Package your Java Application using Docker and Kubernetes

Arun Gupta, @arungupta

Docker Captain

Java Champion

JavaOne Rock Star (4 years)

NetBeans Dream Team

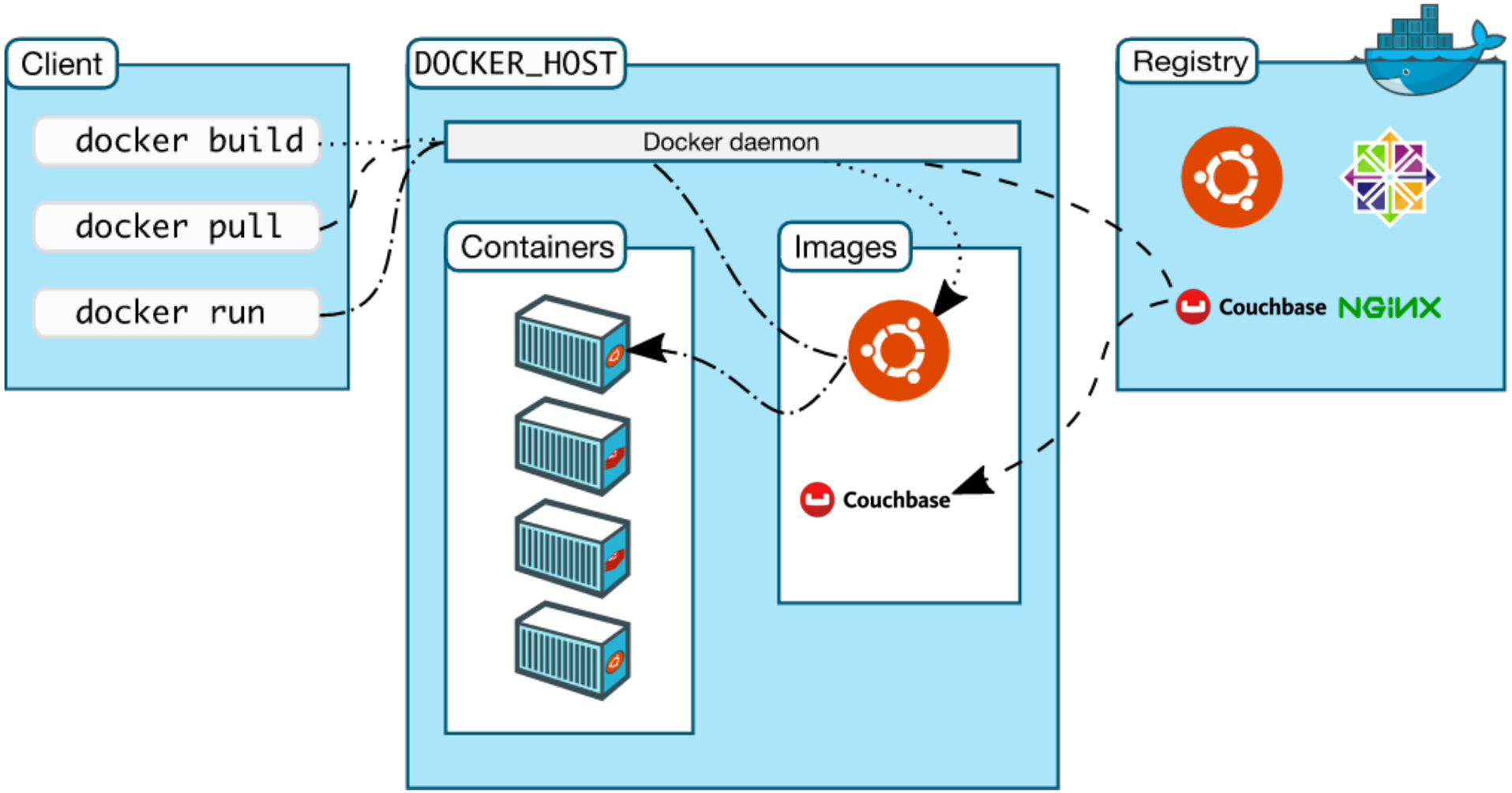
Silicon Valley JUG Leader

Author

Runner

Lifelong learner

Docker Workflow



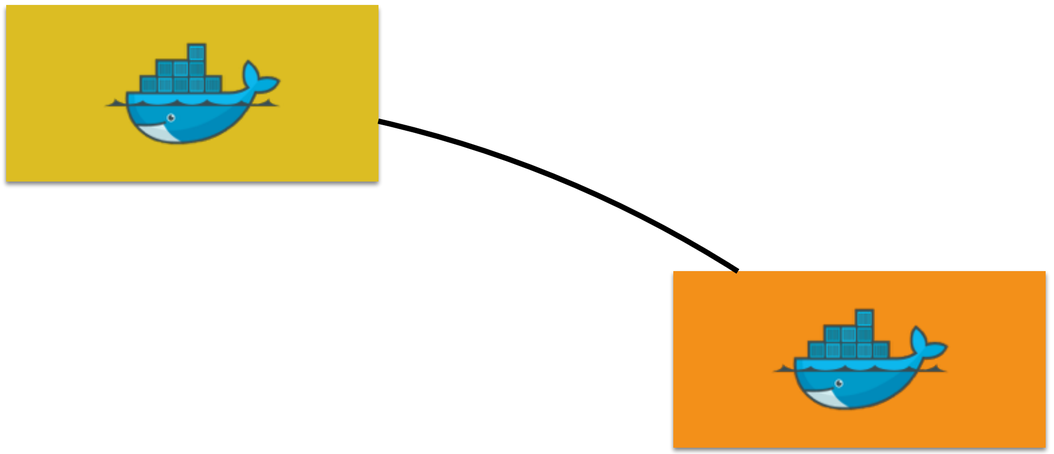
Orchestration Frameworks

- Developer
  - Core concepts
    - Cluster
    - Single container
    - Multi-container
  - Service discovery & LB
  - Persistent Volumes
  - Local development
- Ops
  - Multiple master
  - Scheduler
  - Rules and constraints
  - Monitoring
  - Rolling Update
  - Cloud/commercial support

Core Concepts: Docker Manager

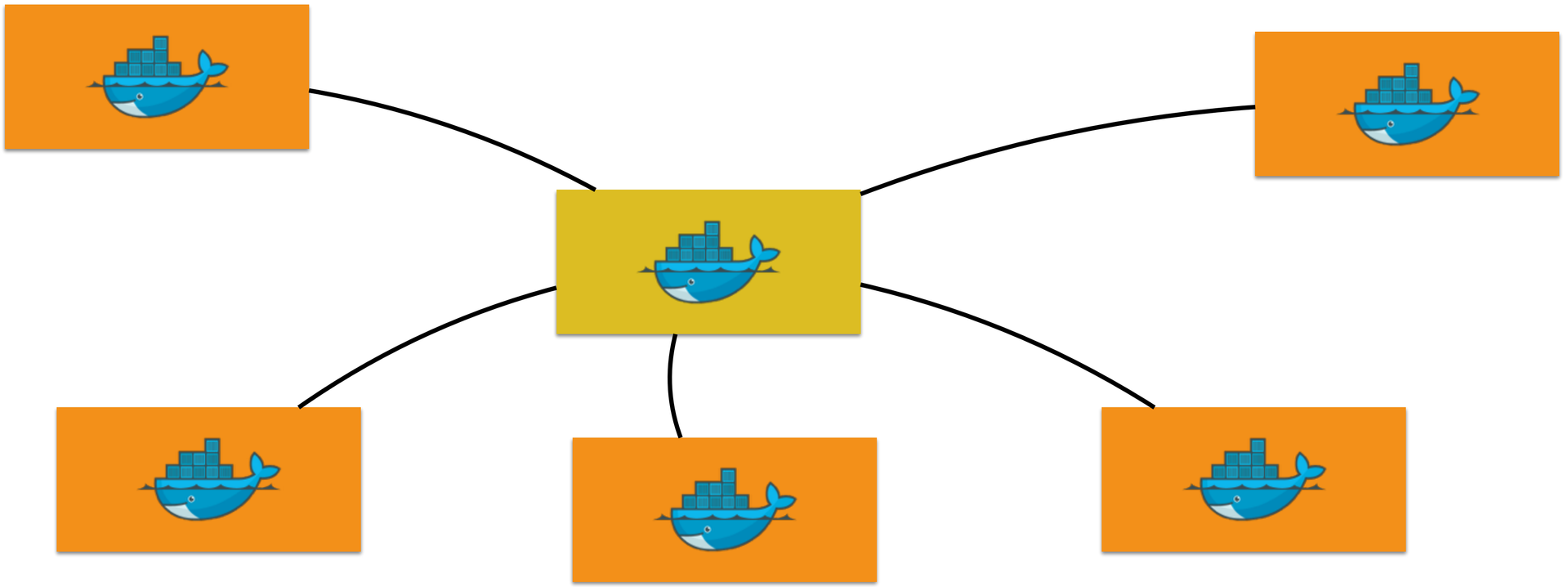
```
docker swarm init --listen-addr <ip>:2377
```

Core Concepts: Docker Worker



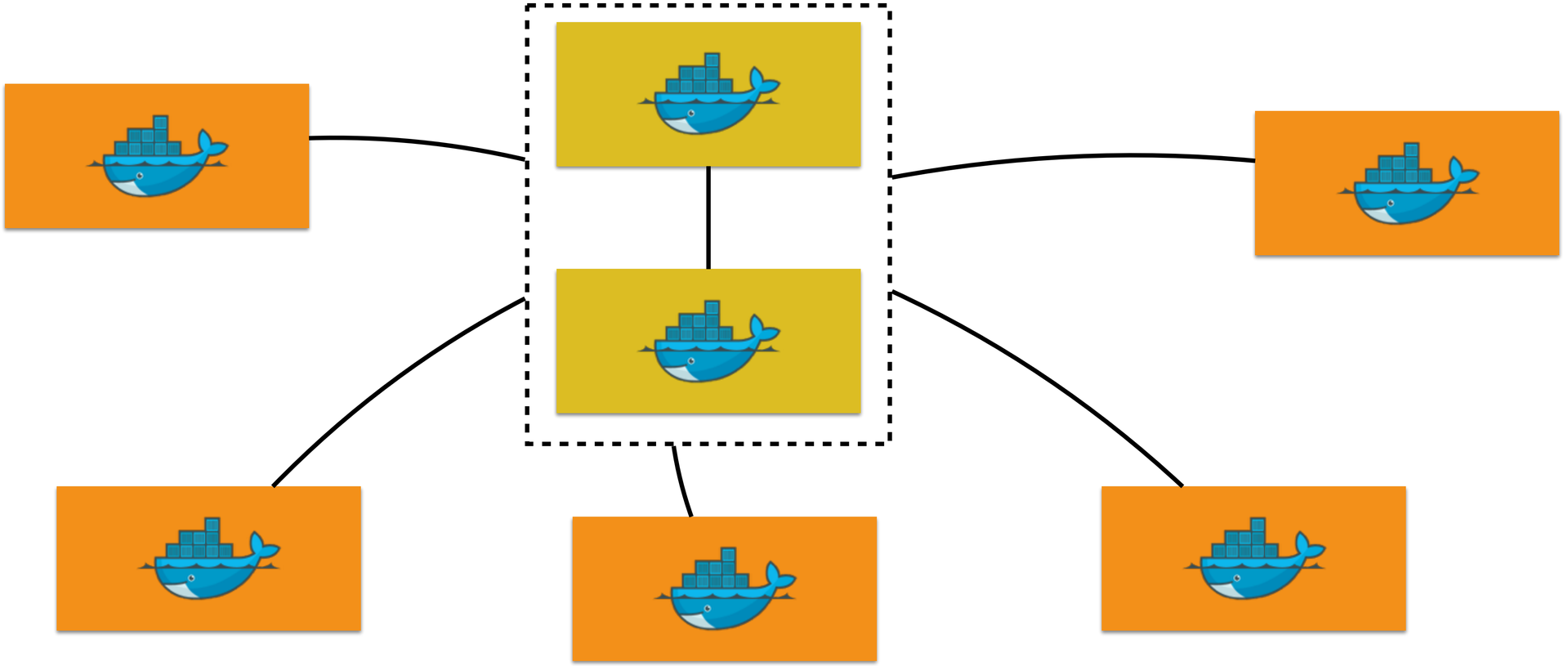
```
docker swarm join --token <worker_token> <manager>:2377
```

Core Concepts: Add More Workers



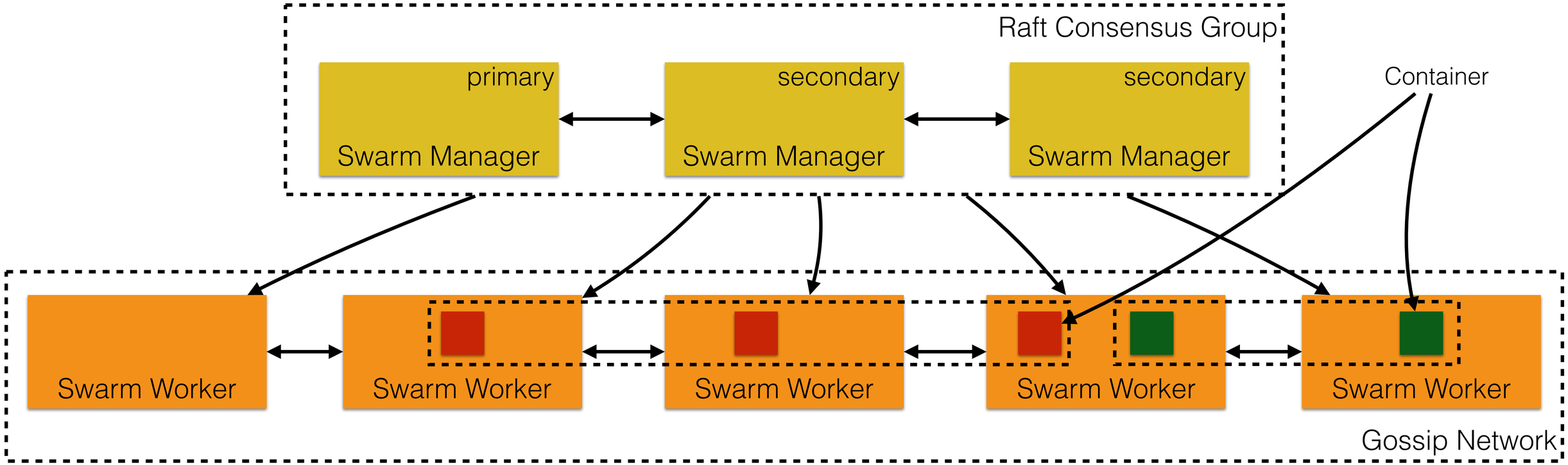
```
docker swarm join --token <worker_token> <manager>:2377
```

Core Concepts: Primary/Secondary Master



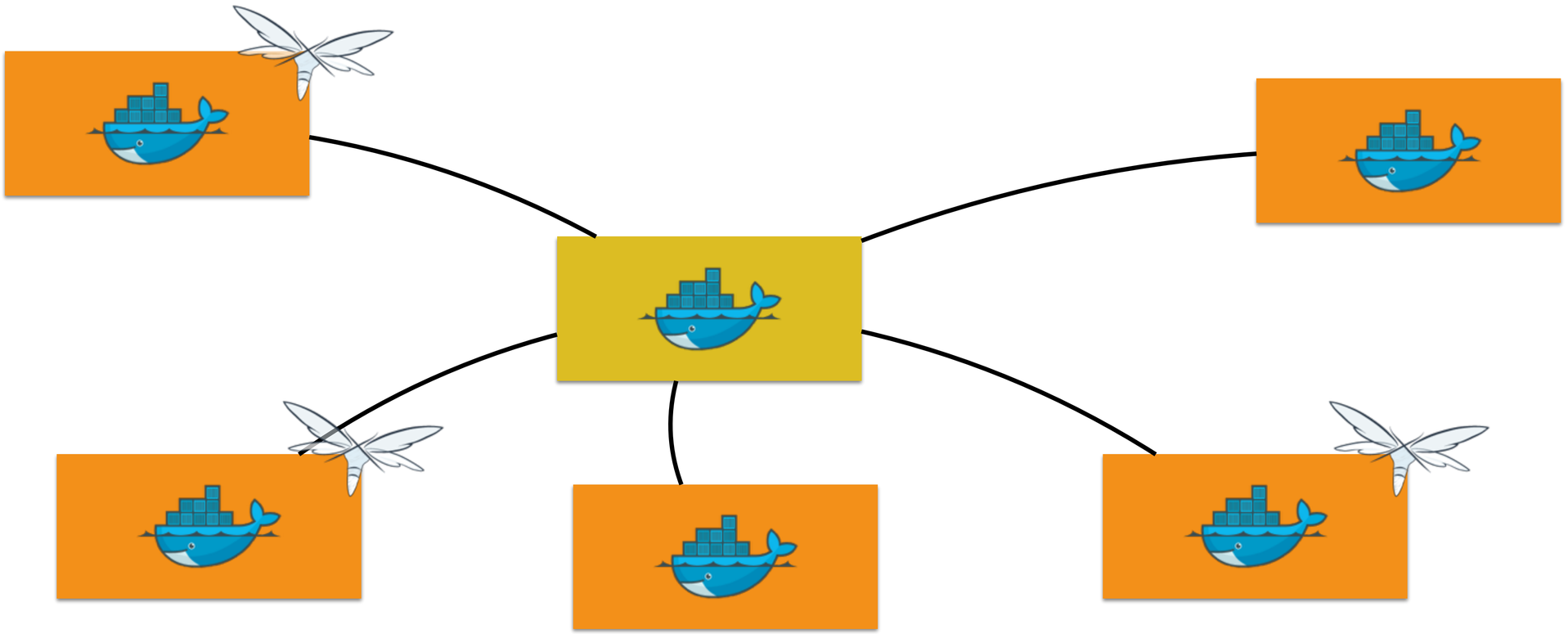
```
docker swarm join --manager --token <manager_token> --listen-
addr <master2>:2377 <master1>:2377
```

Core Concepts: Cluster



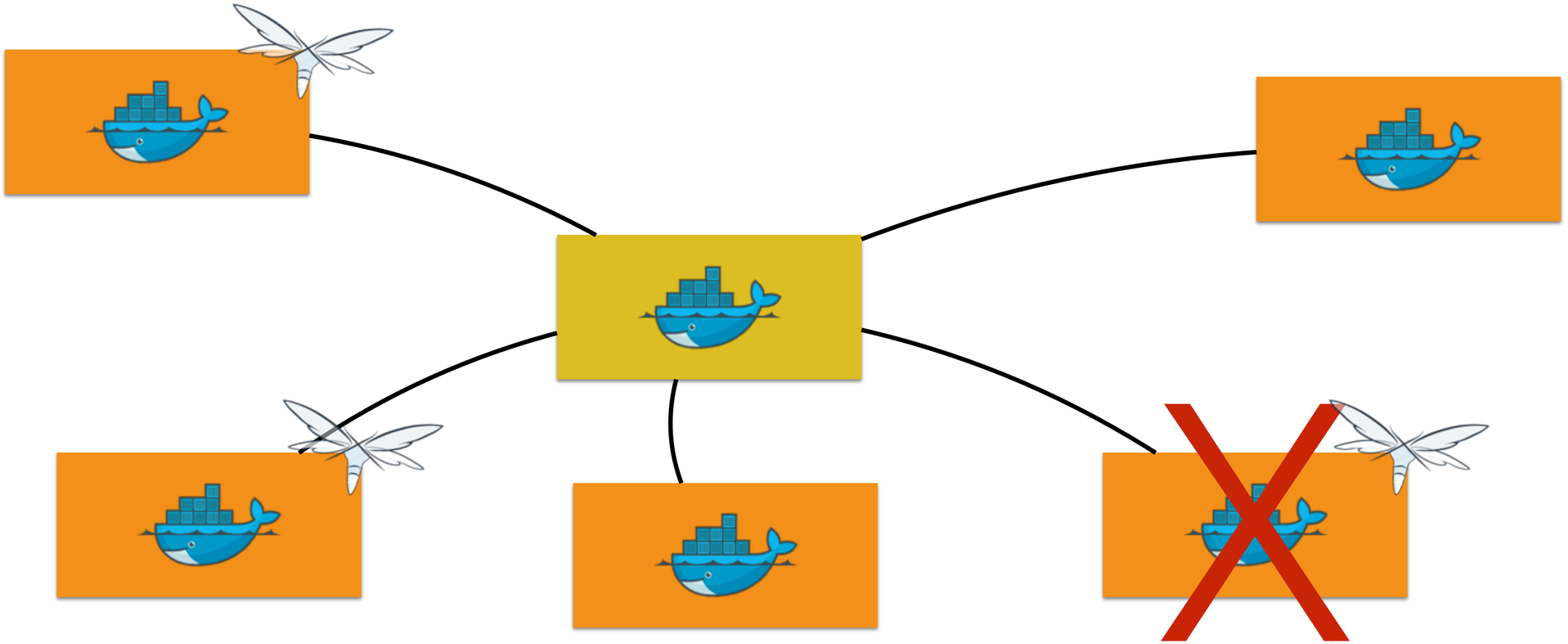
Strongly consistent
Replicated (Raft based)
Extremely fast (in-memory reads)

Core Concepts: Replicated Service

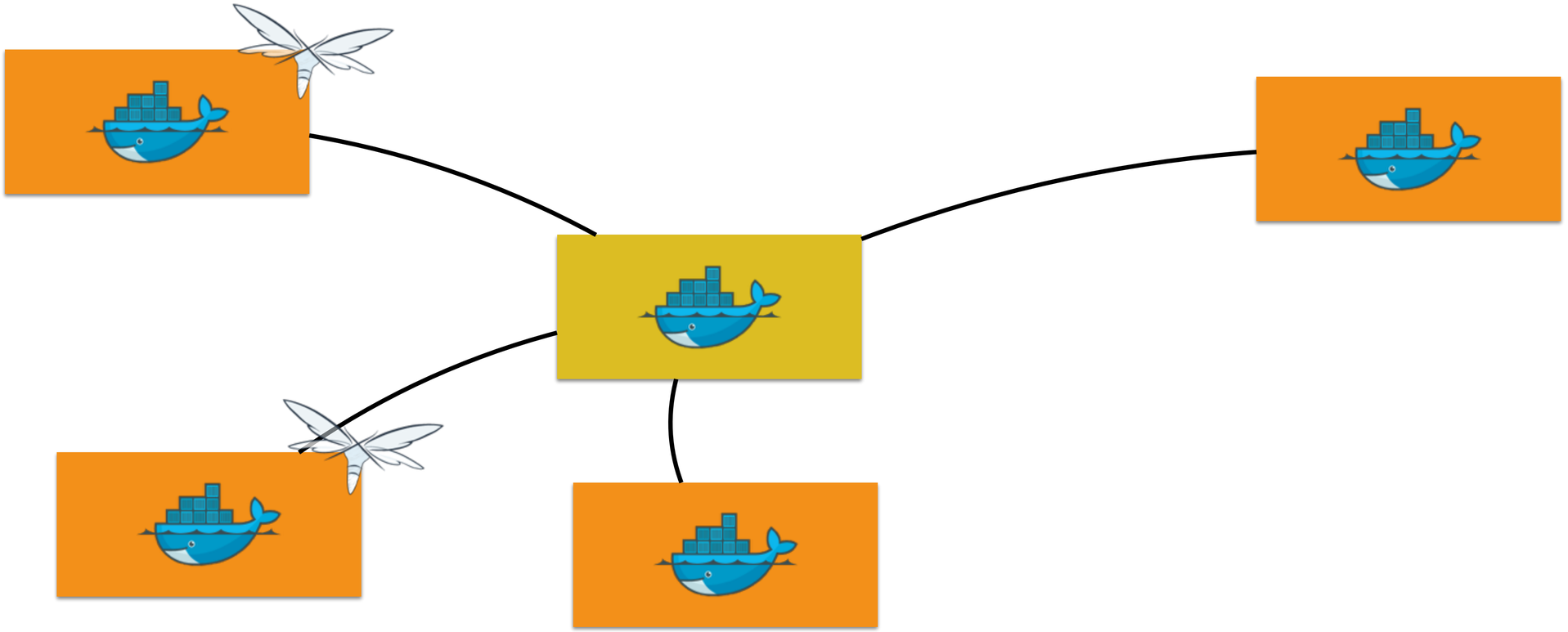


```
docker service create --replicas 3 --name web jboss/wildfly
```

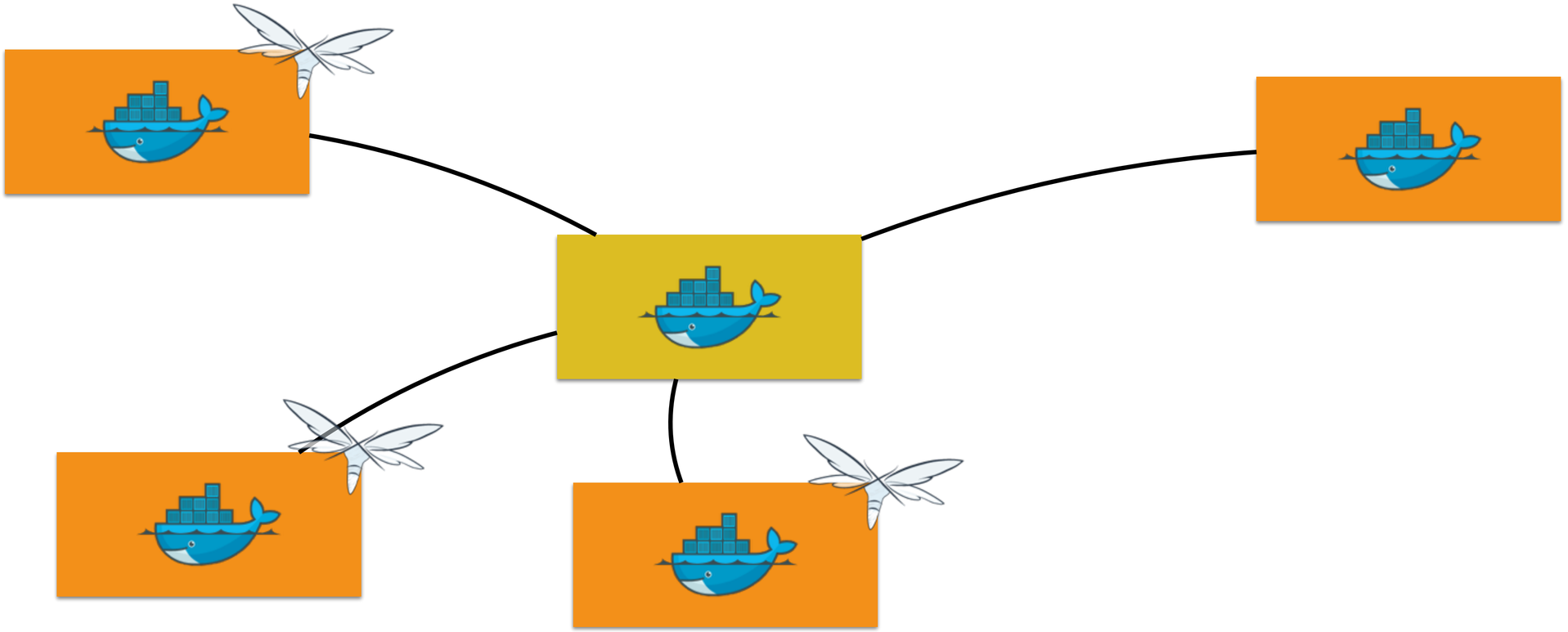
Core Concepts: Node Failure



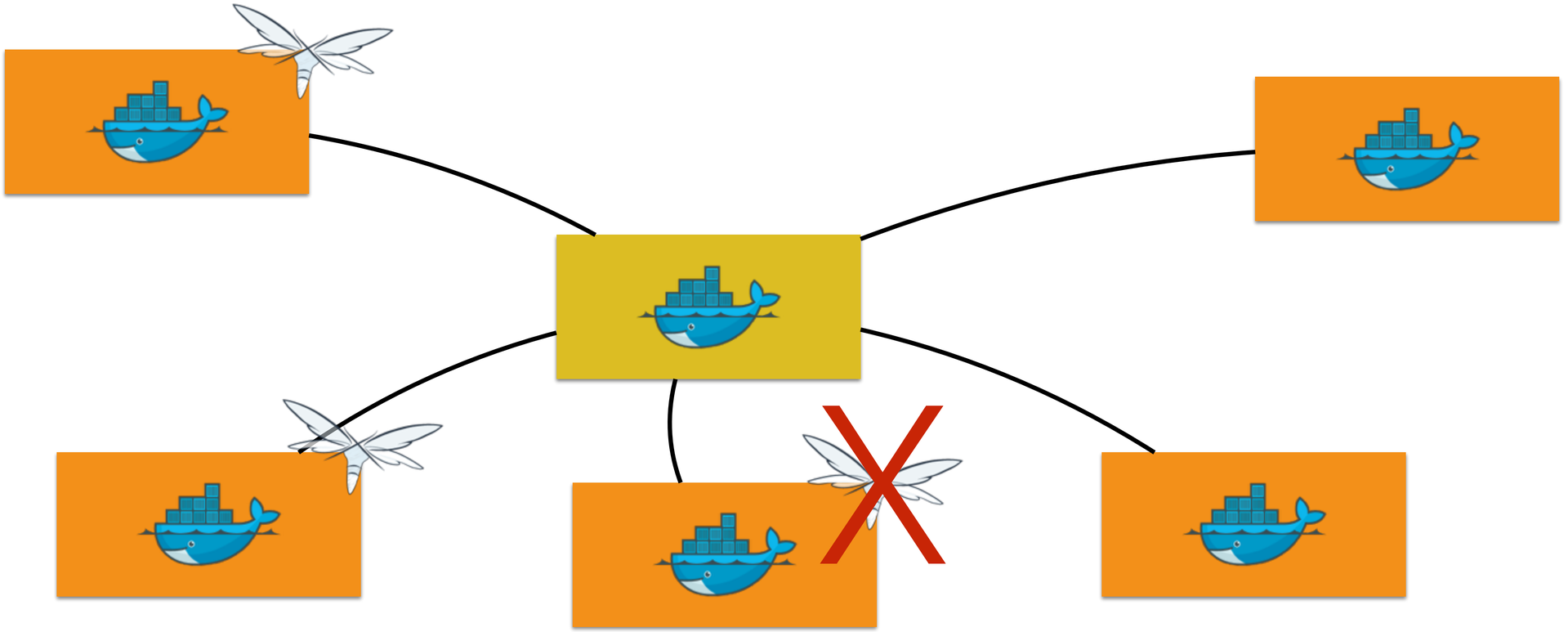
Core Concepts: Desired != Actual



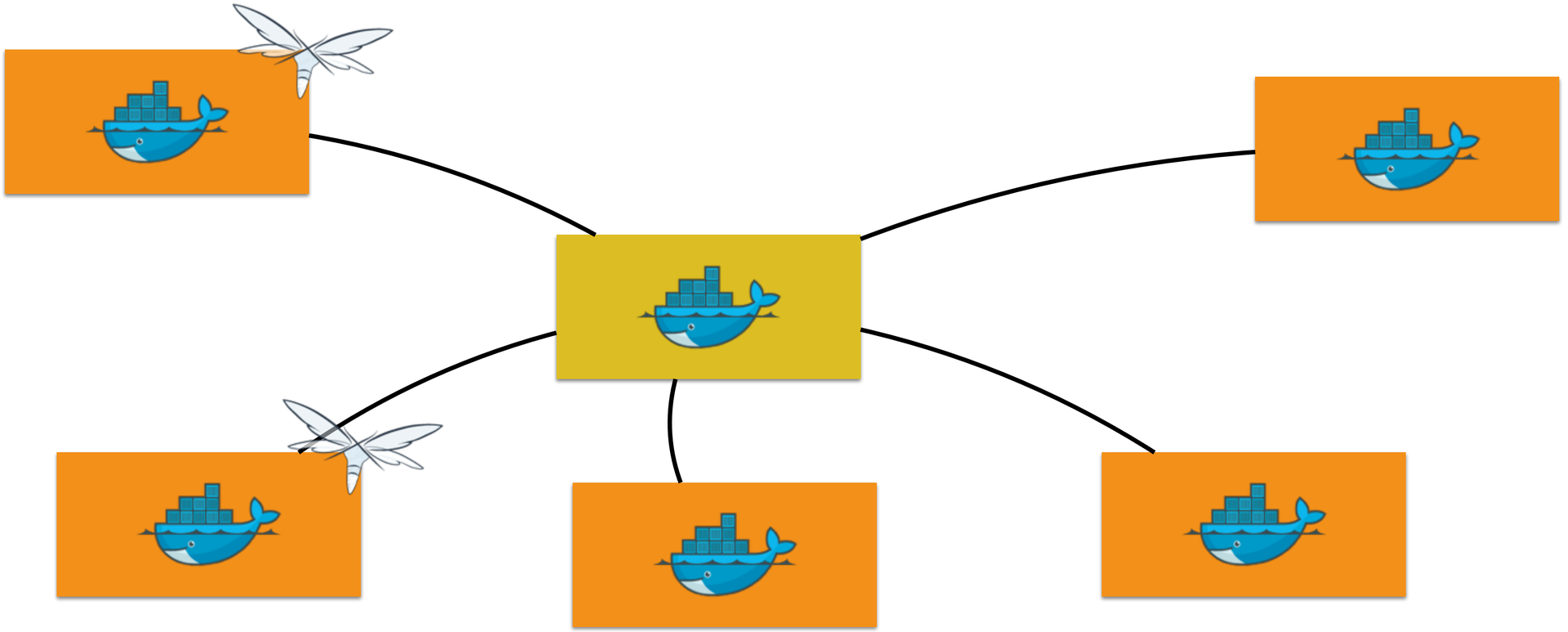
Core Concepts: Reconcile



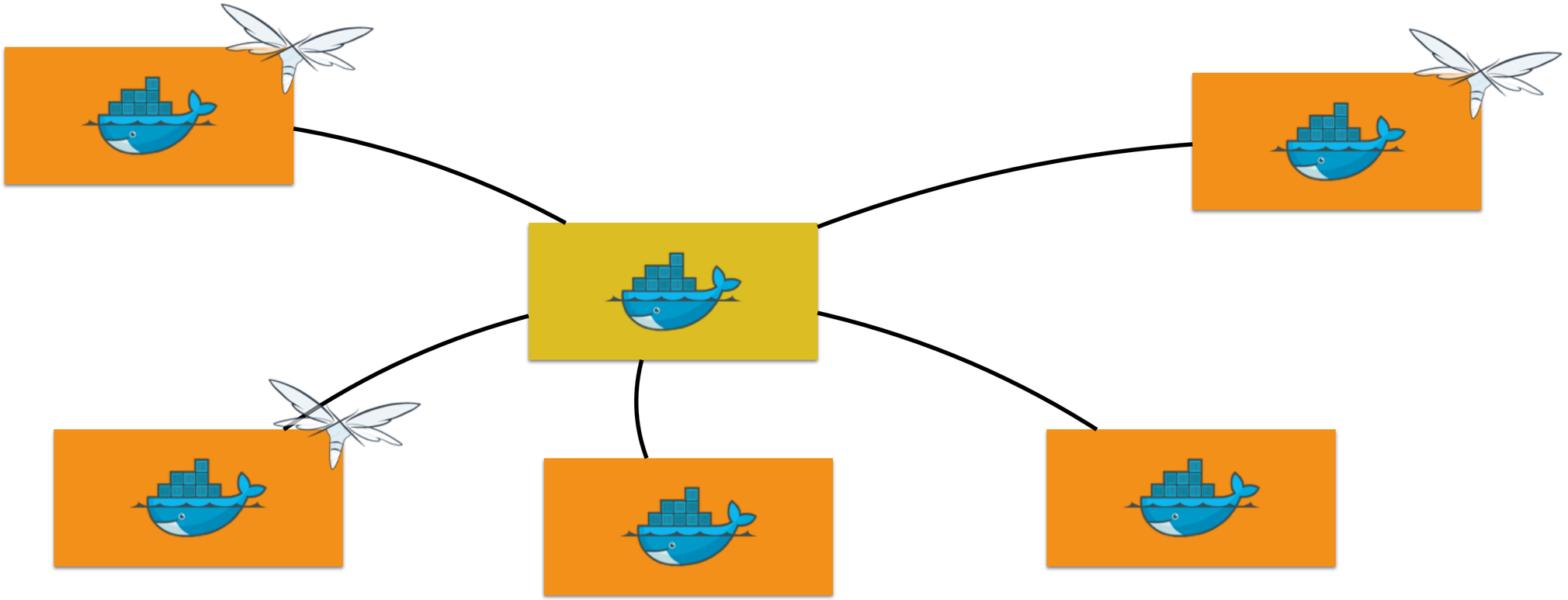
Core Concepts: Container Failure



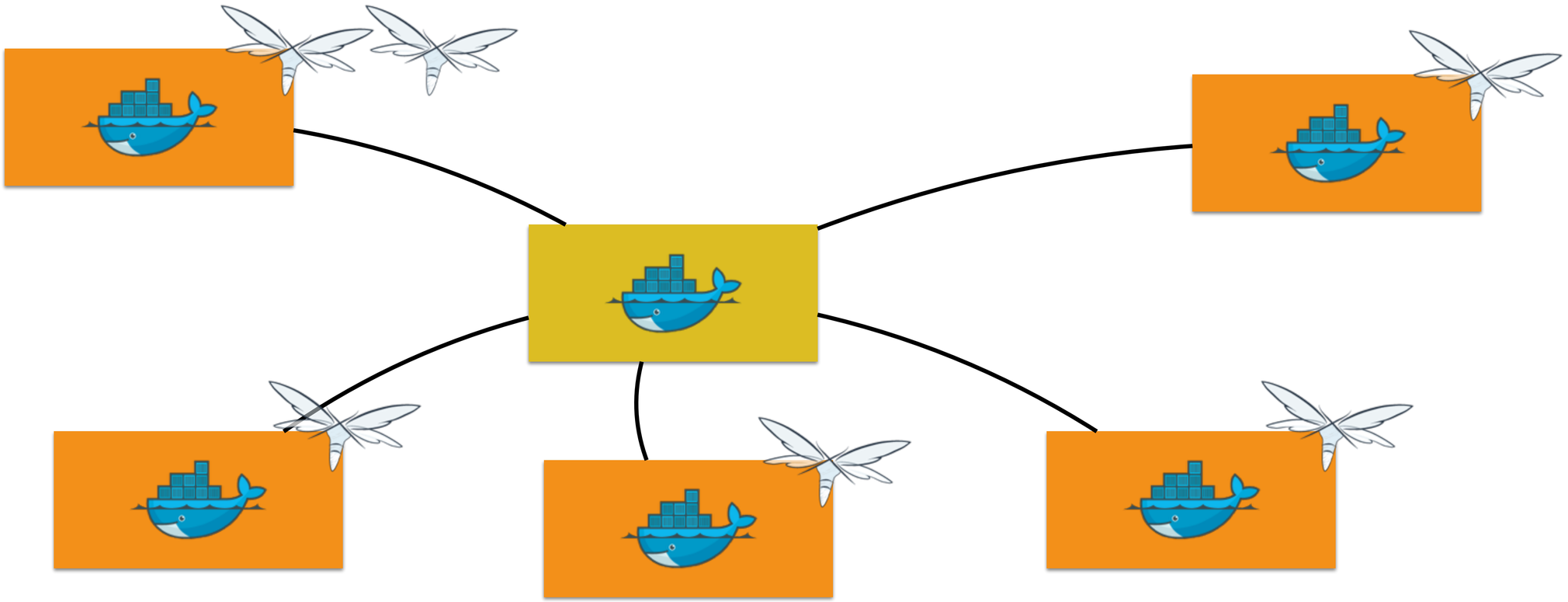
Core Concepts: Desired != Actual



Core Concepts: Reconcile

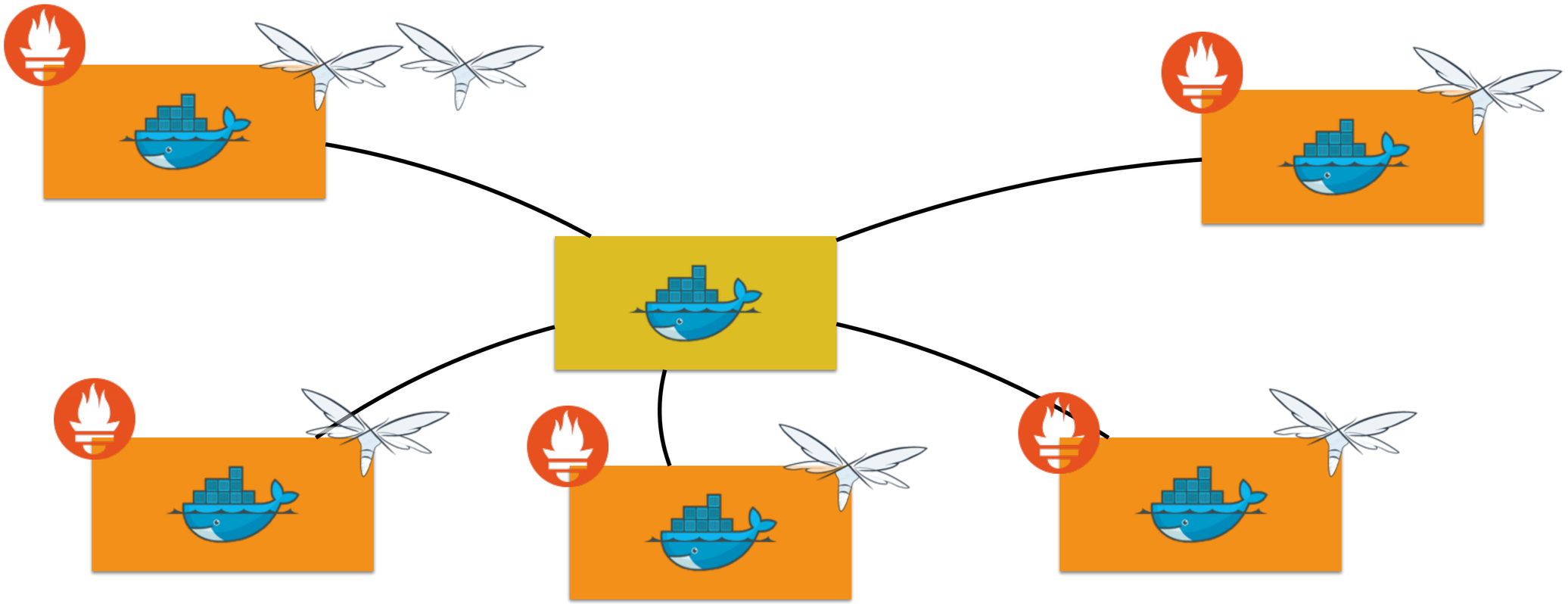


Core Concepts: Scale



```
docker service scale web=6
```

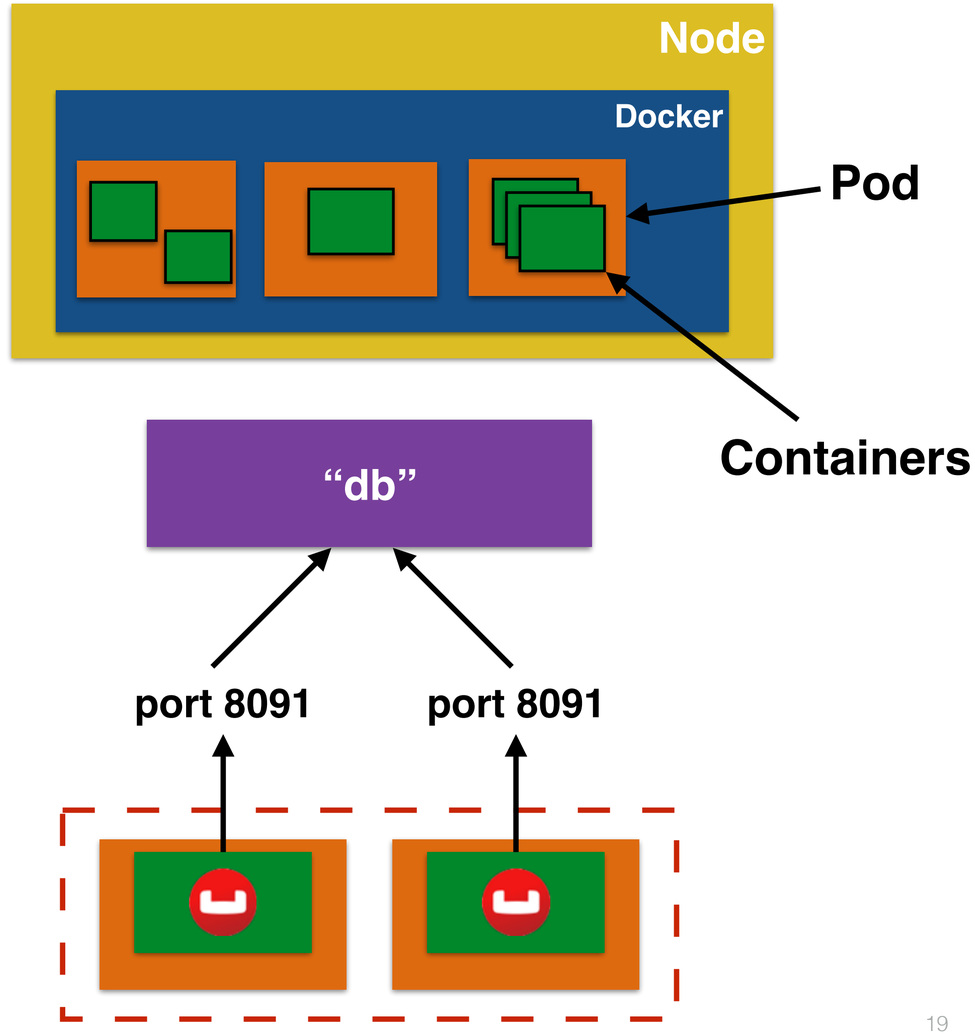
Core Concepts: Global Service



```
docker service create --mode=global --name=prom prom/prometheus
```

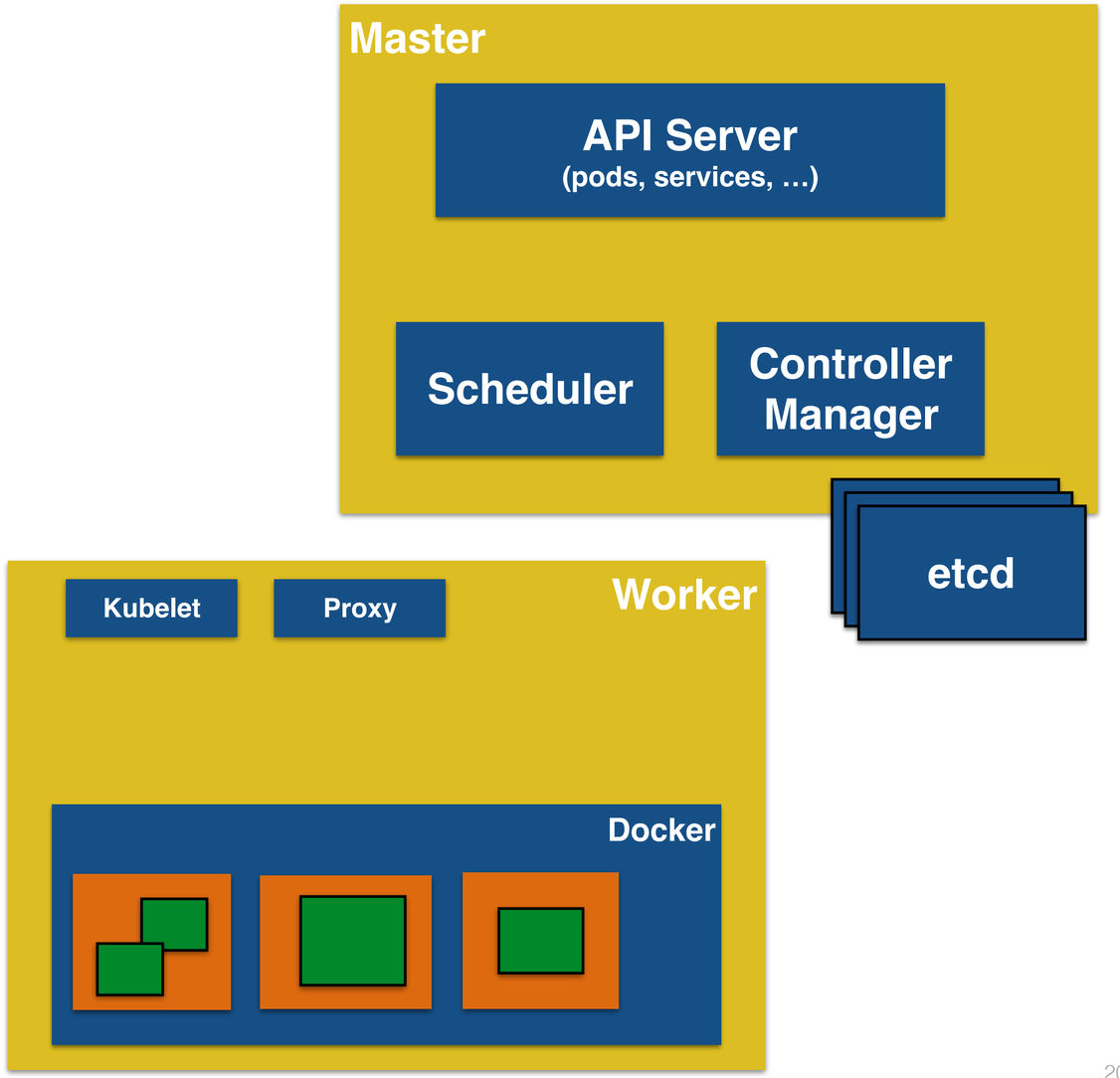
Core Concepts: Kubernetes

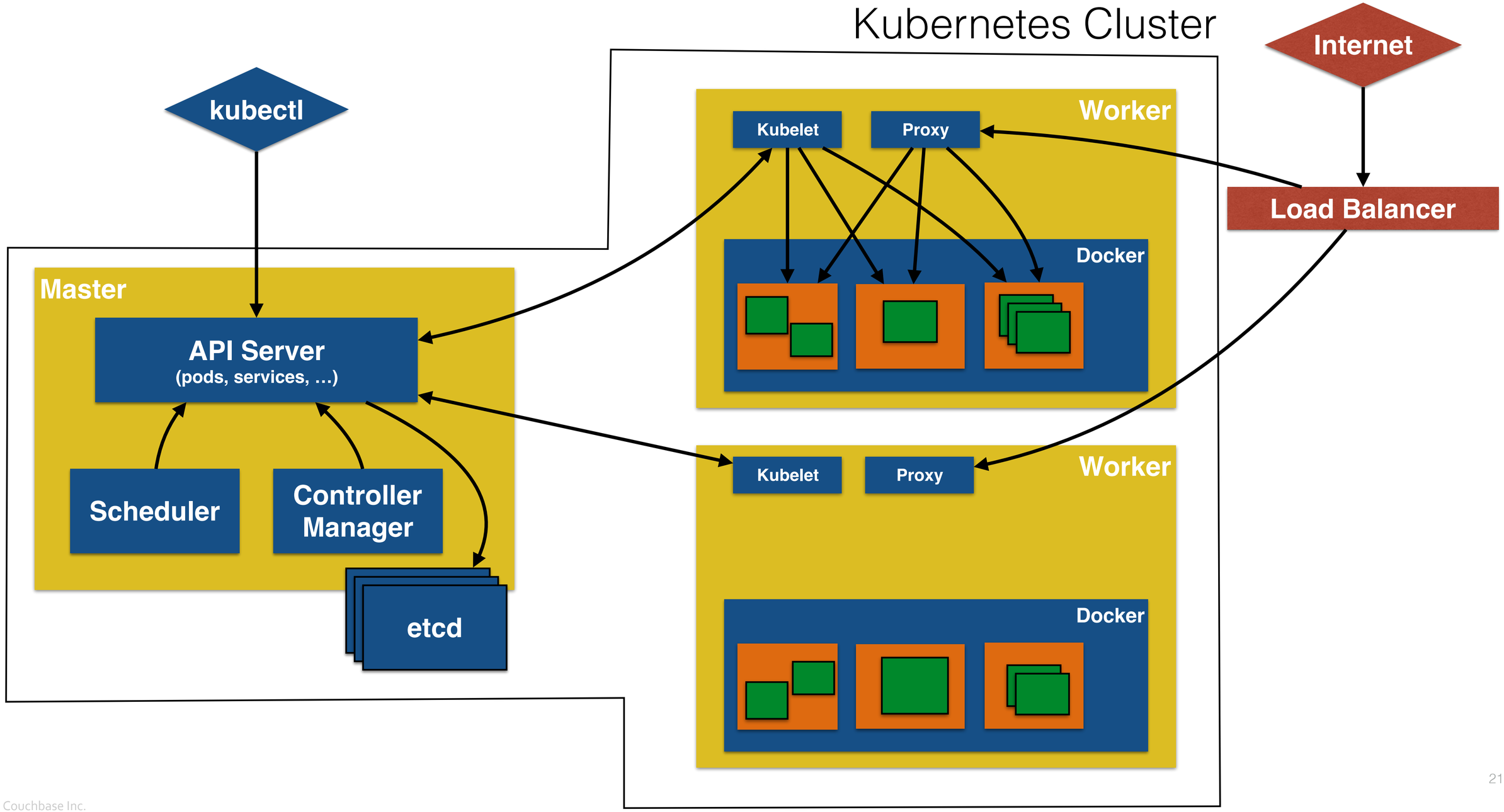
- **Pods**: colocated group of containers that share an IP, namespace, storage volume
- **Replica Set**: manages the lifecycle of pods and ensures specified number are running (next gen Replication Controller)
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to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



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)





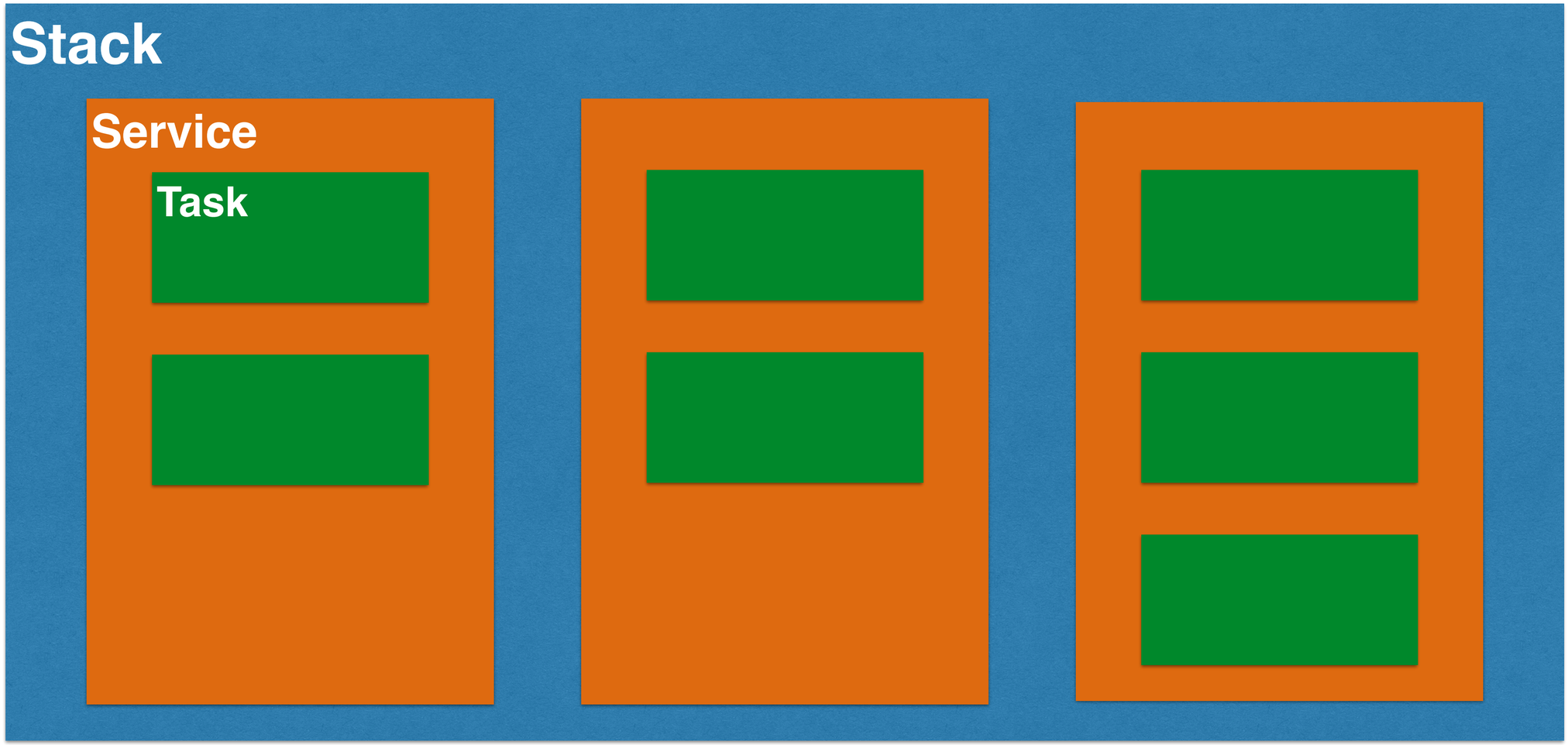
Service Discovery & LB: Docker

- Docker Compose
  - Define and run multi-container applications
  - Configuration defined in one or more files
    - `docker-compose.yml` (default)
    - `docker-compose.override.yml` (default)
    - Multiple files specified using `-f`
  - Deployed as Docker Stack
  - Great for dev, staging, and CI

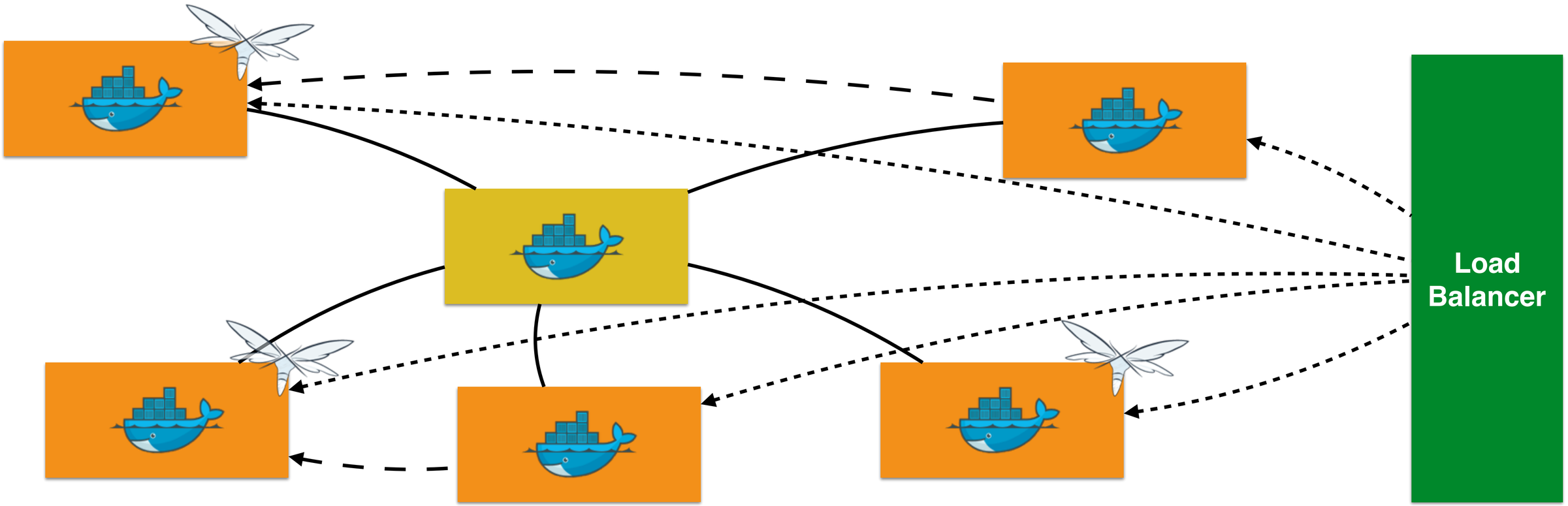
Service Discovery with Docker

```
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
```


Stack
Service
Task

Load Balancing: Docker




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

Service Discovery: Kubernetes

- Service: Abstract a set of pods as a single IP and port
  - Simple TCP/UDP load balancing
- Creates environment variables in other pods or DNS resolution
- Stable endpoint for pods to reference
  - Allows list of pods to change dynamically

Service Discovery with Kubernetes

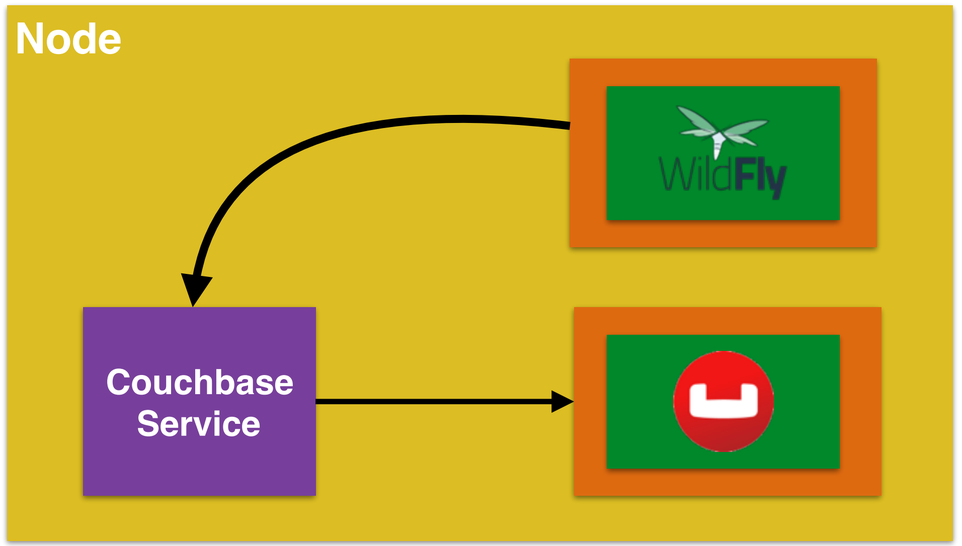
```yaml
apiVersion: v1
kind: Service
metadata:
  name: couchbase-service
spec:
  selector:
    app: couchbase-rs-pod
  ports:
    - name: admin
      port: 8091
    - name: views
      port: 8092
    - name: query
      port: 8093
    - name: memcached
      port: 11210
---
```

```yaml
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
---
```

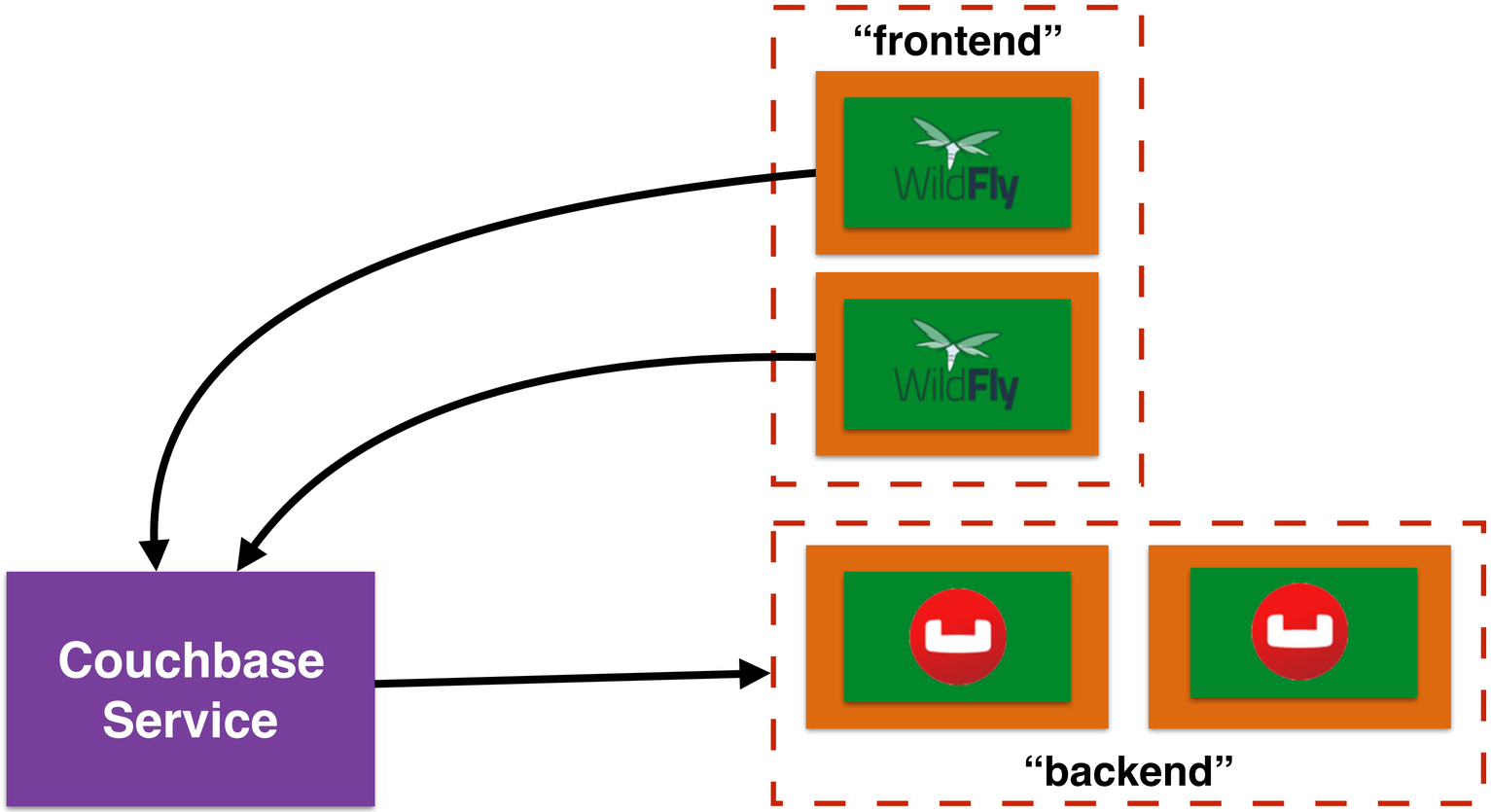
```yaml
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: wildfly-rs
  labels:
    name: wildfly
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rs-pod
        image: arungupta/wildfly-couchbase-javaee:travel
        env:
          name: COUCHBASE_URI
          value: couchbase-service
        ports:
        - containerPort: 8080

```

Couchbase Service: Kubernetes



Service and Replica Set: Kubernetes



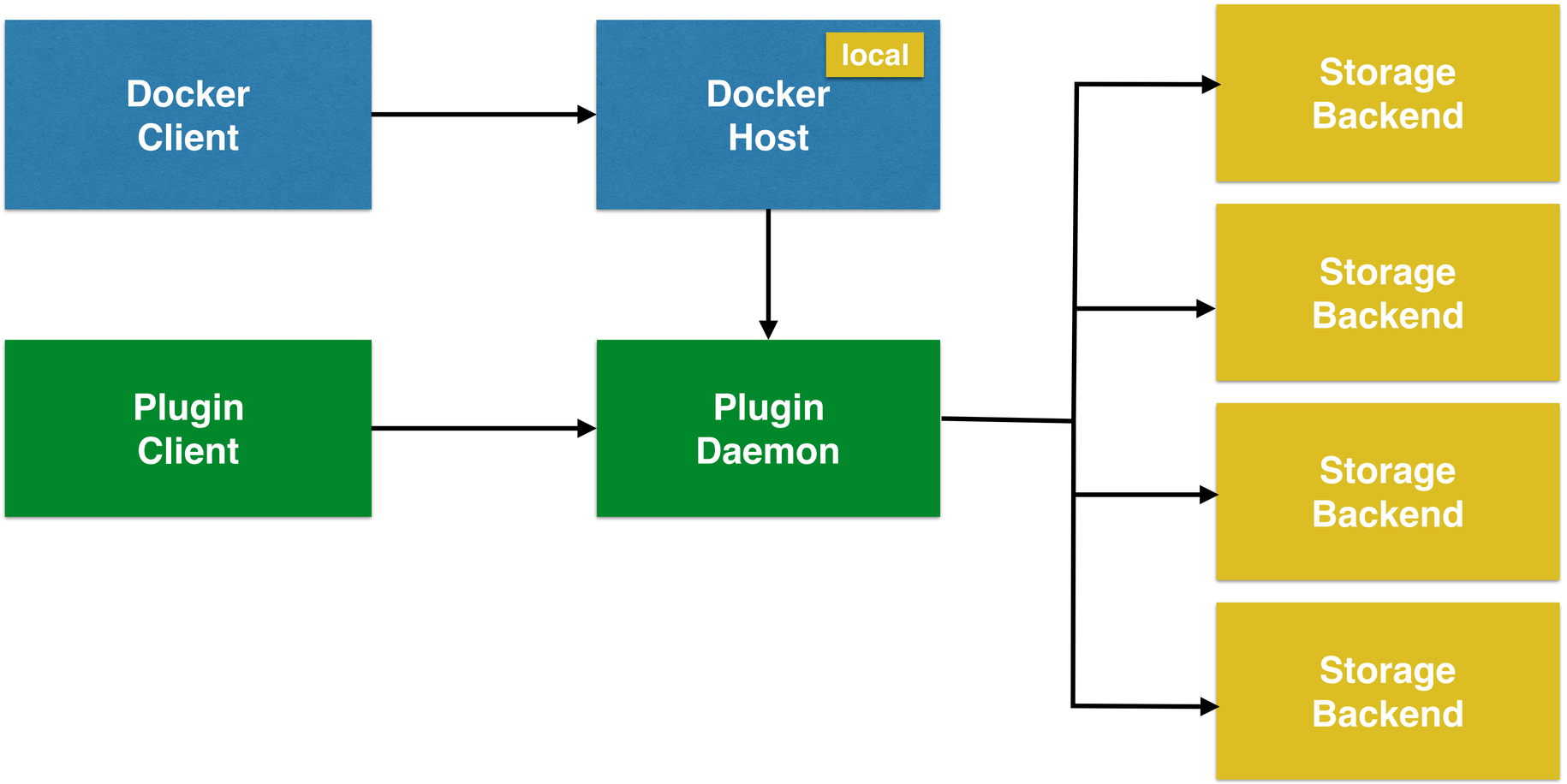
Docker Volumes: Persistent Containers

	Implicit Per-Container	Explicit Per-Container	Per-Host	Multi-Host
What?	Default sandbox	Explicit volume	Directory on host	Storage on distributed file systems
Location	`/var/lib/docker/volumes` on the host	`/var/lib/docker/volumes` on the host	Mounted within container	Ceph, GlusterFS, NFS, …
Container crash	Directory unavailable	Directory unavailable	Yes	Yes
Host crash	Directory unavailable	Directory unavailable	No	Yes
Shared	No	Yes	Yes (host only)	Yes (cluster wide)

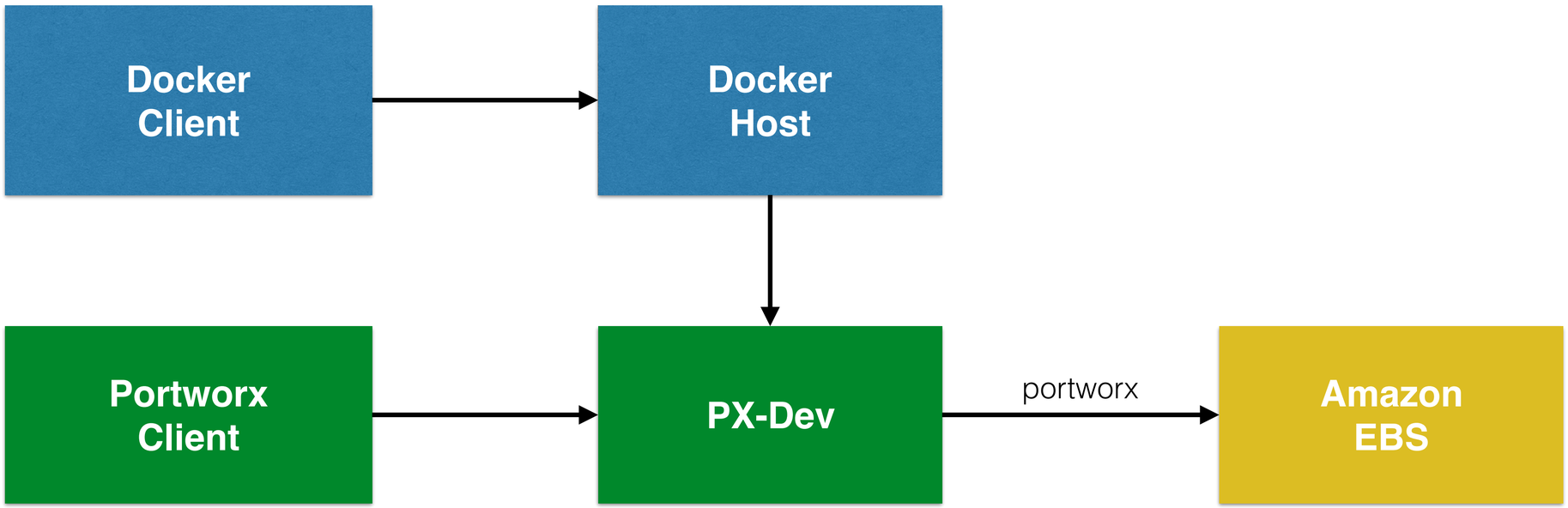
Docker Volume Plugin

- “Batteries included, but replaceable”
- Includes default driver for host-based volumes
- Plugins enables containers to be integrated with external storage systems
  - For example, Amazon EBS, Azure Storage and GCE Persistent Disks

Docker Volume Plugin Architecture



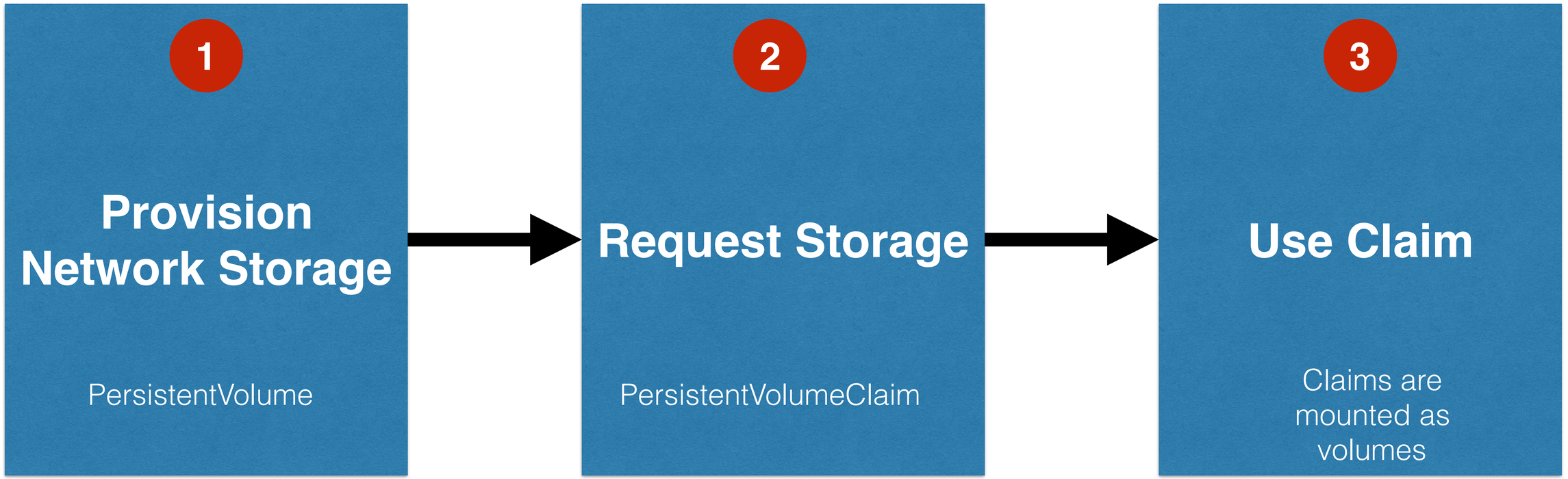
Docker Volume Plugin with Portworx



Kubernetes Volume

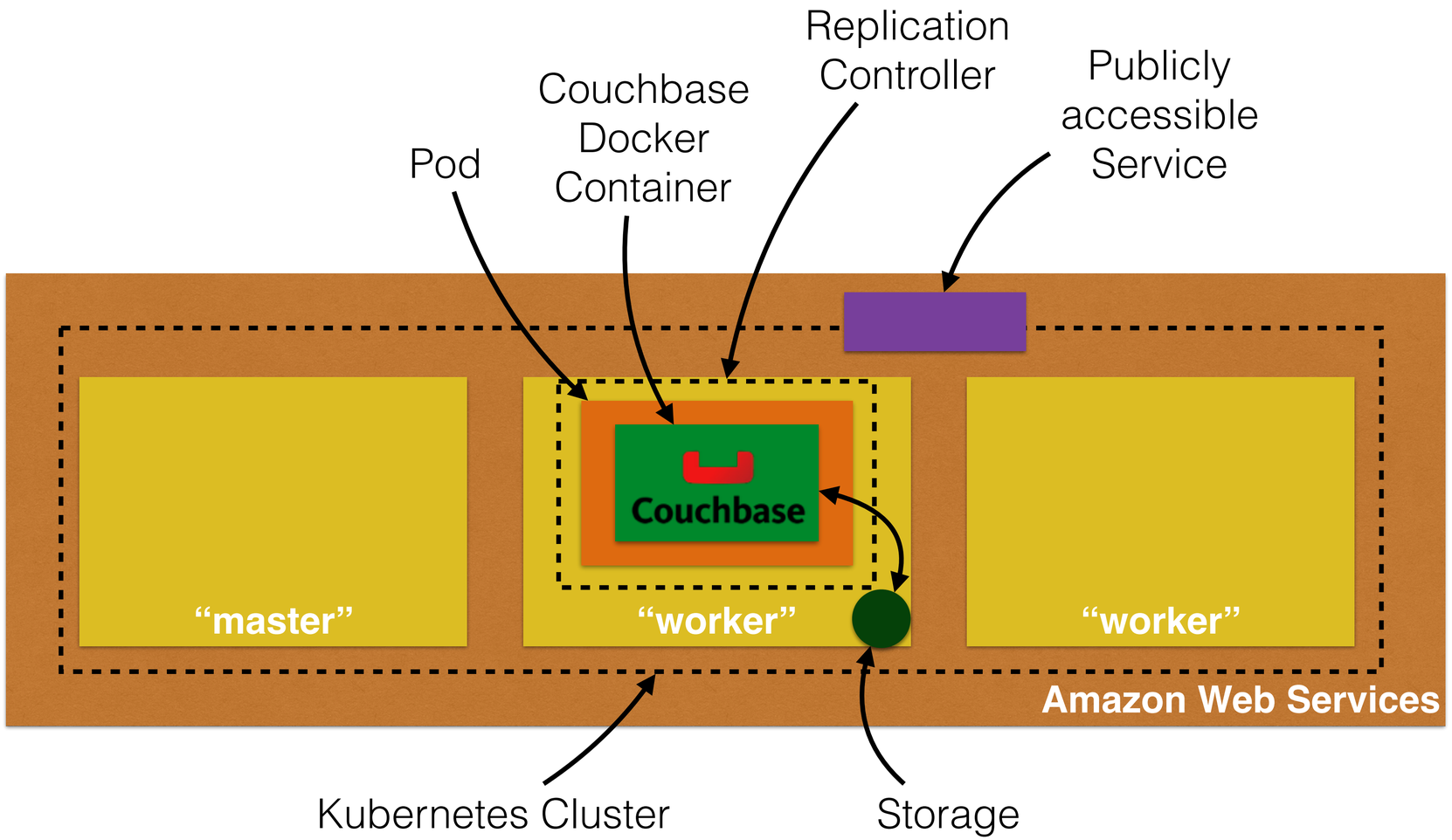
- Directory accessible to the containers in a pod
- Volume outlives any containers in a pod
- Common types
  - `hostPath`
  - `nfs`
  - `awsElasticBlockStore`
  - `gcePersistentDisk`

Kubernetes Persistent Volume

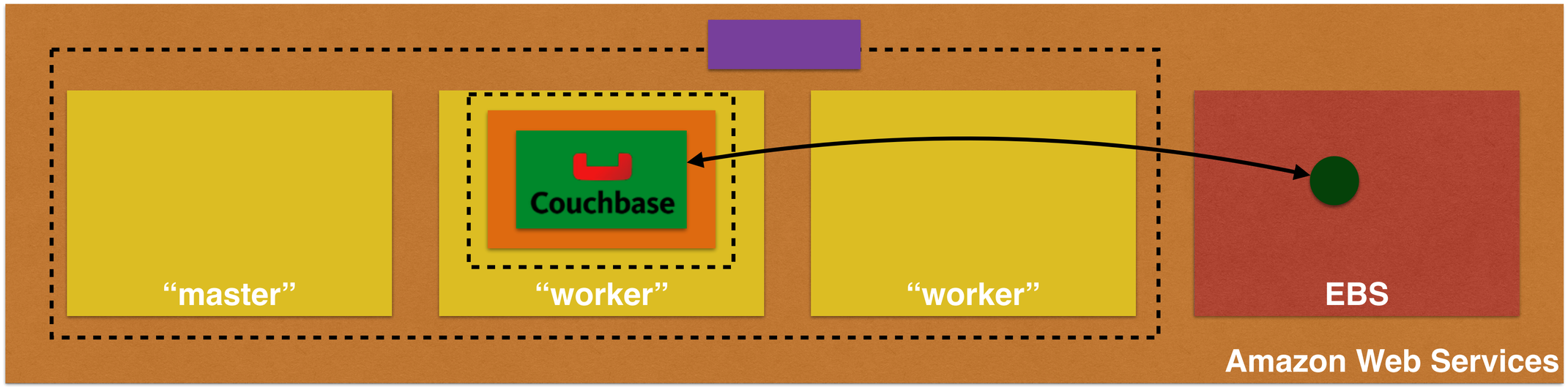


http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs

Kubernetes Volume: Persistent Container

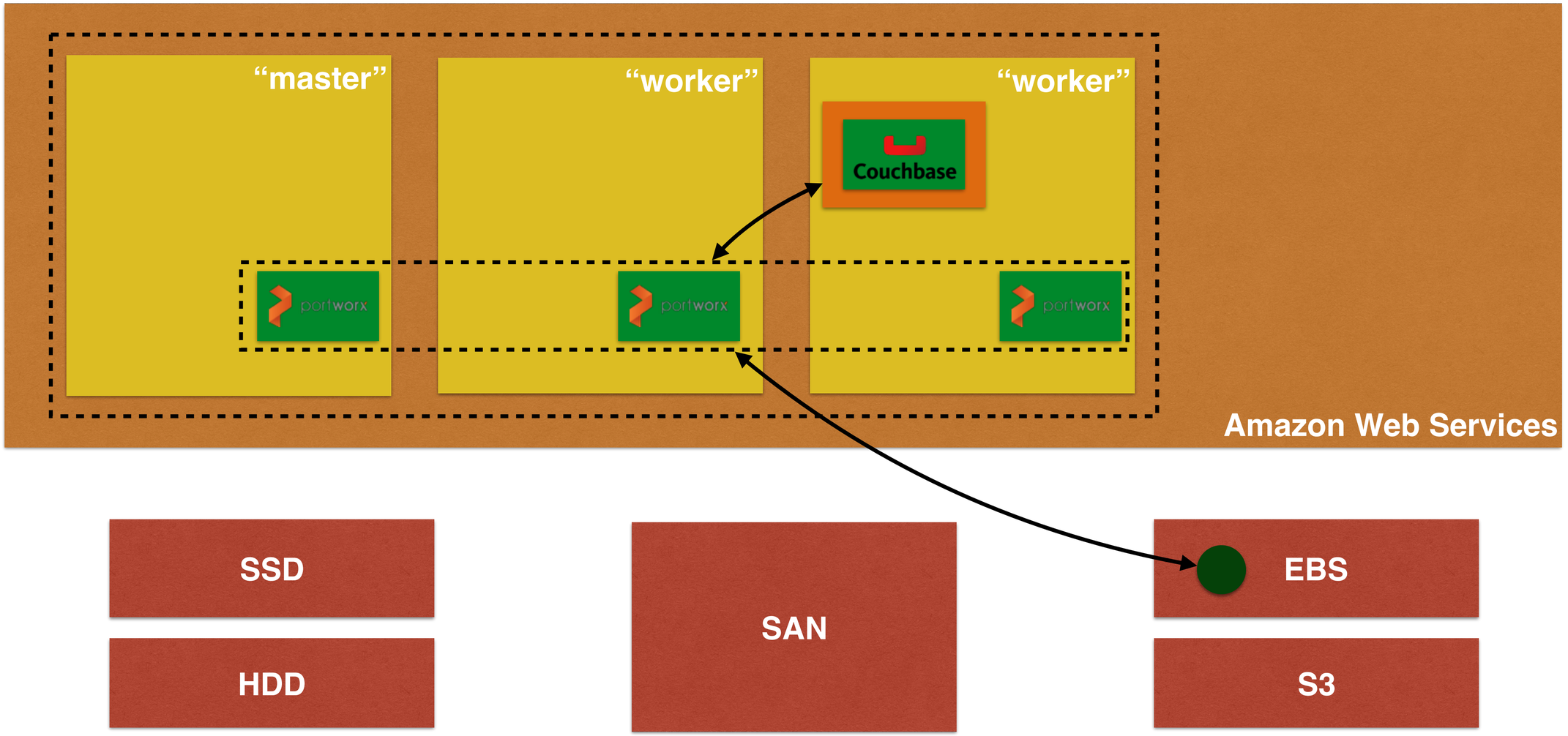


Kubernetes Volume: Persistent Container



http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs

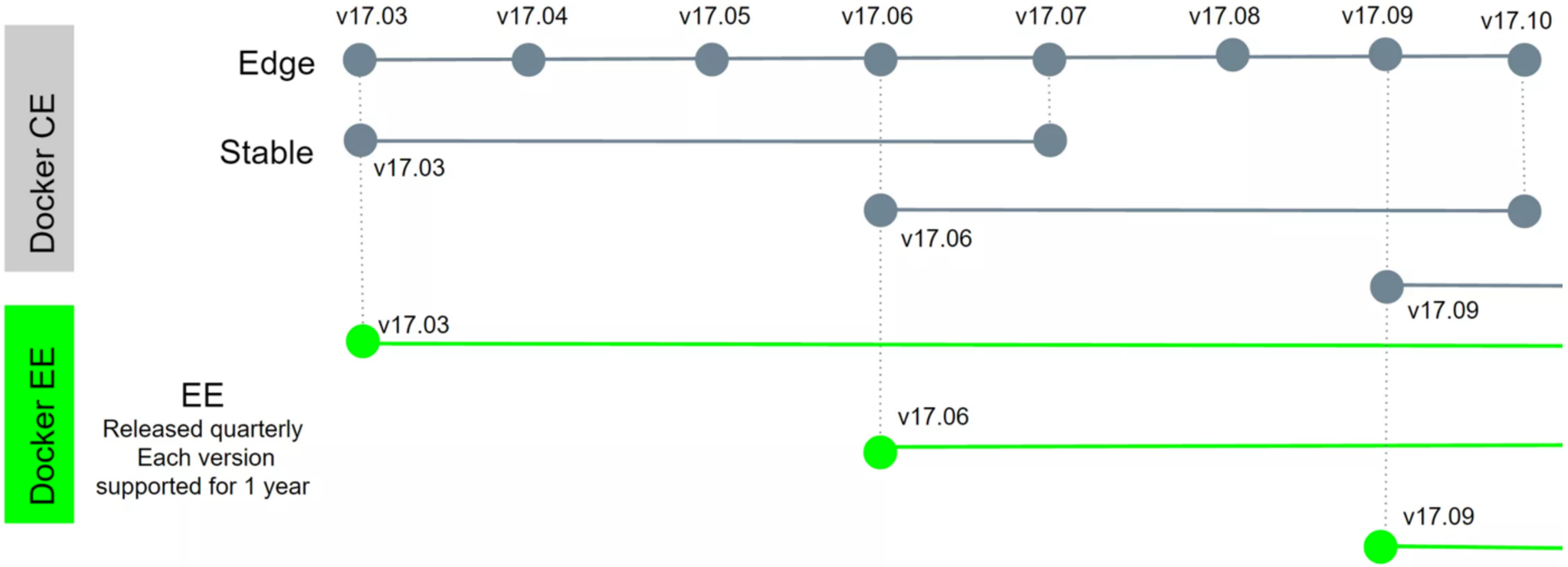
Kubernetes: Persistent Containers with Portworx



https://github.com/arun-gupta/couchbase-kubernetes/tree/master/cluster-petset-portworx

Development: Docker

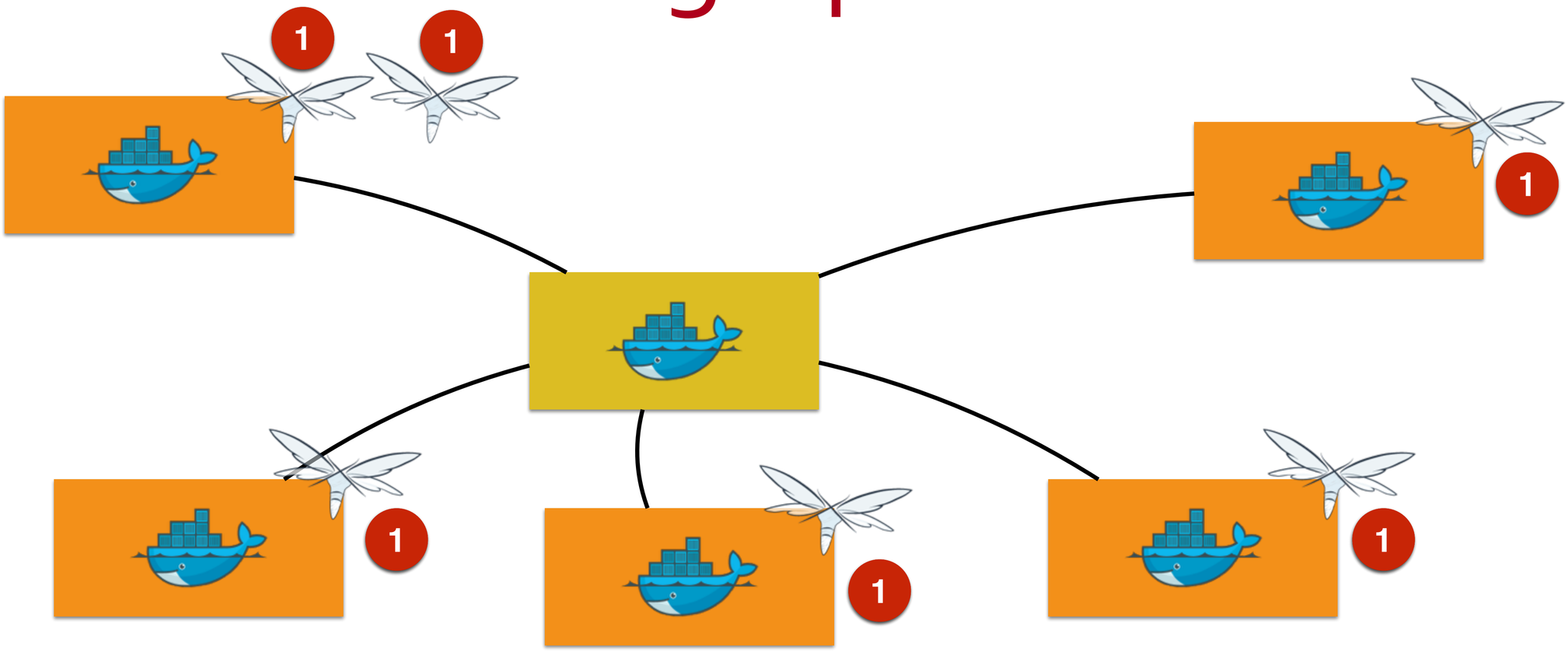
- Docker Community Edition
  - Docker for Mac/Windows/Linux
  - Monthly edge and quarterly stable releases
  - Native desktop or cloud provider experience



Development: Kubernetes

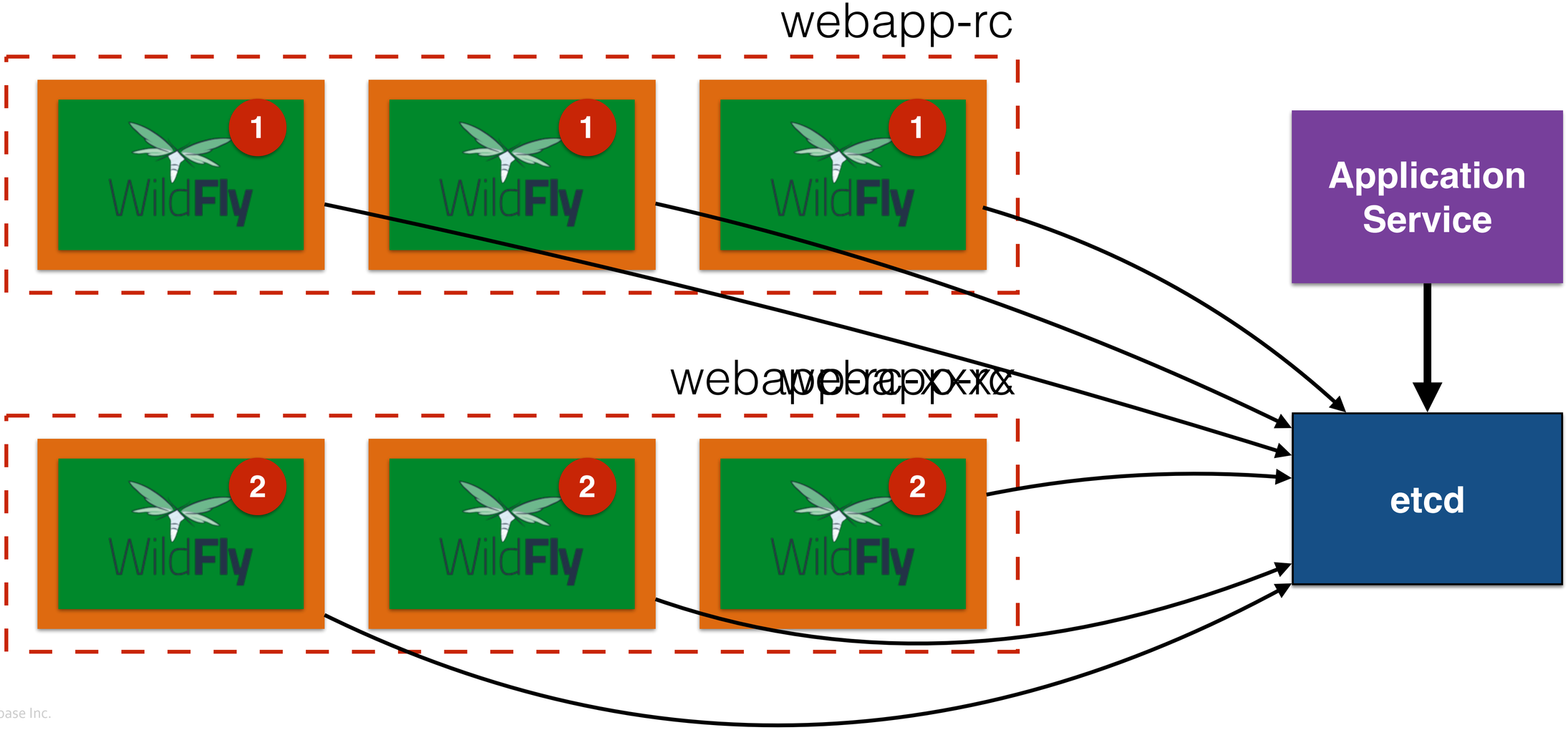
- Single node cluster
  - minikube
- Multi-node cluster
  - kops
  - kube-aws (CoreOS + AWS)
  - kube-up (deprecated)
  - Google Cloud, Azure, Tectonic, …

Rolling Update: Docker




```
docker service update web --image wildfly:2 --update-parallelism
2 --update-delay 10s
```

Rolling Update: Kubernetes

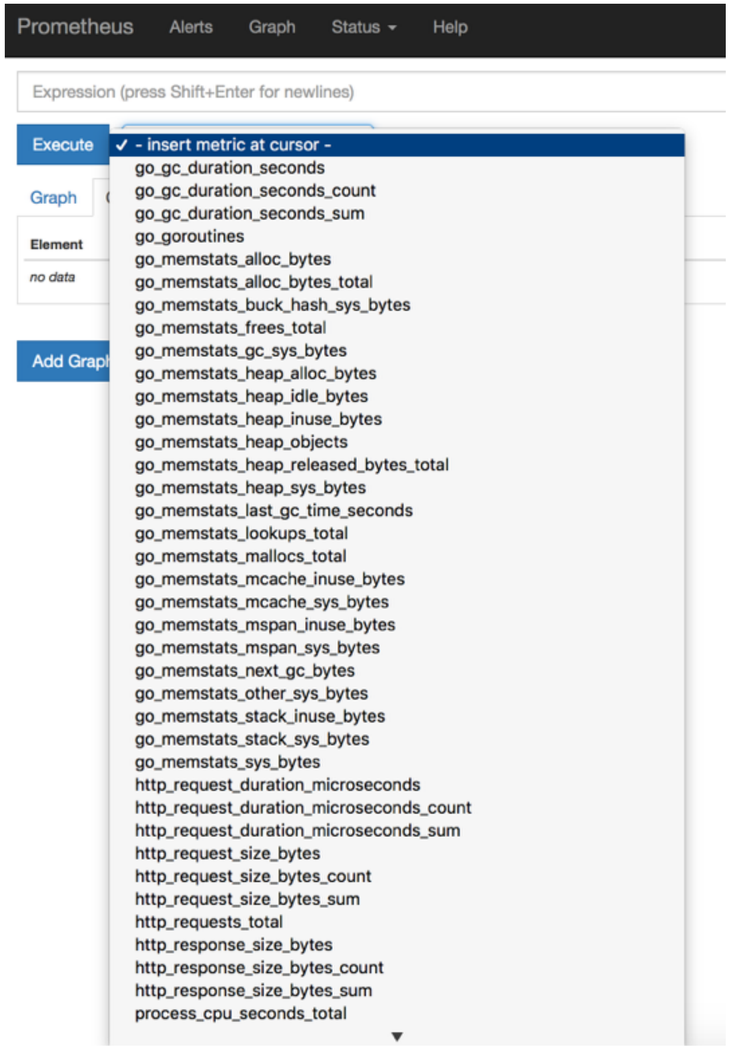


https://github.com/arun-gupta/kubernetes-java-sample/tree/master/rolling-update

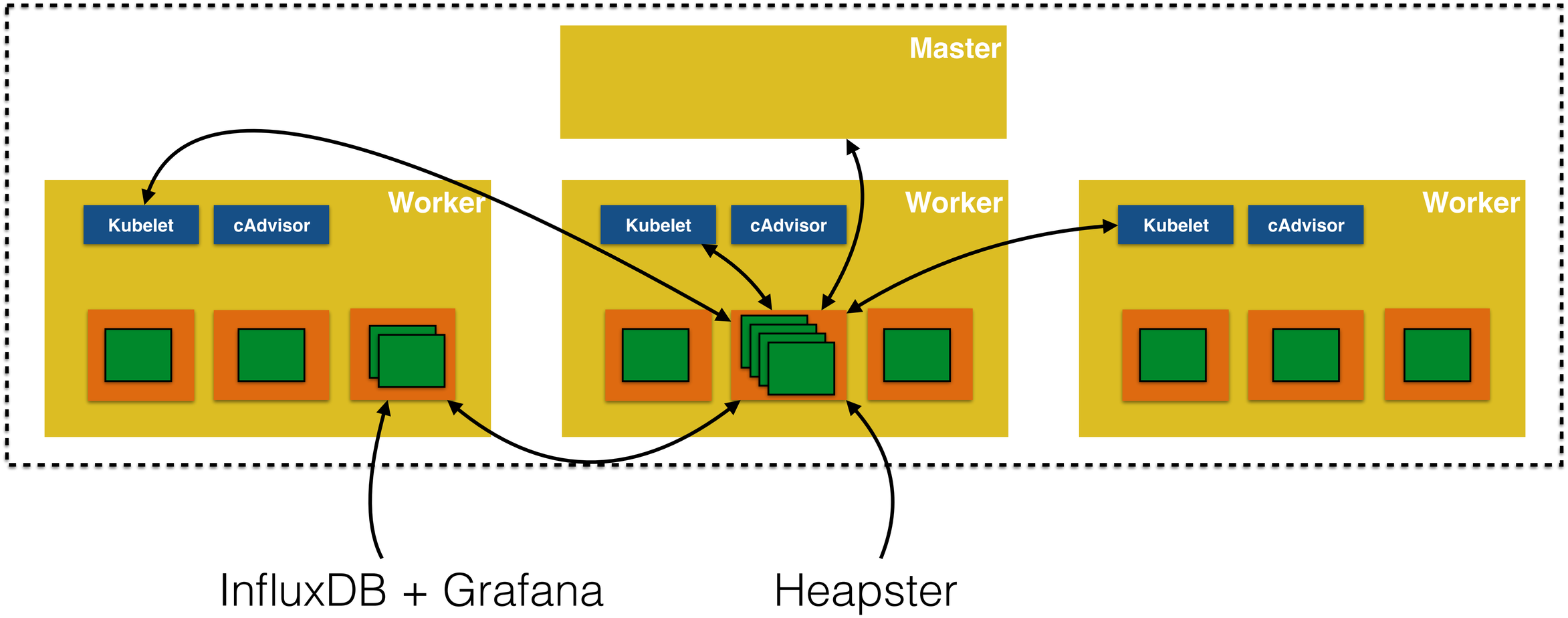
Monitoring: Docker

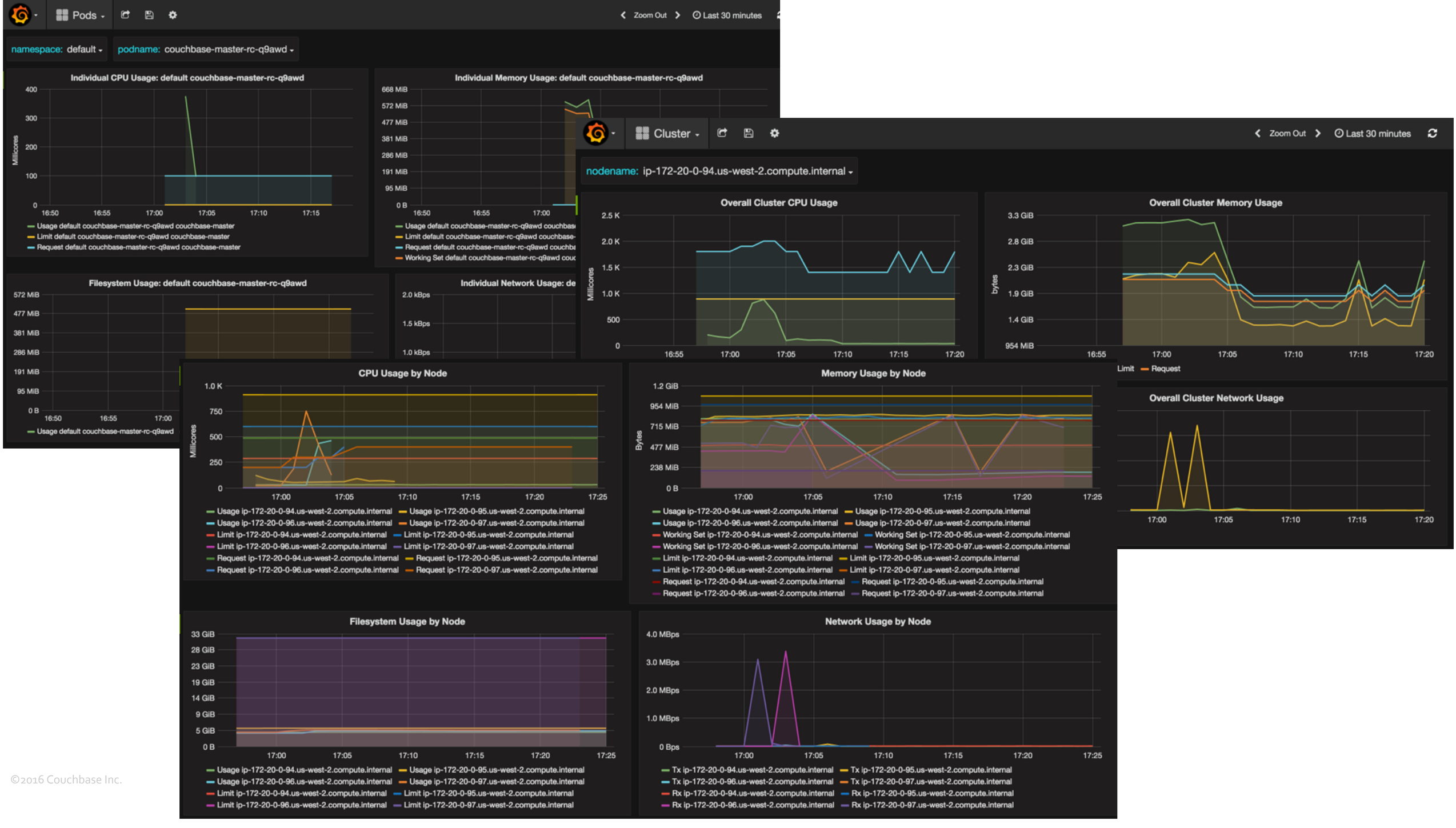
- `docker container stats` CLI
- Docker Remote API
- `docker system events` CLI
- In-built Prometheus endpoint
- cAdvisor

Monitoring: Docker



Monitoring: Kubernetes





Monitoring: Docker and Kubernetes

References

- Docker: docker.io
- Kubernetes: kubernetes.io
- Slides: https://github.com/javaee-samples/docker-java