

Getting Started with



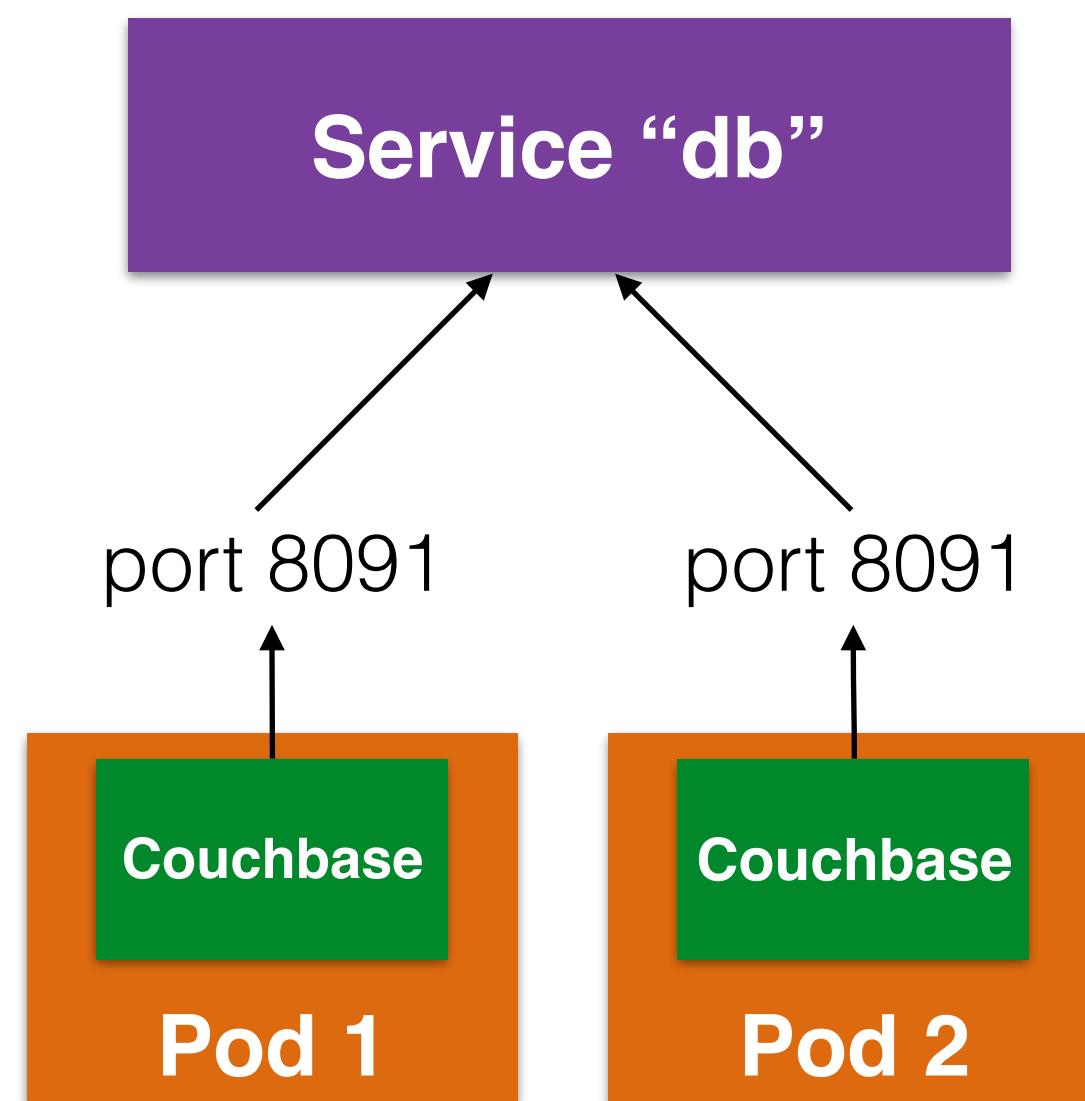
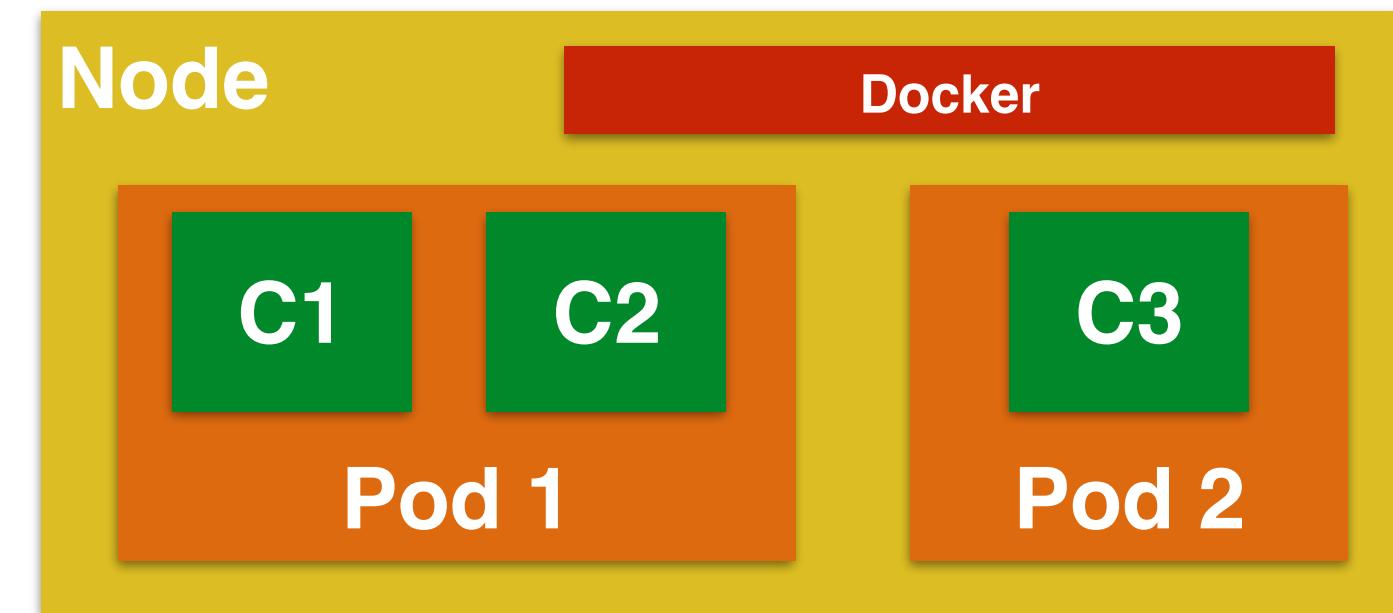
Arun Gupta
Vice President, Developer Advocacy
Couchbase
[@arungupta](https://twitter.com/arungupta), blog.arungupta.me
arun@couchbase.com

Kubernetes

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

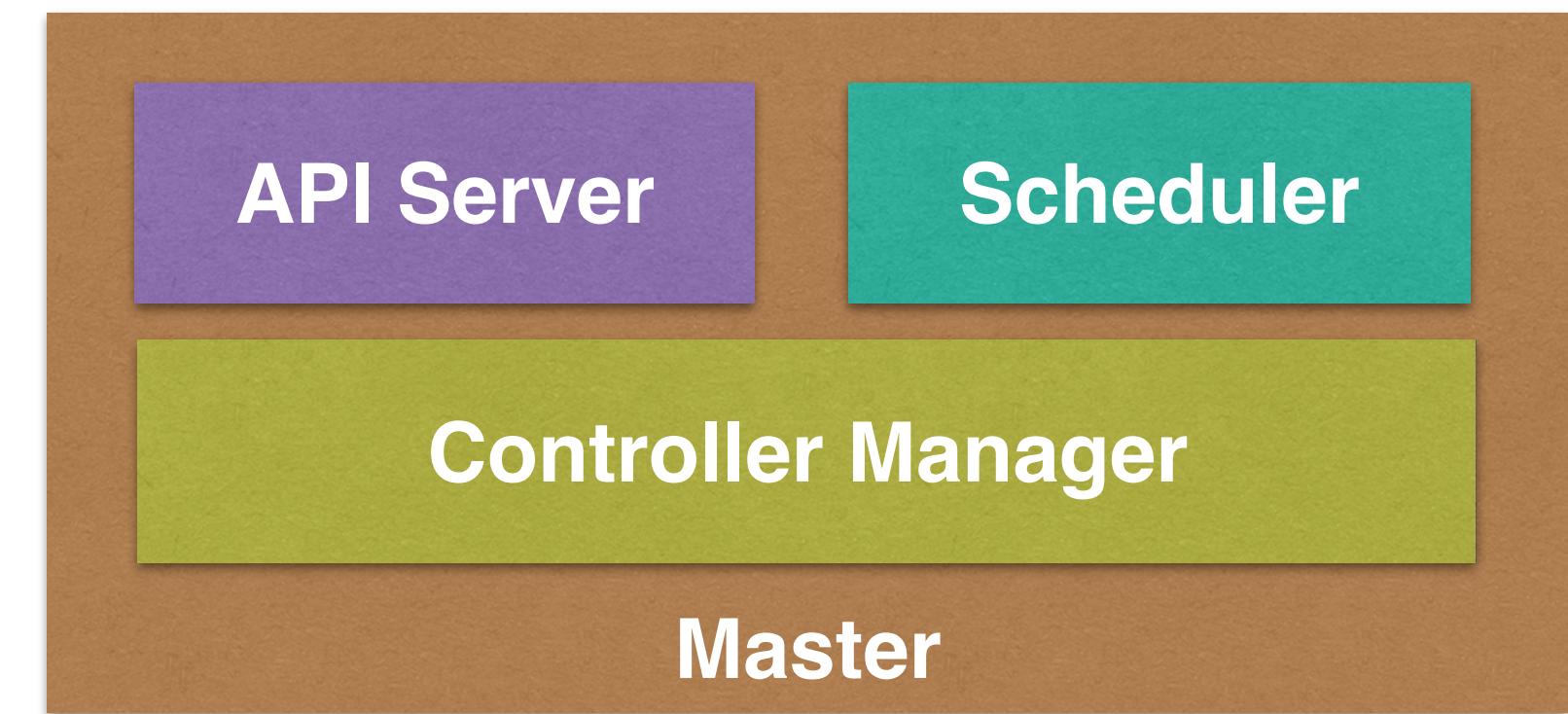
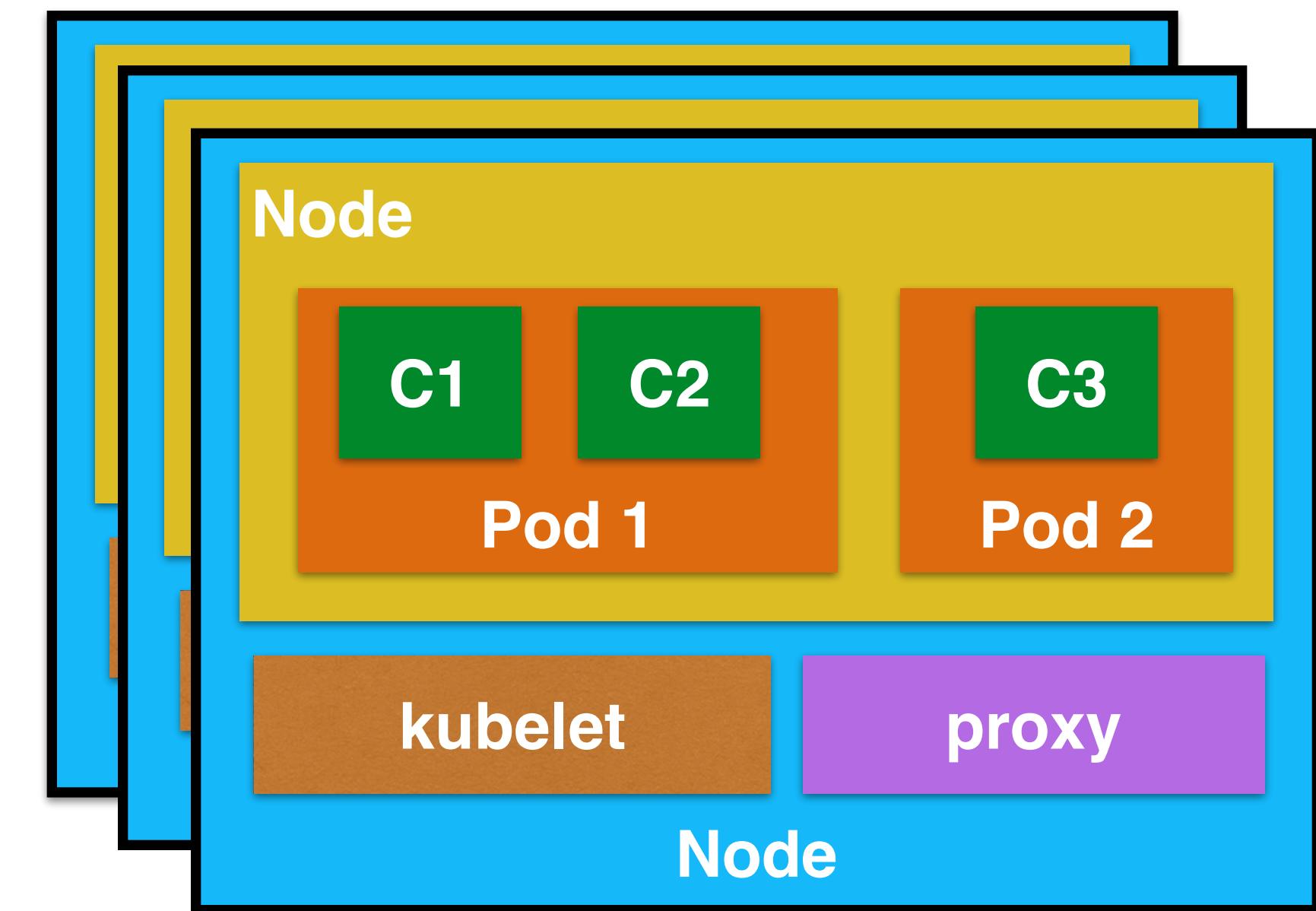
Concepts

- **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
- **Replication Controller**: manages the lifecycle of pods and ensures specified number are running

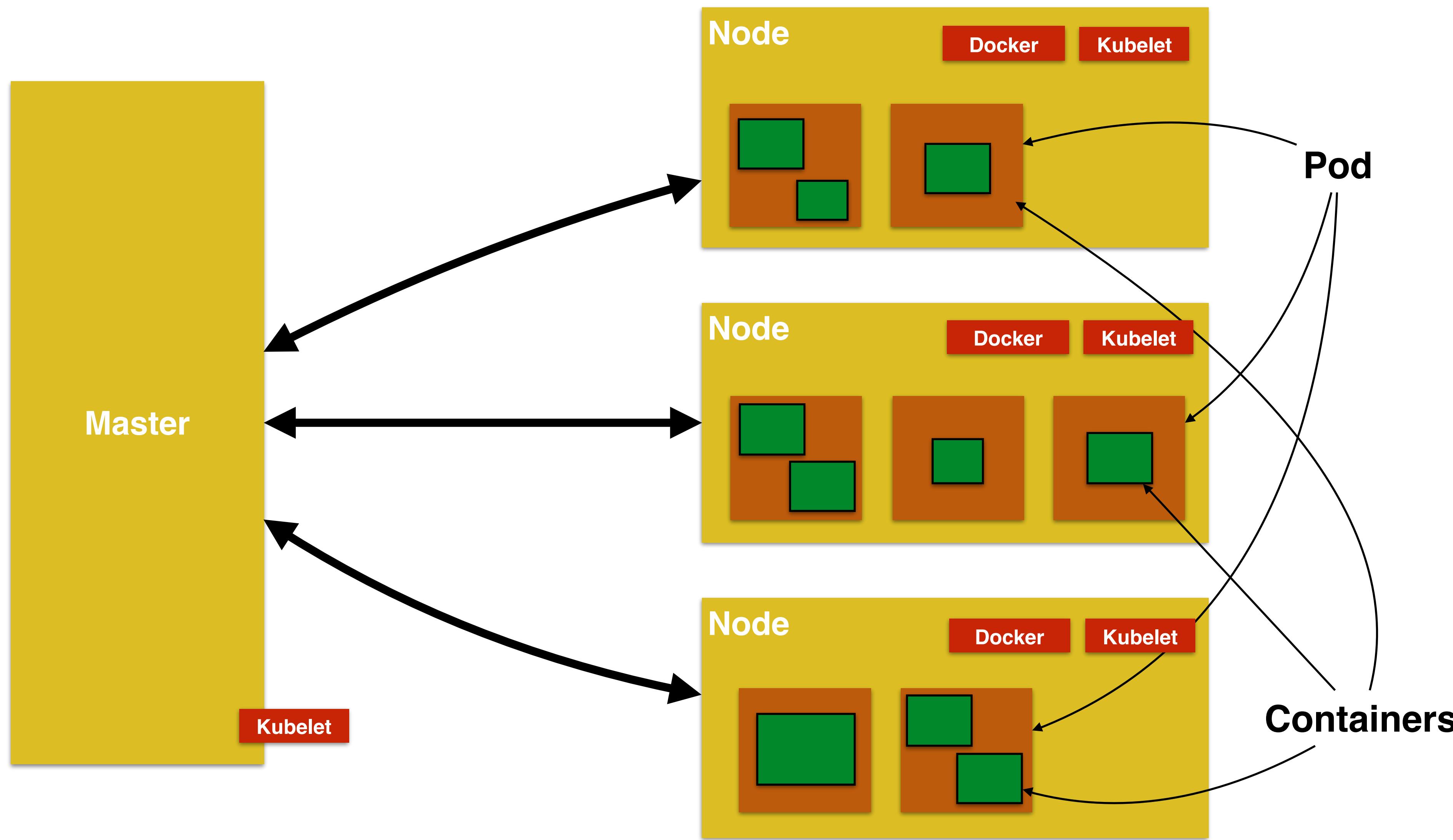


Components

- **Node**: Docker host running *kubelet* (node agent) and *proxy* services
 - Monitored by *systemd* (CentOS) or *monit* (Debian)
- **Master**: hosts cluster-level control services, including the API server, scheduler, and controller manager
- **etcd**: distributed key-value store used to persist Kubernetes system state



Architecture





Master High Availability

- Hack by running a `podmaster` utility
- Proposal
 - Hot Standby
 - Warm Standby
 - Active-Active (Load Balanced)

kubectl

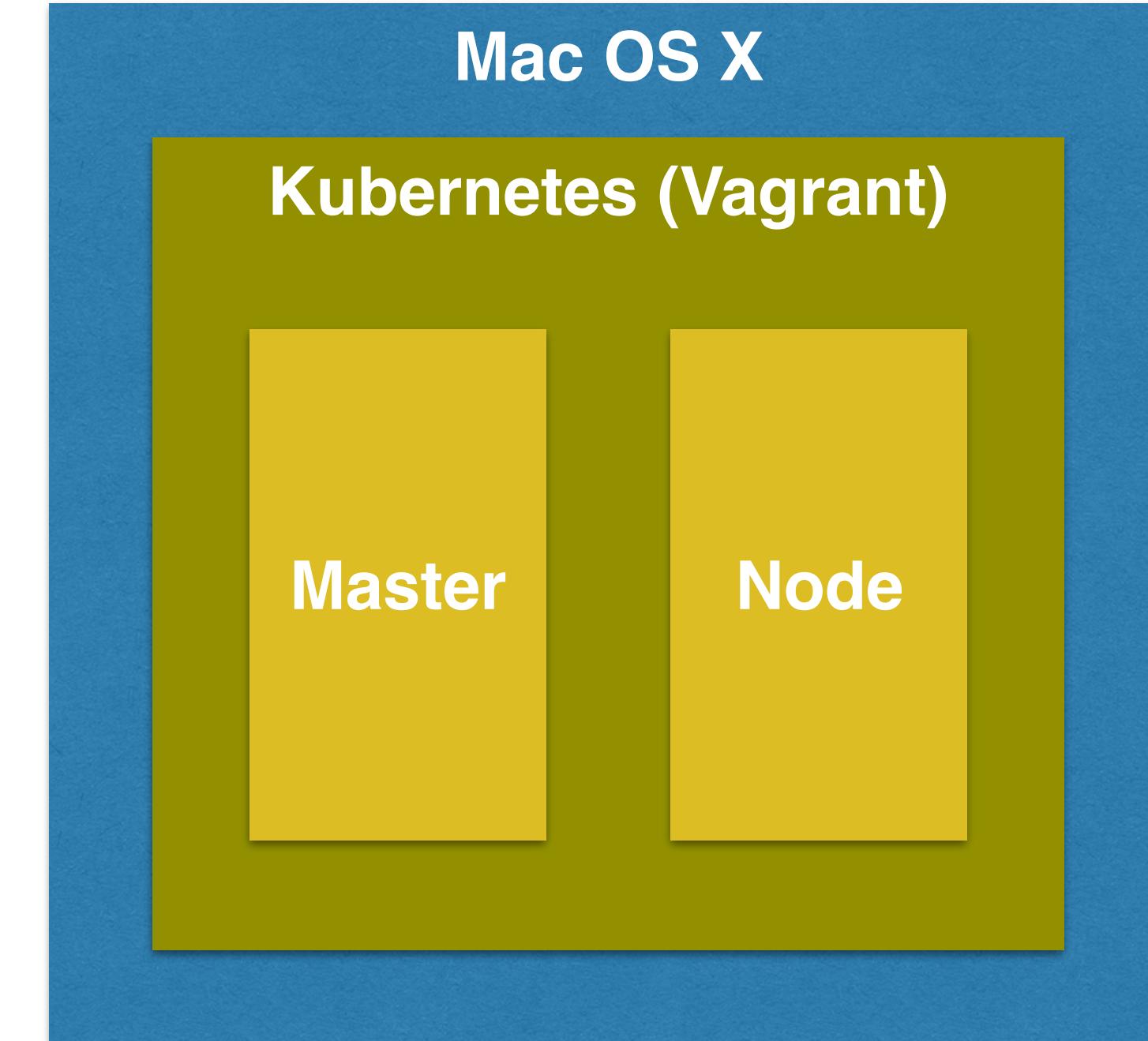
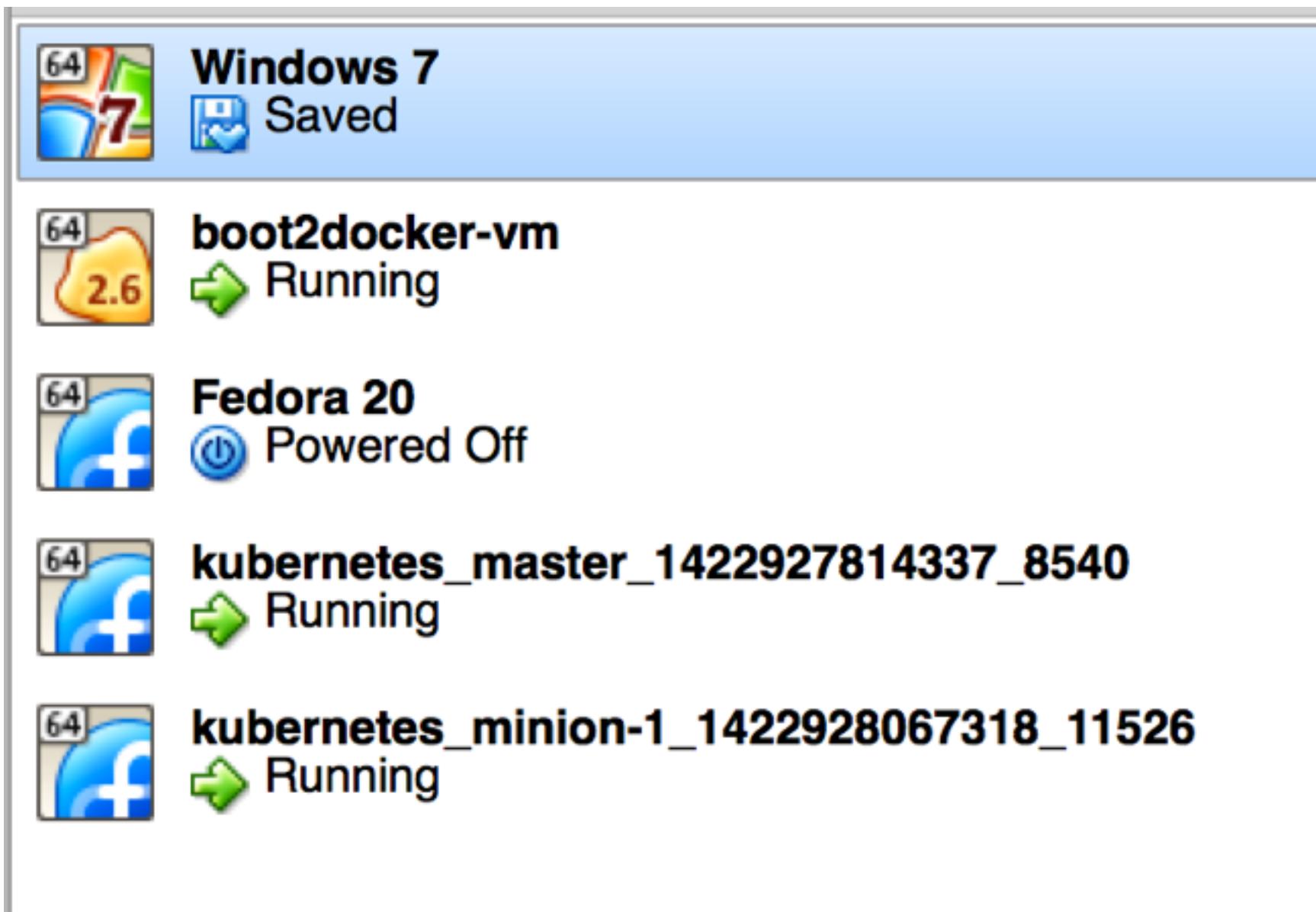
- Controls the Kubernetes cluster manager
- `kubectl get pods or minions`
- `kubectl create -f <filename>`
- `kubectl update or delete`
- `kubectl resize --replicas=3 replicationcontrollers <name>`

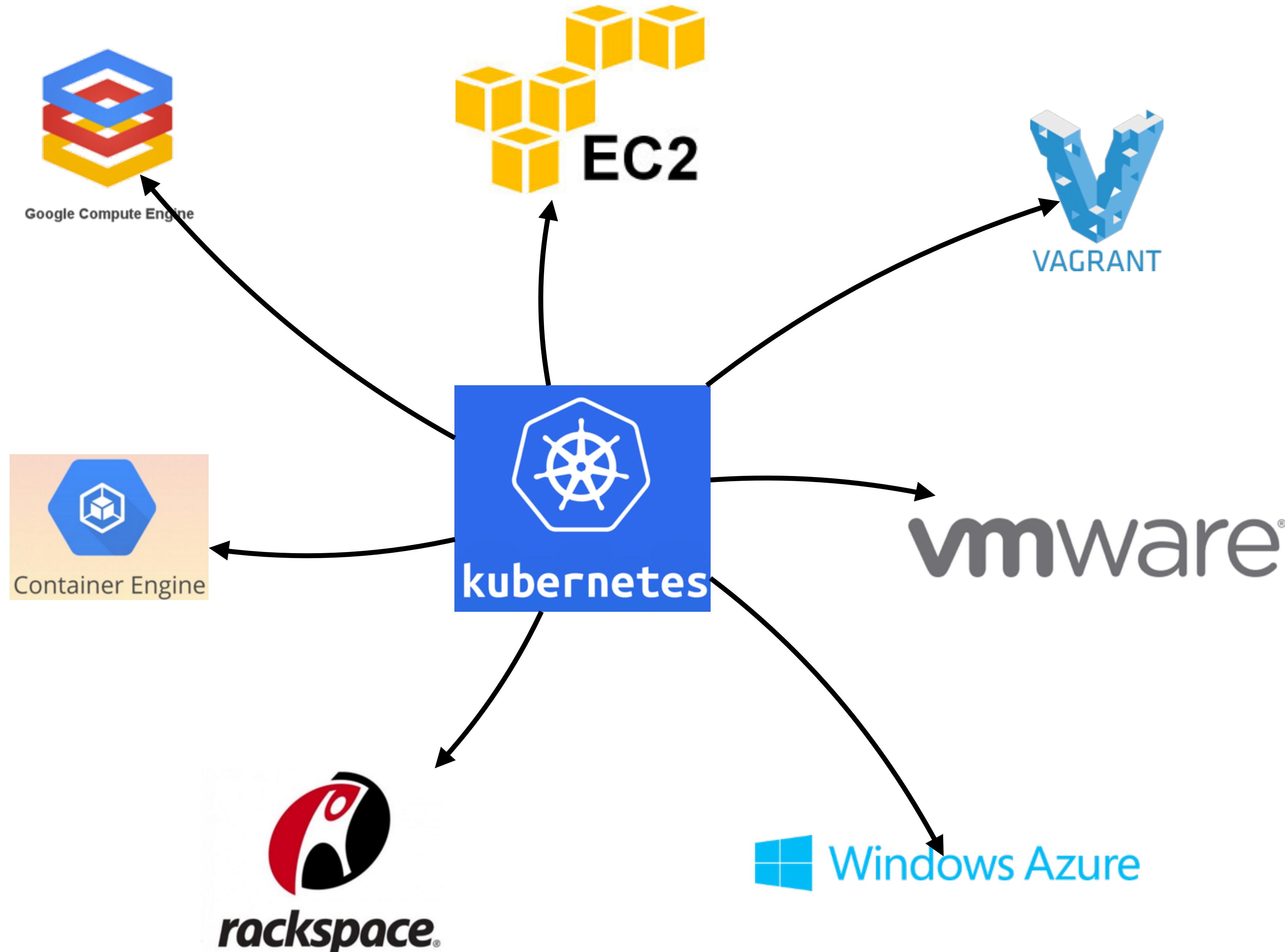
Kubernetes Config

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

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: wildfly-rc
5   labels:
6     name: wildfly
7 spec:
8   replicas: 2
9   template:
10  metadata:
11  labels:
12    name: wildfly
13  spec:
14    containers:
15      - name: wildfly-rc-pod
16        image: jboss/wildfly
17        ports:
18          - containerPort: 8080
```

```
export KUBERNETES_PROVIDER=vagrant  
./cluster/kube-up.sh
```

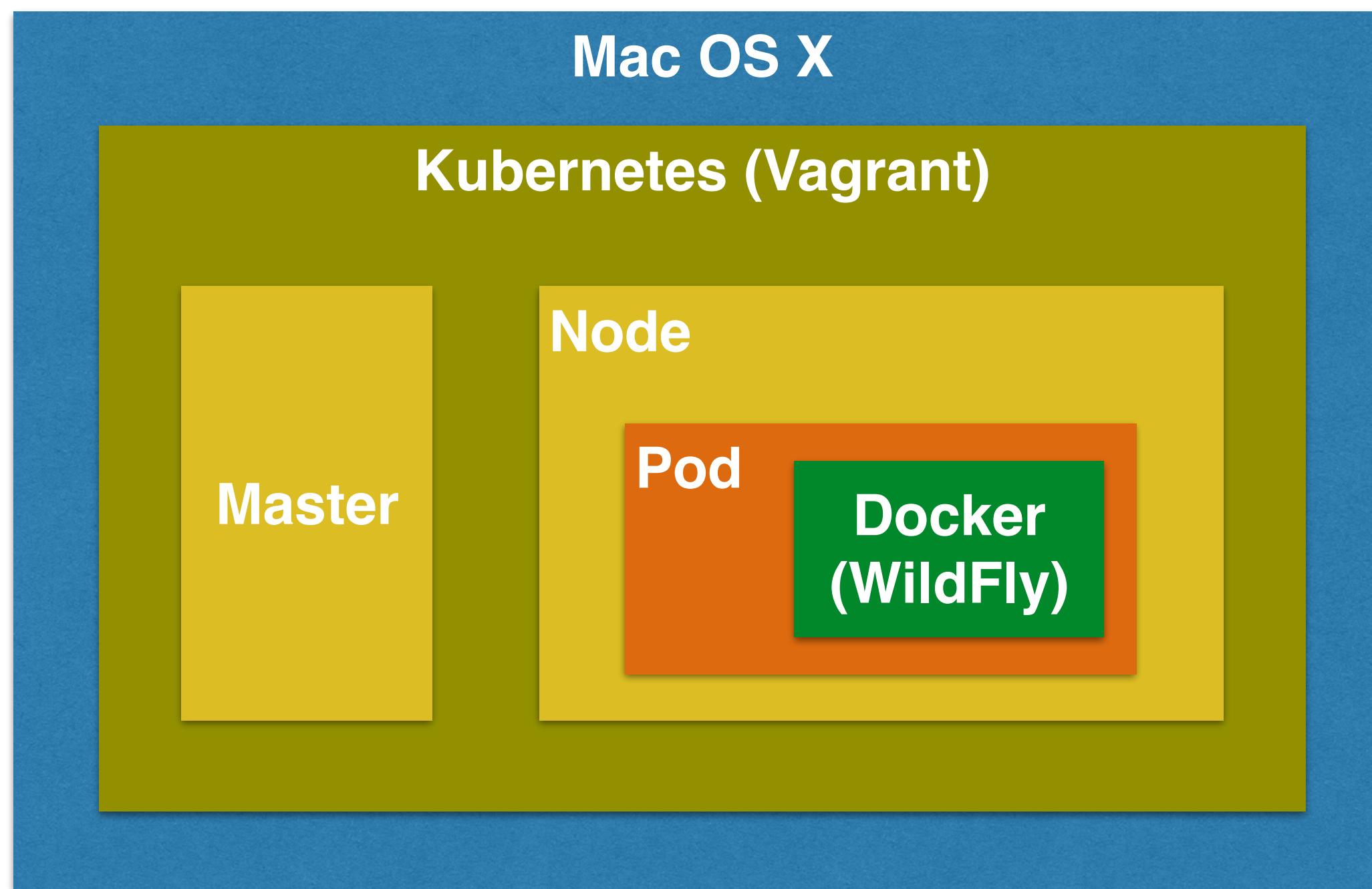




<https://get.k8s.io/>

A Pod with One Container

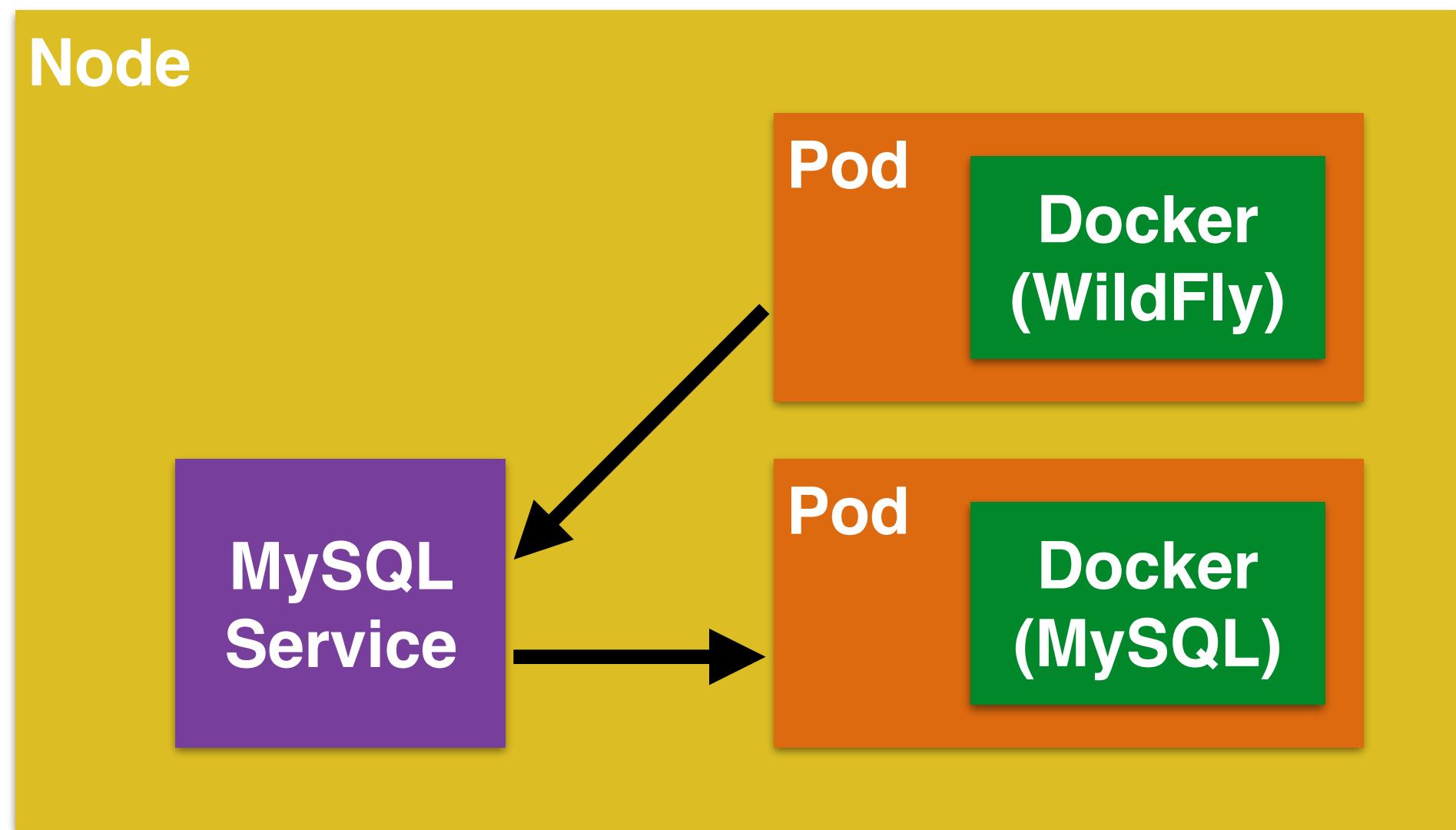
```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: wildfly-pod
5   labels:
6     name: wildfly
7 spec:
8   containers:
9     - image: jboss/wildfly
10    name: wildfly-pod
11    ports:
12      - containerPort: 8080
```



Services

- Abstract a set of pods as a single IP and port
 - Simple TCP/UDP load balancing
- Creates environment variables in other pods
 - Like “Docker links” but across hosts
- Stable endpoint for pods to reference
 - Allows list of pods to change dynamically

Services

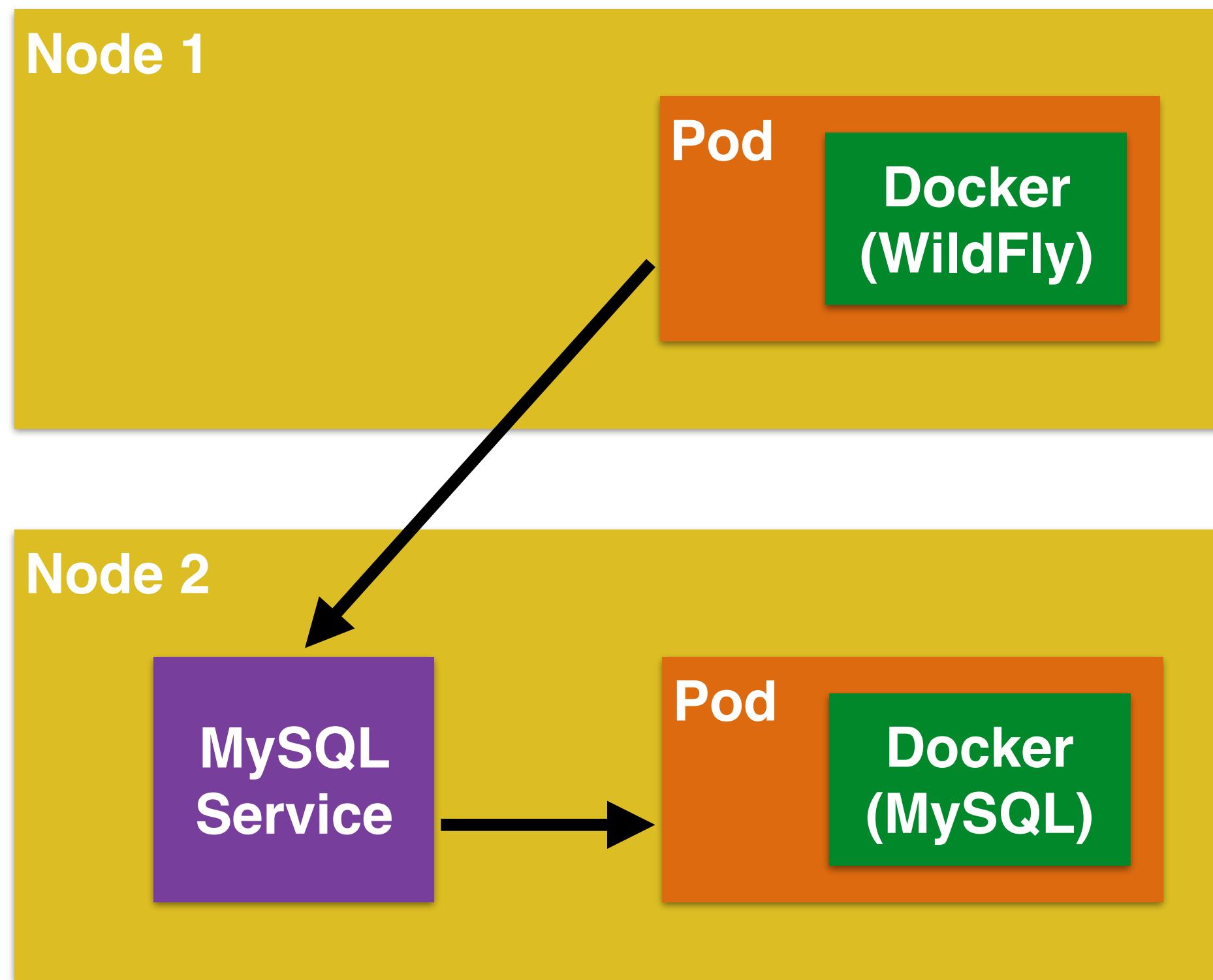


Services

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: mysql-pod
5   labels:
6     name: mysql-pod
7     context: docker-k8s-lab
8 spec:
9   containers:
10    -
11      name: mysql
12      image: mysql:latest
13      env:
14        -
15          name: "MYSQL_USER"
16          value: "mysql"
17        -
18          name: "MYSQL_PASSWORD"
19          value: "mysql"
20        -
21          name: "MYSQL_DATABASE"
22          value: "sample"
23        -
24          name: "MYSQL_ROOT_PASSWORD"
25          value: "supersecret"
26   ports:
27    -
28      containerPort: 3306
```

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: mysql-service
5   labels:
6     name: mysql-pod
7     context: docker-k8s-lab
8 spec:
9   ports:
10    # the port that this service should
11    - port: 3306
12    # label keys and values that must mat
13    selector:
14      name: mysql-pod
15      context: docker-k8s-lab
```

Service across Two Nodes



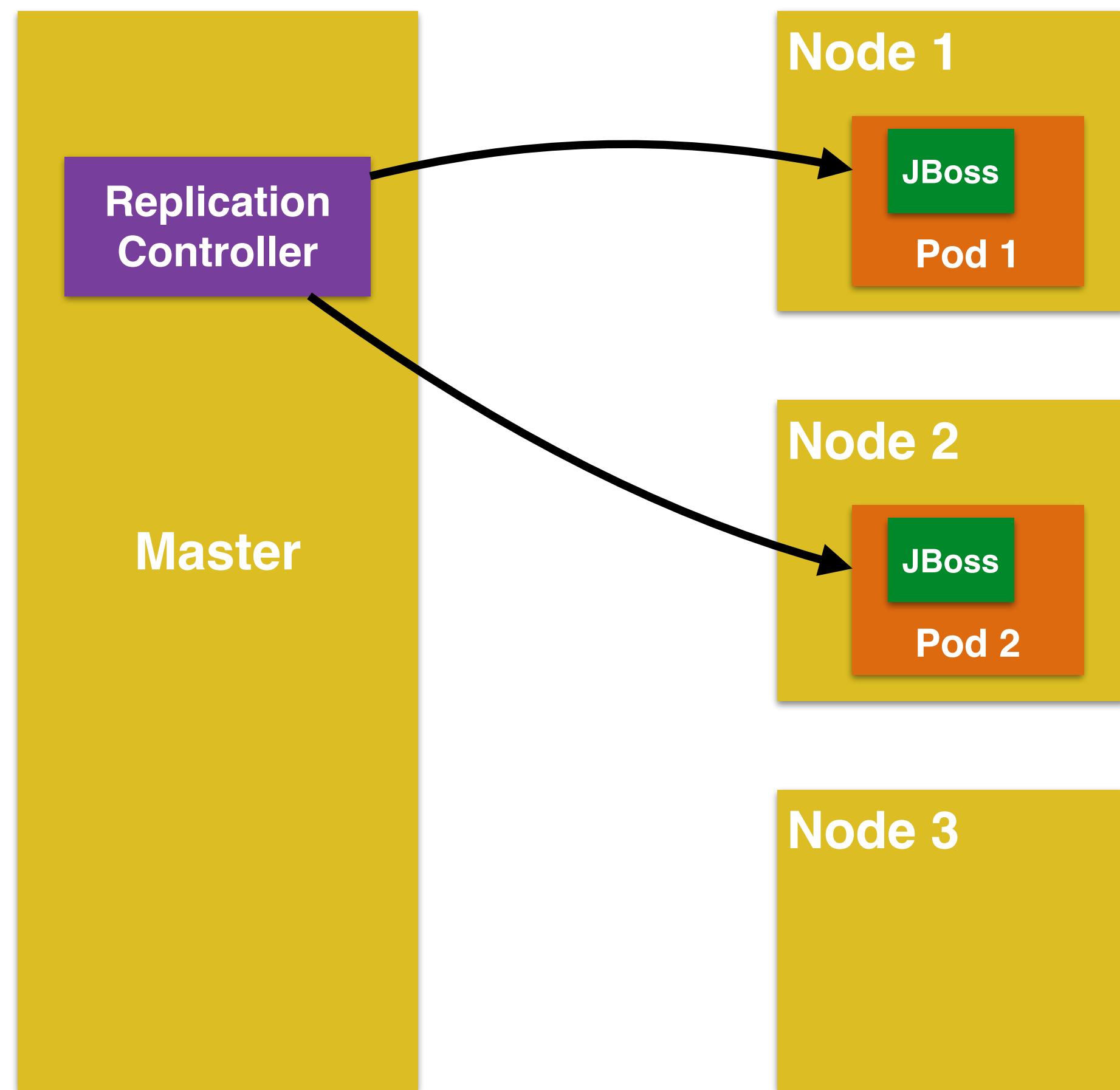
Replication Controller

- Ensures that a specified number of pod "replicas" are running
 - Pod templates are cookie cutters
 - Rescheduling
 - Manual or auto-scale replicas
 - Rolling updates
- Recommended to wrap a Pod or Service in a RC
- Only appropriate for Pods with **Restart=Always** policy (default)

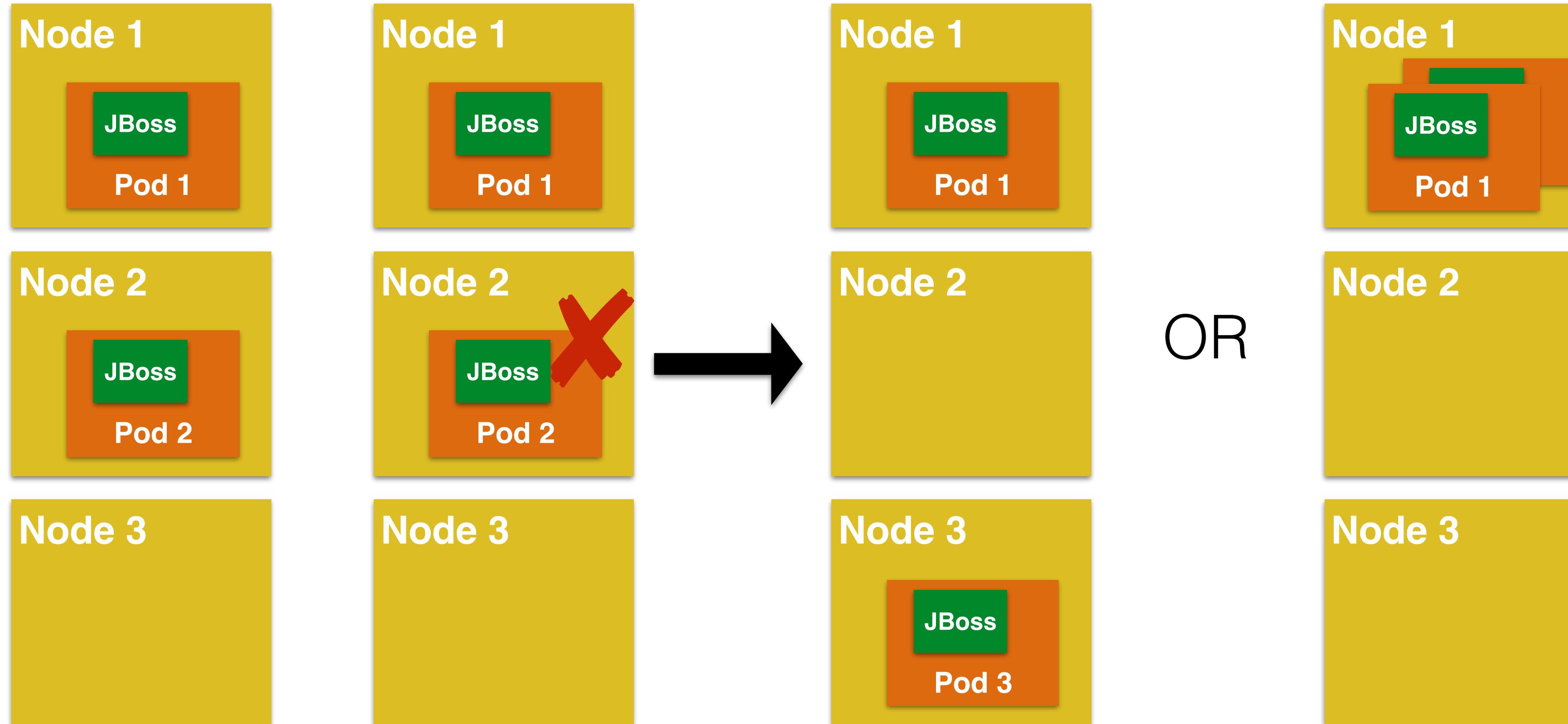
Replication Controller Configuration

```
1  apiVersion: v1
2  kind: ReplicationController
3  metadata:
4    name: wildfly-rc
5    labels:
6      name: wildfly
7      context: docker-k8s-lab
8  spec:
9    replicas: 1
10   template:
11     metadata:
12       labels:
13         name: wildfly
14     spec:
15       containers:
16         - name: wildfly-rc-pod
17           image: arungupta/wildfly-mysql-javaee7:k8s
18           ports:
19             - containerPort: 8080
```

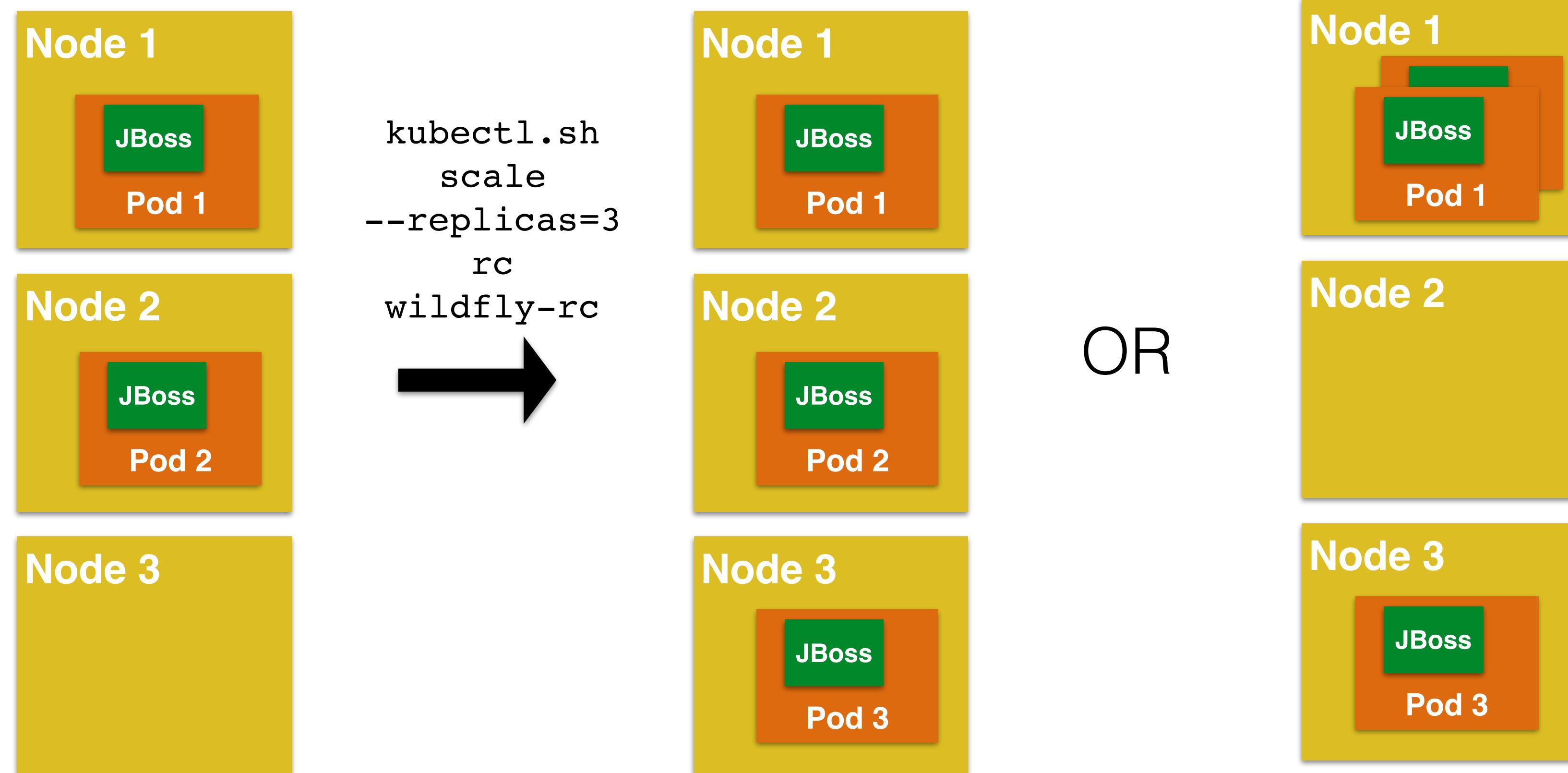
Replication Controller



Replication Controller: Automatic Rescheduling



Replication Controller: Scaling



Sample Production Deployment

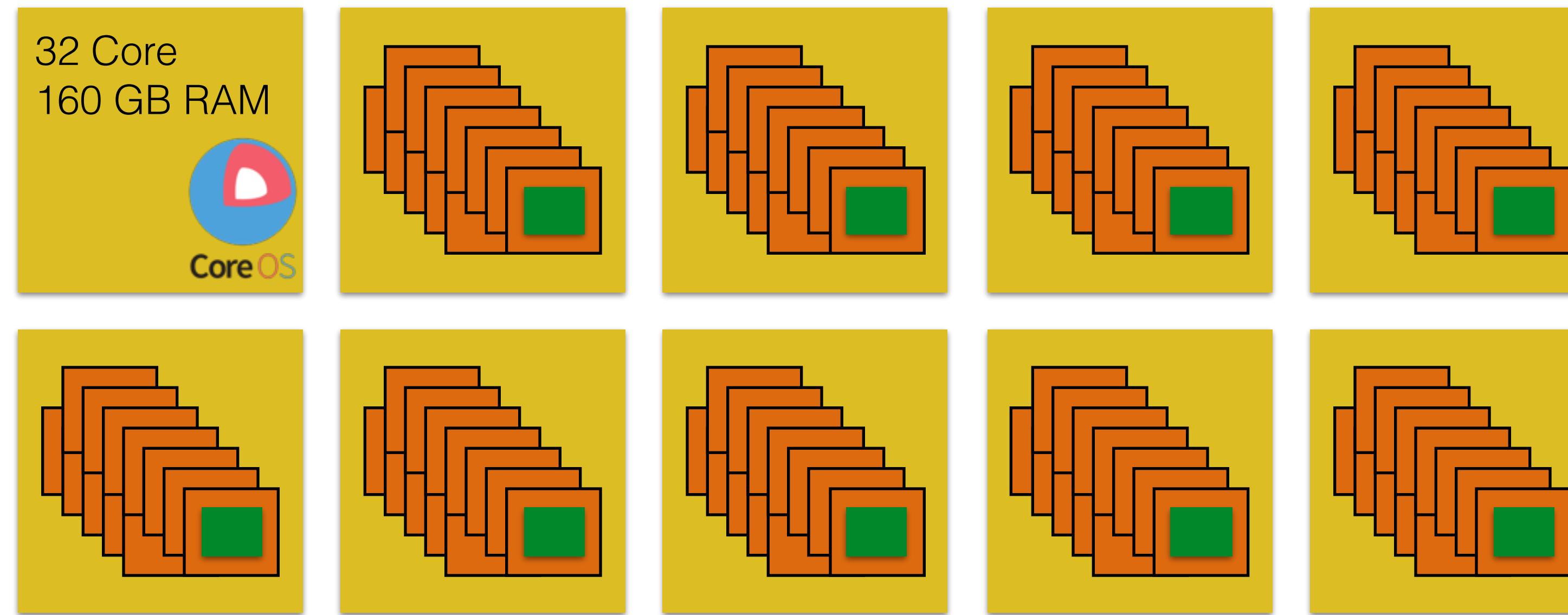
- www.wombatsoftware.de
- shopadvisors.de: E-commerce optimization and monitoring tools for increase of sales



WOMBAT
SOFTWARE



Sample Production Deployment



Load	Containers
Normal	400
Peak	600

Health Checks

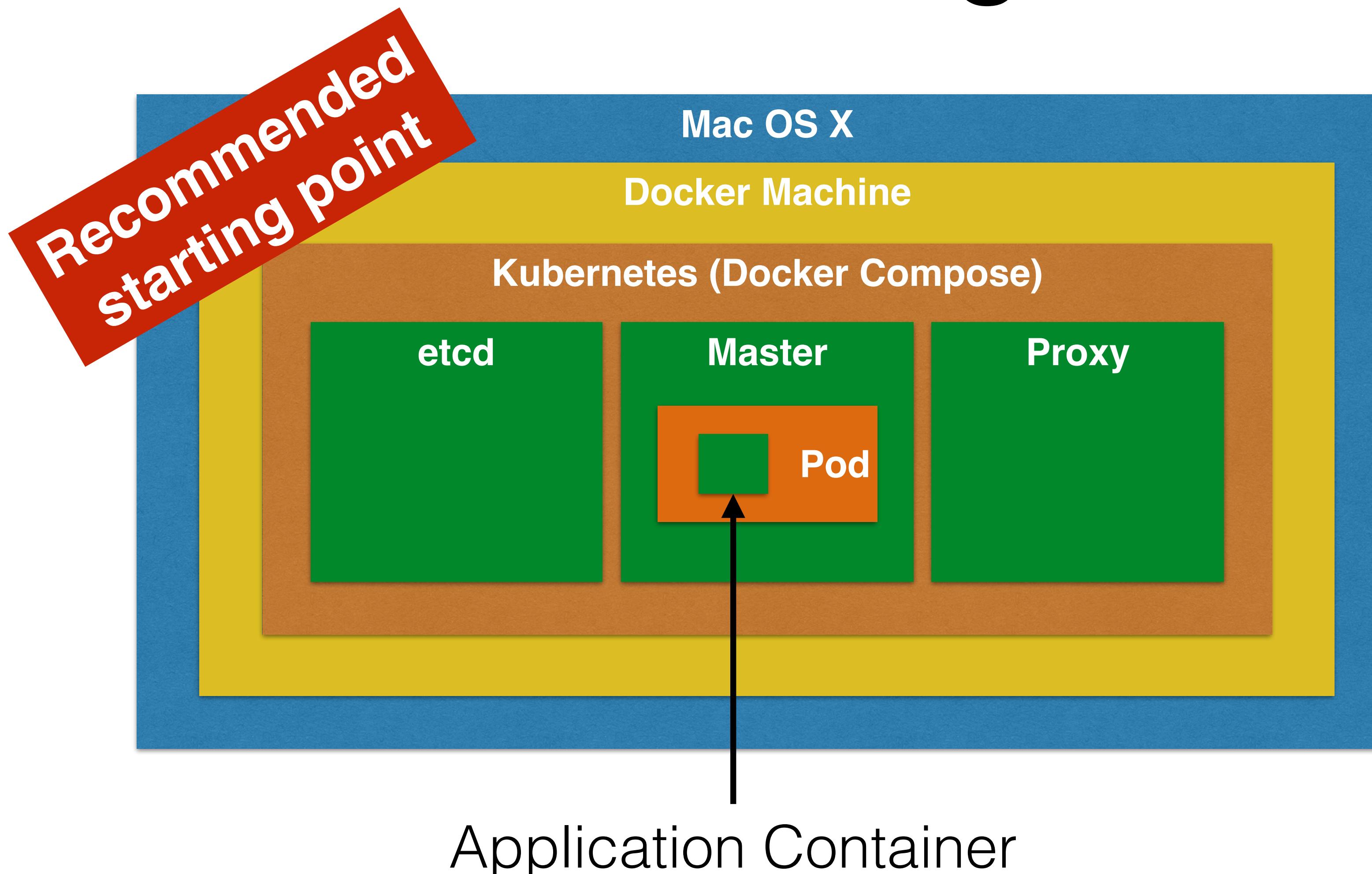
- Restarts Pod, if wrapped in RC
- Application-level health checks
 - HTTP
 - Container Exec
 - TCP Socket
- Health checks performed by Kubelet

Kubernetes using Docker

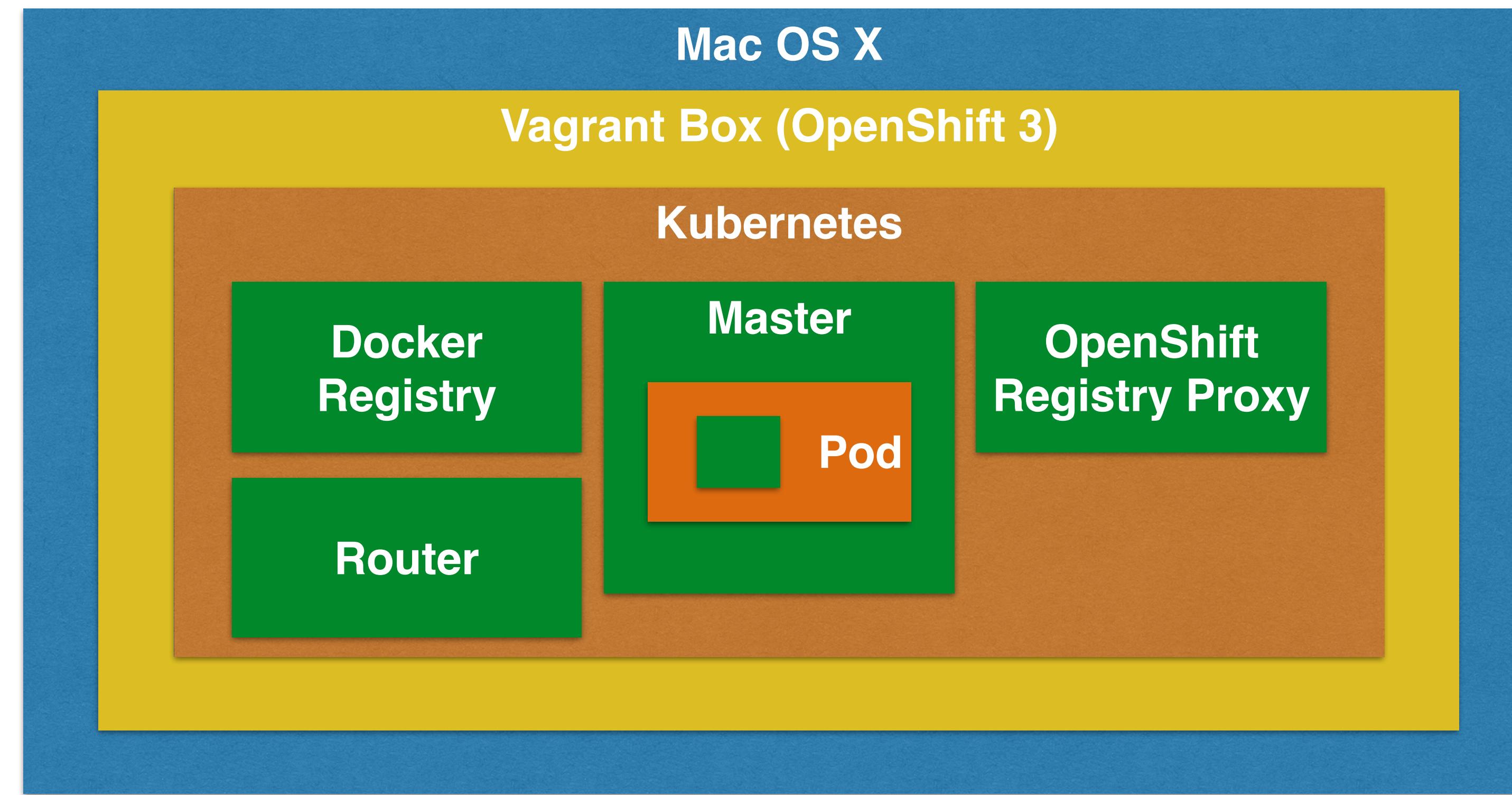
```
1 etcd:
2   image: gcr.io/google_containers/etcd:2.0.9
3   net: "host"
4   entrypoint: /usr/local/bin/etcd --addr=127.0.0.1:4001 --bind-addr=0.0.0.0:4001 --data-dir=/var/lib/etcd
5 master:
6   image: gcr.io/google_containers/hyperkube:v0.21.2
7   net: "host"
8   volumes:
9     - /var/run/docker.sock:/var/run/docker.sock
10  entrypoint: /hyperkube kubelet --api_servers=http://localhost:8080 --v=2 --address=0.0.0.0 --port=8080
11 proxy:
12   image: gcr.io/google_containers/hyperkube:v0.21.2
13   net: "host"
14   privileged: true
15   entrypoint: /hyperkube proxy --master=http://127.0.0.1:8080 --v=2
```

<https://github.com/arun-gupta/docker-images/blob/master/kubernetes/docker-compose.yml>

Kubernetes using Docker



OpenShift 3



References

- github.com/javaee-samples/docker-java
- kubernetes.io/v1.0/docs