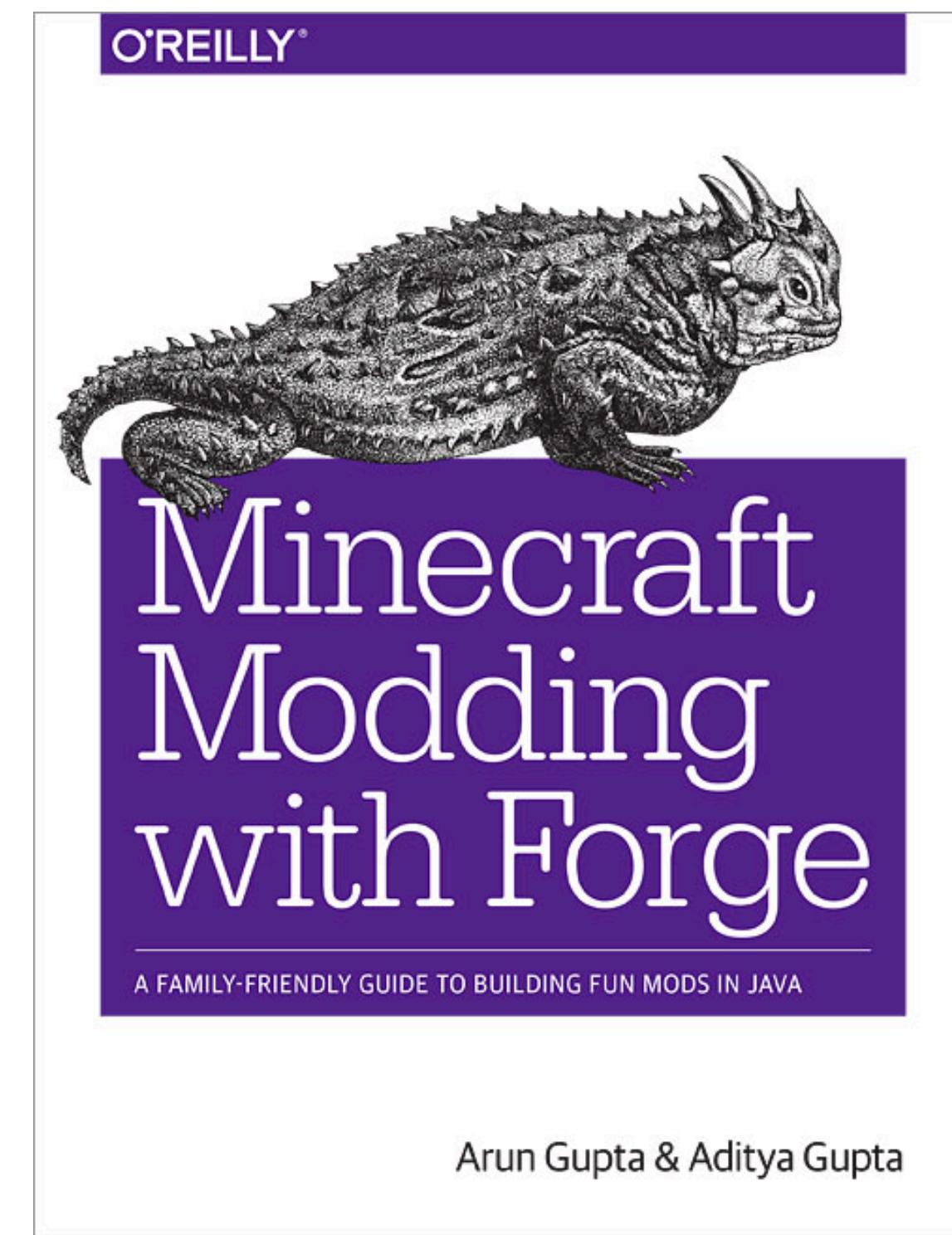


Migrate your traditional VM Clusters to Containers

Arun Gupta, Couchbase
[@arungupta](https://twitter.com/arungupta)





#Devoxx #containers

@arungupta

TM
X
O
>
S
D



#Devoxx #containers

@arungupta



Virtual Machine and Containers



Virtual Machine and Containers

Virtual Machines

Containers

Designed to provide isolated environment to run application



Virtual Machine and Containers

Virtual Machines

Containers

Designed to provide isolated environment to run application

Environment is binary artifact that can be moved between hosts



Virtual Machine and Containers

Virtual Machines

Containers

Designed to provide isolated environment to run application

Environment is binary artifact that can be moved between hosts

Houses

Apartments
(Building, Leasing Office)



Virtual Machine and Containers

Virtual Machines

Containers

Designed to provide isolated environment to run application

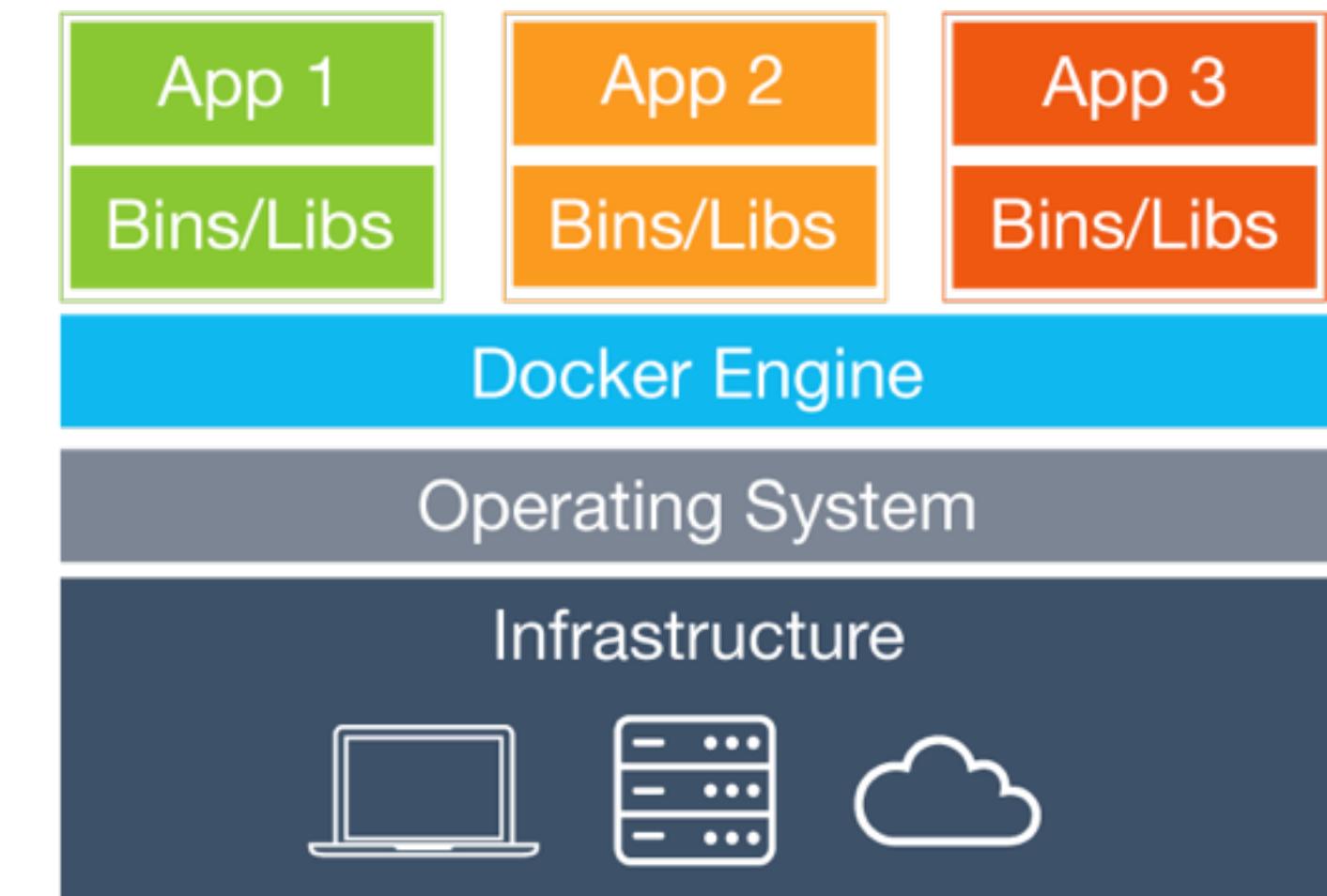
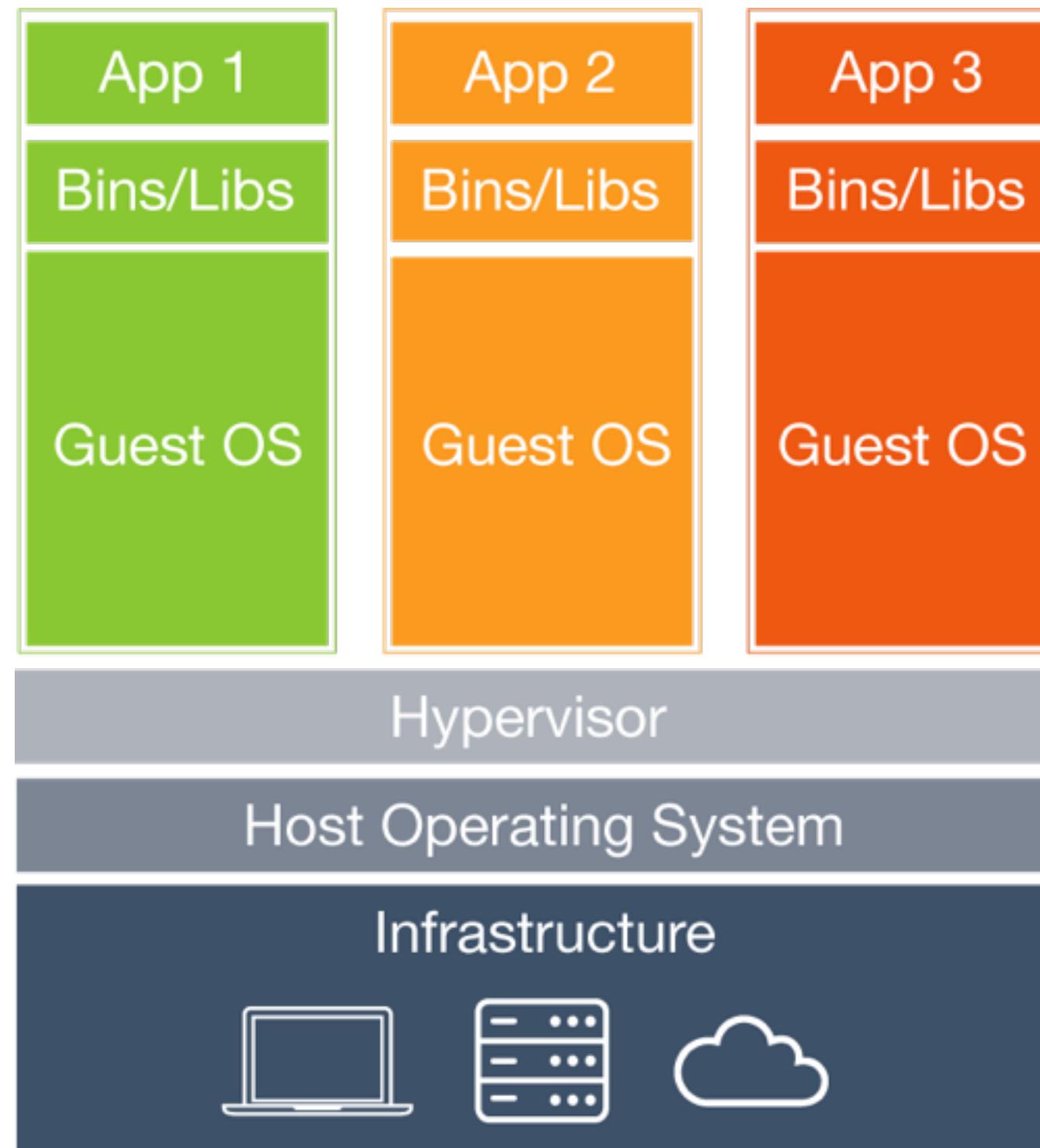
Environment is binary artifact that can be moved between hosts

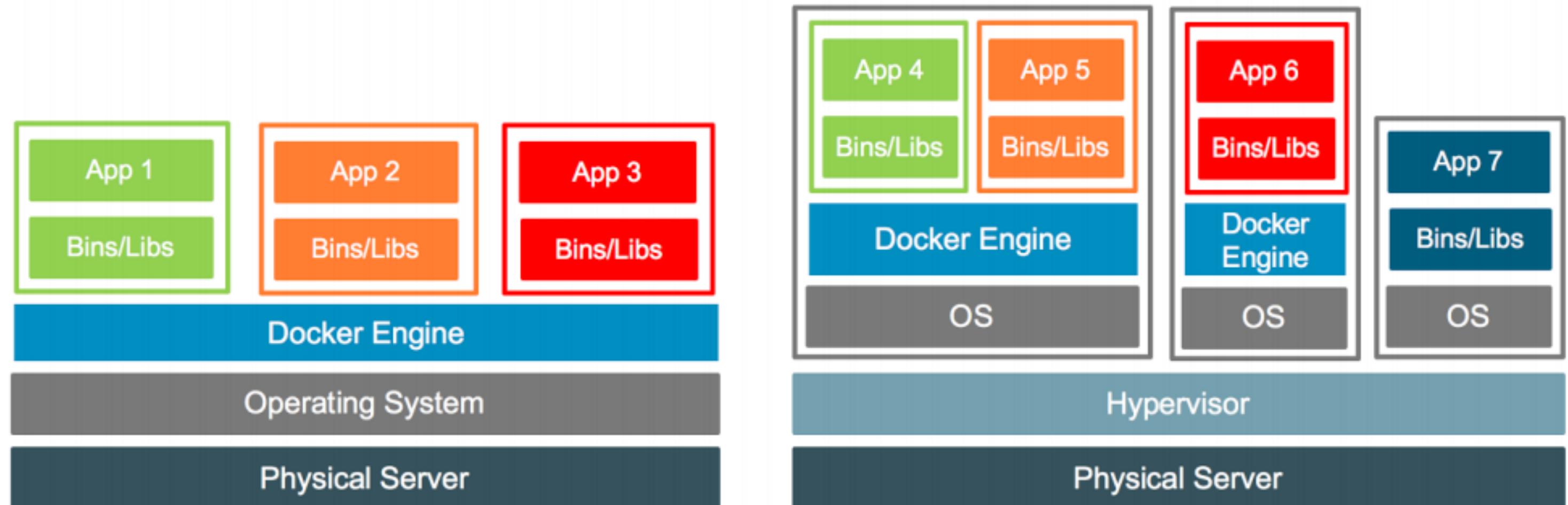
Houses

Apartments
(Building, Leasing Office)

Remove what you don't need

Add what you want







Docker is **not** a
Virtualization
Technology

Docker is an
Application Delivery
Technology





Couchbase Cluster Creation

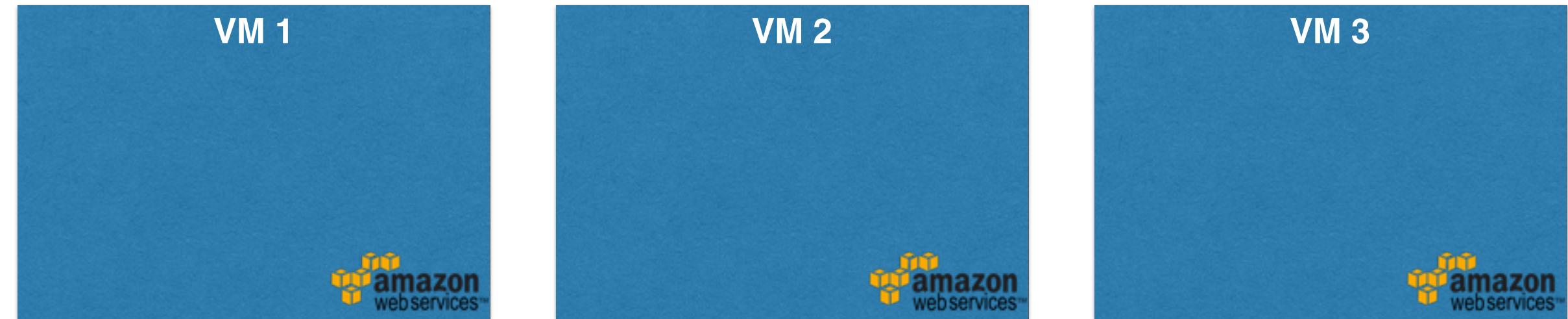
- Amazon Web Services - Classical
- Docker for AWS
- Kubernetes
- Red Hat OpenShift
- Mesos + Marathon



Amazon Web Services - Classical



Amazon Web Services - Classical



- Create VM 1, 2, 3



Amazon Web Services - Classical



- Create VM 1, 2, 3
- Install/Start Couchbase on each VM

TM

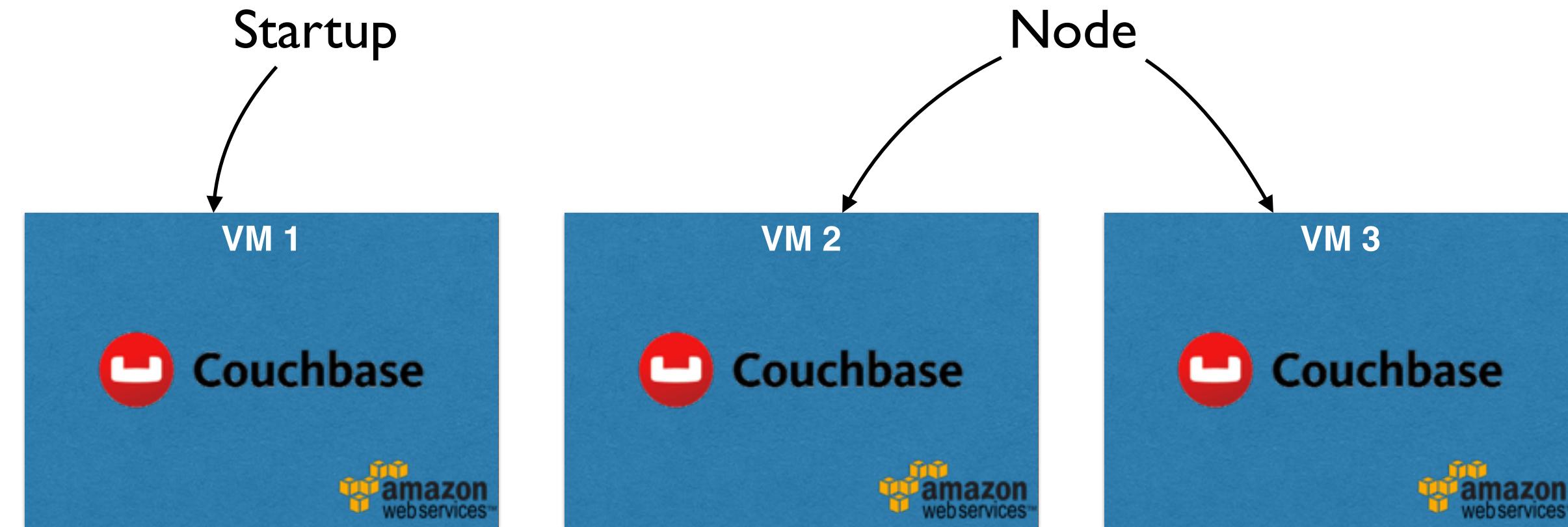
Amazon Web Services - Classical



- Create VM 1, 2, 3
- Install/Start Couchbase on each VM



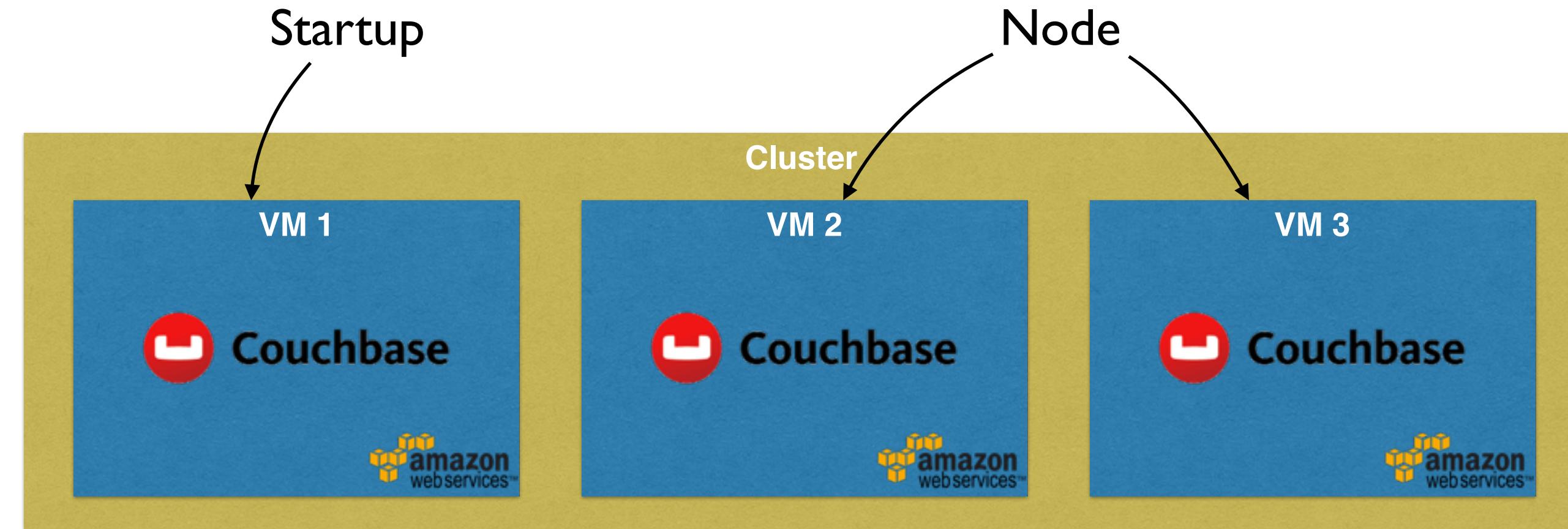
Amazon Web Services - Classical



- Create VM 1, 2, 3
- Install/Start Couchbase on each VM



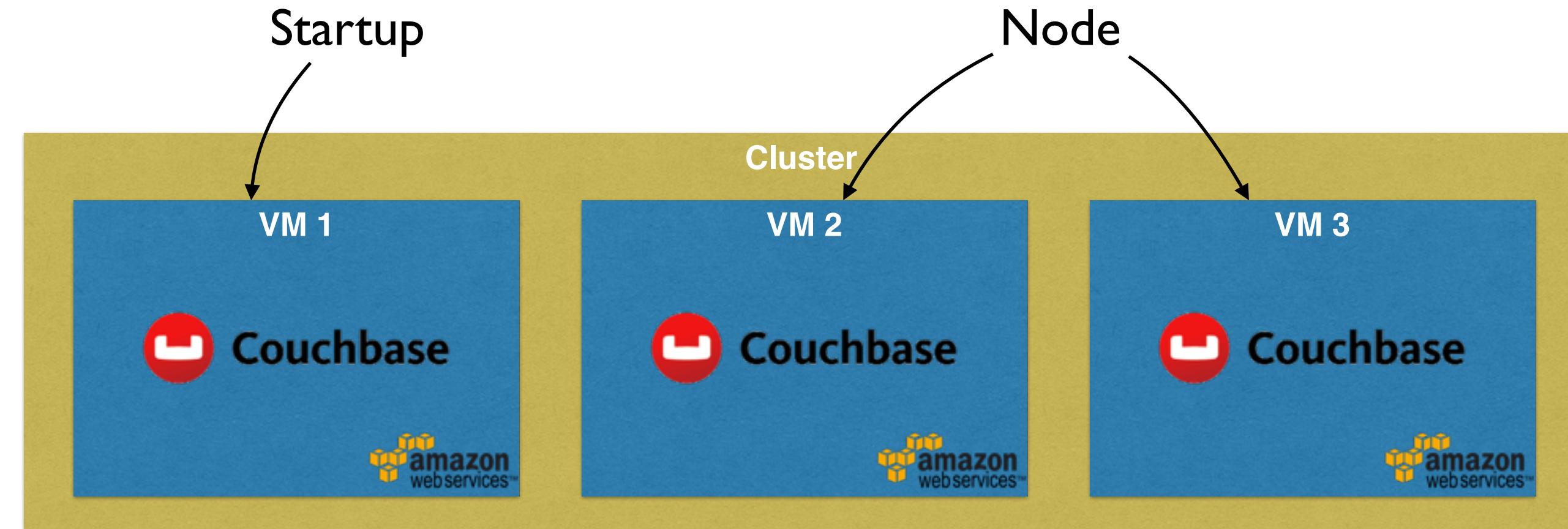
Amazon Web Services - Classical



- Create VM 1, 2, 3
- Install/Start Couchbase on each VM
- Create cluster: Add 2 and 3 to 1



Amazon Web Services - Classical

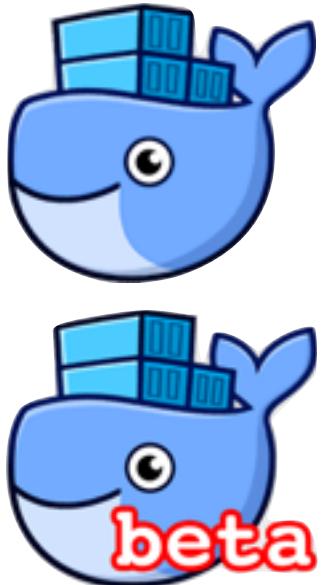


- Create VM 1, 2, 3
- Install/Start Couchbase on each VM
- Create cluster: Add 2 and 3 to 1
- Rebalance cluster

TM

D E V O O X

Docker for Mac/Windows

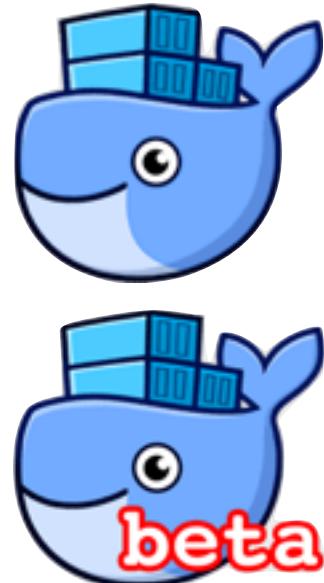


TM



Docker for Mac/Windows

- Native application and UI

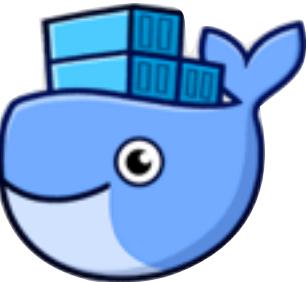


TM



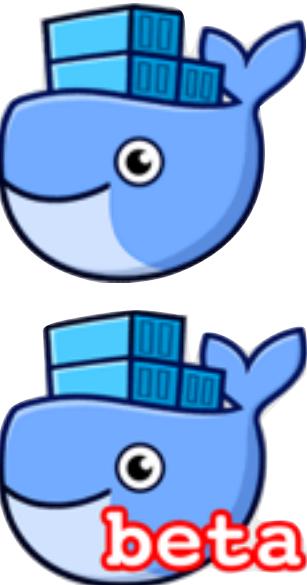
Docker for Mac/Windows

- Native application and UI
- Auto update capability





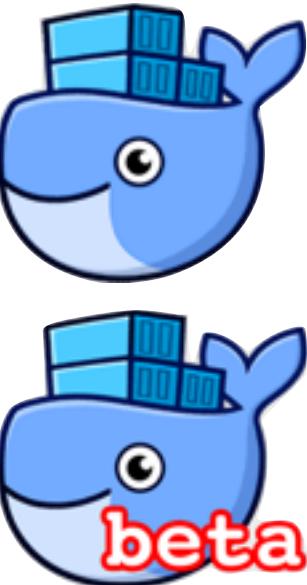
Docker for Mac/Windows



- Native application and UI
- Auto update capability
- No additional software required, e.g. VirtualBox



Docker for Mac/Windows

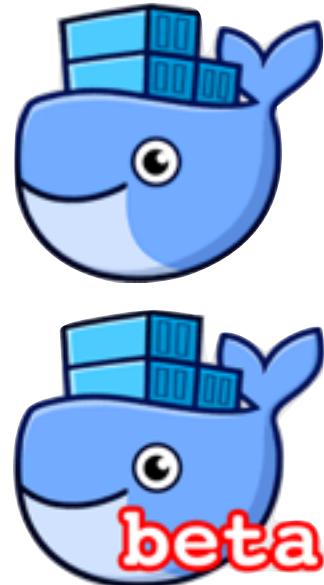


- Native application and UI
- Auto update capability
- No additional software required, e.g. VirtualBox
 - OSX: xhyve VM using Hypervisor.framework
 - Windows: Hyper-VM



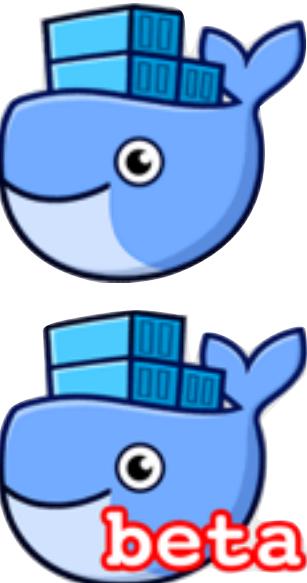
Docker for Mac/Windows

- Native application and UI
- Auto update capability
- No additional software required, e.g. VirtualBox
 - OSX: xhyve VM using Hypervisor.framework
 - Windows: Hyper-VM
- Download: docker.com/getdocker





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-VM
- Download: docker.com/getdocker
- Requires Yosemite 10.10+ or Windows 10 64-bit



Docker for AWS/Azure

TM
X
X
O
V
W
S
D

TM
X
X
O
V
O
S
D

Docker for AWS/Azure



- Amazon Web Services



Docker for AWS/Azure

- Amazon Web Services
- Amazon CloudFormation templates

TM
X
O
V
W
S
D



Docker for AWS/Azure

- Amazon Web Services
 - Amazon CloudFormation templates
 - Integrated with Autoscaling, ELB, and EBS

TM
X
O
V
W
D

A vertical column of icons on the left side of the slide. From top to bottom, they are: a yellow 'TM' symbol, a black 'X' symbol, a black-outlined square 'O', a downward-pointing triangle 'V', a right-pointing triangle 'W', and a black-outlined square 'D'.



Docker for AWS/Azure

- Amazon Web Services
 - Amazon CloudFormation templates
 - Integrated with Autoscaling, ELB, and EBS
- Azure

TM
X
O
V
W
D

A vertical column of icons on the left side of the slide. From top to bottom, they are: a yellow 'X' with a black outline and a trademark symbol; a black 'O'; a black downward-pointing triangle; a large black 'W'; and a black square.



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





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 (public beta soon)





```
4 /entrypoint.sh couchbase-server &
5
6 sleep 15
7
8 # Setup index and memory quota
9 curl -v -X POST http://127.0.0.1:8091/pools/default -d memoryQuota=300 -d indexMemoryQuota=300
10
11 # Setup services
12 curl -v http://127.0.0.1:8091/node/controller/setupServices -d services=kv%2Cn1ql%2Cindex
13
14 # Setup credentials
15 curl -v http://127.0.0.1:8091/settings/web -d port=8091 -d username=Administrator -d password=password
16
17 # Setup Memory Optimized Indexes
18 curl -i -u Administrator:password -X POST http://127.0.0.1:8091/settings/indexes -d 'storageMode=memory_optimized'
19
20 # Load travel-sample bucket
21 curl -v -u Administrator:password -X POST http://127.0.0.1:8091/sampleBuckets/install -d '[{"travel-sample"}]'
22
23 echo "Type: $TYPE"
24
25 if [ "$TYPE" = "WORKER" ]; then
26   sleep 15
27
28 #IP=`hostname -s`
29 IP=`hostname -I | cut -d ' ' -f1`
30
31 echo "Auto Rebalance: $AUTO_REBALANCE"
32 if [ "$AUTO_REBALANCE" = "true" ]; then
33   couchbase-cli rebalance --cluster=$COUCHBASE_MASTER:8091 --user=Administrator --password=password --server-add=$IP --server-add-username=Administrator --server-add-password=password
34 else
35   couchbase-cli server-add --cluster=$COUCHBASE_MASTER:8091 --user=Administrator --password=password --server-add=$IP --server-add-username=Administrator --server-add-password=password
36 fi;
37 fi;
38
39 fg 1
```

TM

Devoxx
2016

Using Docker Services



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```

Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```



Using Docker Services

```
docker service create  
--name couchbase-master  
-p 8091:8091  
--replicas 1  
--network couchbase  
-e TYPE=MASTER  
arungupta/couchbase:swarm
```

```
docker service create  
--name couchbase-worker  
--replicas 1  
--network couchbase  
-e TYPE=WORKER  
-e COUCHBASE_MASTER=  
    couchbase-master.couchbase  
arungupta/couchbase:swarm
```

```
docker service scale couchbase-worker=3
```

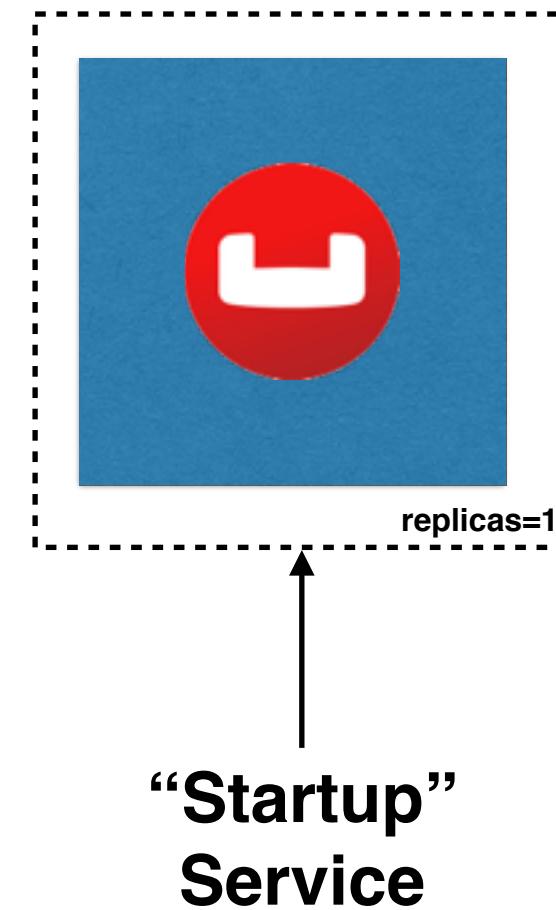


Couchbase in Docker for AWS

TM

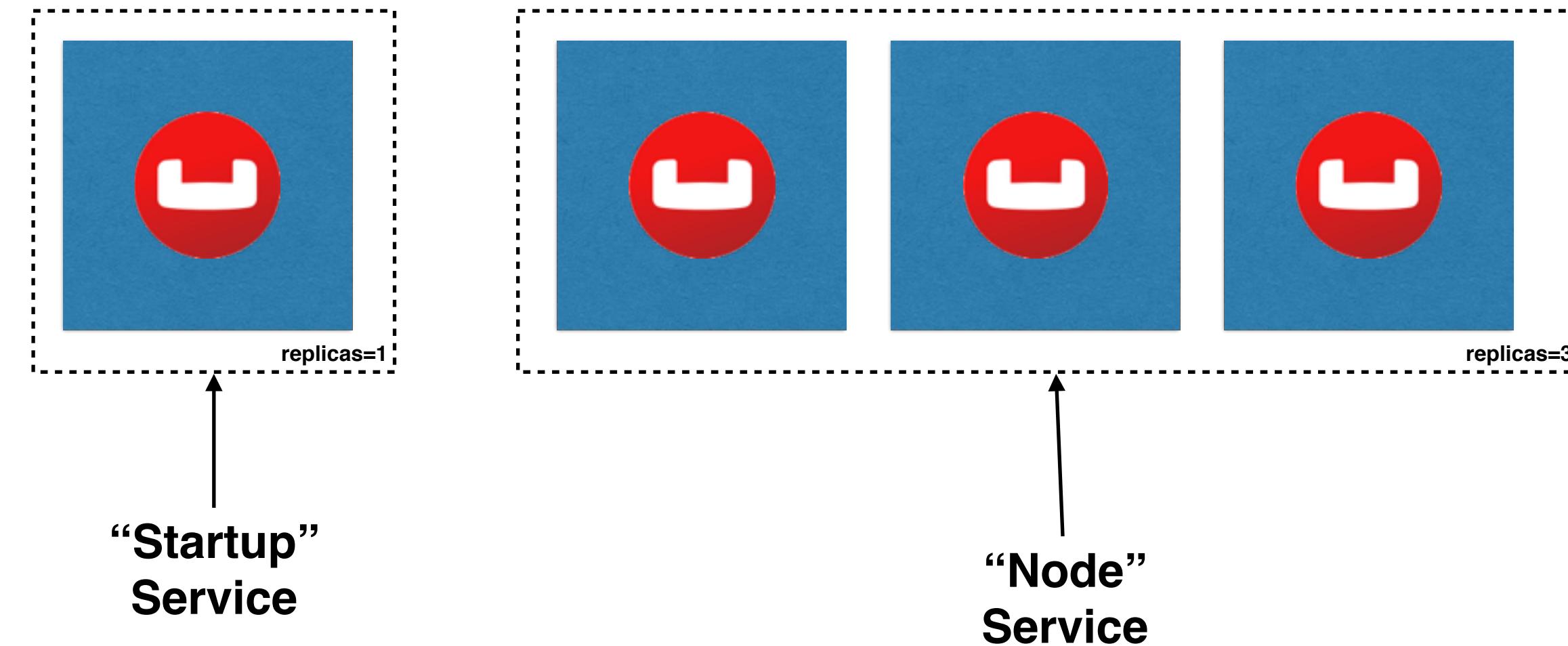


Couchbase in Docker for AWS



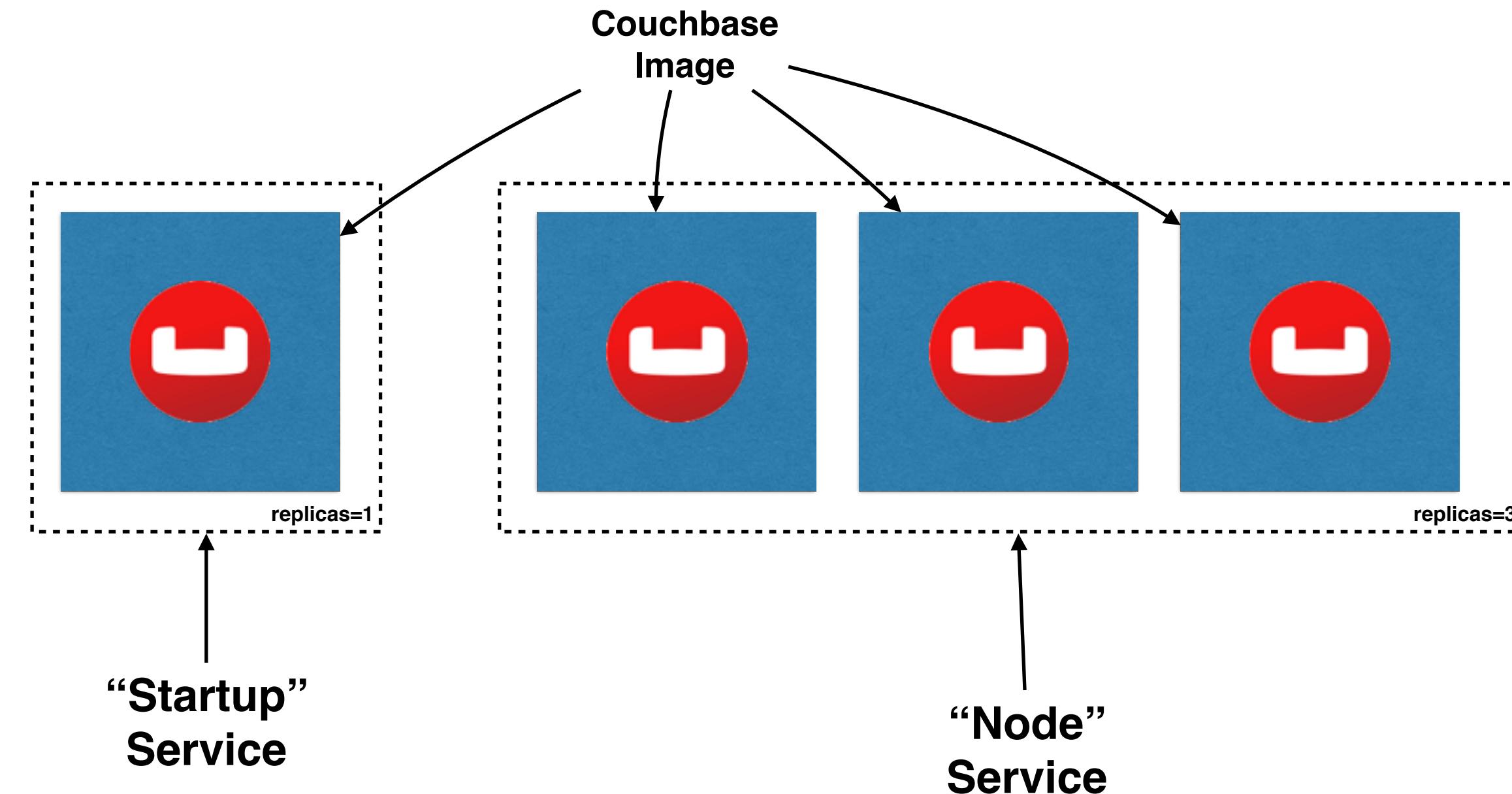
TM

Couchbase in Docker for AWS

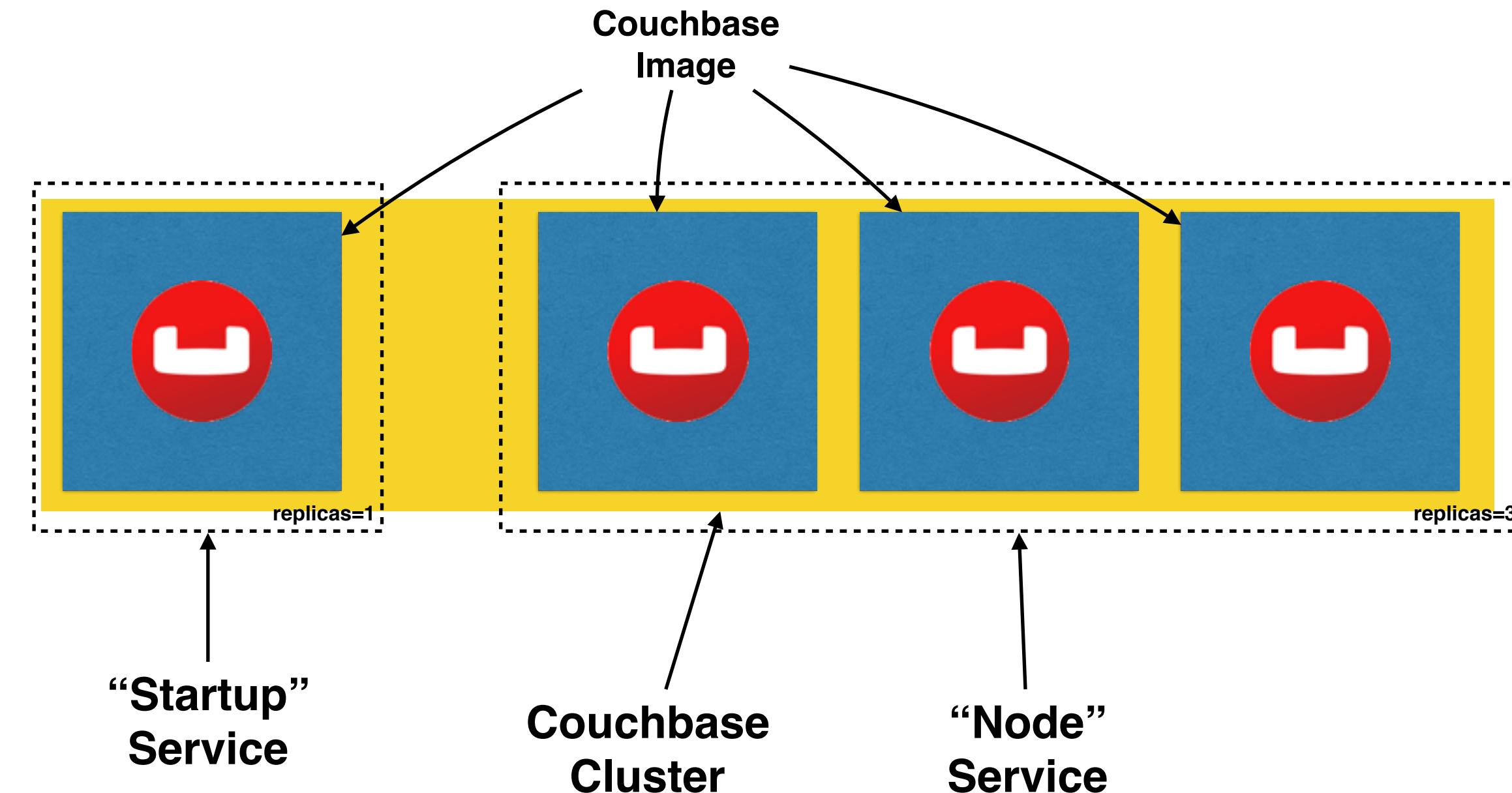


TM

Couchbase in Docker for AWS



Couchbase in Docker for AWS



<http://blog.couchbase.com/2016/november/docker-for-aws-getting-started-video>



Docker Data Center

- Commercially Supported Docker Engine
- Docker Trusted Registry
- Docker Universal Control Plane



Managing Data in Docker



Managing Data in Docker

- Per-container storage (`docker run`)



Managing Data in Docker

- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)



Managing Data in Docker

- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)
- Shared network filesystem: Ceph, GlusterFS, NFS



Managing Data in Docker

- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)
- Shared network filesystem: Ceph, GlusterFS, NFS
- Docker Volume Plugins: Flocker, Portworx, ...



Managing Data in Docker

- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)
- Shared network filesystem: Ceph, GlusterFS, NFS
- Docker Volume Plugins: Flocker, Portworx, ...
- Volumes follow containers across hosts



Managing Data in Docker

- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)
- Shared network filesystem: Ceph, GlusterFS, NFS
- Docker Volume Plugins: Flocker, Portworx, ...
 - Volumes follow containers across hosts
 - EBS, GCE persistent disk, ...

<http://blog.couchbase.com/2016/october/persisting-couchbase-data-across-container-restarts>

TM

Kubernetes Concepts

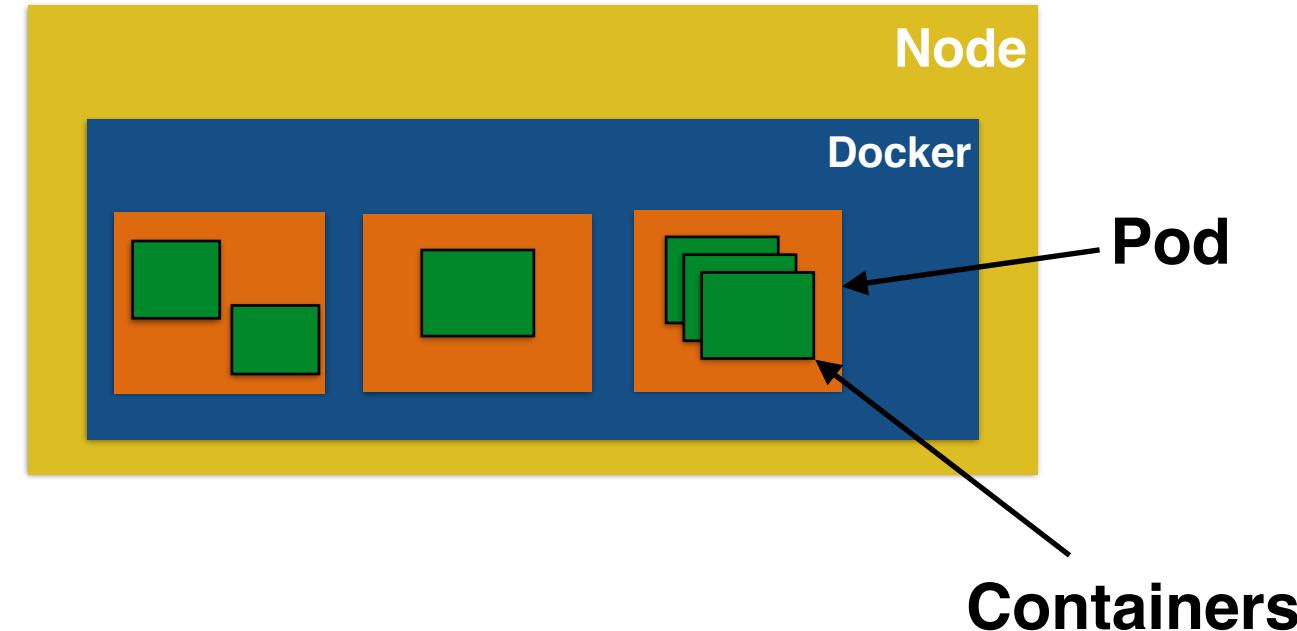


X
O
V
W
S
D



Kubernetes Concepts

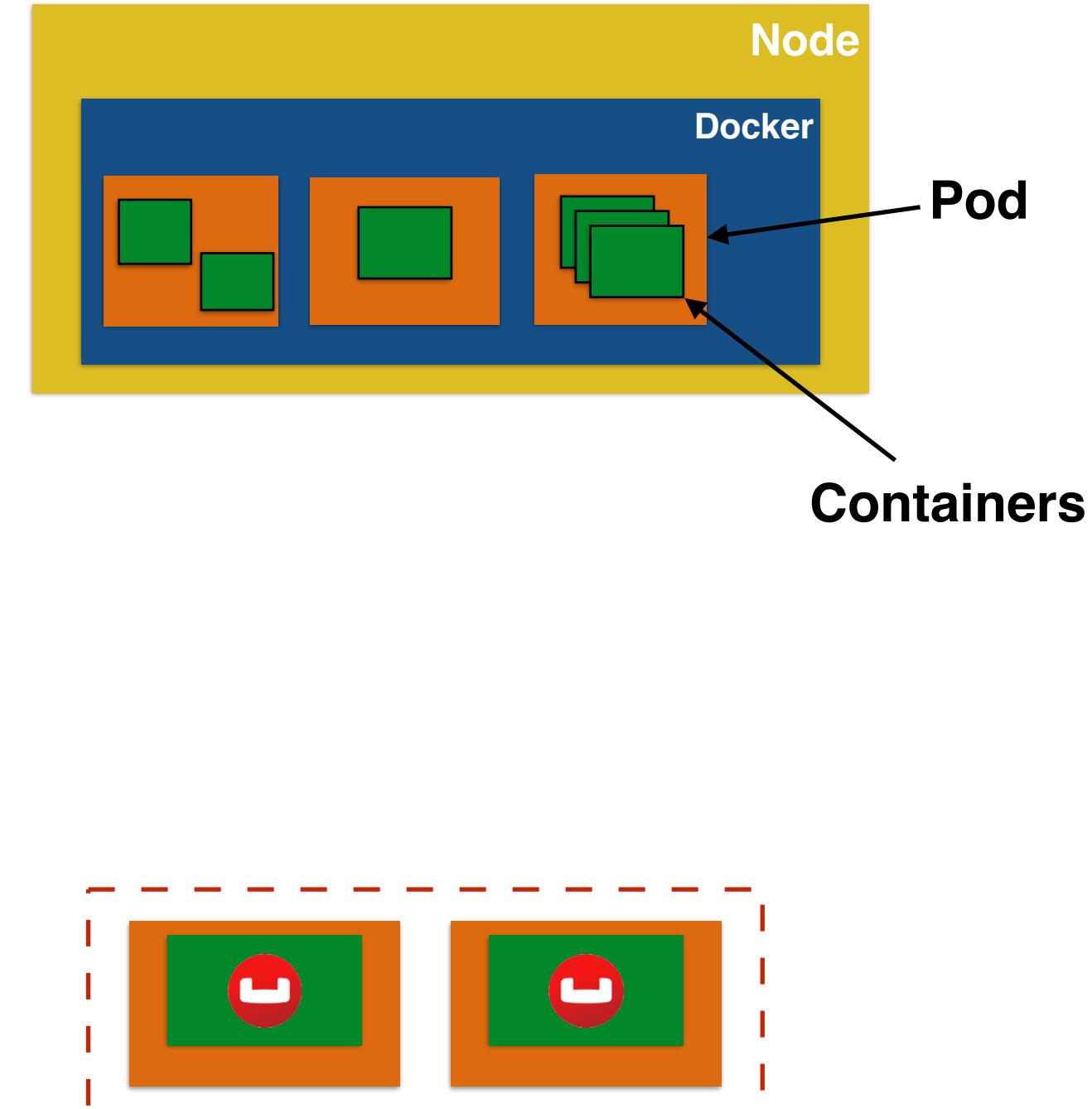
- **Pods:** colocated group of containers that share an IP, namespace, storage volume



Kubernetes Concepts



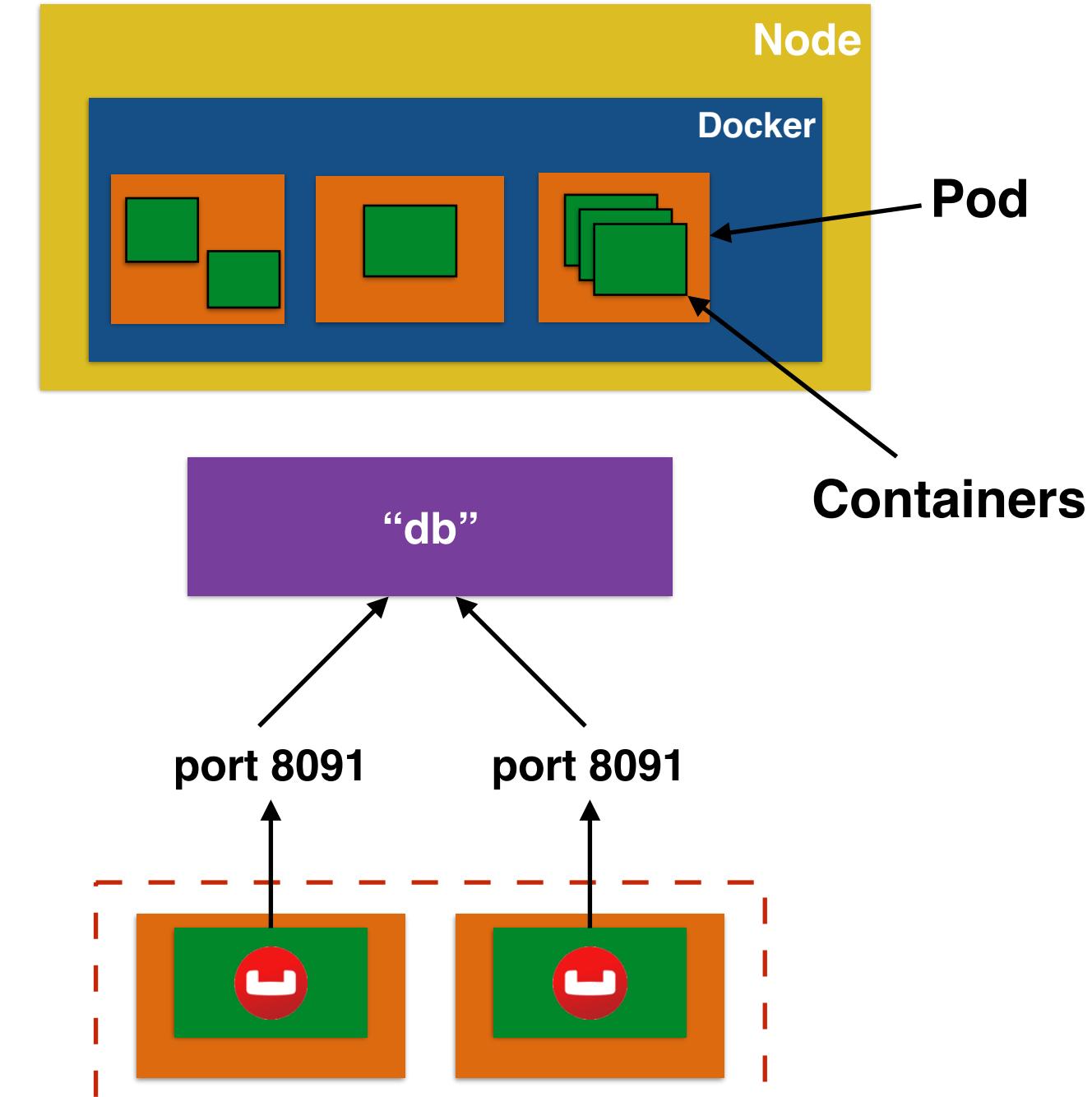
- **Pods:** colocated group of containers that share an IP, namespace, storage volume
- **Replica Set:** manages the lifecycle of pods and ensures specified number are running (next gen Replication Controller)





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



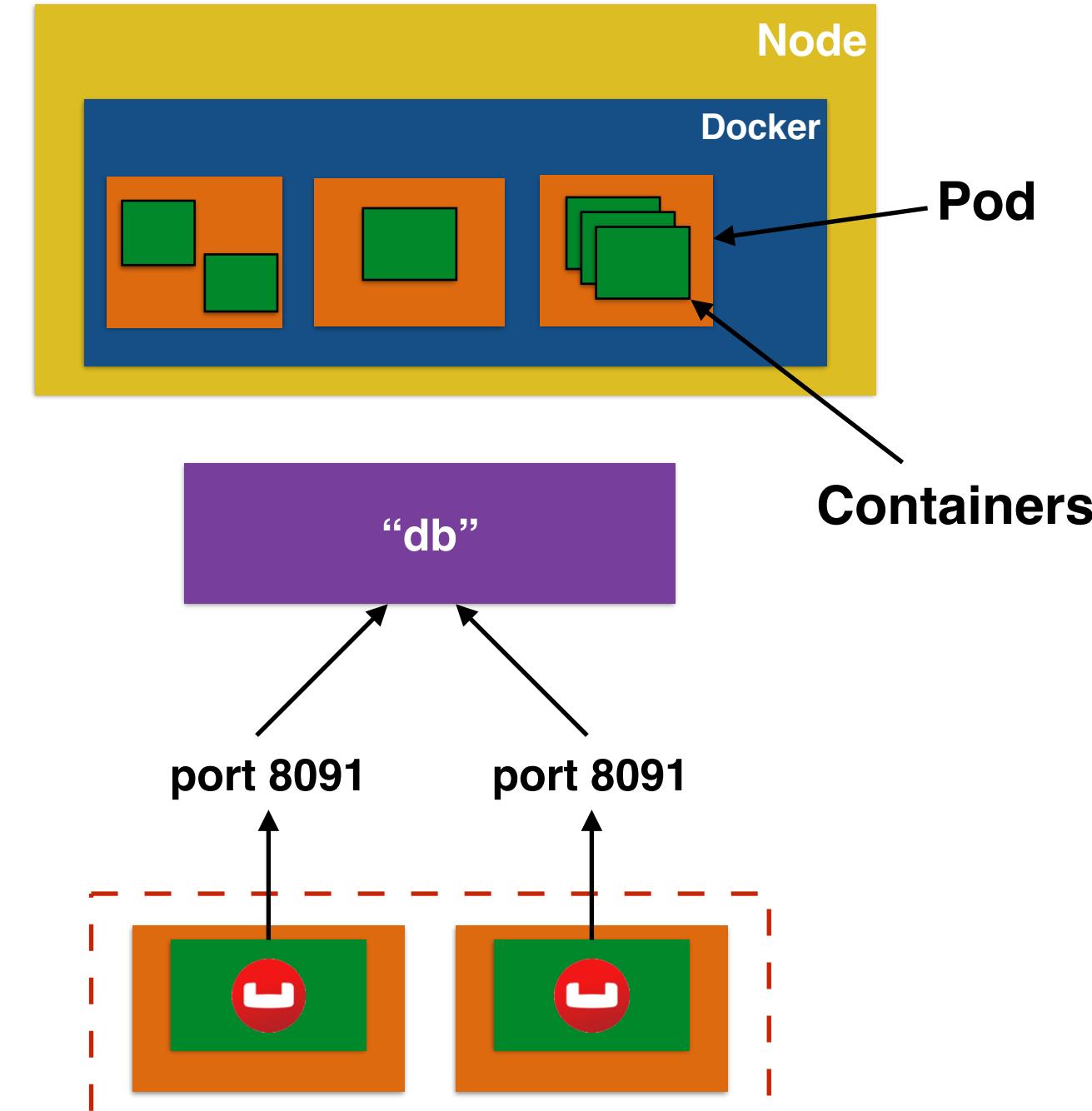


Kubernetes Concepts



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





Kubernetes Components

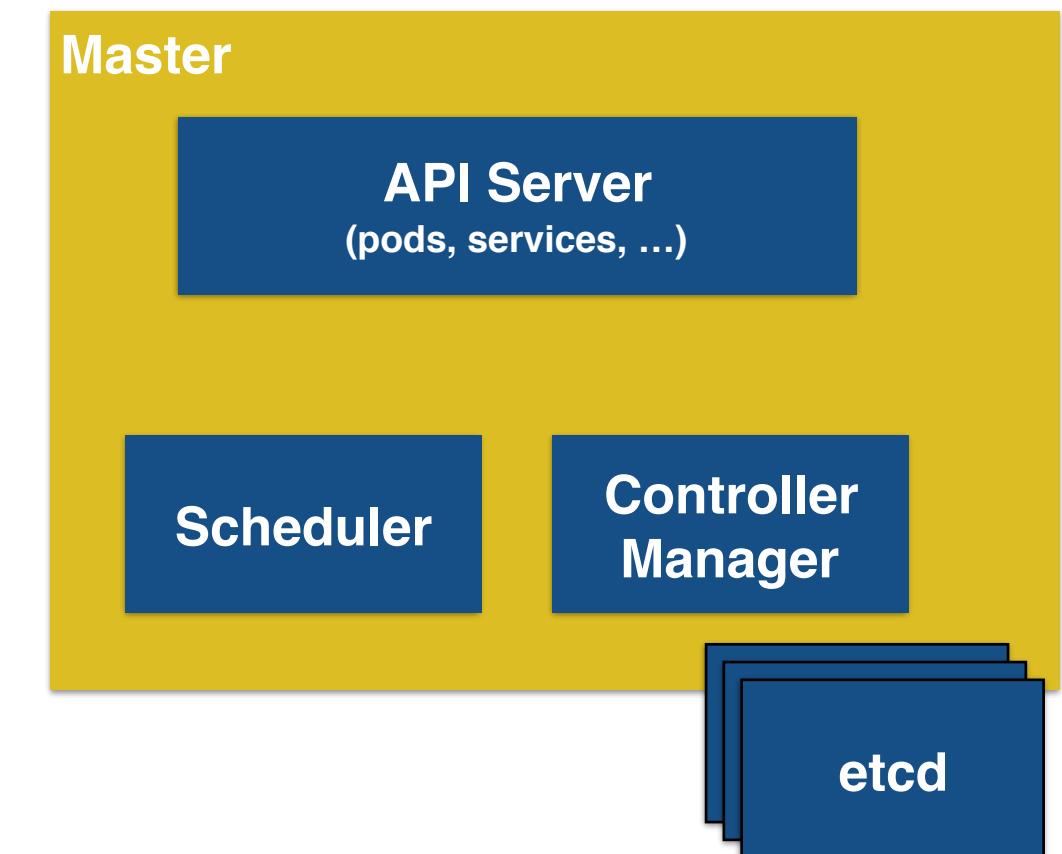


Kubernetes Components

- **Node:** Machine or VM in the cluster

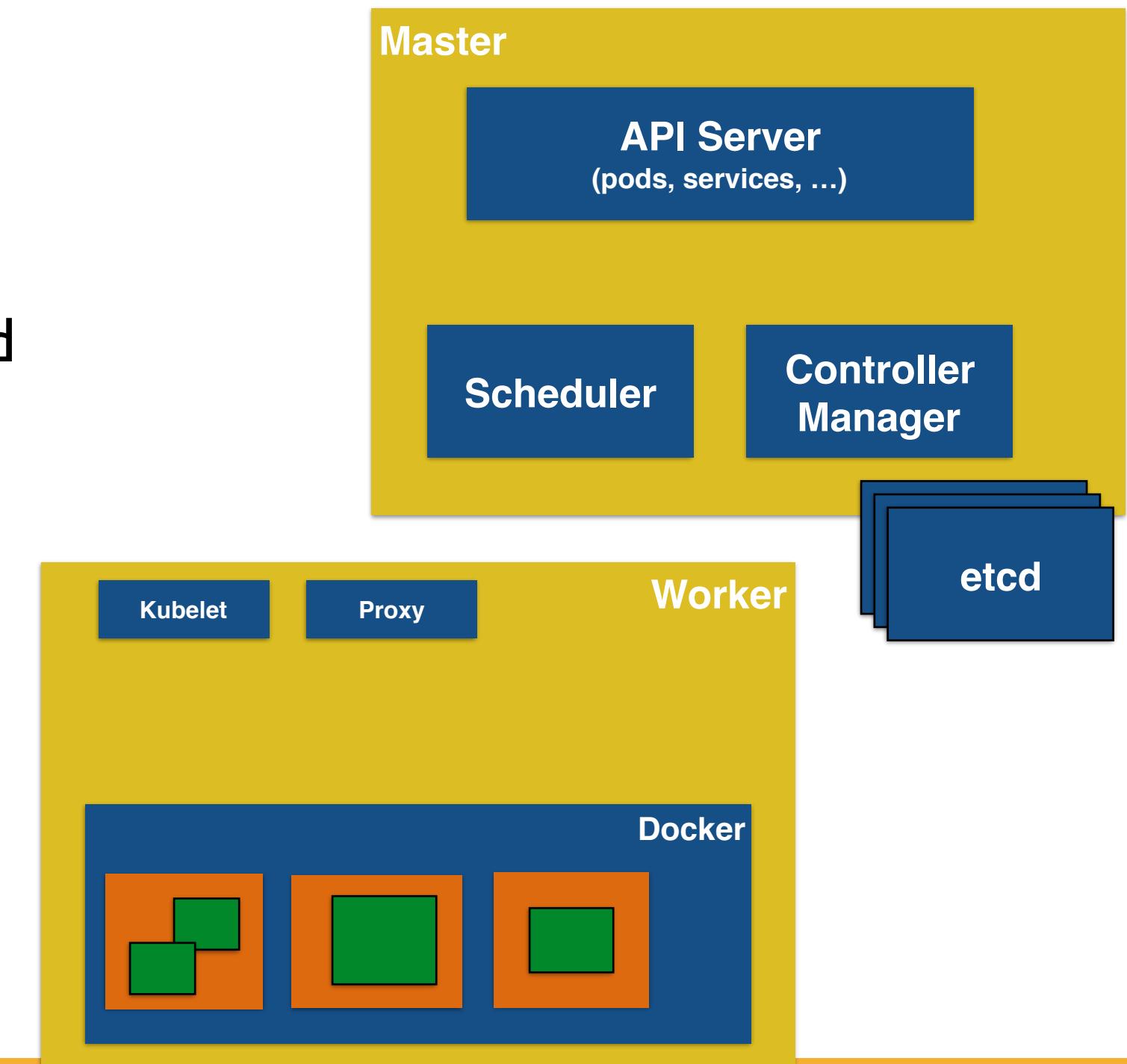
Kubernetes Components

- **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



Kubernetes Components

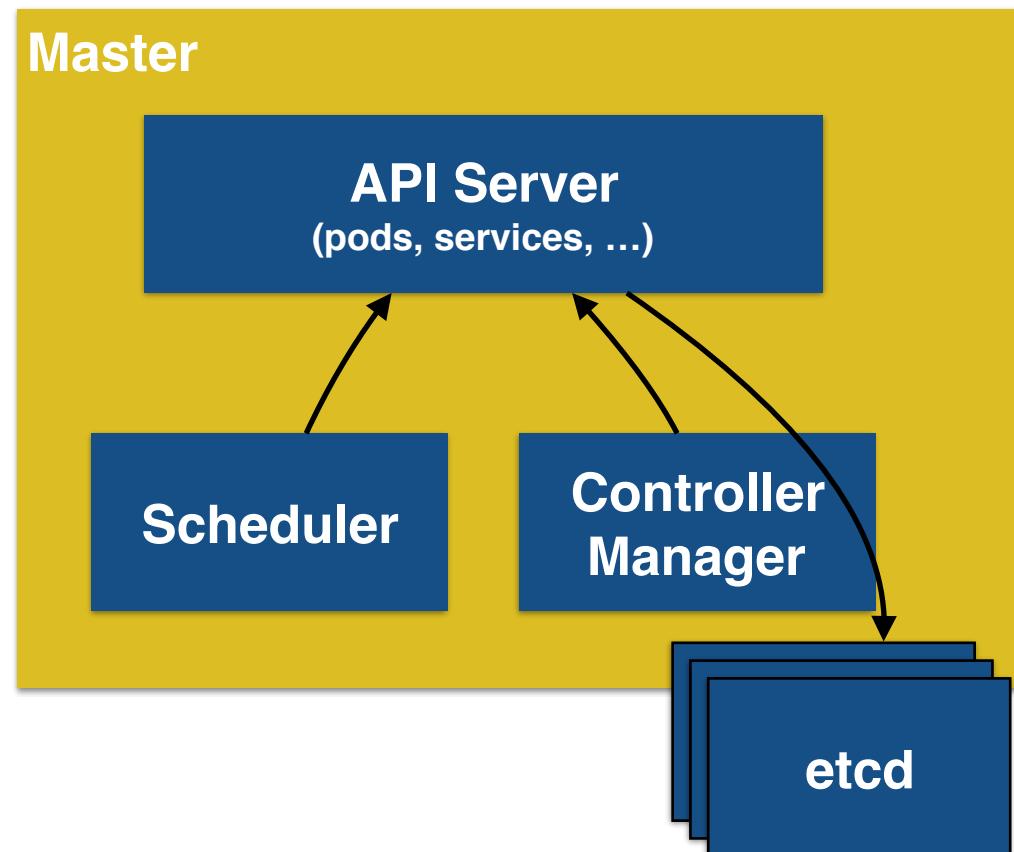
- **Node:** Machine or VM in the cluster
- **Master:** Central control plane, provides unified view of the cluster
- **etcd:** distributed key-value store used to persist Kubernetes system state
- **Worker:** Docker host running *kubelet* (node agent) and *proxy* services
 - Runs pods and containers
 - Monitored by *systemd* (CentOS) or *monit* (Debian)





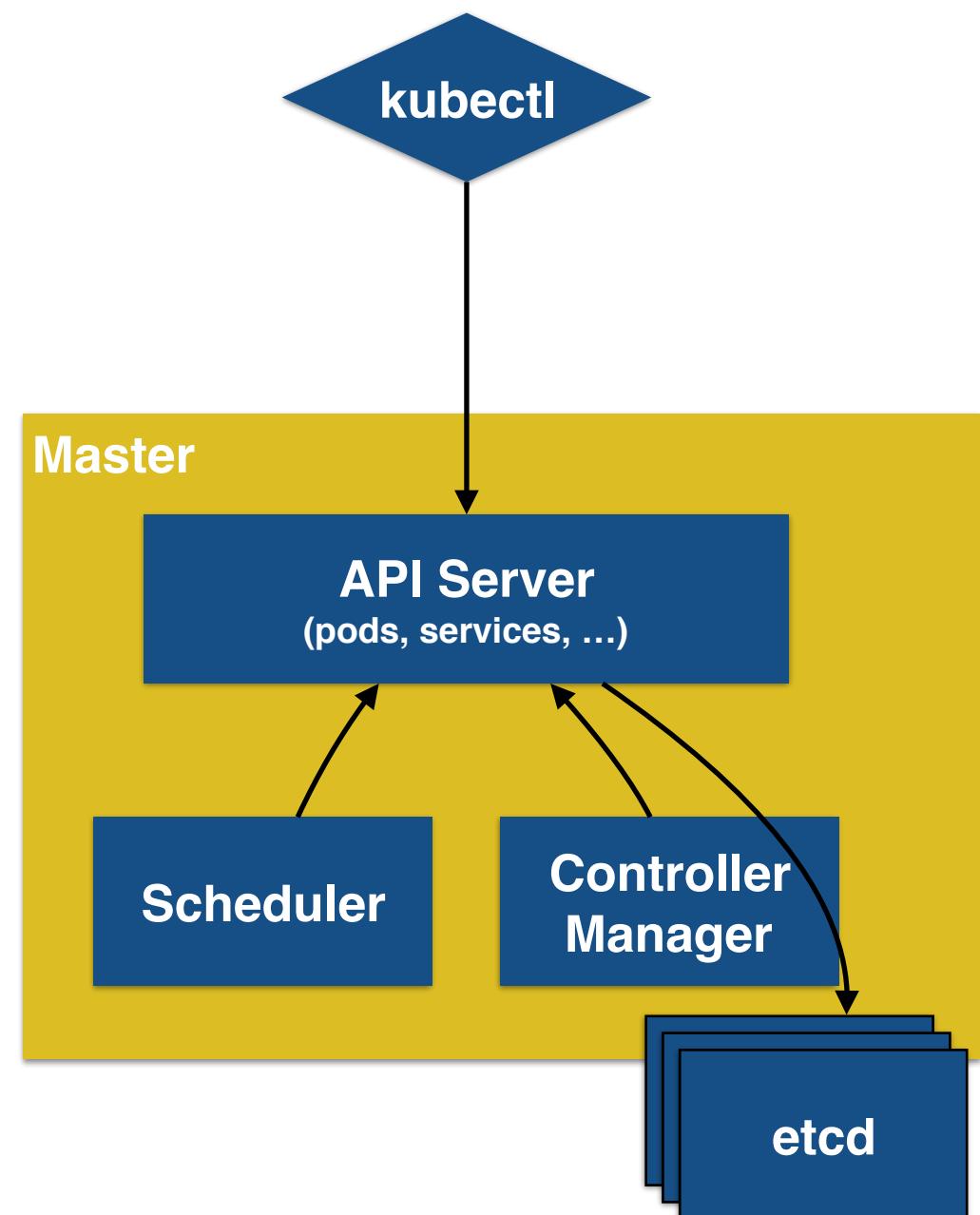
#Devoxx #containers

@arungupta

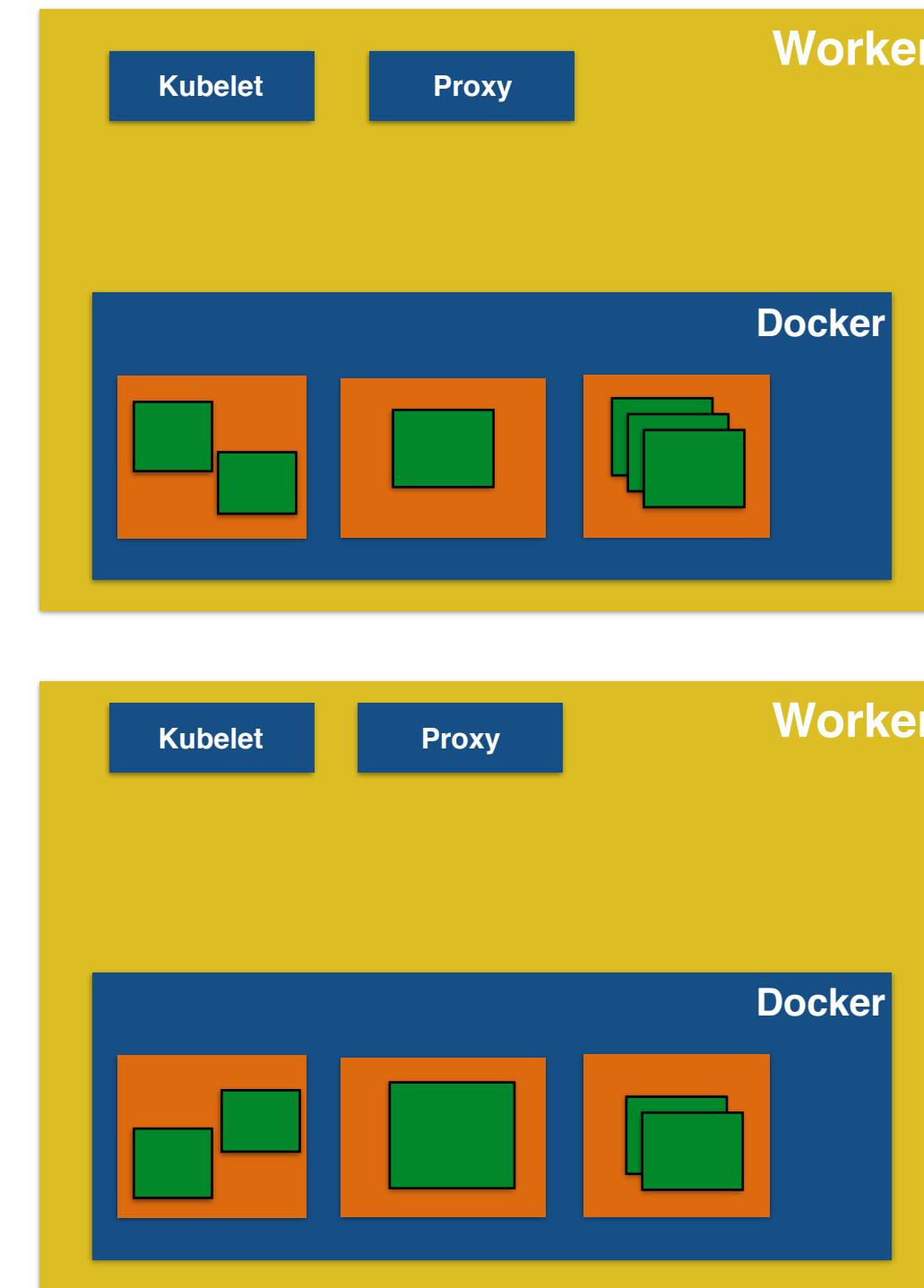
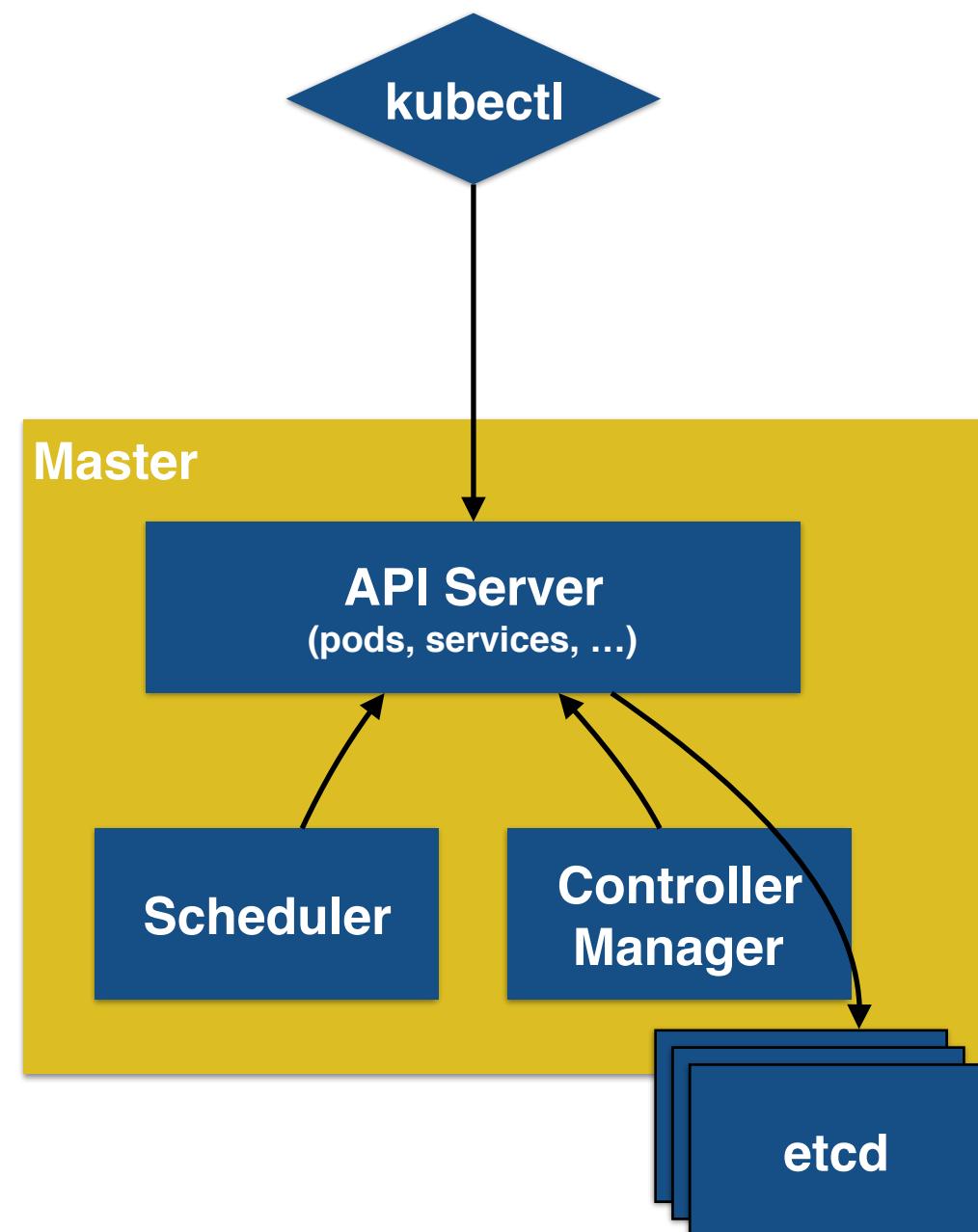


TM

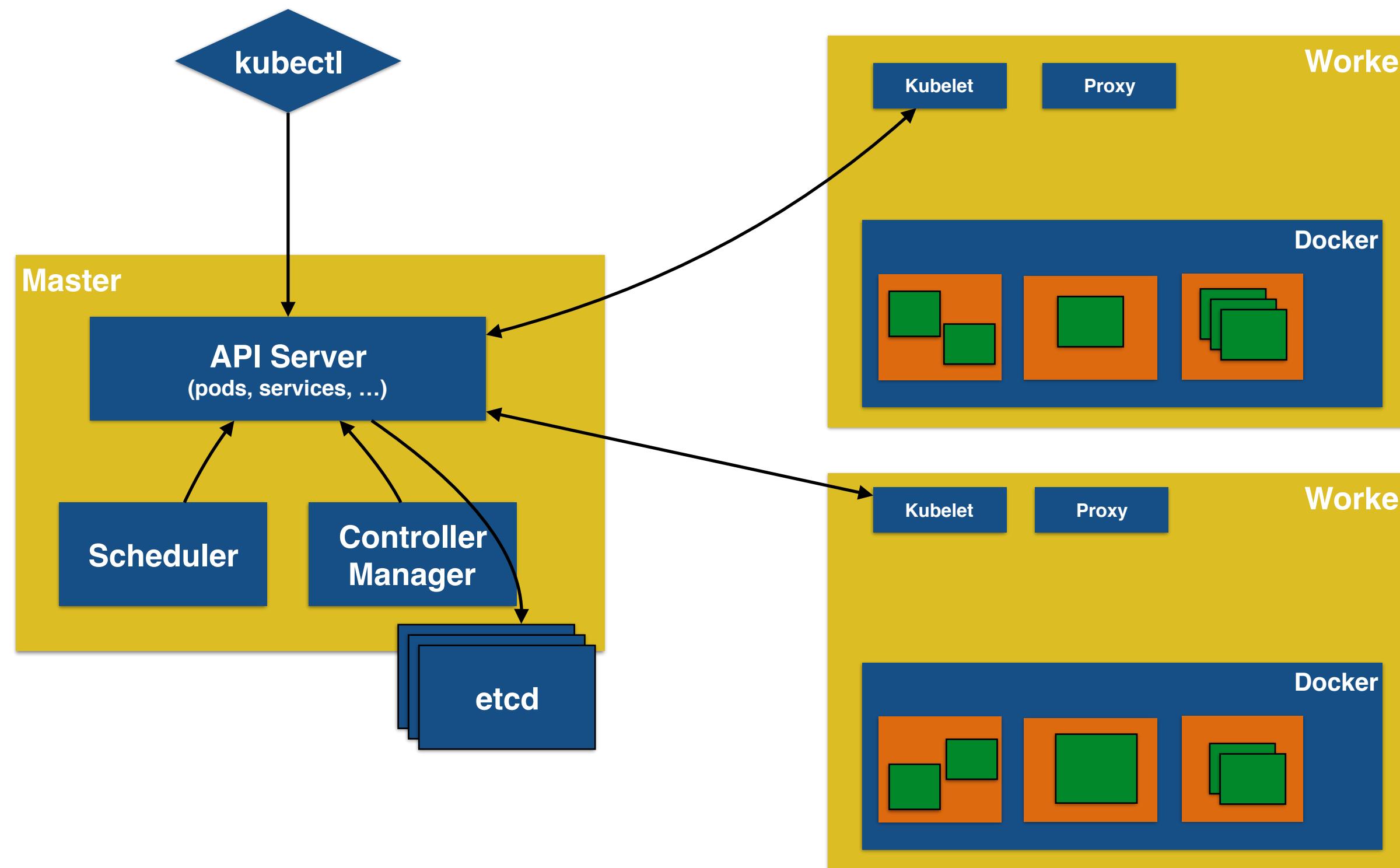
Devoxx 2016



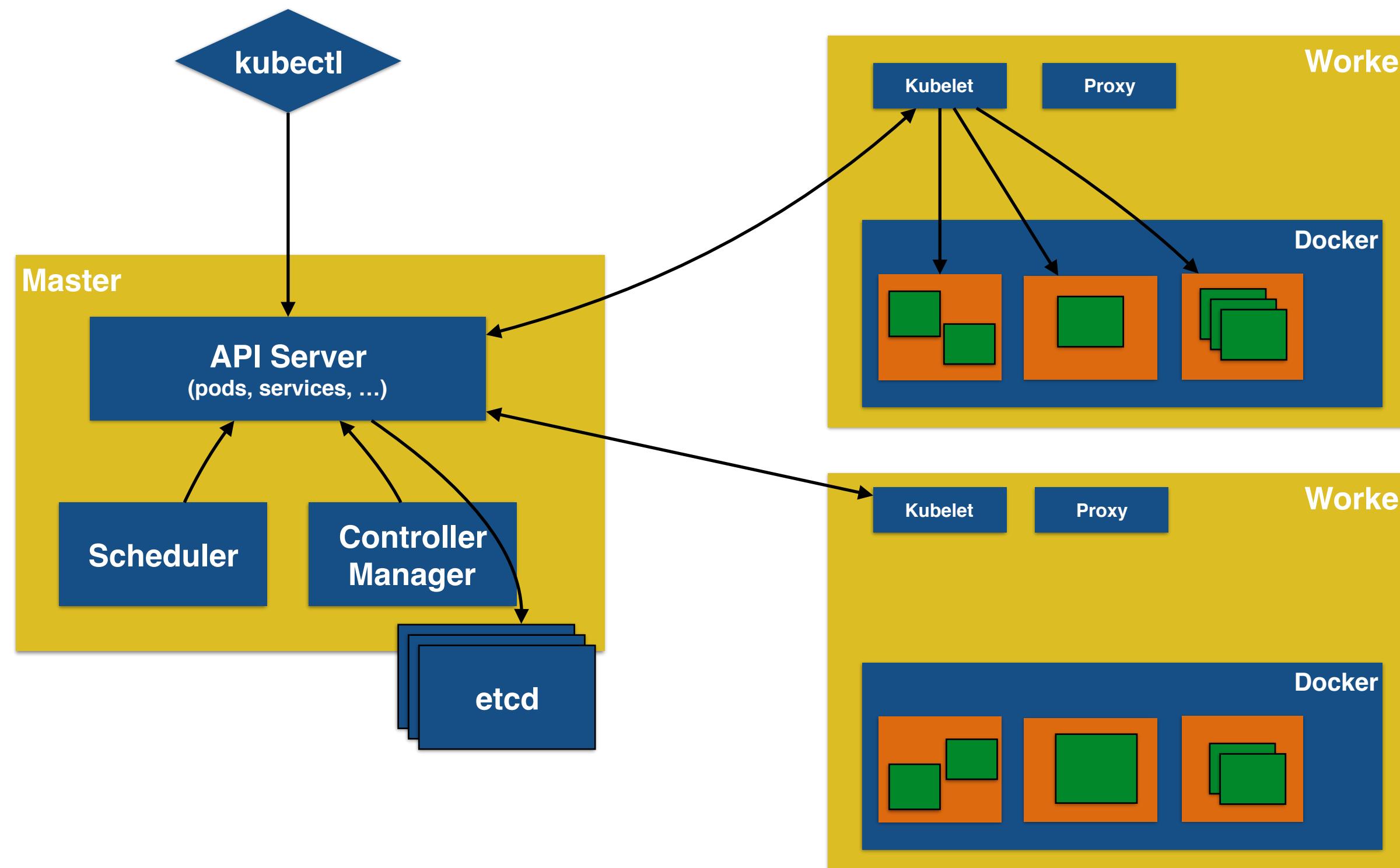
TM



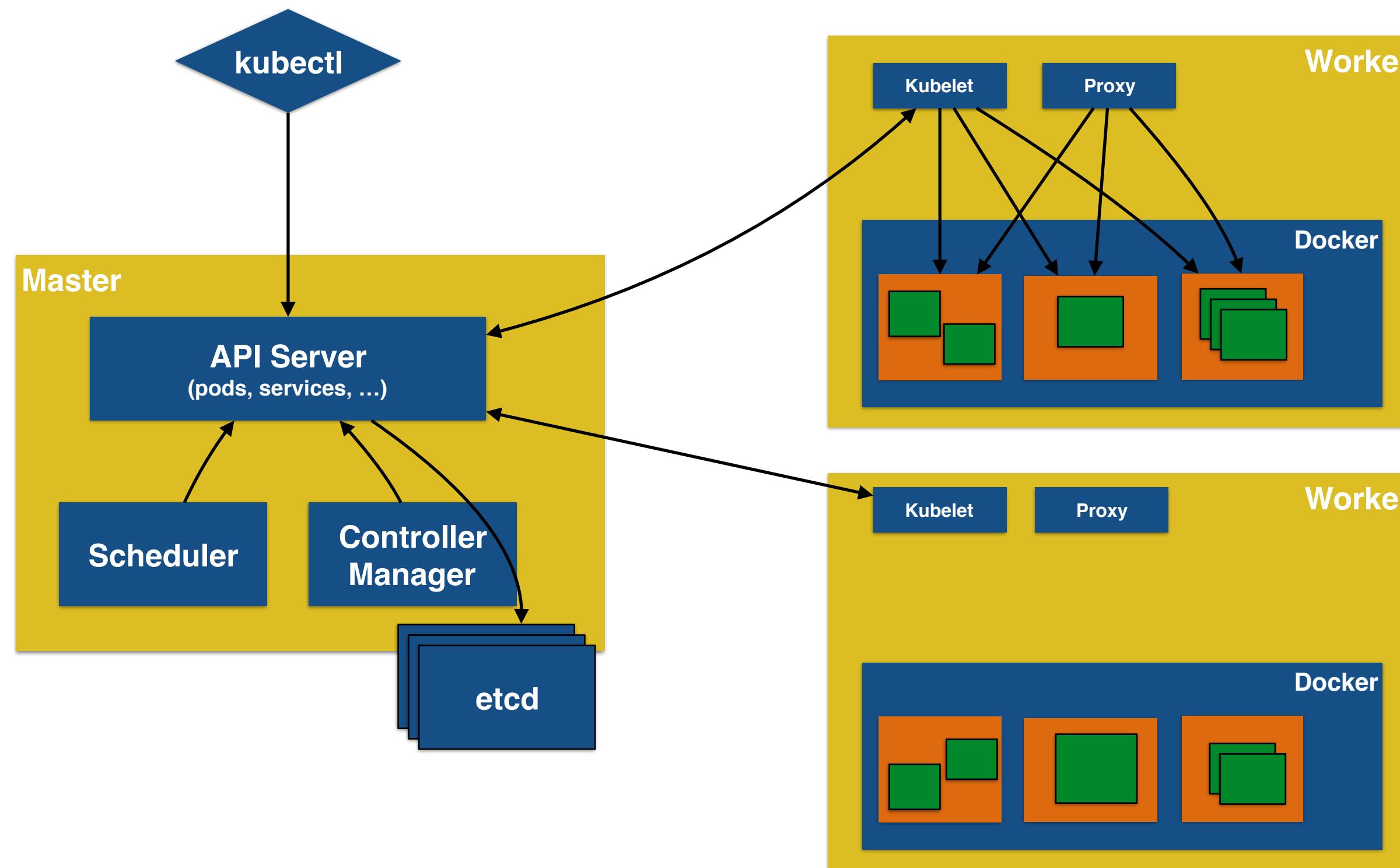
Devoxx

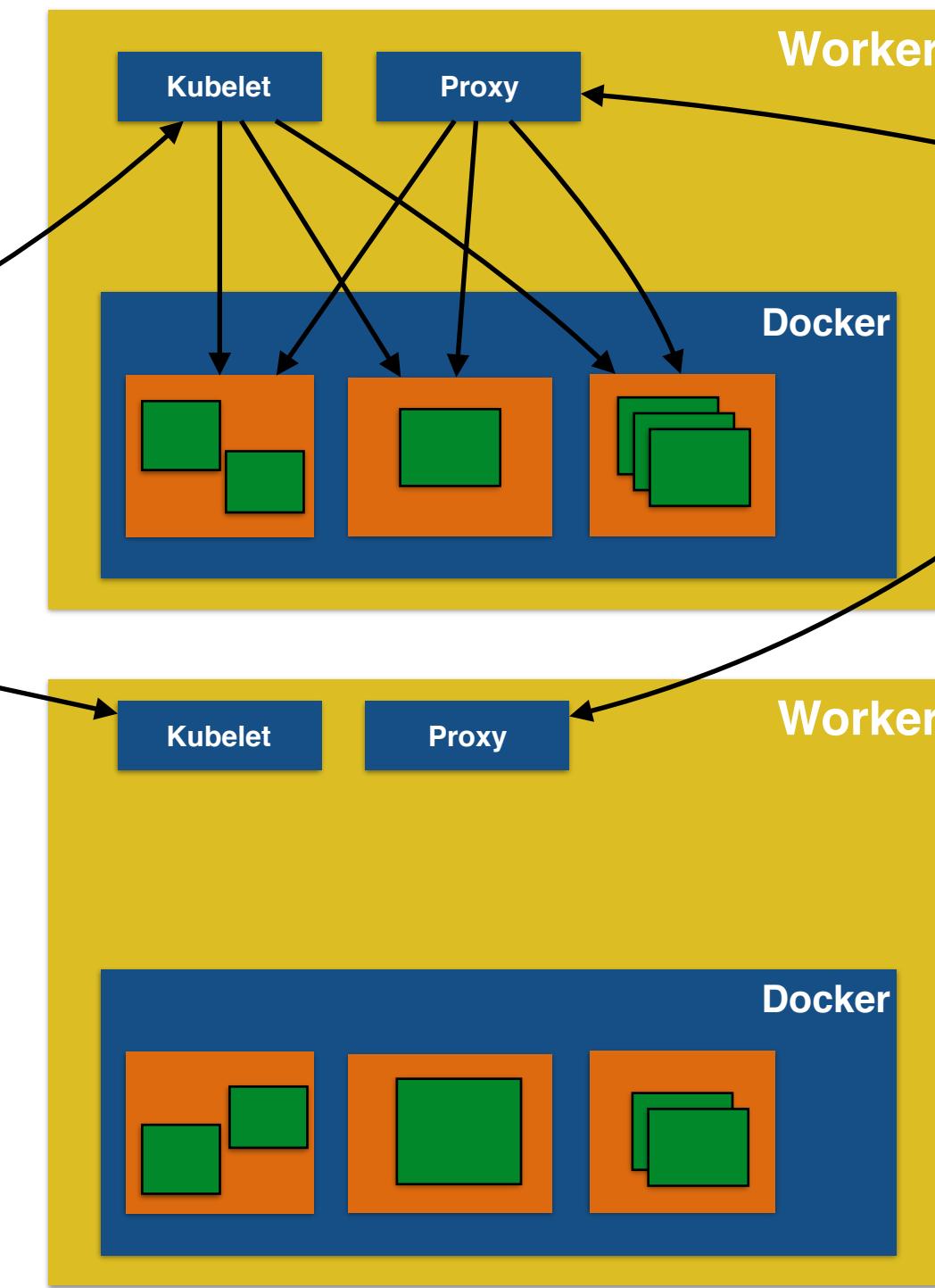
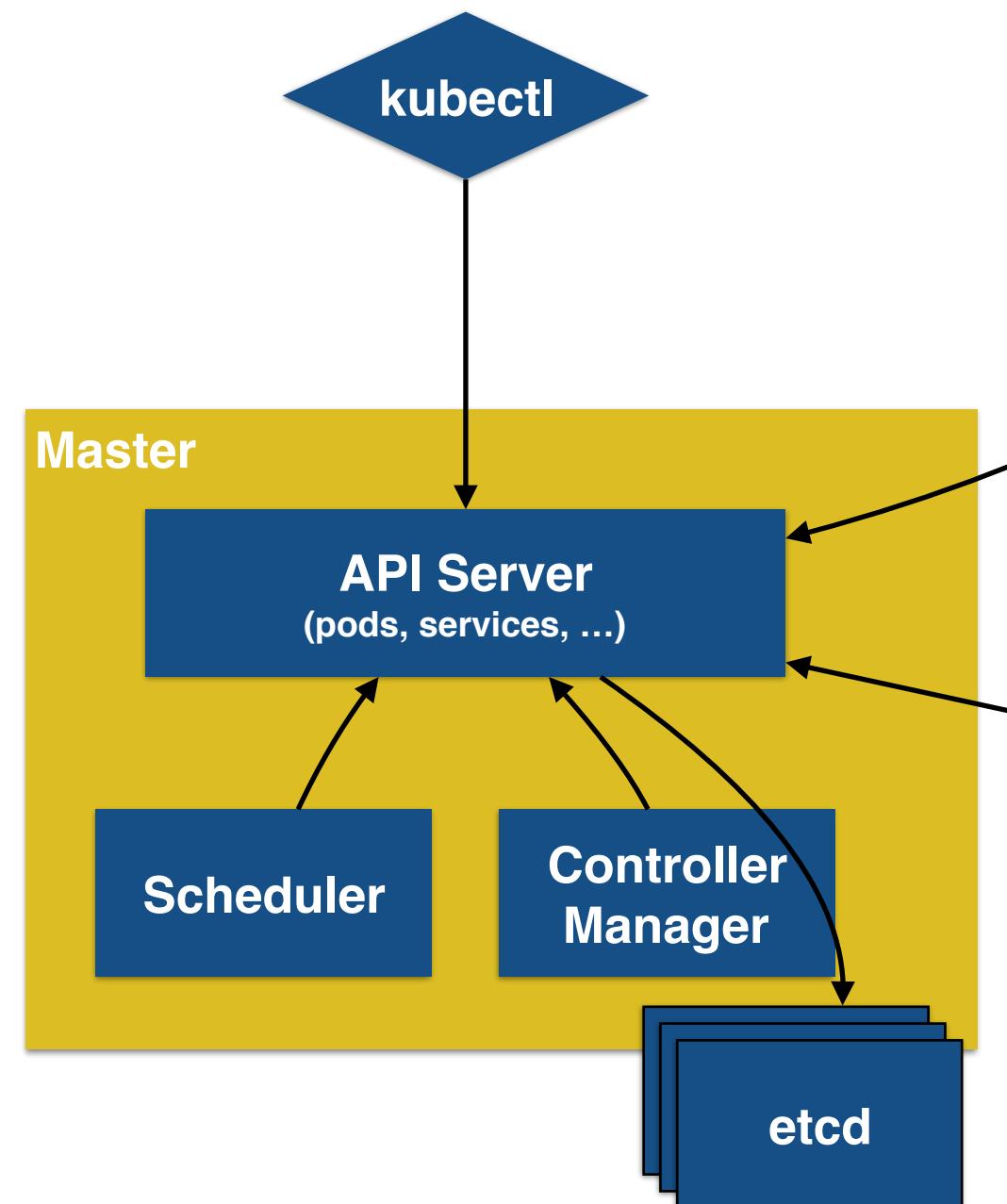


TM

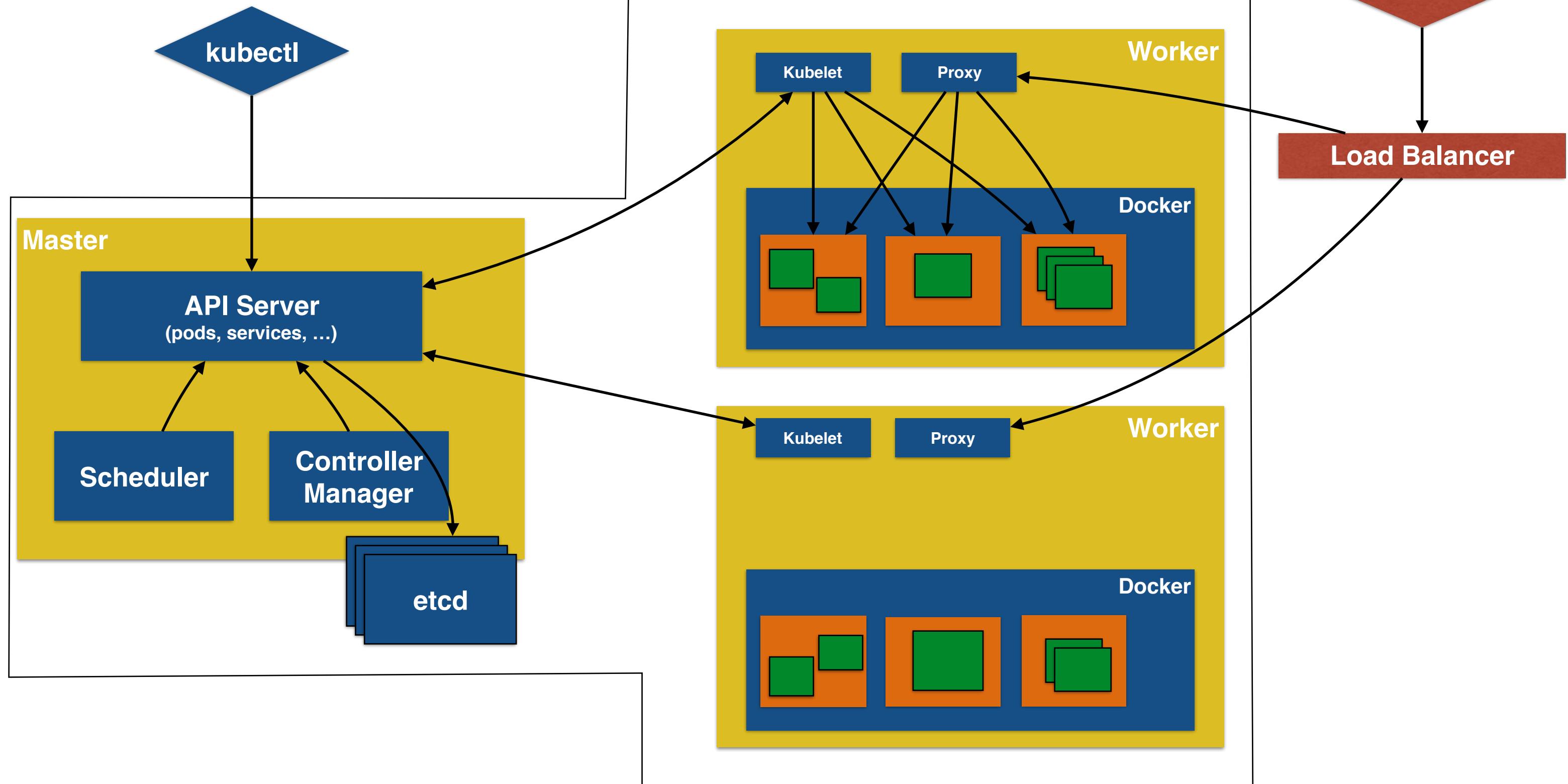


TM





Kubernetes Cluster



TM
X
X
O
V
>
S
D

Minikube



minikube



Minikube



minikube

- Runs a single node cluster in a VM



Minikube



minikube

- Runs a single node cluster in a VM
- Targeted for local development



Minikube



minikube

- Runs a single node cluster in a VM
- Targeted for local development
- minikube start, stop, docker-env, ...



Minikube



minikube

- Runs a single node cluster in a VM
- Targeted for local development
- minikube start, stop, docker-env, ...
- Requires kubectl CLI



Minikube



minikube

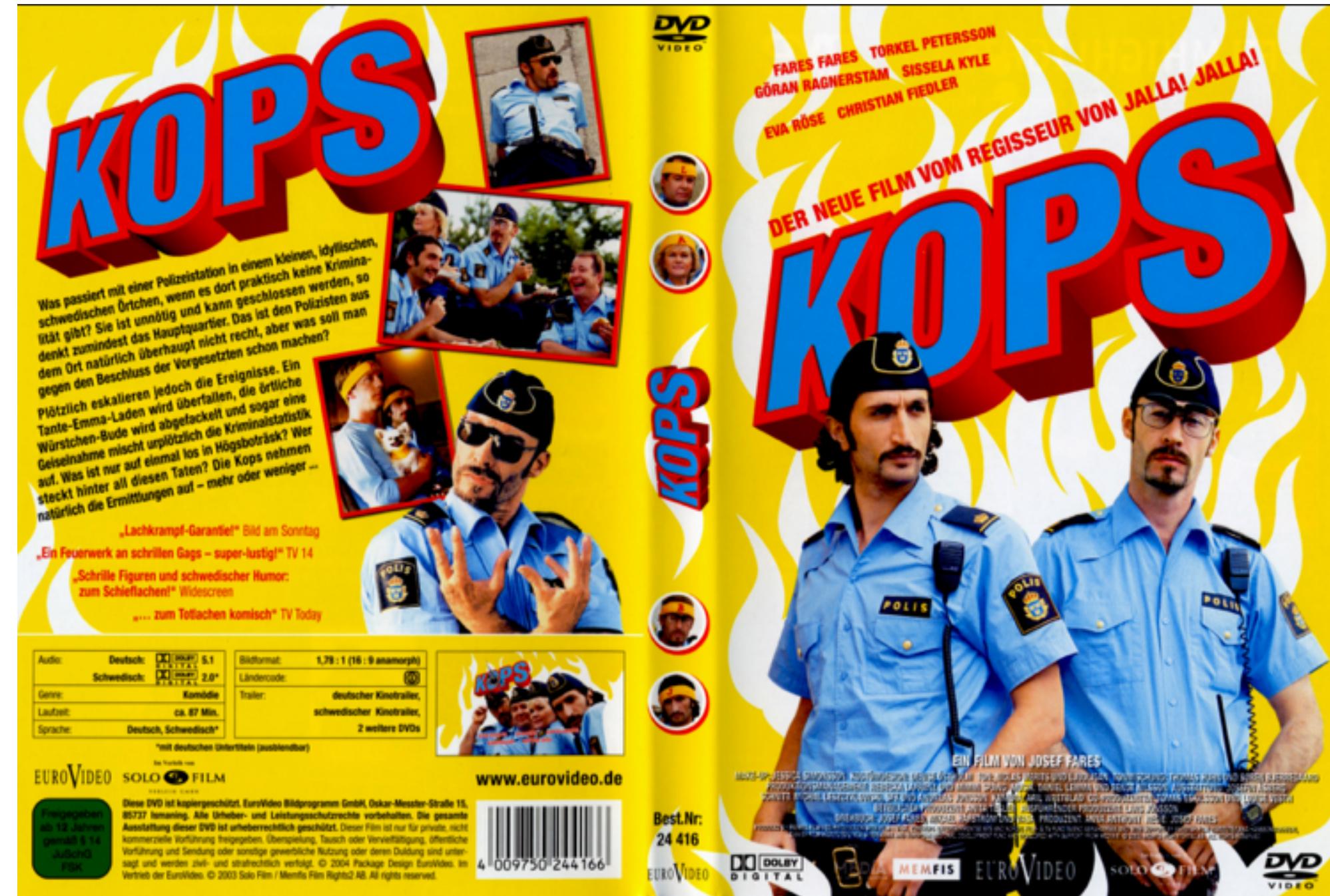
- Runs a single node cluster in a VM
- Targeted for local development
- minikube start, stop, docker-env, ...
- Requires kubectl CLI
- github.com/kubernetes/minikube/releases

<http://blog.couchbase.com/2016/september/minikube-rapid-dev--testing-kubernetes>

TM
X
O
>
U

Devoxx

Multimaster Kubernetes



TM



#Devoxx #

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-master-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-master-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-master-pod
13  spec:
14    containers:
15      - name: couchbase-master
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: MASTER
20        ports:
21          - containerPort: 8091
22 ---
23 apiVersion: v1
24 kind: Service
25 metadata:
26   name: couchbase-master-service
27   labels:
28     app: couchbase-master-service
29 spec:
30   ports:
31     - port: 8091
32   selector:
33     app: couchbase-master-pod
34   type: LoadBalancer
```

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-worker-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-worker-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-worker-pod
13  spec:
14    containers:
15      - name: couchbase-worker
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: "WORKER"
20          - name: COUCHBASE_MASTER
21            value: "couchbase-master-service"
22          - name: AUTO_REBALANCE
23            value: "false"
24   ports:
25     - containerPort: 8091
```

kubectl
scale
--replicas=3
couchbase-worker

@arungupta

TM



```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-master-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-master-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-master-pod
13  spec:
14    containers:
15      - name: couchbase-master
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: MASTER
20        ports:
21          - containerPort: 8091
22 ---
23 apiVersion: v1
24 kind: Service
25 metadata:
26   name: couchbase-master-service
27   labels:
28     app: couchbase-master-service
29 spec:
30   ports:
31     - port: 8091
32   selector:
33     app: couchbase-master-pod
34   type: LoadBalancer
```

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-worker-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-worker-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-worker-pod
13  spec:
14    containers:
15      - name: couchbase-worker
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: "WORKER"
20          - name: COUCHBASE_MASTER
21            value: "couchbase-master-service"
22          - name: AUTO_REBALANCE
23            value: "false"
24   ports:
25     - containerPort: 8091
```

kubectl
scale
--replicas=3
couchbase-worker

TM



#Devoxx #

```

1  apiVersion: v1
2  kind: ReplicationController
3
4    name: couchbase-master-rc
5
6  spec:
7    replicas: 1
8    selector:
9      app: couchbase-master-pod
10   template:
11     metadata:
12       labels:
13         app: couchbase-master-pod
14   spec:
15     containers:
16       - name: couchbase-master
17         image: arungupta/couchbase:k8s
18         env:
19           - name: TYPE
20             value: MASTER
21         ports:
22           - containerPort: 8091
23
24  ---
25
26  apiVersion: v1
27  kind: Service
28  metadata:
29    name: couchbase-master-service
30    labels:
31      app: couchbase-master-service
32  spec:
33    ports:
34      - port: 8091
35    selector:
36      app: couchbase-master-pod
37      type: LoadBalancer

```

```

1  apiVersion: v1
2  kind: ReplicationController
3
4    name: couchbase-worker-rc
5
6  spec:
7    replicas: 1
8    selector:
9      app: couchbase-worker-pod
10   template:
11     metadata:
12       labels:
13         app: couchbase-worker-pod
14   spec:
15     containers:
16       - name: couchbase-worker
17         image: arungupta/couchbase:k8s
18         env:
19           - name: TYPE
20             value: "WORKER"
21           - name: COUCHBASE_MASTER
22             value: "couchbase-master-service"
23           - name: AUTO_REBALANCE
24             value: "false"
25         ports:
26           - containerPort: 8091

```

kubectl
scale
--replicas=3
couchbase-worker

@arungupta

TM



```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-master-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-master-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-master-pod
13  spec:
14    containers:
15      - name: couchbase-master
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: MASTER
20        ports:
21          - containerPort: 8091
22 ---
23 apiVersion: v1
24 kind: Service
25 metadata:
26   name: couchbase-master-service
27   labels:
28     app: couchbase-master-service
29 spec:
30   ports:
31     - port: 8091
32   selector:
33     app: couchbase-master-pod
34   type: LoadBalancer
```

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-worker-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-worker-pod
9   template:
10  metadata:
11    labels:
12      app: couchbase-worker-pod
13  spec:
14    containers:
15      - name: couchbase-worker
16        image: arungupta/couchbase:k8s
17        env:
18          - name: TYPE
19            value: "WORKER"
20          - name: COUCHBASE_MASTER
21            value: "couchbase-master-service"
22          - name: AUTO_REBALANCE
23            value: "false"
24   ports:
25     - containerPort: 8091
```

kubectl
scale
--replicas=3
couchbase-worker



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

```
1 apiVersion: v1
2 kind: ReplicationController
3 metadata:
4   name: couchbase-worker-rc
5 spec:
6   replicas: 1
7   selector:
8     app: couchbase-worker-pod
9   template:
10    metadata:
11      labels:
12        app: couchbase-worker-pod
13   spec:
14     containers:
15       - name: couchbase-worker
16         image: arungupta/couchbase:k8s
17         env:
18           - name: TYPE
19             value: "WORKER"
20           - name: COUCHBASE_MASTER
21             value: "couchbase-master-service"
22           - name: AUTO_REBALANCE
23             value: "false"
24     ports:
25       - containerPort: 8091
```

```
kuectl  
scale  
--replicas=3  
couchbase-worker
```

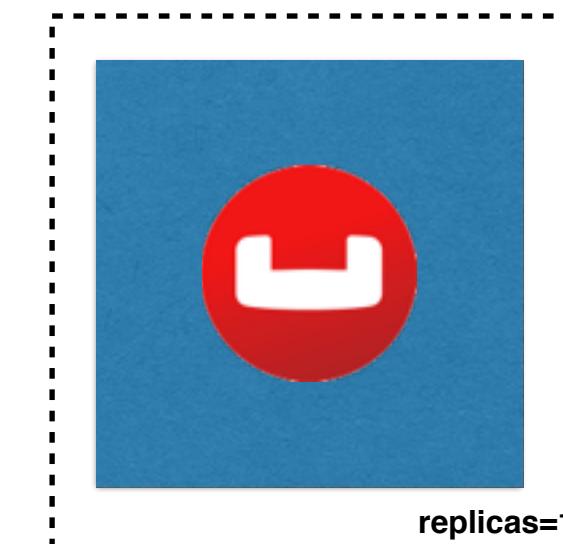


Couchbase in Kubernetes

<http://blog.kubernetes.io/2016/08/create-couchbase-cluster-using-kubernetes.html>



Couchbase in Kubernetes

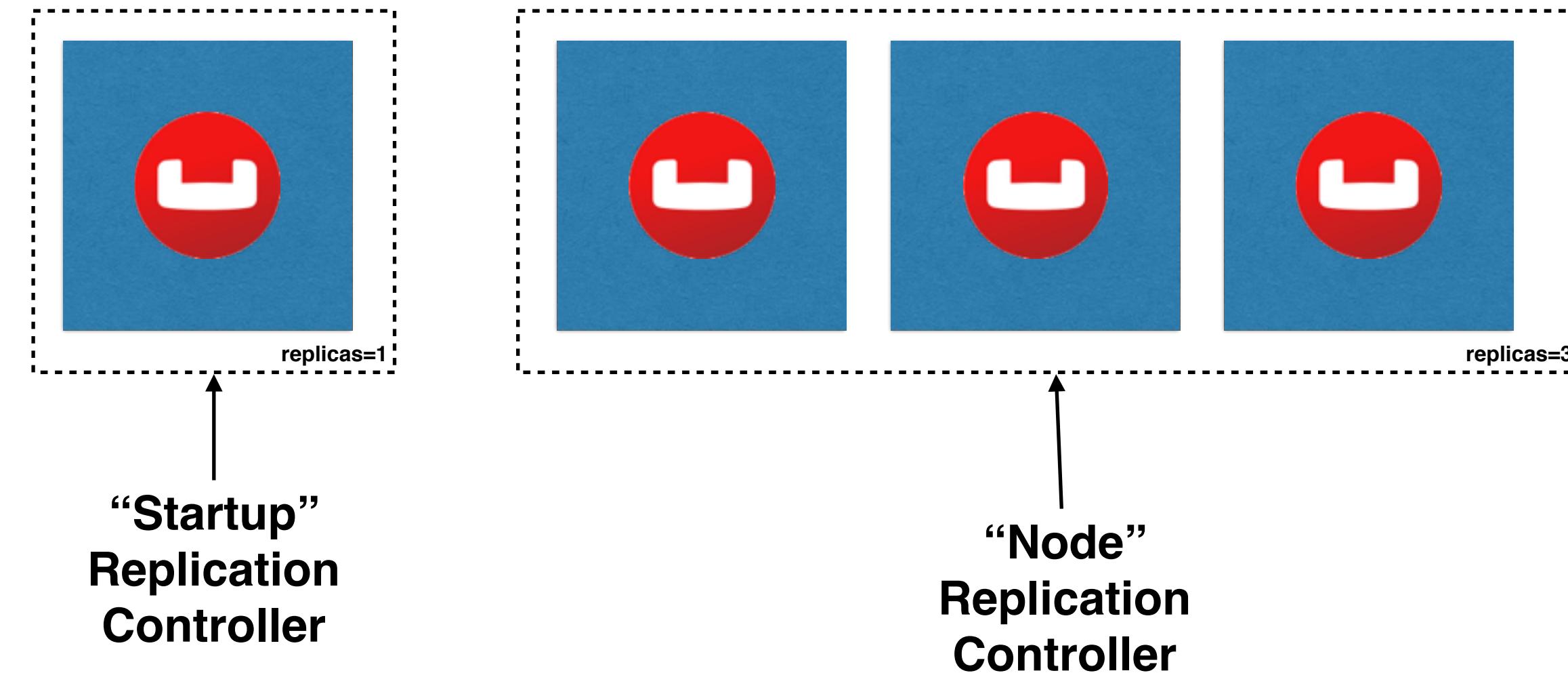


**“Startup”
Replication
Controller**

<http://blog.kubernetes.io/2016/08/create-couchbase-cluster-using-kubernetes.html>

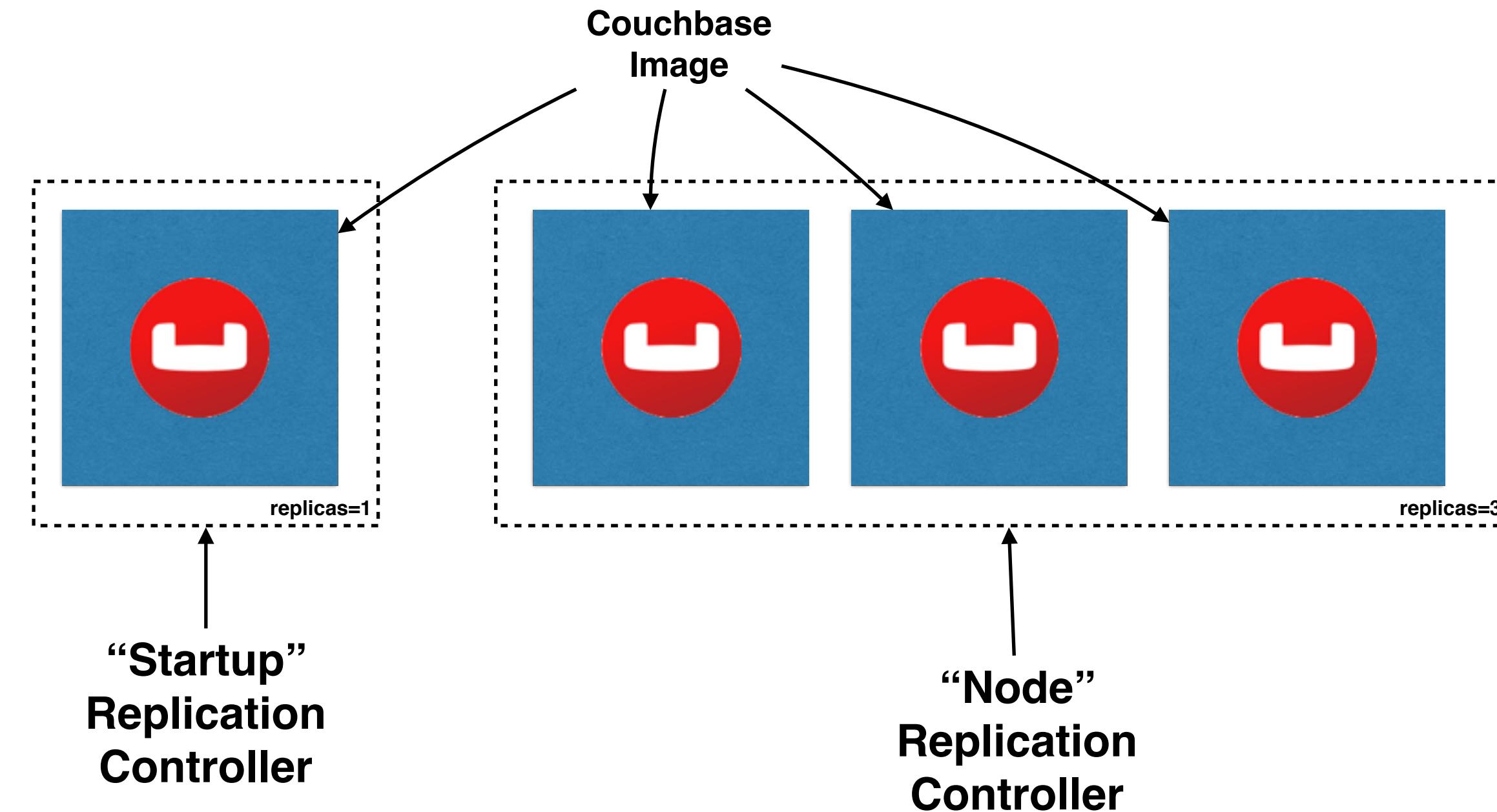
TM

Couchbase in Kubernetes



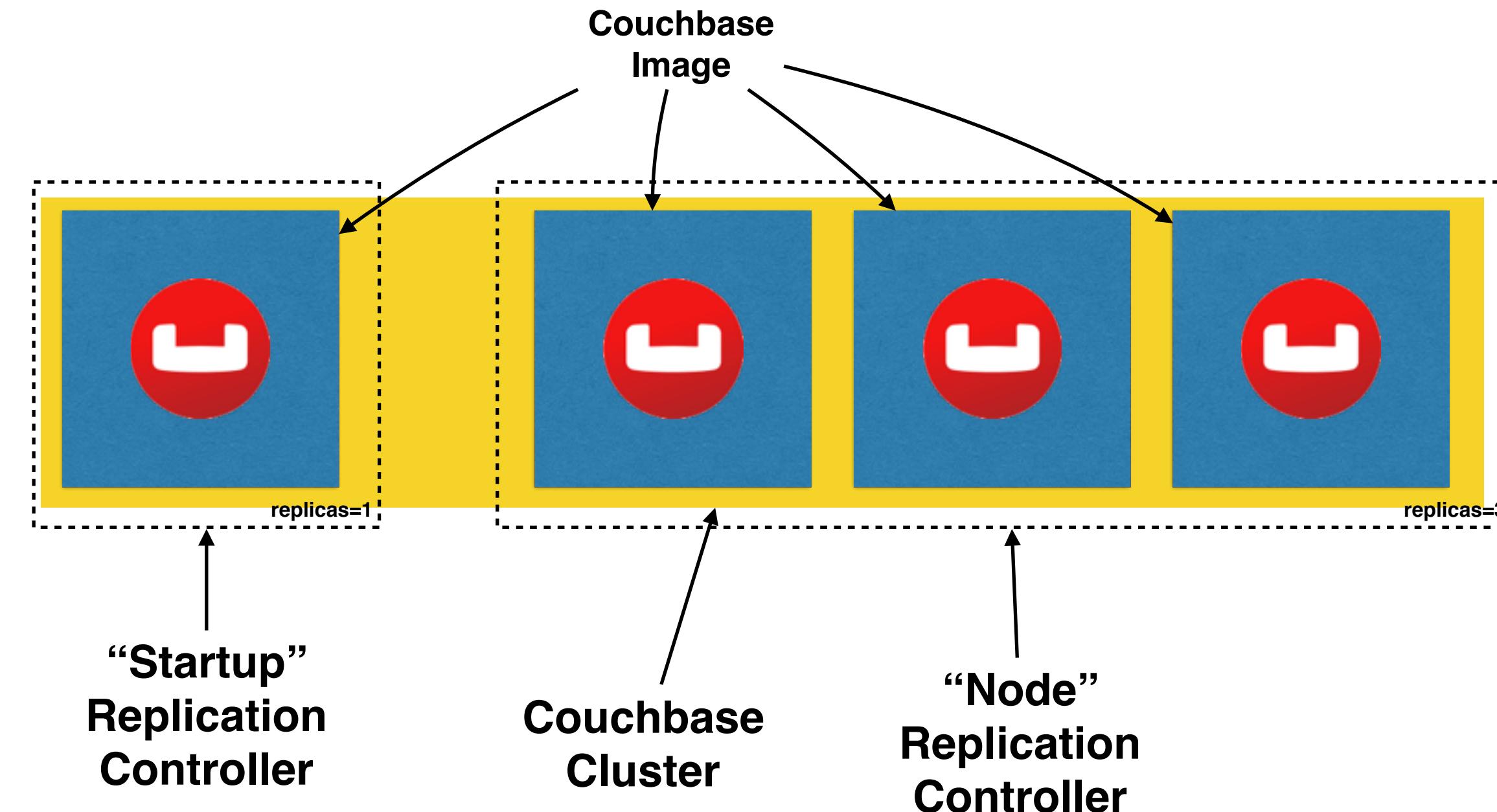
<http://blog.kubernetes.io/2016/08/create-couchbase-cluster-using-kubernetes.html>

Couchbase in Kubernetes



<http://blog.kubernetes.io/2016/08/create-couchbase-cluster-using-kubernetes.html>

Couchbase in Kubernetes



<http://blog.kubernetes.io/2016/08/create-couchbase-cluster-using-kubernetes.html>



CPU usage history

A line chart titled "CPU usage history" showing CPU usage in cores over time. The Y-axis ranges from 0 to 1.13 cores, and the X-axis shows times from 16:04 to 16:06. The green line starts at 1.00 core at 16:04, dips slightly to 0.95 at 16:05, and then rises back to 1.00 at 16:06.

Memory usage history

A line chart titled "Memory usage history" showing memory usage in bytes over time. The Y-axis ranges from 0 to 1.89 GiB, and the X-axis shows times from 16:03 to 16:06. The blue line starts at 0.429 MiB at 16:03, rises steadily to 0.858 MiB at 16:04, 1.26 GiB at 16:05, and reaches 1.68 GiB at 16:06.

Replication controllers

Name	Labels	Pods	Age	Images
couchbase-master-rc	app: couchbase-master-pod	1 / 1	3 minutes	arungupta/couchbase:k8s
couchbase-worker-rc	app: couchbase-worker-pod	3 / 3	a minute	arungupta/couchbase:k8s

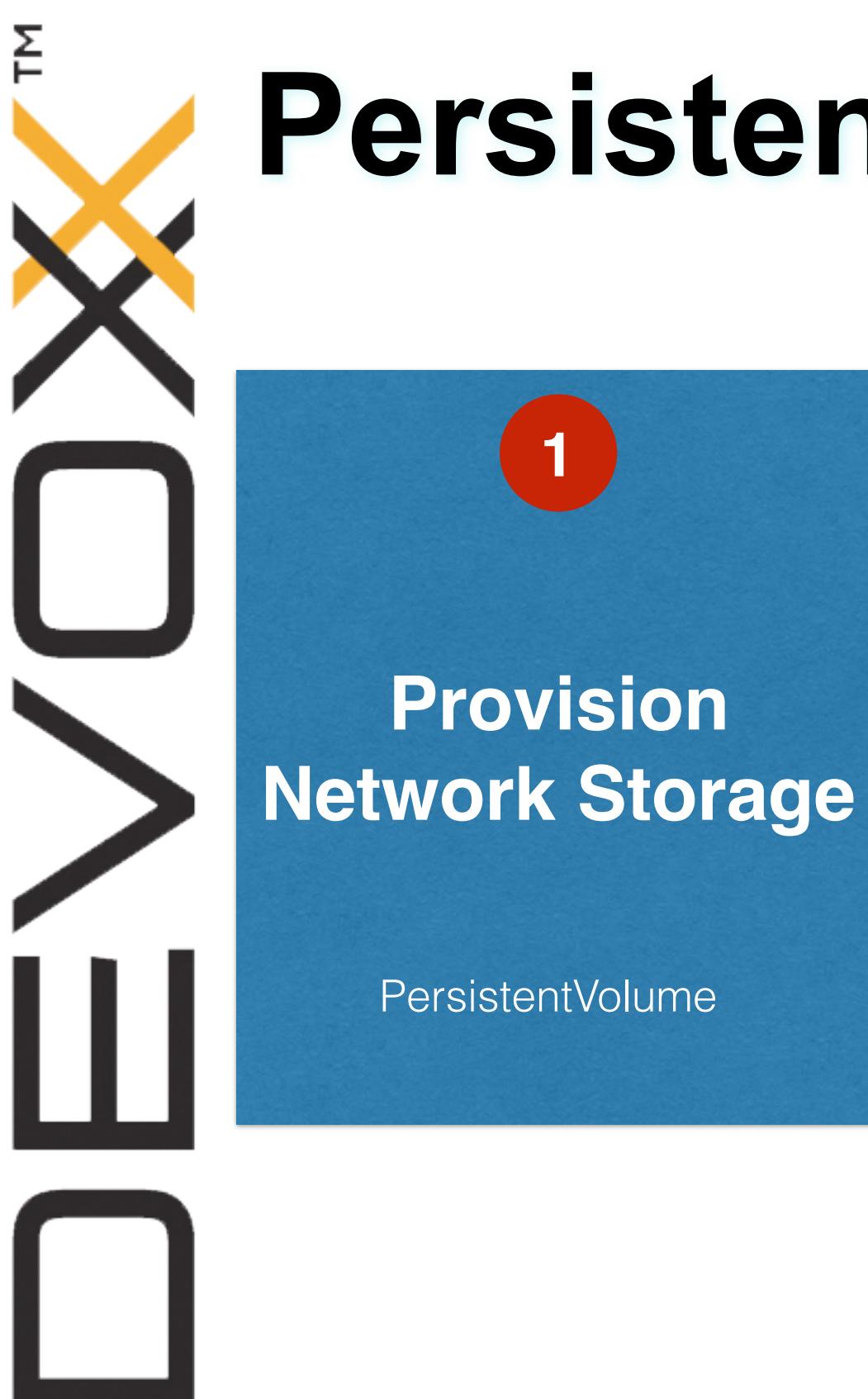
Pods

Name	Status	Restarts	Age	Cluster IP	CPU (cores)	Memory (bytes)	Actions	
couchbase-master-rc...	Running	0	3 minutes	10.244.1.3		1.024		708 Ki
couchbase-worker-rc-l...	Running	0	55 seconds	10.244.0.4		-		310.121 Mi
couchbase-worker-rc-...	Running	0	55 seconds	10.244.2.10		-		377.246 Mi
couchbase-worker-rc-...	Running	0	a minute	10.244.1.4		0.298		547.105 Mi

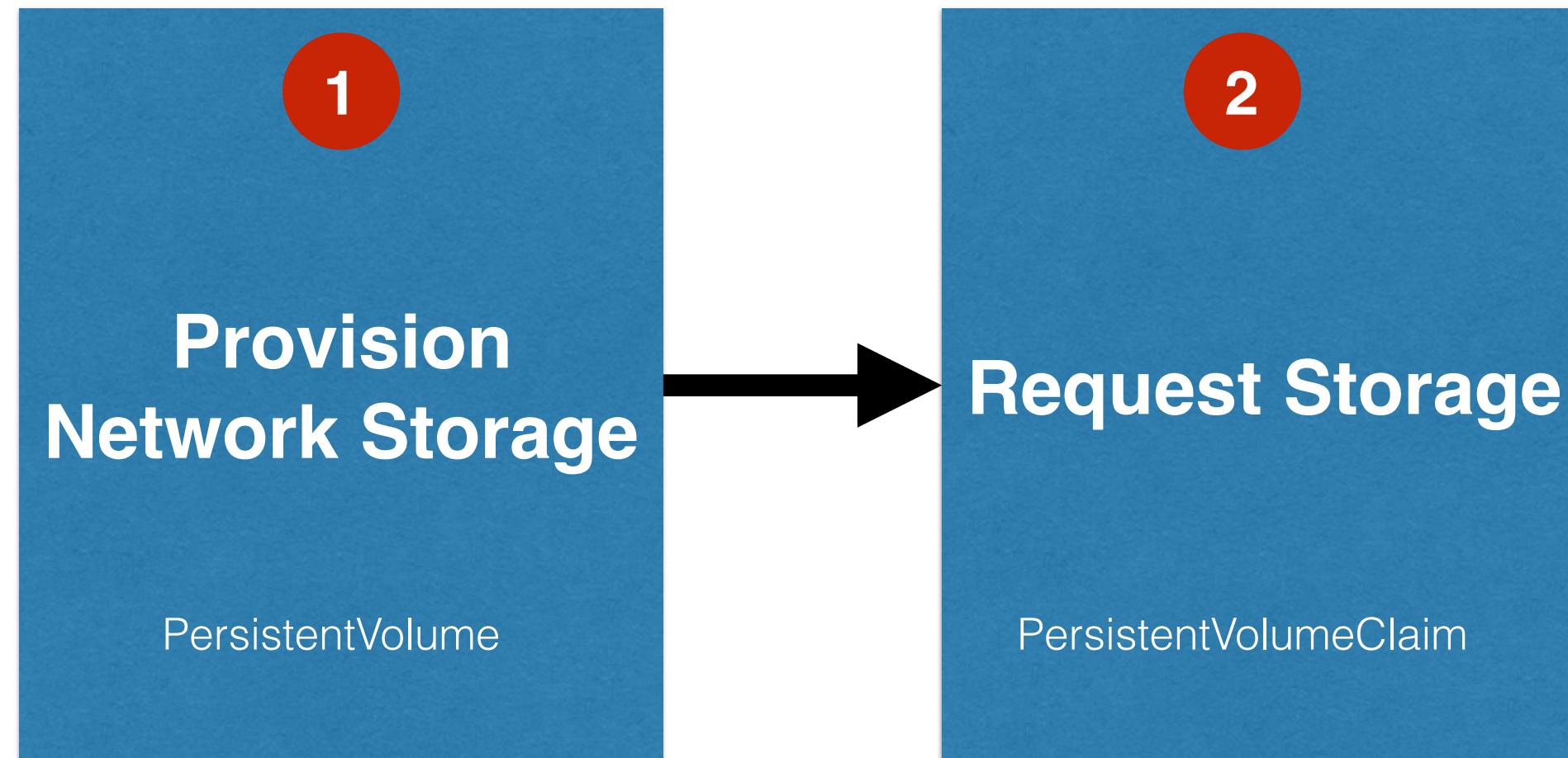
#Devoxx #containers

@arungupta

Persistent Volume

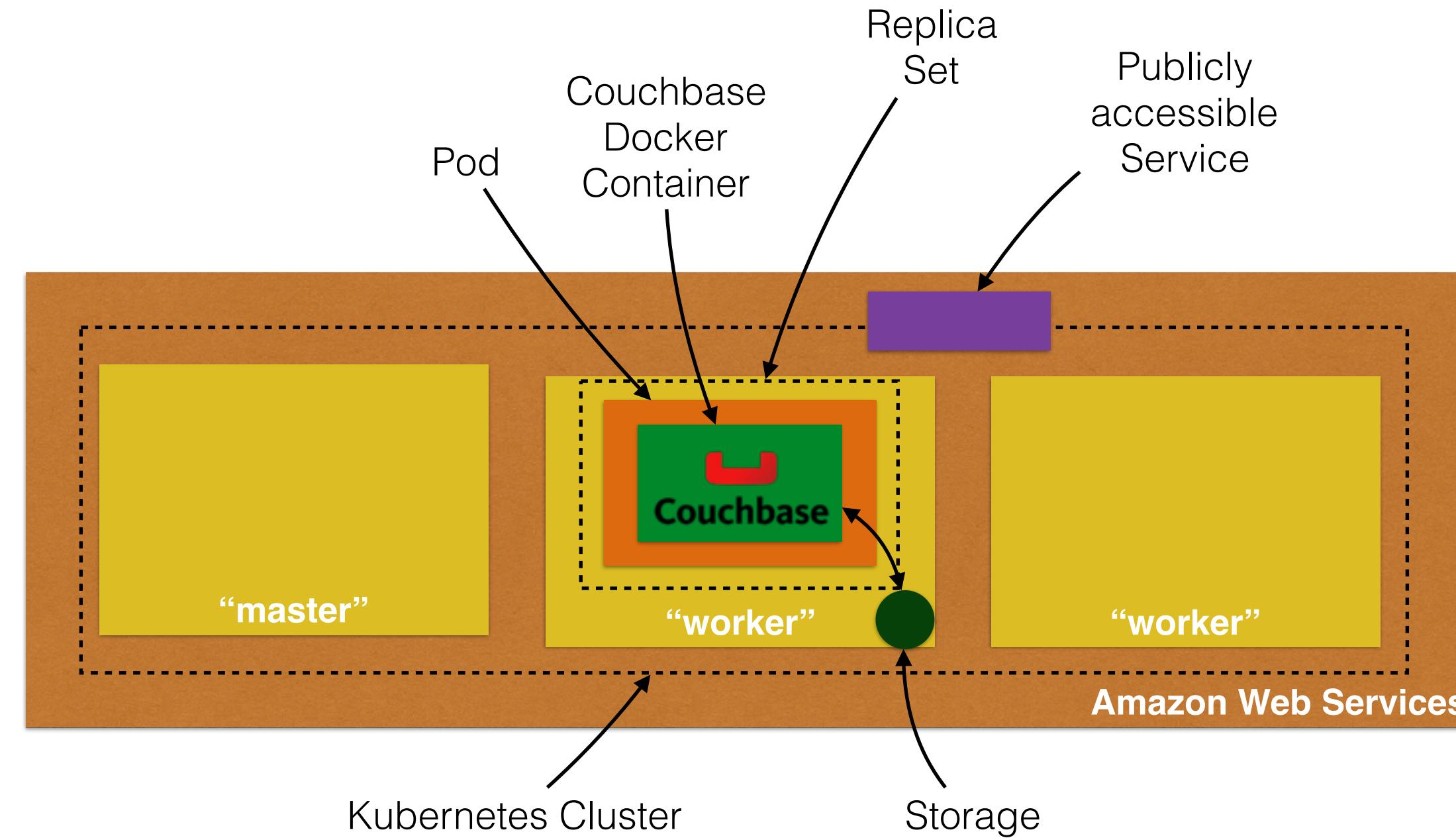


Persistent Volume

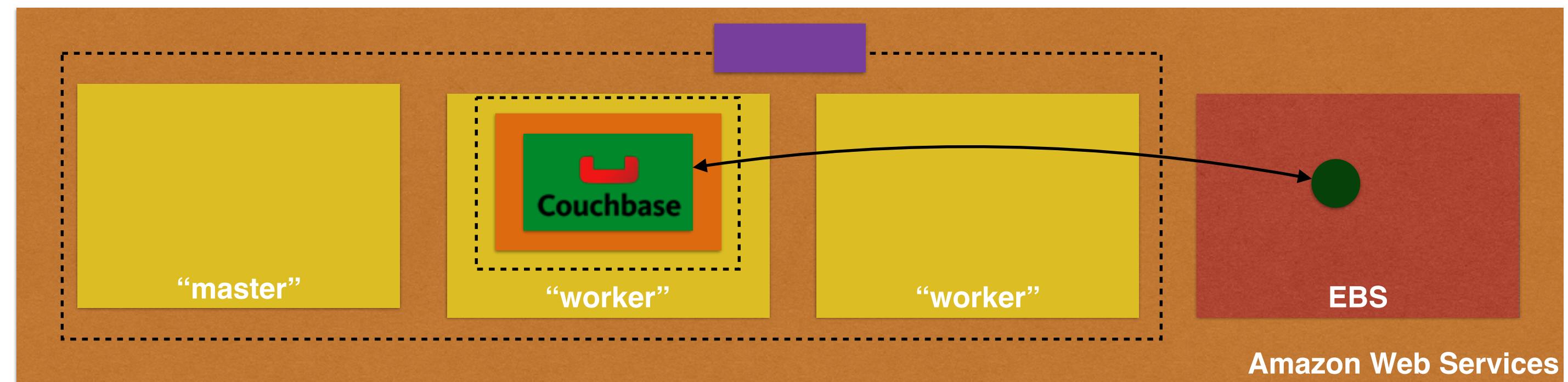


Persistent Volume



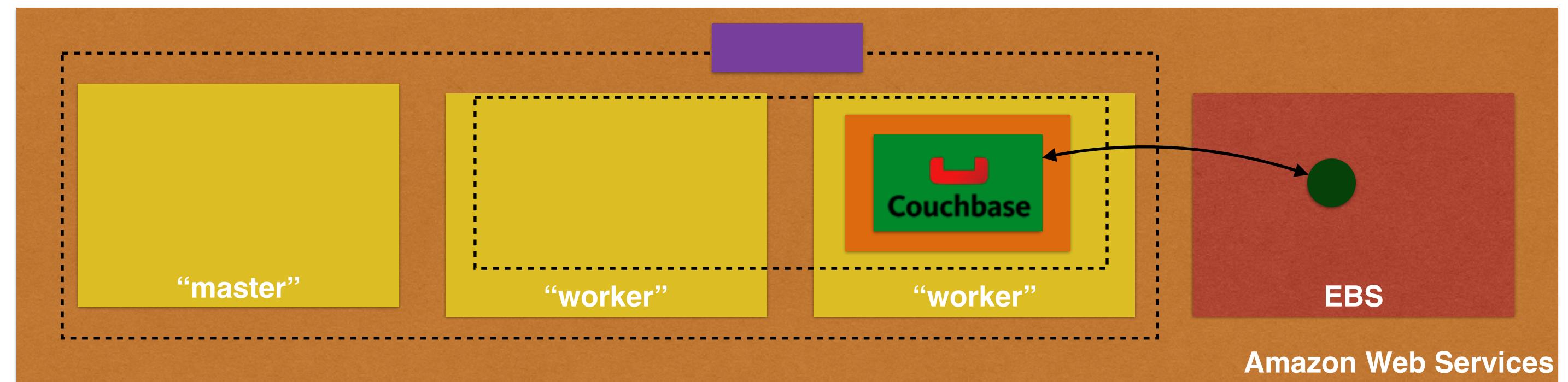


<http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs>



- Nodes and EBS should be on the same Region/AZ

<http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs>





Pet Set



Pet Set

- Alpha resource introduced in 1.3



Pet Set

- Alpha resource introduced in 1.3
- Stateful pods



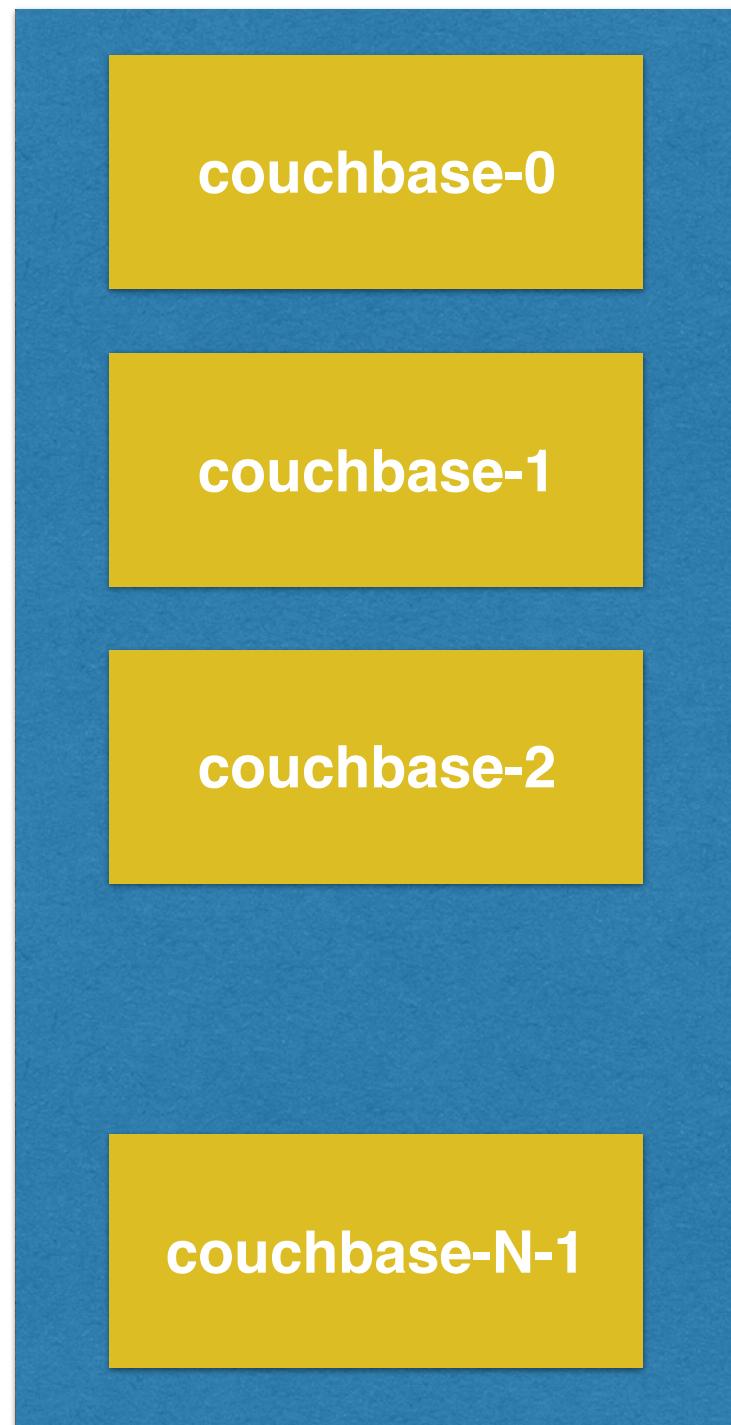
Pet Set

- Alpha resource introduced in 1.3
- Stateful pods
- PetSet has 0..N-1 Pets



Pet Set

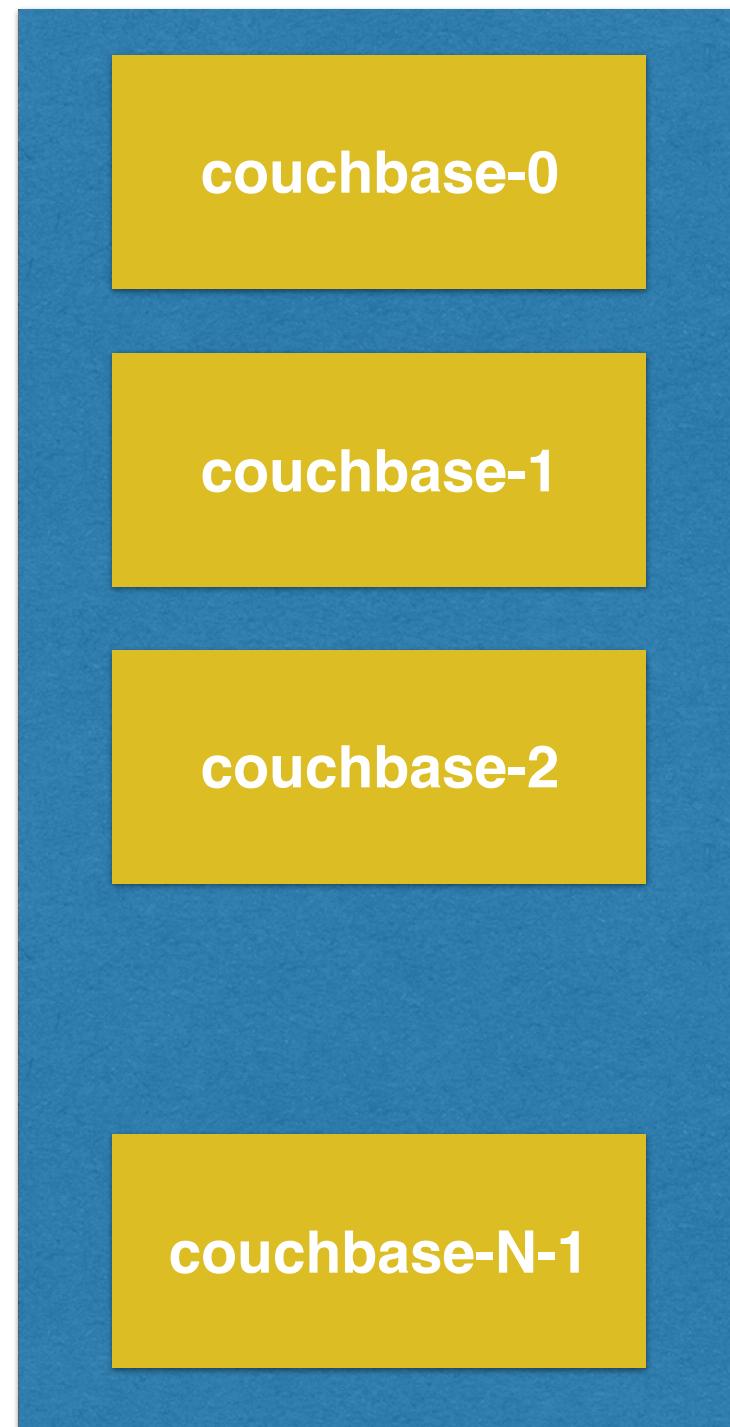
- Alpha resource introduced in 1.3
- Stateful pods
- PetSet has 0..N-1 Pets
- Pet has deterministic name, and a unique identity
 - stable hostname
 - ordinal index
 - stable storage linked to ordinal & hostname





Pet Set

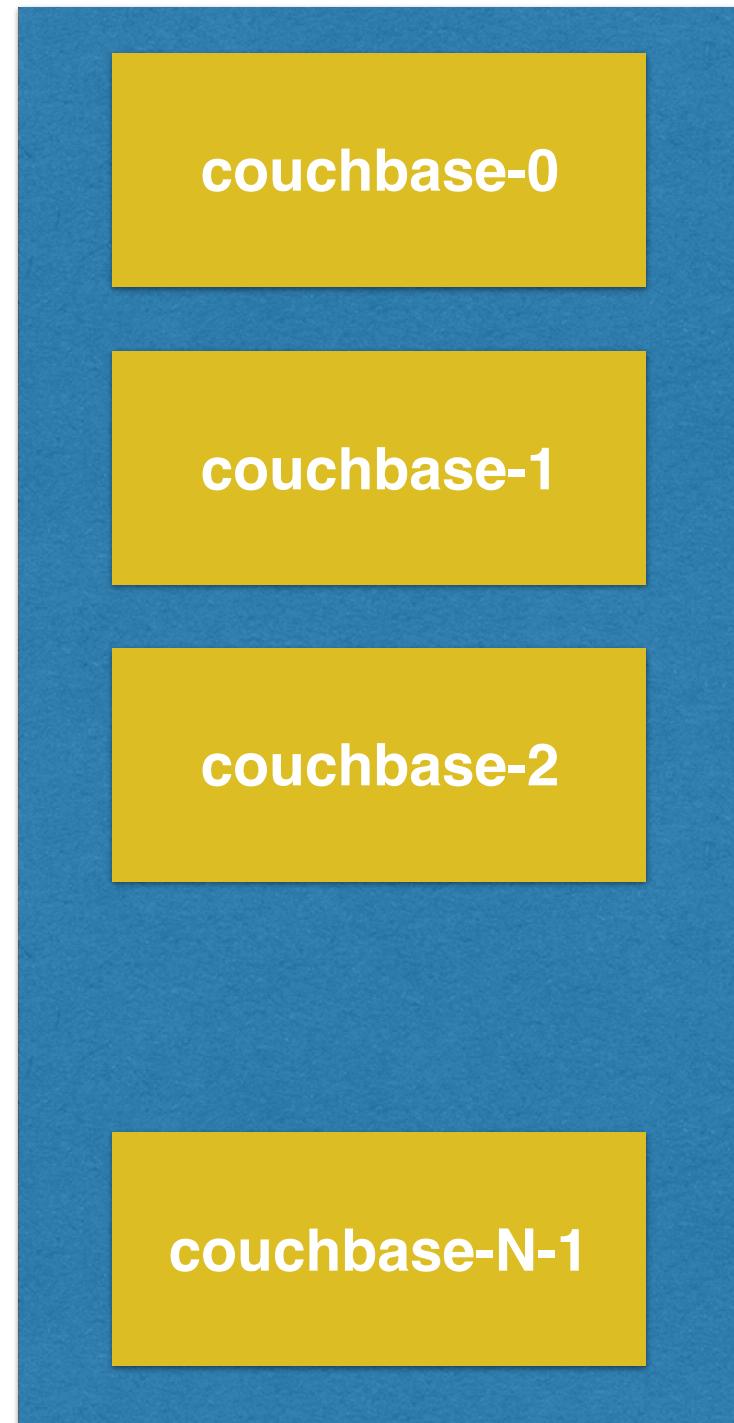
- Alpha resource introduced in 1.3
- Stateful pods
- PetSet has 0..N-1 Pets
- Pet has deterministic name, and a unique identity
 - stable hostname
 - ordinal index
 - stable storage linked to ordinal & hostname
- Each Pet has at most one pod





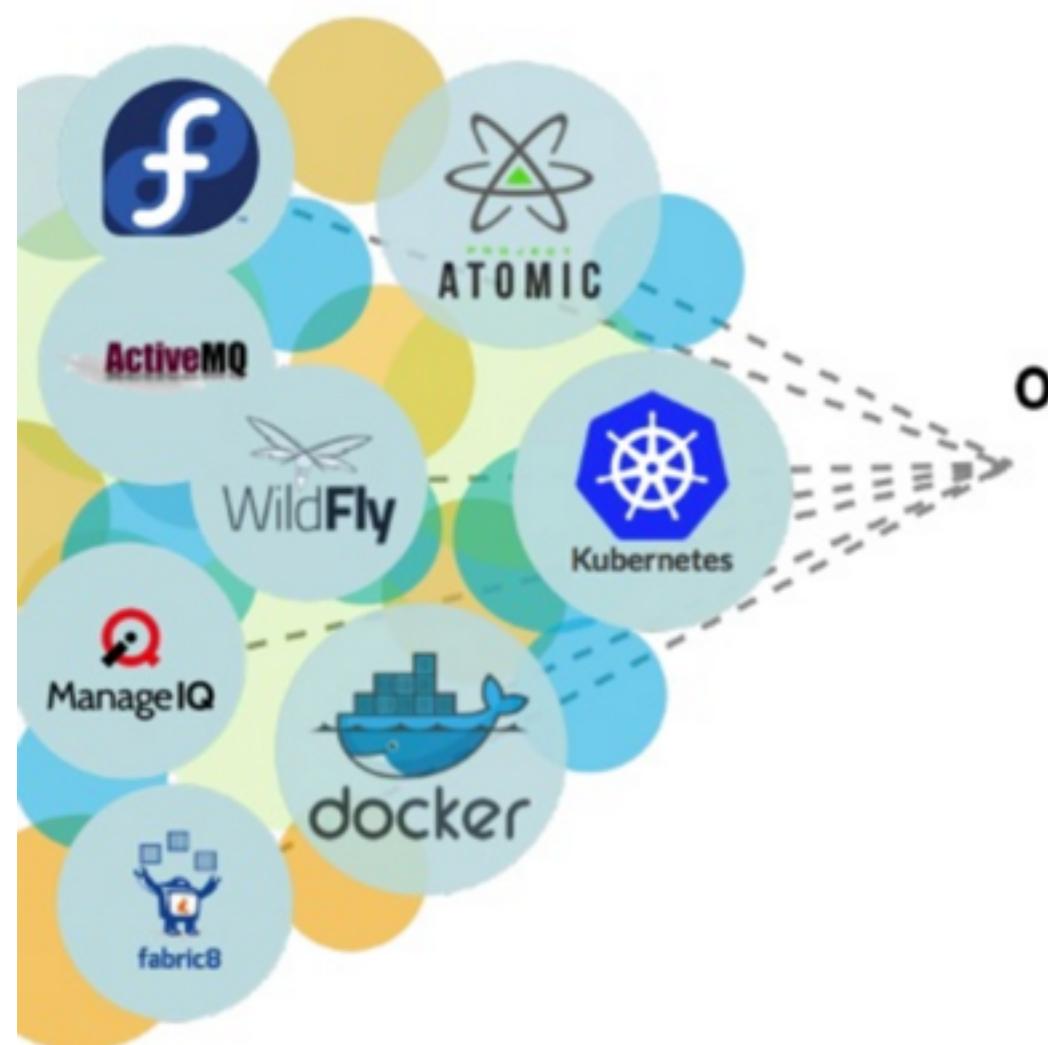
Pet Set

- Alpha resource introduced in 1.3
- Stateful pods
- PetSet has 0..N-1 Pets
- Pet has deterministic name, and a unique identity
 - stable hostname
 - ordinal index
 - stable storage linked to ordinal & hostname
- Each Pet has at most one pod
- Pet Set has at most one Pet with a given identity

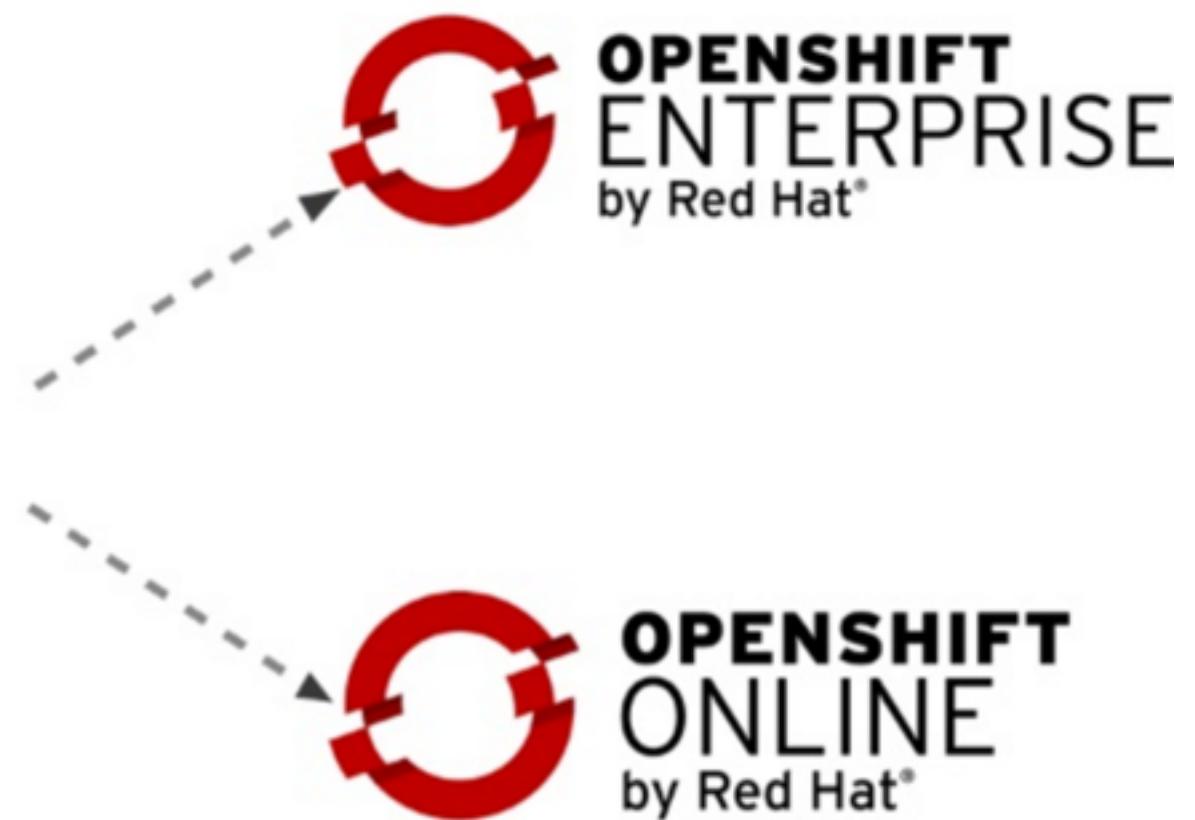


<https://github.com/arun-gupta/couchbase-kubernetes/tree/master/cluster-petset>

TM
Devoxx



OPENSHIFT
origin



OpenShift 3



DEVOPS TOOLS & USER EXPERIENCE

LANGUAGE RUNTIMES, MIDDLEWARE,
DATABASES AND OTHER SERVICES

CONTAINER ORCHESTRATION & MANAGEMENT

CONTAINER API

CONTAINER HOST

OpenShift Hub

Deploy Your Favorite Languages, Frameworks, and Databases in One Click.

Search the Hub



Primed Partners [Browse all](#)



Cloudmunch



Pachyderm



dynatrace



Nuage Networks
from Nokia



Iron.io



Juniper
Networks



NGINX



Couchbase



Click2Cloud



GitLab



Diamanti



NetApp



6fusion



3scale API
Management



Sysdig



CrunchyData



CloudBees



Roambee/T-
Systems

OpenShift Hub

Deploy Your Favorite Languages, Frameworks, and Databases in One Click.

Search the Hub



Primed Partners [Browse all](#)



Cloudmunch



Pachyderm



dynatrace



Nuage Networks
from Nokia



Iron.io



JUNIPER
NETWORKS



NetApp



NGINX



Couchbase



Click2Cloud



GitLab



Diamanti



NetApp



6fusion



3scale API
Management



Sysdig



CrunchyData



CloudBees



Roambee/T-
Systems

TM

X
X
O
>
H
D

Projects Project couchbase Add to project

Overview

Applications >

Builds >

Resources >

Storage

Monitoring

COUCHBASE MASTER SERVICE

couchbase-master-service

Replication Controller couchbase-master-rc - 8 minutes ago

CONTAINER: COUCHBASE-MASTER

Image: arungupta/couchbase:k8s

Ports: 8091/TCP

1 pod

COUCHBASE WORKER SERVICE

couchbase-worker-service

Replication Controller couchbase-worker-rc - a minute ago

CONTAINER: COUCHBASE-WORKER

Image: arungupta/couchbase:k8s

Ports: 8091/TCP

3 pods

blog.openshift.com/openshift-ecosystem-couchbase-openshift-nosql-applications/

#Devoxx #containers

@arungupta

TM
X
O
>
S
D

Mesos





Mesos



- Open source cluster manager from UC Berkeley

D E V O O X



Mesos



- Open source cluster manager from UC Berkeley
- Provides resource isolation and sharing across distributed applications

D
E
V
O
W
O
R
K
S



Mesos



- Open source cluster manager from UC Berkeley
- Provides resource isolation and sharing across distributed applications
- Run distributed systems on the same pool of nodes





Mesos



- Open source cluster manager from UC Berkeley
- Provides resource isolation and sharing across distributed applications
- Run distributed systems on the same pool of nodes
 - Hadoop, Spark, Jenkins, Couchbase, ...



Mesos



- Open source cluster manager from UC Berkeley
- Provides resource isolation and sharing across distributed applications
- Run distributed systems on the same pool of nodes
 - Hadoop, Spark, Jenkins, Couchbase, ...
- Cluster monitoring



Mesos



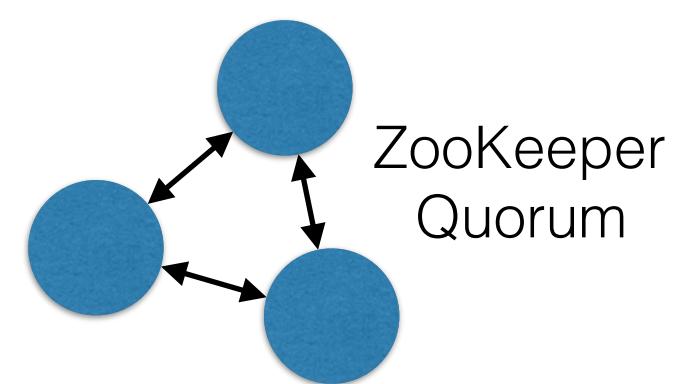
- Open source cluster manager from UC Berkeley
- Provides resource isolation and sharing across distributed applications
- Run distributed systems on the same pool of nodes
 - Hadoop, Spark, Jenkins, Couchbase, ...
- Cluster monitoring
- Tasks isolated via Linux containers



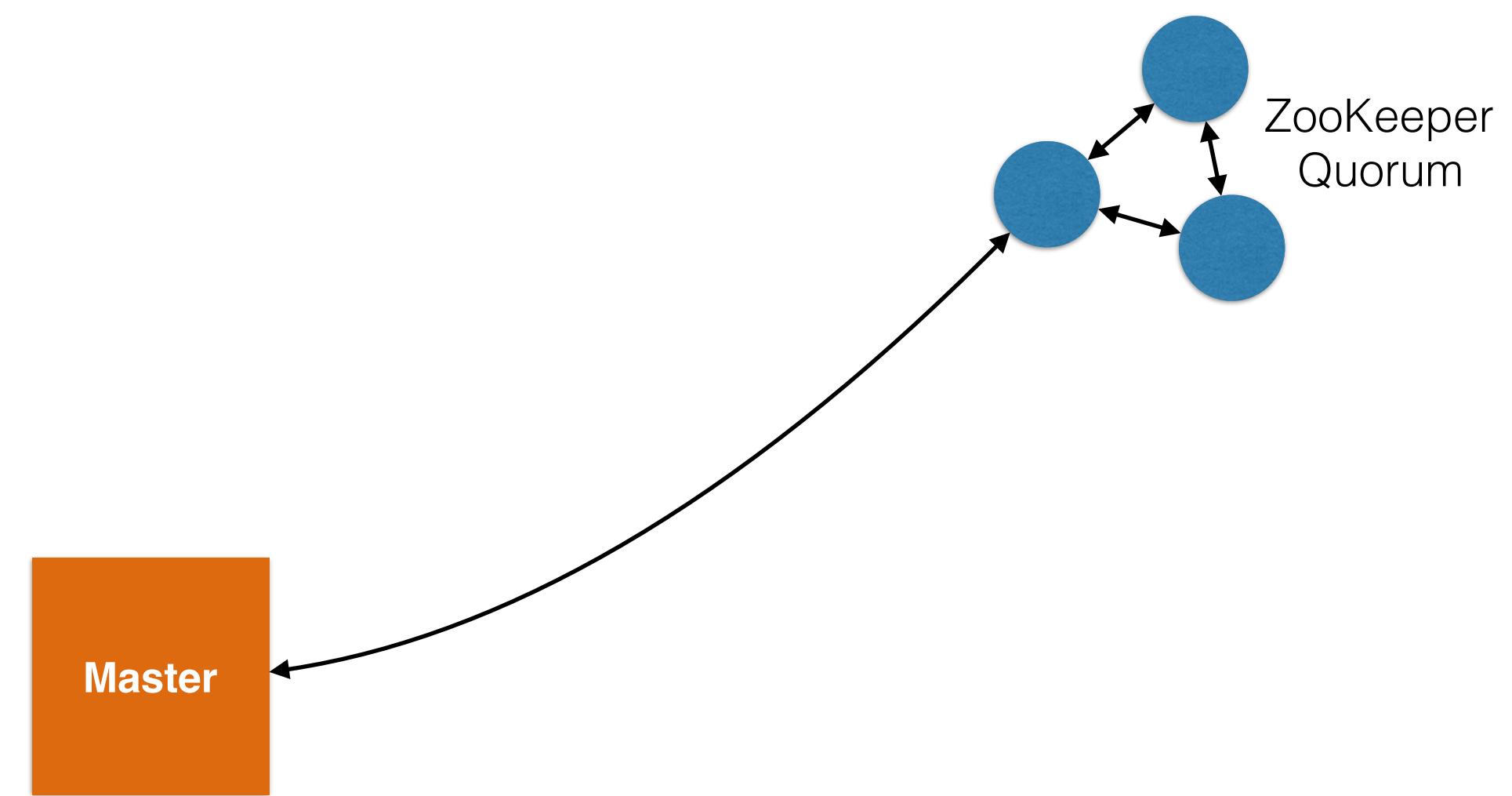
<http://mesos.apache.org/documentation/latest/architecture/>

#Devoxx #containers

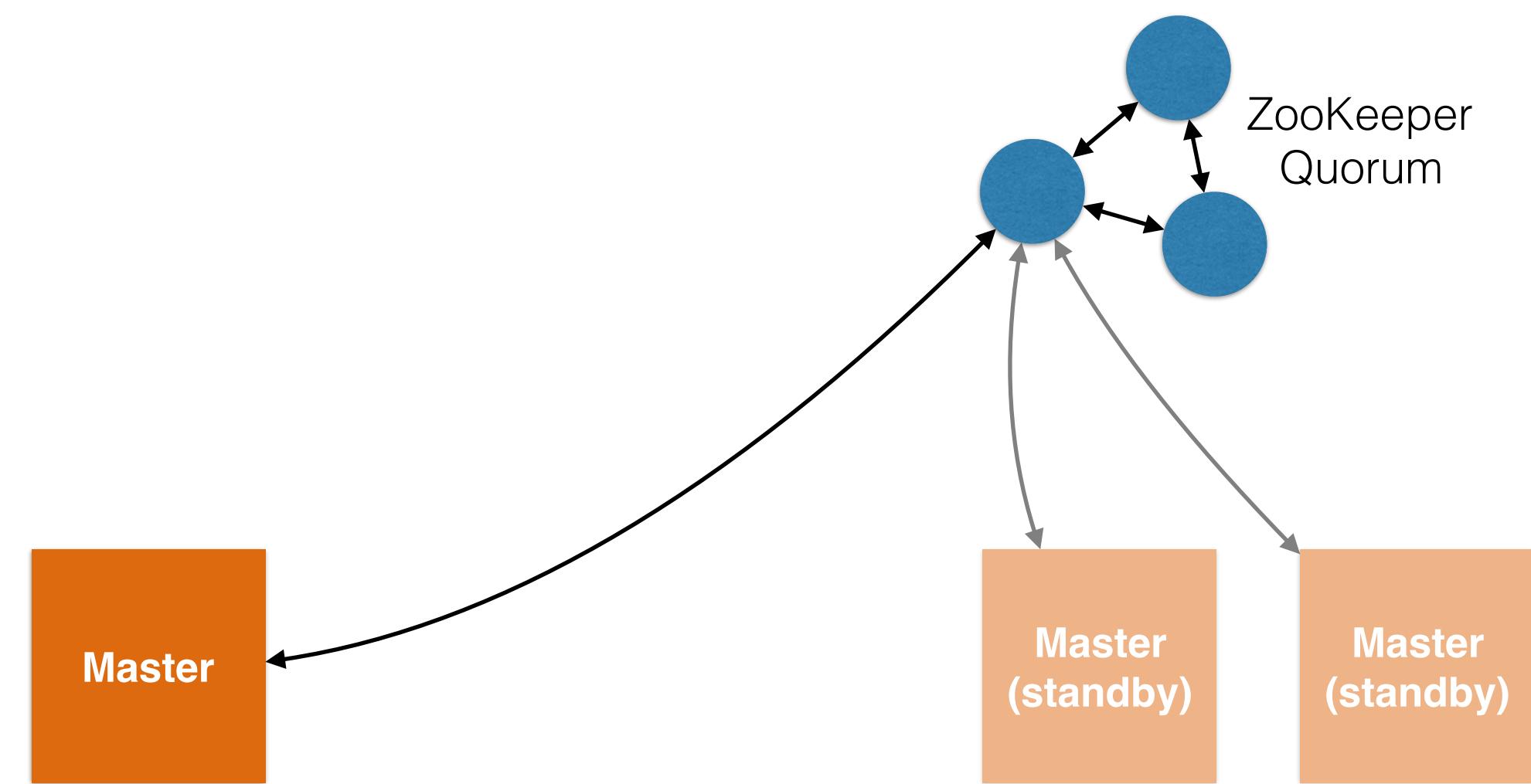
@arungupta



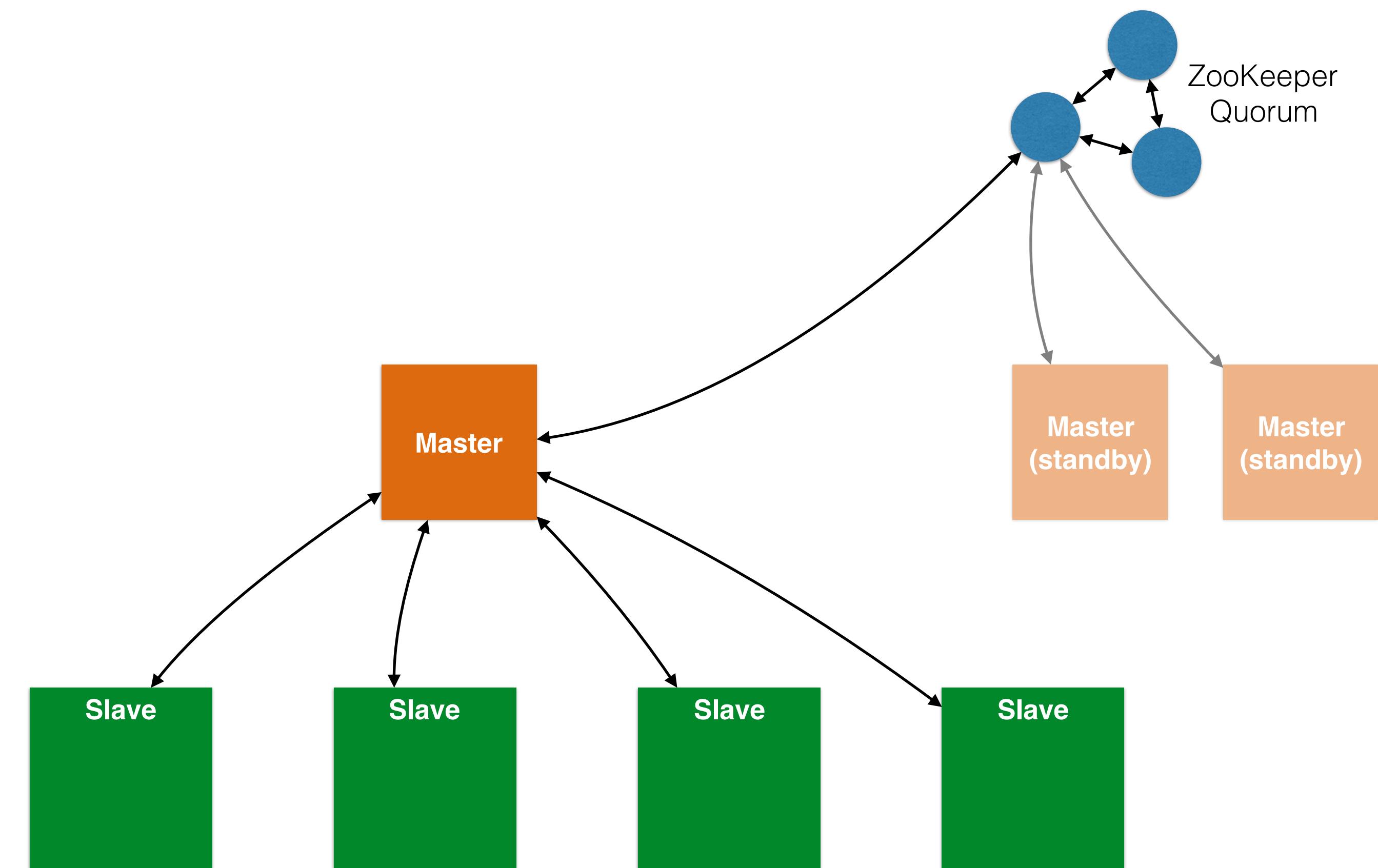
<http://mesos.apache.org/documentation/latest/architecture/>



<http://mesos.apache.org/documentation/latest/architecture/>

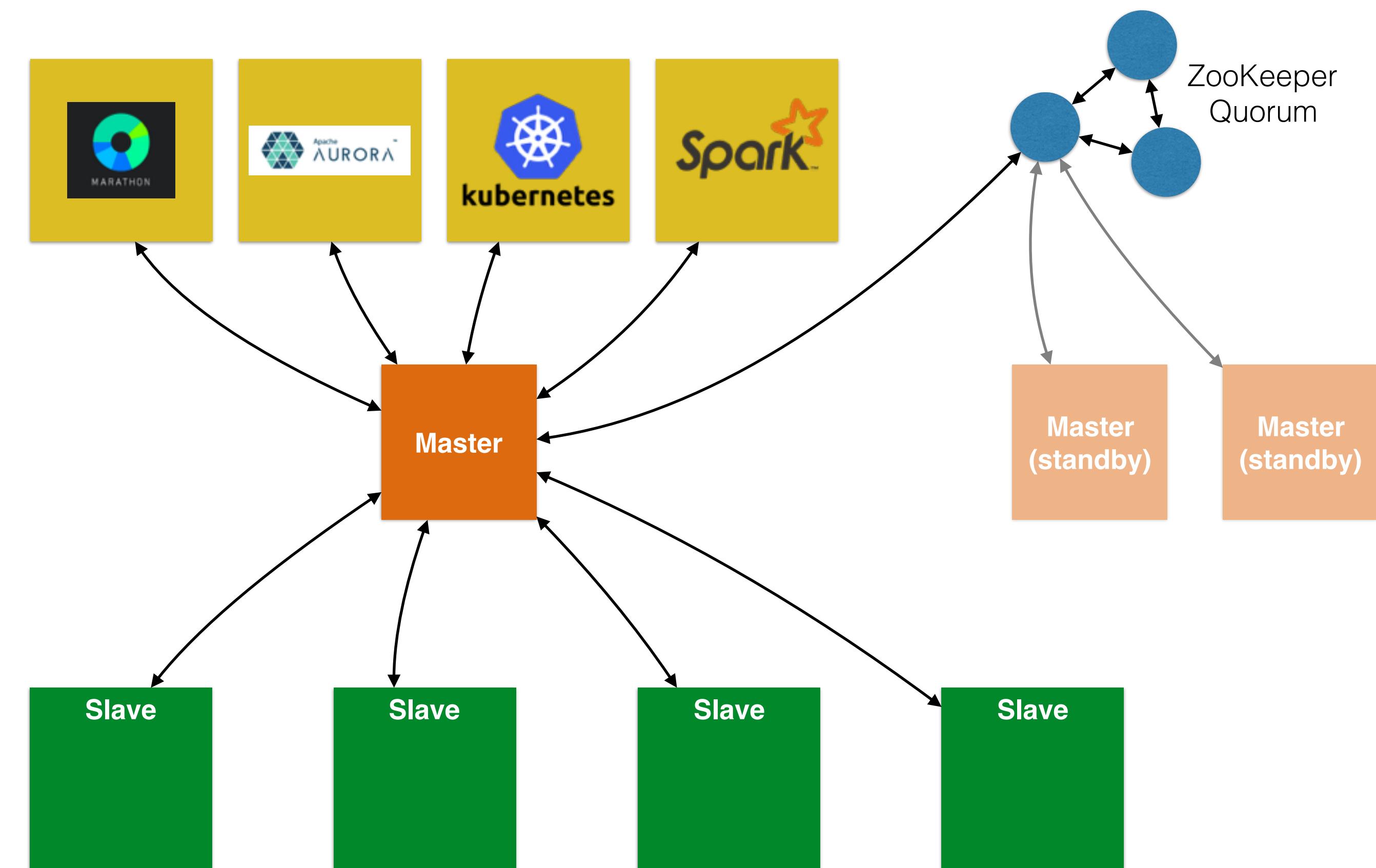


<http://mesos.apache.org/documentation/latest/architecture/>

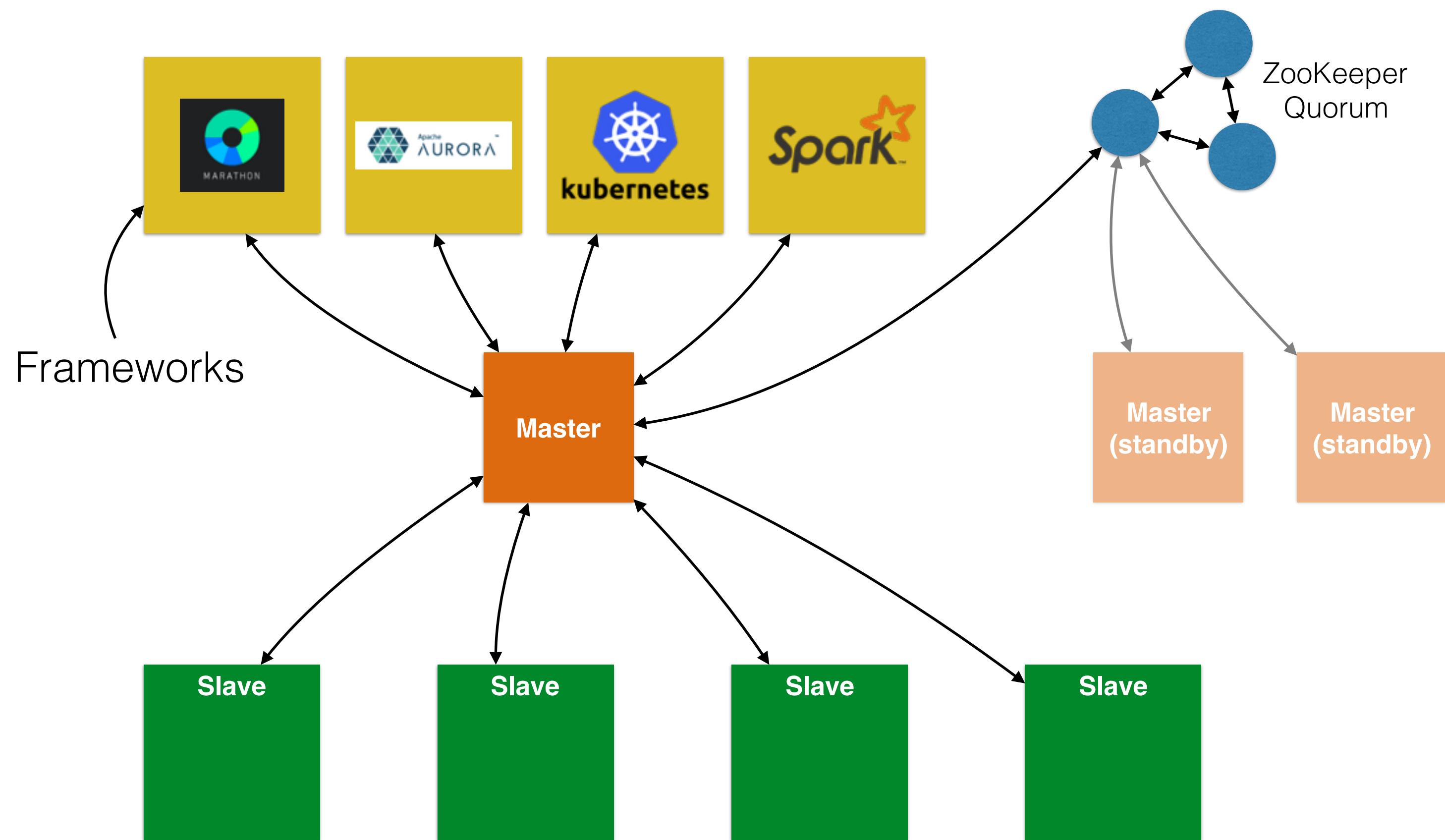


<http://mesos.apache.org/documentation/latest/architecture/>

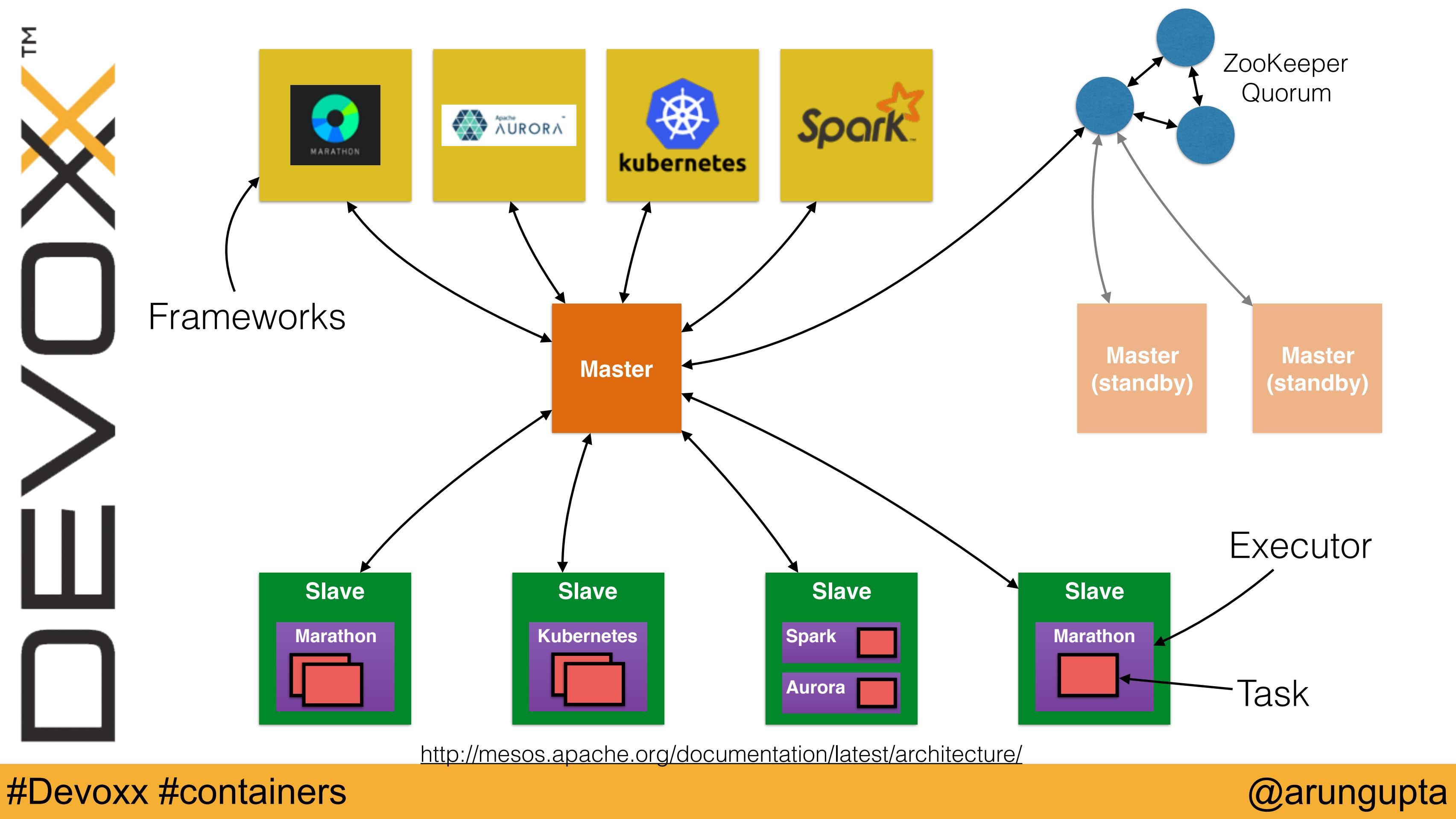
DEVOXX



<http://mesos.apache.org/documentation/latest/architecture/>



<http://mesos.apache.org/documentation/latest/architecture/>





DC/OS



- Logical compliment for Mesos
 - ZooKeeper, Docker repo, Master, Slave, ...
- Includes UI
- Pre-bundled services like Marathon and Chronos
- Advanced security
- Enterprise support



```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm",
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    },
32    "labels": {
33        "HAPROXY_GROUP": "external",
34        "HAPROXY_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35    },
36    "healthChecks": [
37        {
38            "protocol": "HTTP",
39            "path": "/pools",
40            "ignoreHttp1xx": false
41        }
42    ],
43    "env": {
44        "TYPE": "MASTER"
45    },
46    "ipAddress": {
47        "networkName": "dcos"
48    }
49 }
```

TM

X

O

>

U

D

#Devoxx #dc

```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm", // Line 15
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    },
32    "labels": {
33        "HAProxy_GROUP": "external",
34        "HAProxy_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35    },
36    "healthChecks": [
37        {
38            "protocol": "HTTP",
39            "path": "/pools",
40            "ignoreHttp1xx": false
41        }
42    ],
43    "env": {
44        "TYPE": "MASTER"
45    },
46    "ipAddress": {
47        "networkName": "dcos"
48    }
49 }
```

@arungupta



```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm",
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    },
32    "labels": {
33        "HAPROXY_GROUP": "external",
34        "HAPROXY_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35    },
36    "healthChecks": [
37        {
38            "protocol": "HTTP",
39            "path": "/pools",
40            "ignoreHttp1xx": false
41        }
42    ],
43    "env": {
44        "TYPE": "MASTER"
45    },
46    "ipAddress": {
47        "networkName": "dcos"
48    }
49 }
```

TM



```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm",
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    }
```

```
32     "labels": {
33         "HAProxy_GROUP": "external",
34         "HAProxy_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35     },
36     "healthChecks": [
37         {
38             "protocol": "HTTP",
39             "path": "/pools",
40             "ignoreHttp1xx": false
41         }
42     ],
43     "env": {
44         "TYPE": "MASTER"
45     },
46     "ipAddress": {
47         "networkName": "dcos"
48     }
49 }
```



```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm",
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    }
32    "labels": {
33        "HAProxy_GROUP": "external",
34        "HAProxy_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35    },
36    "healthChecks": [
37        {
38            "protocol": "HTTP",
39            "path": "/pools",
40            "ignoreHttp1xx": false
41        }
42    ],
43    "env": {
44        "TYPE": "MASTER"
45    },
46    "ipAddress": {
47        "networkName": "dcos"
48    }
49 }
```



```
1 {
2     "id": "/couchbase-startup",
3     "cmd": null,
4     "cpus": 4,
5     "mem": 4096,
6     "disk": 4096,
7     "instances": 1,
8     "executor": null,
9     "fetch": null,
10    "constraints": null,
11    "acceptedResourceRoles": null,
12    "user": null,
13    "container": {
14        "docker": {
15            "image": "arungupta/couchbase:swarm",
16            "forcePullImage": false,
17            "privileged": false,
18            "portMappings": [
19                {
20                    "containerPort": 8091,
21                    "protocol": "tcp",
22                    "name": "admin",
23                    "servicePort": 8091,
24                    "labels": {
25                        "VIP_0": "/couchbase-startup:8091"
26                    }
27                }
28            ],
29            "network": "USER"
30        }
31    }
```

```
32     "labels": {
33         "HAPROXY_GROUP": "external",
34         "HAPROXY_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35     },
36     "healthChecks": [
37         {
38             "protocol": "HTTP",
39             "path": "/pools",
40             "ignoreHttp1xx": false
41         }
42     ],
43     "env": {
44         "TYPE": "MASTER"
45     },
46     "ipAddress": {
47         "networkName": "dcos"
48     }
49 }
```

TM



```
1  {
2    "volumes": null,
3    "id": "/couchbase-worker",
4    "cmd": null,
5    "args": null,
6    "user": null,
7    "env": {
8      "TYPE": "WORKER",
9      "COUCHBASE_MASTER": "couchbase-startup.marathon"
10     },
11    "instances": 1,
12    "cpus": 4,
13    "mem": 4096,
14    "disk": 4096,
15    "gpus": 0,
16    "executor": null,
17    "constraints": null,
18    "fetch": null,
19    "storeUrls": null,
20    "backoffSeconds": 1,
21    "backoffFactor": 1.15,
22    "maxLaunchDelaySeconds": 3600,
23
24  "container": {
25    "docker": {
26      "image": "arungupta/couchbase:swarm",
27      "forcePullImage": false,
28      "privileged": false,
29      "portMappings": [
30        {
31          "containerPort": 8091,
32          "protocol": "tcp",
33          "name": "admin",
34          "servicePort": 8091,
35          "labels": {
36            "VIP_0": "/couchbase-worker:8091"
37          }
38        }
39      ],
40      "network": "USER"
41    },
42    "healthChecks": [
43      {
44        "protocol": "HTTP",
45        "path": "/pools",
46        "ignoreHttp1xx": false
47      }
48    ],
49    "readinessChecks": null,
50    "dependencies": null,
51    "upgradeStrategy": {
52      "minimumHealthCapacity": 1,
53      "maximumOverCapacity": 1
54    },
55    "labels": null,
56    "acceptedResourceRoles": null,
57    "ipAddress": {
58      "networkName": "dcos"
59    },
60    "residency": null,
61    "secrets": null,
62    "taskKillGracePeriodSeconds": null
63  }
```



```
1  {
2      "volumes": null,
3      "id": "/couchbase-worker",
4      "cmd": null,
5      "args": null,
6      "user": null,
7      "env": {
8          "TYPE": "WORKER",
9          "COUCHBASE_MASTER": "couchbase-startup.marathon"
10     },
11     "instances": 1,
12     "cpus": 4,
13     "mem": 4096,
14     "disk": 4096,
15     "gpus": 0,
16     "executor": null,
17     "constraints": null,
18     "fetch": null,
19     "storeUrls": null,
20     "backoffSeconds": 1,
21     "backoffFactor": 1.15,
22     "maxLaunchDelaySeconds": 3600,
23
24         "container": {
25             "docker": {
26                 "image": "arungupta/couchbase:swarm",
27                 "forcePullImage": false,
28                 "privileged": false,
29                 "portMappings": [
30                     {
31                         "containerPort": 8091,
32                         "protocol": "tcp",
33                         "name": "admin",
34                         "servicePort": 8091,
35                         "labels": {
36                             "VIP_0": "/couchbase-worker:8091"
37                         }
38                     }
39                 ],
40                 "network": "USER"
41             },
42             "healthChecks": [
43                 {
44                     "protocol": "HTTP",
45                     "path": "/pools",
46                     "ignoreHttp1xx": false
47                 }
48             ],
49             "readinessChecks": null,
50             "dependencies": null,
51             "upgradeStrategy": {
52                 "minimumHealthCapacity": 1,
53                 "maximumOverCapacity": 1
54             },
55             "labels": null,
56             "acceptedResourceRoles": null,
57             "ipAddress": {
58                 "networkName": "dcos"
59             },
60             "residency": null,
61             "secrets": null,
62             "taskKillGracePeriodSeconds": null
63         }
64     }
65 }
```



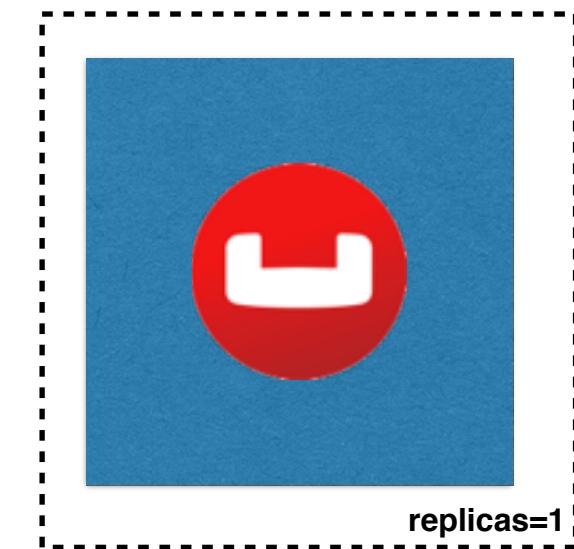
```
1  {
2    "volumes": null,
3    "id": "/couchbase-worker",
4    "cmd": null,
5    "args": null,
6    "user": null,
7    "env": [
8      {"TYPE": "WORKER",
9       "COUCHBASE_MASTER": "couchbase-startup.marathon"
10    },
11    "instances": 1,
12    "cpus": 4,
13    "mem": 4096,
14    "disk": 4096,
15    "gpus": 0,
16    "executor": null,
17    "constraints": null,
18    "fetch": null,
19    "storeUrls": null,
20    "backoffSeconds": 1,
21    "backoffFactor": 1.15,
22    "maxLaunchDelaySeconds": 3600,
23
24  "container": {
25    "docker": {
26      "image": "arungupta/couchbase:swarm",
27      "forcePullImage": false,
28      "privileged": false,
29      "portMappings": [
30        {
31          "containerPort": 8091,
32          "protocol": "tcp",
33          "name": "admin",
34          "servicePort": 8091,
35          "labels": {
36            "VIP_0": "/couchbase-worker:8091"
37          }
38        }
39      ],
40      "network": "USER"
41    },
42    "healthChecks": [
43      {
44        "protocol": "HTTP",
45        "path": "/pools",
46        "ignoreHttp1xx": false
47      }
48    ],
49    "readinessChecks": null,
50    "dependencies": null,
51    "upgradeStrategy": {
52      "minimumHealthCapacity": 1,
53      "maximumOverCapacity": 1
54    },
55    "labels": null,
56    "acceptedResourceRoles": null,
57    "ipAddress": {
58      "networkName": "dcos"
59    },
60    "residency": null,
61    "secrets": null,
62    "taskKillGracePeriodSeconds": null
63  }
```



Couchbase in DC/OS



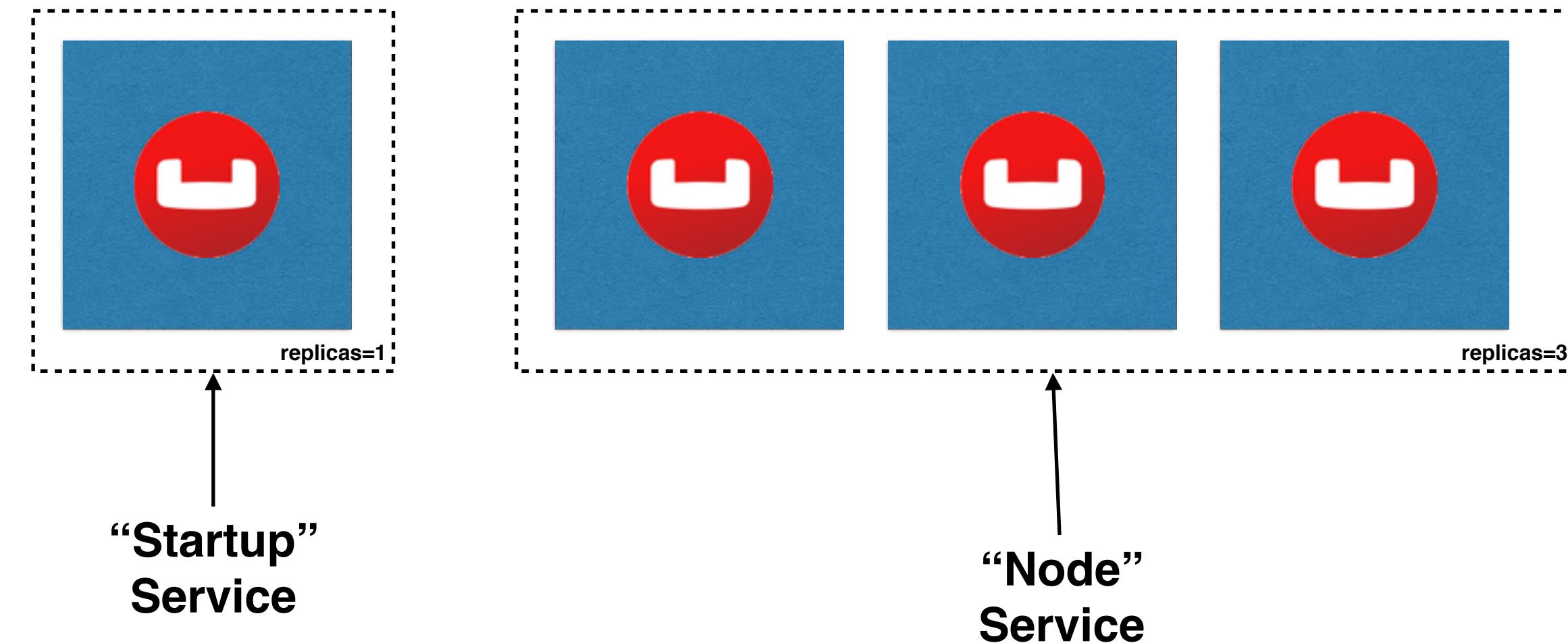
Couchbase in DC/OS



“Startup”
Service

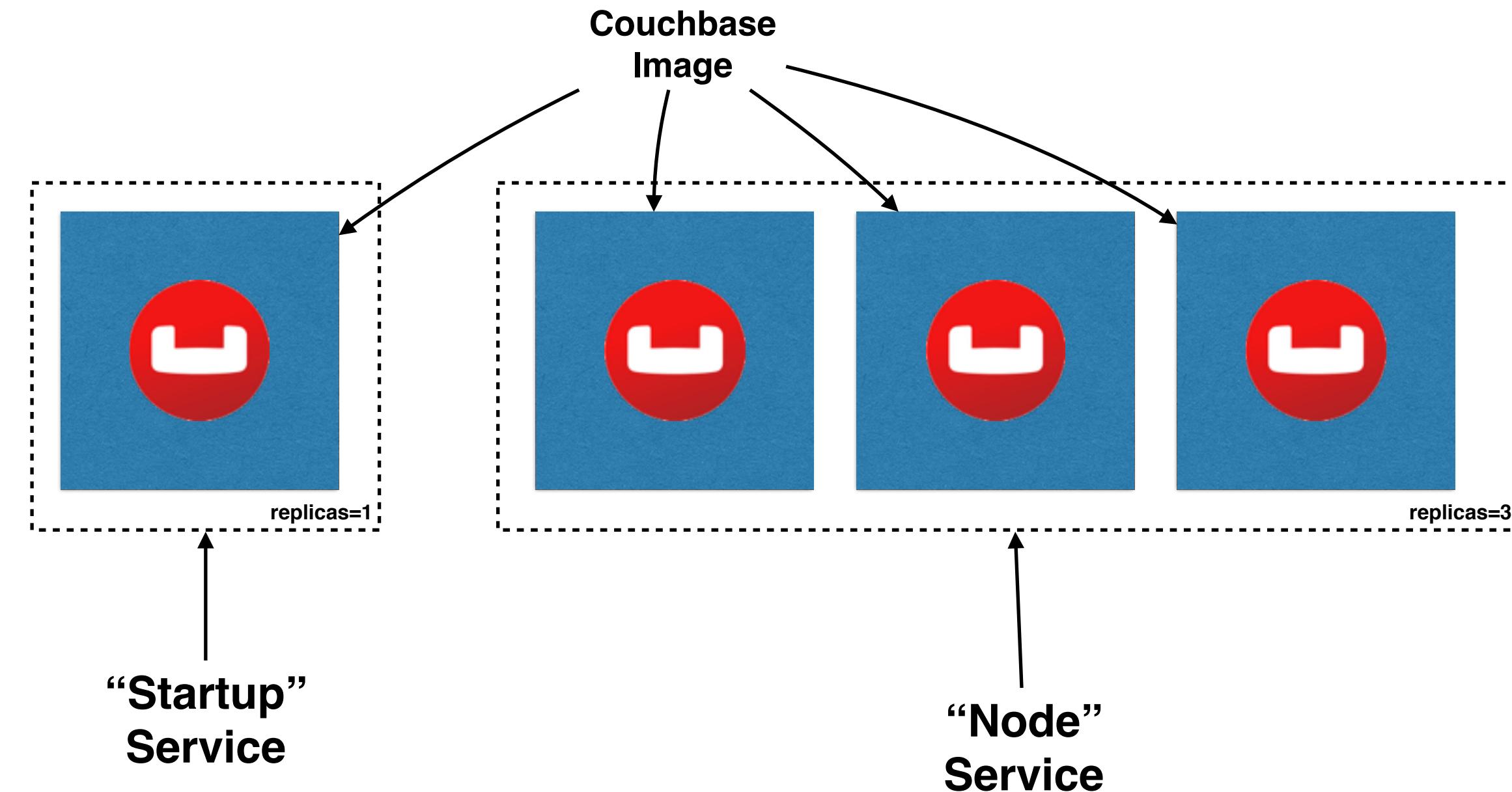
TM

Couchbase in DC/OS

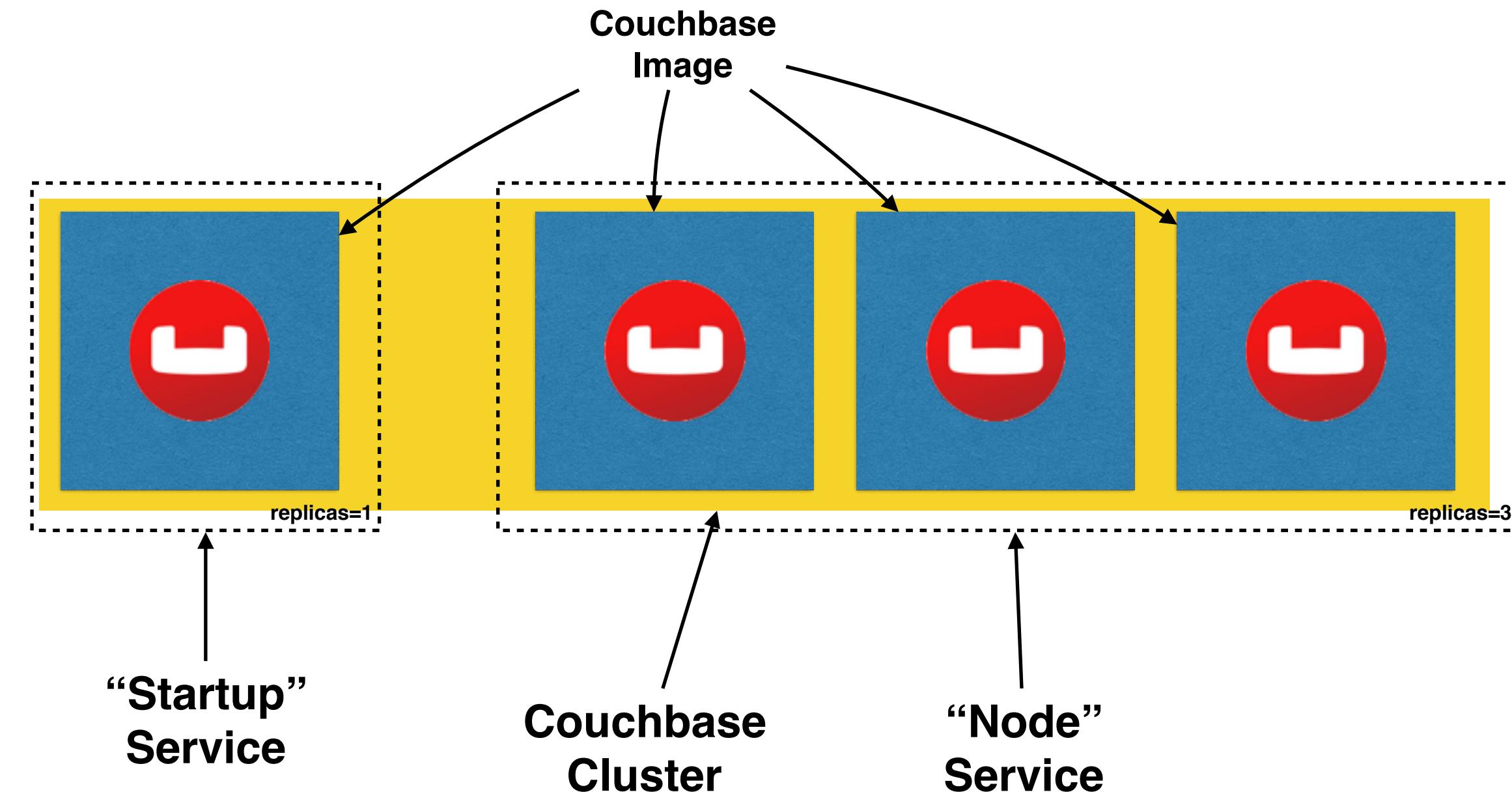


TM

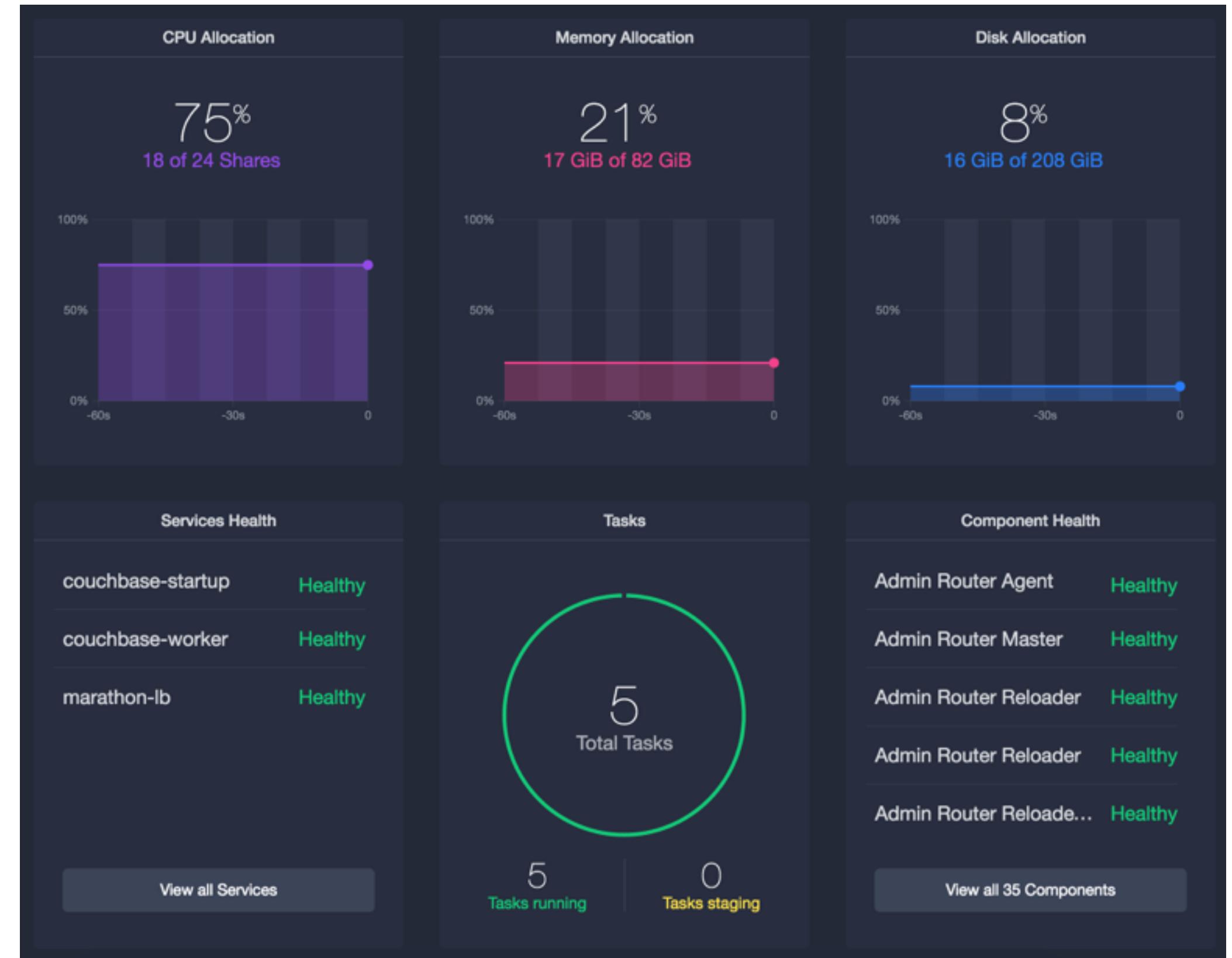
Couchbase in DC/OS



Couchbase in DC/OS



<https://github.com/arun-gupta/couchbase-dcos>



TM



Services Deployments

HEALTH

- Healthy
- Unhealthy
- Idle
- N/A

STATUS

- Running
- Deploying
- Suspended
- Delayed
- Waiting

Services

Search

Create Group Deploy Service

NAME ▲	STATUS ⓘ	CPU	MEM	DISK
couchbase-startup	Running (1/1)	4	4 GiB	4 GiB
couchbase-worker	Running (3/3)	4	4 GiB	4 GiB
marathon-lb	Running (1/1)	2	1 GiB	0 B

Filter by Label ▾

OTHER

- Universe
- Volumes

TM

X
X
O
>
S
D

Persistent Volumes



Persistent Volumes

- Local
 - Tasks are “pinned” to the node
 - New volumes are created as app scales



Persistent Volumes

- Local
 - Tasks are “pinned” to the node
 - New volumes are created as app scales
- External:Amazon EBS
 - Tasks can run on any host
 - Marathon may schedule task on another host, with associated data
 - Apps can only be scaled to a single instance



Virtual Machine Concerns



Virtual Machine Concerns

- How do I back up a container?
 - Containers are stateless and immutable
 - State is stored in a Docker volume



Virtual Machine Concerns

- How do I back up a container?
 - Containers are stateless and immutable
 - State is stored in a Docker volume
- How do I patch a container?
 - Containers are stateless and immutable
 - Terminate containers and start updated ones



Ebook: Docker for the Virtualization Admin

Understand the differences between containers and VMs

A natural response when first working with Docker containers is to try and frame them in terms of virtual machines. Oftentimes we hear people describe Docker containers as “lightweight VMs”.

This is completely understandable, and many people have done the exact same thing when they first started working with Docker. It’s easy to connect those dots as both technologies share some characteristics, but the key is that the underlying architecture is fundamentally different between containers and virtual machines.

This Ebook Covers:

- Why Containers are not VMs
- How Containers and VMs can be used together for capacity optimization
- Running containers on physical servers vs virtual machines
- Getting started with Docker and containers

Com	
First Na	Arun
Last Na	Gupta
Compar	Couch
Compar	arun.g
Phone N	408476
Country	United

TM
X
O
>
U
D



References

- Docker: docker.io
- Kubernetes: kubernetes.io
- OpenShift: openshift.io
- DC/OS: dcos.io
- Couchbase on Containers: couchbase.com/containers