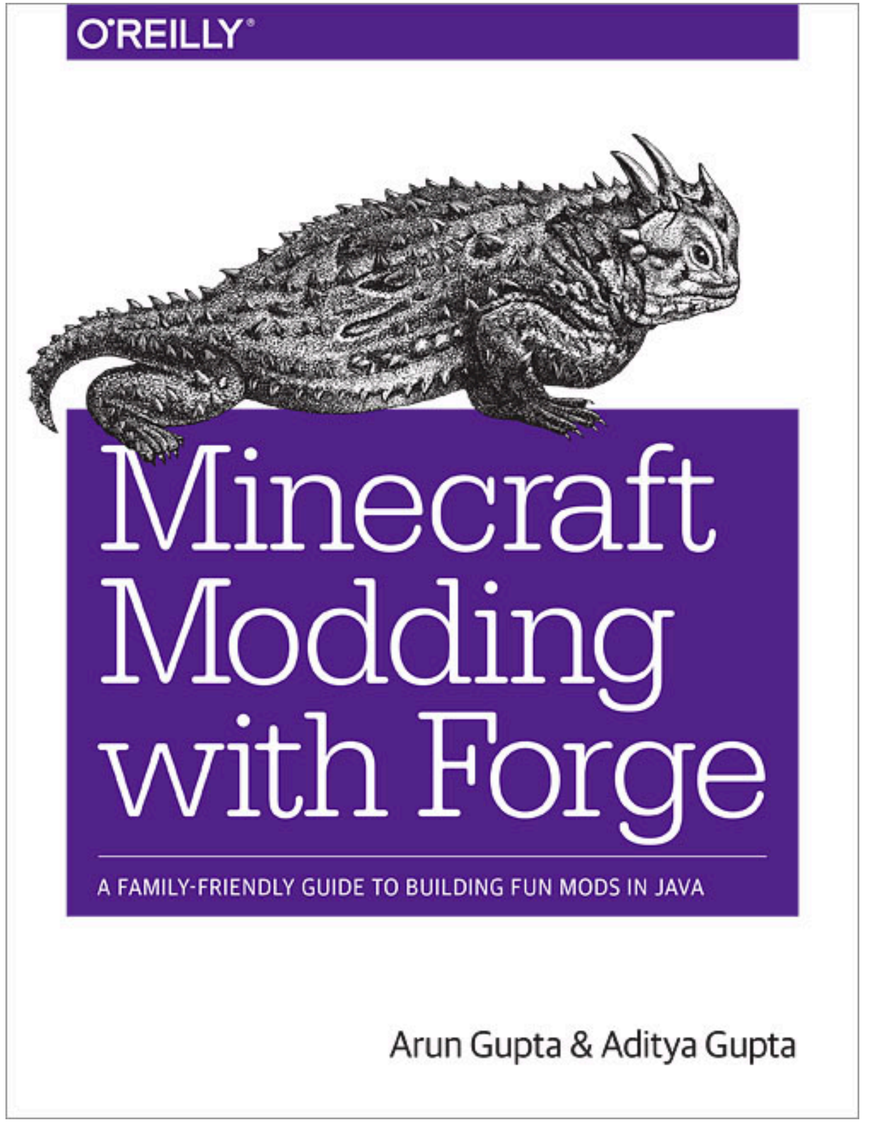
Getting Started with Kubernetes

Arun Gupta, @arungupta
VP Developer Advocacy, Couchbase

Couchbase

DEVOXX™ 4KIDS USA



233

DZone REFCARDZ

BROUGHT TO YOU IN PARTNERSHIP WITH MESOSPHERE

Get More Refcardz! Visit DZone.com/Refcardz

CONTENTS

Kubernetes

BY ARUN GUPTA

WHAT IS KUBERNETES?

Kubernetes (kubernetes.io) is an open-source orchestration system for managing containerized applications across multiple hosts, providing basic mechanisms for the deployment, maintenance, and scaling of applications.

Kubernetes, or "k8s" or "kube" for short, allows the user to declaratively specify the desired state of a cluster using high-level primitives. For example, the user may specify that they want three instances of the Couchbase server container running. Kubernetes' self-healing mechanisms, such as auto-restarting, re-scheduling, and replicating containers then converge the actual state towards the desired state.

Kubernetes supports Docker and Rocket containers. An abstraction around the containerization layer will allow for other container image formats and runtimes to be supported in the future.

KEY CONCEPTS OF KUBERNETES

POD

A Pod is the smallest deployable unit that can be created, scheduled, and managed. It's a logical collection of containers that belong to an application.

Each resource in Kubernetes is defined using a configuration file. For example, a Couchbase pod can be defined with the following `.yaml` file:

A Replication Controller creating two instances of a Couchbase pod can be defined as:

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: couchbase-controller
spec:
  # Two replicas of the Pod to be created
  replicas: 2
  # Identifies the label key and value on the Pod that
  # this Replication Controller is responsible for
  managing
  selector:
    app: couchbase-rc-pod
  # 'cookie cutter' used for creating new pods when
  necessary
  template:
    metadata:
      labels:
        # label key and value on the pod.
        # These must match the selector above.
        app: couchbase-rc-pod
    spec:
      containers:
      - name: couchbase
        image: couchbase
        ports:
        - containerPort: 8091
```

SERVICE

Each Pod is assigned a unique IP address. If the Pod is inside a Replication Controller, then the pod is recreated but may be given

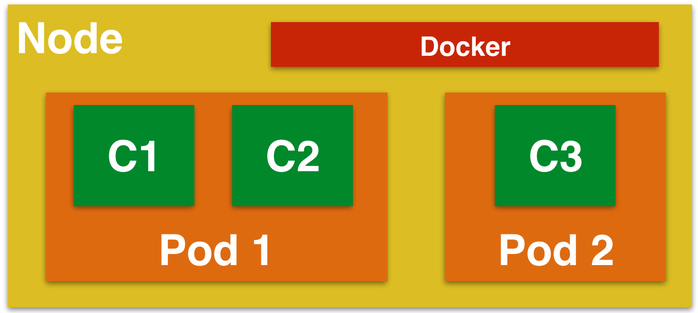
dzone.com/refcardz/kubernetes-essentials

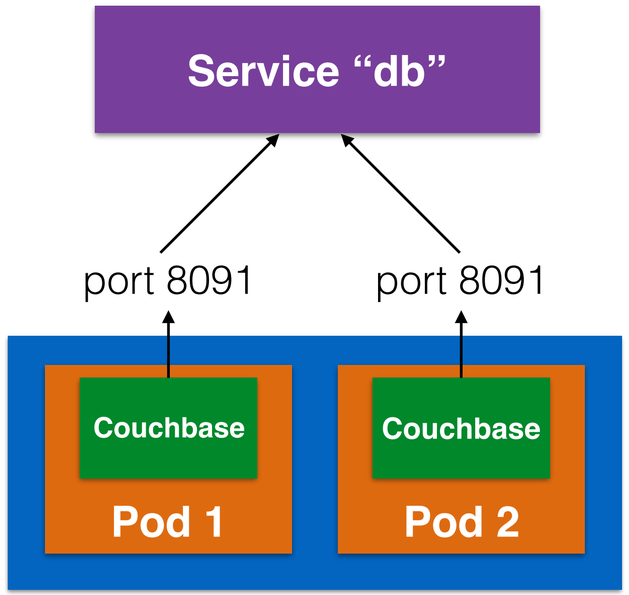
Kubernetes

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

Concepts

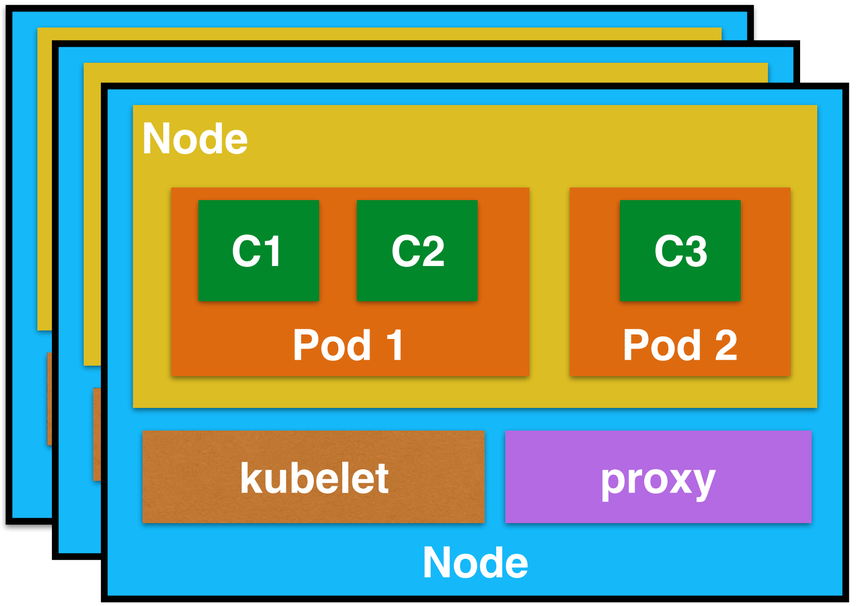
- **Pods**: collocated group of Docker containers that share an IP and storage volume
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to organize and select group of objects
- **Replication Controller**: manages the lifecycle of pods and ensures specified number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

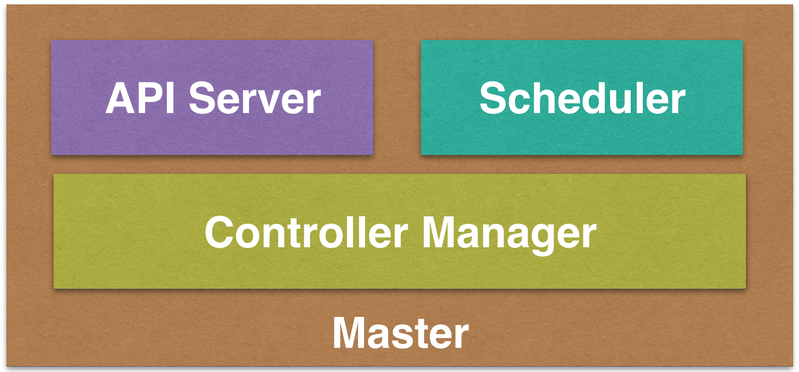




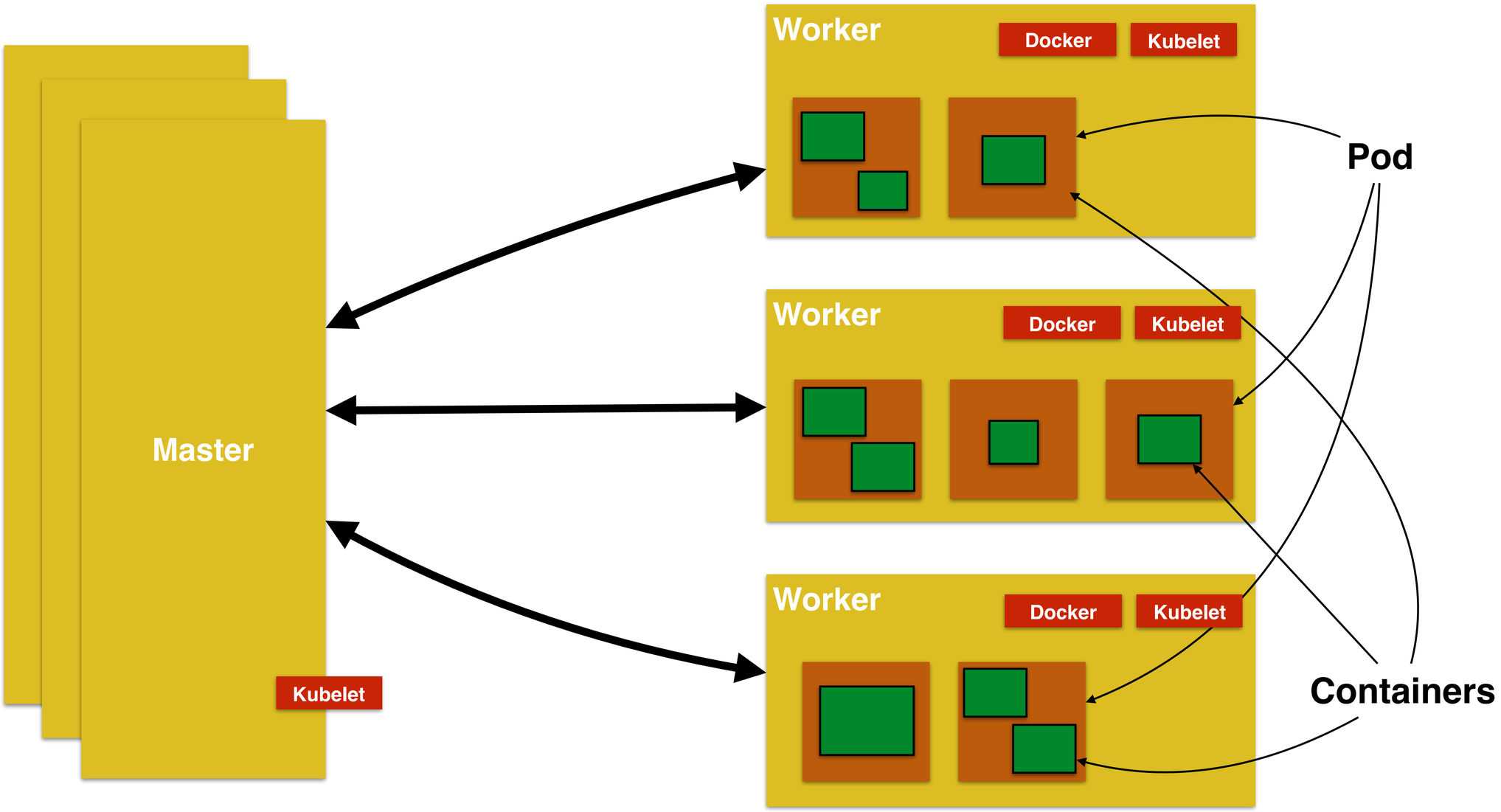
Components

- **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)

http://kubernetes.io/v1.1/docs/proposals/high-availability.html

kubectl

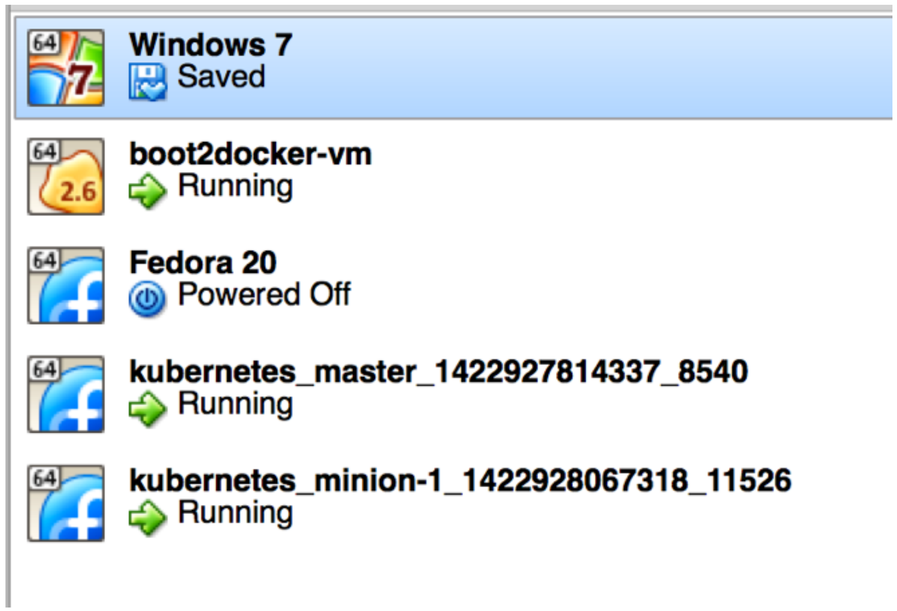
- Controls the Kubernetes cluster manager
- `kubectl get pods` or `minions`
- `kubectl create -f <filename>`
- `kubectl update` or `delete`
- `kubectl resize –replicas=3 replicationcontrollers <name>`

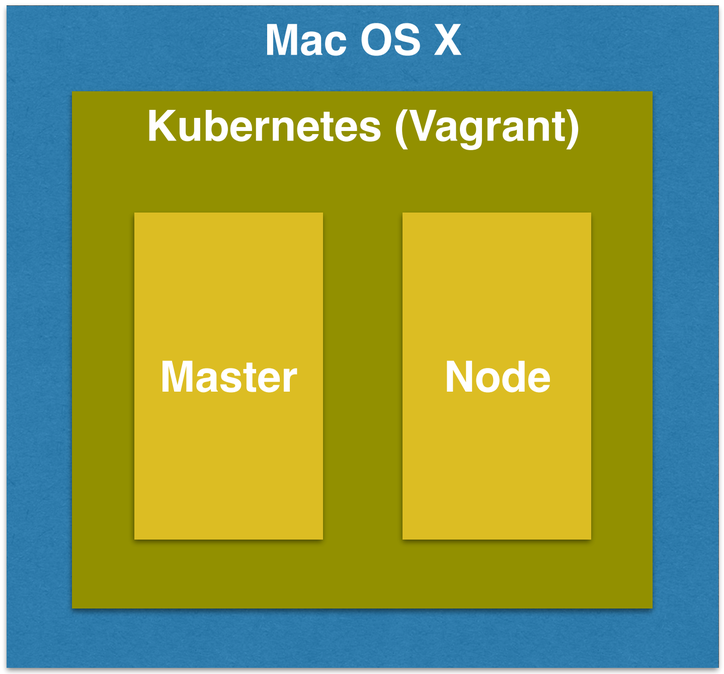
Kubernetes Config

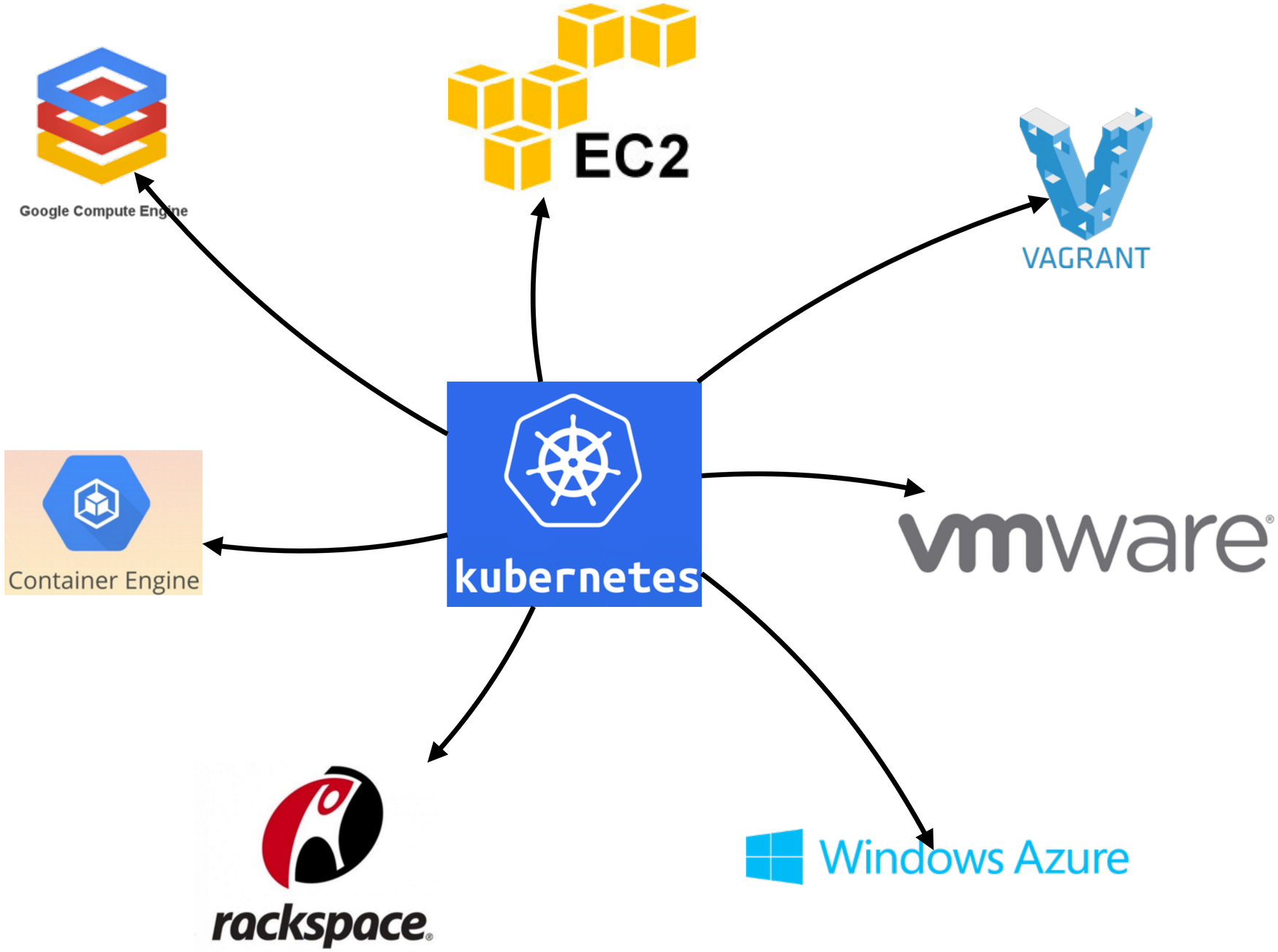
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```

```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
```



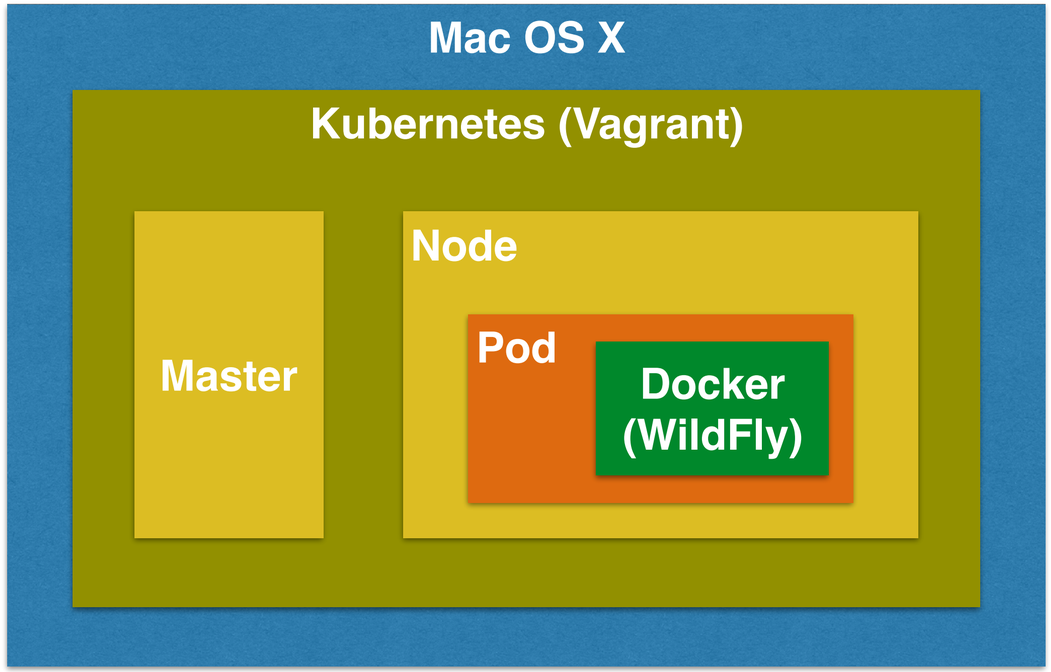




https://get.k8s.io/

A Pod with One Container

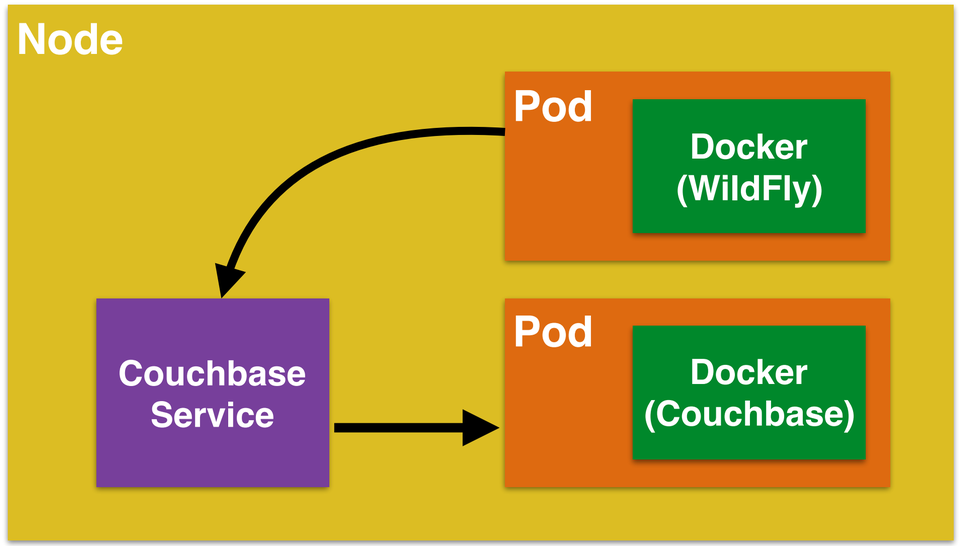
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```



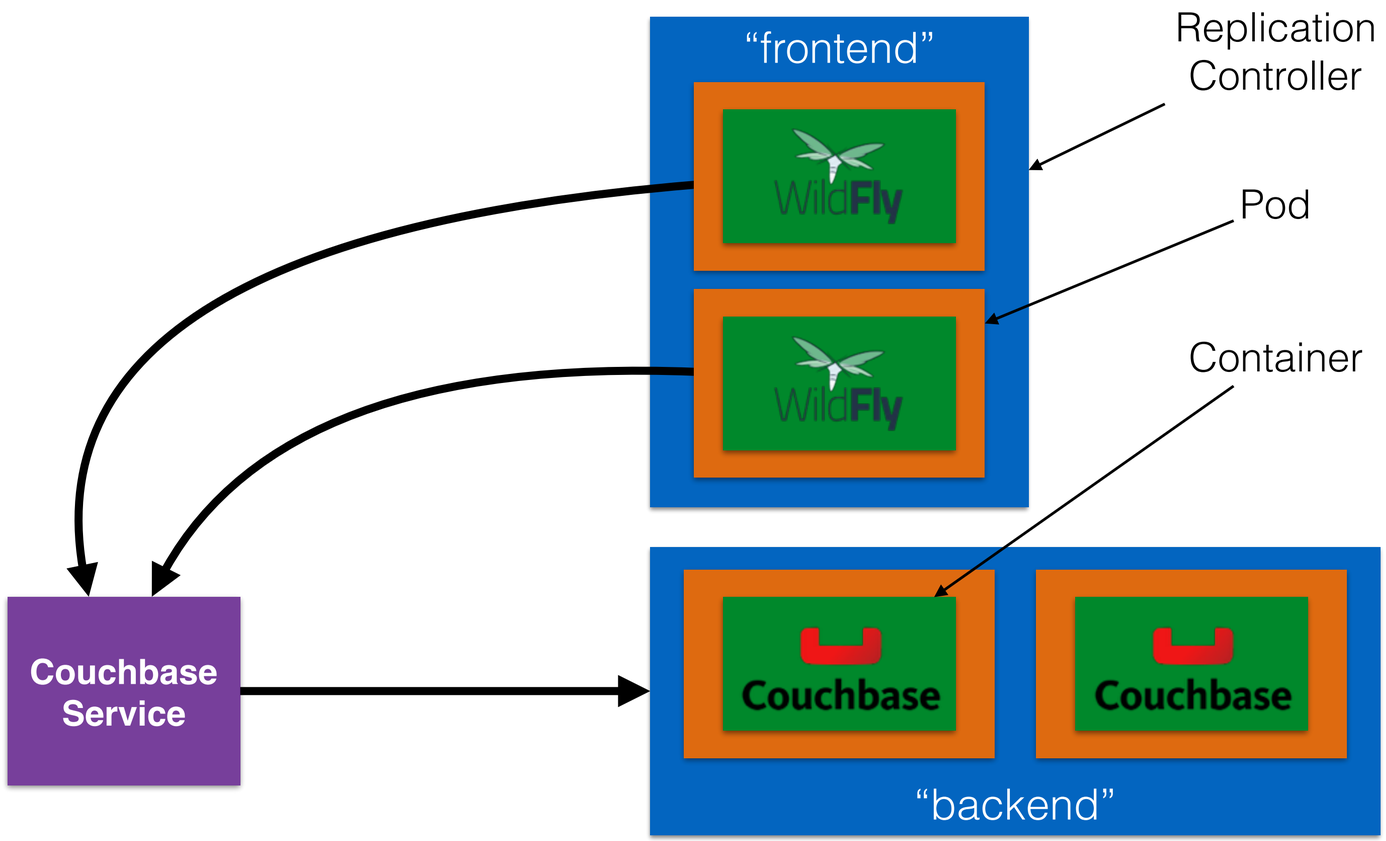
Services

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

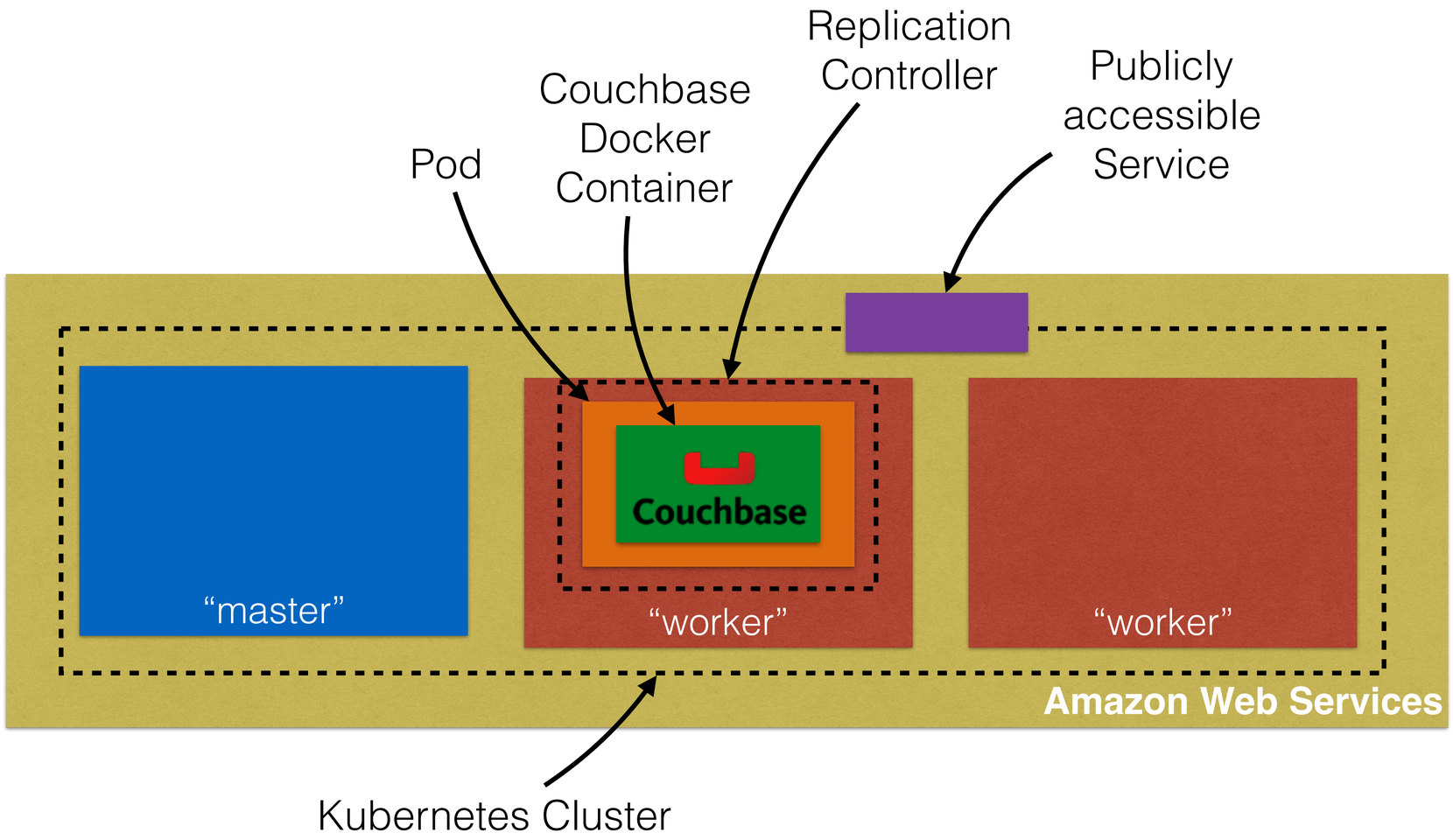
Services



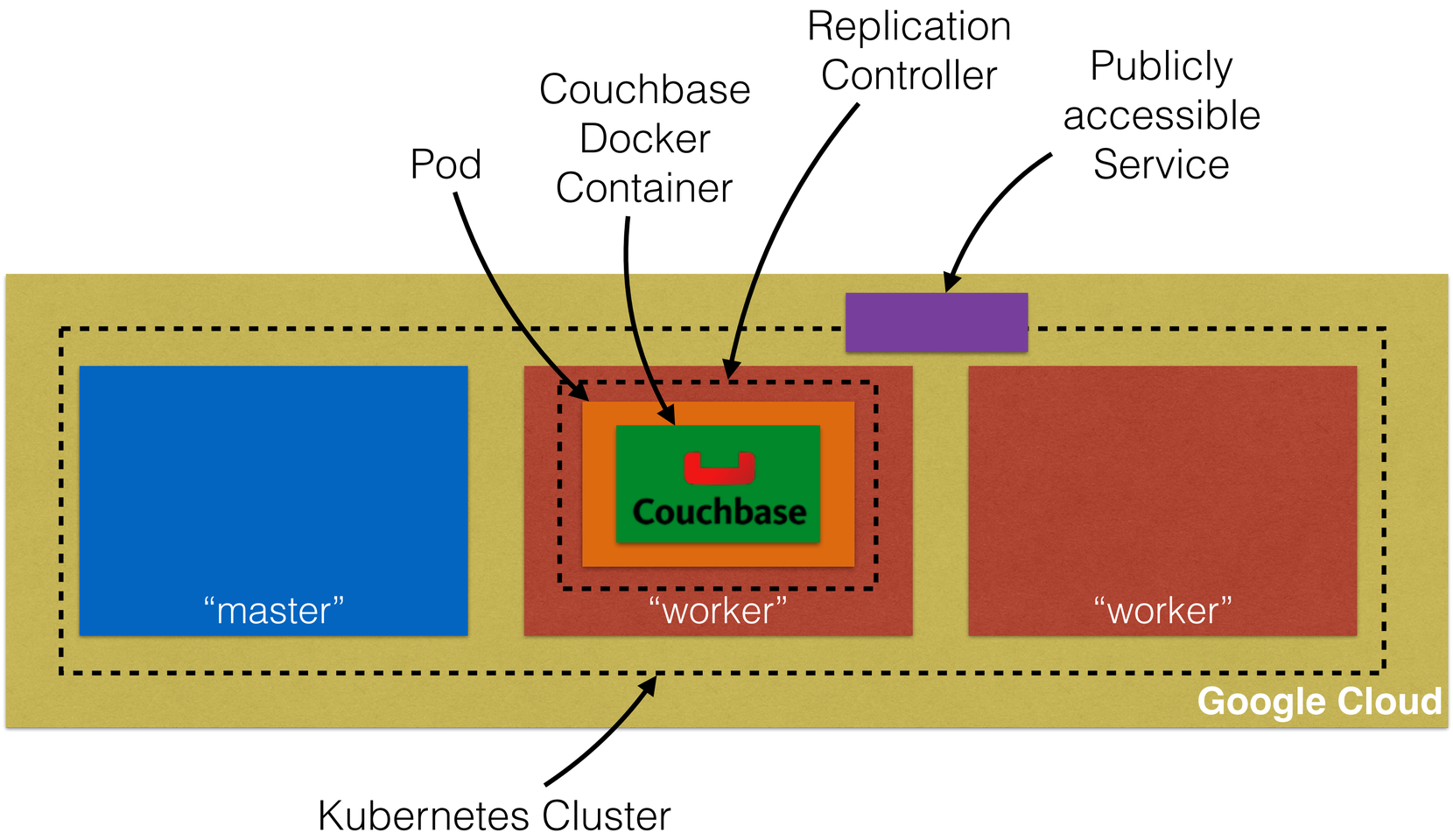
Services



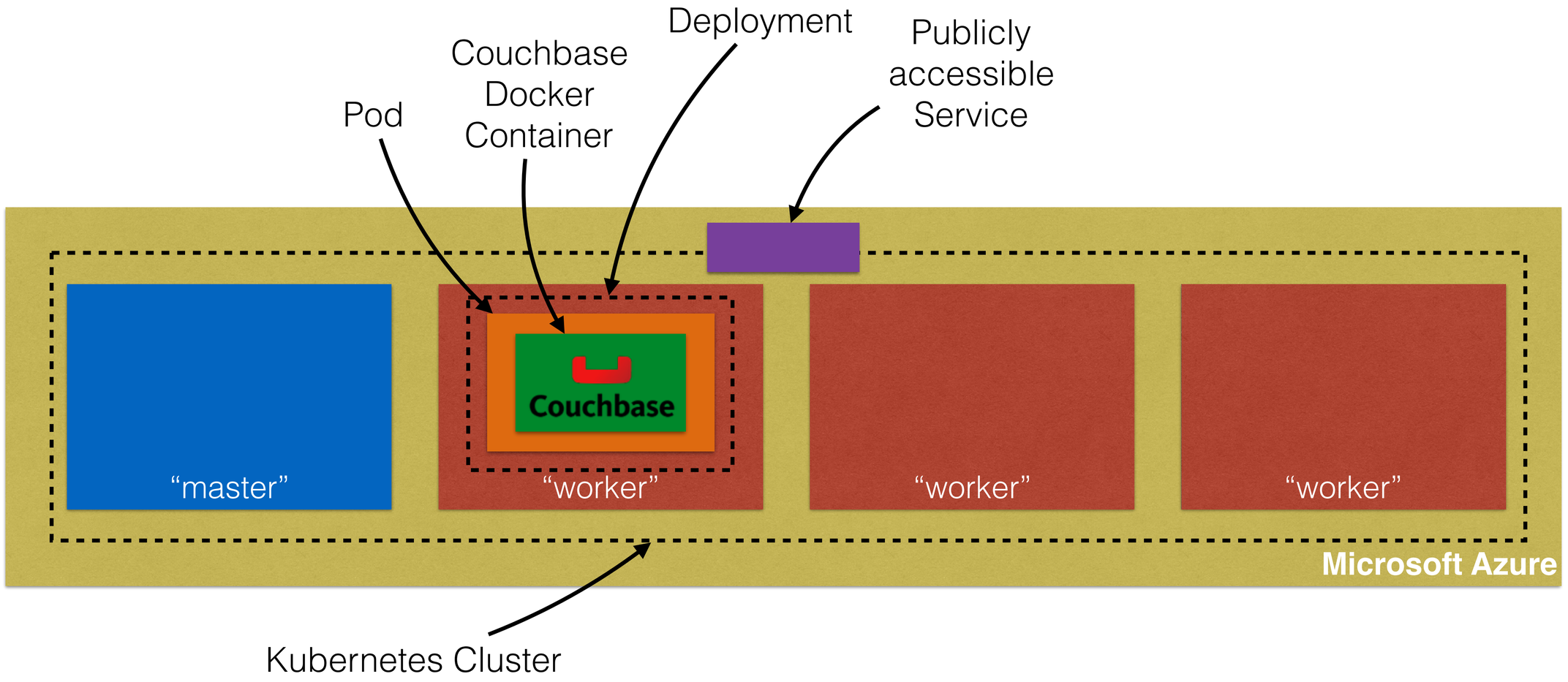
Services - AWS



Services - GCE



Services - Azure

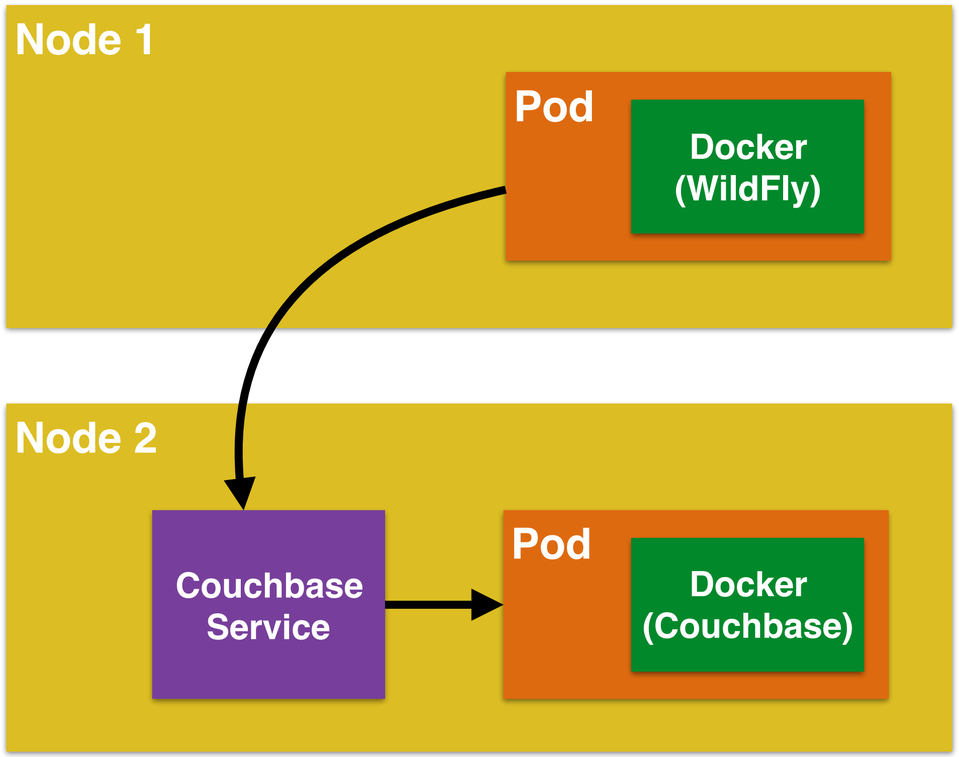


Services

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: couchbase-rc
  labels:
    name: couchbase-rc
    context: docker-k8s-lab
spec:
  replicas: 1
  template:
    metadata:
      name: couchbase-rc-pod
      labels:
        name: couchbase-rc-pod
        context: docker-k8s-lab
    spec:
      containers:
      - name: couchbase-rc-pod
        image: couchbase/server
        volumeMounts:
        - mountPath: "/opt/couchbase/var"
          name: mypd
        ports:
        - containerPort: 8091
          hostPort: 8091
        - containerPort: 8092
        - containerPort: 8093
        - containerPort: 11210
        -
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: couchbase-service
  labels:
    name: couchbase-service-pod
    context: docker-k8s-lab
spec:
  ports:
    - port: 8091
      port: 8092
      port: 8093
      port: 11210
  # label keys and values that must
  selector:
    name: couchbase-rc-pod
    context: docker-k8s-lab
```

Service across Two Nodes



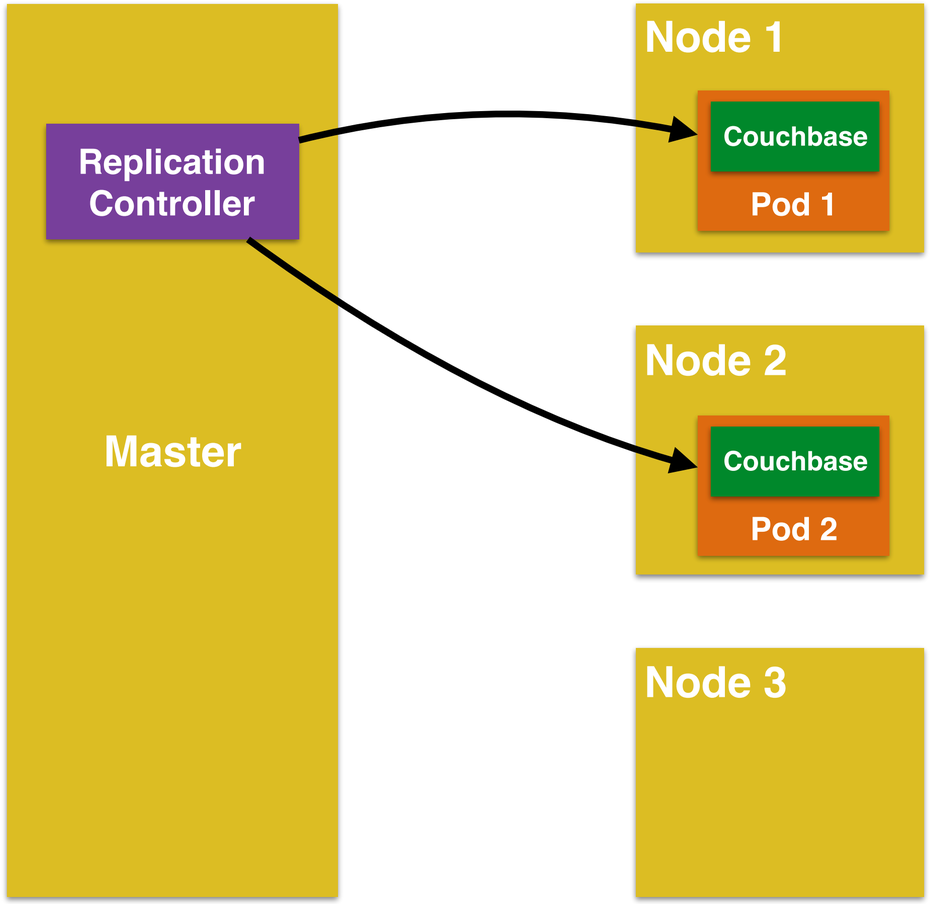
Replication Controller

- 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)

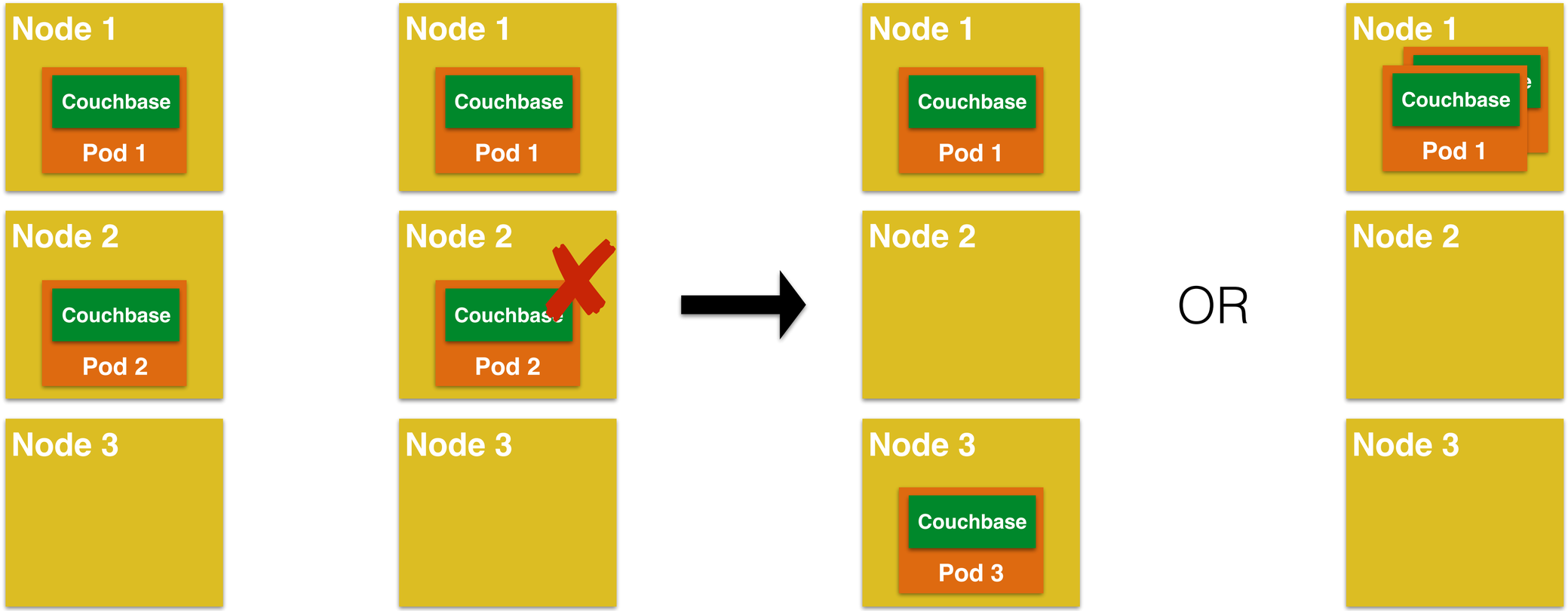
Replication Controller Configuration

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: wildfly-rc
  labels:
    name: wildfly
    context: docker-k8s-lab
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rc-pod
        image: arungupta/wildfly-mysql-javaee7:k8s
        ports:
        - containerPort: 8080
```

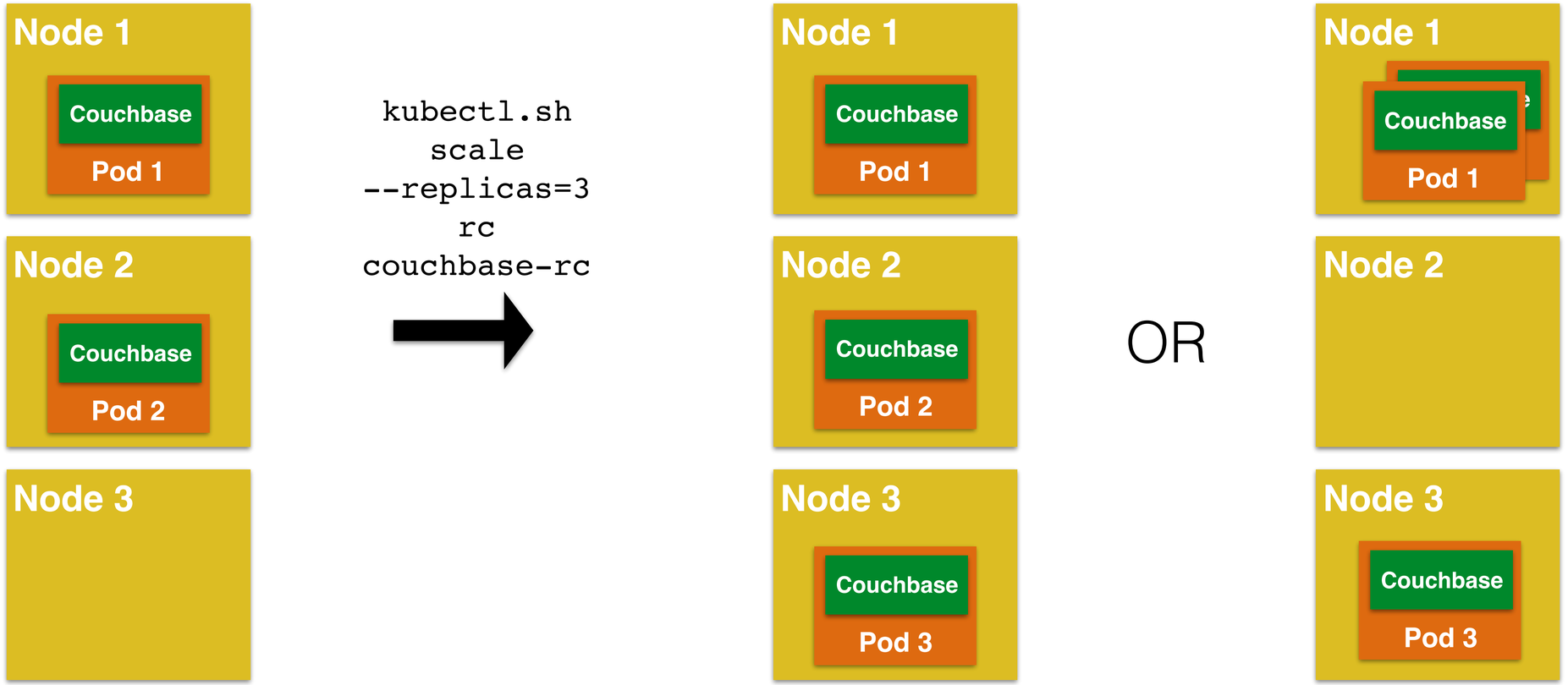
Replication Controller



Replication Controller: Automatic Rescheduling



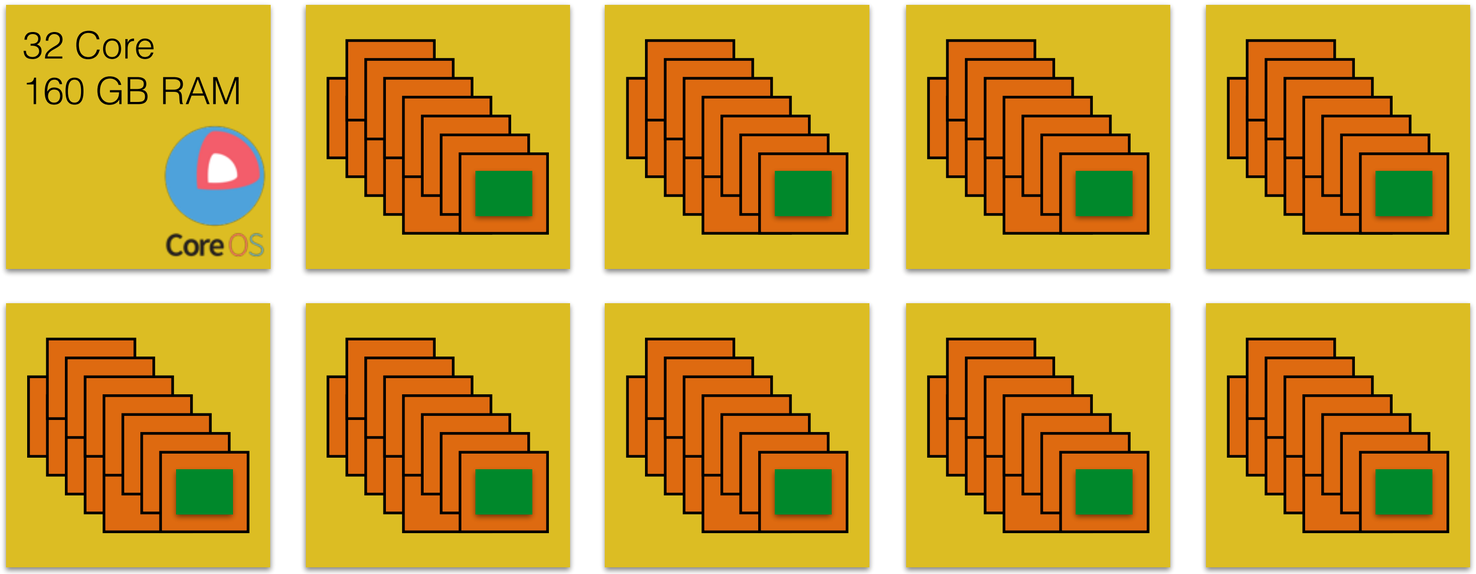
Replication Controller: Scaling



Sample Production Deployment

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

Sample Production Deployment



Load	Containers
Normal	400
Peak	600

Health Checks

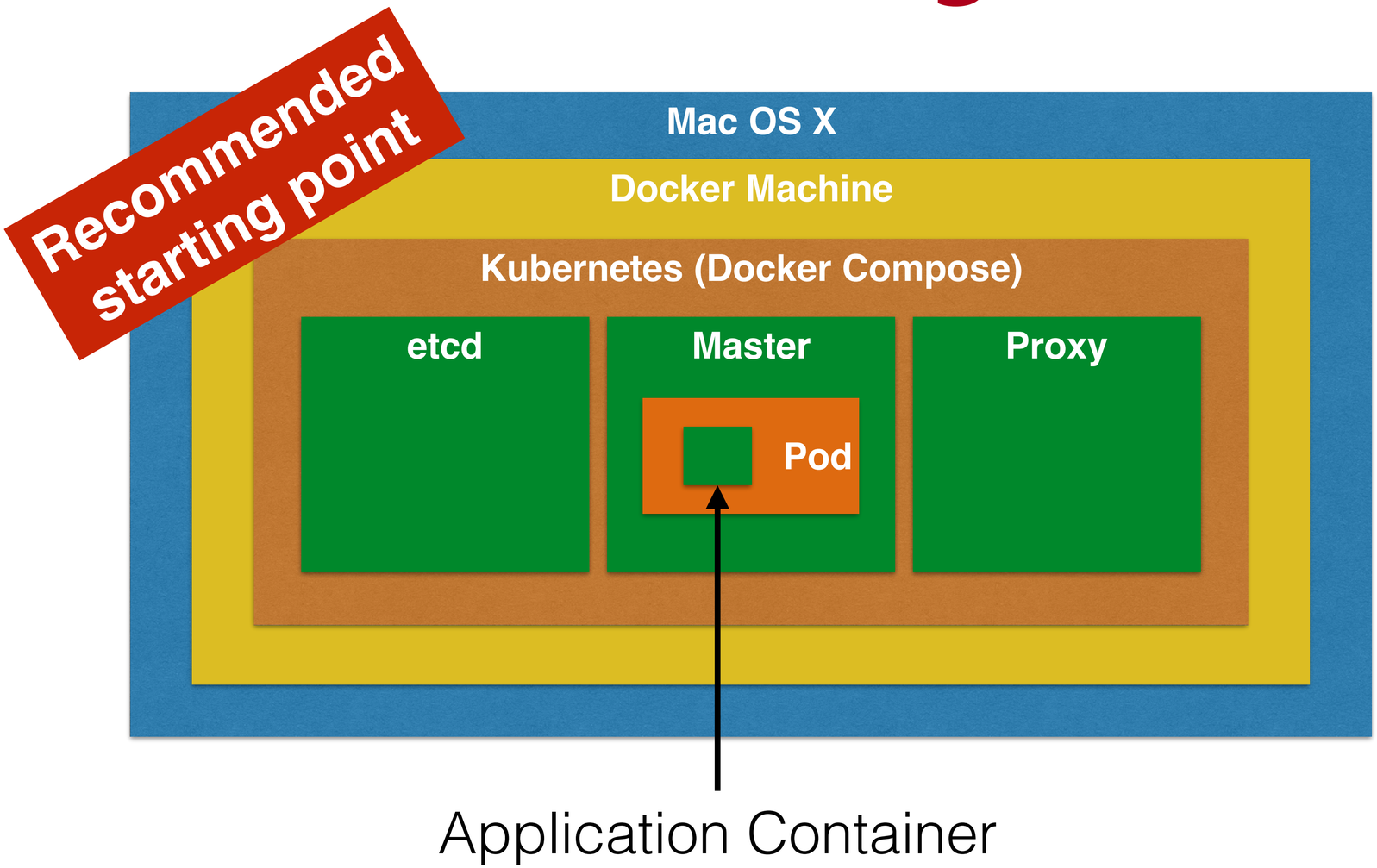
- Restarts Pod, if wrapped in RC
- Application-level health checks
  - HTTP
  - Container Exec
  - TCP Socket
- Health checks performed by Kubelet

Kubernetes using Docker

```
etcd:
  image: gcr.io/google_containers/etcd:2.0.9
  net: "host"
  entrypoint: /usr/local/bin/etcd --addr=127.0.0.1:4001 --bind-addr=0.0.0.0:4001 --data-dir=/
master:
  image: gcr.io/google_containers/hyperkube:v0.21.2
  net: "host"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  entrypoint: /hyperkube kubelet --api_servers=http://localhost:8080 --v=2 --address=0.0.0.0
proxy:
  image: gcr.io/google_containers/hyperkube:v0.21.2
  net: "host"
  privileged: true
  entrypoint: /hyperkube proxy --master=http://127.0.0.1:8080 --v=2
```

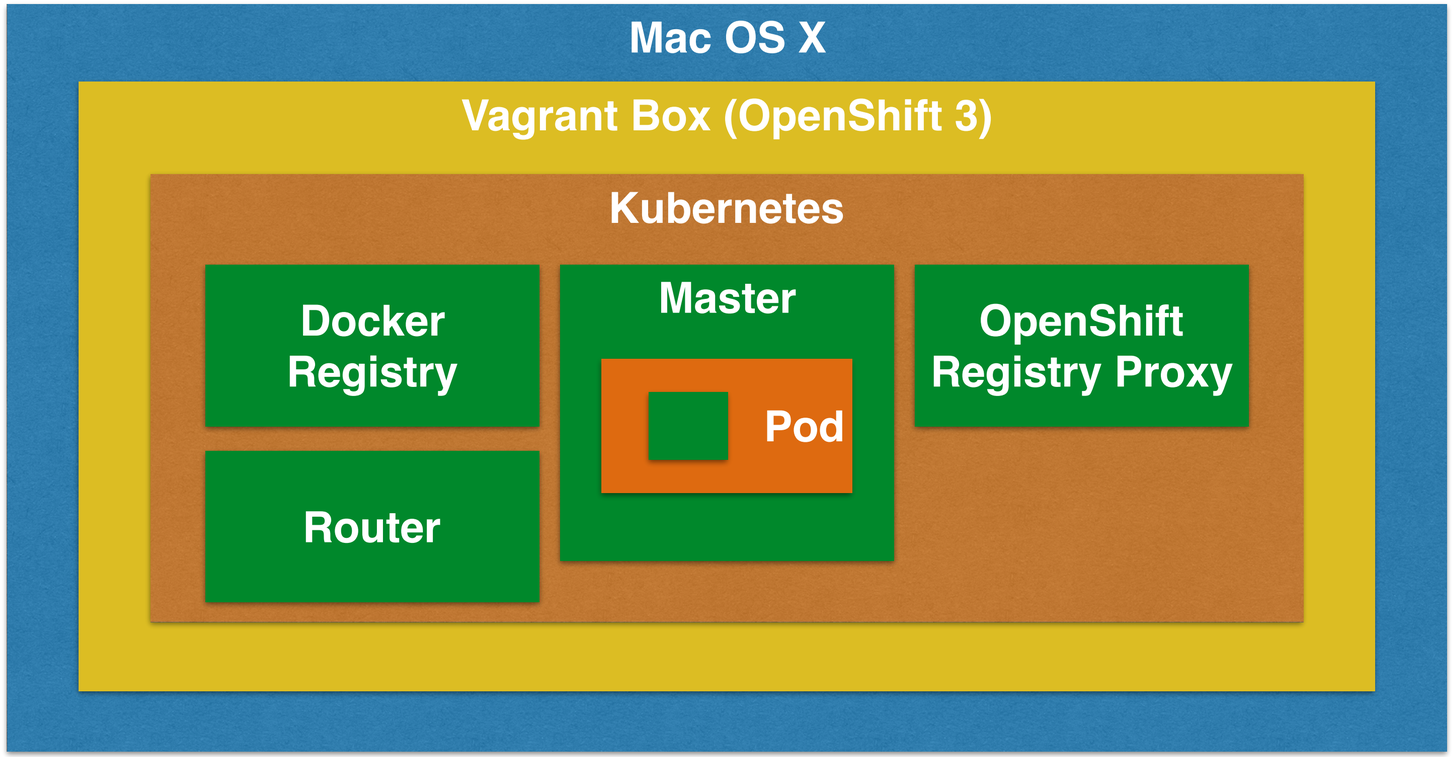

https://github.com/arun-gupta/docker-images/blob/master/kubernetes/docker-compose.yml

Kubernetes using Docker



http://kubernetes.io/v1.0/docs/getting-started-guides/docker.html

OpenShift 3



References

- github.com/javaee-samples/docker-java
- kubernetes.io
- Containers recipe: couchbase.com/containers