Kubernetes Introduction

Arun Gupta
Vice President, Developer Advocacy
@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

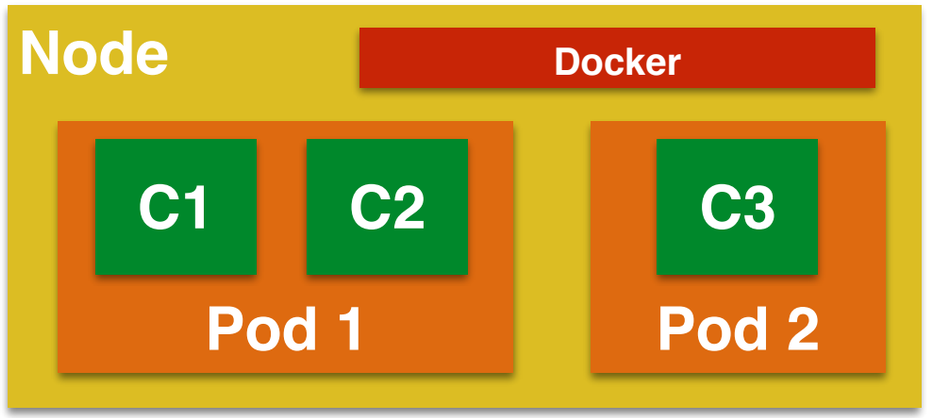
Concepts

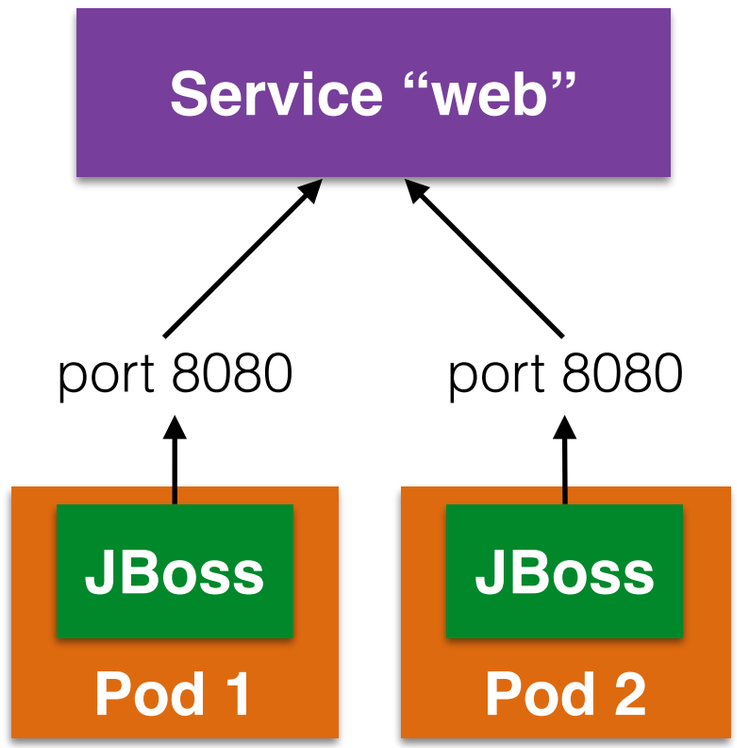
- **Pods**: collocated group of Docker containers that share an IP and storage volume

- **Service**: Single, stable name for a set of pods, also acts as LB

- **Label**: used to organize and select group of objects

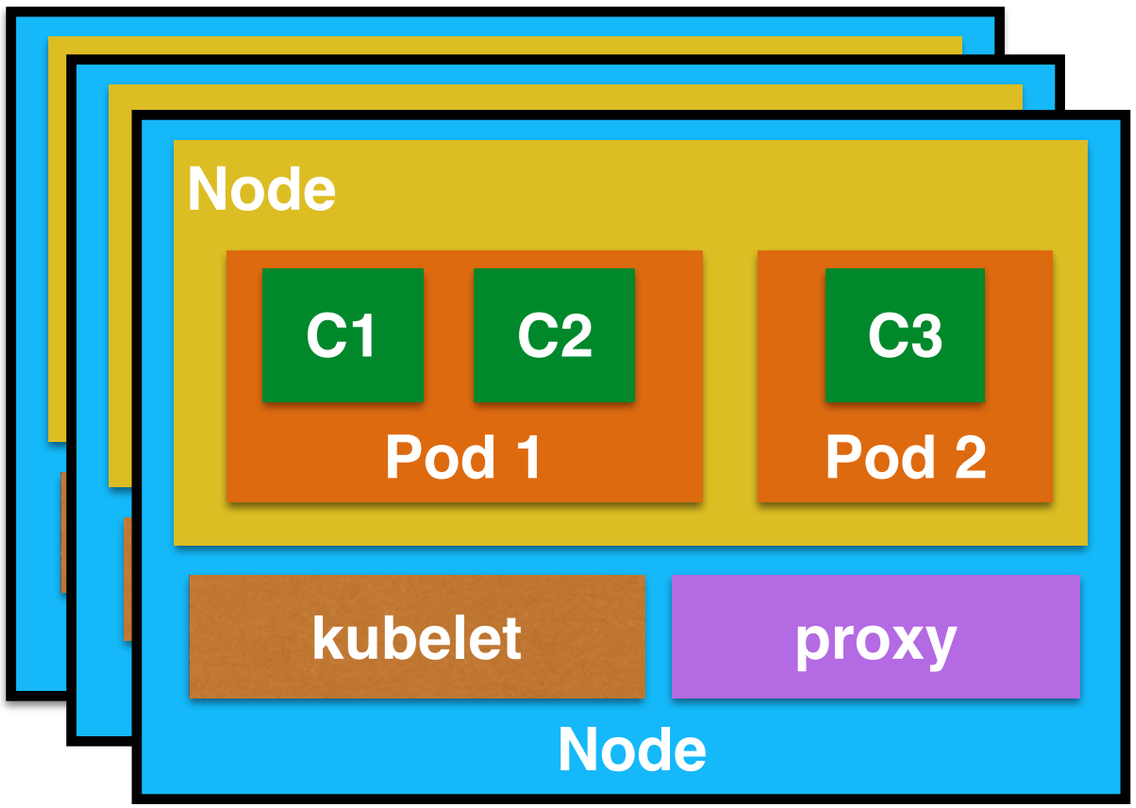
- **Replication Controller**: manages the lifecycle of pods and ensures specified number are running

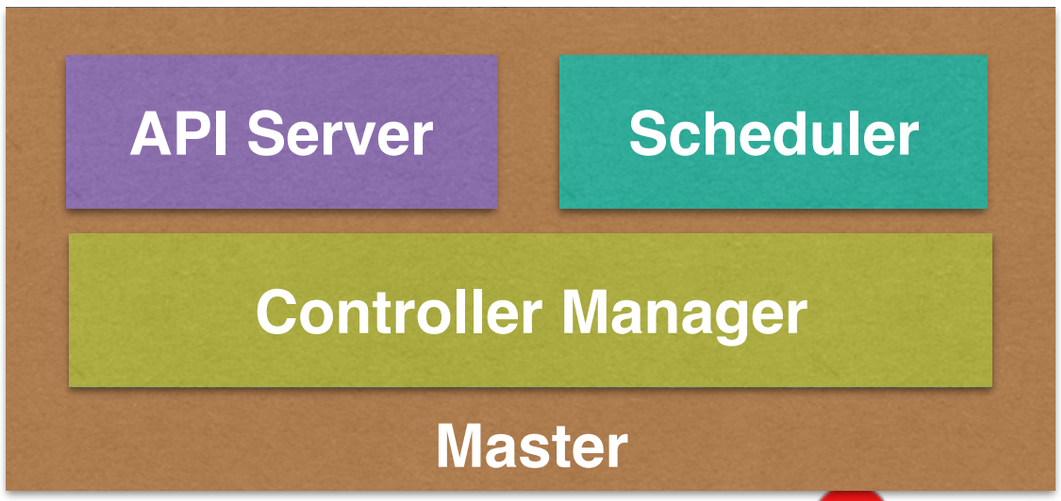




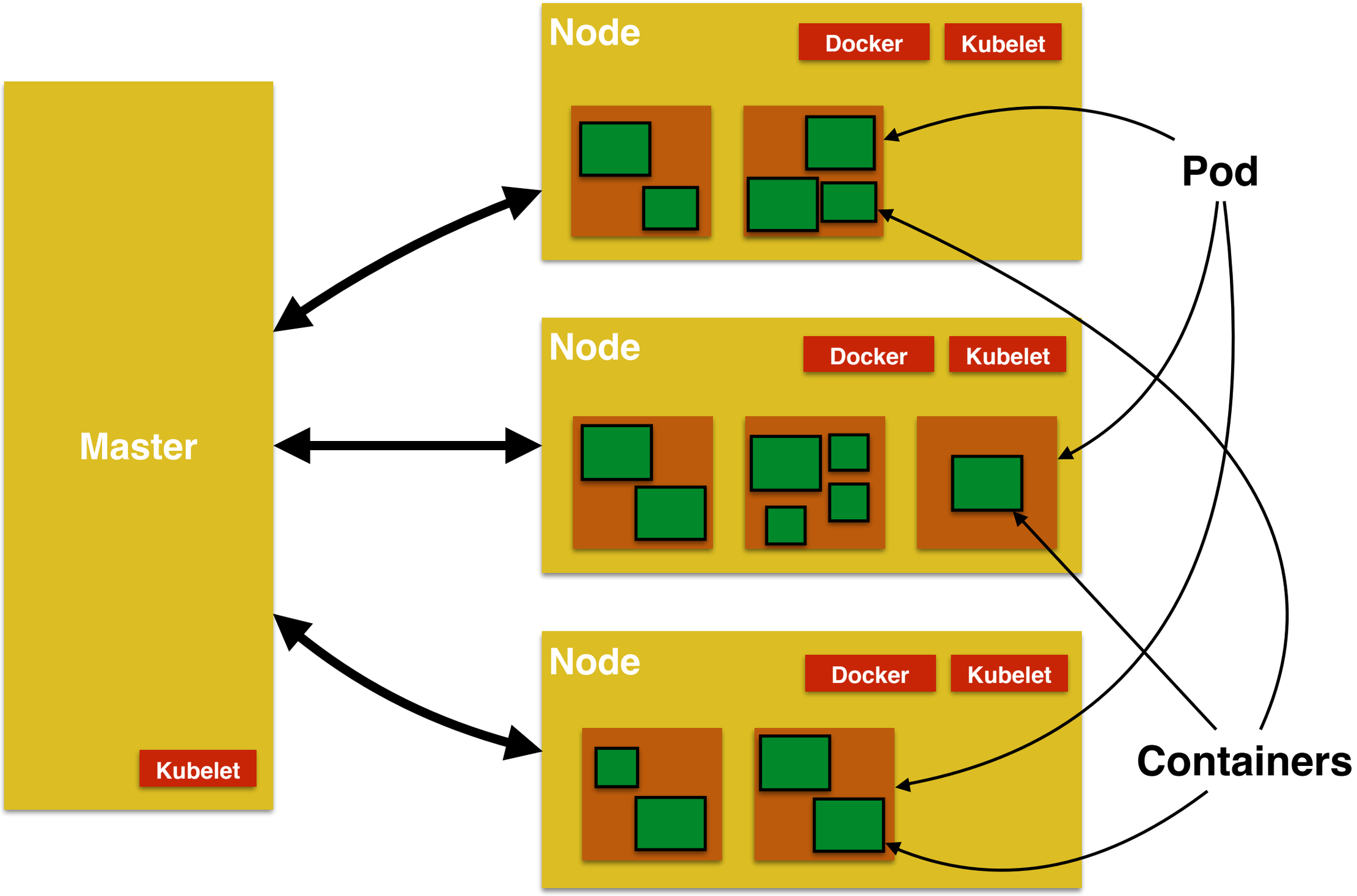
Components

- **Cluster**: compute resources on top of which container are built
- **Node**: Docker host running *kubelet* (node agent) and *proxy* services
- **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



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>`

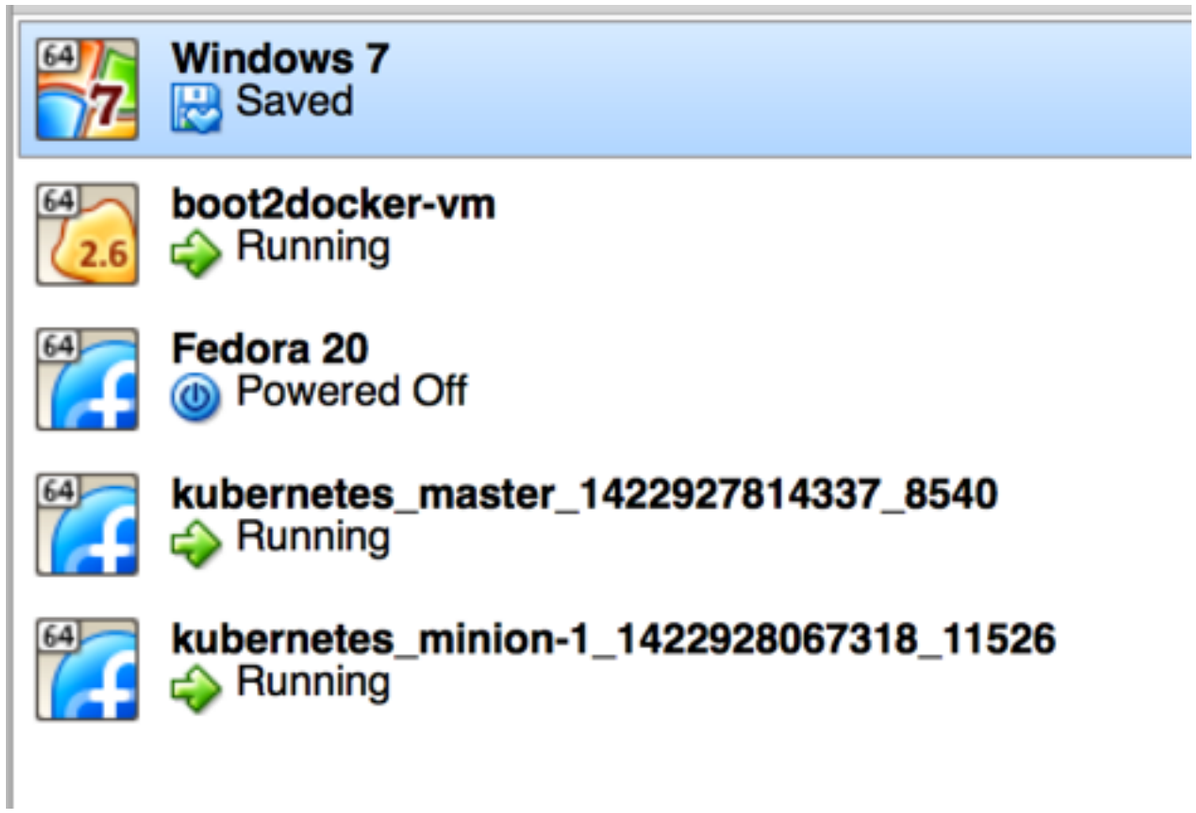
Couchbase

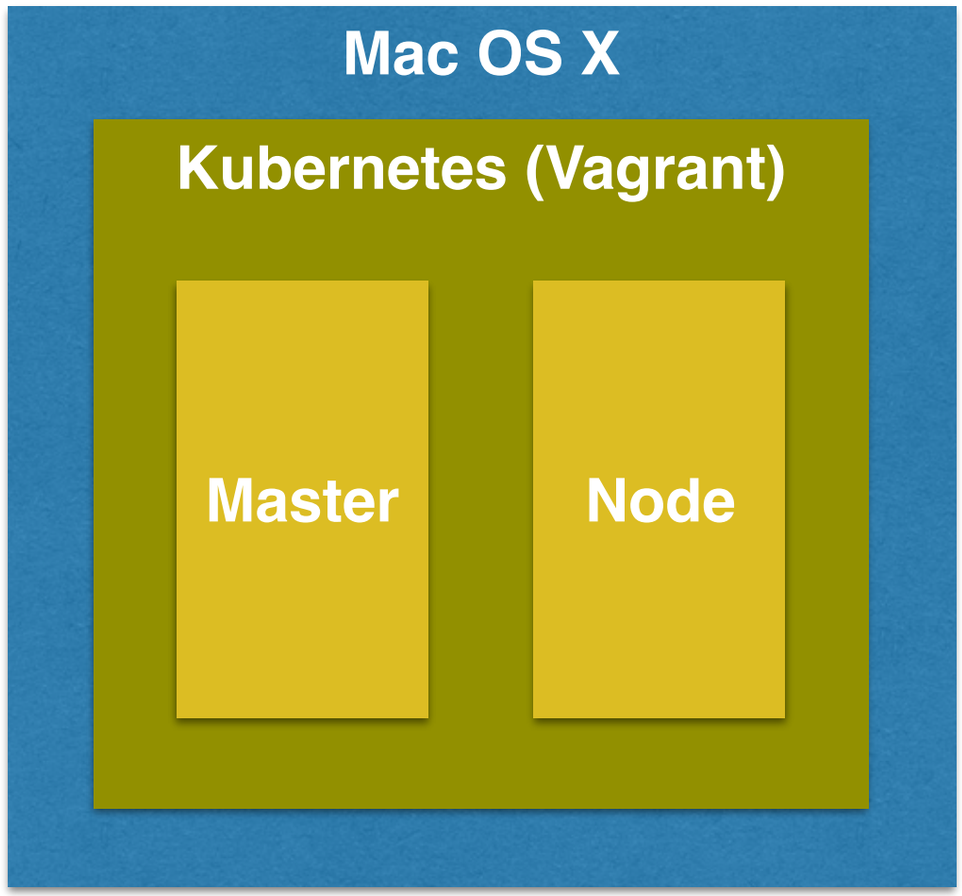
Kubernetes Config

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: wildfly-rc
  labels:
    name: wildfly
spec:
  replicas: 2
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rc-pod
        image: jboss/wildfly
        ports:
        - containerPort: 8080
```

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

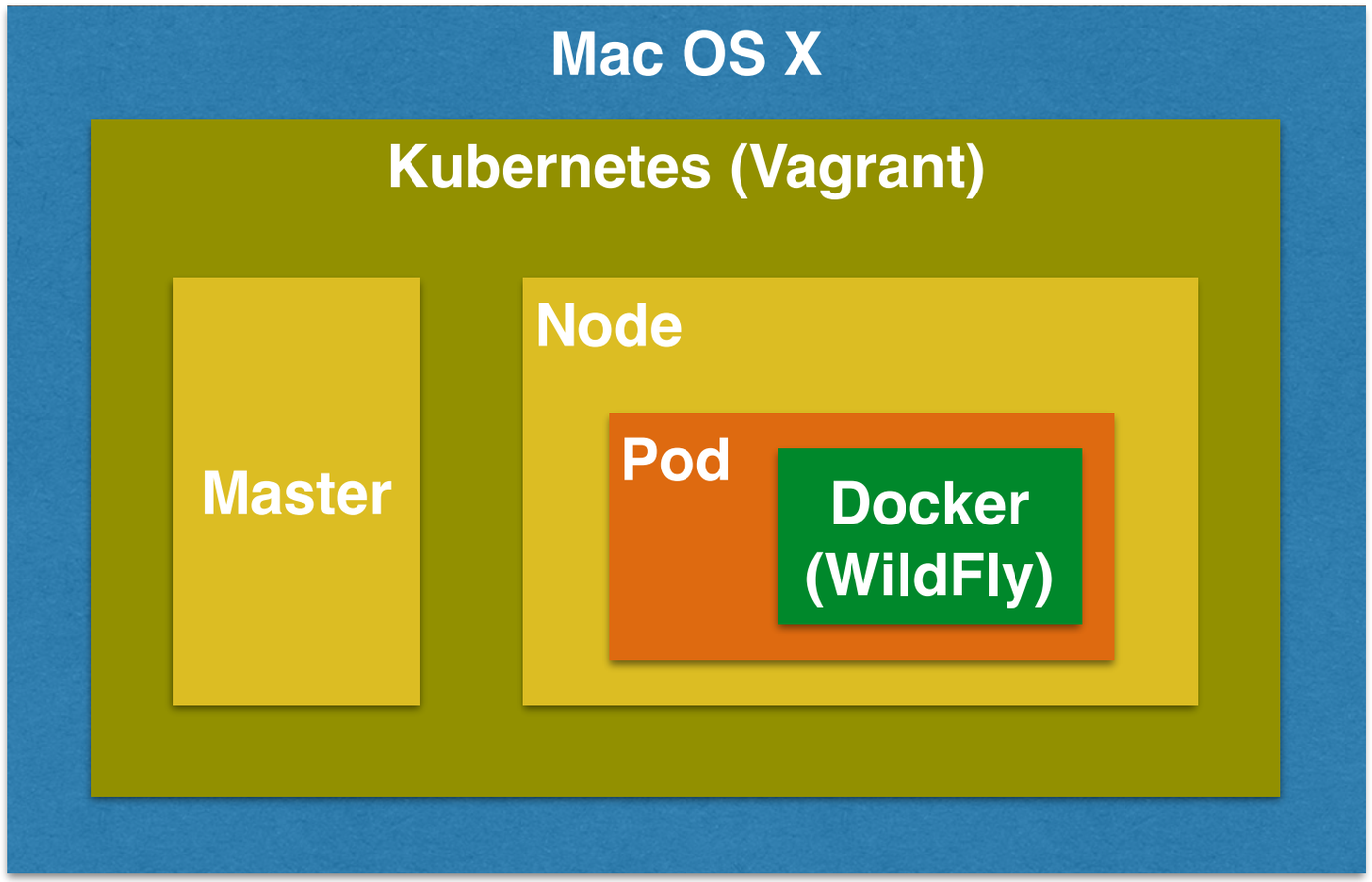




Couchbase

A Pod with One Container

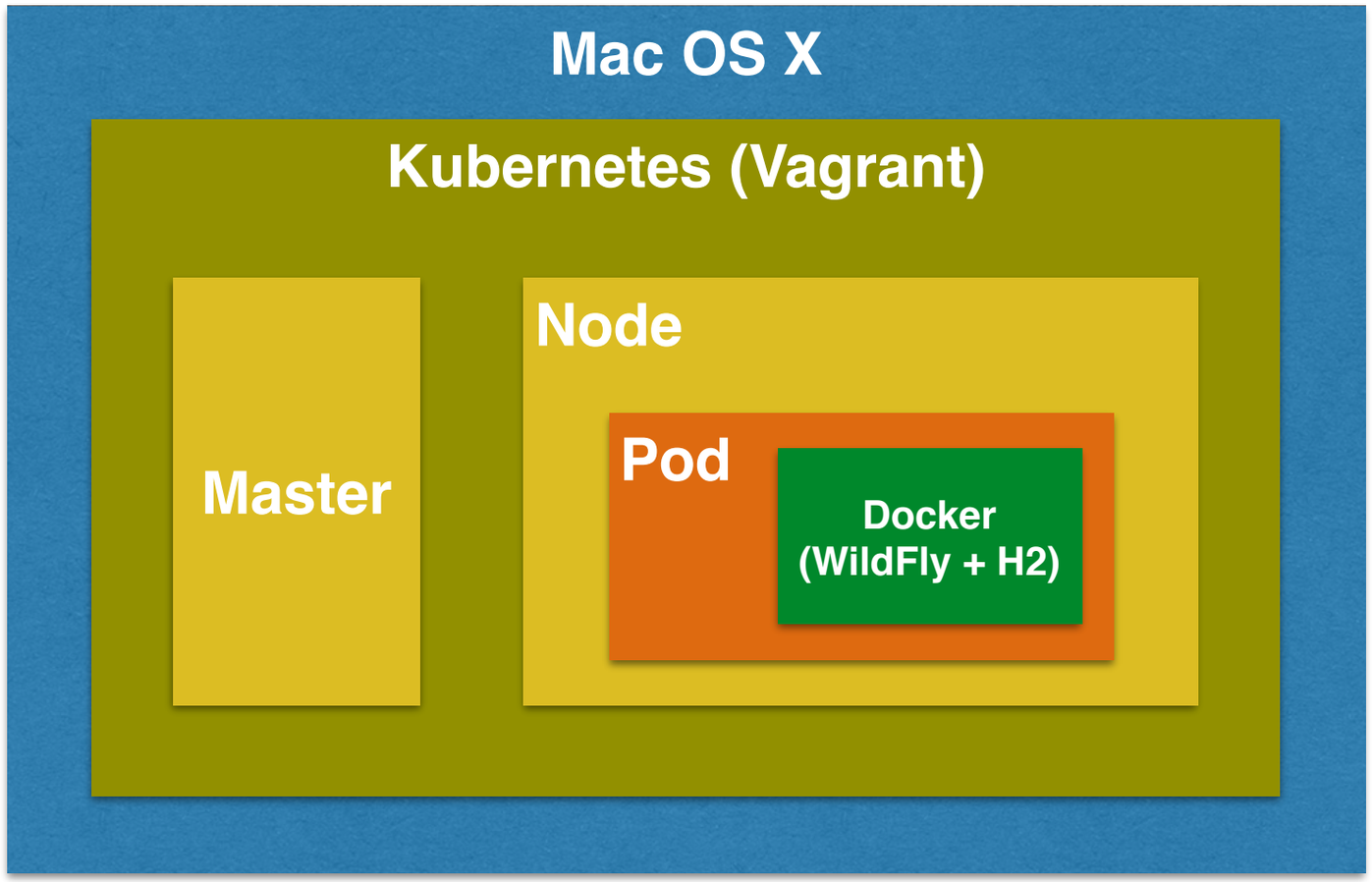
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```



Couchbase

Java EE Application Deployed in a Pod with One Container

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: javaee7-hol
  labels:
    name: javaee7-hol
spec:
  replicas: 1
  selector:
    name: javaee7-hol
  template:
    metadata:
      labels:
        name: javaee7-hol
    spec:
      containers:
      - name: master
        image: arungupta/javaee7-hol
        ports:
        - containerPort: 8080
          hostPort: 8080
```



Couchbase

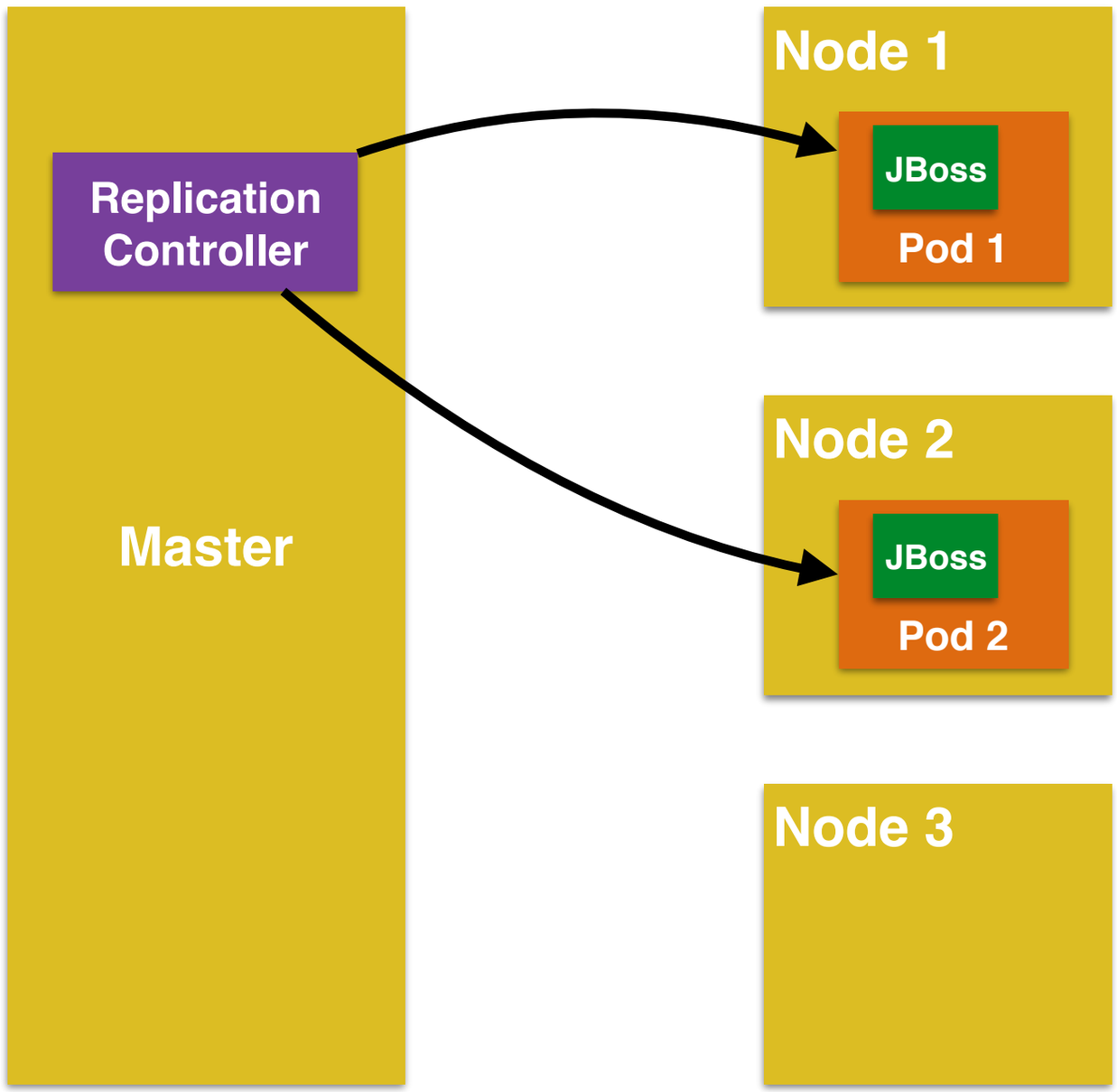
Replication Controller

- Ensures that a specified number of pod "replicas" are running at any one time

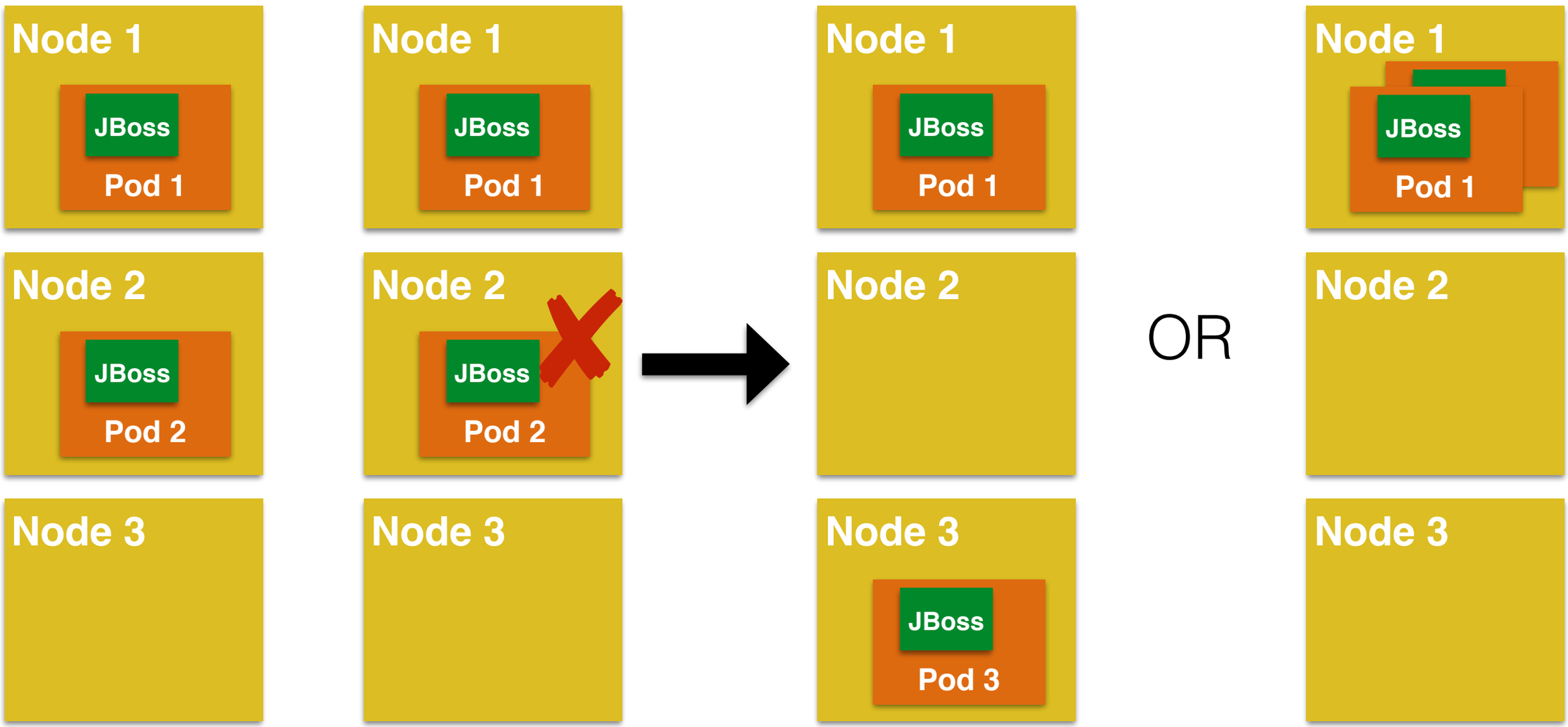
- Recommended to wrap a Pod in a RC

- Only appropriate for Pods with `Restart=Always` policy (default)

Replication Controller

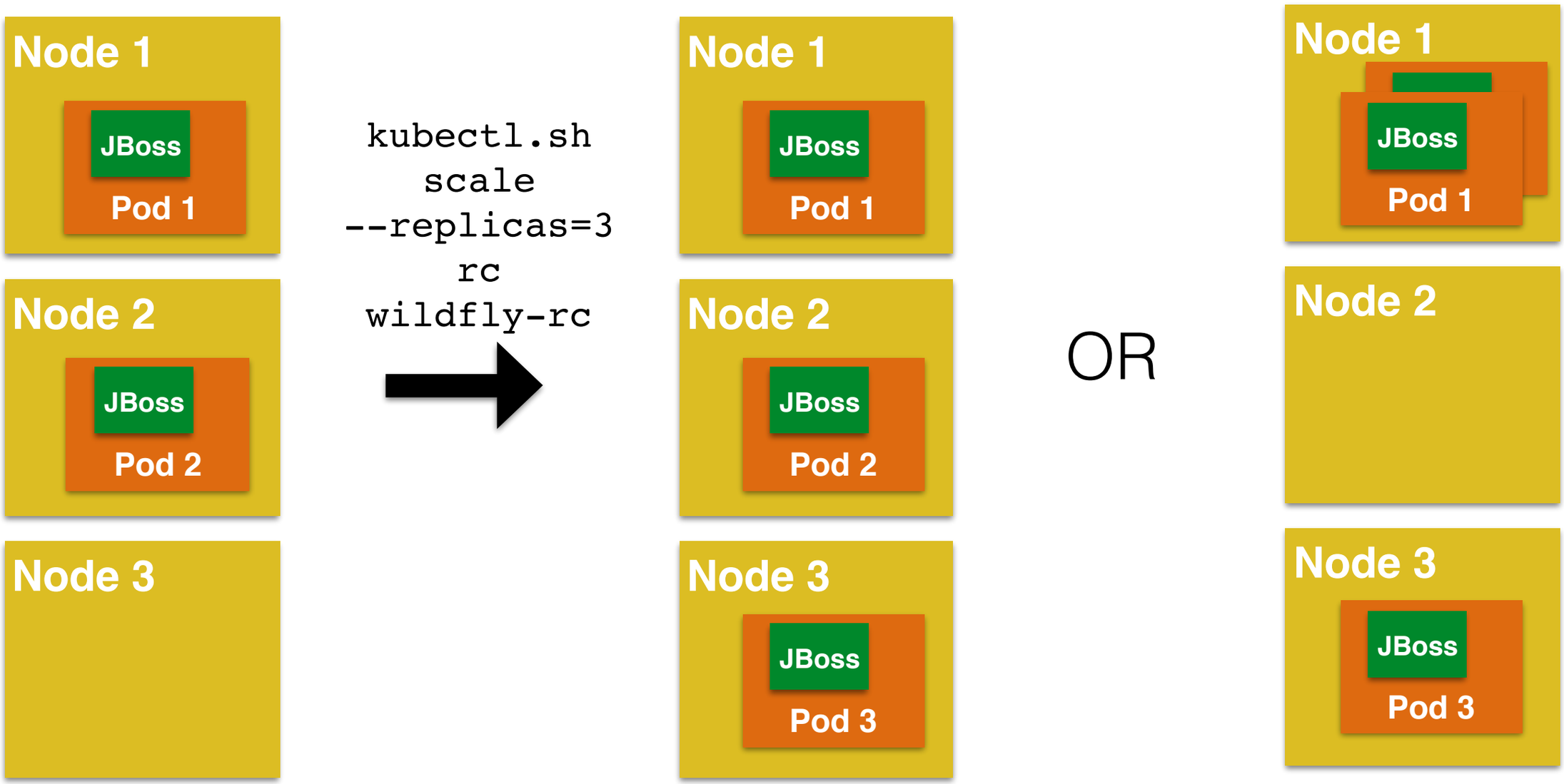


Replication Controller: Automatic Rescheduling



Couchbase

Replication Controller: Scaling



Services

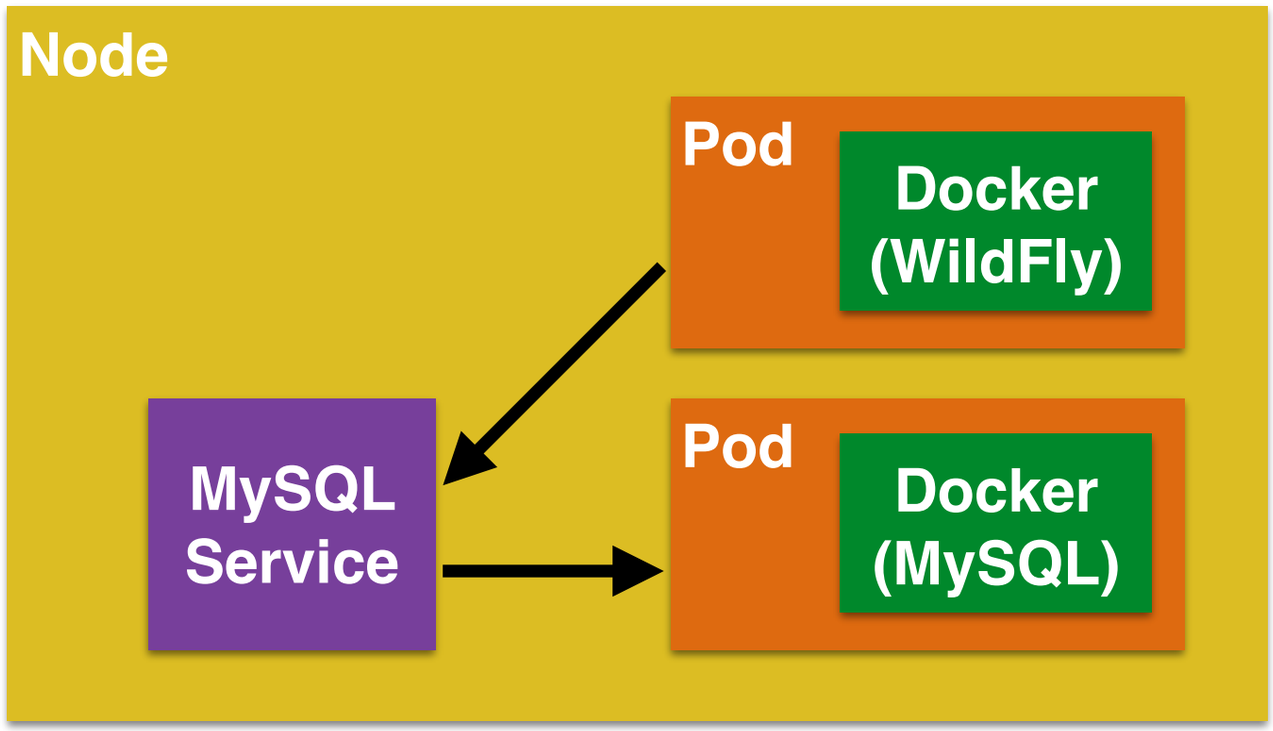
- Abstract a set of pods as a single IP and port
  - Simple TCP/UDP load balancing
- Creates environment variables in other pods
  - Like “Docker links” but across hosts
- Stable endpoint for pods to reference
  - Allows list of pods to change dynamically

kubernetes

Couchbase

Services

```
{
  "id": "mysql",
  "kind": "Pod",
  "apiVersion": "v1beta1",
  "desiredState": {
    "manifest": {
      "version": "v1beta1",
      "id": "mysql",
      "containers": [{
        "name": "mysql",
        "image": "mysql:latest",
        "cpu": 100,
        "env": [
              {
                  "name": "MYSQL_USER",
                  "value": "mysql"
              },
              {
                  "name": "MYSQL_PASSWORD",
                  "value": "mysql"
              },
              {
                  "name": "MYSQL_DATABASE",
                  "value": "sample"
              },
              {
                  "name": "MYSQL_ROOT_PASSWORD",
                  "value": "supersecret"
              }
          ],
        "ports": [{
          "containerPort": 3306,
          "hostPort": 3306
        }]
      }]
    }
  },
  "labels": {
    "name": "mysql"
  }
}
```

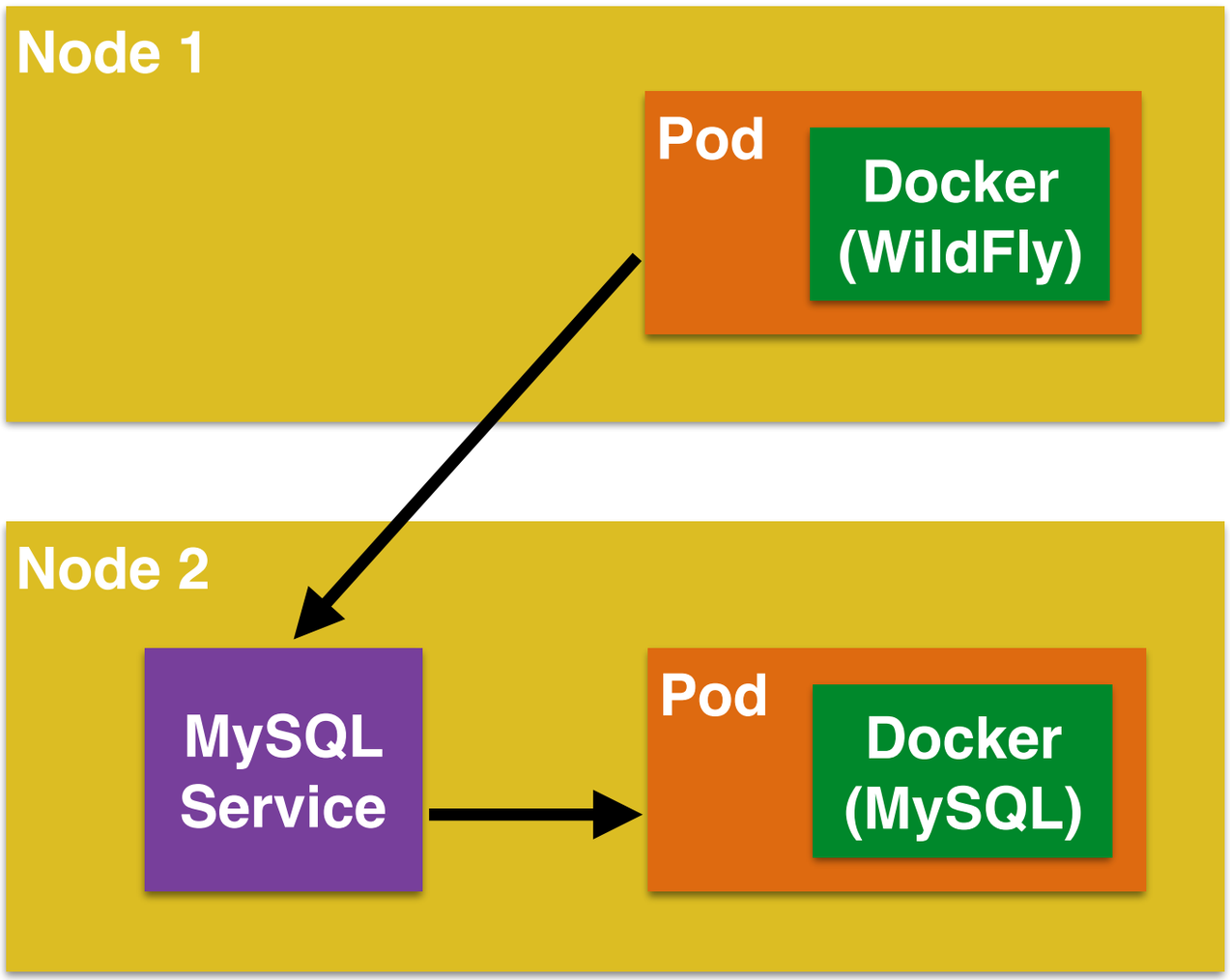



```
{
  "id": "mysql",
  "kind": "Service",
  "apiVersion": "v1beta1",
  "port": 3306,
  "containerPort": 3306,
  "selector": {
    "name": "mysql"
  },
  "labels": {
    "name": "mysql"
  }
}
```

http://blog.arungupta.me/mysql-kubernetes-service-access-wildfly-pod-techtip72/

Couchbase

Two Nodes

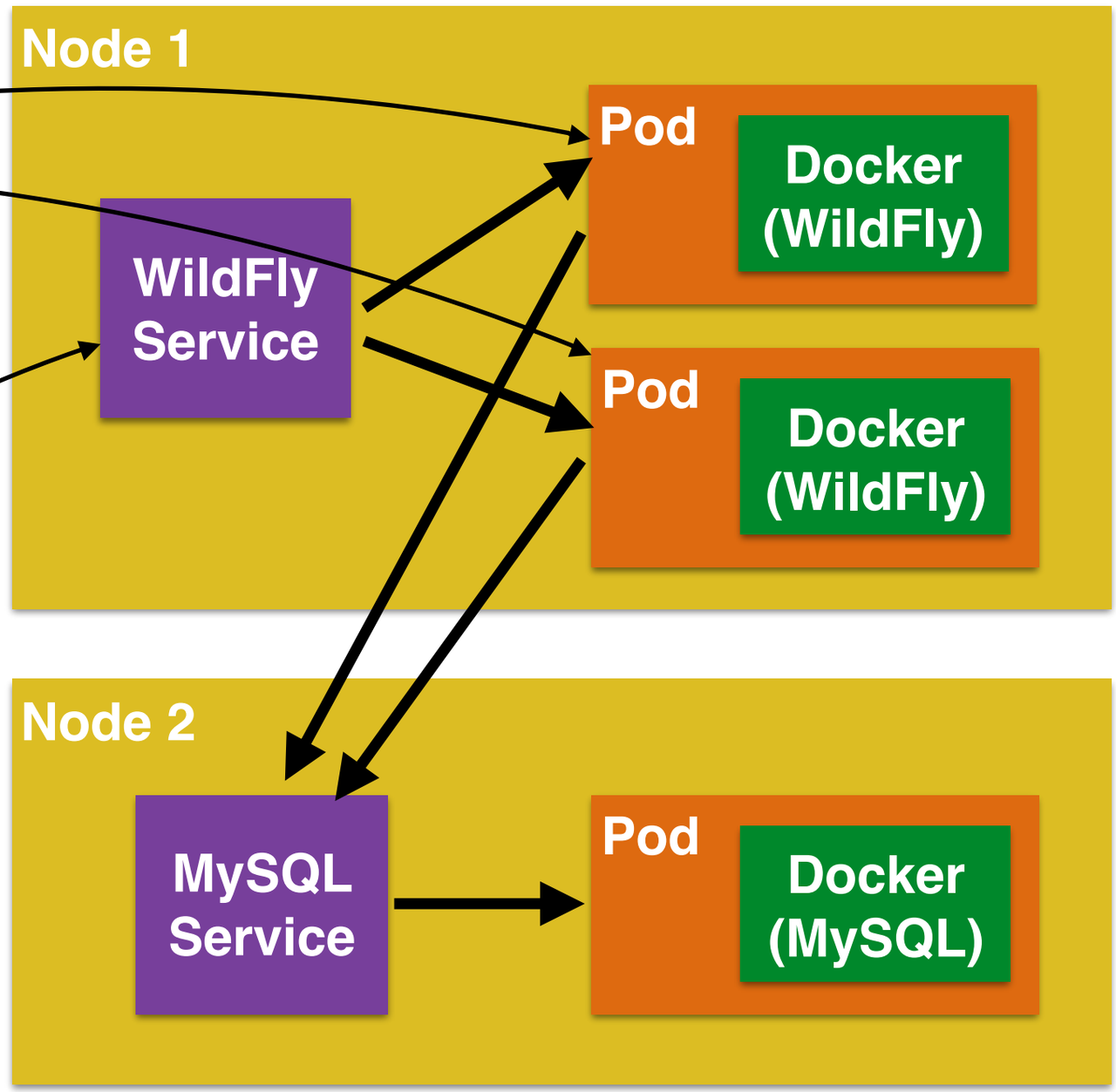

Node 1
Pod
Docker
(WildFly)
Node 2
MySQL
Service
Pod
Docker
(MySQL)

Replication Controller

- Ensures specified number of pod “replicas” are running
- Pod templates are cookie cutters
- Rescheduling
- Manual or auto-scale replicas
- Rolling updates

```yaml
id: wildfly-controller
kind: ReplicationController
apiVersion: v1beta1
desiredStat
  replica : 2
  replicaSe tor:
    name: wildfly
  podTemplate:
    desiredState:
      manifest:
        version: v1beta1
        id: wildfly
        containers:
          - name: javaee7-hol
            image: arungupta/javaee7-hol
    labels:
      name: wildfly
```

```yaml
id: wildfly
ki d: Service
apiVe beta1
port: 8080
containerPort: 8080
selector:
  name: wildfly
labels:
  name: wildfly
```



Health Checks

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