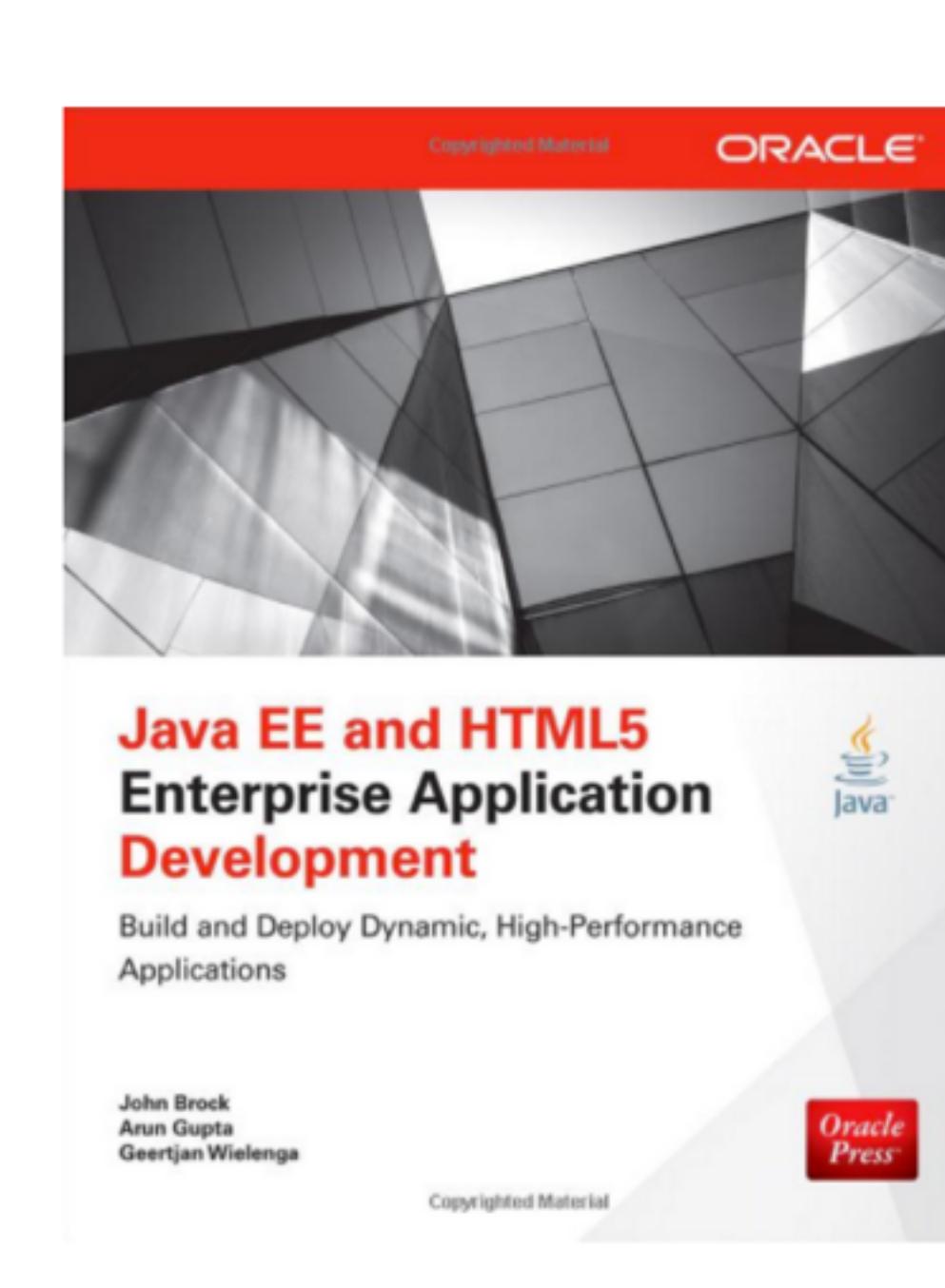
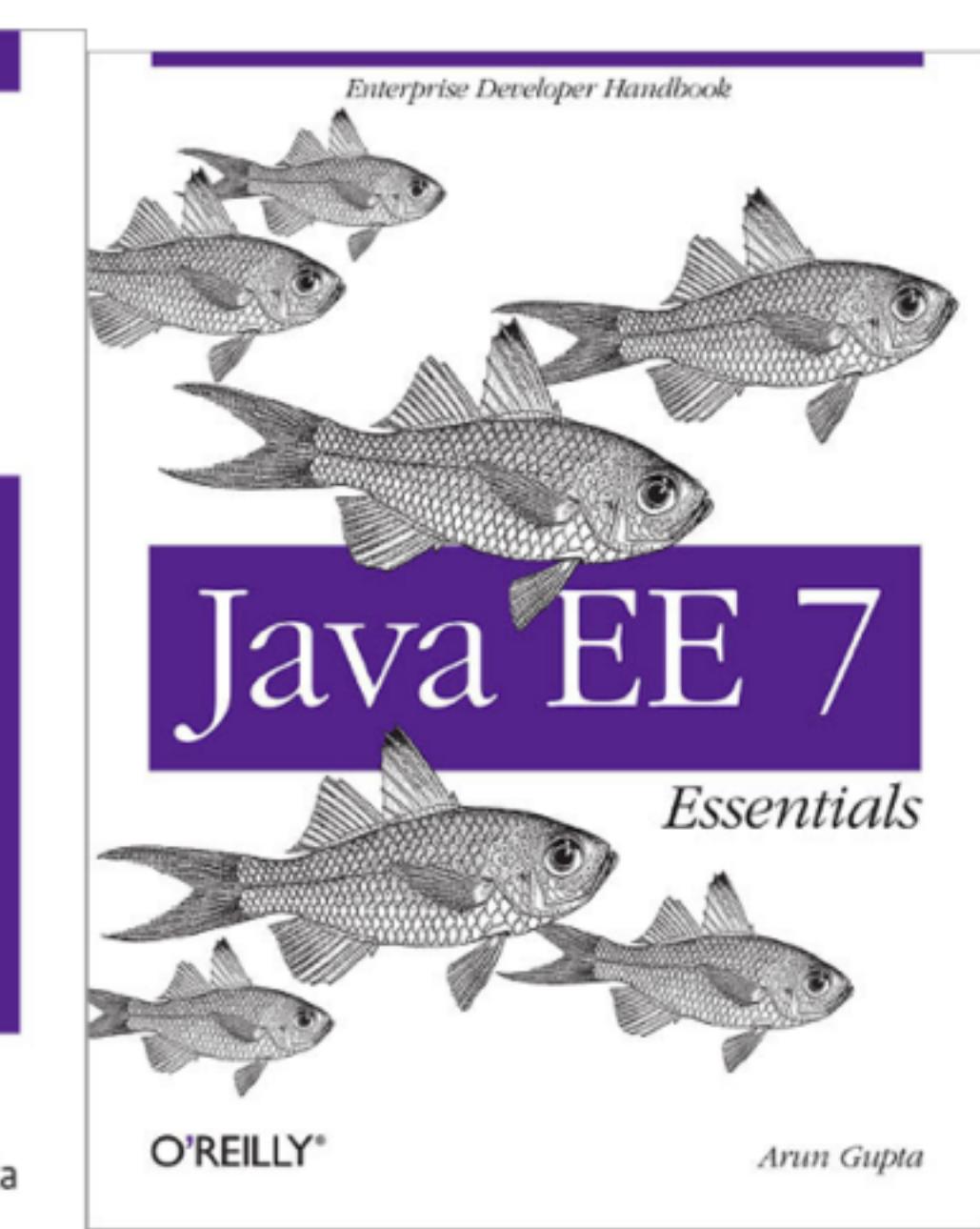
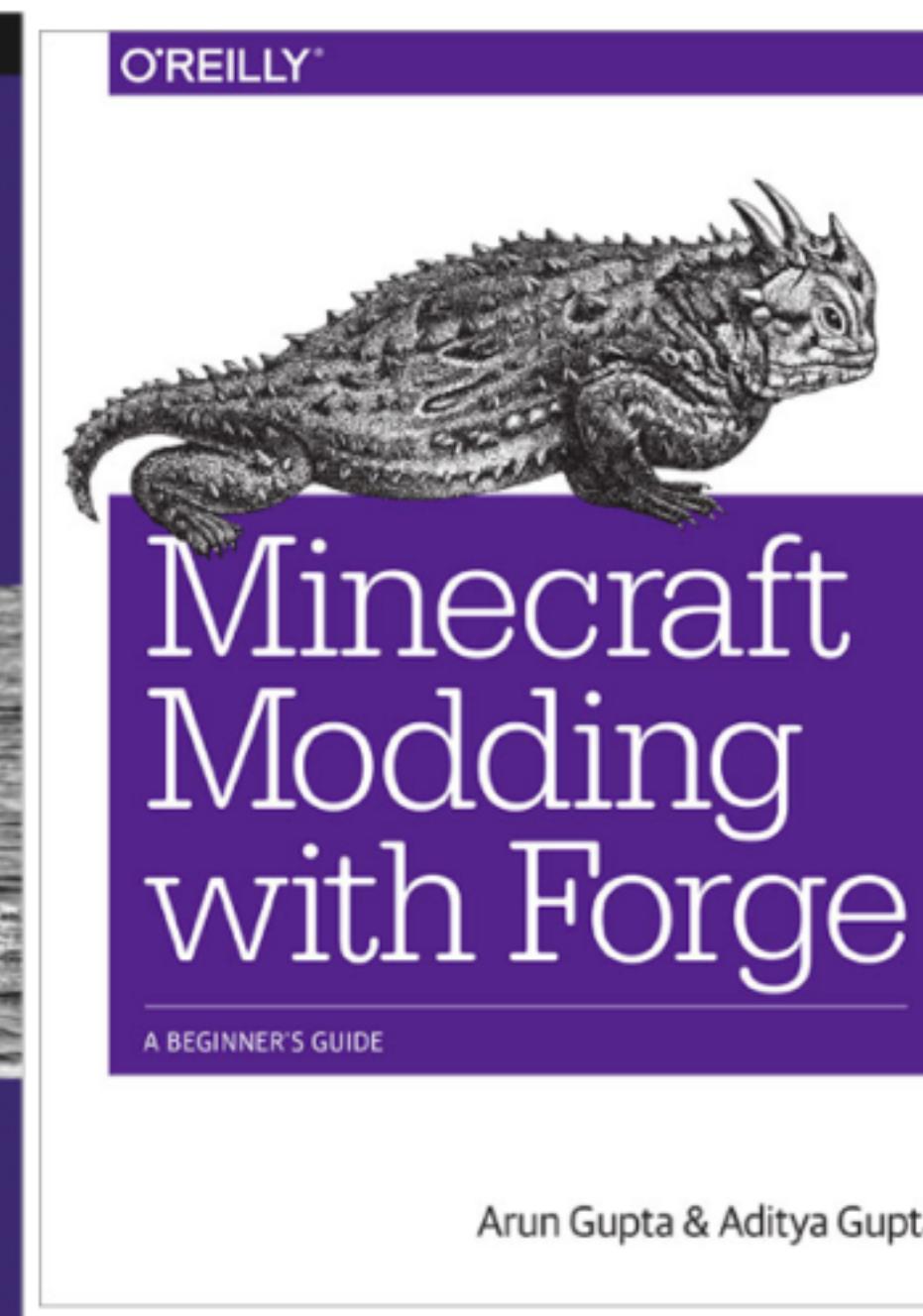
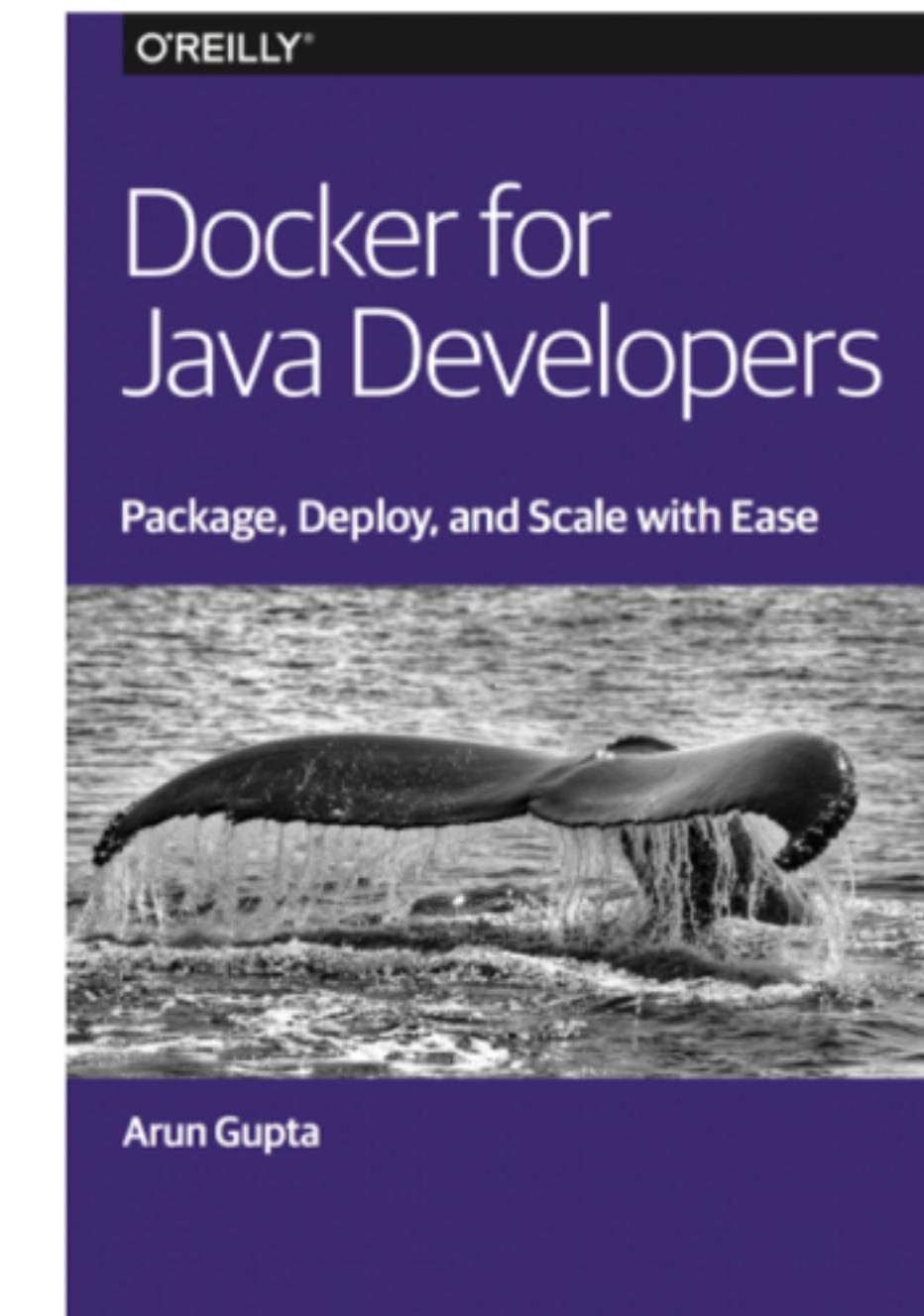


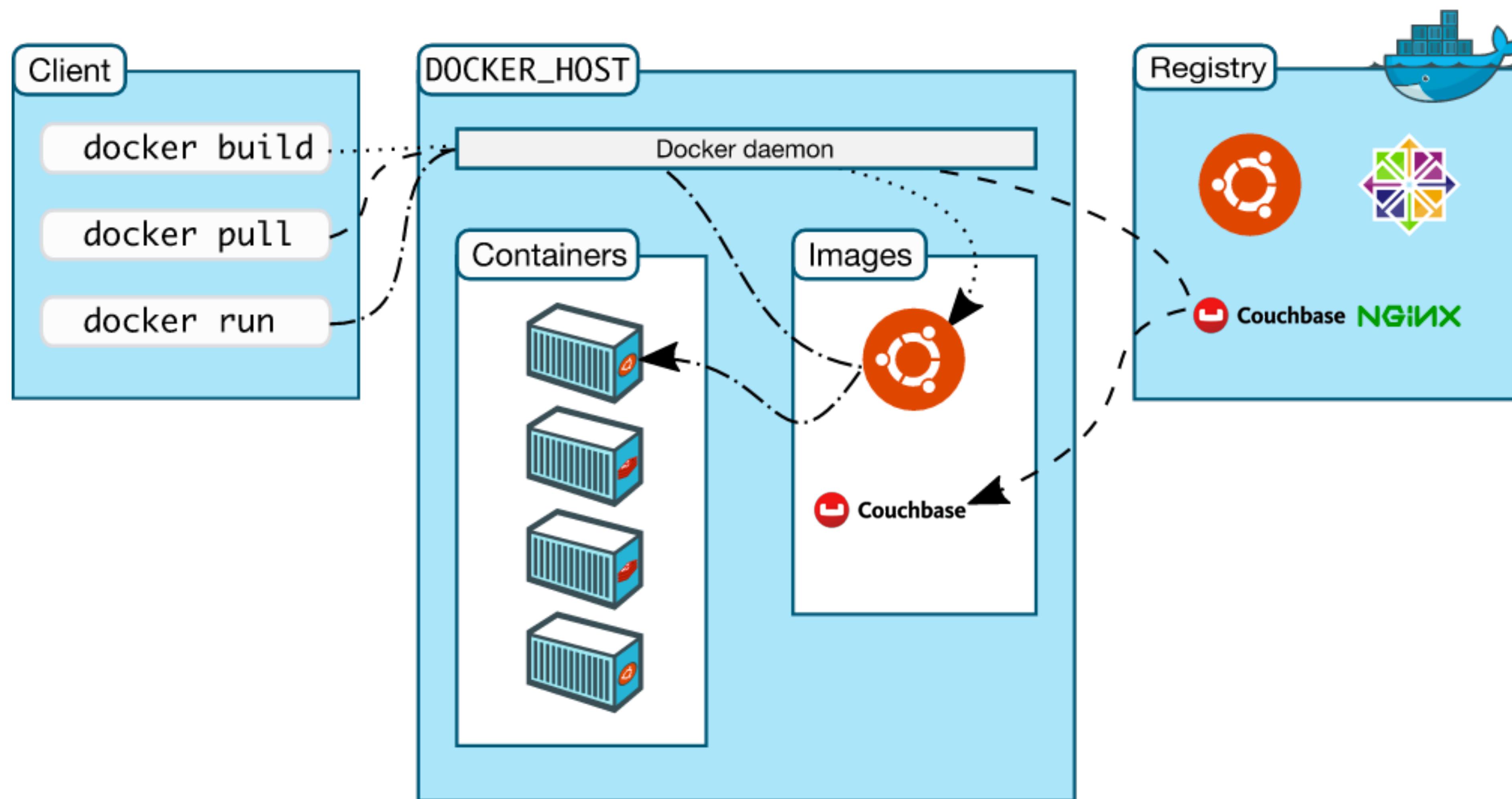


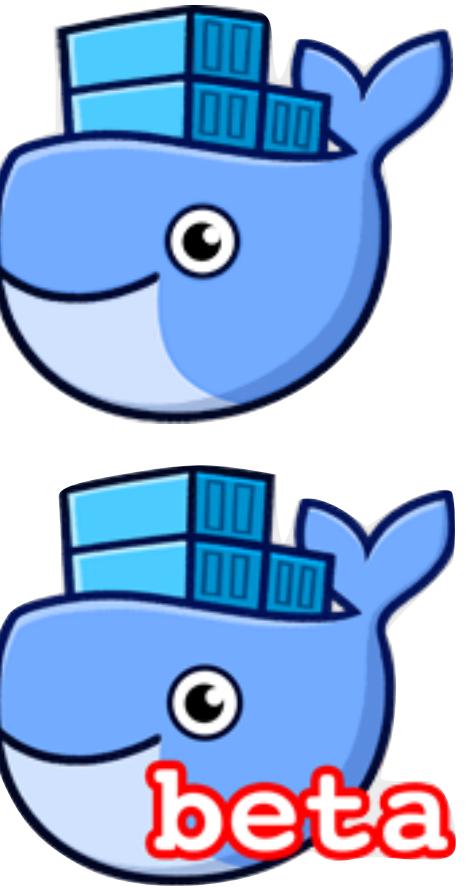
Docker and Kubernetes Recipes

Arun Gupta, @arungupta
VP Developer Advocacy, Couchbase



Docker Workflow





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

Docker for AWS/Azure

- Amazon Web Services
 - Amazon CloudFormation templates
 - Integrated with Autoscaling, ELB, and EBS
- Azure
 - Integrated with VM Scale Sets for autoscaling, Azure Load Balancer, Azure Storage
- beta.docker.com



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 - One Service

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

docker-compose up -d

Docker Compose - Two Services



Docker Compose - Two Services

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



Overriding Services in Docker Compose

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

docker-compose.yml

```
web:  
  ports:  
    - 9080:8080
```

docker-compose.override.yml

docker-compose up -d

Dev/Prod with Compose

```
db-dev:  
  image: arungupta/couchbase  
  ports:  
    - . . .  
  
web:  
  image: arungupta/wildfly  
  environment:  
    - COUCHBASE_URI=db-dev:8093  
  ports:  
    - 8080:8080
```

docker-compose.yml

docker-compose up -d

```
web:  
  environment:  
    - COUCHBASE_URI=db-prod:8093  
  ports:  
    - 80:8080  
  
db-prod:  
  image: . . .
```

production.yml

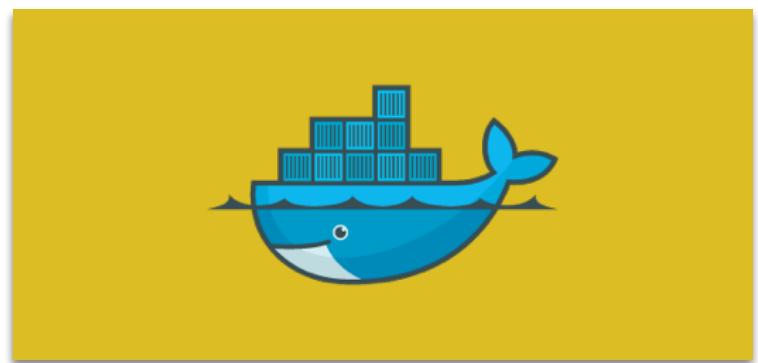
docker-compose up
-f docker-compose.yml
-f production.yml
-d



Swarm Mode

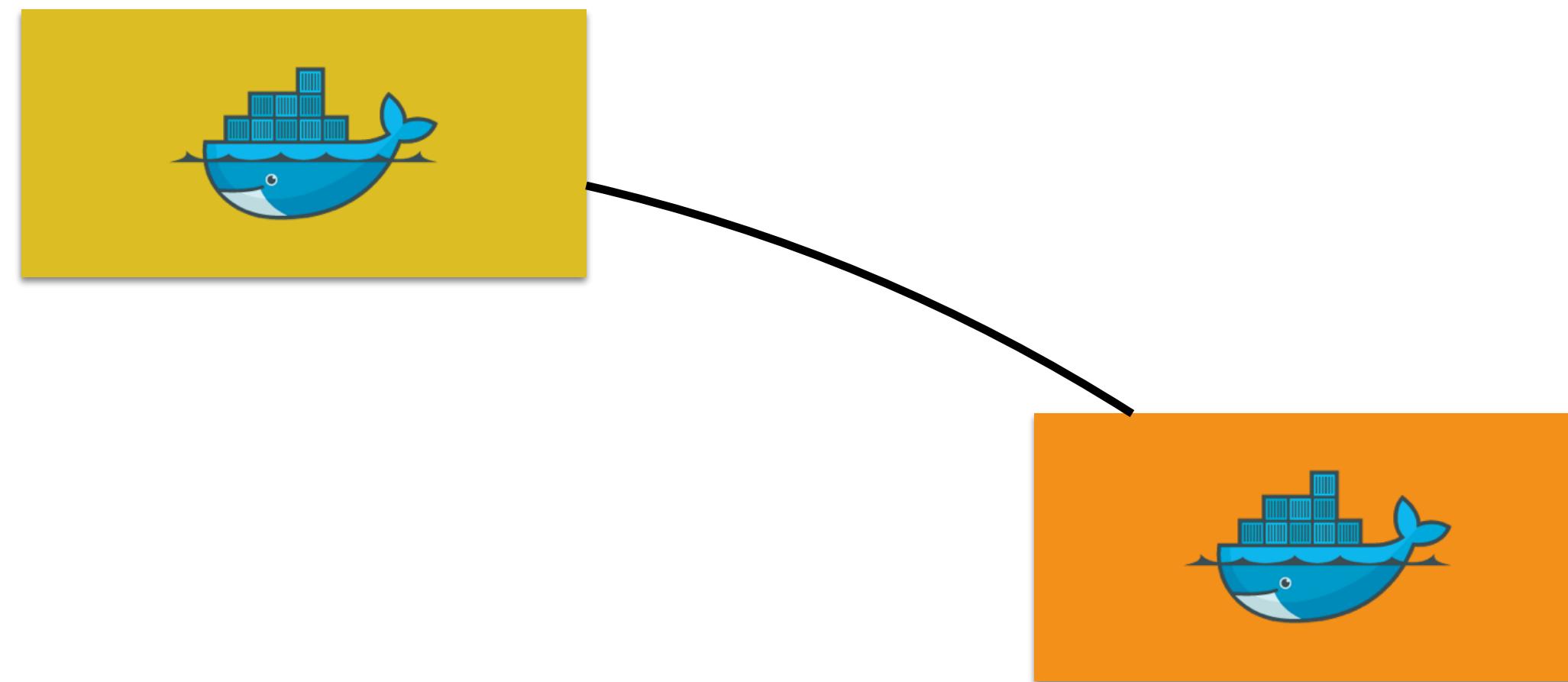
- New in 1.12
- Natively managing a cluster of Docker Engines called a Swarm
- Docker CLI to create a swarm, deploy apps, and manage swarm
- No Single Point of Failure (SPOF)
- Declarative state model
- Self-organizing, self-healing
- Service discovery, load balancing and scaling
- Rolling updates
- Optional feature, need to be explicitly enabled

Swarm Mode: Initialize



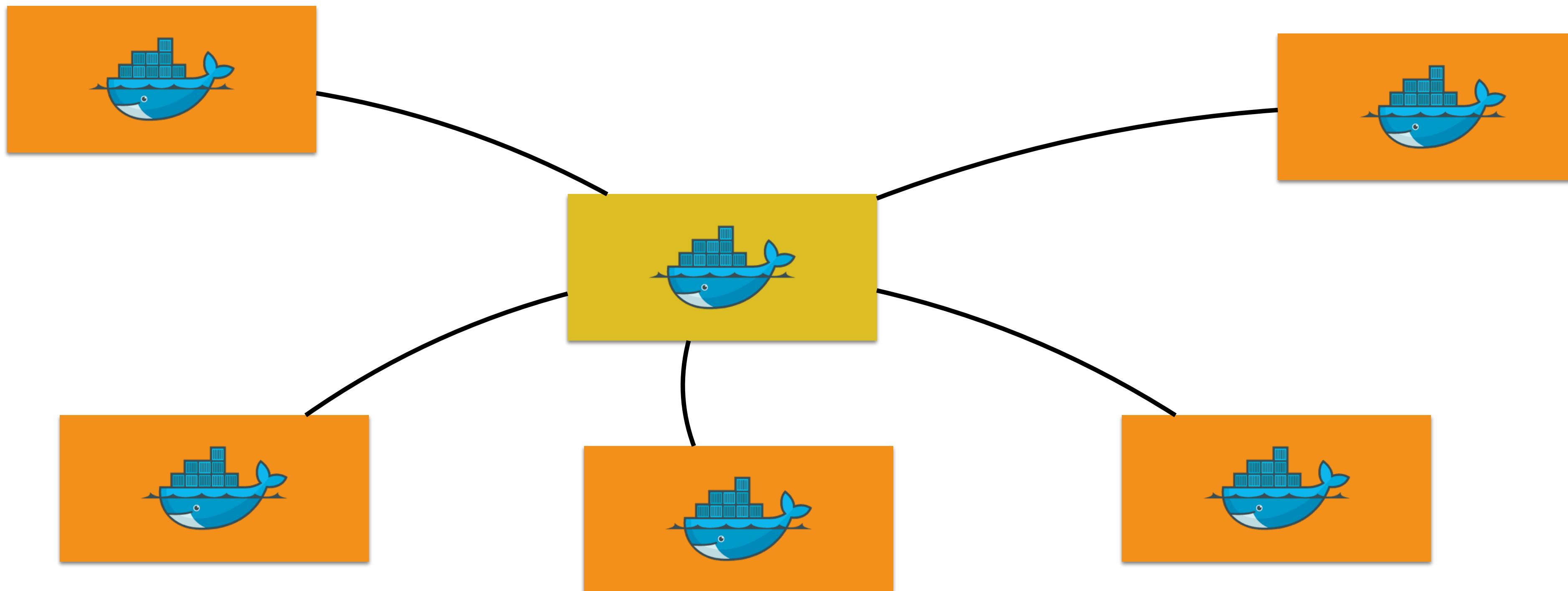
```
docker swarm init --listen-addr <ip>:2377
```

Swarm Mode: Add Worker



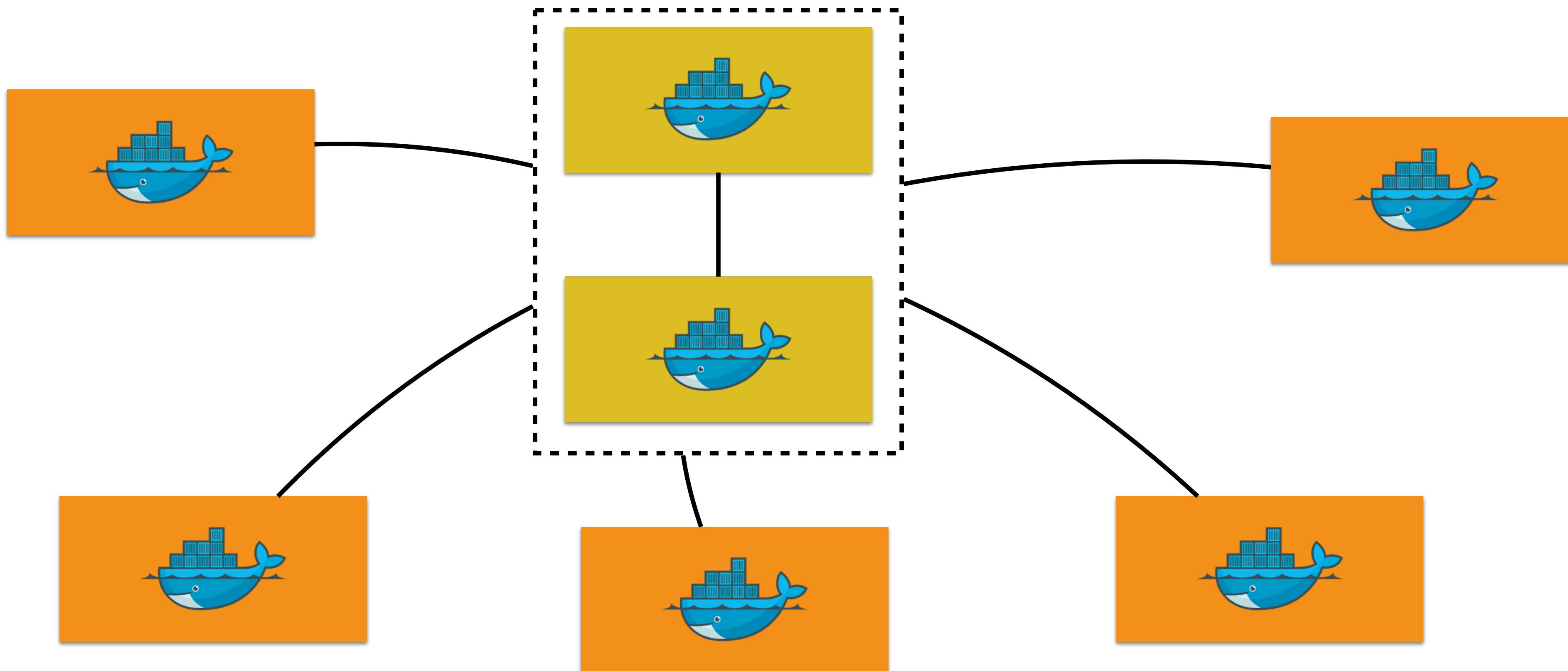
```
docker swarm join --token <worker_token> <manager>:2377
```

Swarm Mode: Add More Workers



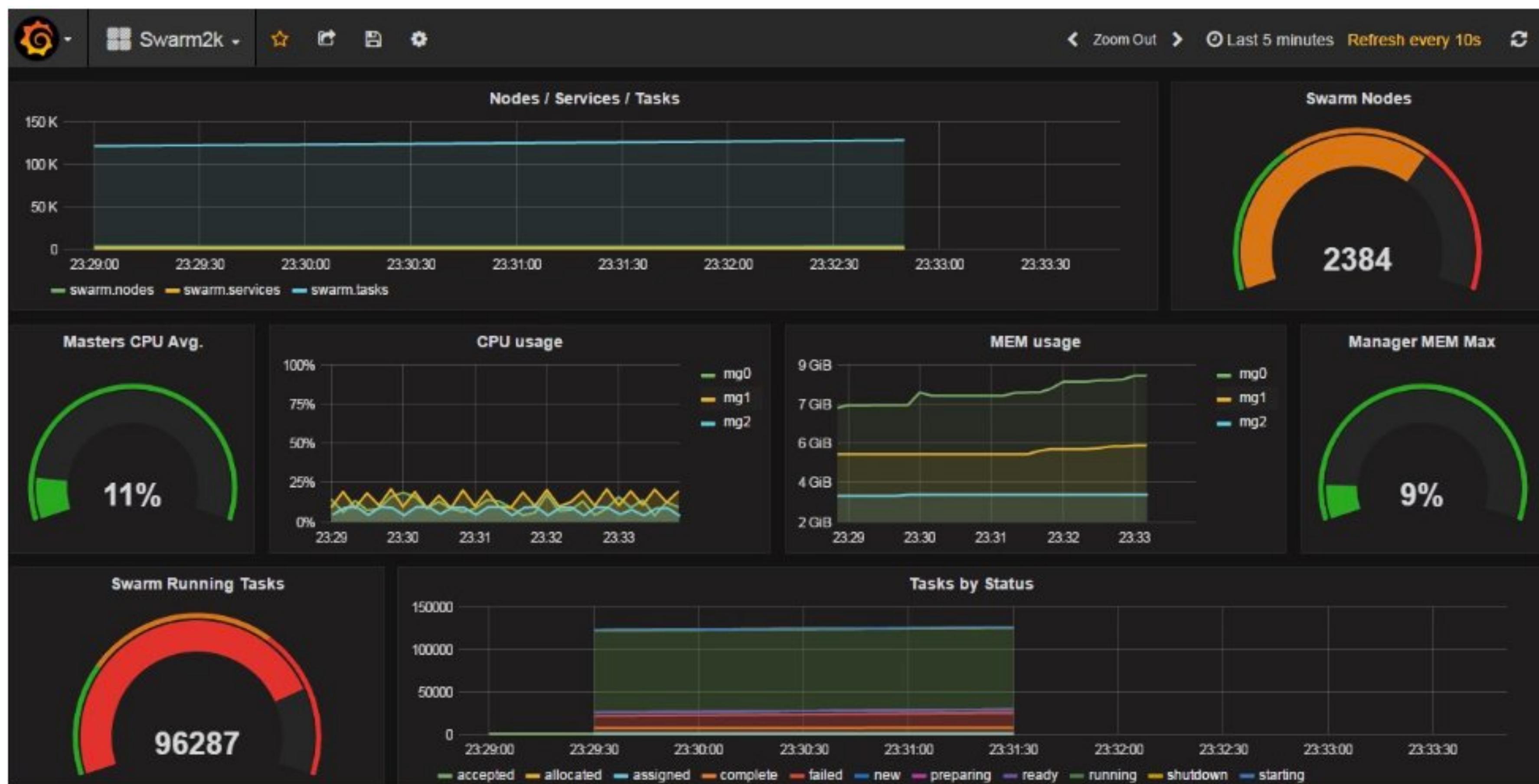
```
docker swarm join --token <worker_token> <manager>:2377
```

Swarm Mode: Primary/Secondary Master

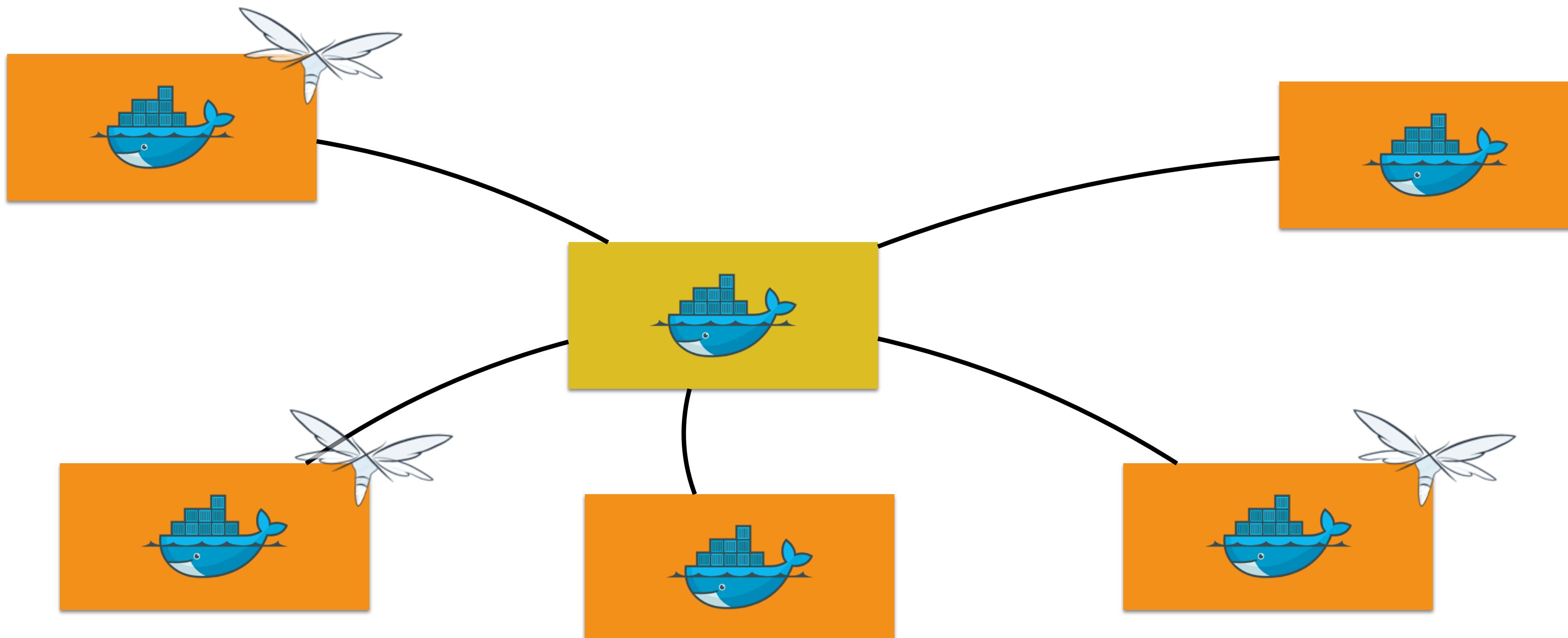


```
docker swarm join --manager --token <manager_token> --listen-  
addr <master2>:2377 <master1>:2377
```

Swarm Mode in Production

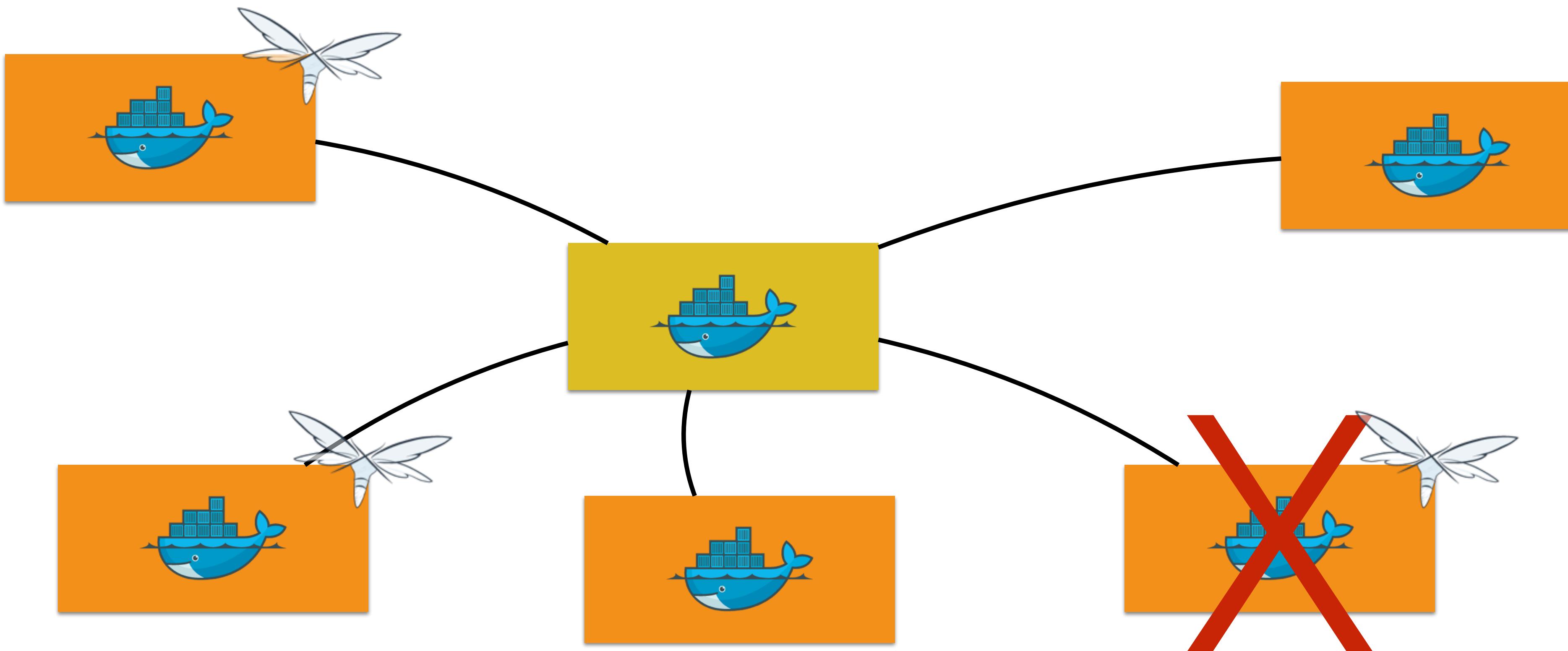


Swarm Mode: Replicated Service

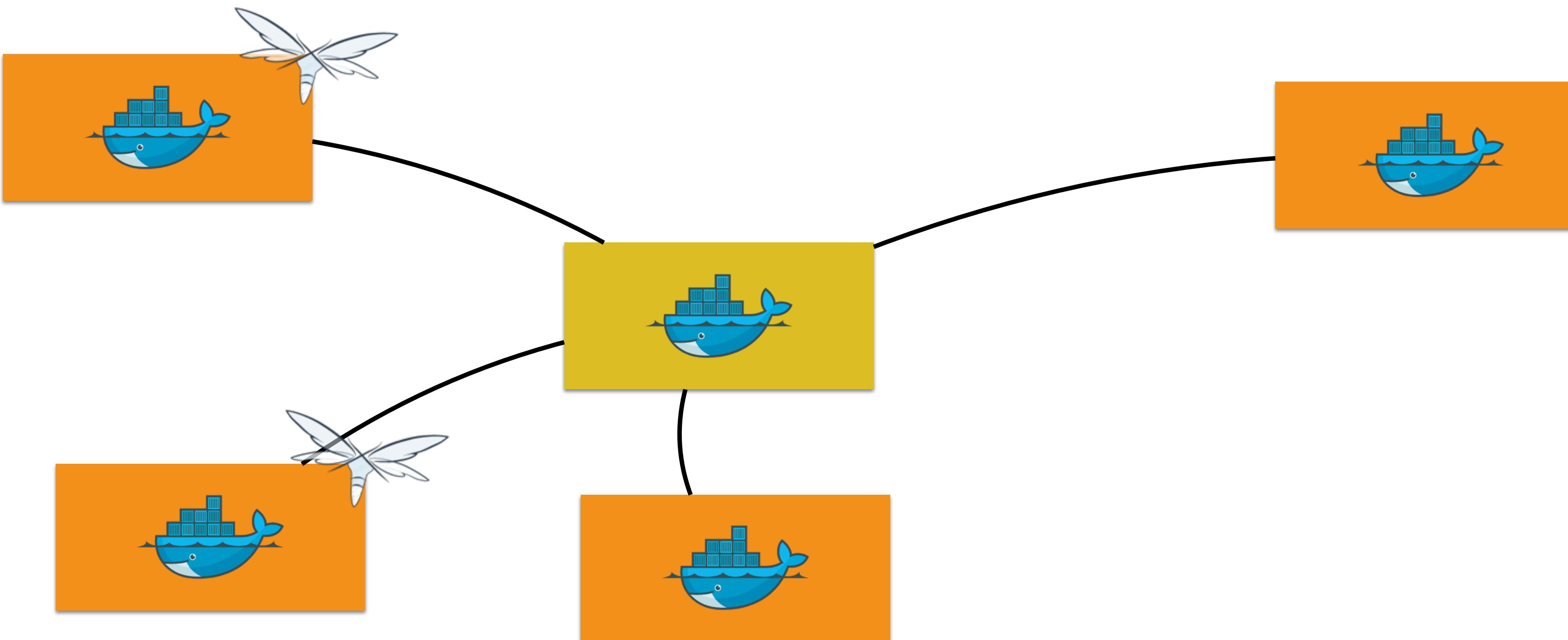


```
docker service create --replicas 3 --name web jboss/wildfly
```

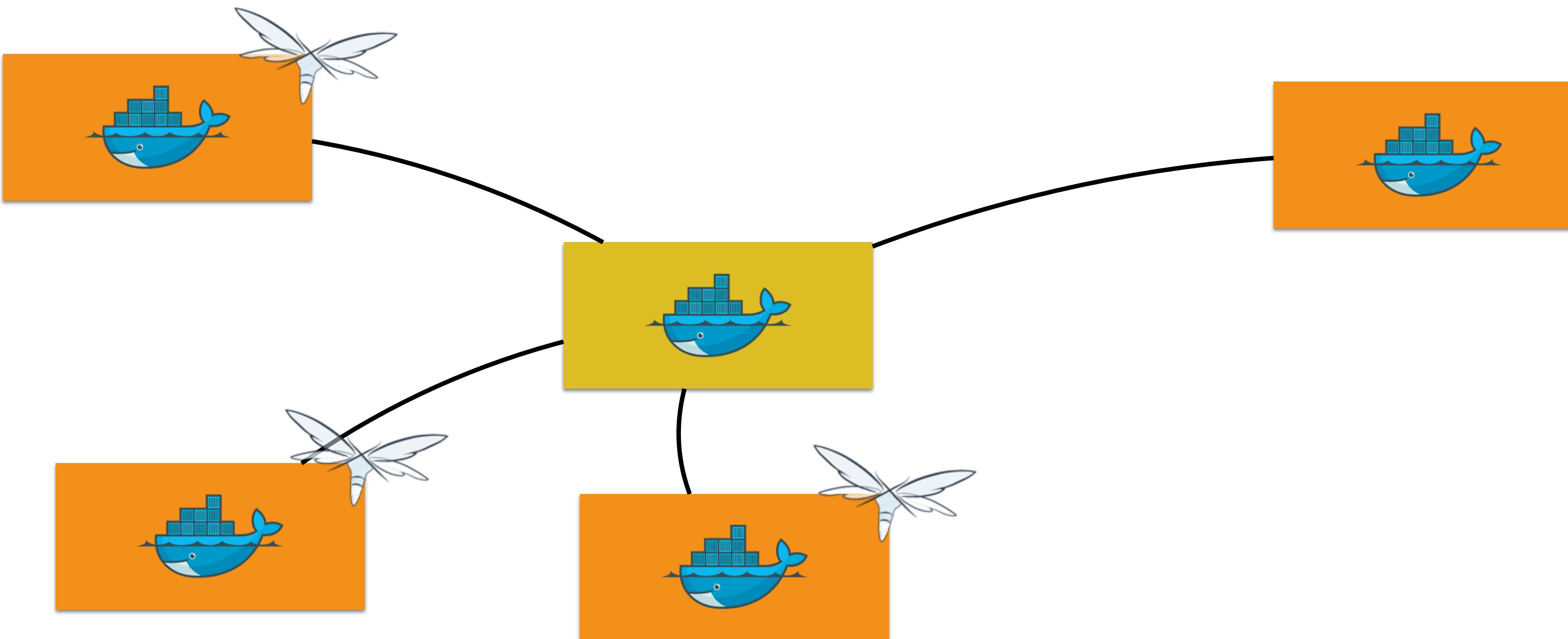
Swarm Mode: Node Failure



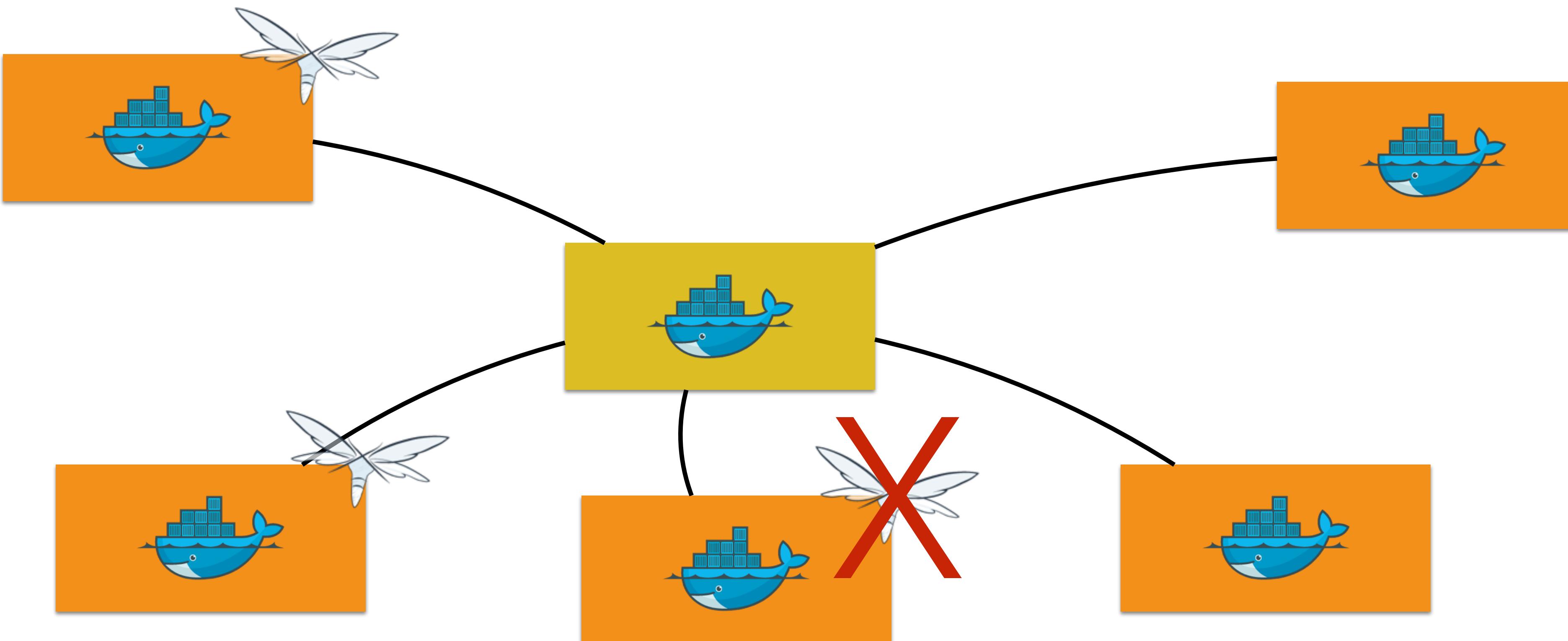
Swarm Mode: Desired != Actual



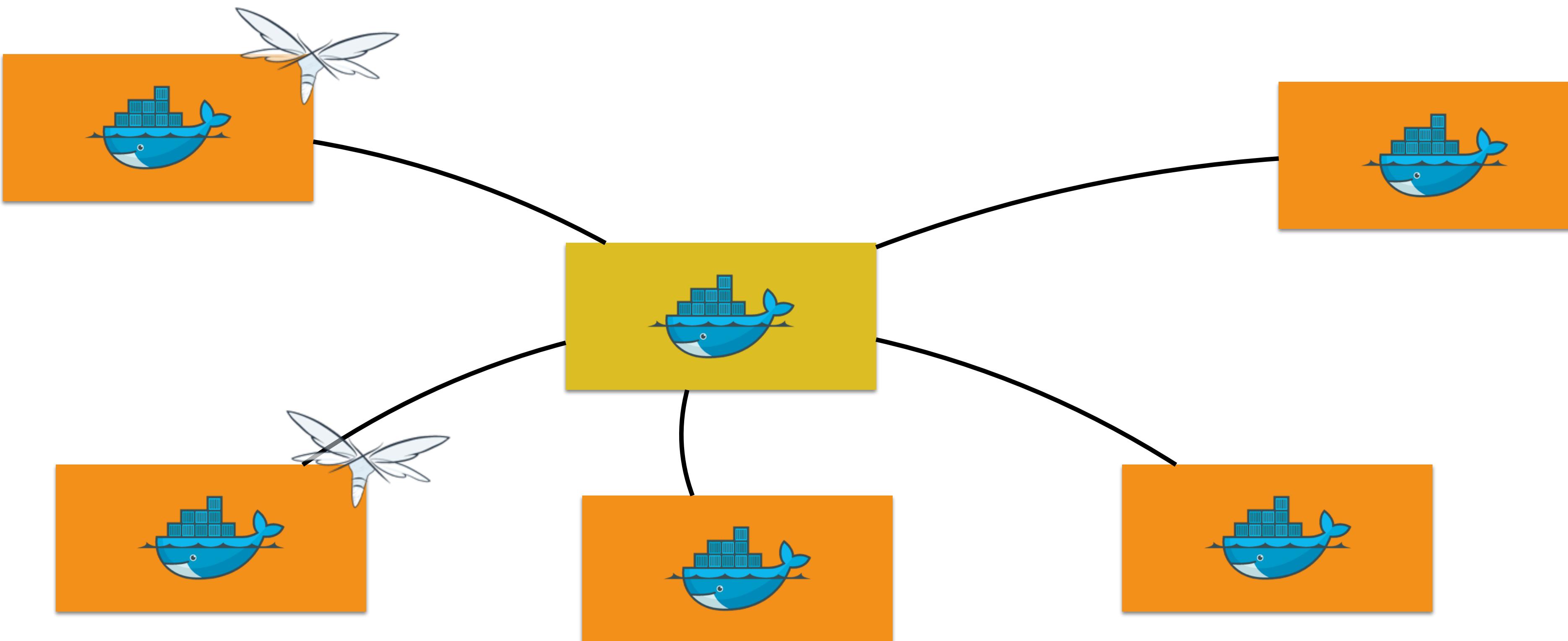
Swarm Mode: Reconcile



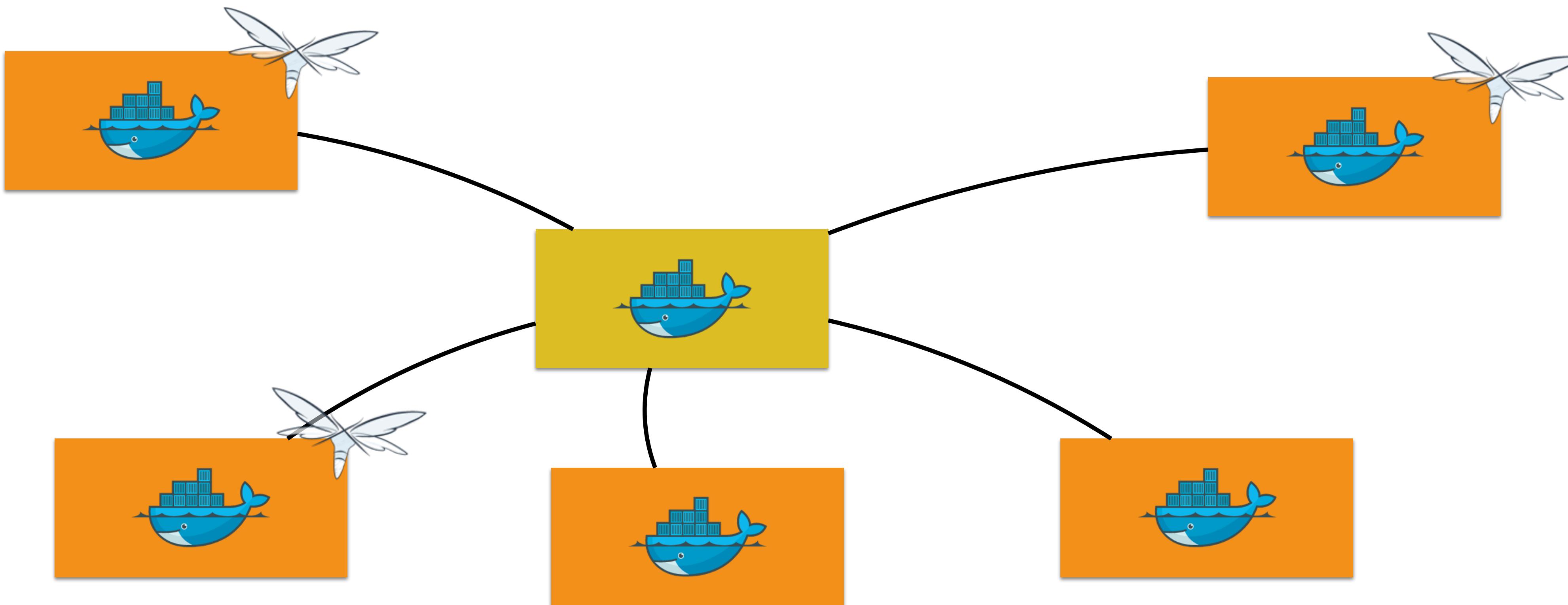
Swarm Mode: Container Failure



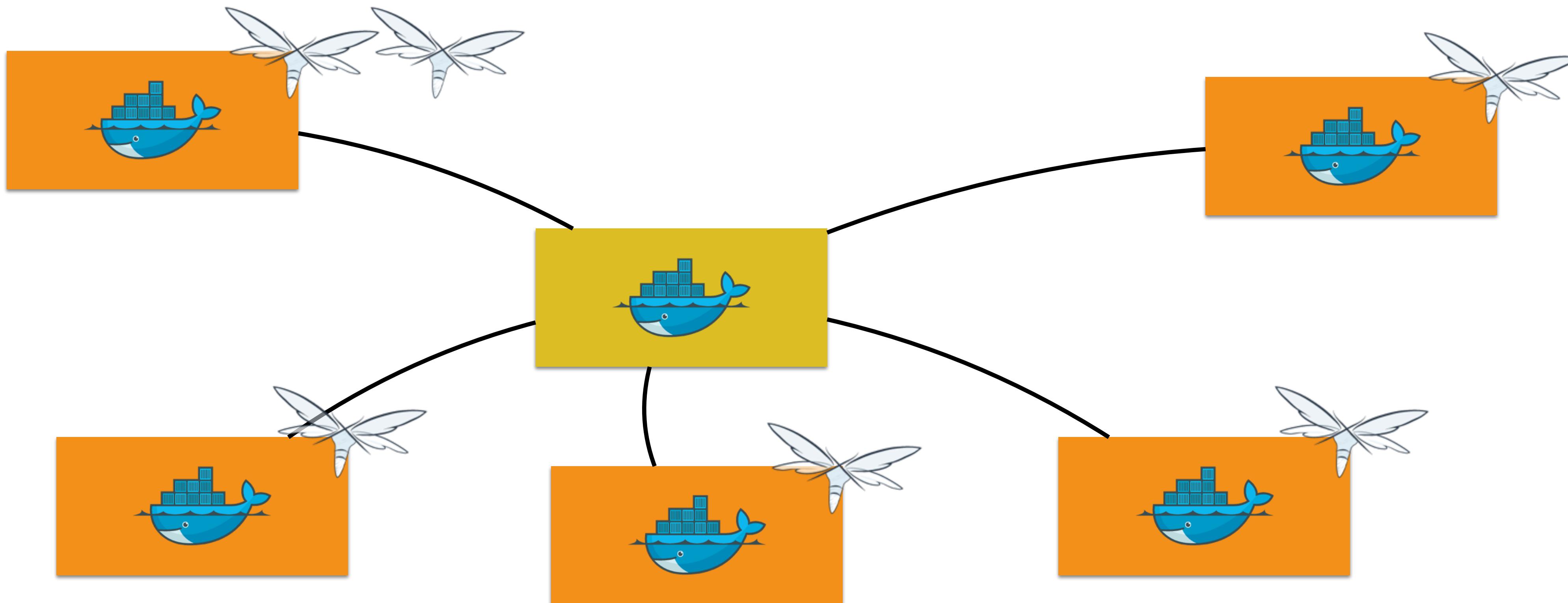
Swarm Mode: Desired != Actual



Swarm Mode: Reconcile

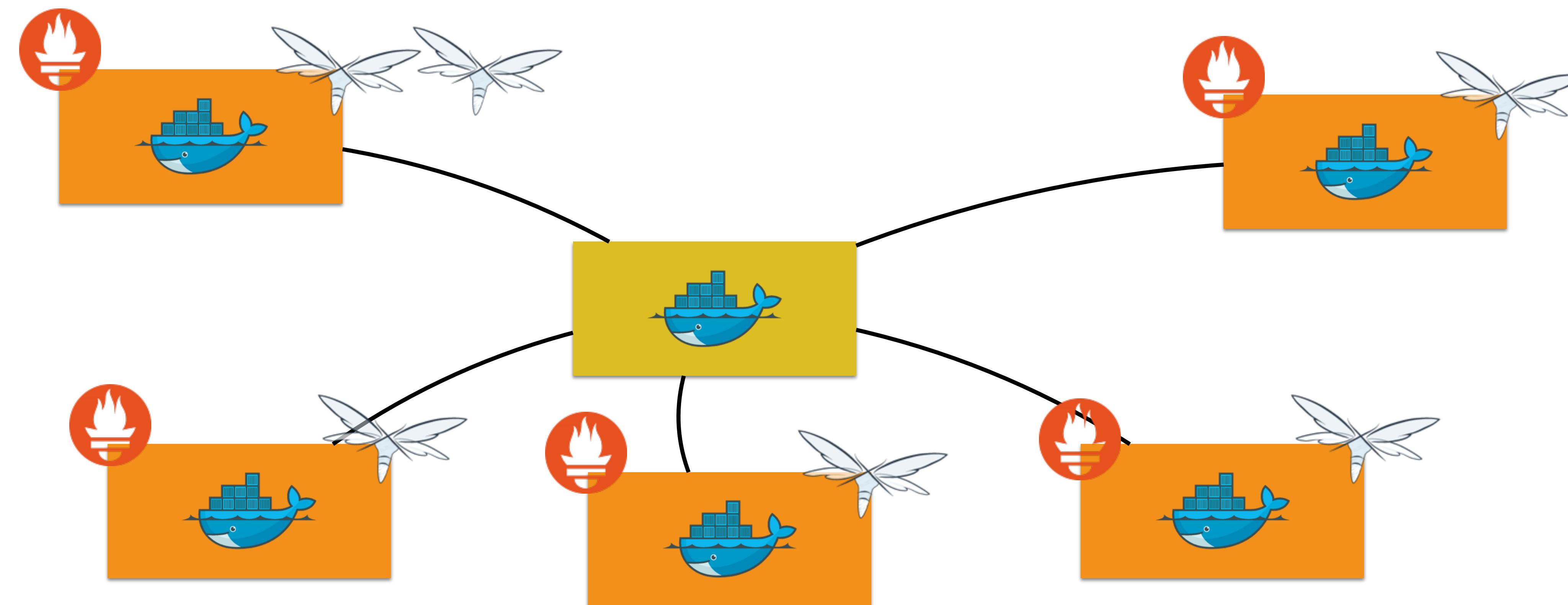


Swarm Mode: Scale



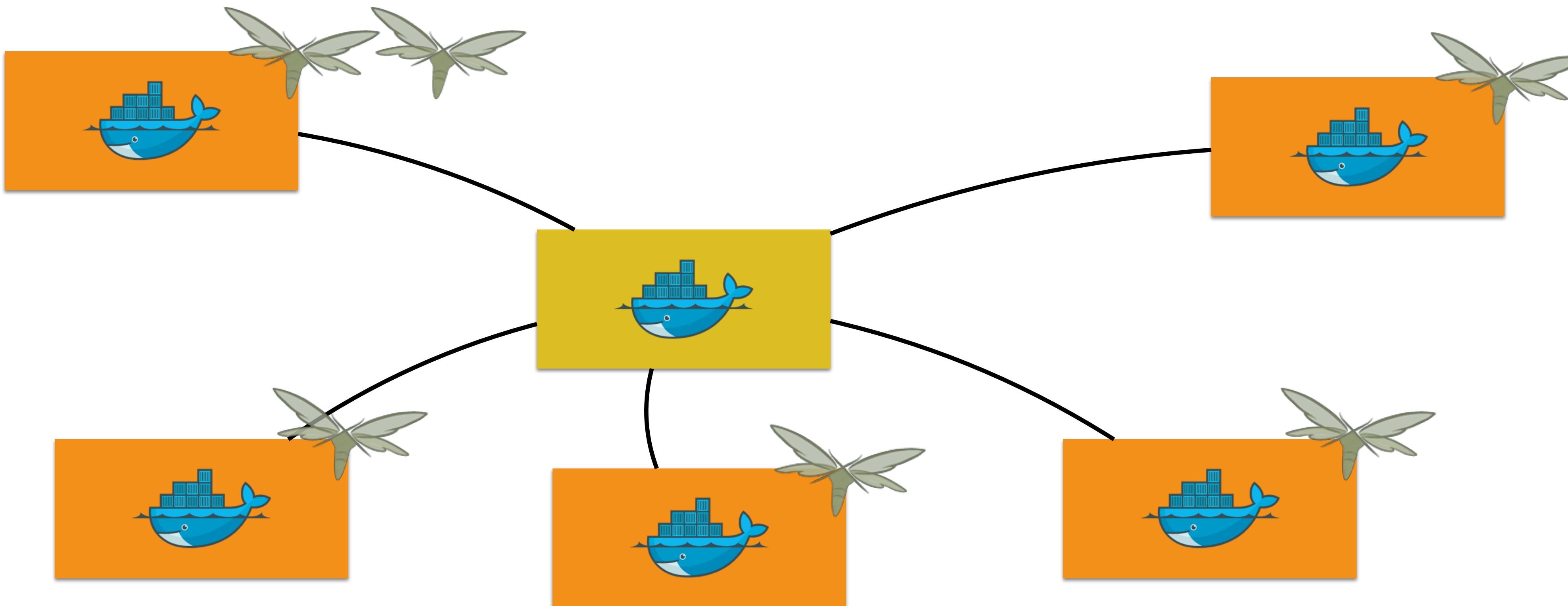
```
docker service scale web=6
```

Swarm Mode: Global Service



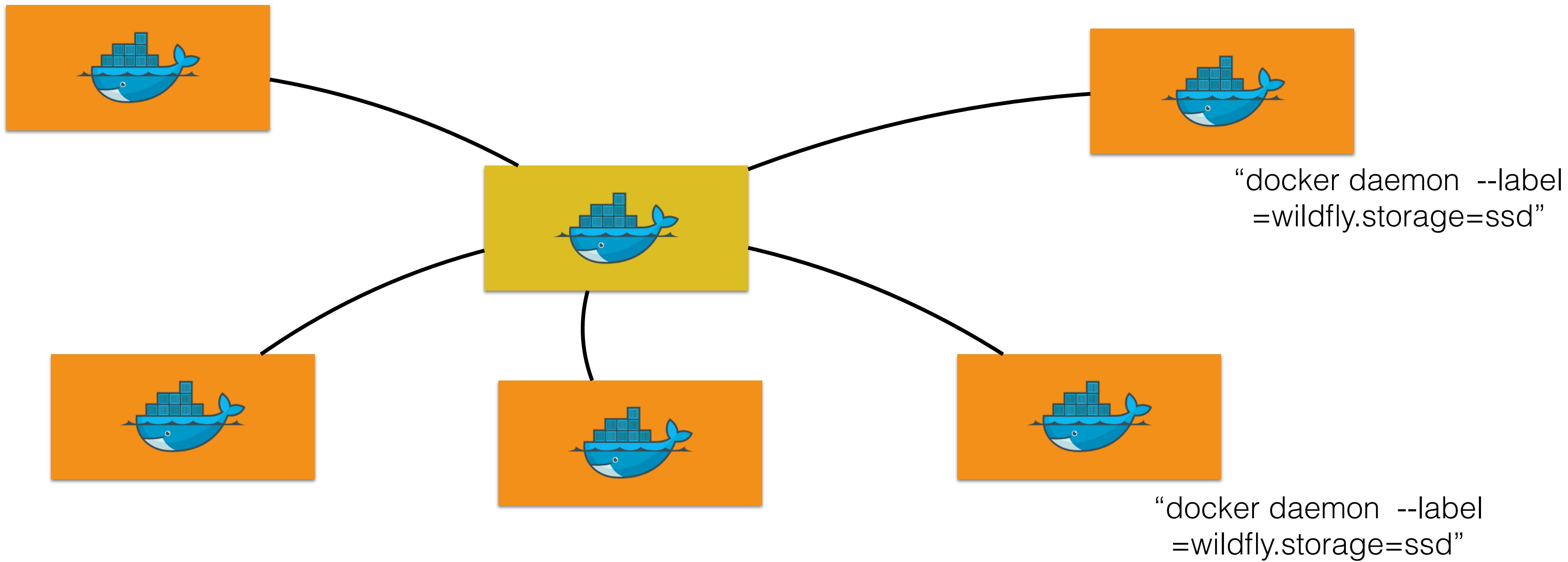
```
docker service create --mode=global --name=prom prom/prometheus
```

Swarm Mode: Rolling Updates



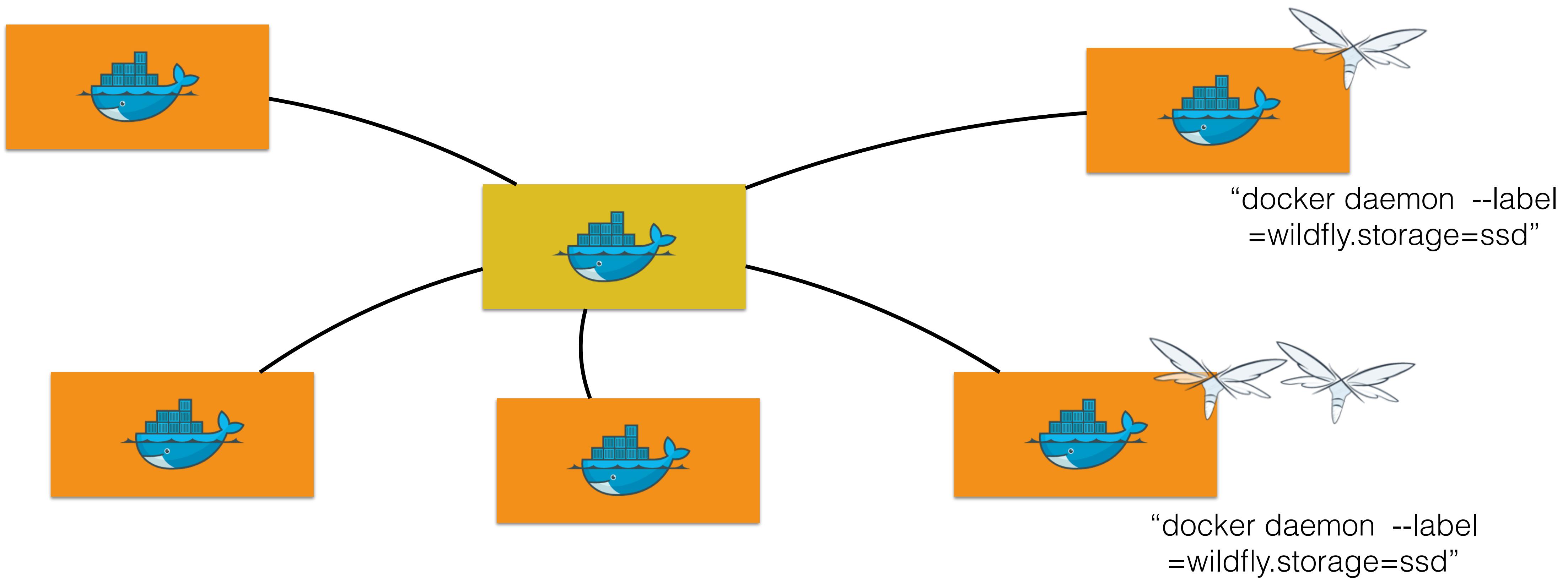
```
docker service update web --image wildfly:2 --update-parallelism  
2 --update-delay 10s
```

Swarm Mode: Label



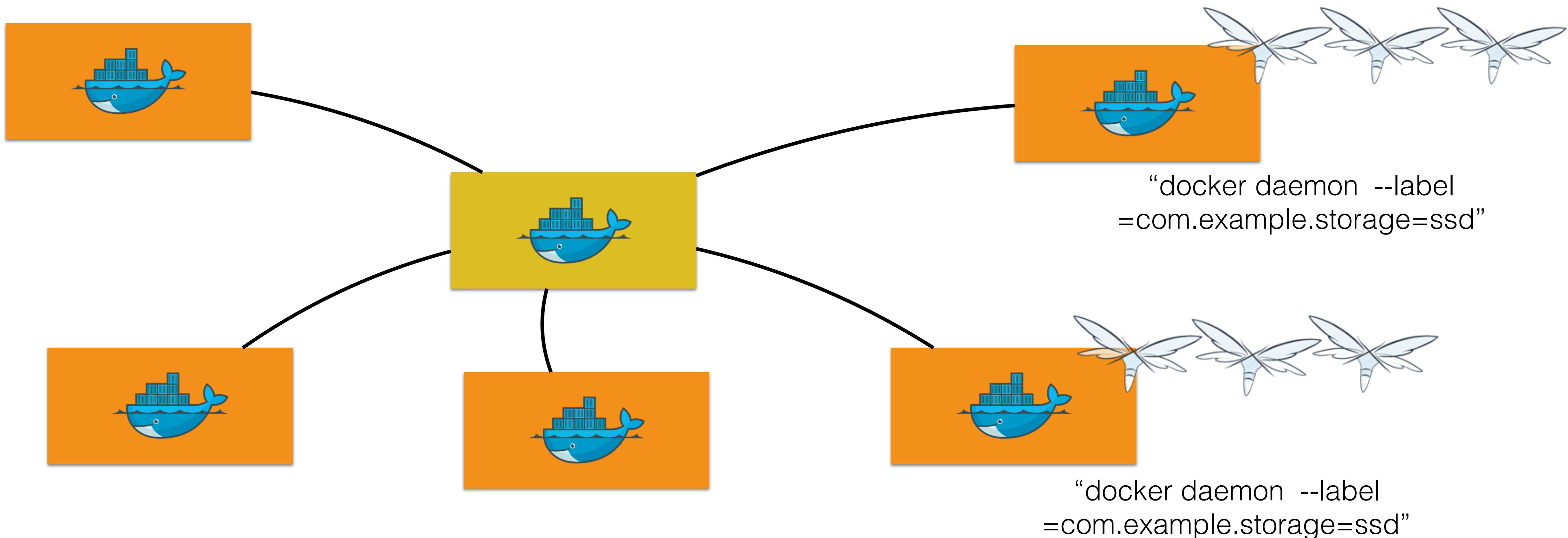
```
DOCKER_OPTS="--label=wildfly.storage=ssd"
```

Swarm Mode: Constraints



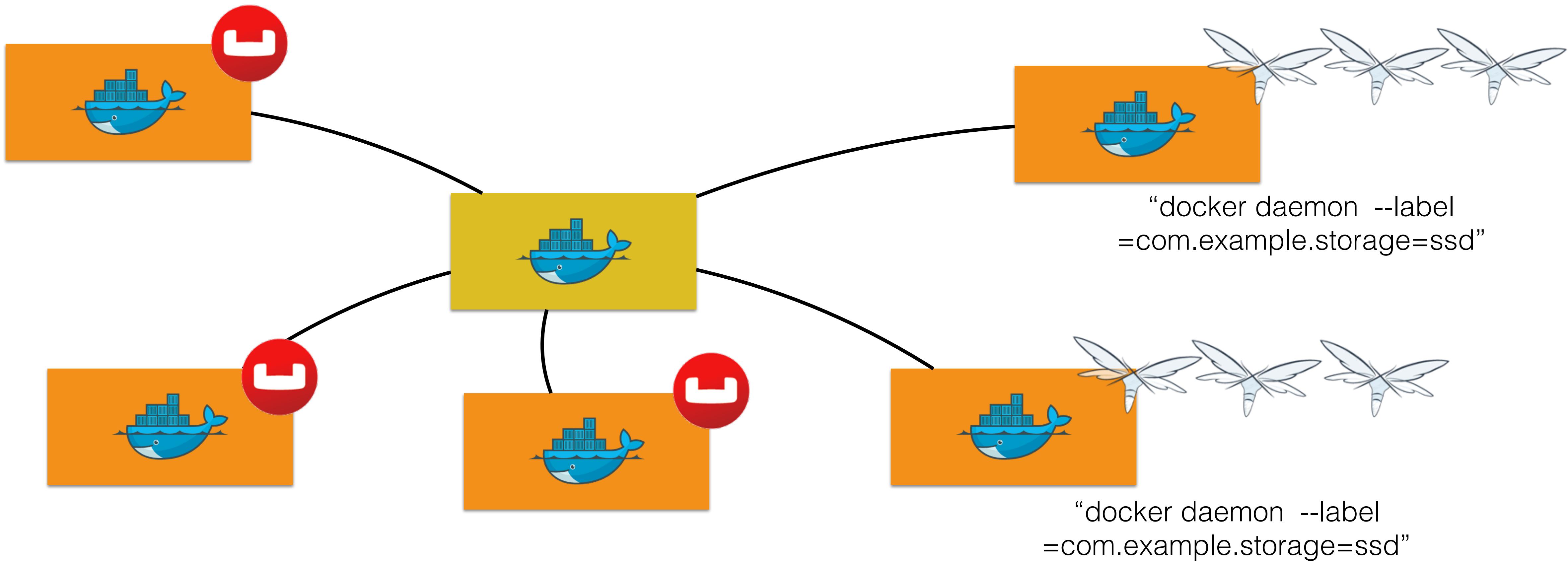
```
docker service create --replicas=3 --name=web --constraint  
engine.labels.wildfly.storage==ssd jboss/wildfly
```

Swarm Mode: Constraints



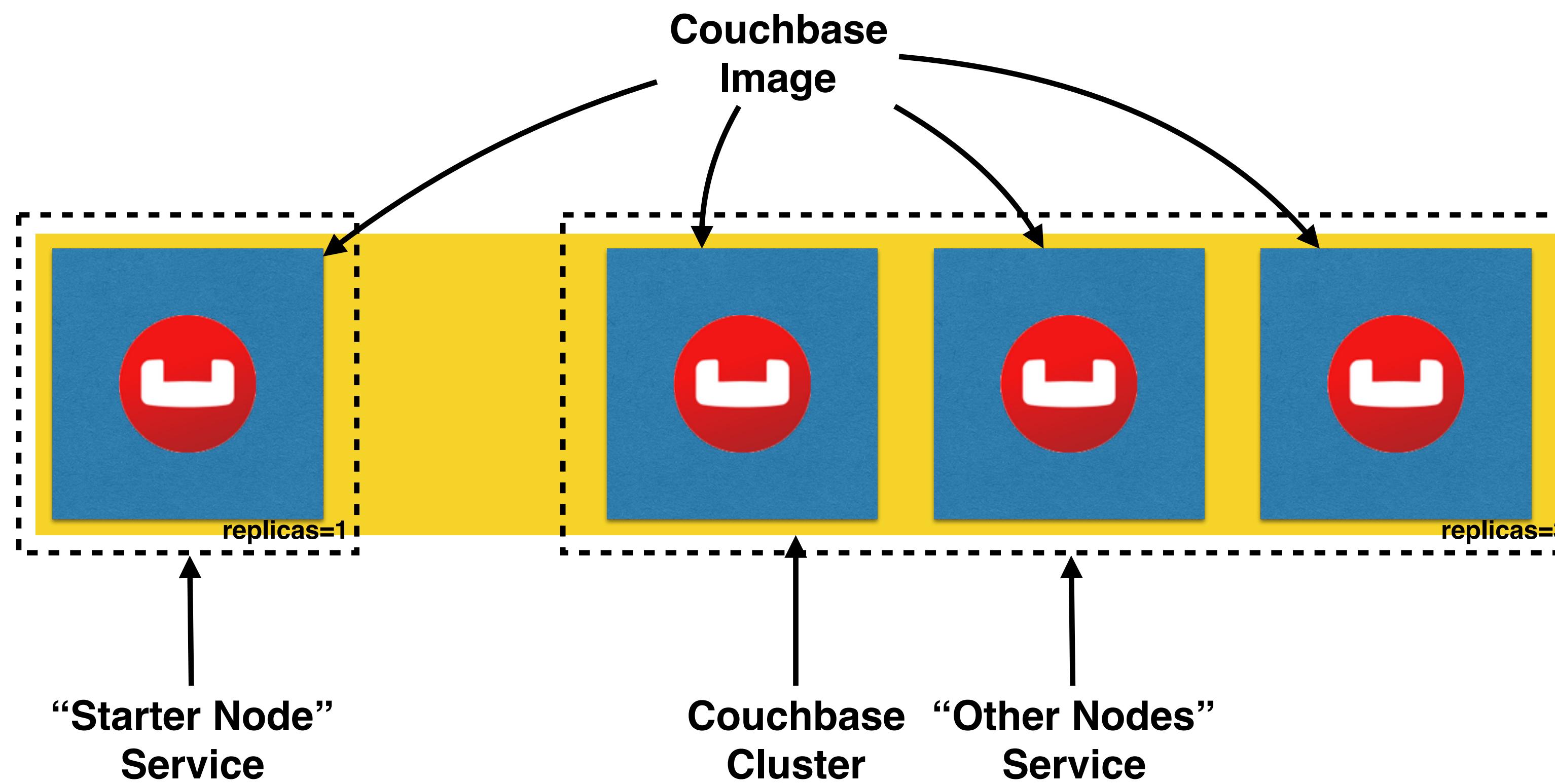
```
docker service scale web=6
```

Swarm Mode: Constraints



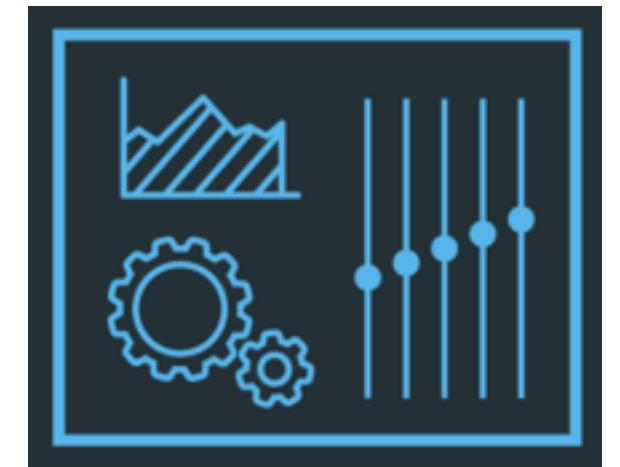
```
docker service create --replicas=3 --name=db couchbase
```

Couchbase Cluster using Docker Services



Monitoring Docker Containers

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



cAdvisor



Docker Support in Java IDEs

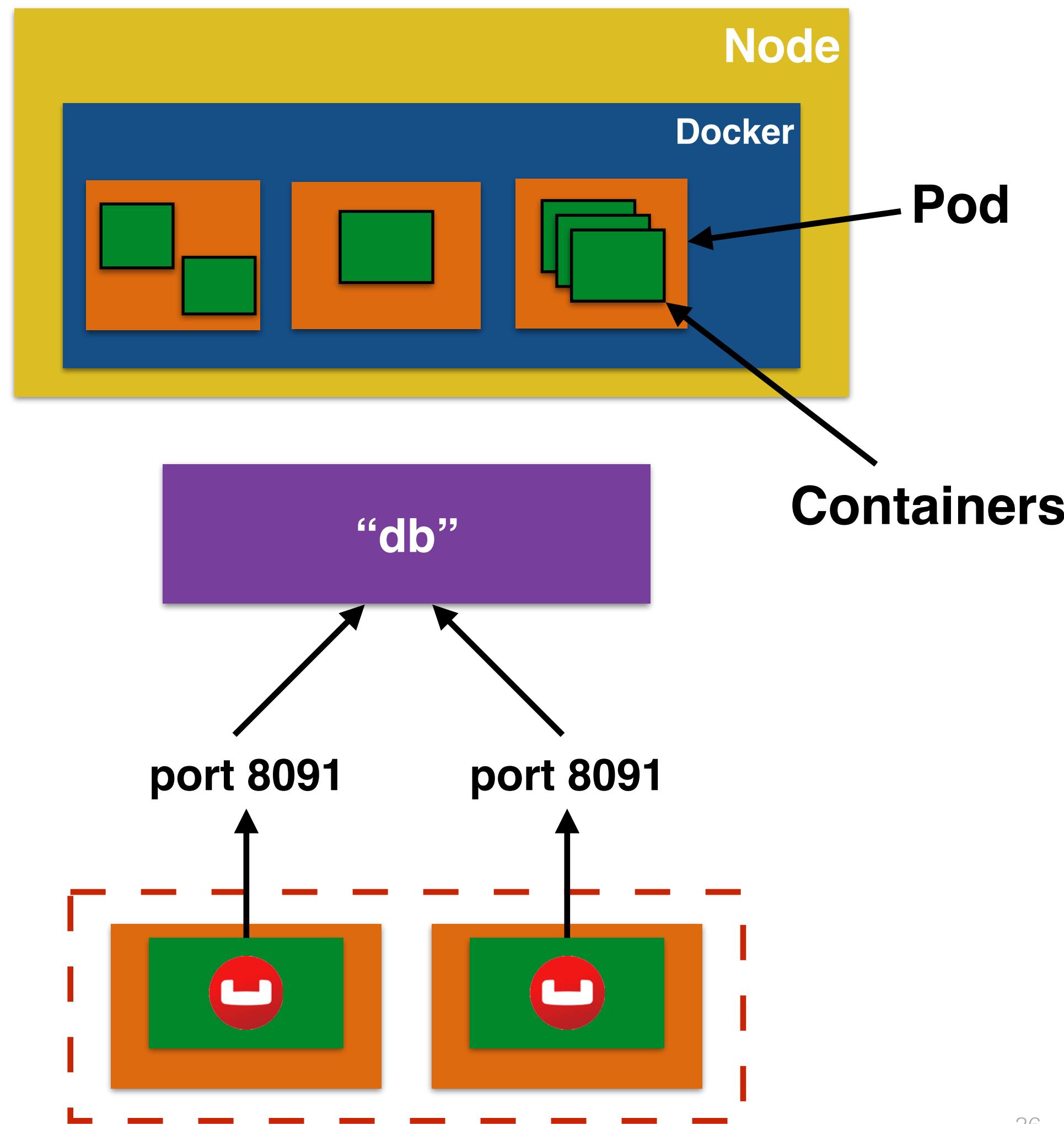


Kubernetes

- Open source orchestration system for containers
 - Docker, rkt, OCI, ...
- Provide declarative primitives for the “desired state”
 - Self-healing
 - Auto-restarting
 - Schedule across hosts
 - Replicating

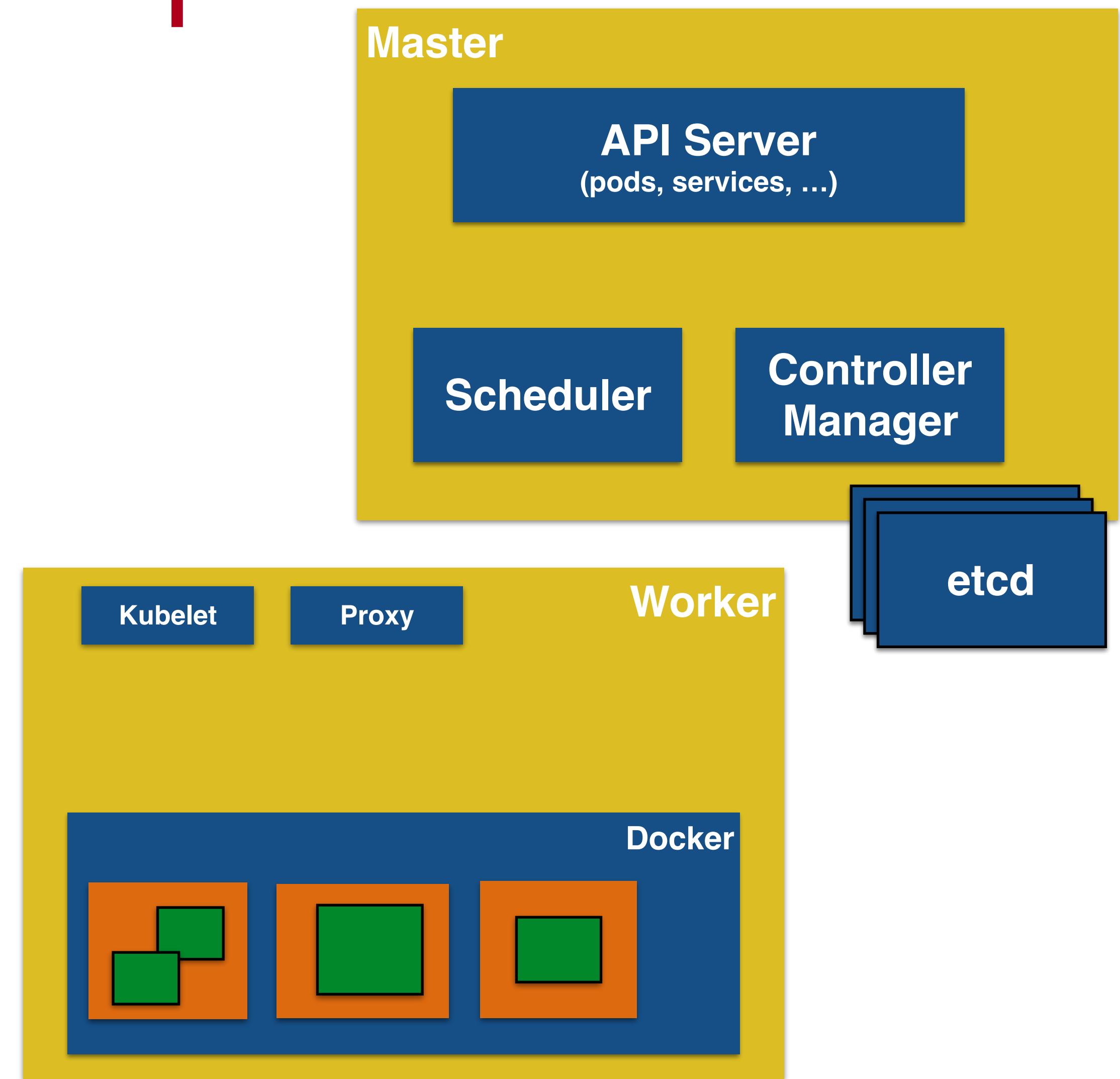
Kubernetes Concepts

- **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

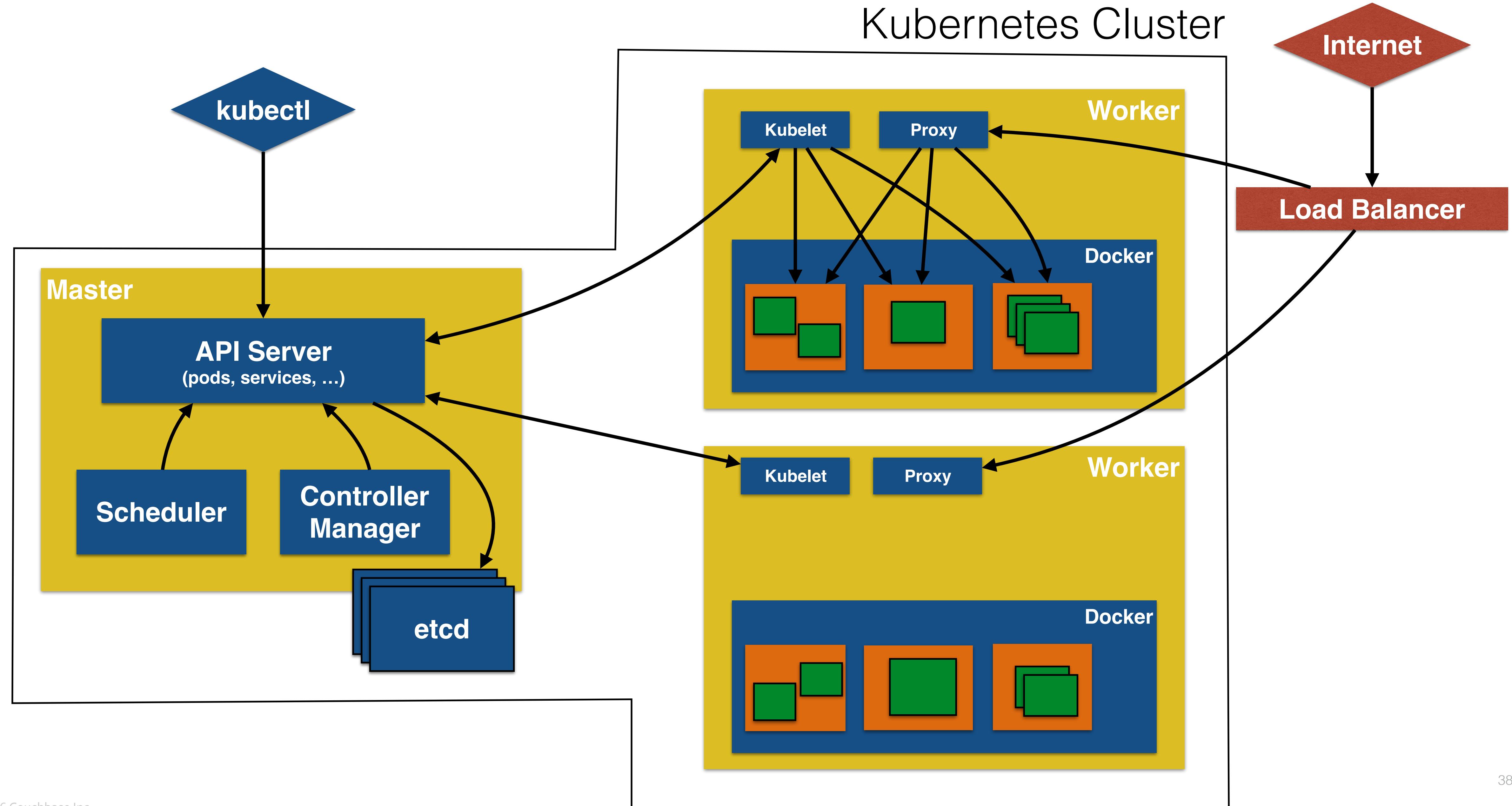


Kubernetes Components

- **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)



Kubernetes Cluster



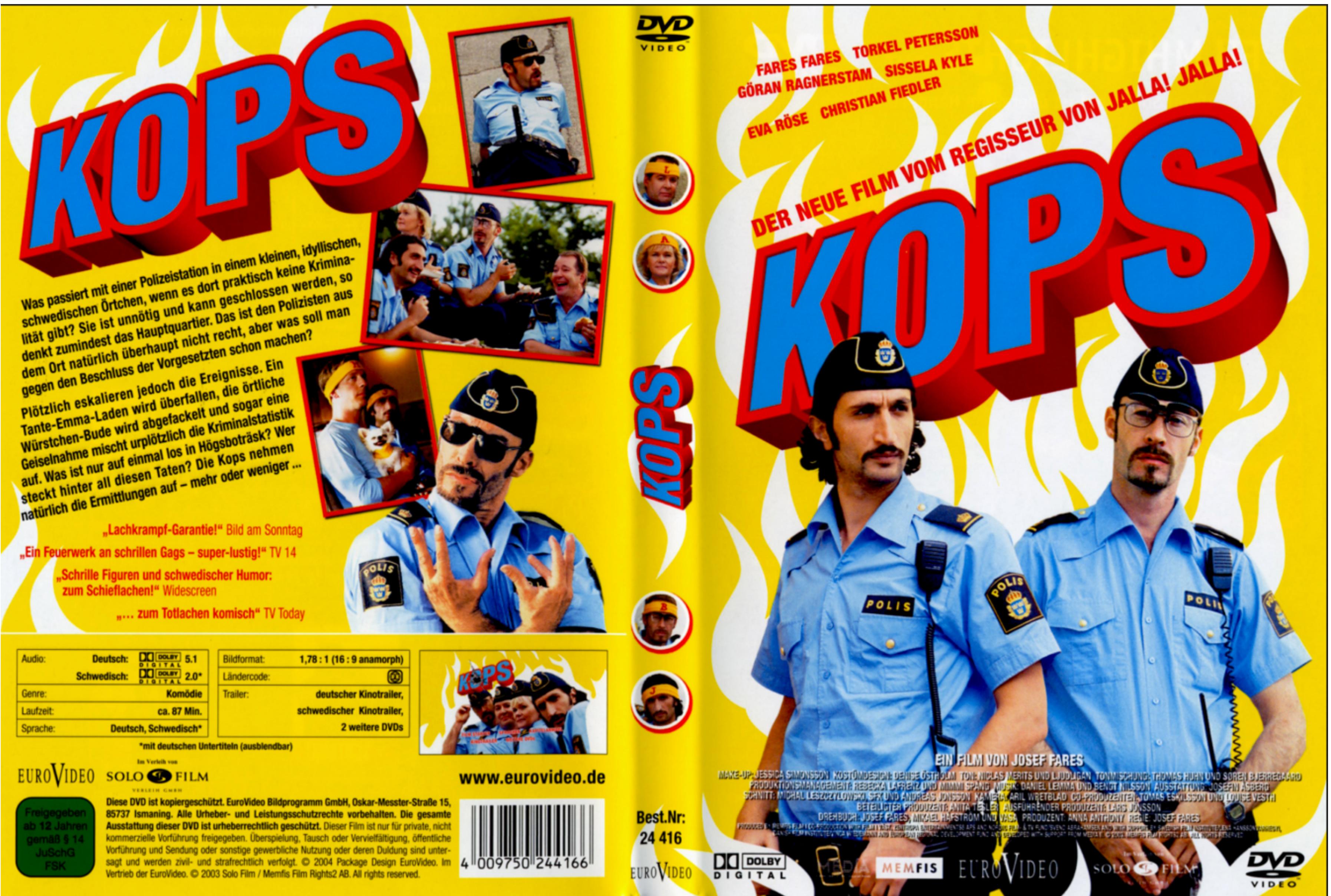
kubectl

- Controls the Kubernetes cluster manager
- CRUD Kubernetes resources
 - `create, get, describe, delete, ...`
 - `kubectl create -f <filename>`
- `kubectl get nodes or pods`
- `kubectl scale --replicas=3 rc/<name>`



Minikube

- Runs a single node cluster in a VM
- Targeted for local development
- `minikube start, stop, docker-env, ...`
- Requires `kubectl` CLI
- github.com/kubernetes/minikube/releases



Kubernetes Pod Configuration

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: wildfly-pod
5    labels:
6      name: wildfly-pod
7  spec:
8    containers:
9      - name: wildfly
10     image: jboss/wildfly
11     ports:
12       - containerPort: 8080
```

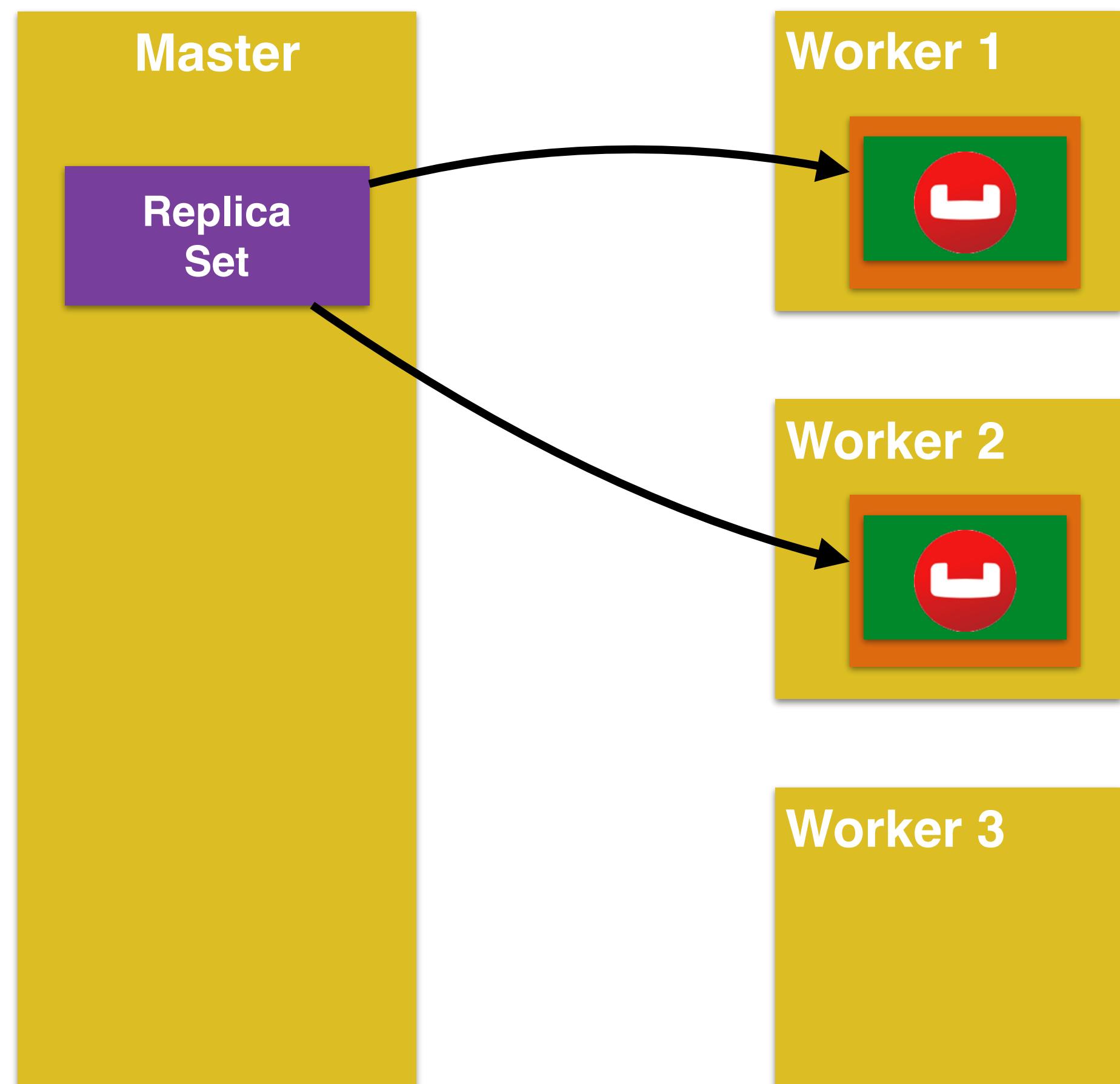
Replica Set

- Ensures that a specified number of pod "replicas" are running
 - Pod templates are cookie cutters
 - Rescheduling
 - Manual or auto-scale replicas
 - Rolling updates
- Set-based selector requirement
 - Expression: `key, operator, value`
 - Operators: `In, NotIn, Exists, DoesNotExist`
- Generally created with `Deployment`
- Only appropriate for pods with `Restart=Always` policy (default)

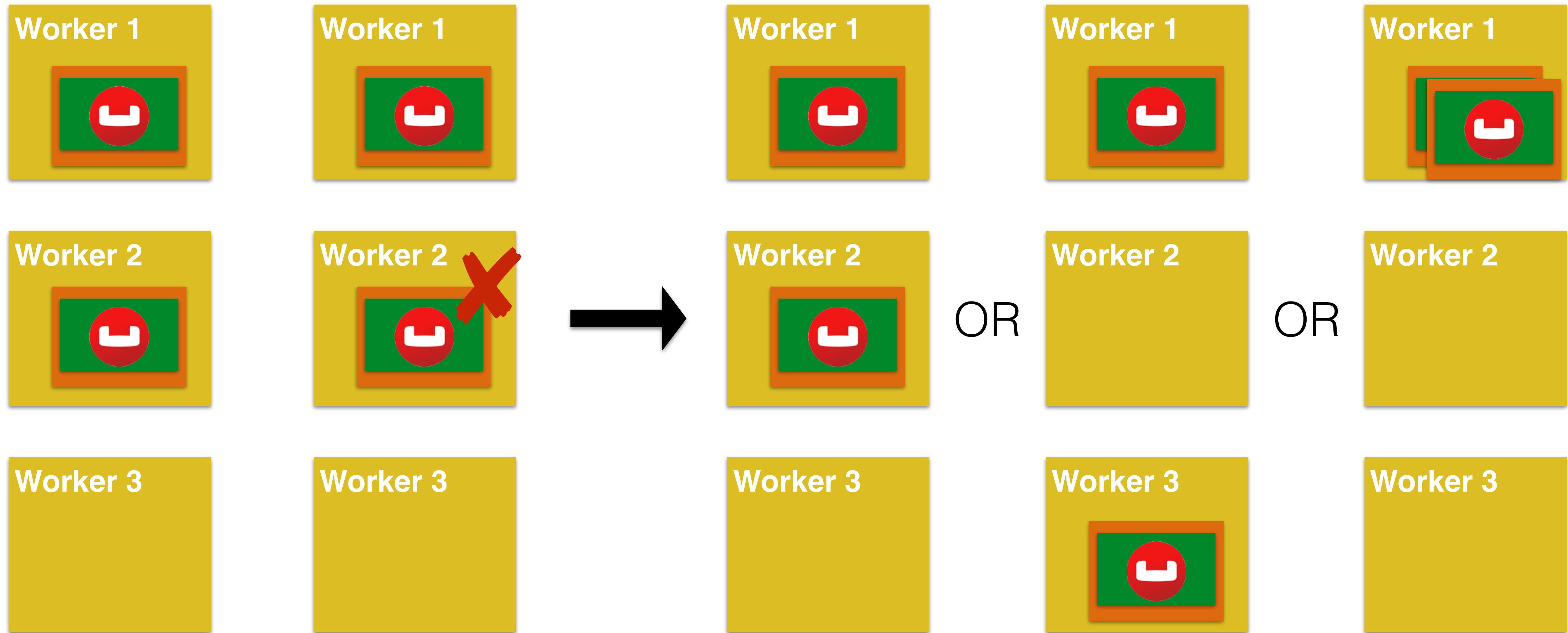
Replica Set Configuration

```
1  apiVersion: extensions/v1beta1
2  kind: ReplicaSet
3  metadata:
4    name: wildfly-rs
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: wildfly-rs-pod
10     matchExpressions:
11       - {key: tier, operator: In, values: ["backend"]}
12       - {key: environment, operator: NotIn, values: ["dev"]}
13   template:
14     metadata:
15       labels:
16         app: wildfly-rs-pod
17         tier: backend
18         environment: dev
19     spec:
20       containers:
21         - name: wildfly
22           image: jboss/wildfly
23       ports:
24         - containerPort: 8080
```

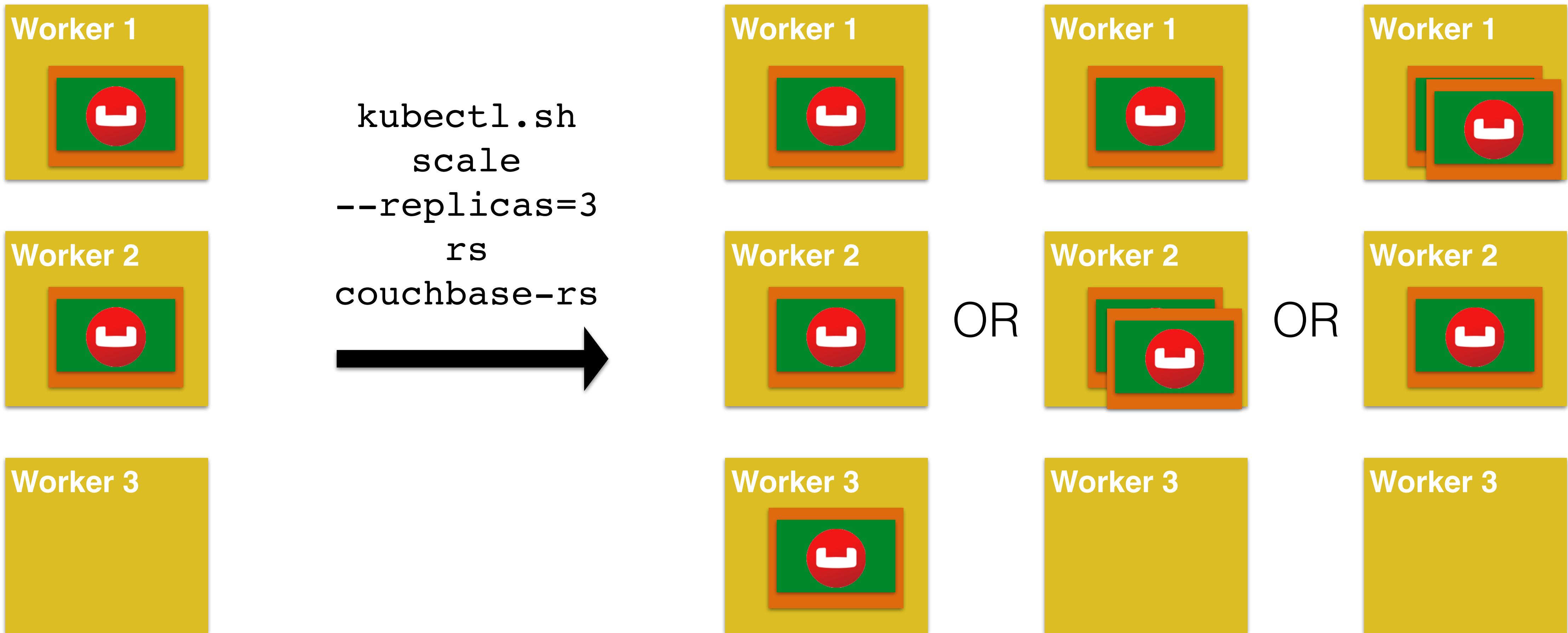
Replica Set



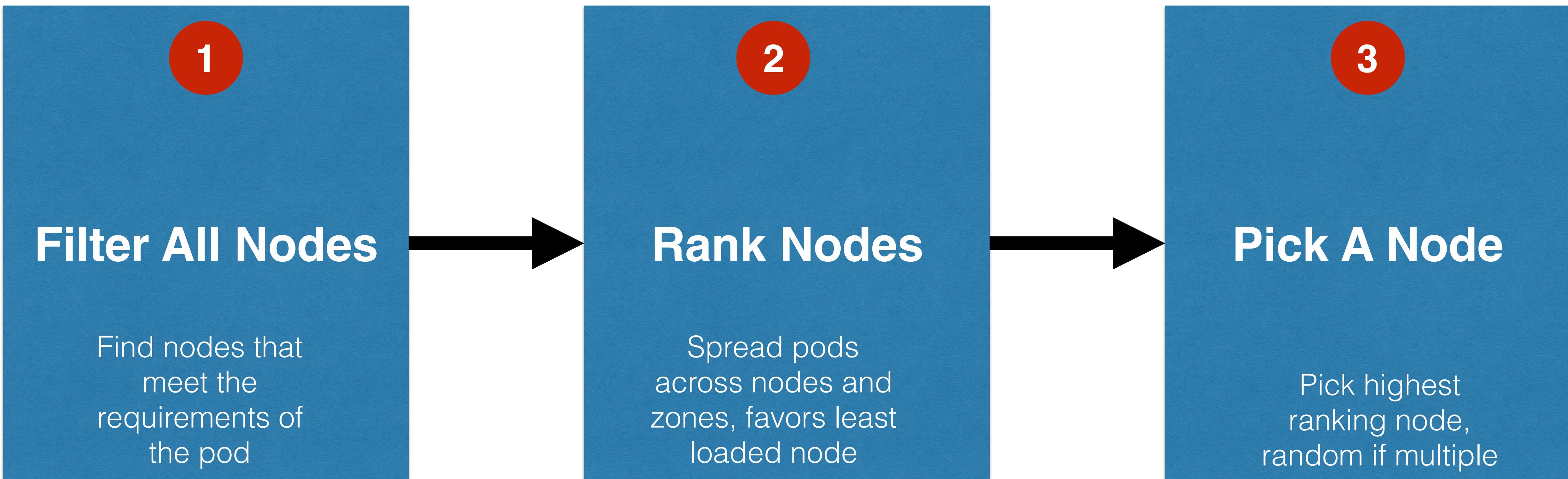
RS: “Actual” vs “Desired” State



RS: Scale Pods



Kubernetes Scheduling Algorithm



Services

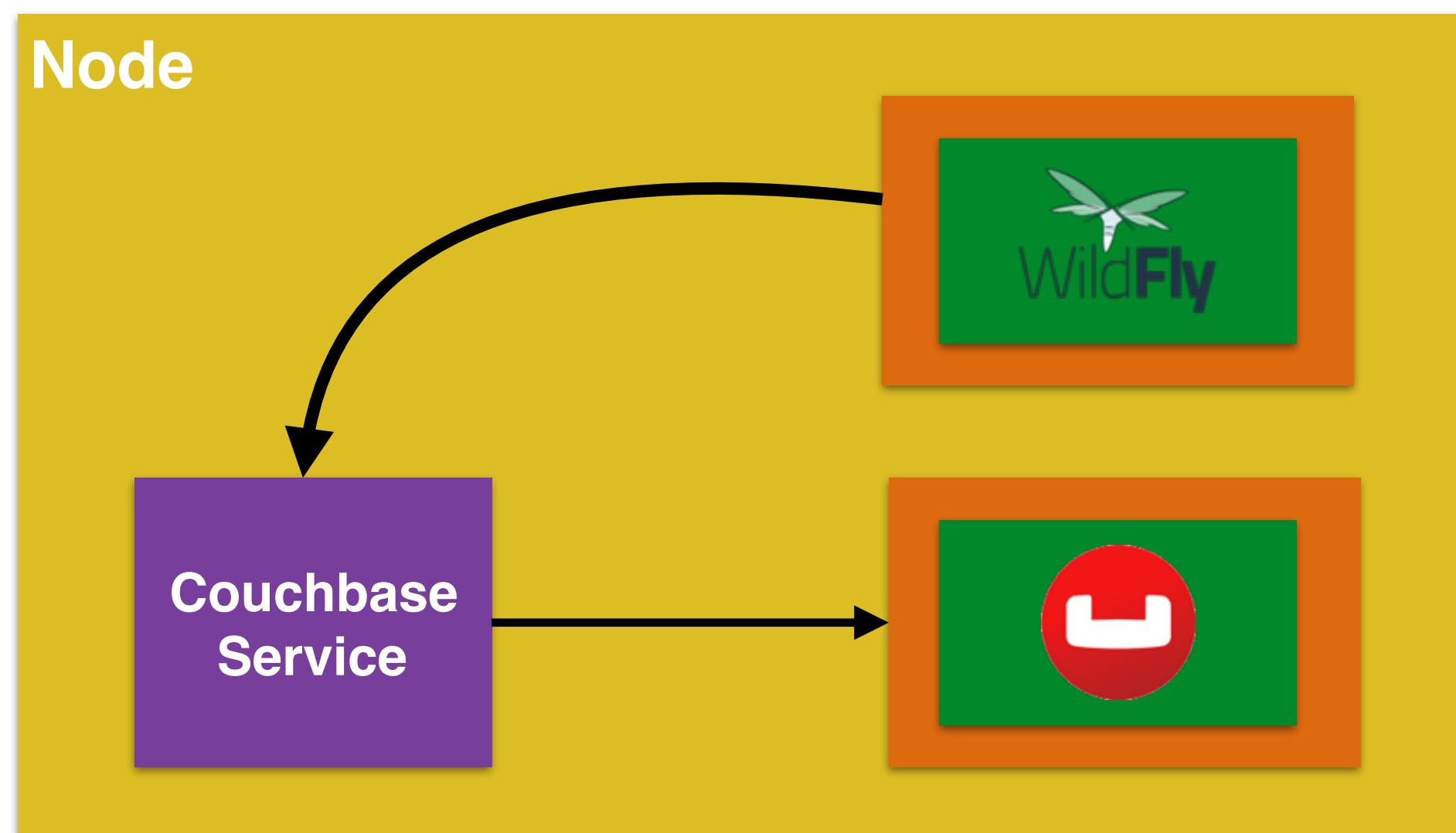
- Abstract a set of pods as a single IP and port
 - Simple TCP/UDP load balancing
- Creates environment variables in other pods or DNS resolution
- Stable endpoint for pods to reference
 - Allows list of pods to change dynamically

Kubernetes Service Configuration

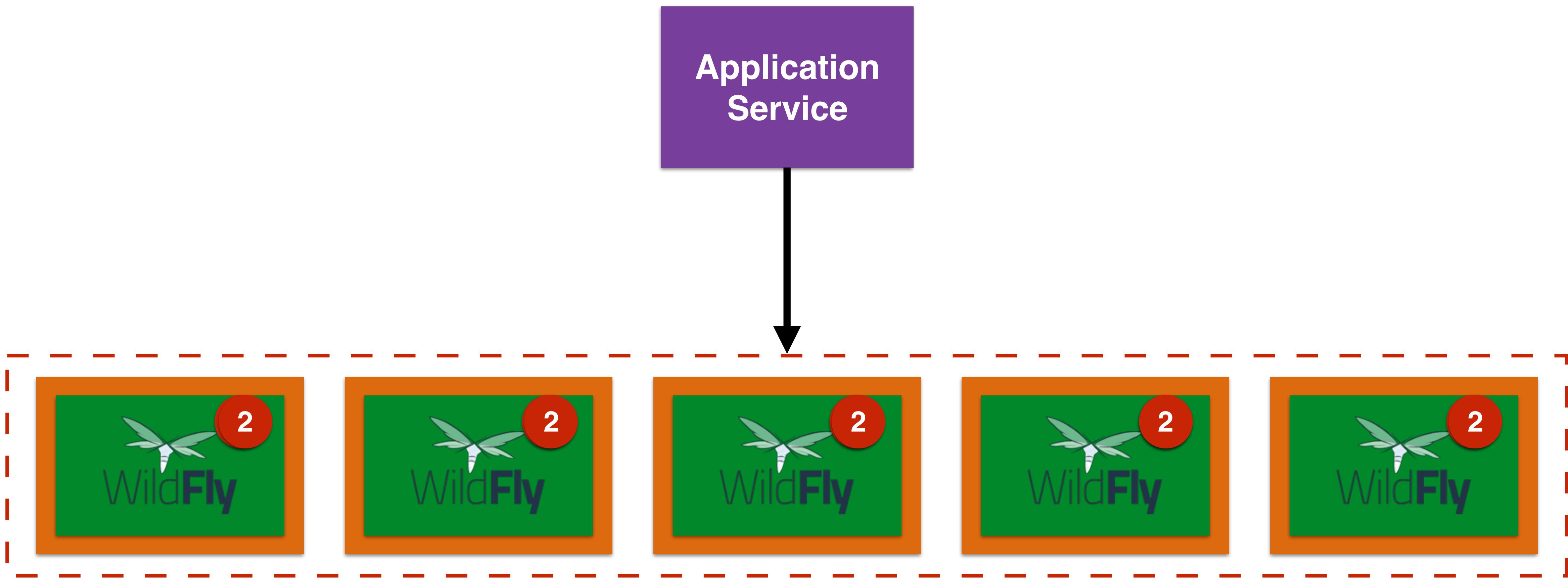
```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: couchbase-service
5  spec:
6    selector:
7      app: couchbase-rc-pod
8    ports:
9      - name: admin
10     port: 8091
11      - name: query
12        port: 8093
13  ---
14
15  apiVersion: v1
16  kind: ReplicationController
17  metadata:
18    name: couchbase-rc
19  spec:
20    replicas: 2
21    selector:
22      app: couchbase-rc-pod
23    template:
24      metadata:
25        labels:
26          app: couchbase-rc-pod
27
28        containers:
29          - name: couchbase
30            image: couchbase
31            ports:
32              - containerPort: 8091
```

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: couchbase-service
5  spec:
6    selector:
7      app: couchbase-rc-pod
8    ports:
9      - name: admin
10     port: 8091
11      - name: query
12        port: 8093
13  ---
14
15  apiVersion: v1
16  kind: ReplicationController
17  metadata:
18    name: couchbase-rc
19  spec:
20    replicas: 2
21    selector:
22      app: couchbase-rc-pod
23    template:
24      metadata:
25        labels:
26          app: couchbase-rc-pod
27
28        containers:
29          - name: couchbase
30            image: couchbase
31            ports:
32              - containerPort: 8091
```

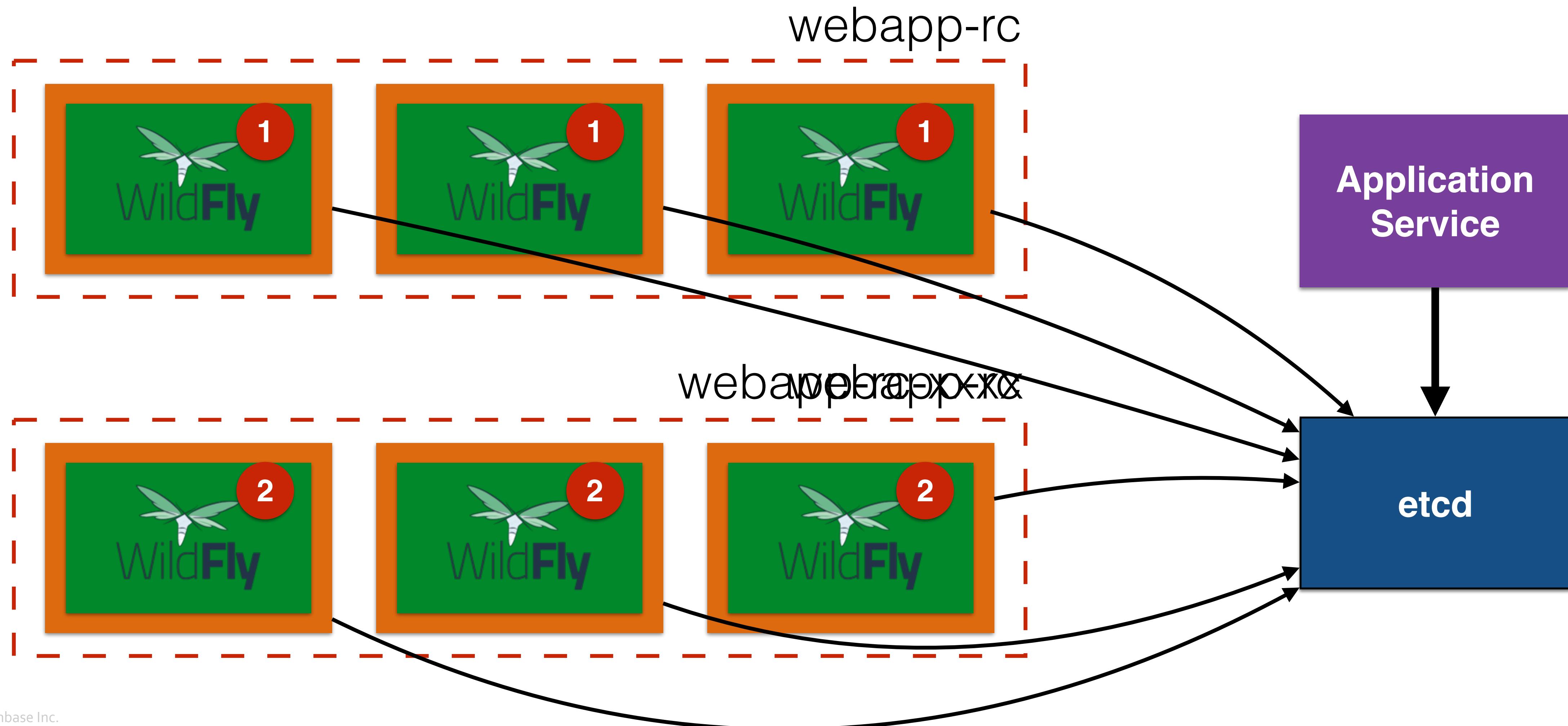
Couchbase Service



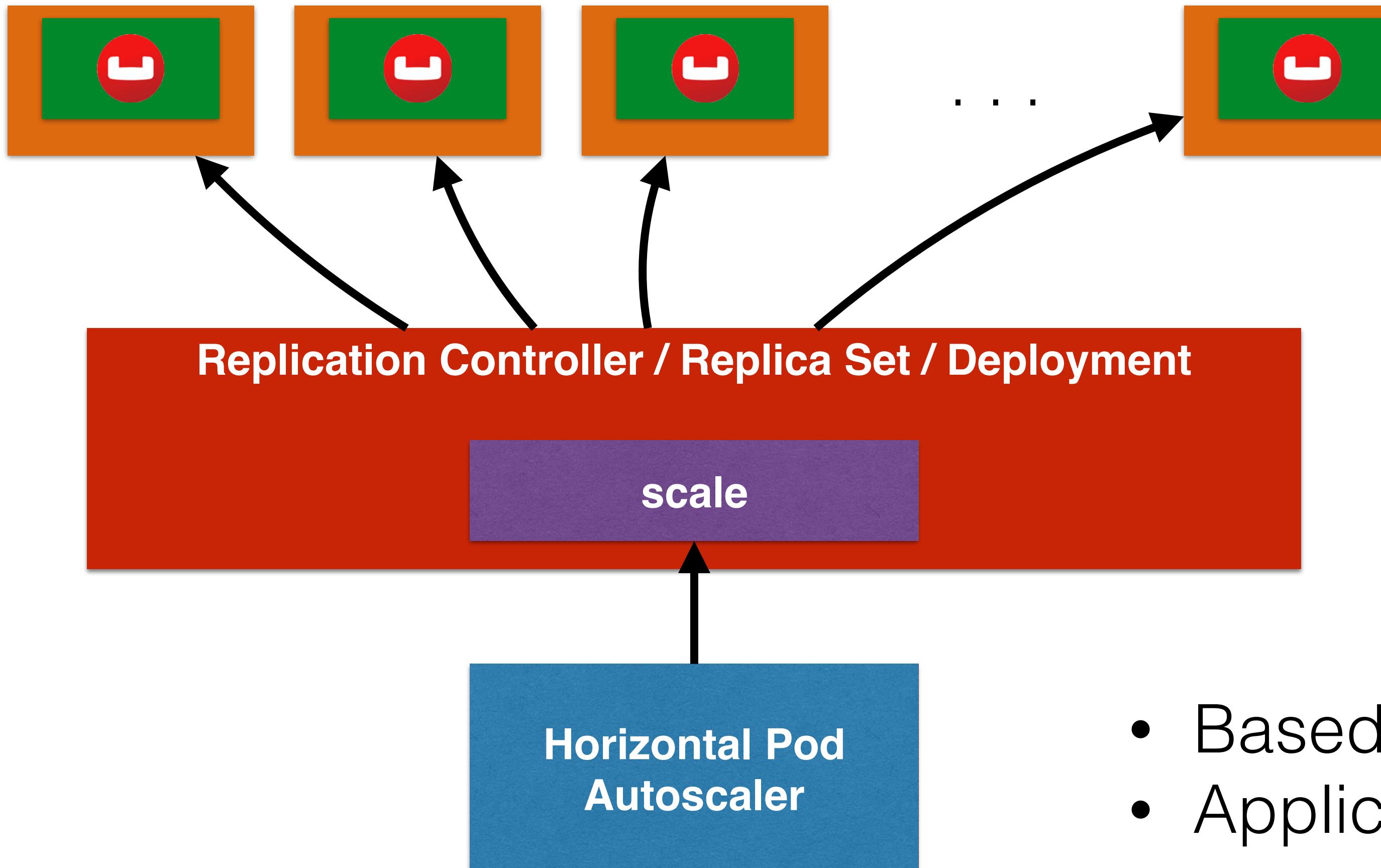
Rolling Updates



Rolling Updates



Horizontal Pod Autoscaling



- Based on observed CPU utilization
- Application provided metrics (health)

HPA Configuration

```
1 apiVersion: autoscaling/v1
2 kind: HorizontalPodAutoscaler
3 metadata:
4   name: wildfly-scaler
5 spec:
6   scaleTargetRef:
7     kind: ReplicaSet
8     name: wildfly-rs
9   minReplicas: 3
10  maxReplicas: 10
11  targetCPUUtilizationPercentage: 50
```

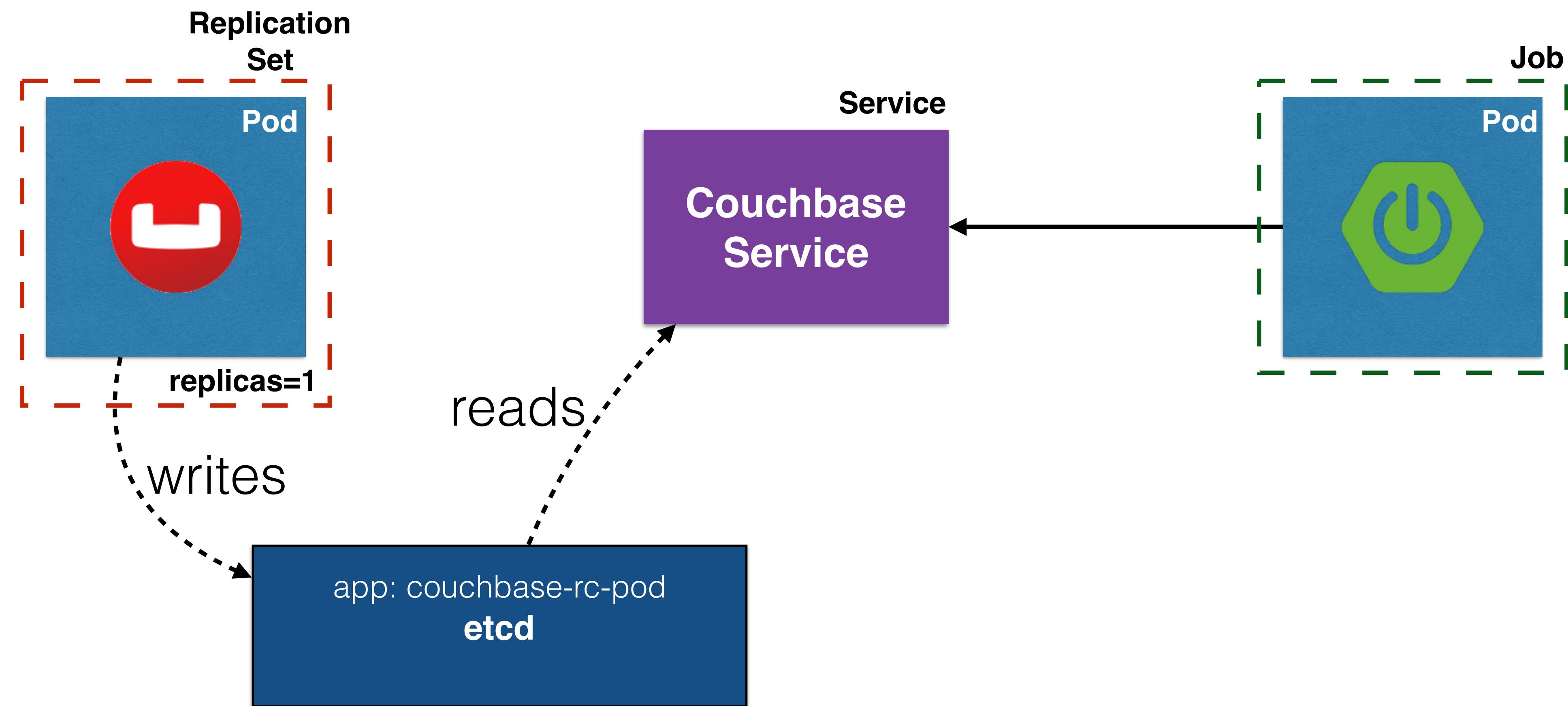
Batch Jobs

- Run-once jobs
 - Replication Controller, Replica Set or Deployments not suitable
- Three types
 - Non-parallel: only one pod is started
 - Parallel:
 - With a fixed completion count
 - With a work queue

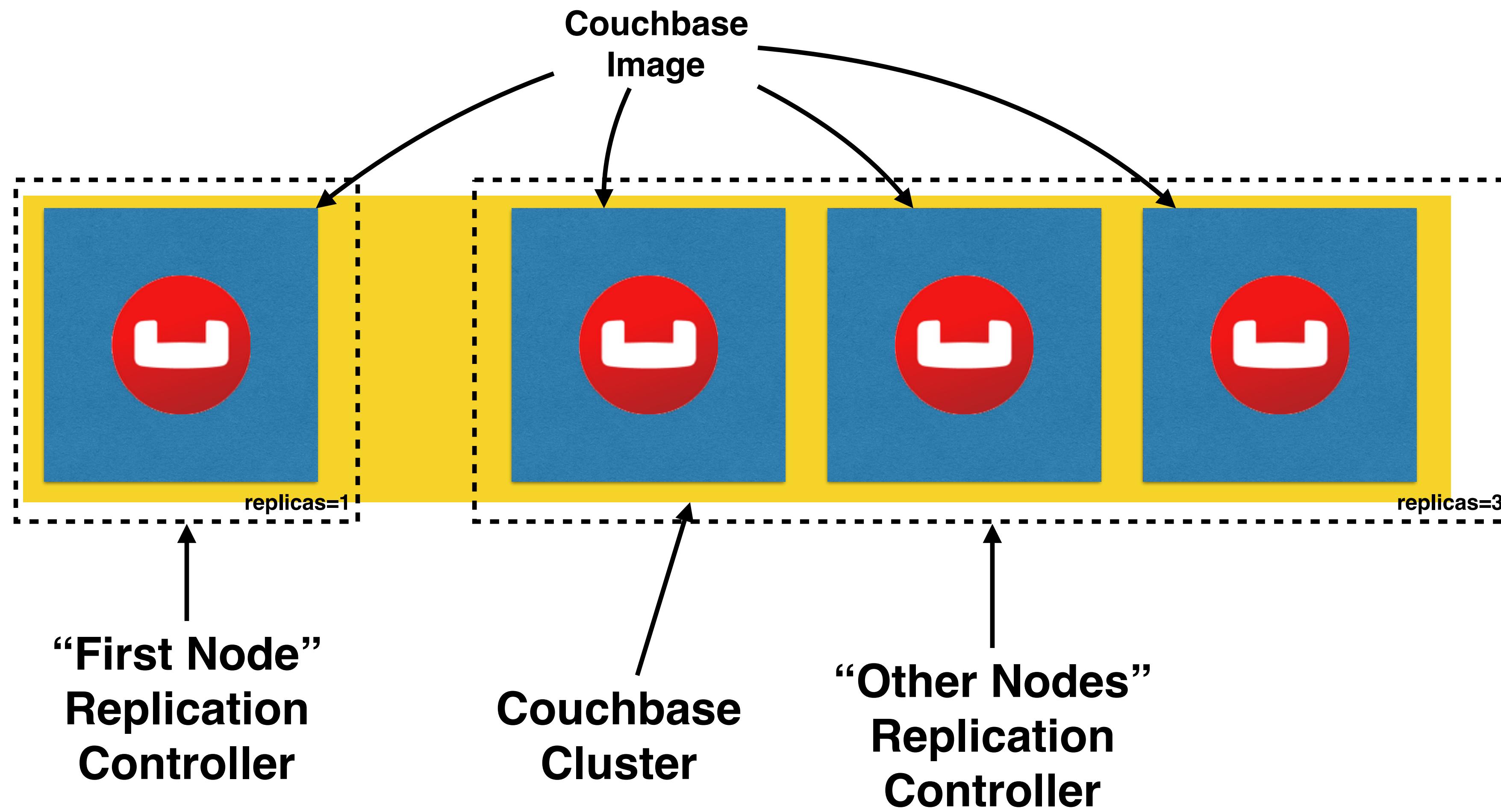
Run-Once Job Specification

```
1  apiVersion: batch/v1
2  kind: Job
3  metadata:
4    name: wait
5  spec:
6    template:
7      metadata:
8        name: wait
9      spec:
10        containers:
11          - name: wait
12            image: ubuntu
13            command: ["sleep", "20"]
14            restartPolicy: Never
```

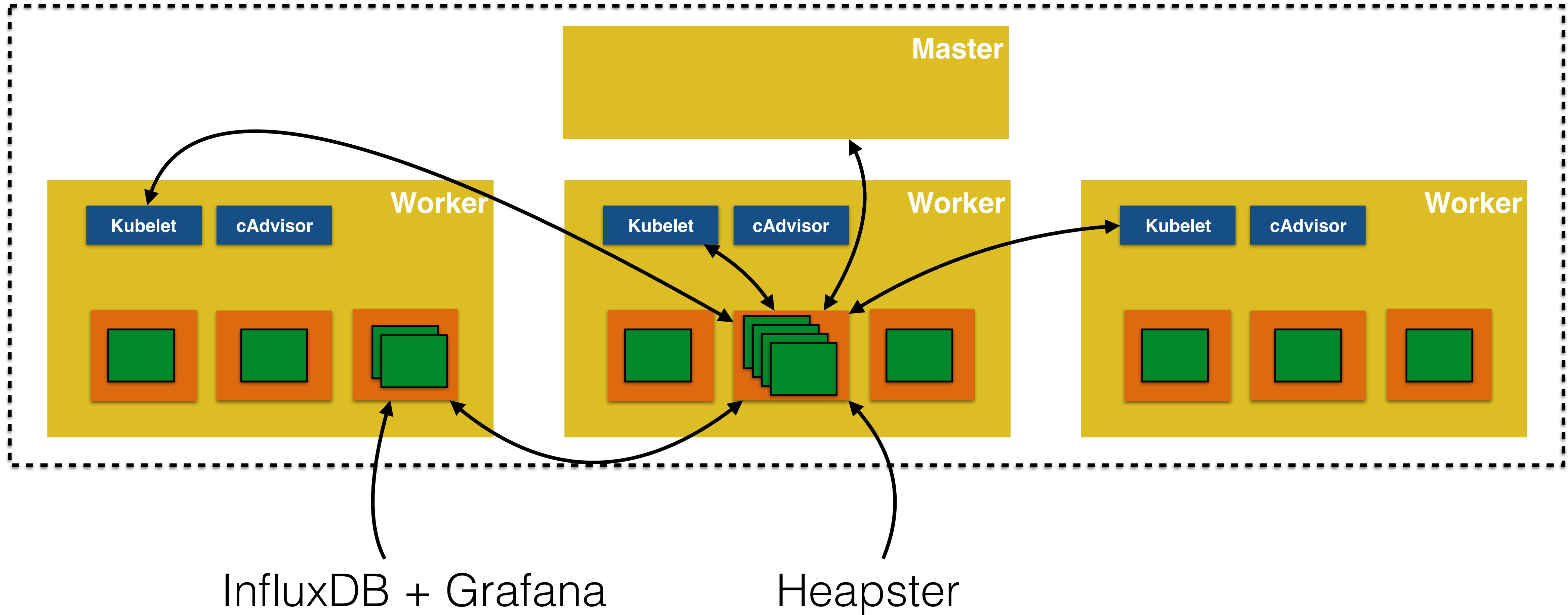
Java Application in Kubernetes

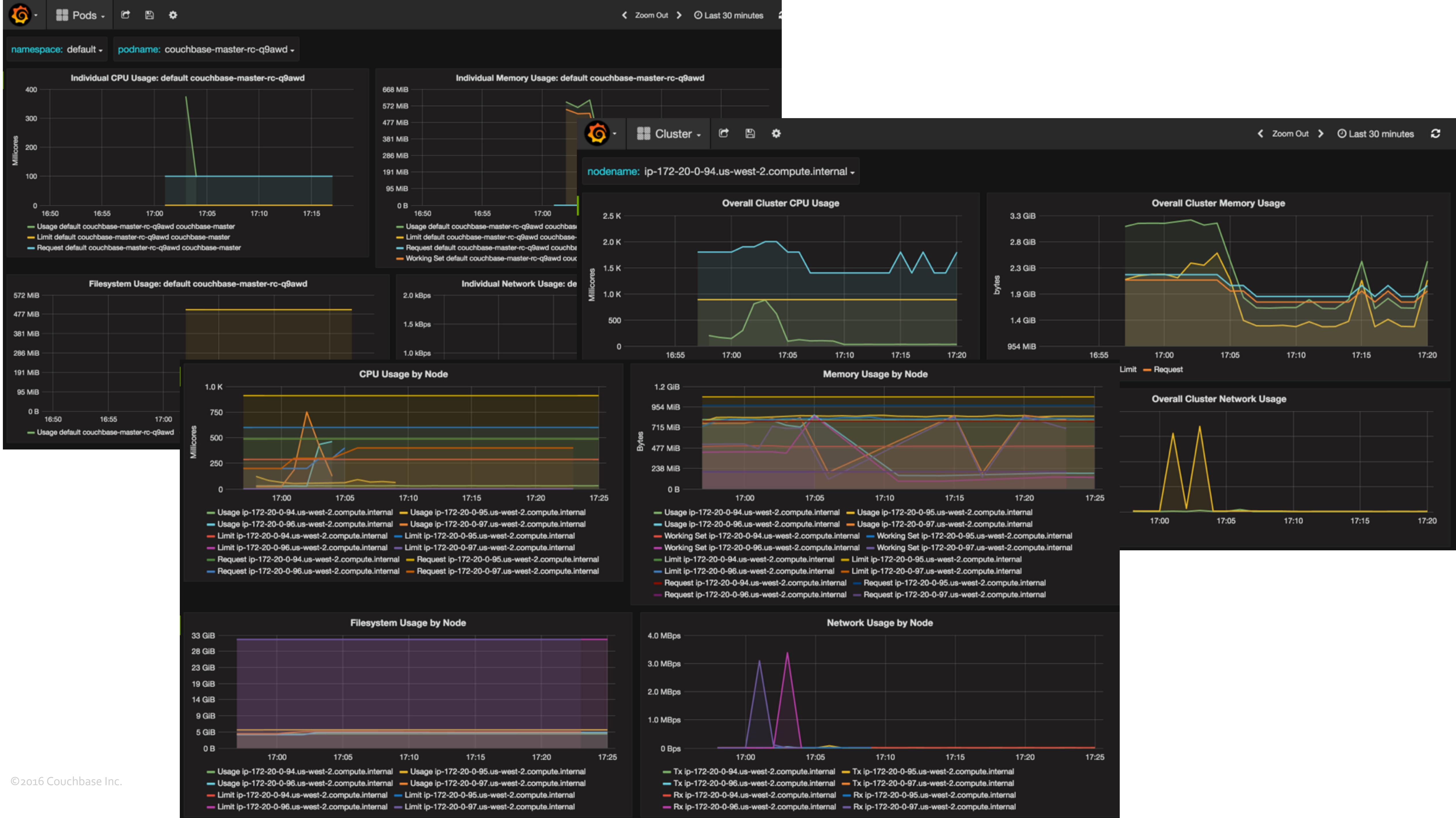


Couchbase Cluster on Kubernetes



Monitoring Kubernetes Resources





References

- Slides: github.com/javaee-samples/docker-java/tree/master/slides
- Docker Lab: github.com/docker/labs/tree/master/java
- Kubernetes Lab: github.com/arun-gupta/kubernetes-java-sample