

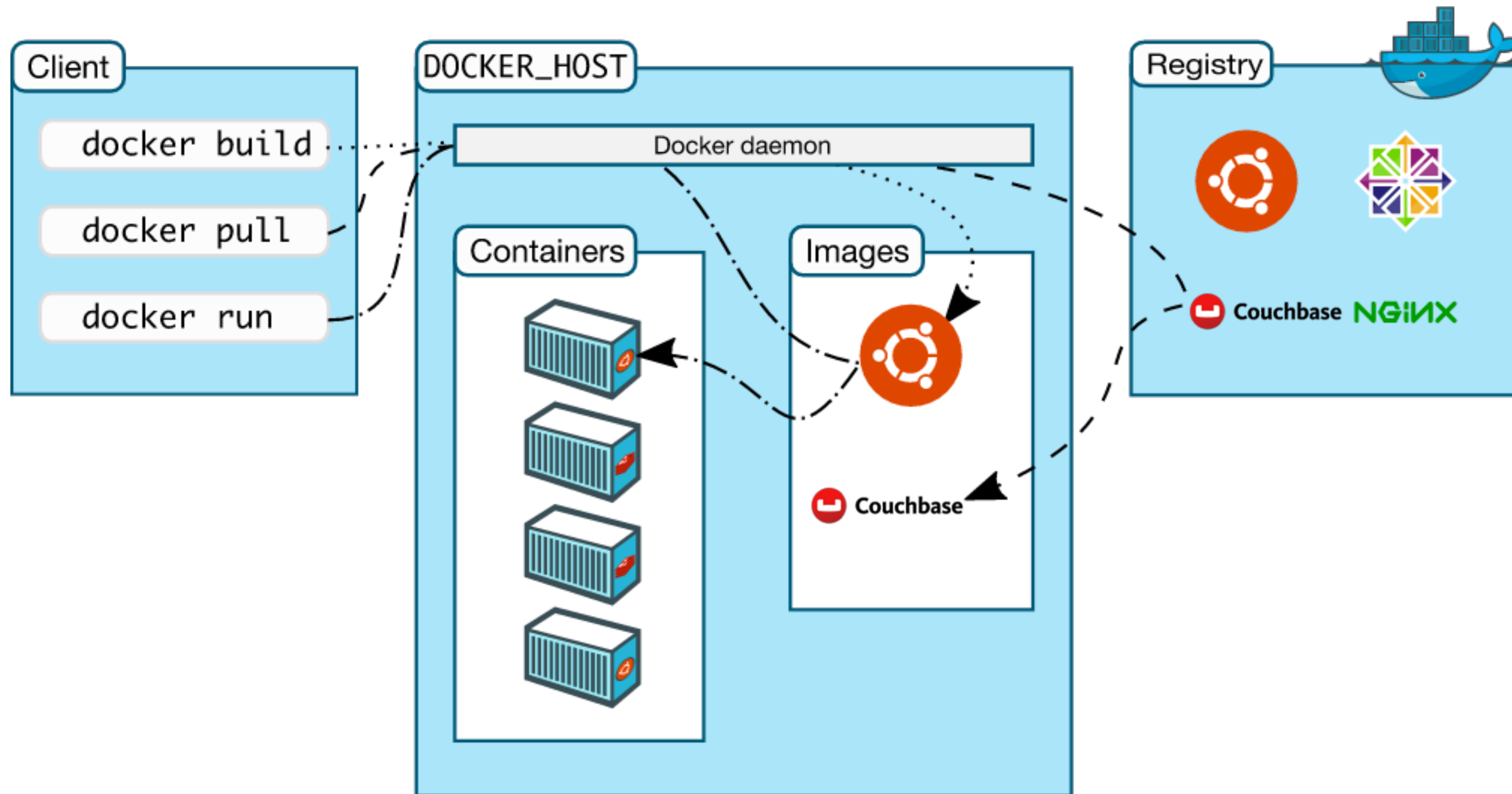
Package your Java Application using Docker and Kubernetes

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

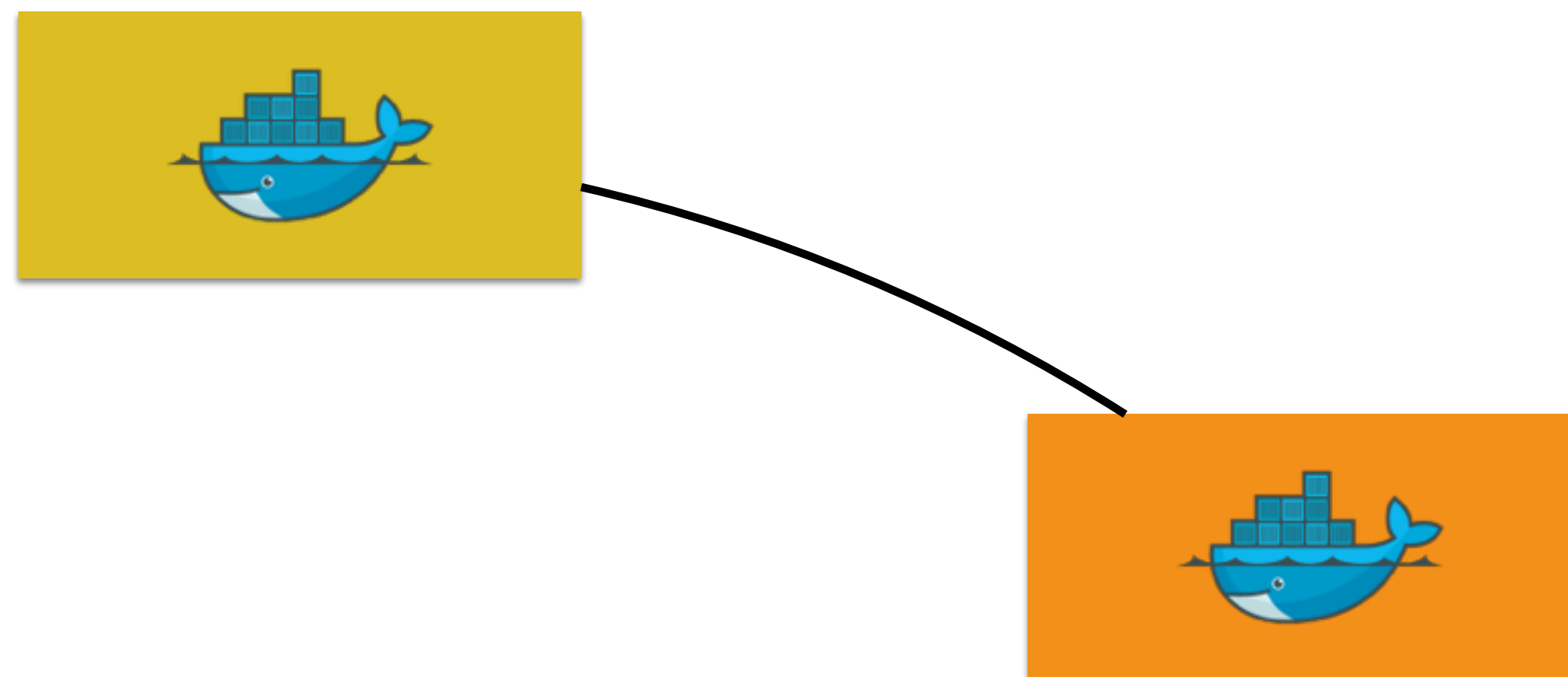
- Multiple master
- Scheduler
- Rules and constraints
- Monitoring
- Rolling Update
- Cloud/commercial support

Core Concepts: Docker Manager



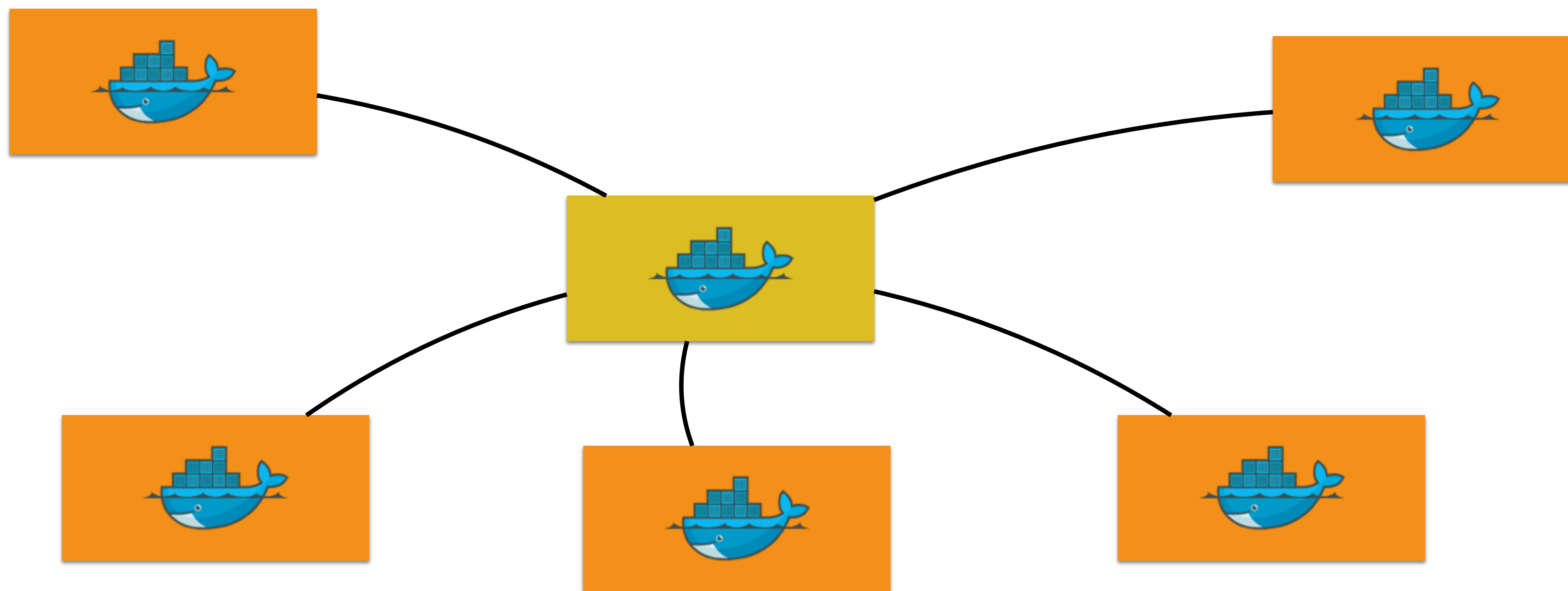
```
docker swarm init --listen-addr <ip>:2377
```

Core Concepts: Docker Worker



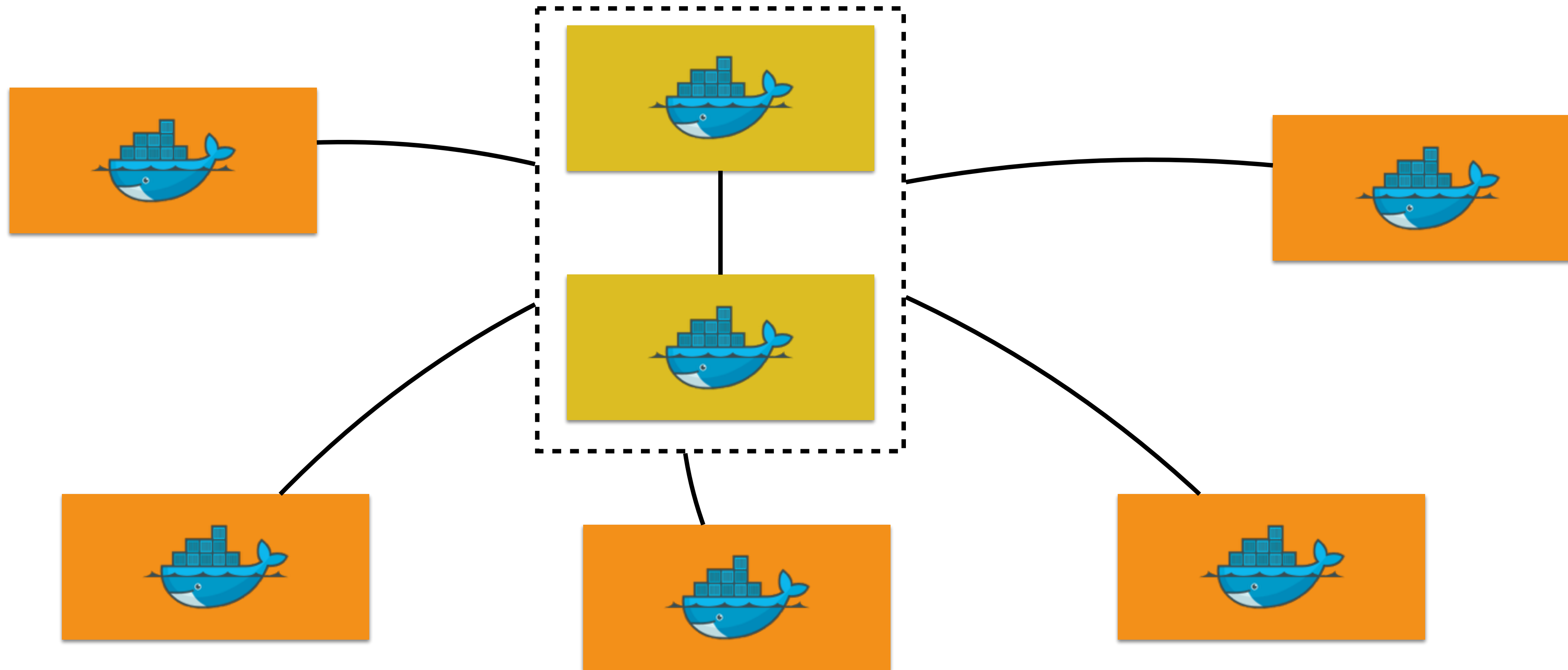
```
docker swarm join --token <worker_token> <manager>:2377
```

Core Concepts: Add More Workers



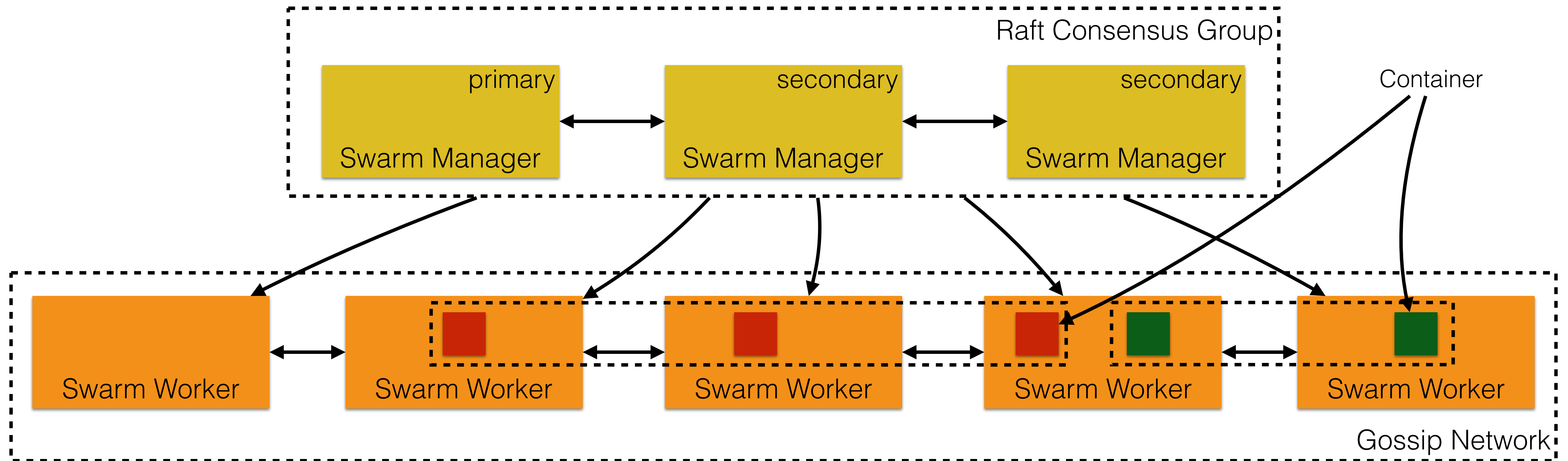
```
docker swarm join --token <worker_token> <manager>:2377
```

Core Concepts: Primary/Secondary Master



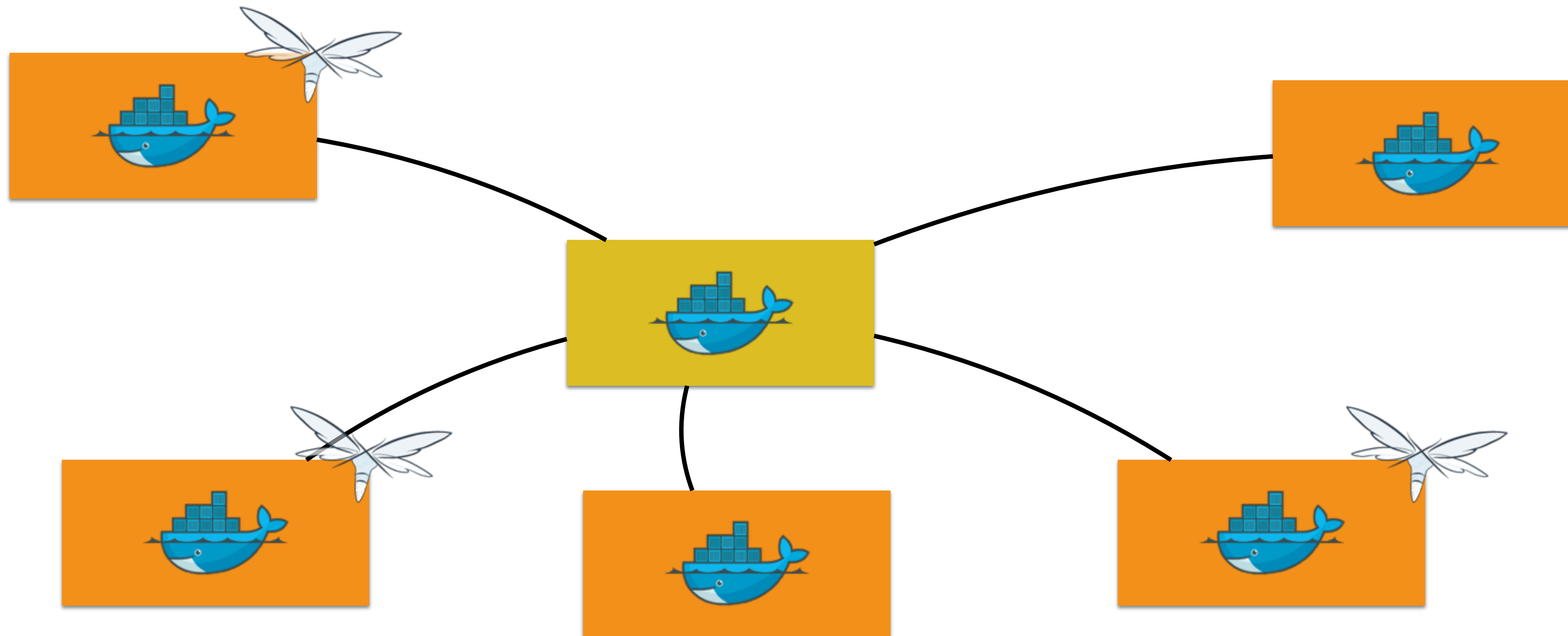
```
docker swarm join --manager --token <manager_token> --listen-addr <master2>:2377 <master1>:2377
```


Core Concepts: Cluster



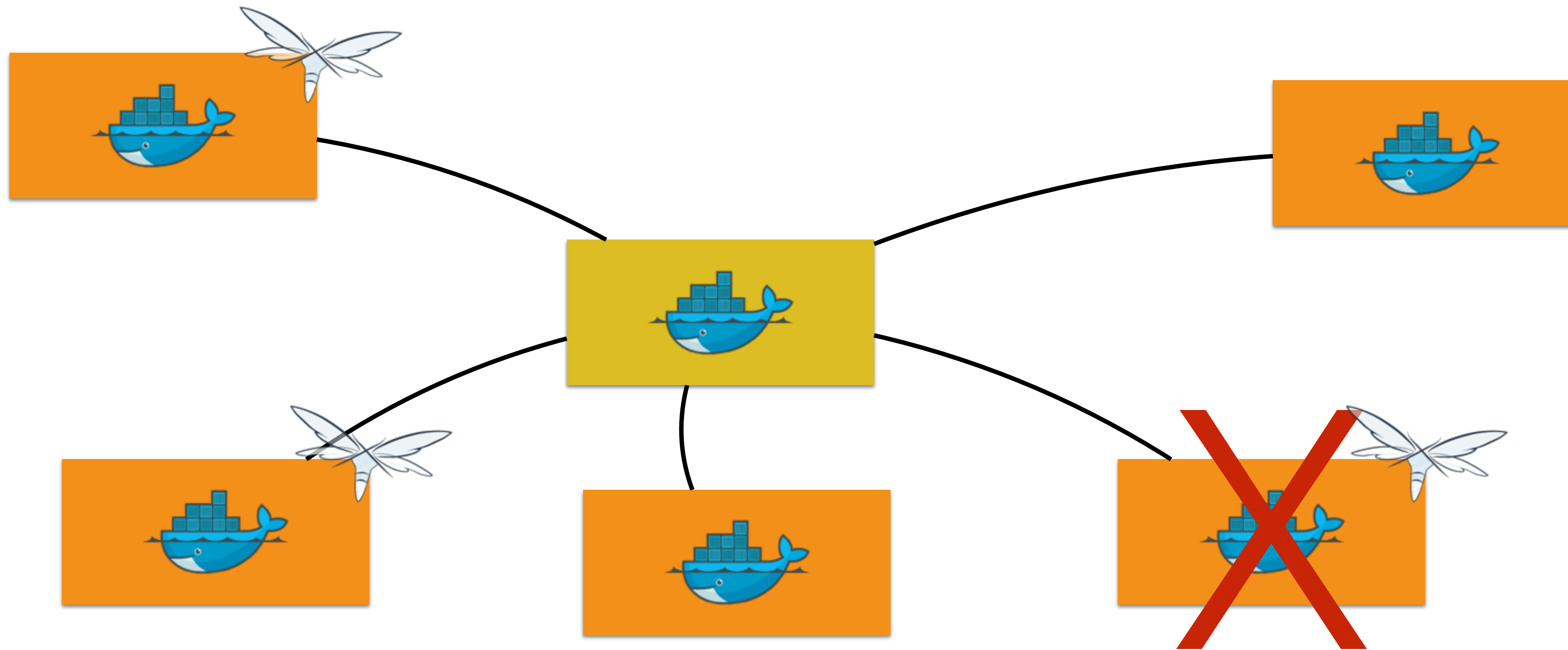
Strongly consistent
Replicated (Raft based)
Extremely fast (in-memory reads)

Core Concepts: Replicated Service

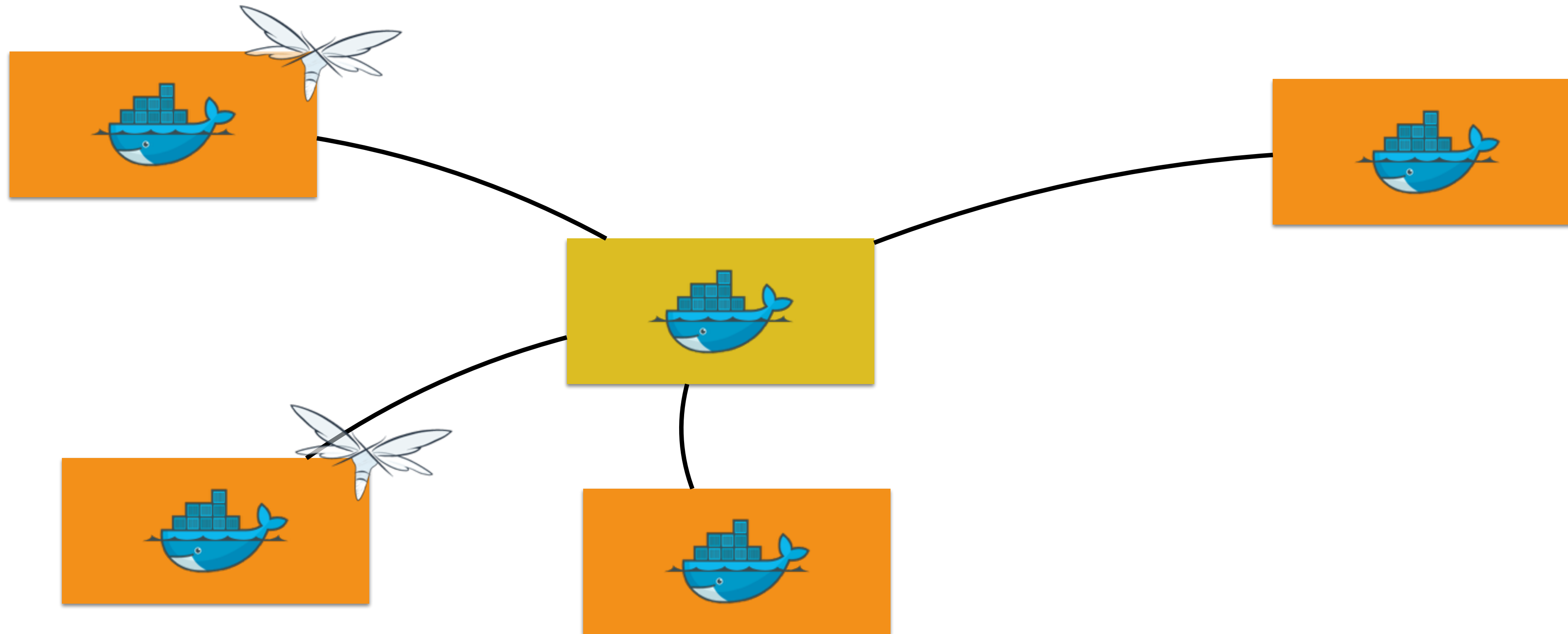


```
docker service create --replicas 3 --name web jboss/wildfly
```

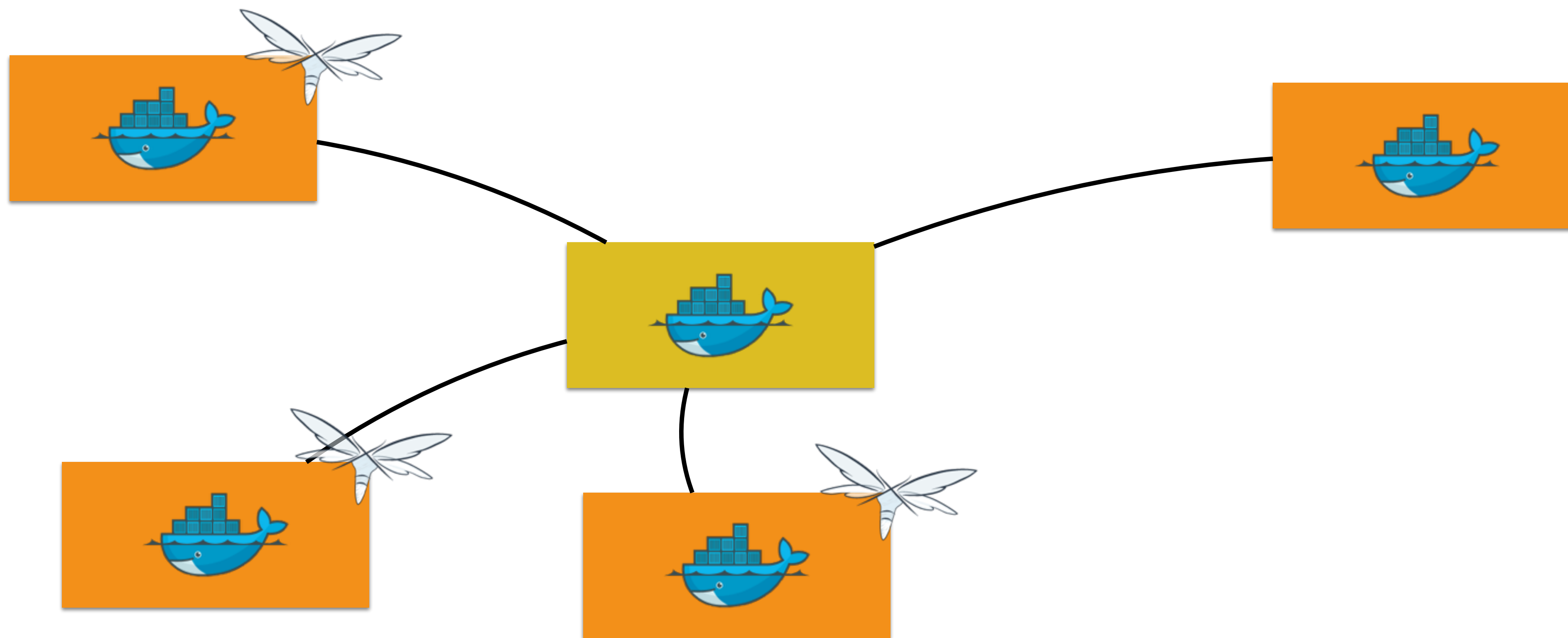
Core Concepts: Node Failure



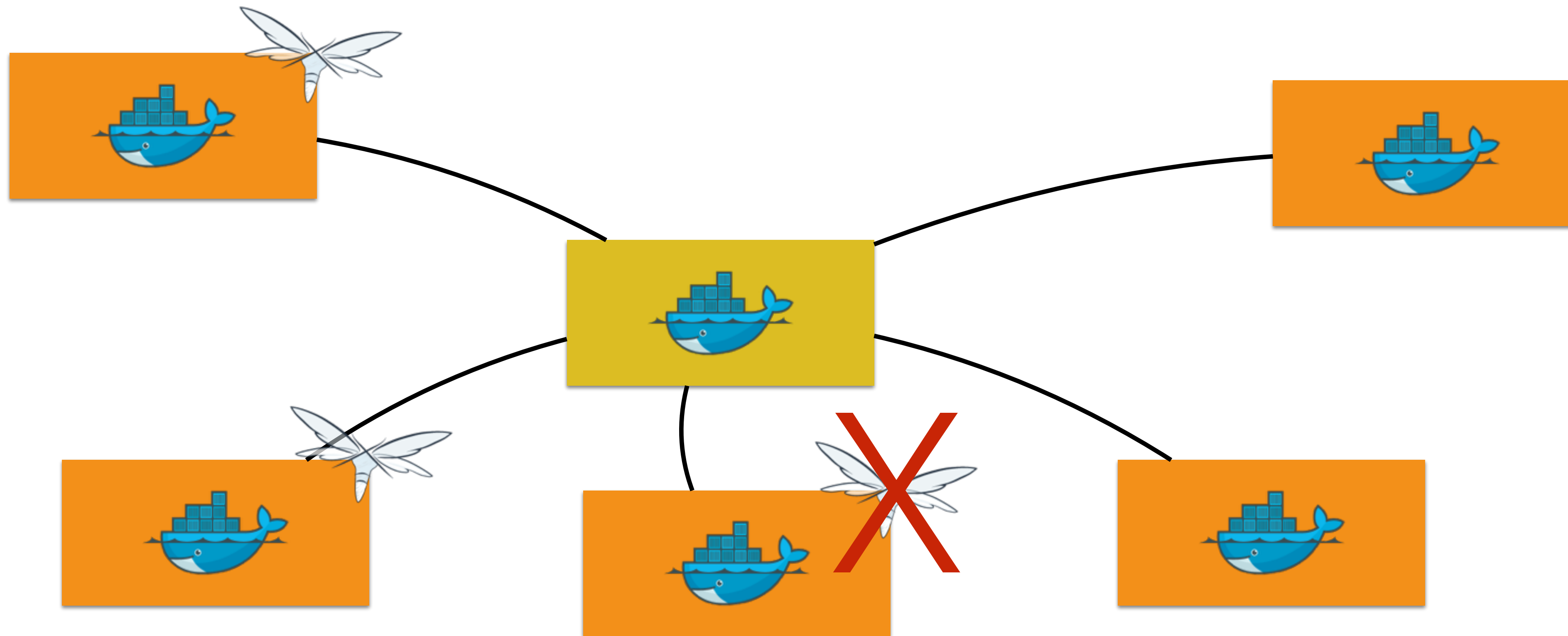
Core Concepts: Desired \neq Actual



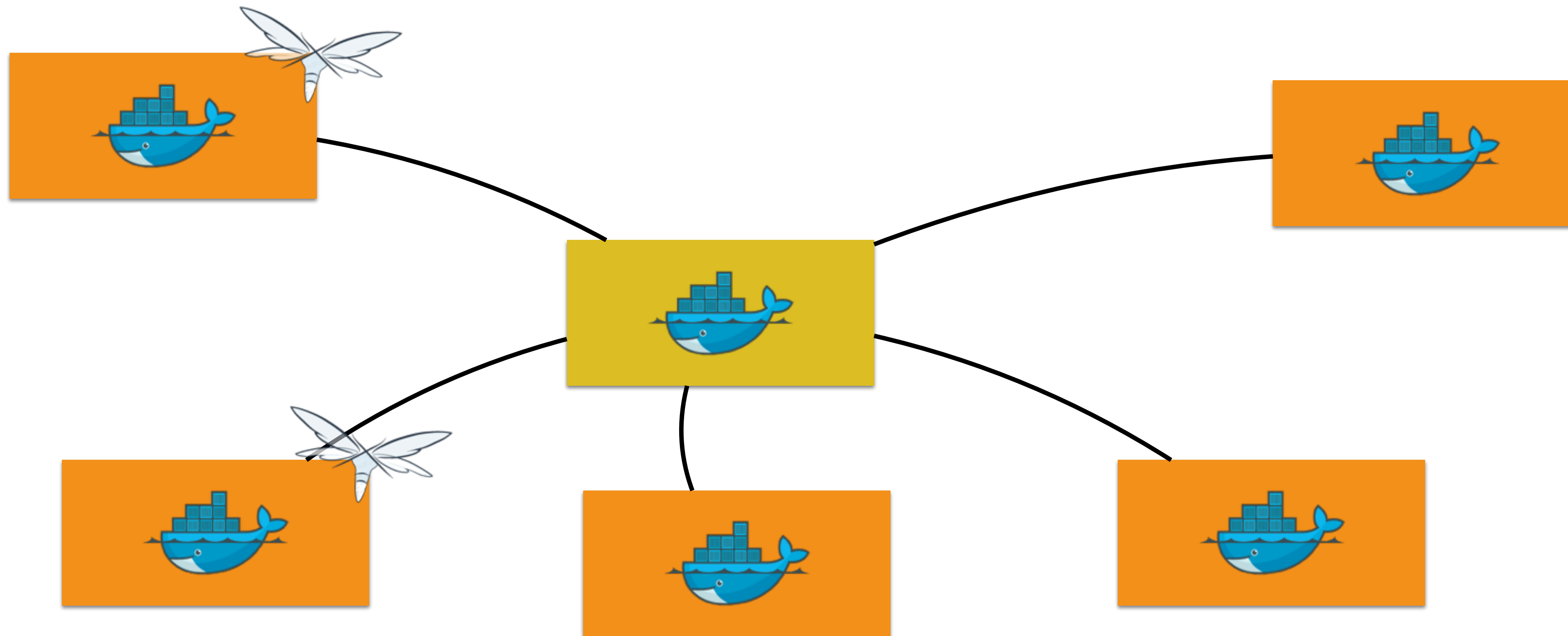
Core Concepts: Reconcile



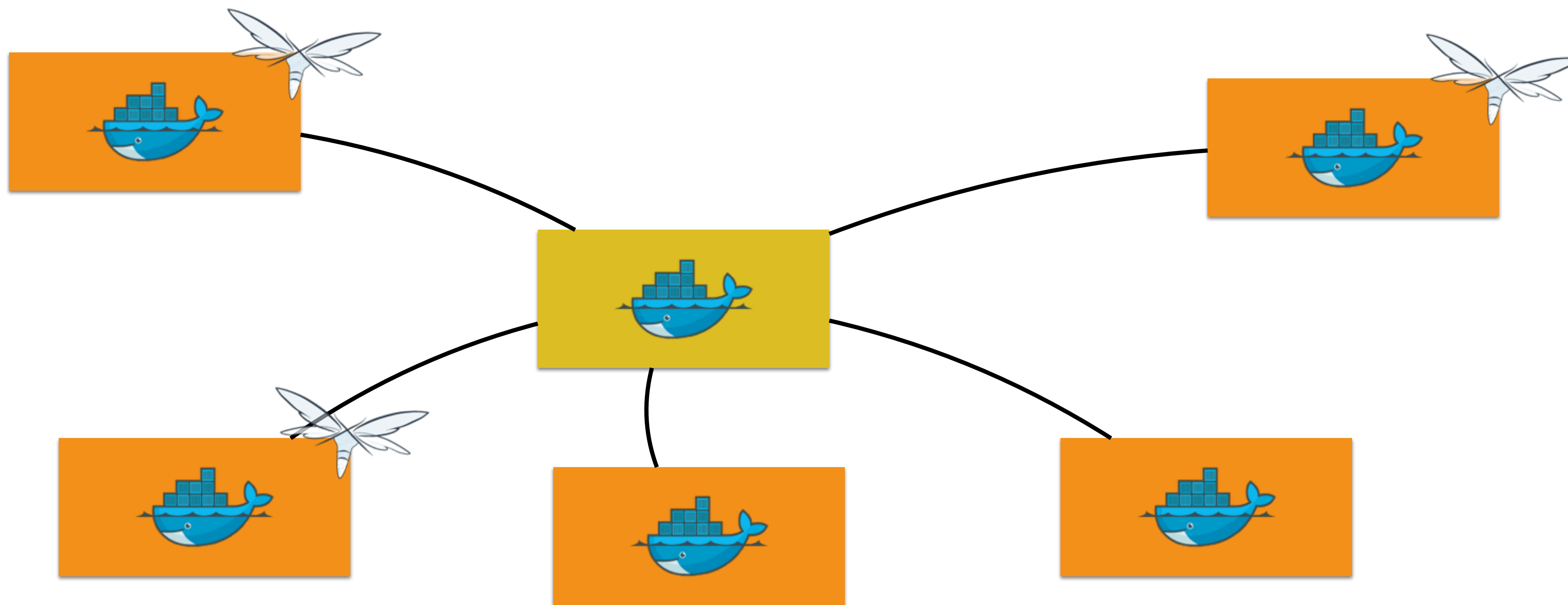
Core Concepts: Container Failure



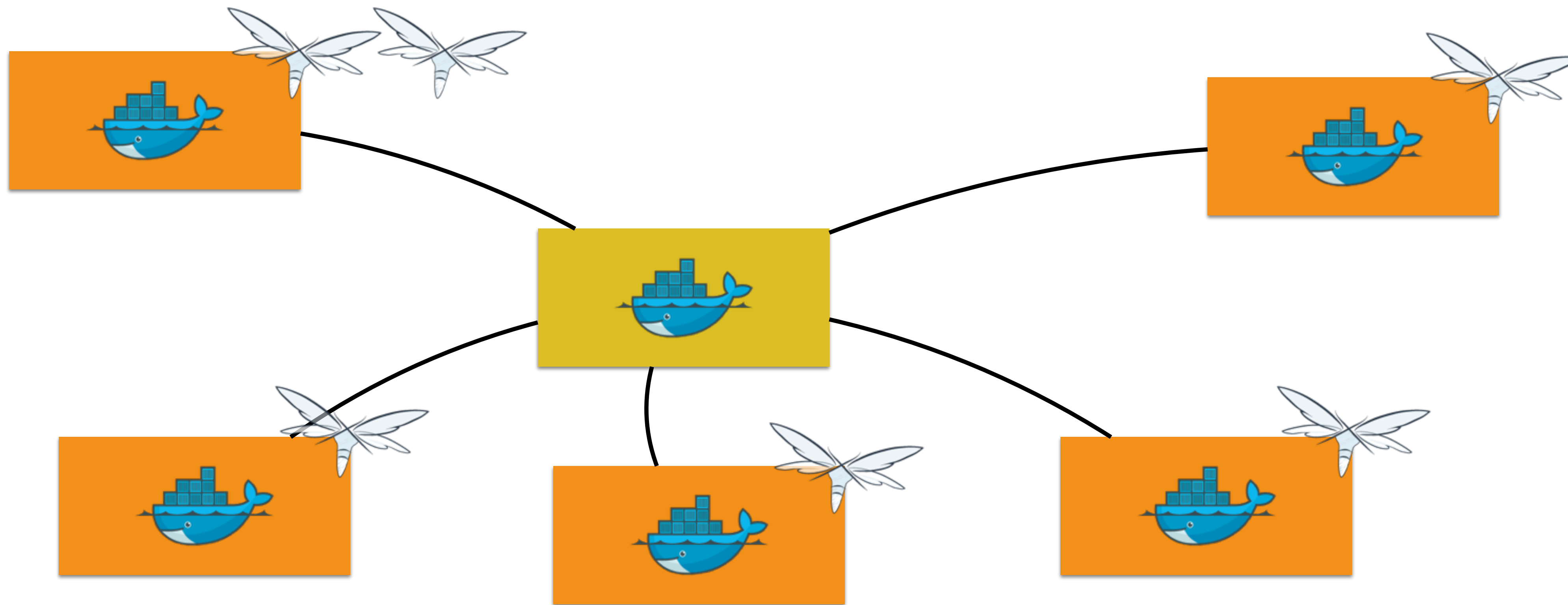
Core Concepts: Desired \neq Actual



Core Concepts: Reconcile

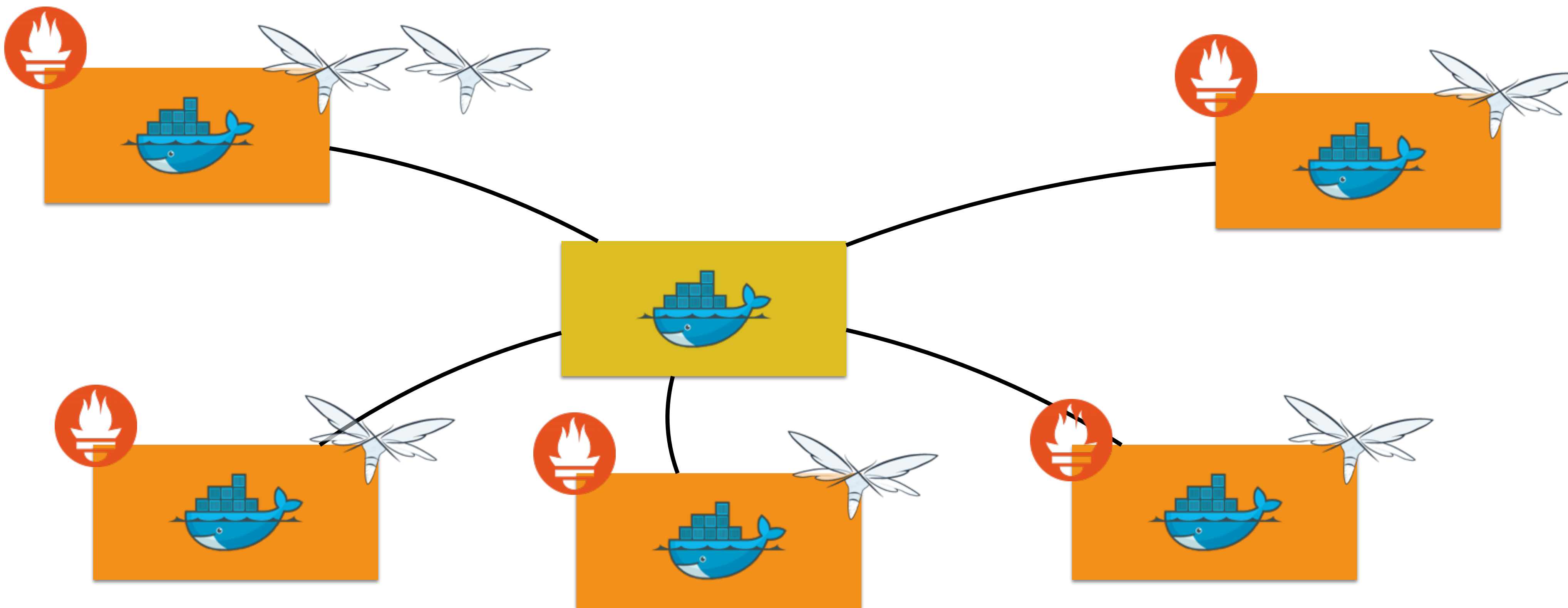


Core Concepts: Scale



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

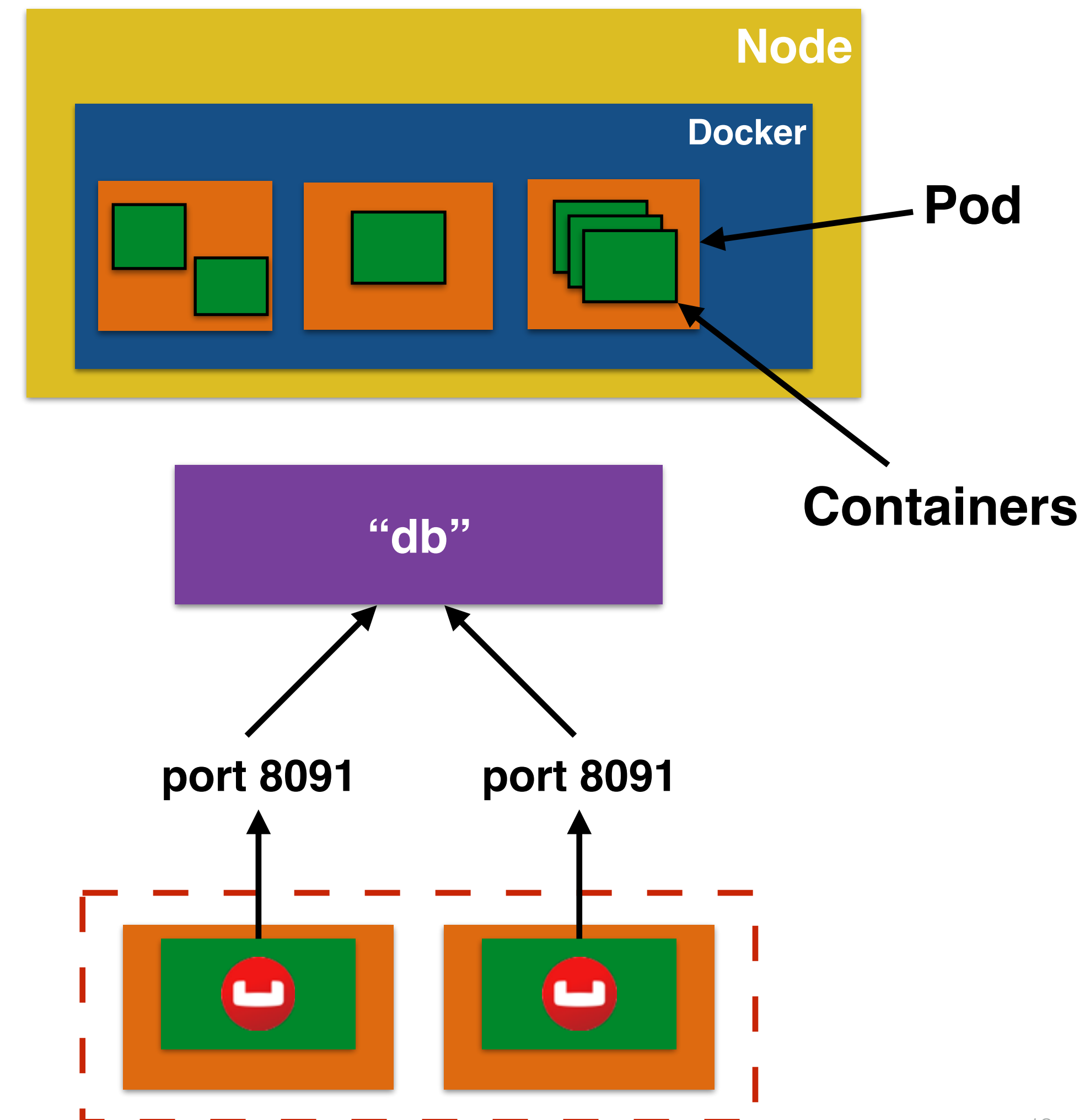
Core Concepts: Global Service



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

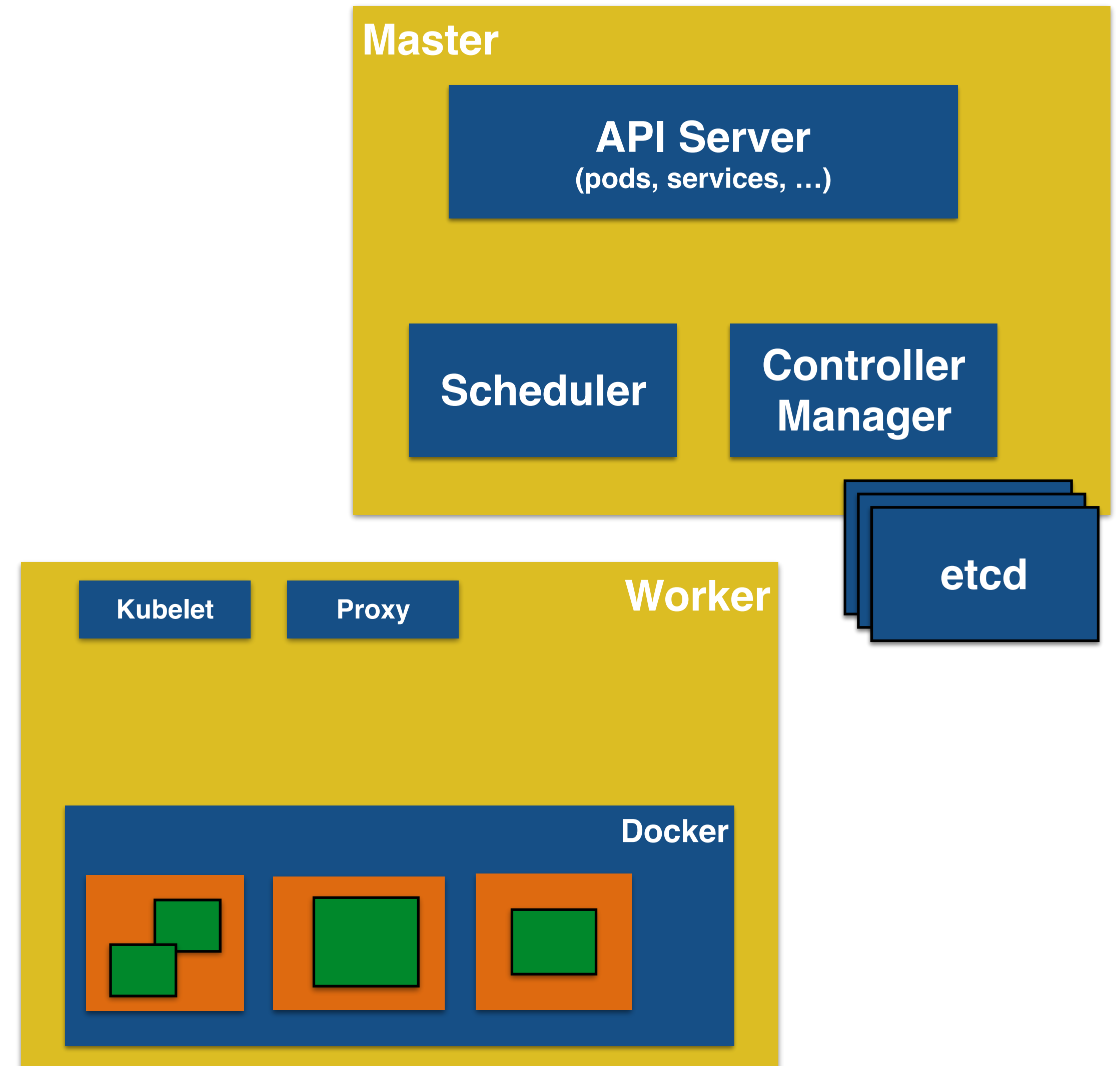
Core Concepts: Kubernetes

- **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

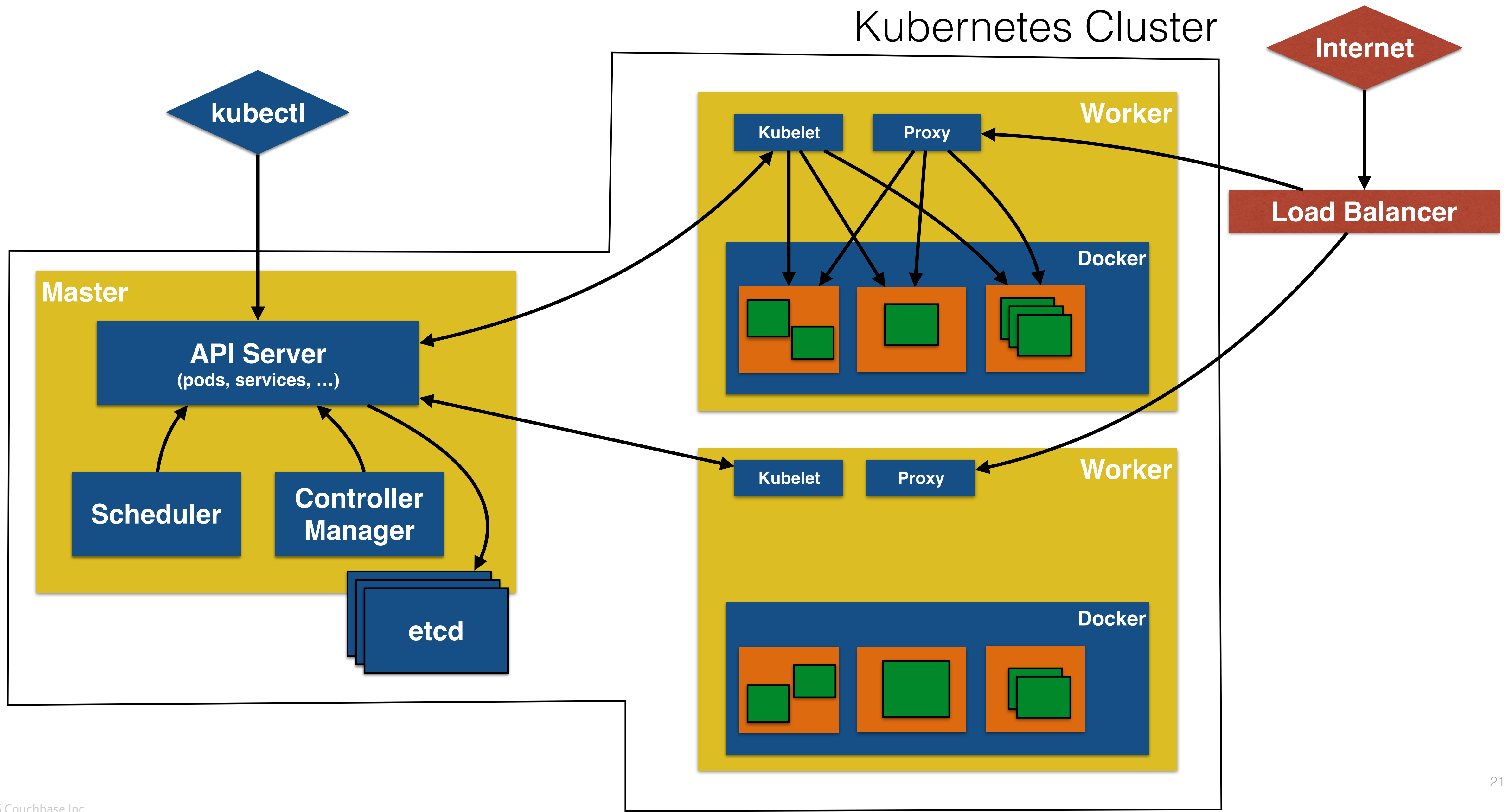


Core Concepts: Kubernetes

- **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)



Kubernetes Cluster



Service Discovery & LB: Docker



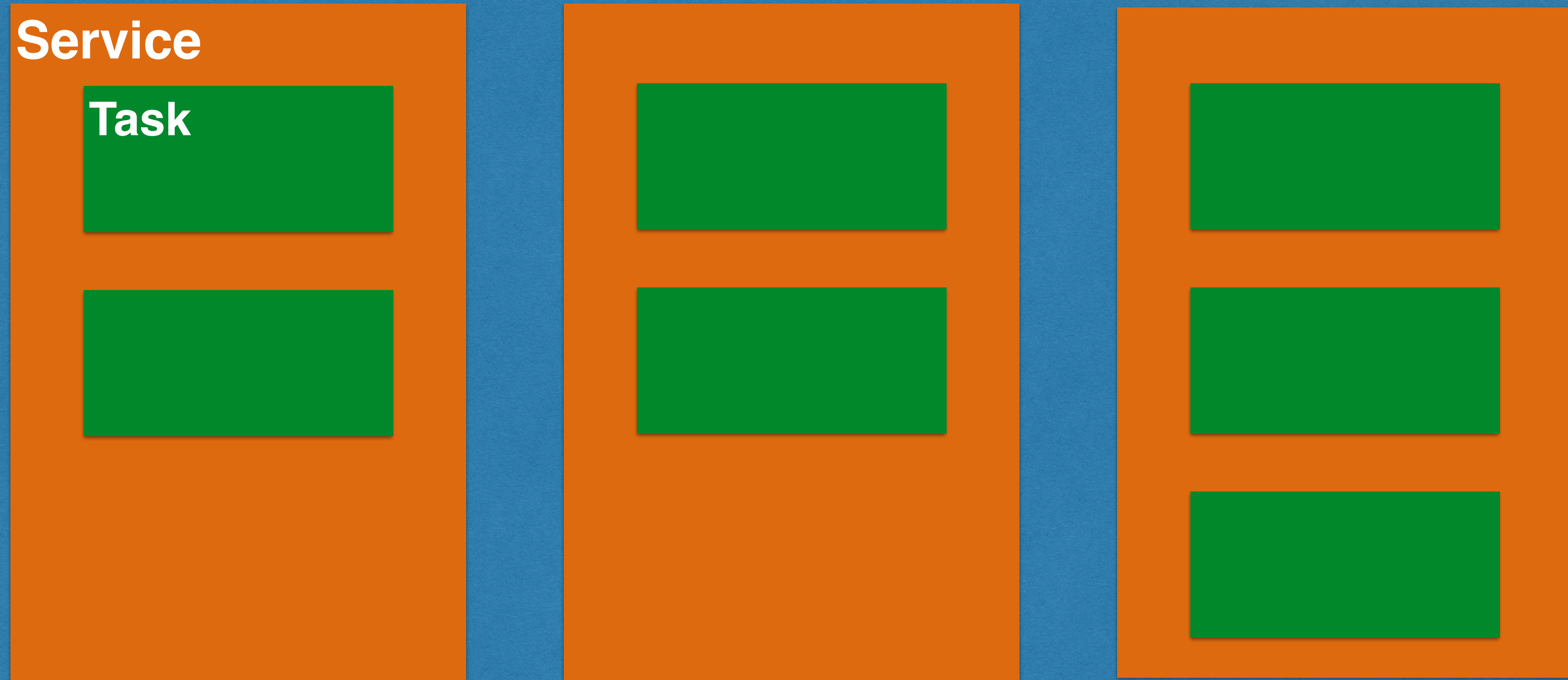
- Docker Compose
 - Define and run multi-container applications
 - Configuration defined in one or more files
 - `docker-compose.yml` (default)
 - `docker-compose.override.yml` (default)
 - Multiple files specified using `-f`
 - Deployed as Docker Stack
 - Great for dev, staging, and CI

Service Discovery with Docker

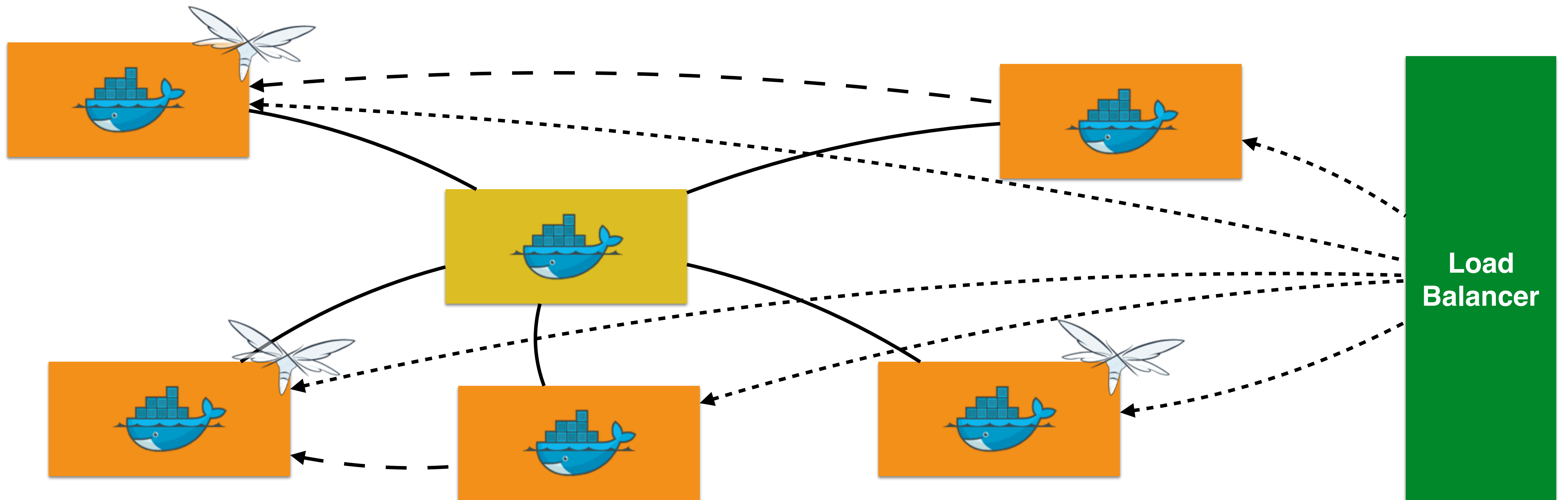
```
1  version: "3"
2  services:
3    db:
4      image: arungupta/couchbase:travel
5      ports:
6        - 8091:8091
7        - 8092:8092
8        - 8093:8093
9        - 11210:11210
10   web:
11     image: arungupta/wildfly-couchbase-javaee:travel
12     environment:
13       - COUCHBASE_URI=db
14 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

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

Stack



Load Balancing: Docker



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

Service Discovery: Kubernetes

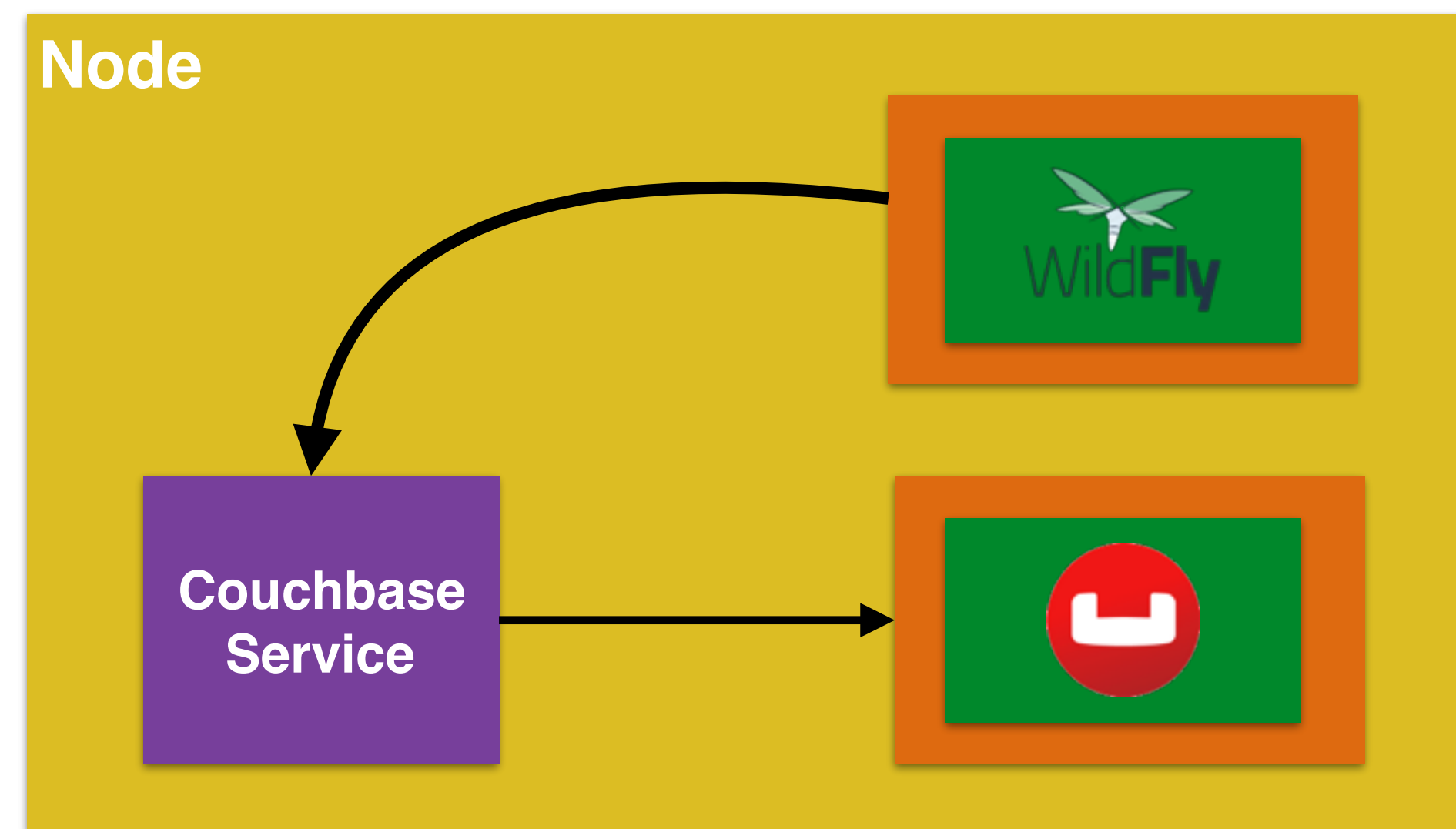
- Service: Abstract a set of pods as a single IP and port
 - Simple TCP/UDP load balancing
- Creates environment variables in other pods or DNS resolution
- Stable endpoint for pods to reference
 - Allows list of pods to change dynamically

Service Discovery with Kubernetes

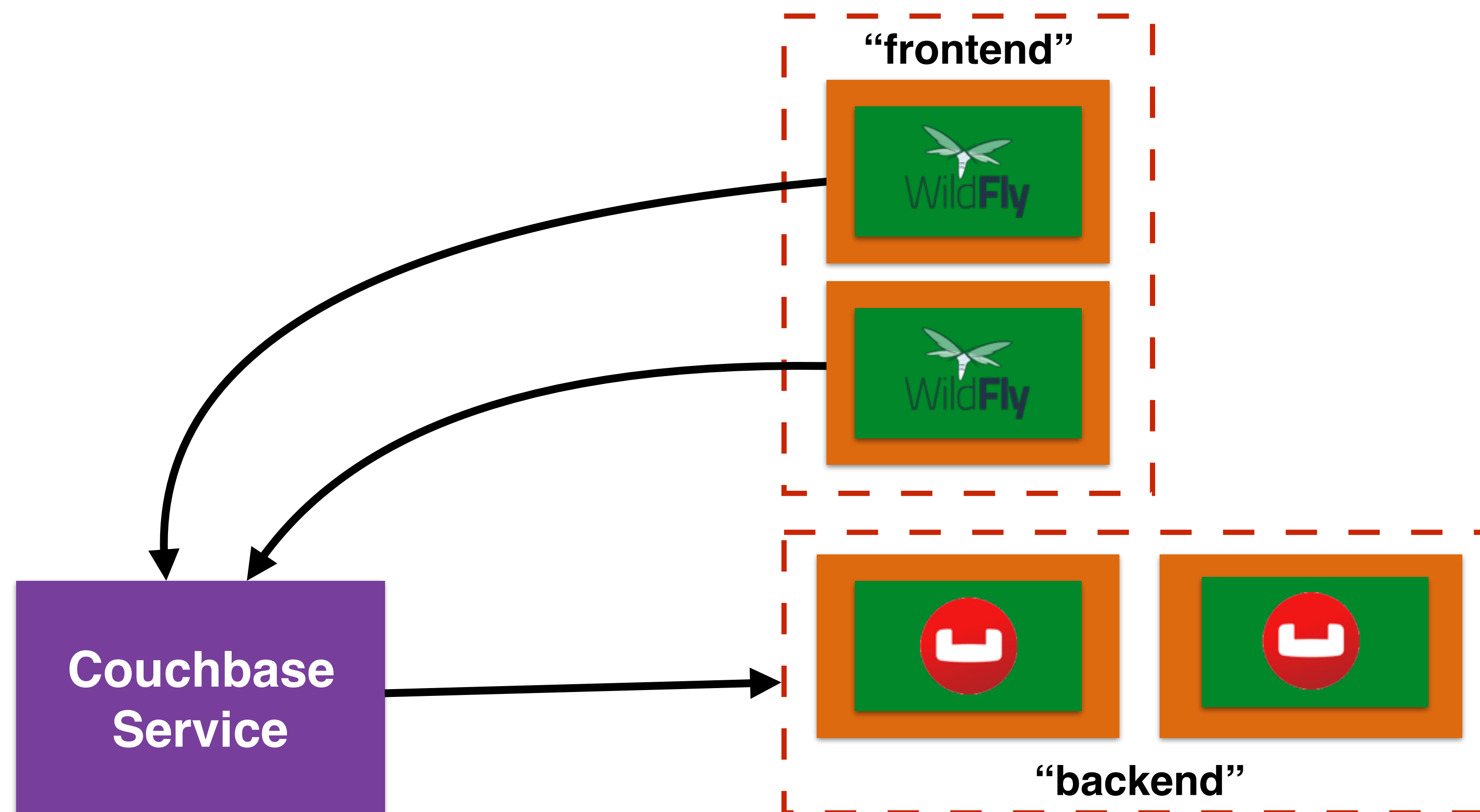
```
1  apiVersion: v1
2  kind: Service
3  metadata:
4  name: couchbase-service
5  spec:
6  selector:
7  app: couchbase-rs-pod
8  ports:
9  - name: admin
10  port: 8091
11  - name: views
12  port: 8092
13  - name: query
14  port: 8093
15  - name: memcached
16  port: 11210
17  ---
18  apiVersion: extensions/v1beta1
19  kind: ReplicaSet
20  metadata:
21  name: couchbase-rs
22  spec:
23  replicas: 1
24  template:
25  metadata:
26  labels:
27  app: couchbase-rs-pod
28  spec:
29  containers:
30  - name: couchbase
31  image: arungupta/couchbase:travel
32  ports:
33  - containerPort: 8091
34  - containerPort: 8092
35  - containerPort: 8093
36  - containerPort: 11210
37  ---
38  apiVersion: extensions/v1beta1
39  kind: ReplicaSet
40  metadata:
41  name: wildfly-rs
42  labels:
43  name: wildfly
44  spec:
45  replicas: 1
46  template:
47  metadata:
48  labels:
49  name: wildfly
50  spec:
51  containers:
52  - name: wildfly-rs-pod
53  image: arungupta/wildfly-couchbase-javaee:travel
54  env:
55  name: COUCHBASE_URI
56  value: couchbase-service
57  ports:
58  - containerPort: 8080
59
```

The diagram illustrates the service discovery configuration in Kubernetes. It shows three YAML snippets: a Service, a ReplicaSet for couchbase-rs, and a ReplicaSet for wildfly-rs. Red circles highlight key configuration elements: the Service name 'couchbase-service', the selector 'app: couchbase-rs-pod', the ReplicaSet template label 'app: couchbase-rs-pod', and the environment variable 'name: COUCHBASE_URI' with 'value: couchbase-service'. Arrows indicate the flow of information: from the Service name to the ReplicaSet selector, and from the ReplicaSet label to the environment variable value.

Couchbase Service: Kubernetes



Service and Replica Set: Kubernetes



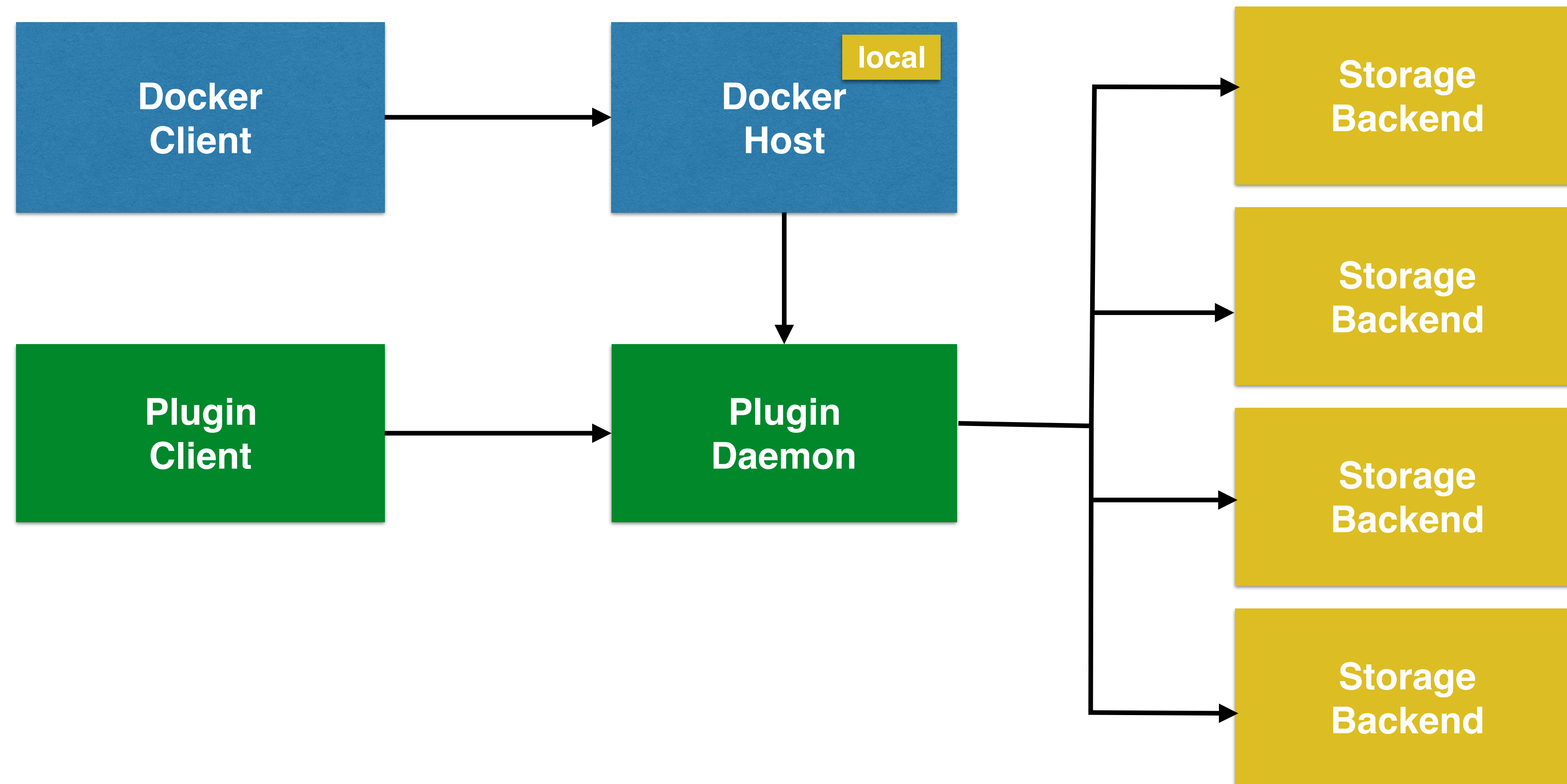
Docker Volumes: Persistent Containers

	Implicit Per-Container	Explicit Per-Container	Per-Host	Multi-Host
What?	Default sandbox	Explicit volume	Directory on host	Storage on distributed file systems
Location	<code>/var/lib/docker/volumes</code> on the host	<code>/var/lib/docker/volumes</code> 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)

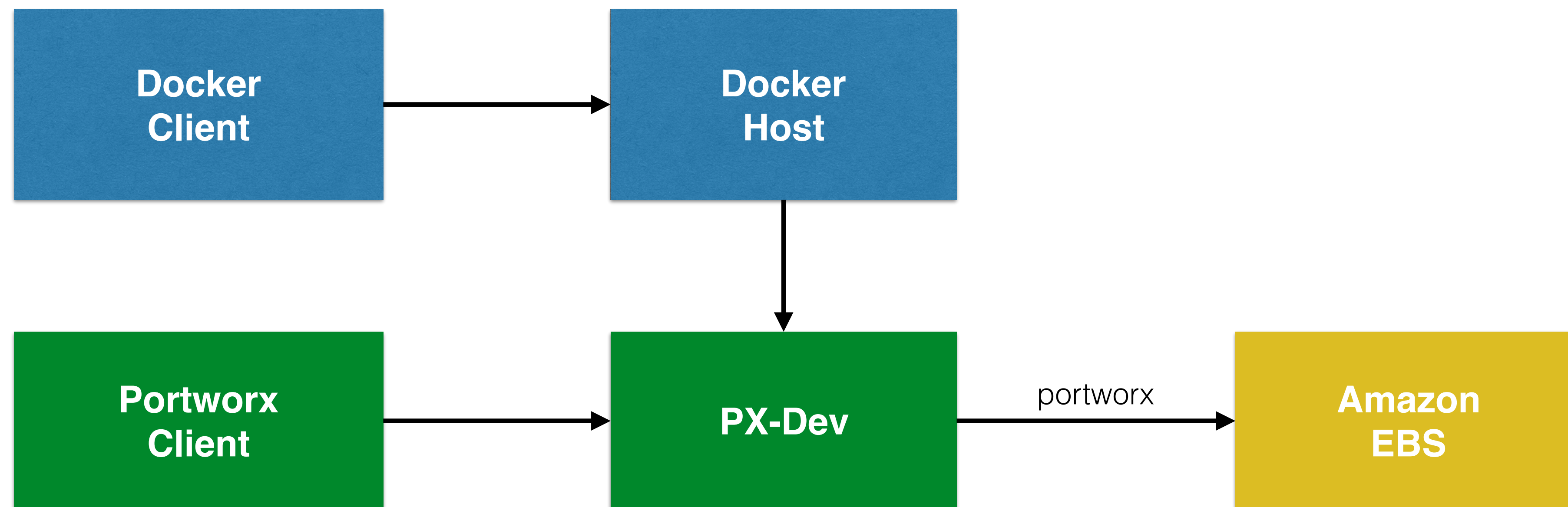
Docker Volume Plugin

- “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



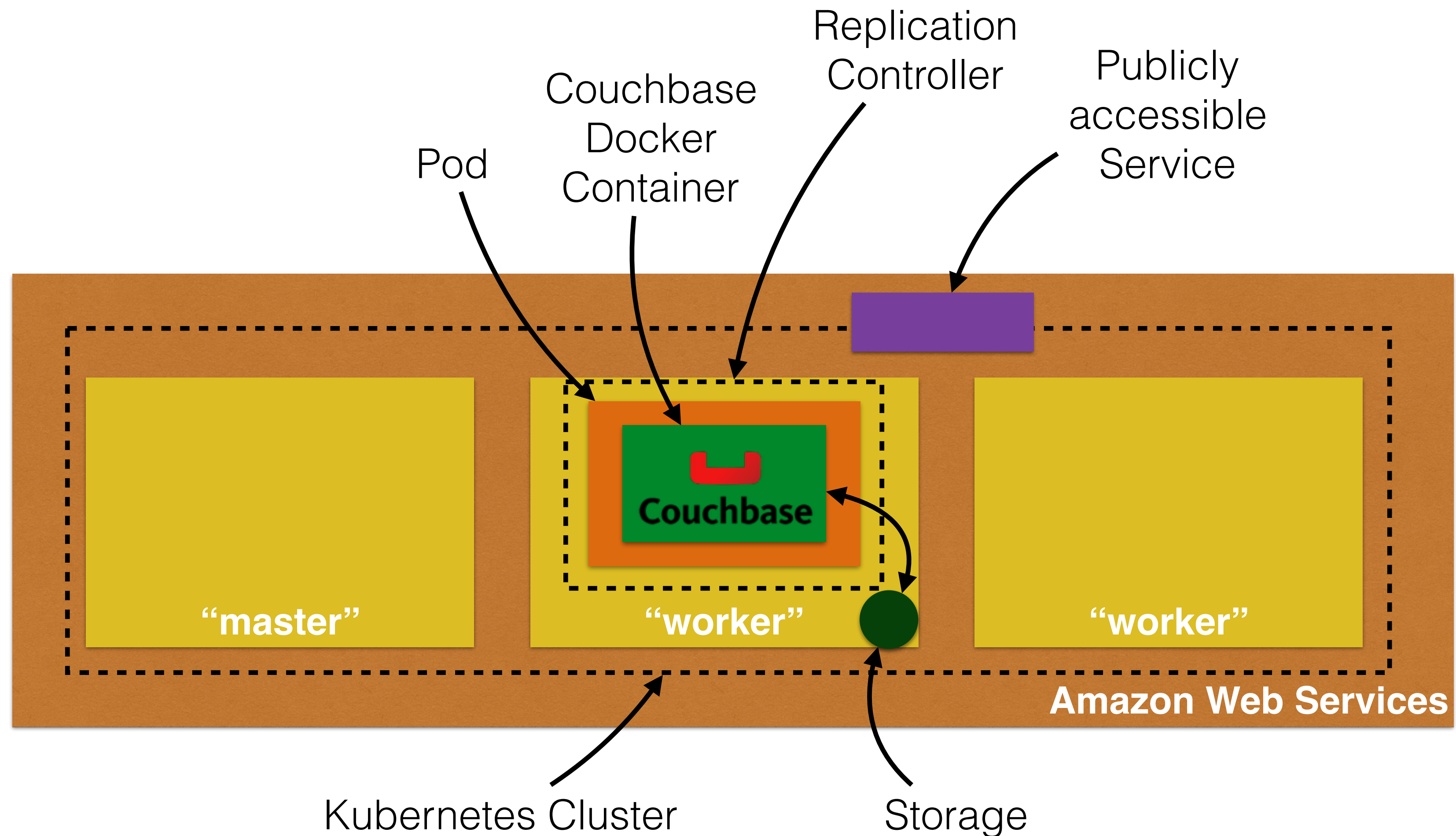
Kubernetes Volume

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

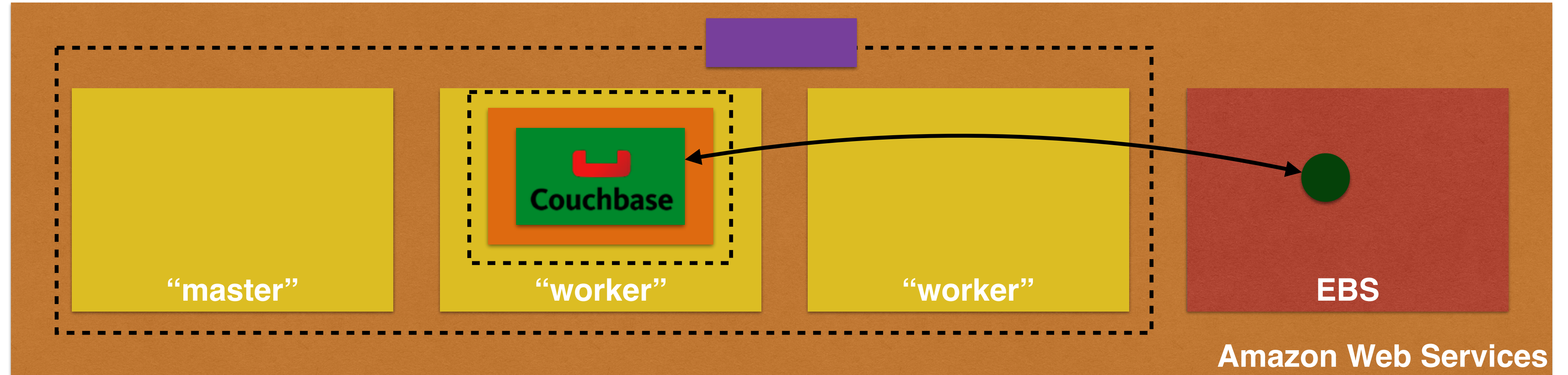
Kubernetes Persistent Volume



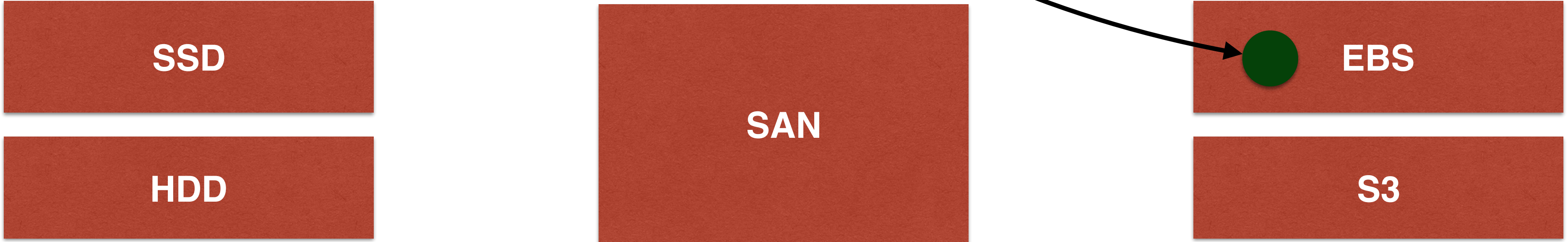
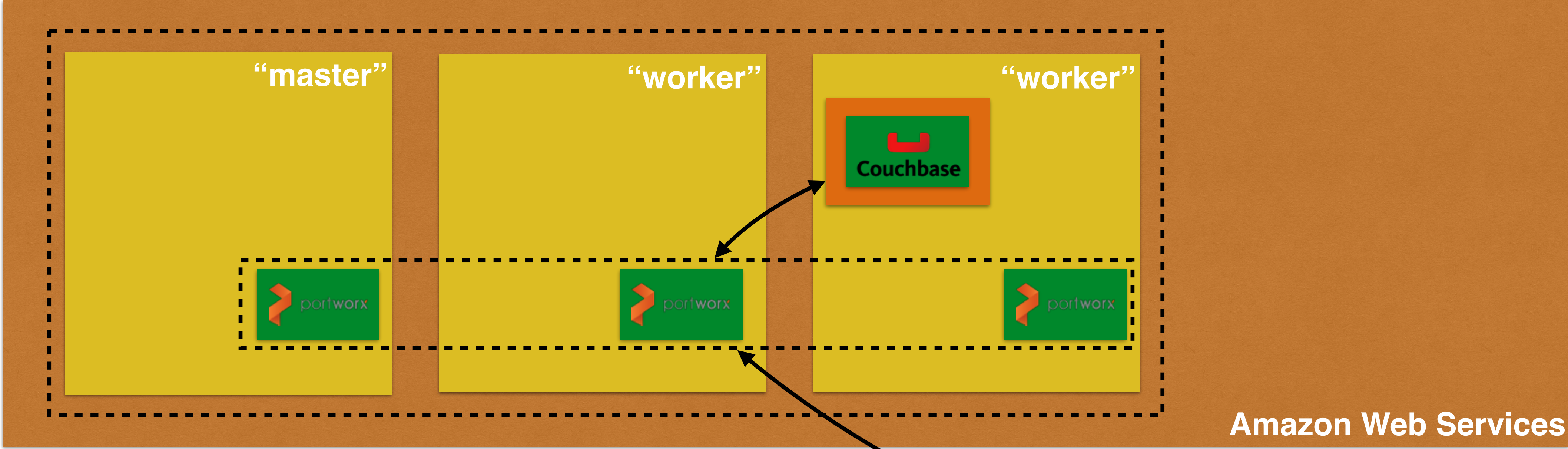
Kubernetes Volume: Persistent Container



Kubernetes Volume: Persistent Container



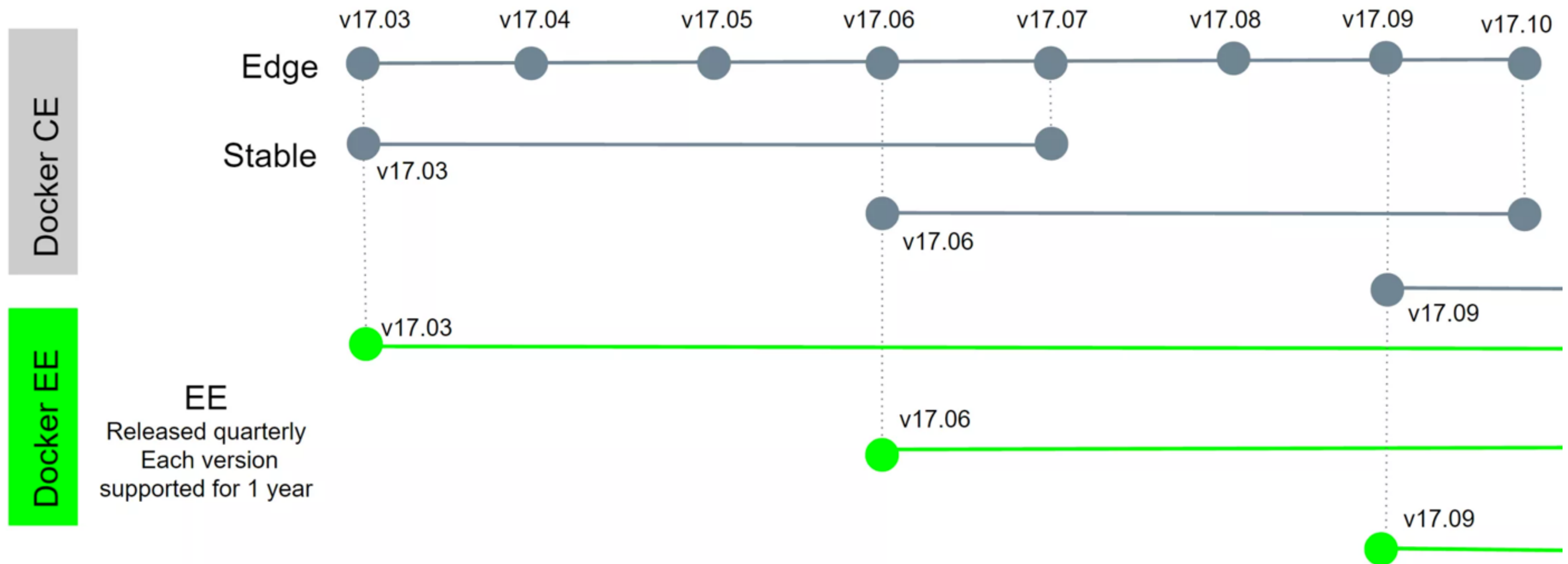
Kubernetes: Persistent Containers with Portworx



Development: Docker



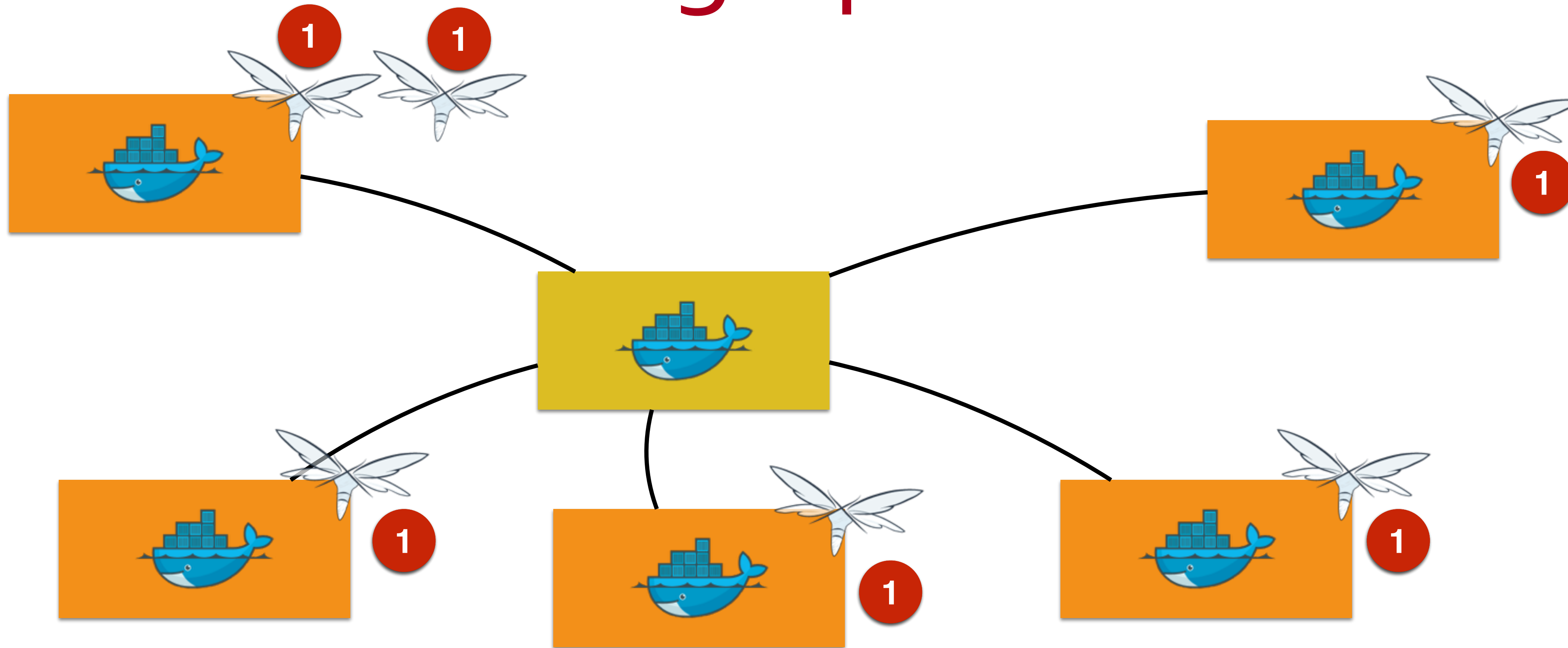
- Docker Community Edition
 - Docker for Mac/Windows/Linux
 - Monthly edge and quarterly stable releases
 - Native desktop or cloud provider experience



Development: Kubernetes

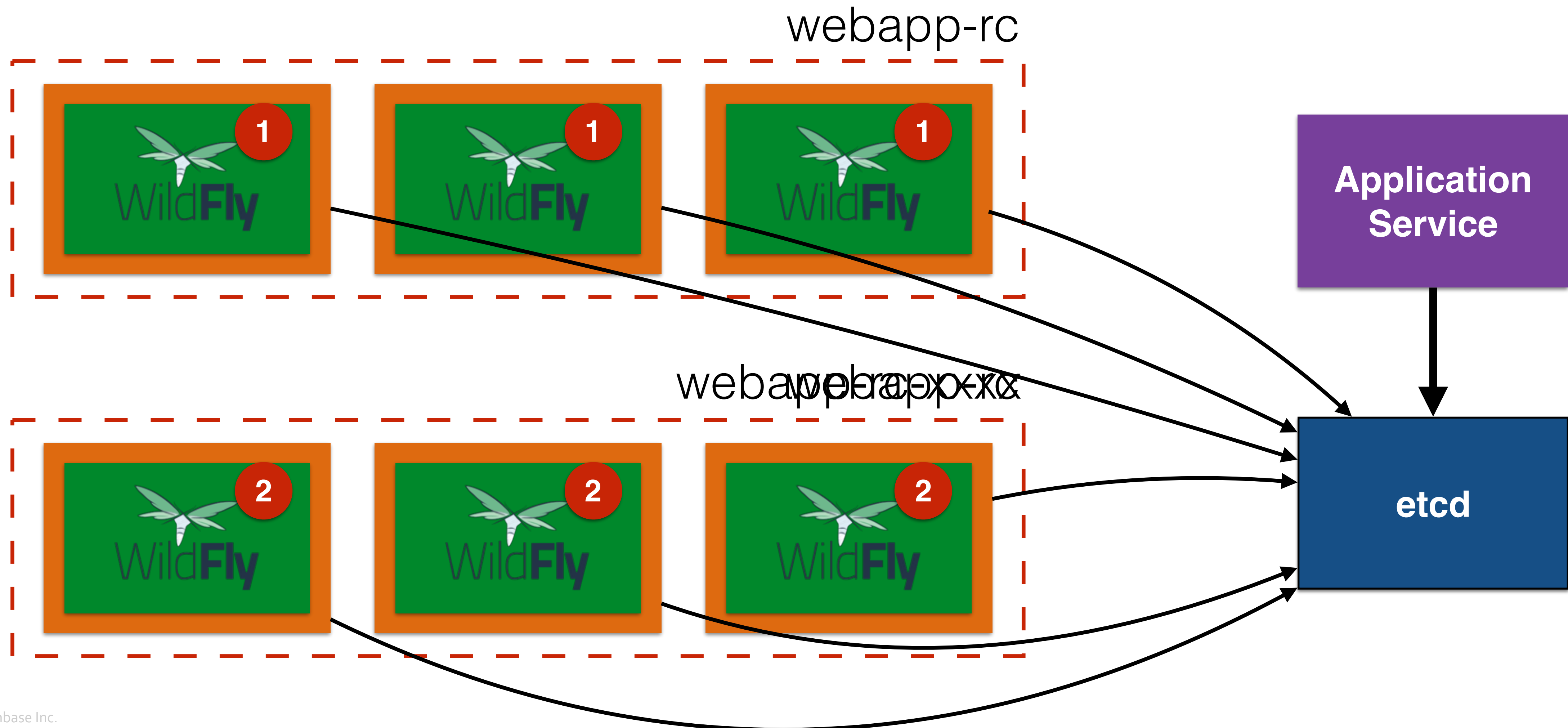
- Single node cluster
 - minikube
- Multi-node cluster
 - kops
 - kube-aws (CoreOS + AWS)
 - kube-up (deprecated)
 - Google Cloud, Azure, Tectonic, ...

Rolling Update: Docker



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

Rolling Update: Kubernetes



Monitoring: Docker

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

Monitoring: Docker

Prometheus Alerts Graph Status Help

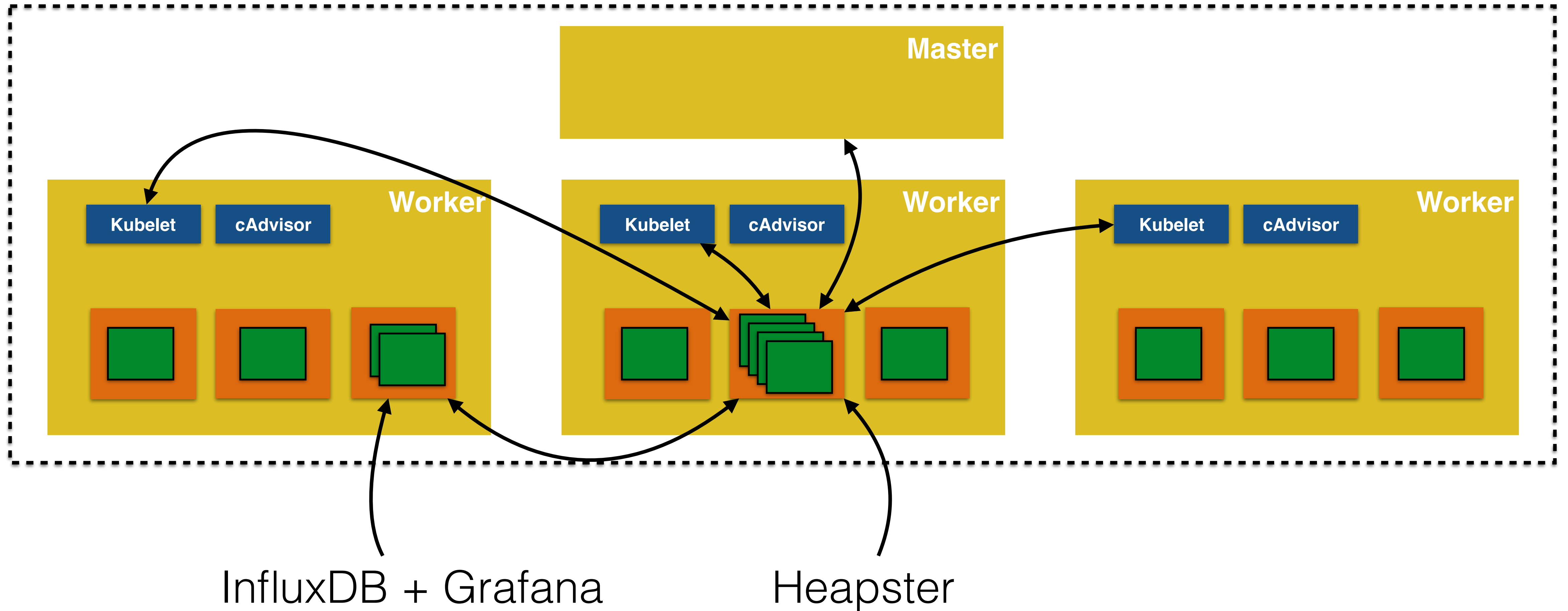
Expression (press Shift+Enter for newlines)

Execute ✓ - insert metric at cursor -

- go_gc_duration_seconds
- go_gc_duration_seconds_count
- go_gc_duration_seconds_sum
- go_goroutines
- go_memstats_alloc_bytes
- go_memstats_alloc_bytes_total
- go_memstats_buck_hash_sys_bytes
- go_memstats_frees_total
- go_memstats_gc_sys_bytes
- go_memstats_heap_alloc_bytes
- go_memstats_heap_idle_bytes
- go_memstats_heap_inuse_bytes
- go_memstats_heap_objects
- go_memstats_heap_released_bytes_total
- go_memstats_heap_sys_bytes
- go_memstats_last_gc_time_seconds
- go_memstats_lookups_total
- go_memstats_mallocs_total
- go_memstats_mcache_inuse_bytes
- go_memstats_mcache_sys_bytes
- go_memstats_mspan_inuse_bytes
- go_memstats_mspan_sys_bytes
- go_memstats_next_gc_bytes
- go_memstats_other_sys_bytes
- go_memstats_stack_inuse_bytes
- go_memstats_stack_sys_bytes
- go_memstats_sys_bytes
- http_request_duration_microseconds
- http_request_duration_microseconds_count
- http_request_duration_microseconds_sum
- http_request_size_bytes
- http_request_size_bytes_count
- http_request_size_bytes_sum
- http_requests_total
- http_response_size_bytes
- http_response_size_bytes_count
- http_response_size_bytes_sum
- process_cpu_seconds_total



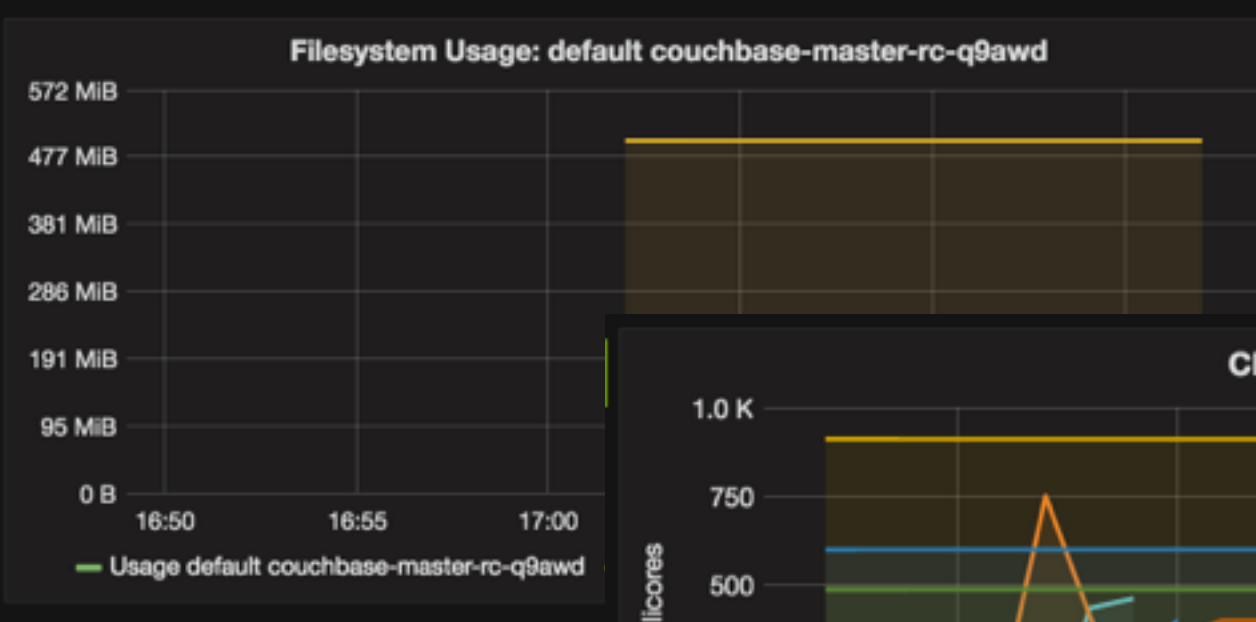
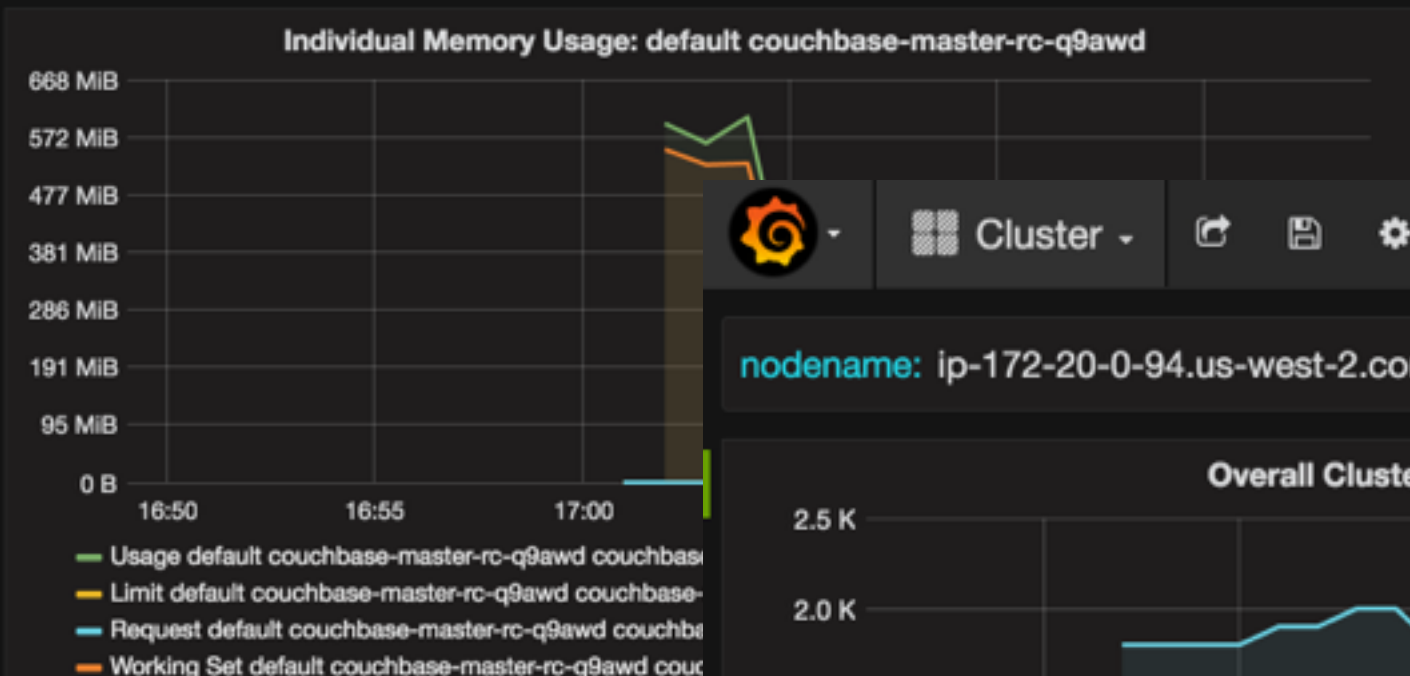
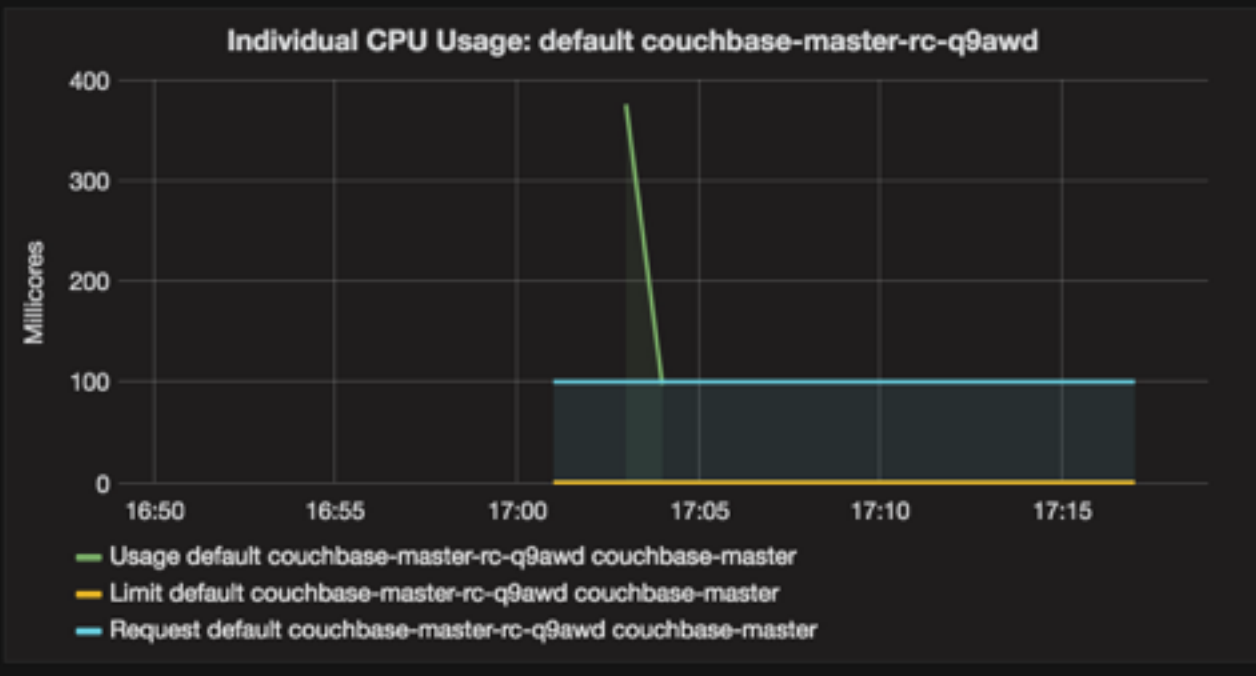
Monitoring: Kubernetes



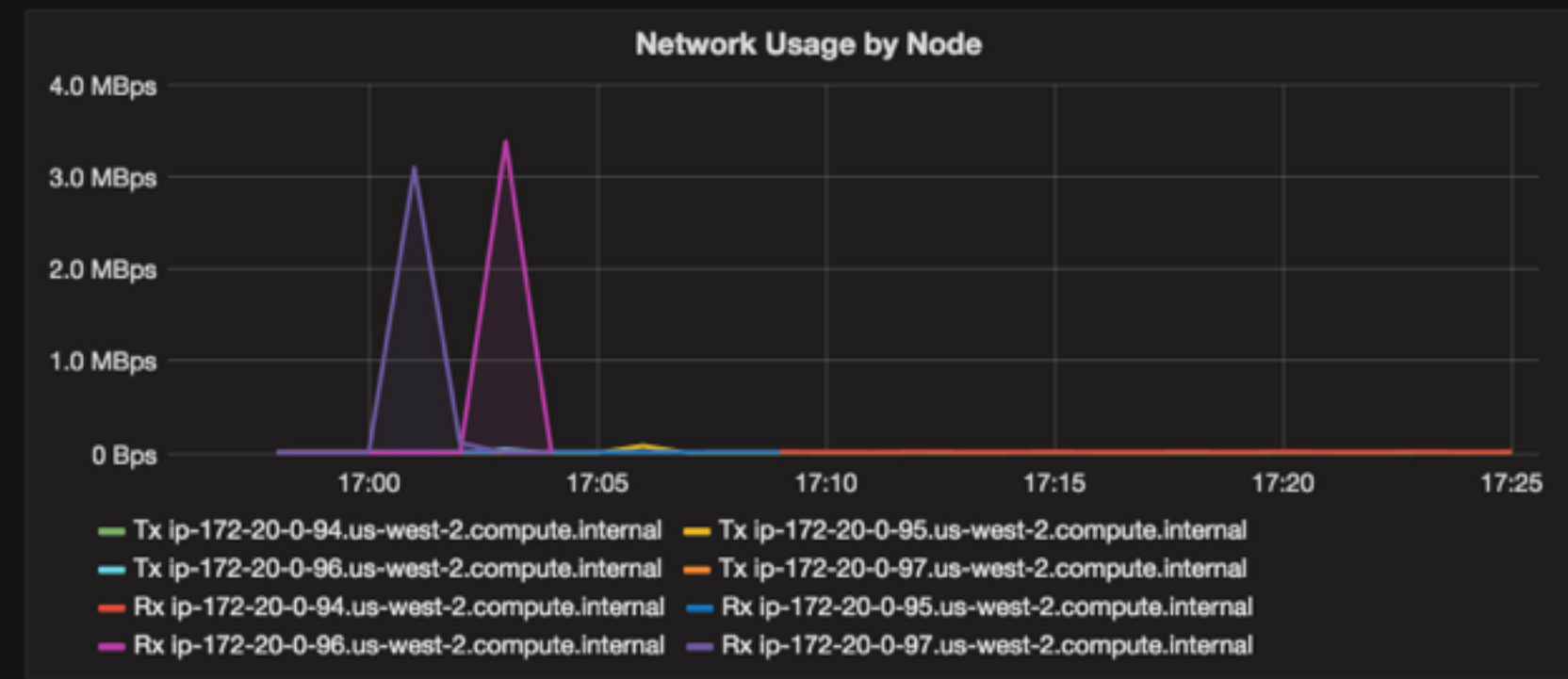
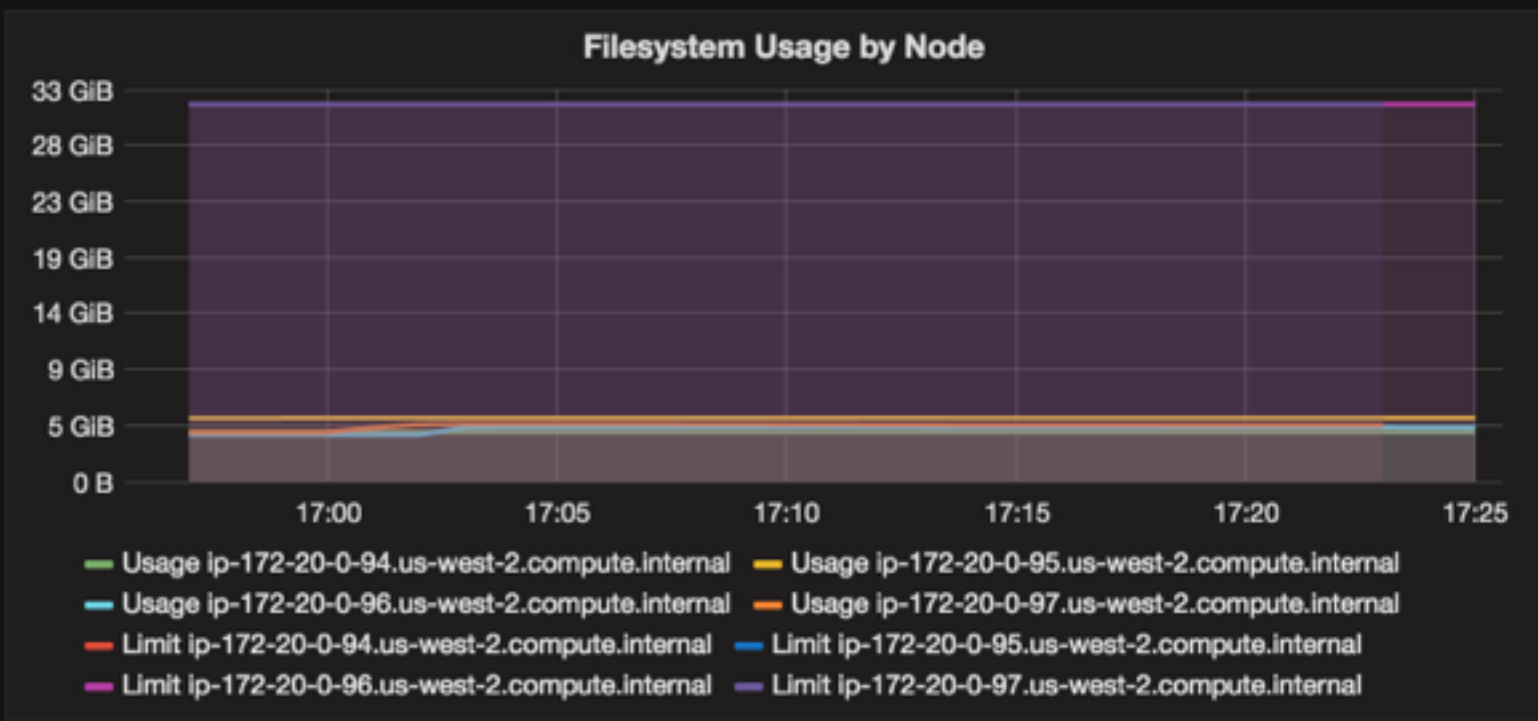
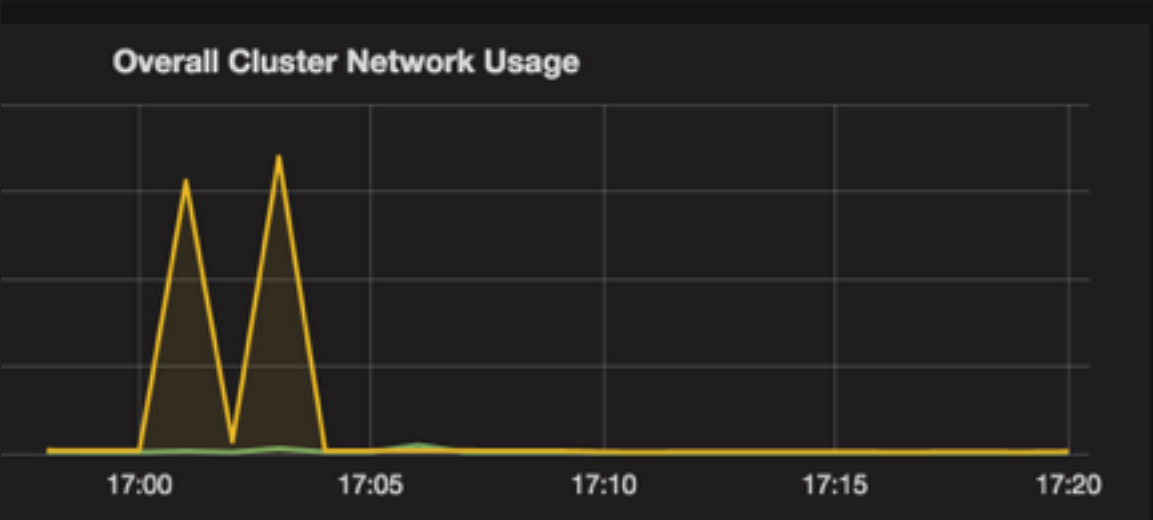
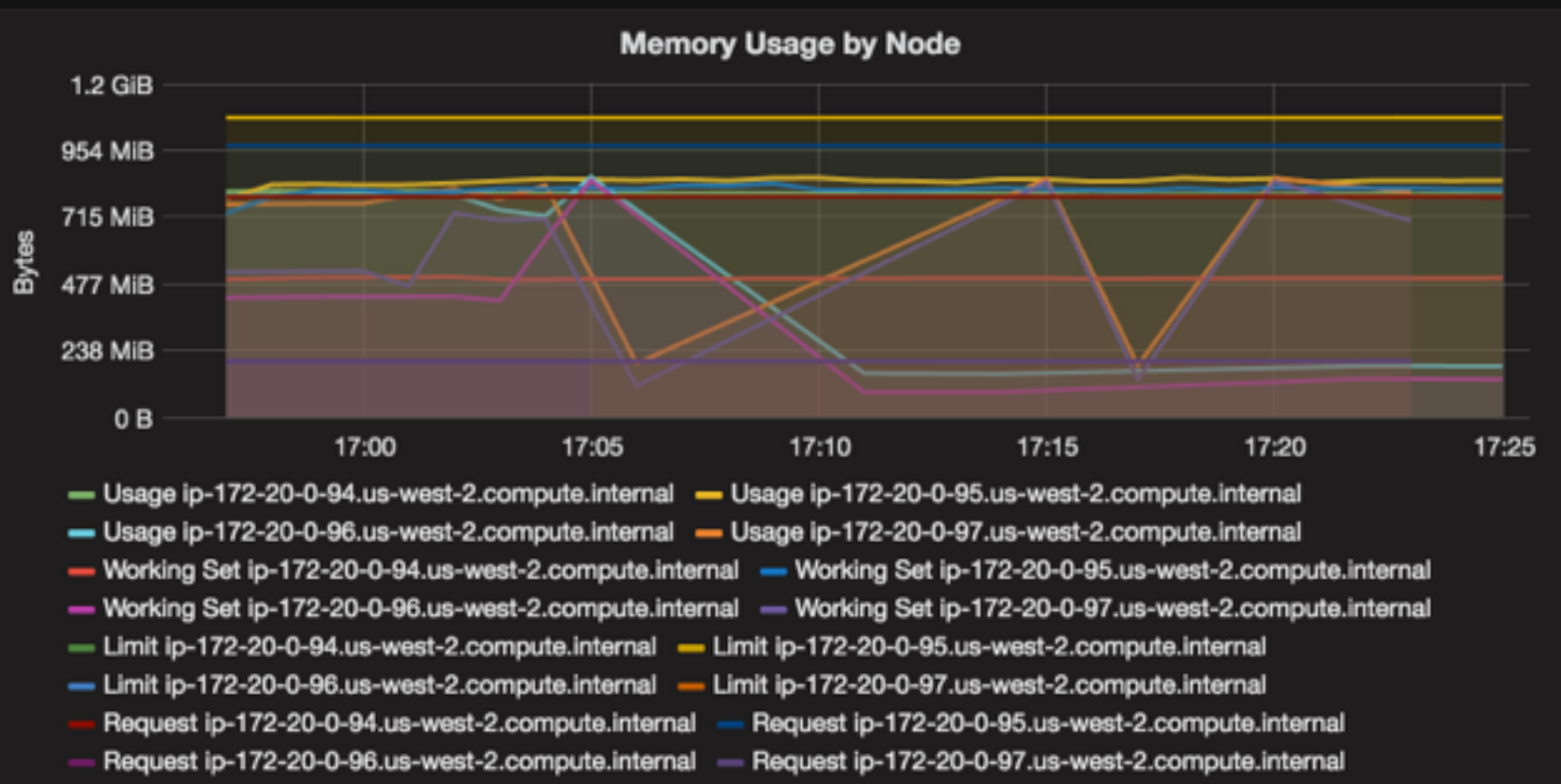
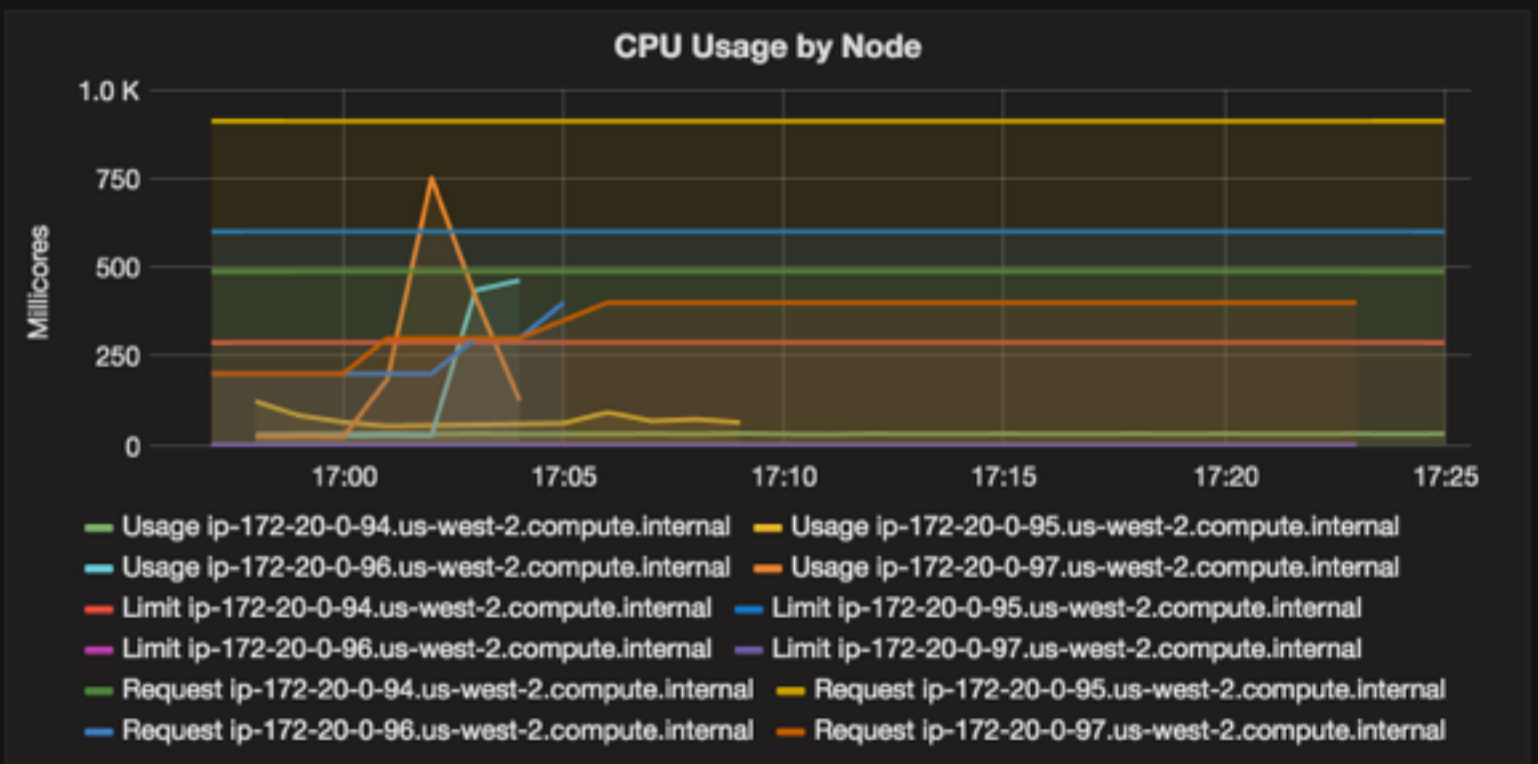
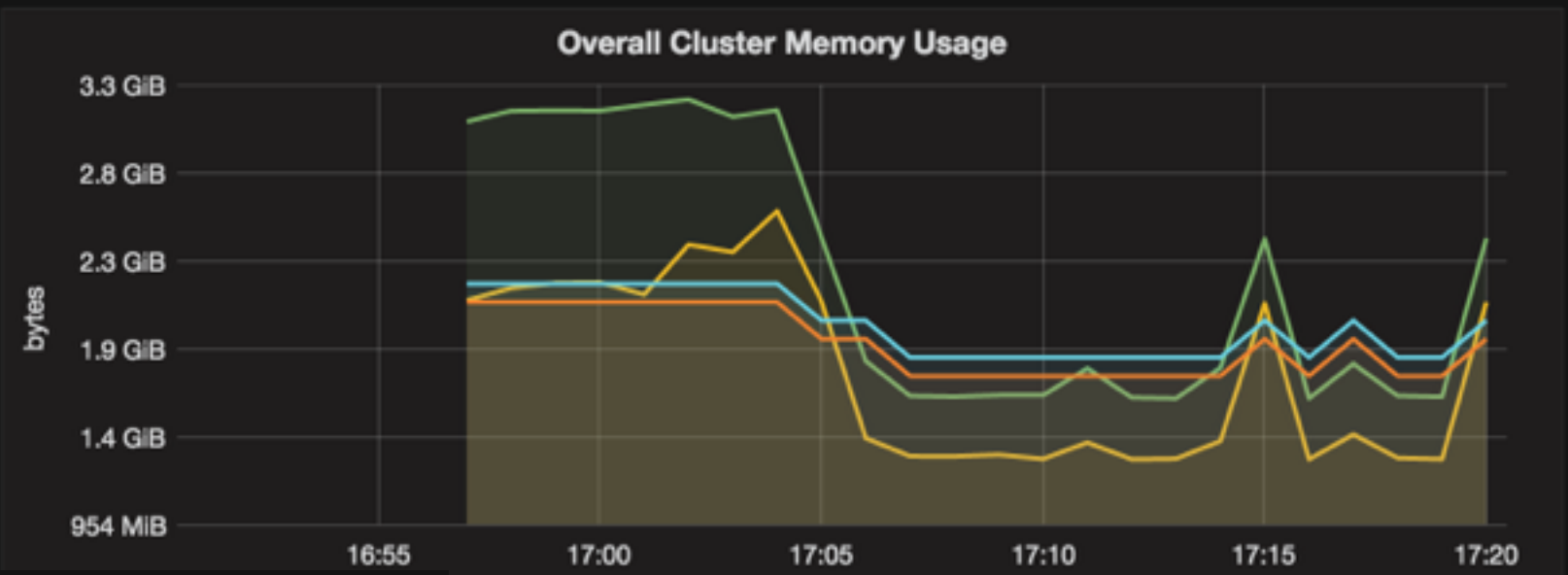
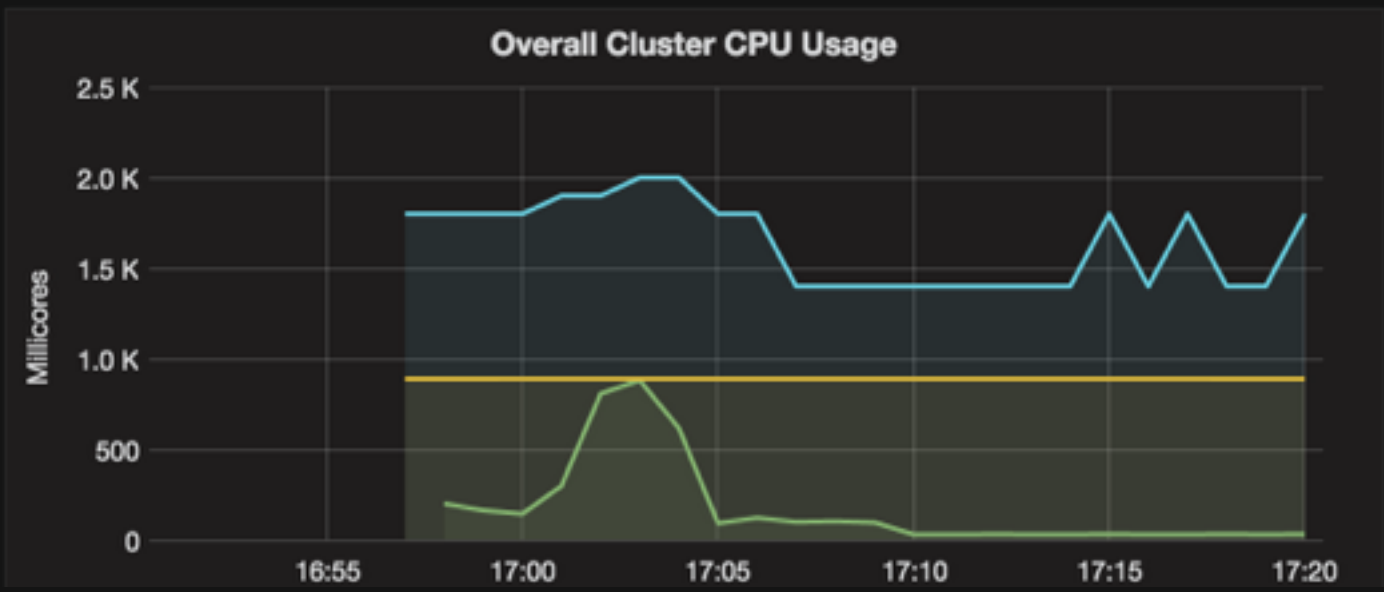
InfluxDB + Grafana

Heapster

namespace: default | podname: couchbase-master-rc-q9awd



Cluster | nodename: ip-172-20-0-94.us-west-2.compute.internal



Monitoring: Docker and Kubernetes



References

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