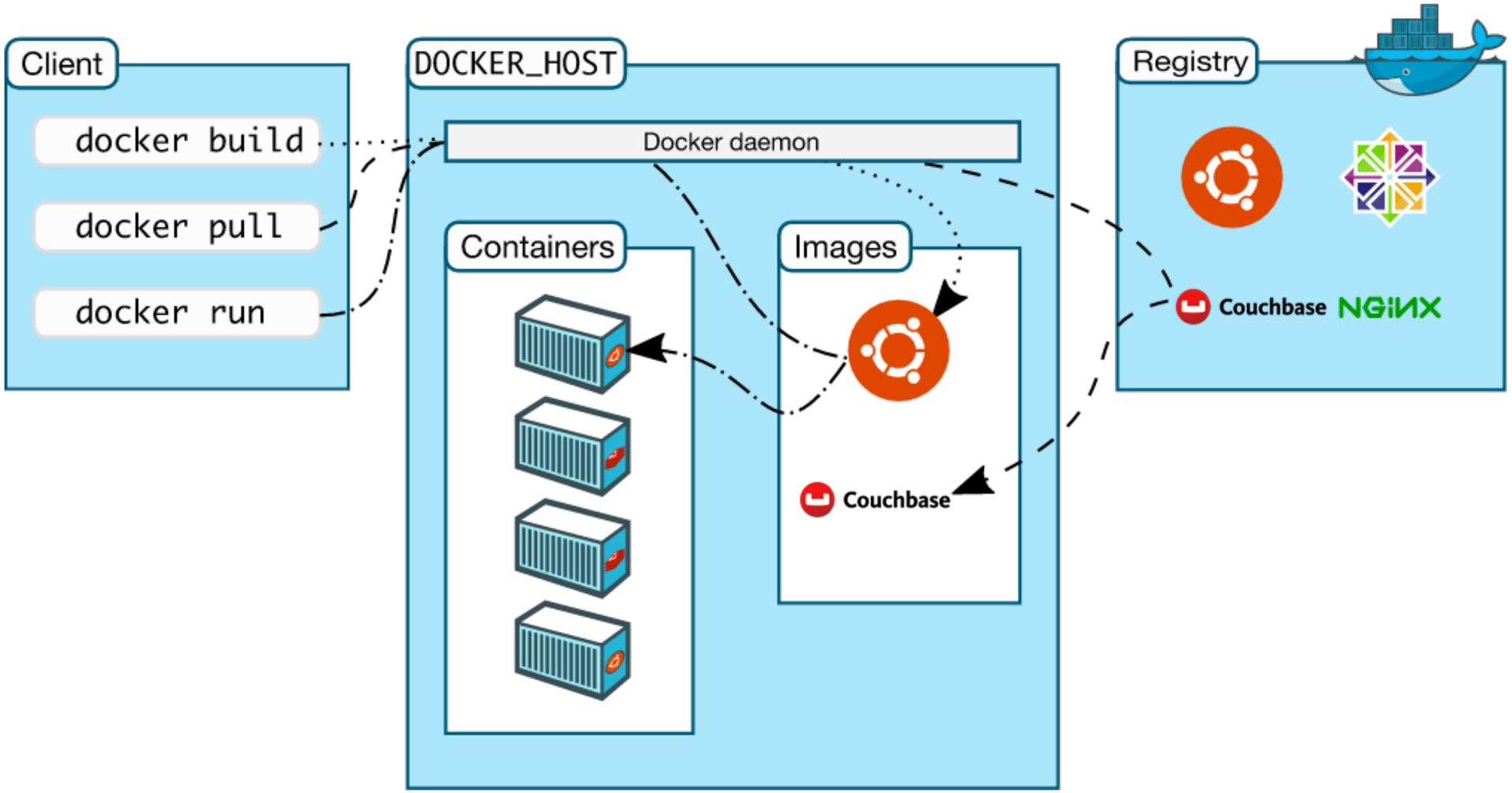
Service Discovery using Docker and Kubernetes

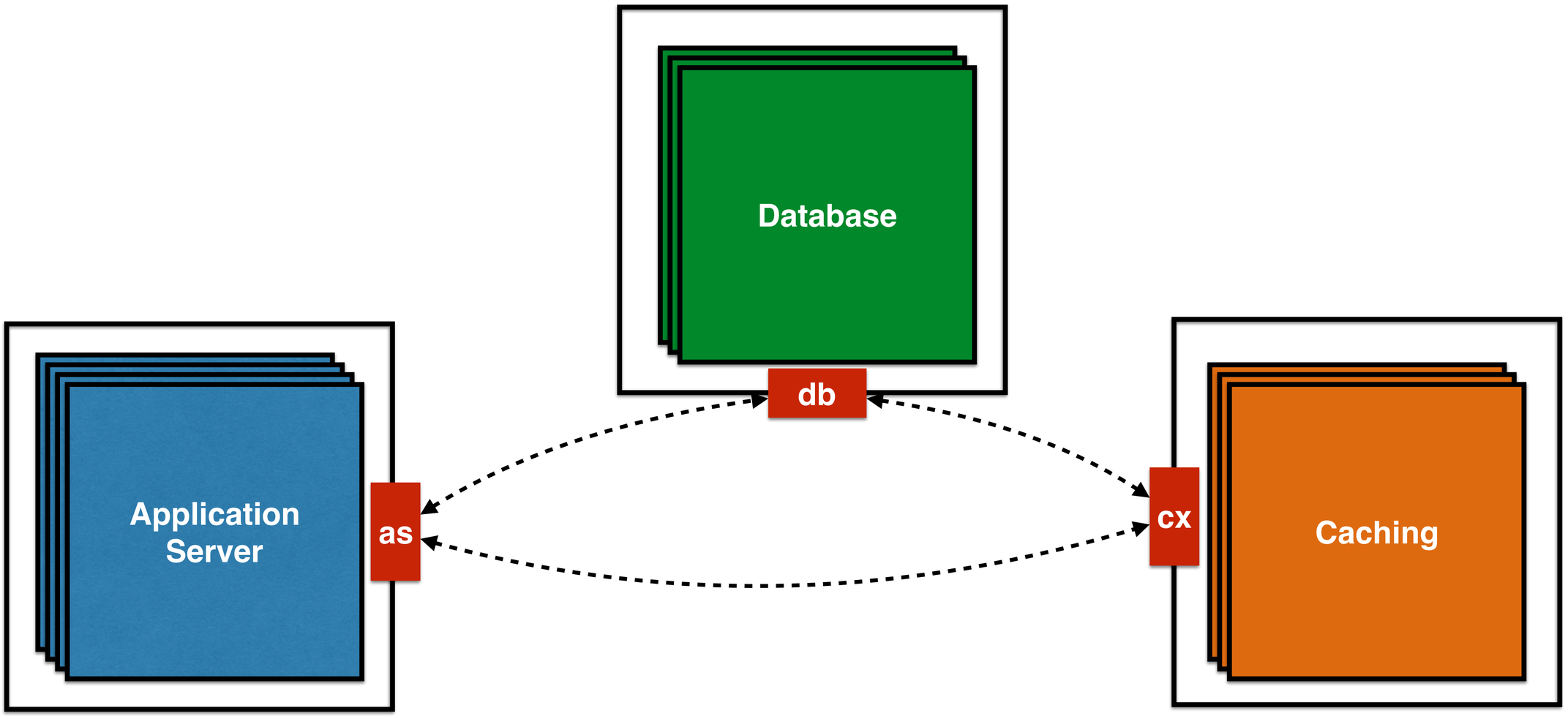
Arun Gupta, @arungupta

Couchbase

Docker Workflow

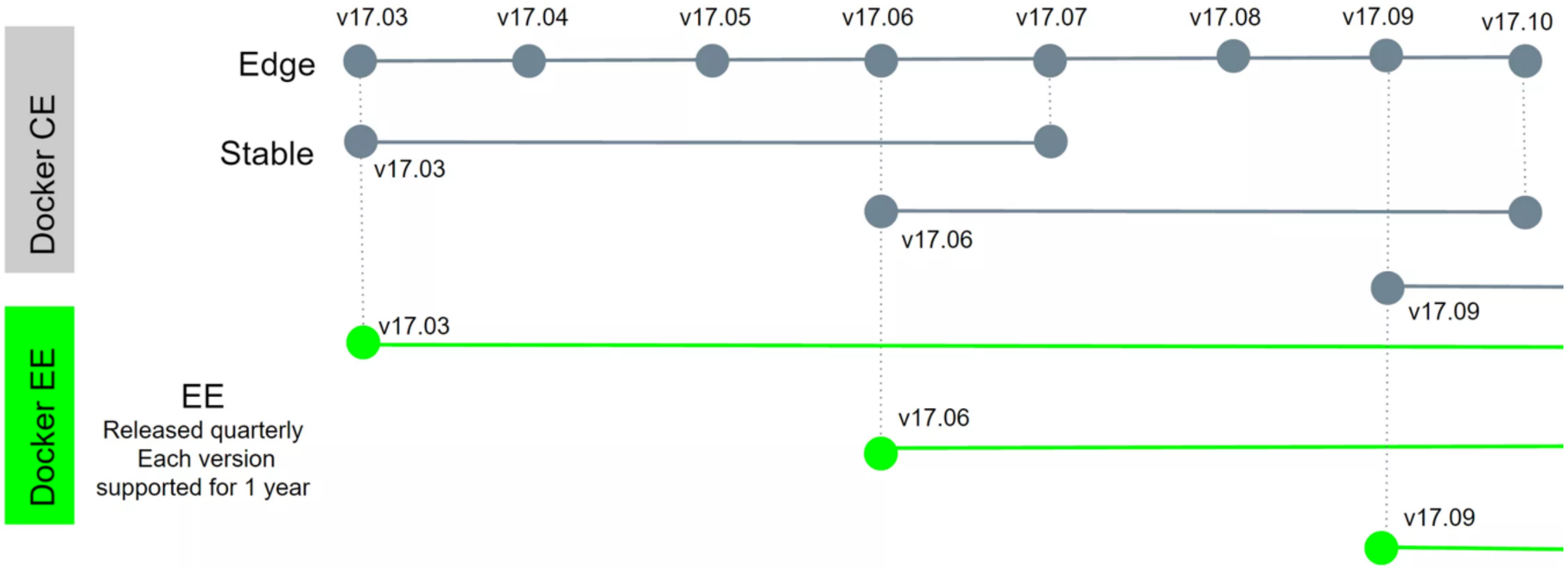


Why Service Discovery?

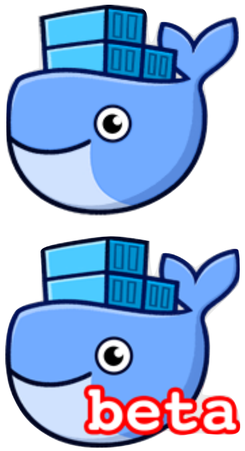


Development using Docker

- Docker Community Edition
  - Docker for Mac/Windows/Linux
  - Monthly edge and quarterly stable releases
  - Native desktop or cloud provider experience



Docker for Mac/Windows

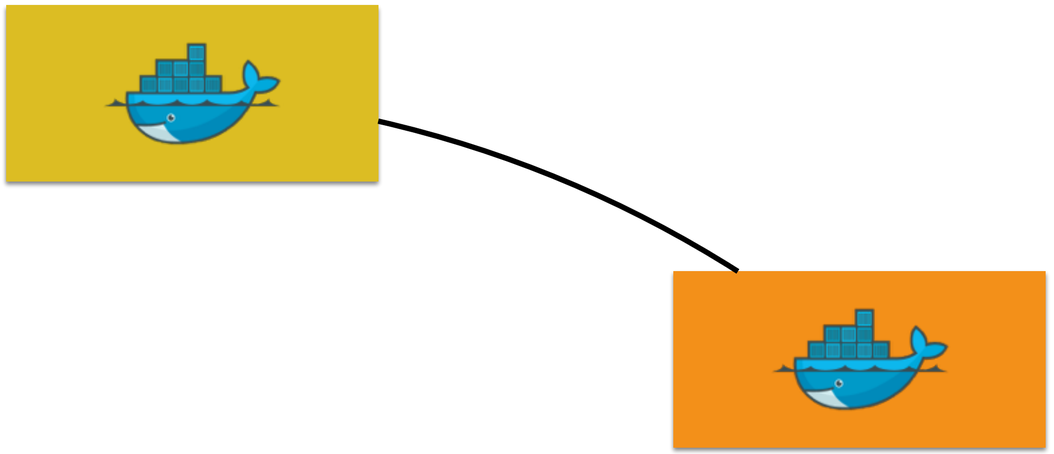


- Native application and UI
- Auto update capability
- No additional software required, e.g. VirtualBox
  - OSX: xhyve VM using **`Hypervisor.framework`**
  - Windows: Hyper-V VM
- Download: docker.com/getdocker
- Requires Yosemite 10.10+ or Windows 10 64-bit

Swarm-mode: Initialize

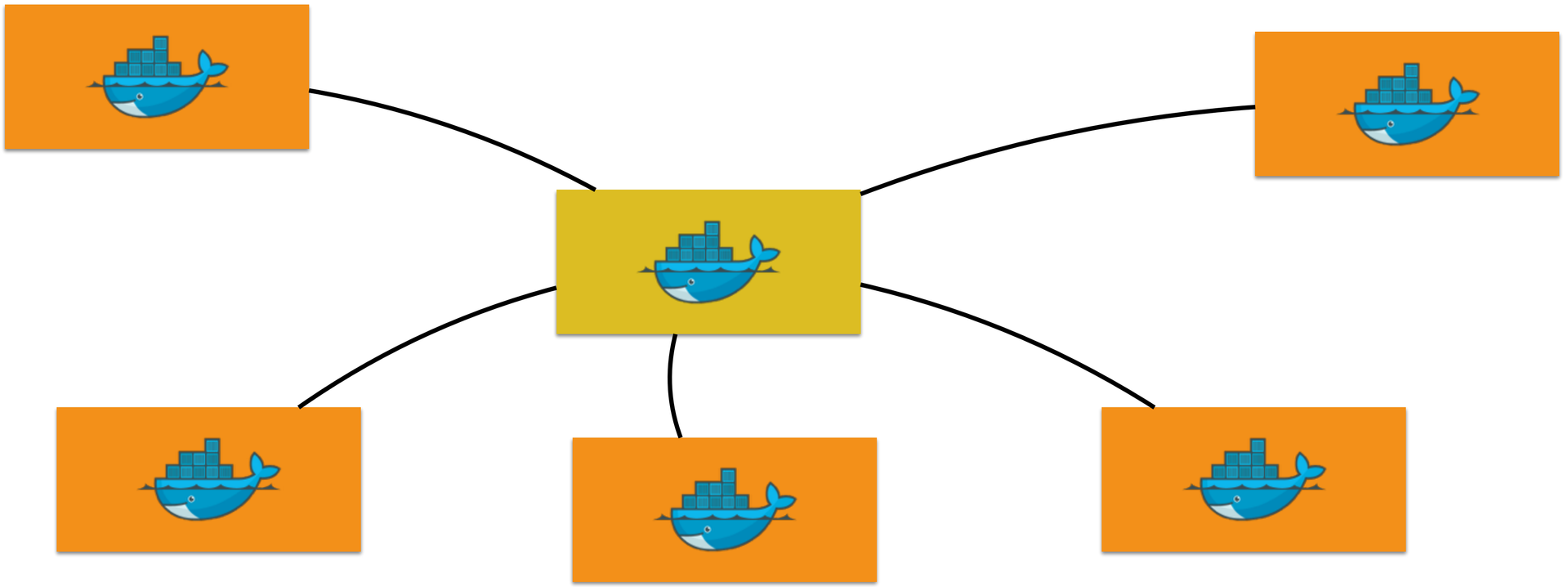
```
docker swarm init --listen-addr <ip>:2377 --secret <SECRET>
```

Swarm-mode: Add Worker



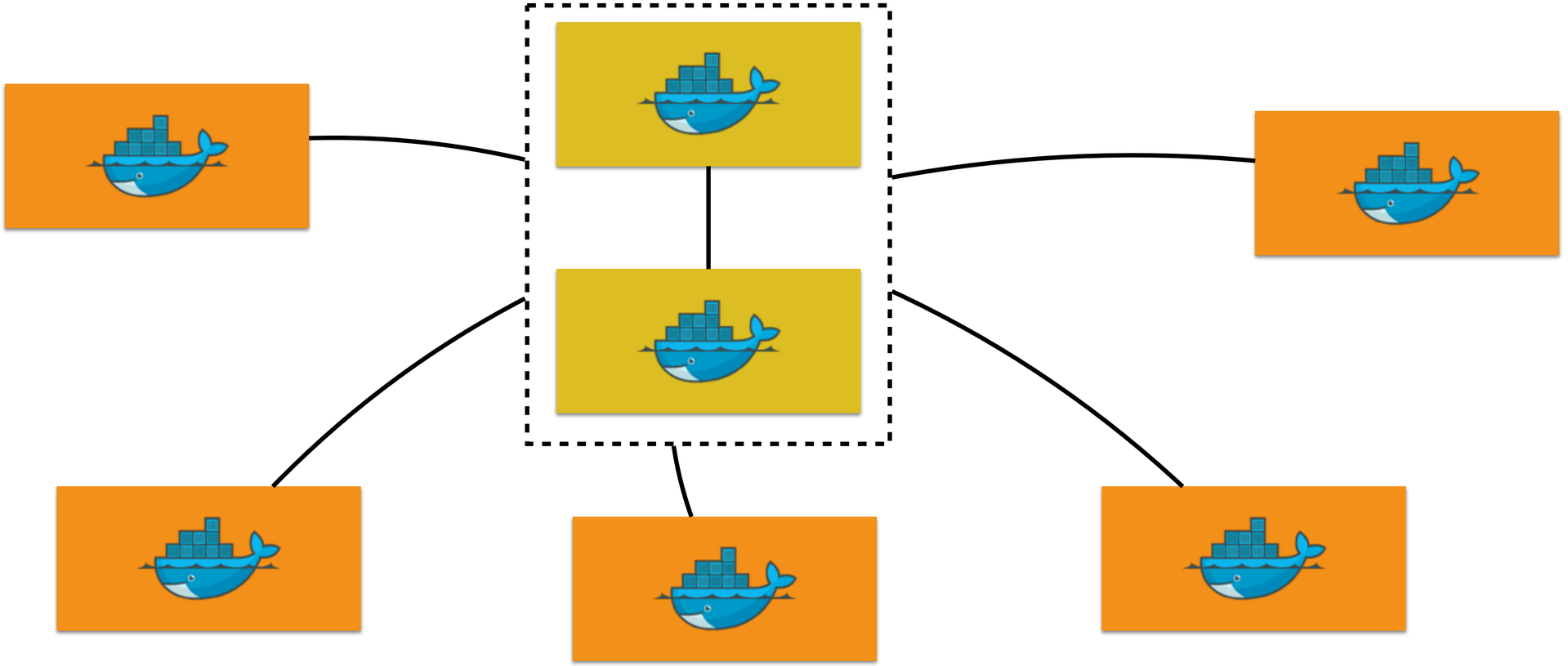
```
docker swarm join --secret <SECRET> <manager>:2377
```

Swarm-mode: Add More Workers



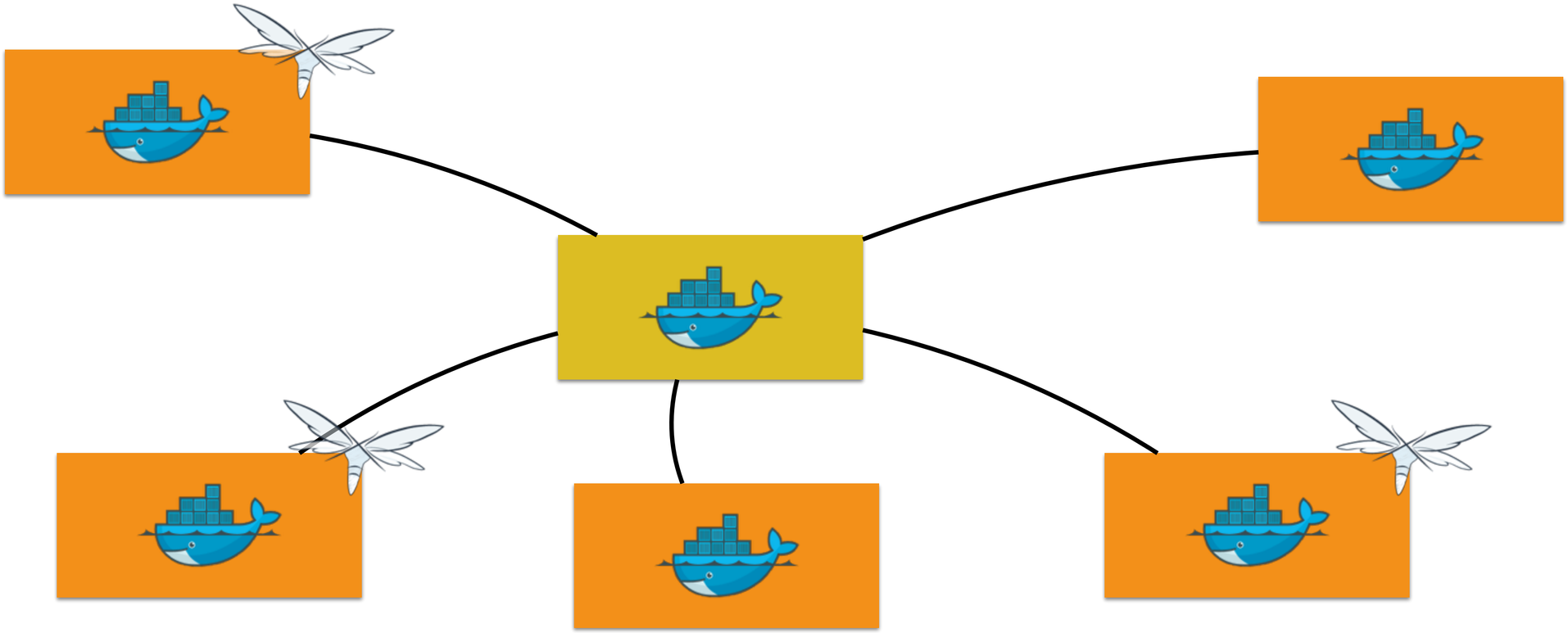
```
docker swarm join --secret <SECRET> <manager>:2377
```

Swarm-mode: Primary/Secondary Master



```
docker swarm join --manager --secret <SECRET> --listen-addr
<master2>:2377 <master1>:2377
```

Swarm-mode: Replicated Service

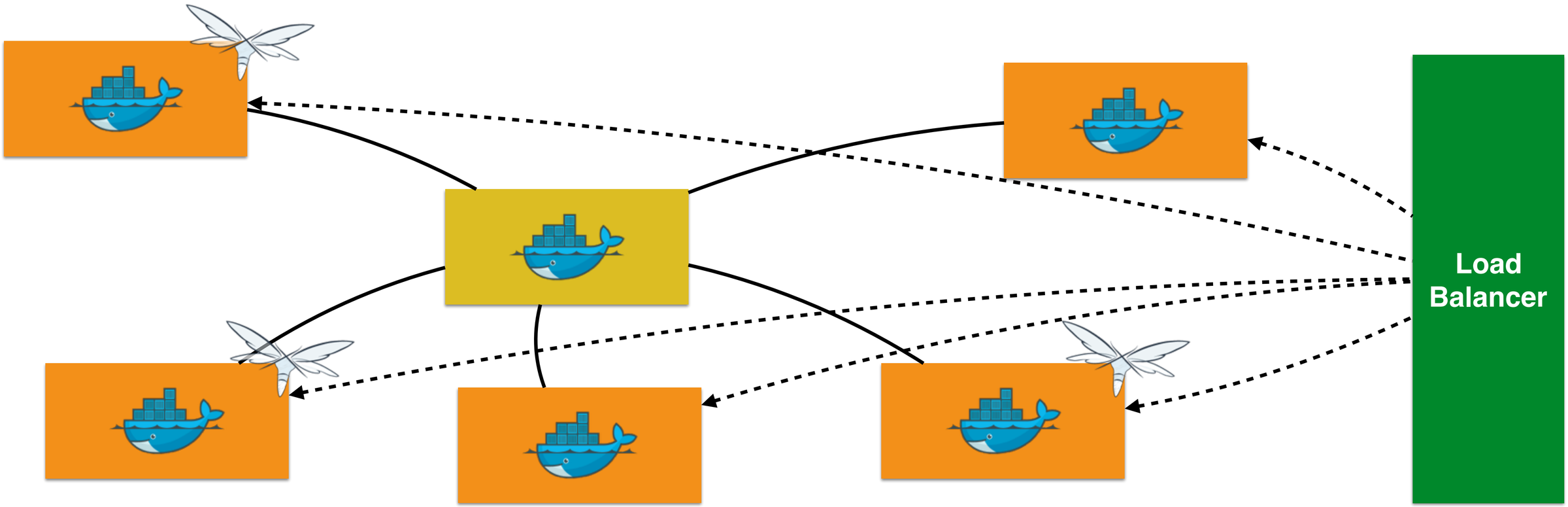


```
docker service create --replicas 3 --name web jboss/wildfly
```

Swarm-mode: Routing Mesh

- Load balancers are host-aware, not container-aware
- Swarm mode introduces container-aware routing mesh
- Reroutes traffic from any host to a container
  - Reserves a Swarm-wide ingress port
  - Uses DNS-based service discovery

Swarm-mode: Routing Mesh




```
docker service create --replicas 3 --name web -p 8080:8080 jboss/
wildfly
```

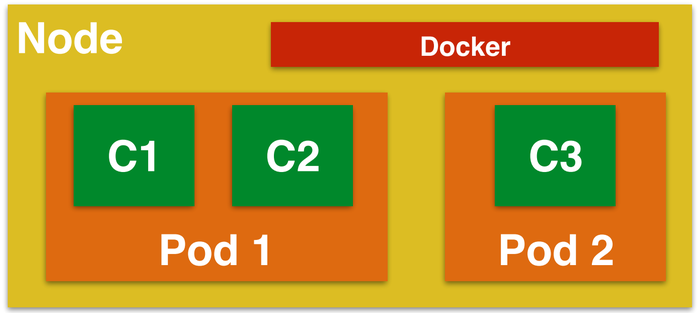
Service Discovery with Docker

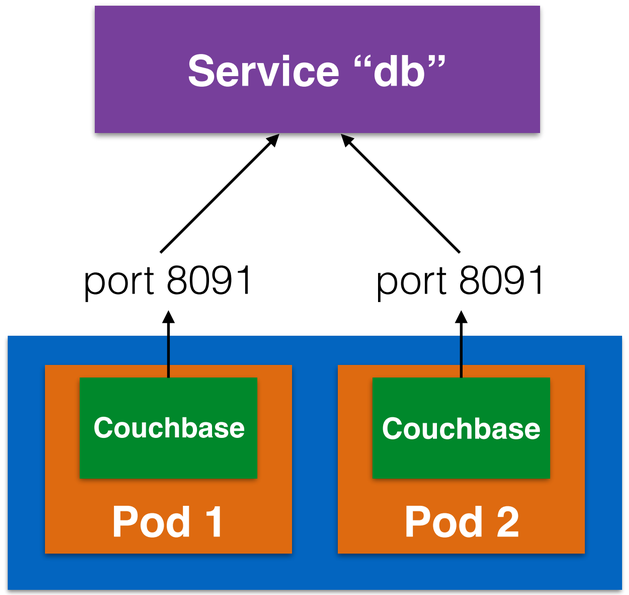
```
1   version: "3"
2   services:
3     db:
4       image: arungupta/couchbase:travel
5       ports:
6         - 8091:8091
7         - 8092:8092
8         - 8093:8093
9         - 11210:11210
10    web:
11      image: arungupta/wildfly-couchbase-javaee:travel
12      environment:
13        - COUCHBASE_URI=db
14      ports:
15        - 8080:8080
16        - 9990:9990
```

```
docker stack deploy --compose-file=docker-compose.yml webapp
```

Kubernetes Concepts

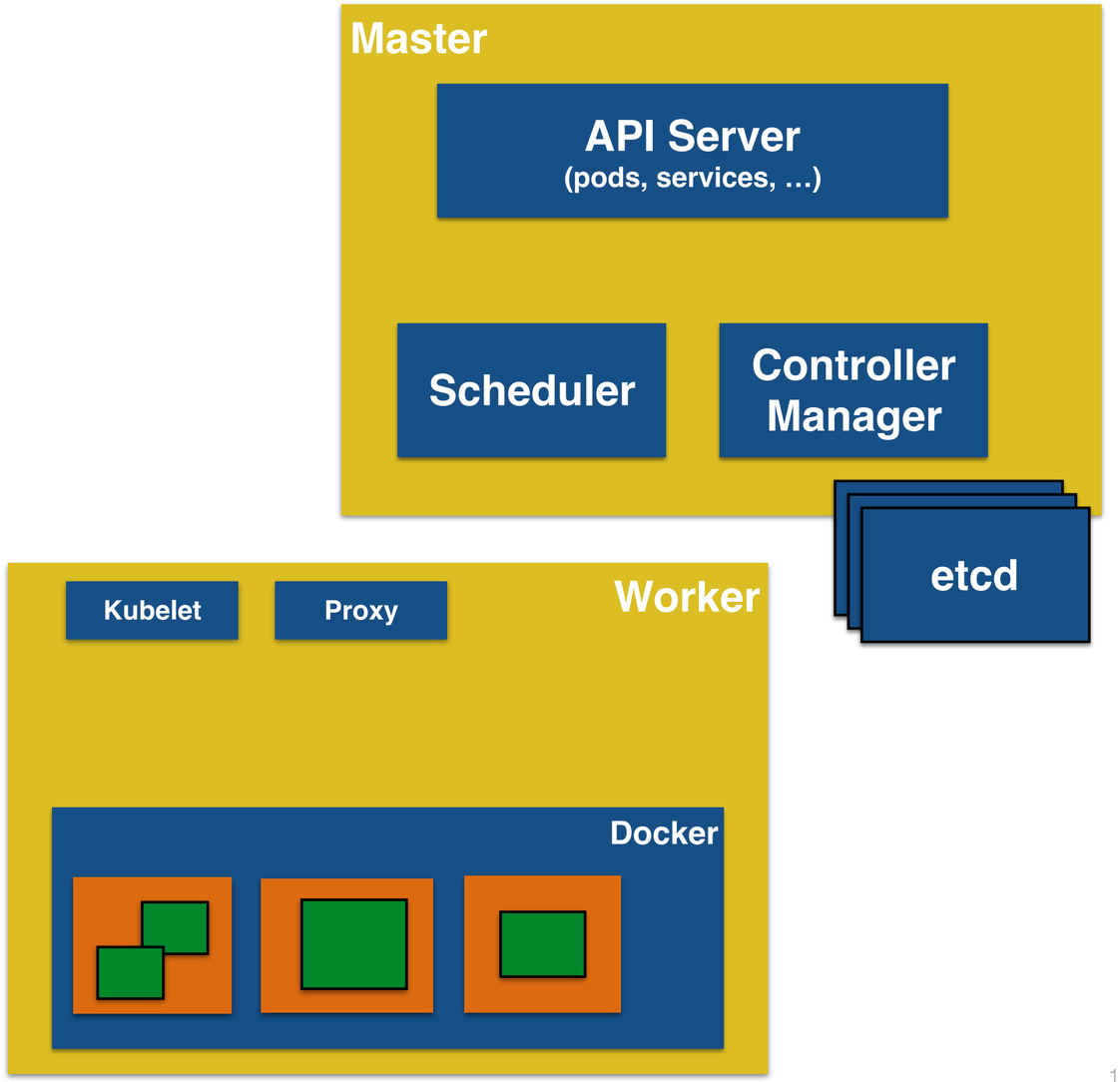
- **Pods**: collocated group of Docker containers that share an IP and storage volume
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to organize and select group of objects
- **Replica Set**: manages the lifecycle of pods and ensures specified number are running

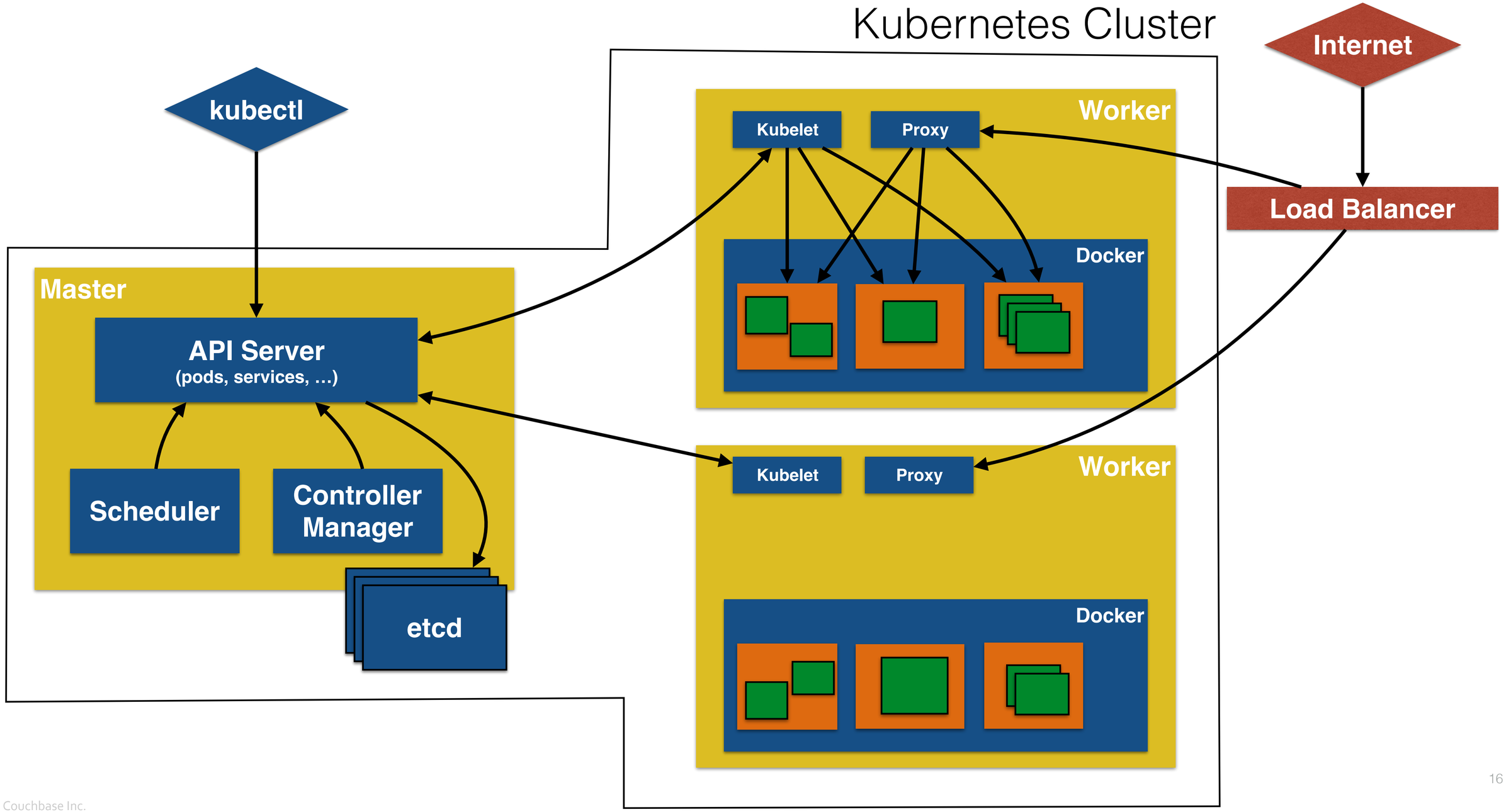




Core Concepts: Kubernetes

- **Node**: Machine or VM in the cluster
- **Master**: Central control plane, provides unified view of the cluster
  - **etcd**: distributed key-value store used to persist Kubernetes system state
- **Worker**: Docker host running *kubelet* (node agent) and *proxy* services
  - Runs pods and containers
  - Monitored by *systemd* (CentOS) or *monit* (Debian)





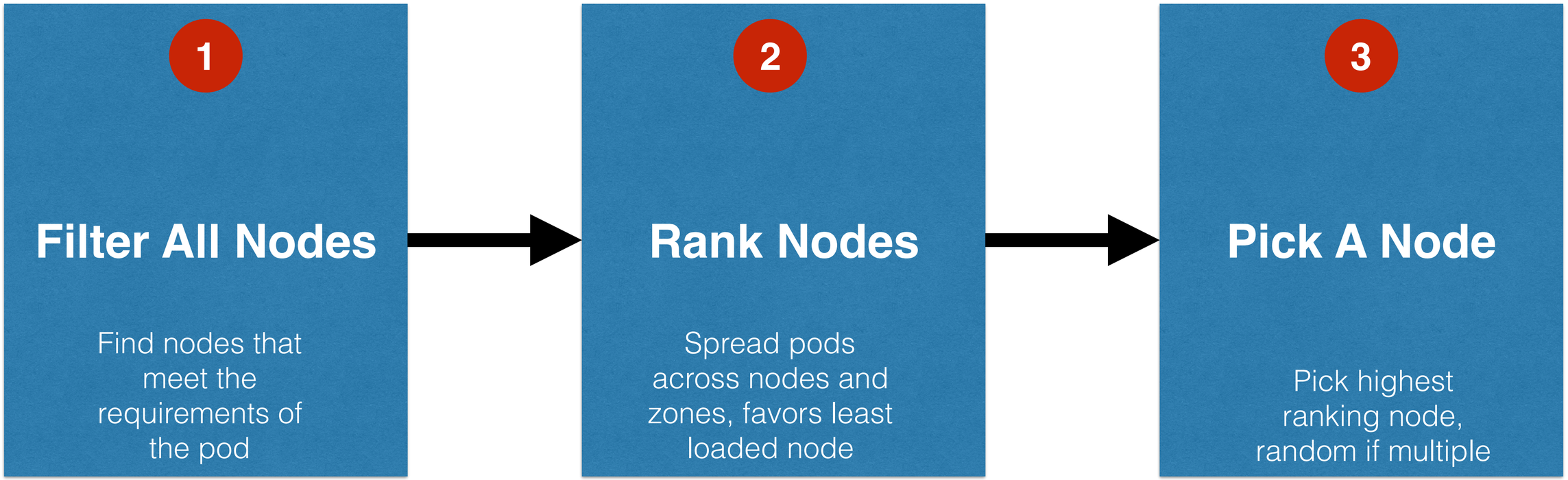
Development using Kubernetes

- Single node cluster
  - minikube
- Multi-node cluster
  - kops
  - kube-aws (CoreOS + AWS)
  - kube-up (deprecated)
  - Google Cloud, Azure, Tectonic, …

kubectl

- Controls the Kubernetes cluster manager
- `kubectl get pods` or `minions`
- `kubectl create -f <filename>`
- `kubectl update` or `delete`
- `kubectl resize –replicas=3 replicaset <name>`

Kubernetes Scheduling Algorithm

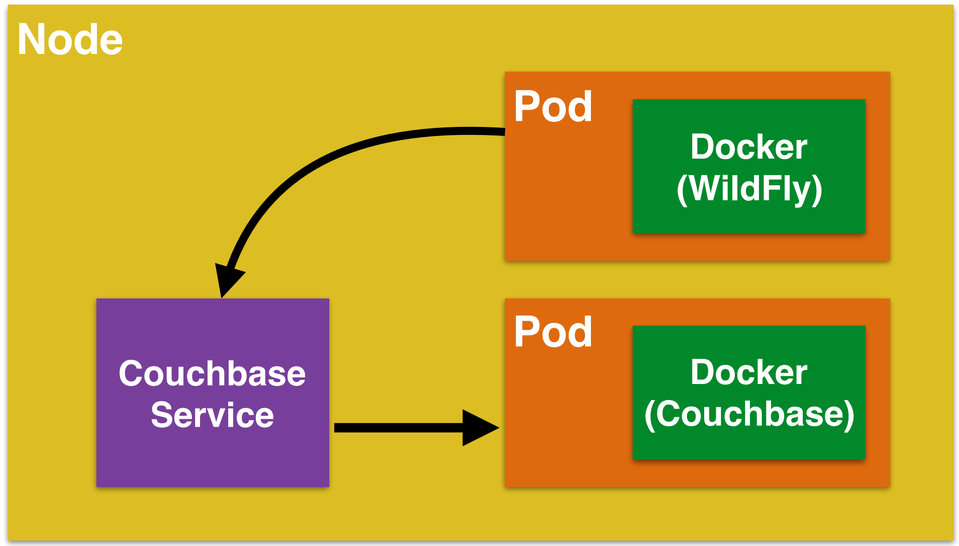


Kubernetes Configuration File

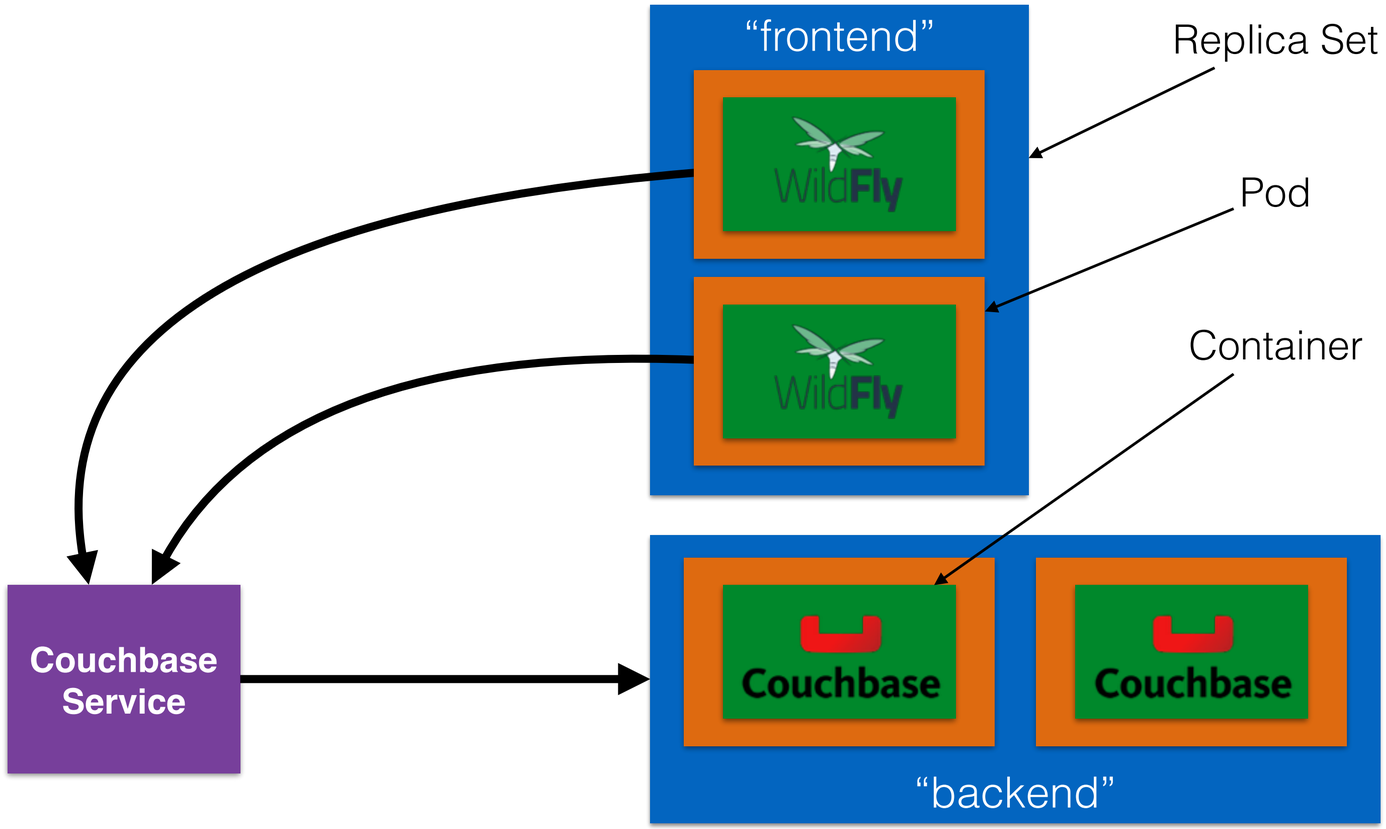
```yaml
apiVersion: v1
kind: Service
metadata:
  name: couchbase-service
spec:
  selector:
    app: couchbase-rs-pod
  ports:
    - name: admin
      port: 8091
    - name: views
      port: 8092
    - name: query
      port: 8093
    - name: memcached
      port: 11210
---
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: couchbase-rs
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: couchbase-rs-pod
    spec:
      containers:
      - name: couchbase
        image: arungupta/couchbase:travel
        ports:
        - containerPort: 8091
        - containerPort: 8092
        - containerPort: 8093
        - containerPort: 11210
---
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: wildfly-rs
  labels:
    name: wildfly
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rs-pod
        image: arungupta/wildfly-couchbase-javaee:travel
        env:
          name: COUCHBASE_URI
          value: couchbase-service
        ports:
        - containerPort: 8080

```

Services



Services



References

- Docker: github.com/docker/labs/tree/master/developer-tools/java
- Kubernetes: github.com/arun-gupta/kubernetes-java-sample

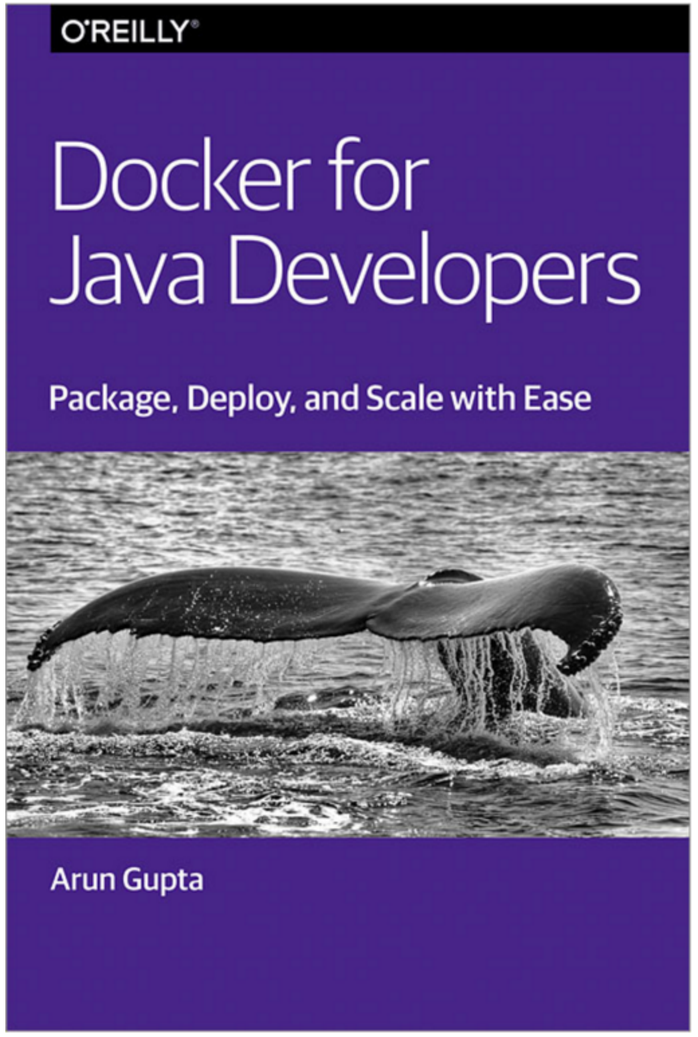
ONLINE COURSE

Arun Gupta

KUBERNETES FOR JAVA DEVELOPERS

September 23 | 10AM-12PM

bit.ly/kubejava



bit.ly/dockerjava

Thanks!

Arun Gupta, @arungupta

github.com/javaee-samples/docker-java/tree/master/slides