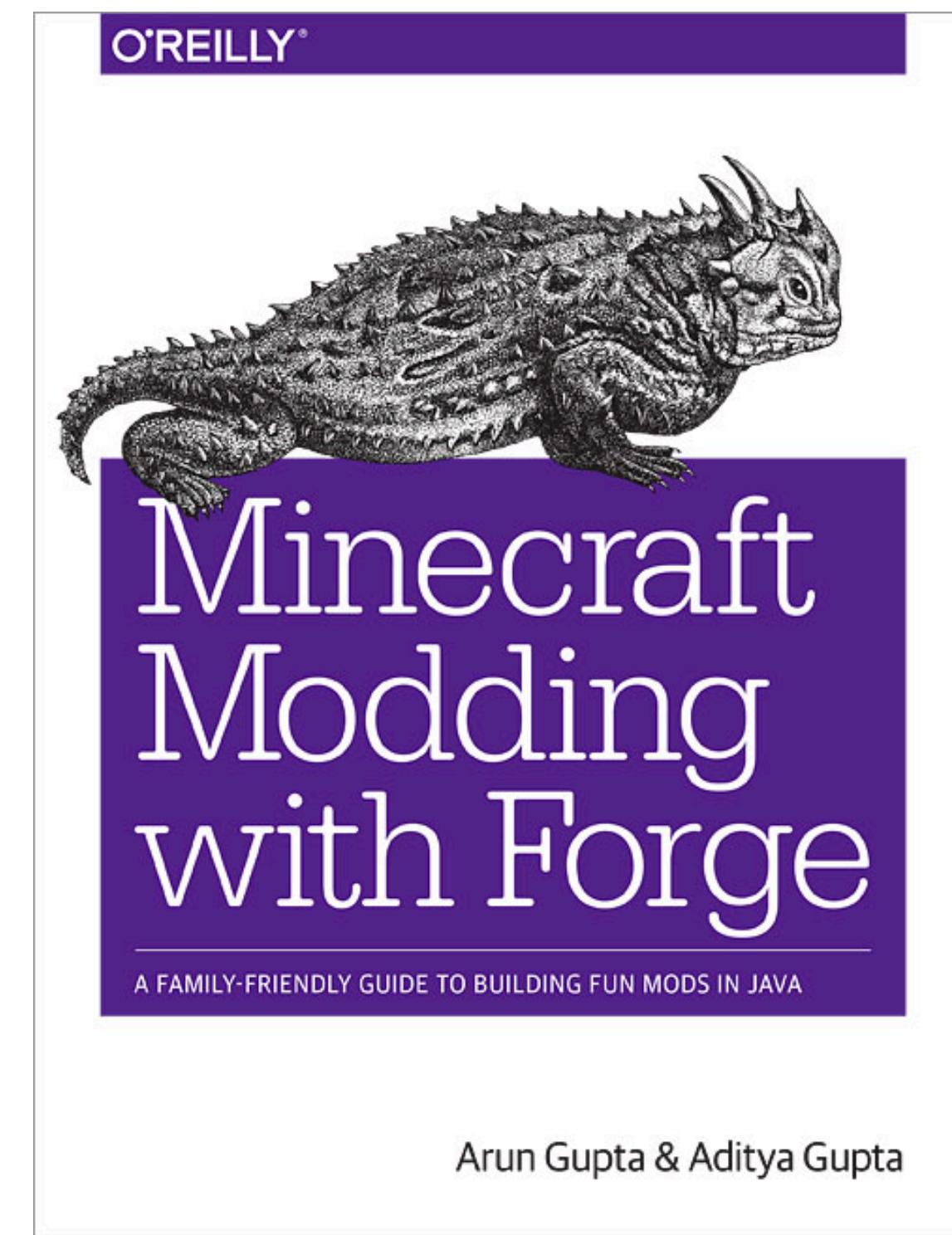


Migrate your traditional VM Clusters to Containers

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





#Devoxx #containers

@arungupta

TM
X
O
>
S
D



#Devoxx #containers

@arungupta



Virtual Machine and Containers

Virtual Machines

Containers

Designed to provide isolated environment to run application

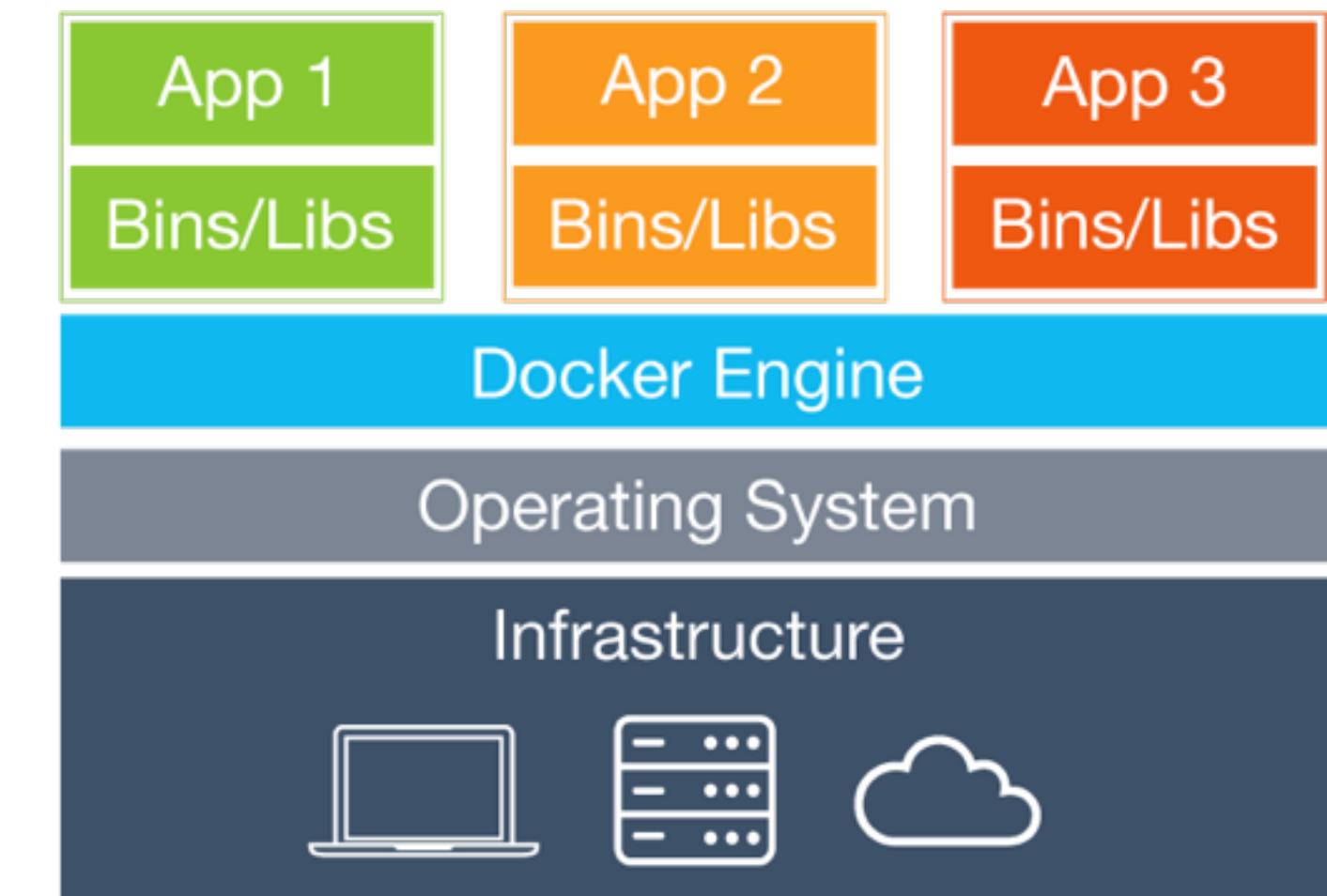
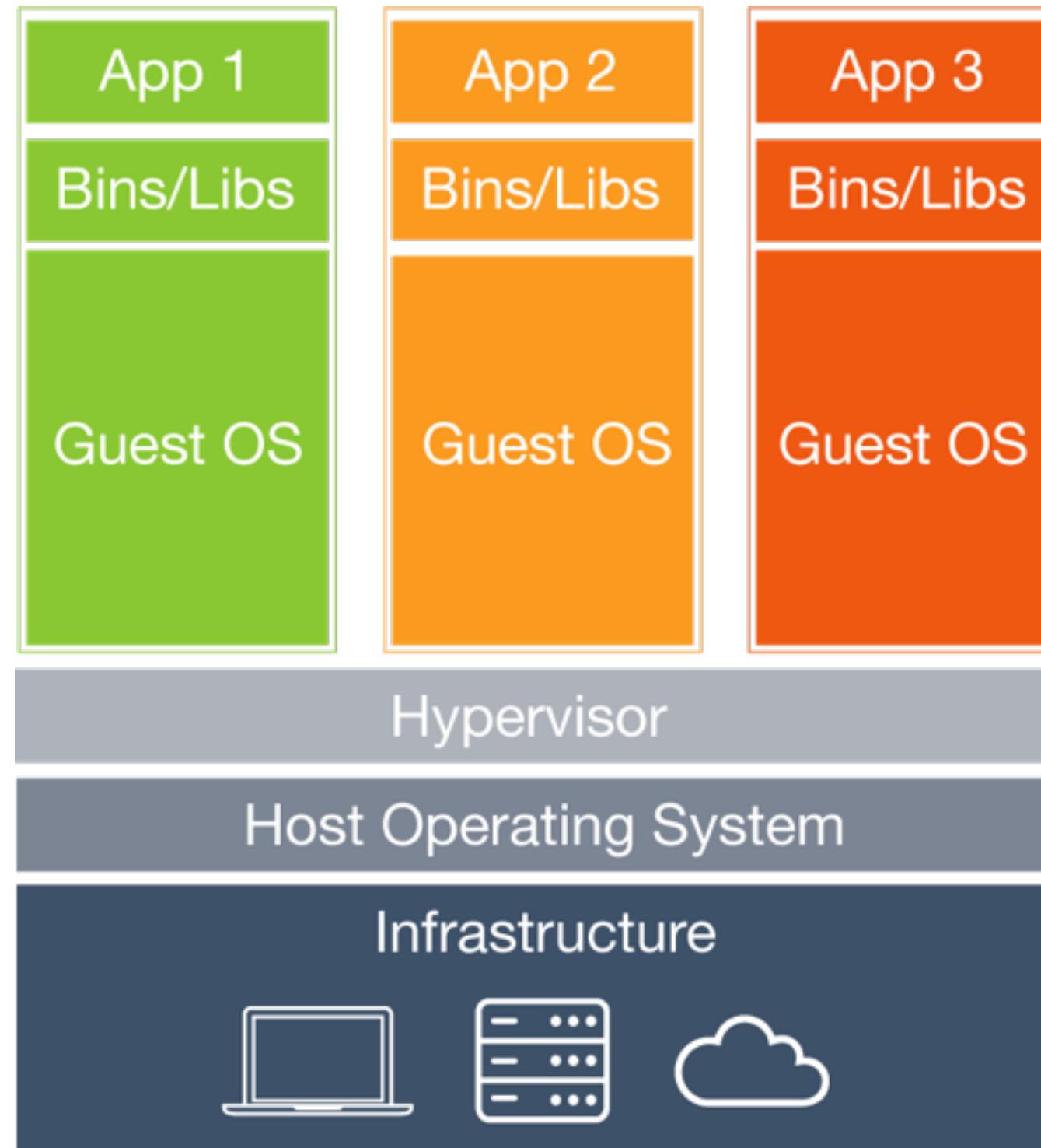
Environment is binary artifact that can be moved between hosts

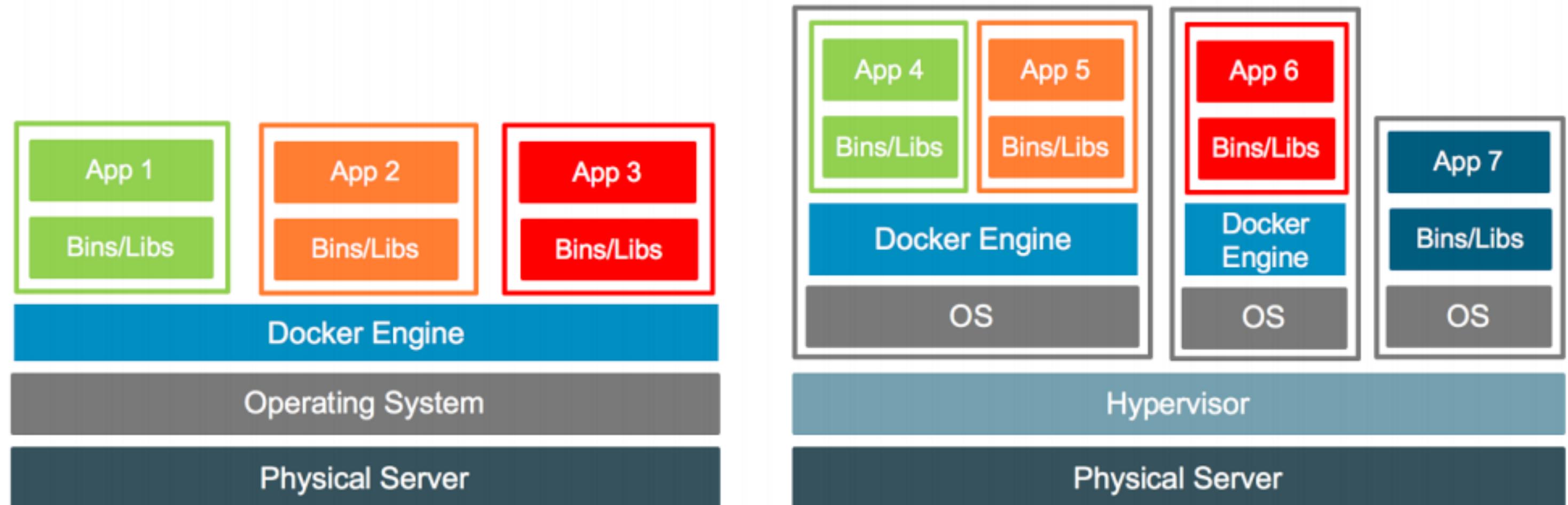
Houses

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



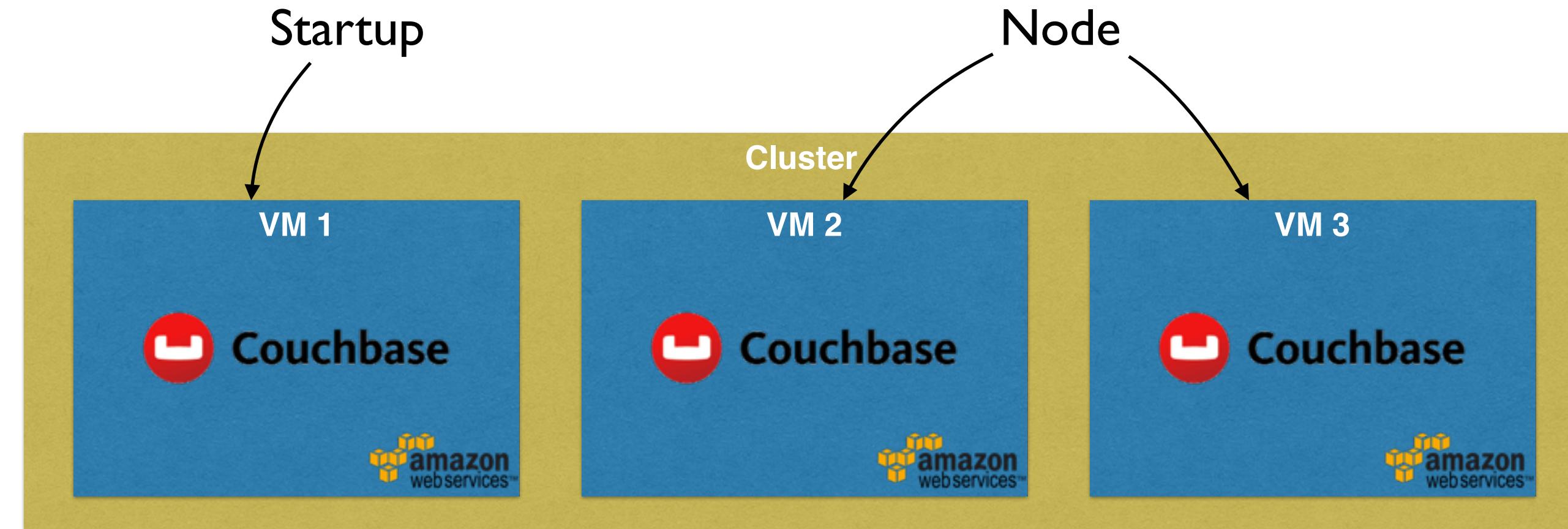


Couchbase Cluster Creation

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



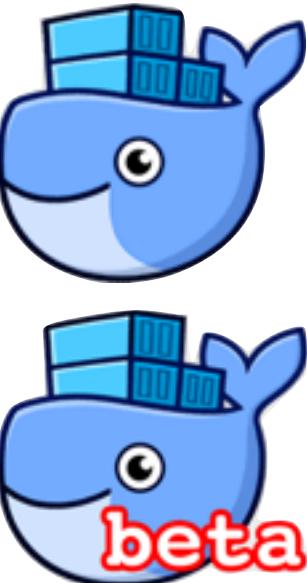
Amazon Web Services - Classical



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



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

- 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
34 else
35   couchbase-cli server-add --cluster=$COUCHBASE_MASTER:8091 --user=Administrator --password=password --server-add=$IP --server-add-username
36 fi;
37 fi;
38
39 fg 1
```

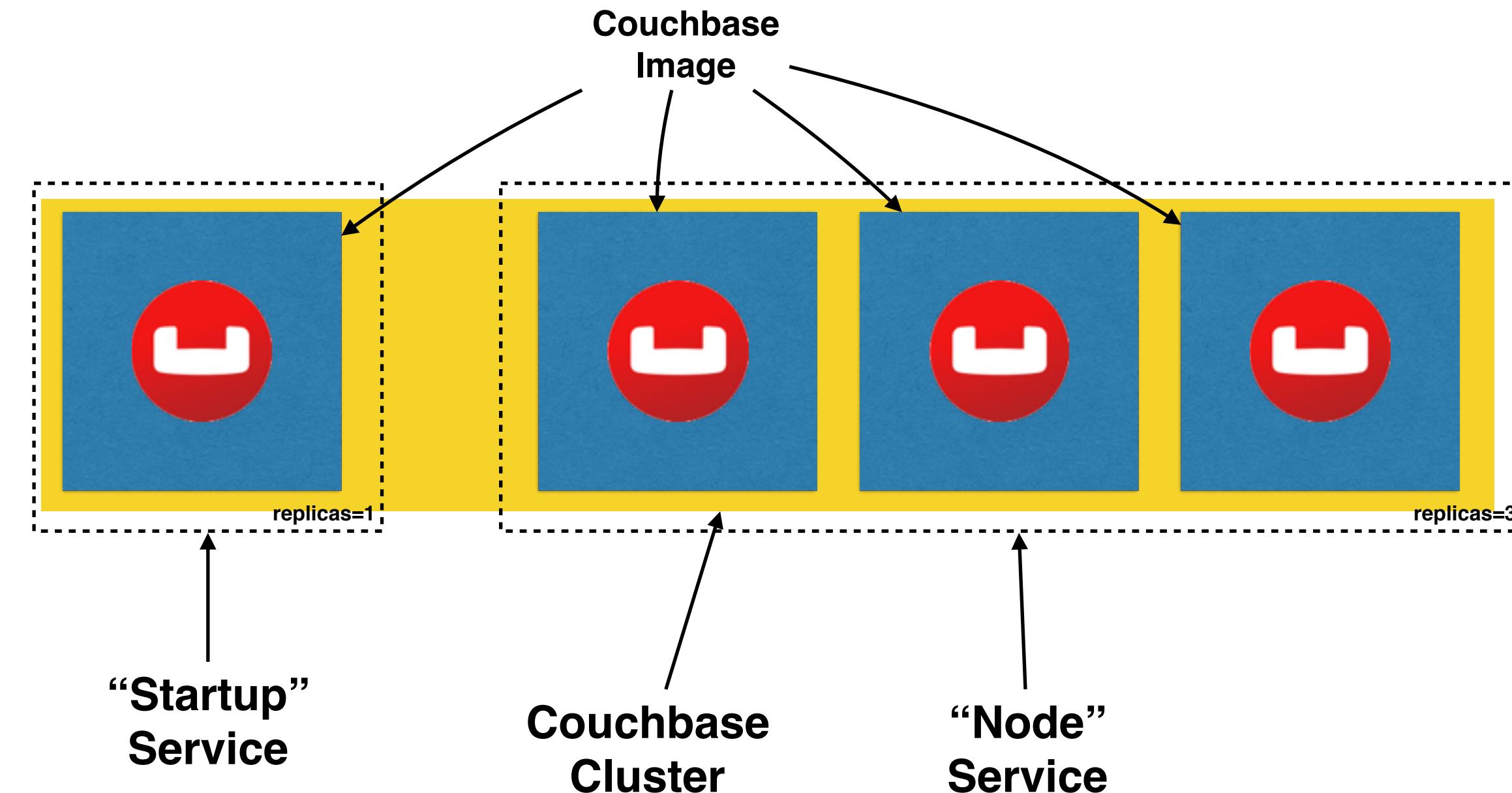
Using Docker Services

```
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



<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

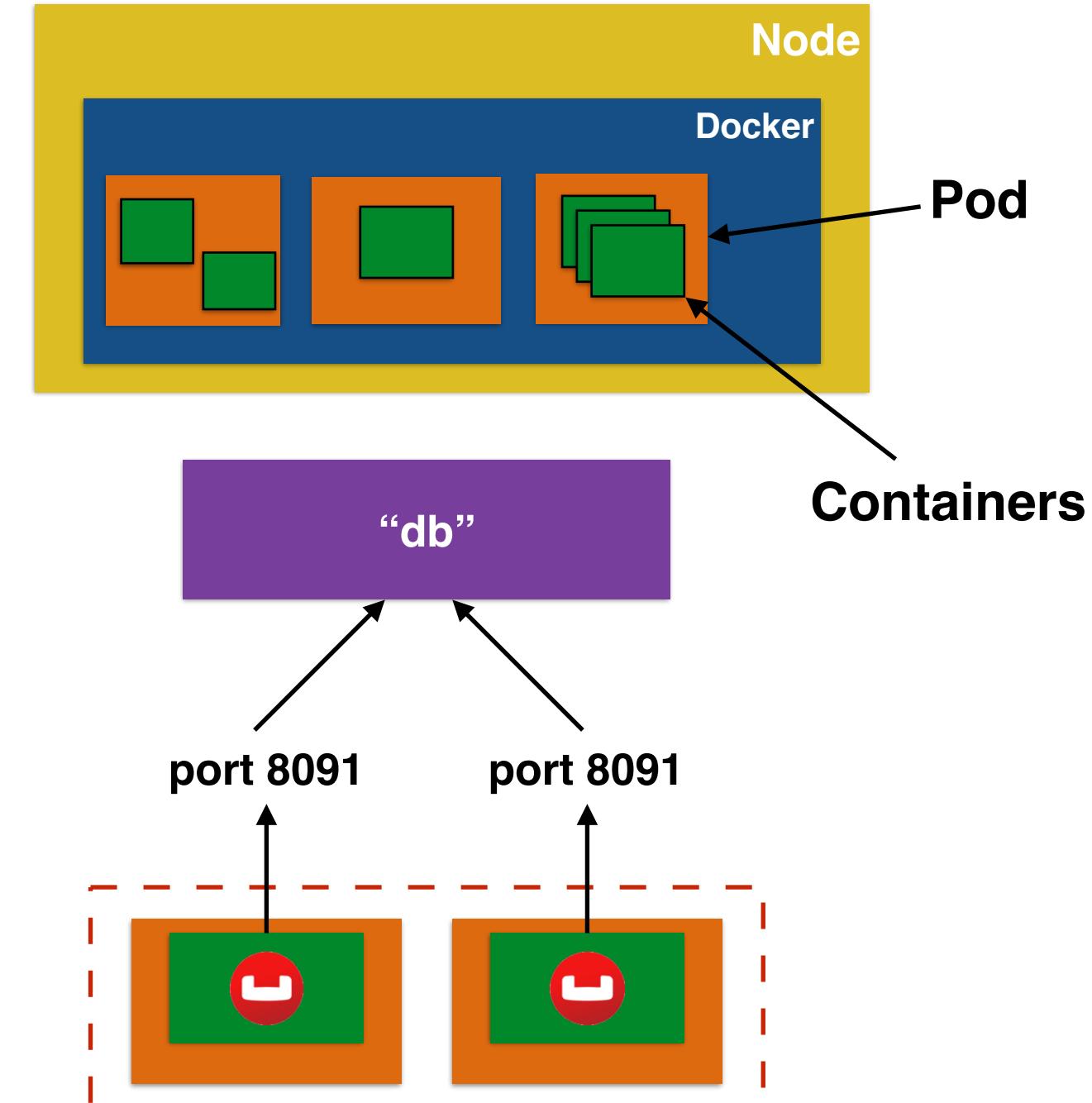
- Per-container storage (`docker run`)
- Explicit host directory mapping (`-v`)
- Shared network filesystem: Ceph, GlusterFS, NFS
- Docker Volume Plugins: Flocker, Portworx, ...
 - Volumes follow containers across hosts
 - EBS, GCE persistent disk, ...

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



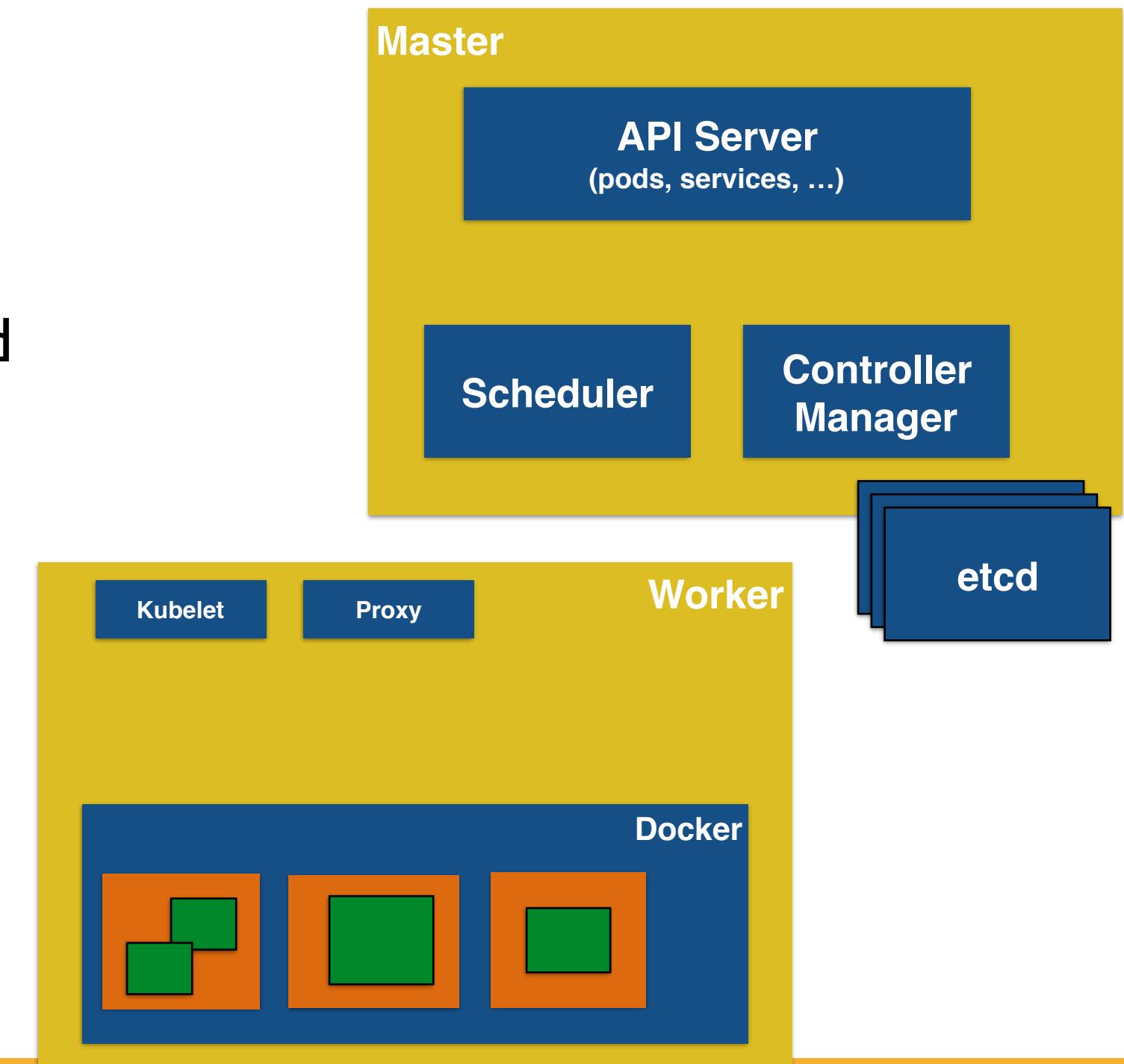
Kubernetes Concepts

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

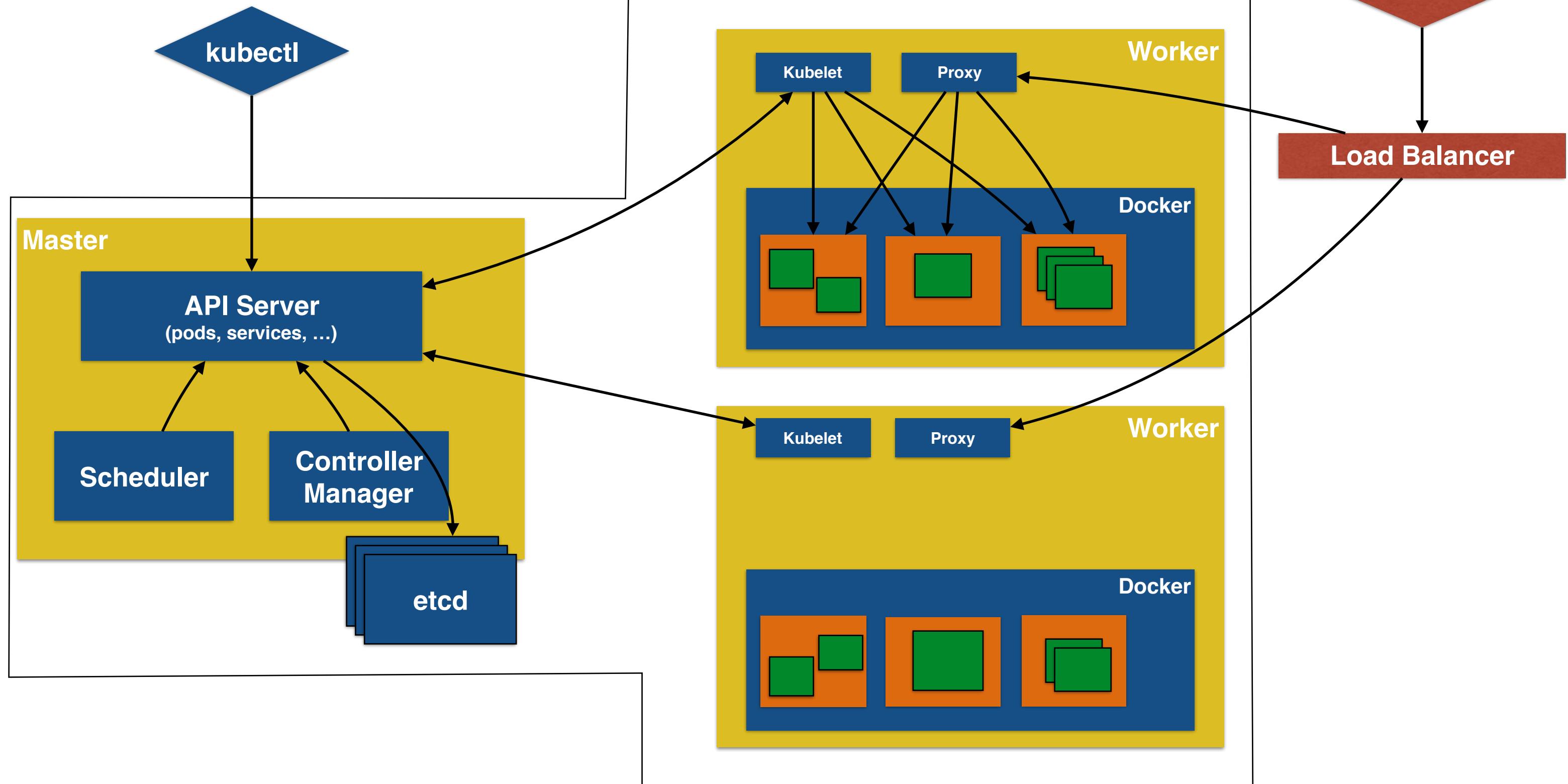


Kubernetes Components

- **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)



Kubernetes Cluster





Minikube



minikube

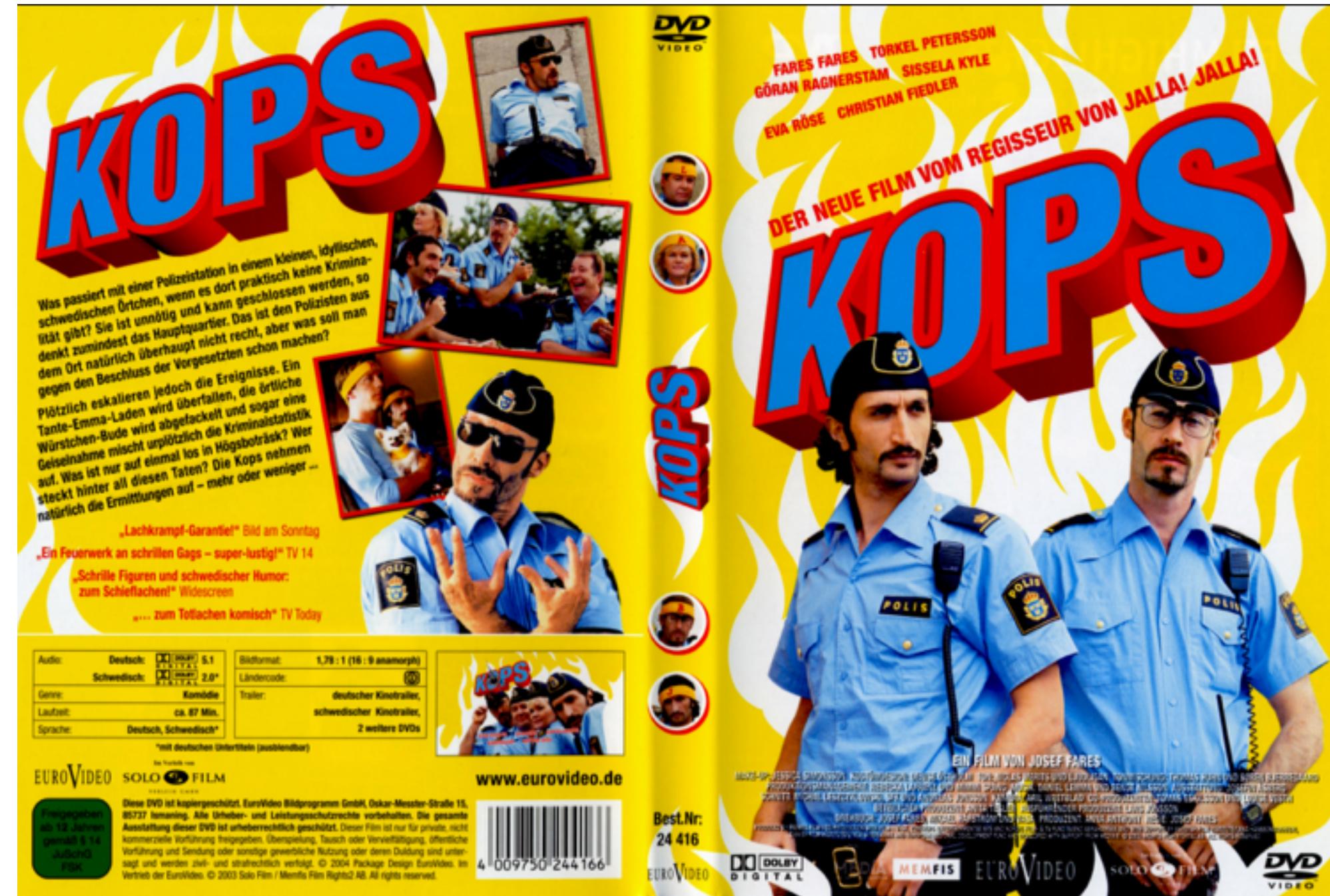
- Runs a single node cluster in a VM
- Targeted for local development
- minikube start, stop, docker-env, ...
- Requires kubectl CLI
- github.com/kubernetes/minikube/releases

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

TM
X
O
>
U

Devoxx

Multimaster Kubernetes



<http://blog.couchbase.com/2016/november/multimaster-kubernetes-cluster-amazon-kops>



```

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
15   spec:
16     containers:
17       - name: couchbase-master
18         image: arungupta/couchbase:k8s
19         env:
20           - name: TYPE
21             value: MASTER
22         ports:
23           - containerPort: 8091
24
25
26  apiVersion: v1
27  kind: Service
28  metadata:
29    name: couchbase-master-service
30    labels:
31      app: couchbase-master-service
32
33  spec:
34    ports:
35      - port: 8091
36    selector:
37      app: couchbase-master-pod
38      type: LoadBalancer

```

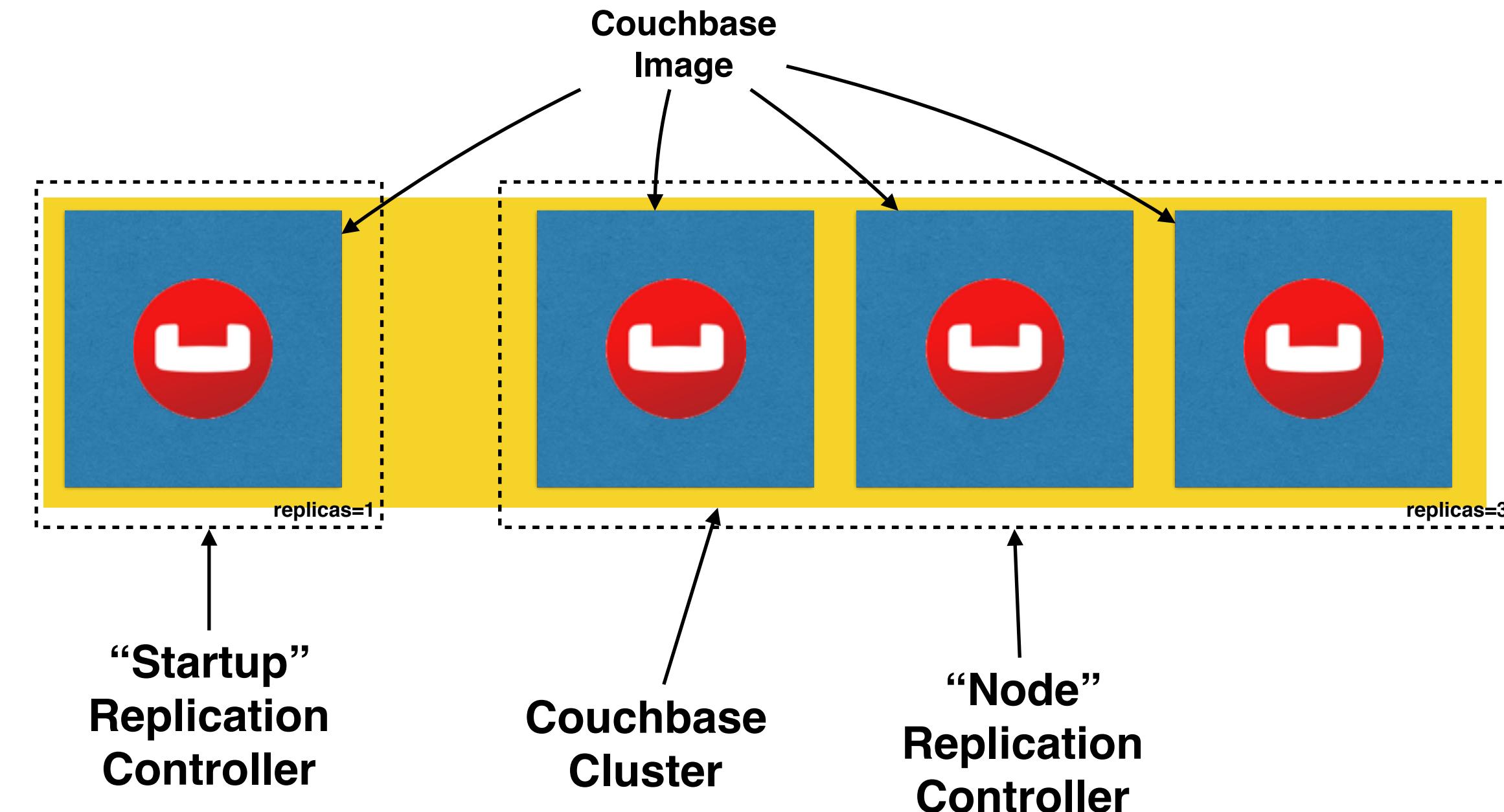
```

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
15   spec:
16     containers:
17       - name: couchbase-worker
18         image: arungupta/couchbase:k8s
19         env:
20           - name: TYPE
21             value: "WORKER"
22           - name: COUCHBASE_MASTER
23             value: "couchbase-master-service"
24           - name: AUTO_REBALANCE
25             value: "false"
26
27   ports:
28     - containerPort: 8091

```

kubectl
scale
--replicas=3
couchbase-worker

Couchbase in Kubernetes



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



CPU usage history



A line chart showing CPU usage over time. The Y-axis is labeled 'CPU (cores)' and ranges from 0 to 1.13. The X-axis is labeled 'Time' and shows two points: 16:04 and 16:06. A green line starts at 1.00 at 16:04, dips slightly to 0.95 at 16:05, and then rises back to 1.00 at 16:06. The area under the line is shaded light green.

Memory usage history

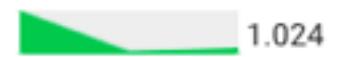


A line chart showing memory usage over time. The Y-axis is labeled 'Memory (bytes)' and has labels for 429 Mi, 858 Mi, 1.26 Gi, 1.68 Gi, and 1.89 Gi. The X-axis is labeled 'Time' and shows three points: 16:03, 16:04, and 16:06. A blue line starts at 429 Mi at 16:03, rises to 858 Mi at 16:04, and continues to rise to 1.89 Gi at 16:06. The area under the line is shaded light blue.

Replication controllers

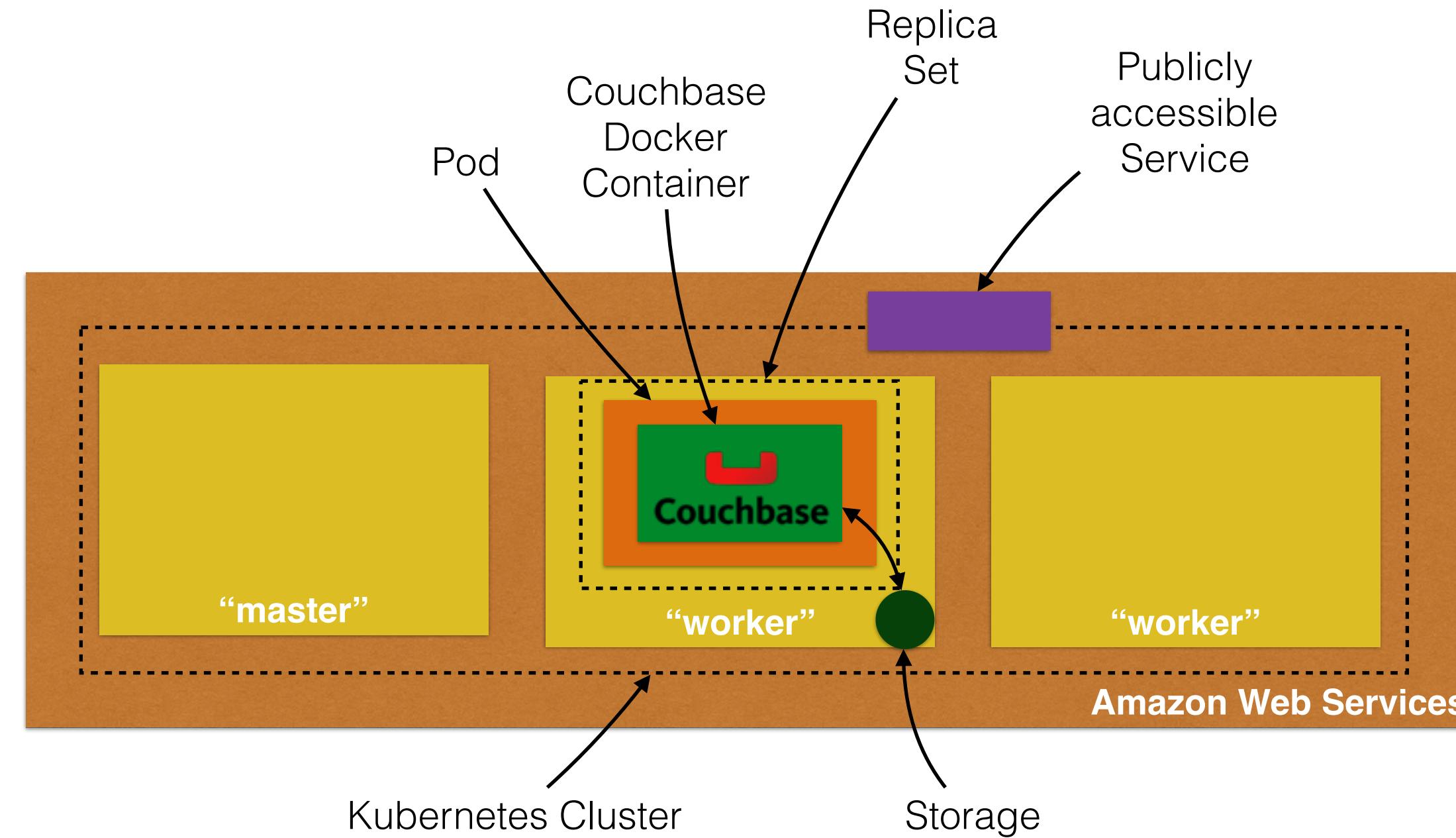
Name	Labels	Pods	Age	Images
couchbase-master-rc	app: couchbase-master-pod	1 / 1	3 minutes	arungupta/couchbase:k8s
couchbase-worker-rc	app: couchbase-worker-pod	3 / 3	a minute	arungupta/couchbase:k8s

Pods

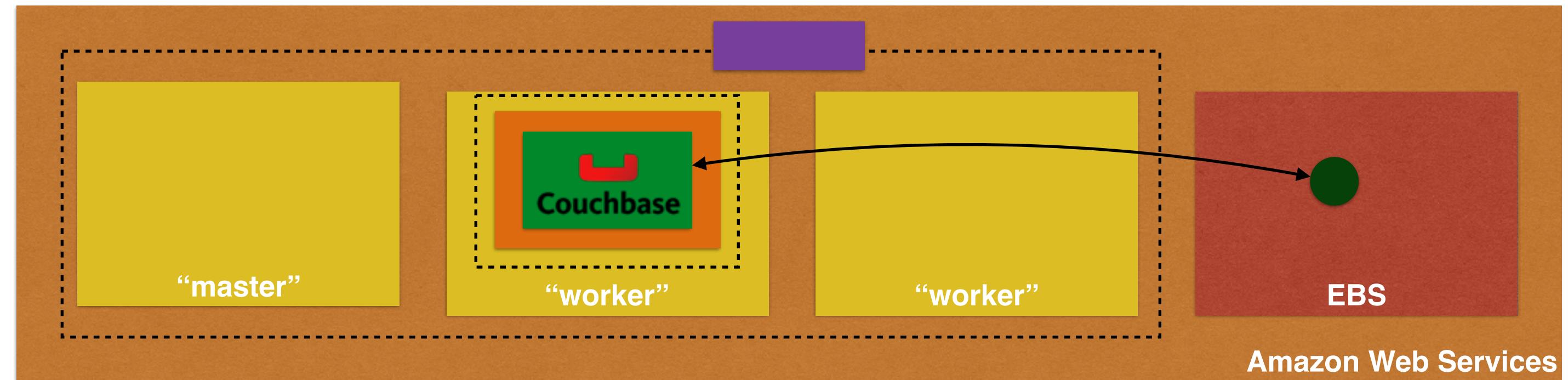
Name	Status	Restarts	Age	Cluster IP	CPU (cores)	Memory (bytes)	⋮	⋮
couchbase-master-rc...	Running	0	3 minutes	10.244.1.3	 1.024	 708 Ki		
couchbase-worker-rc-l...	Running	0	55 seconds	10.244.0.4	-	 310.121 Mi		
couchbase-worker-rc...	Running	0	55 seconds	10.244.2.10	-	 377.246 Mi		
couchbase-worker-rc...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

Persistent Volume



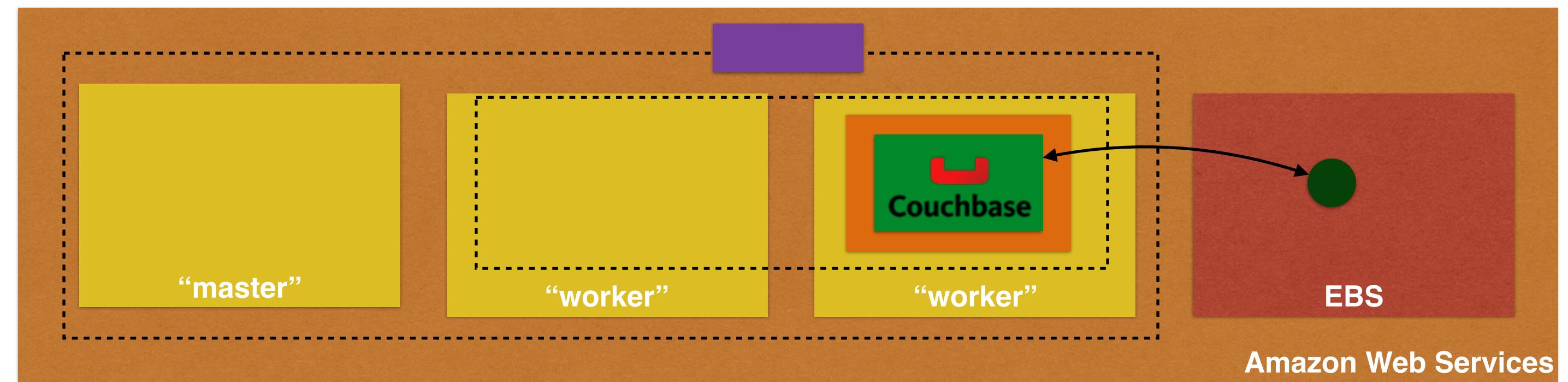


<http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs>



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

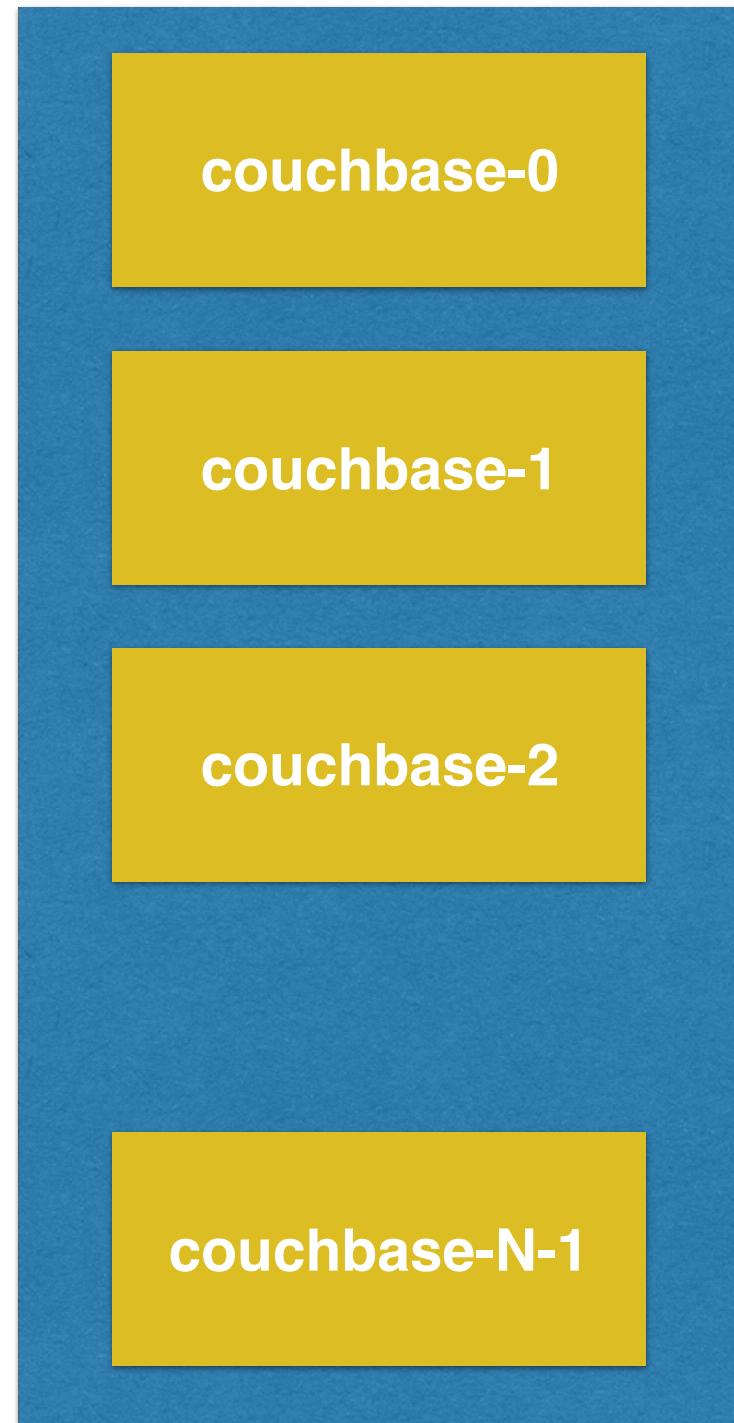
<http://blog.couchbase.com/2016/july/stateful-containers-kubernetes-amazon-ebs>





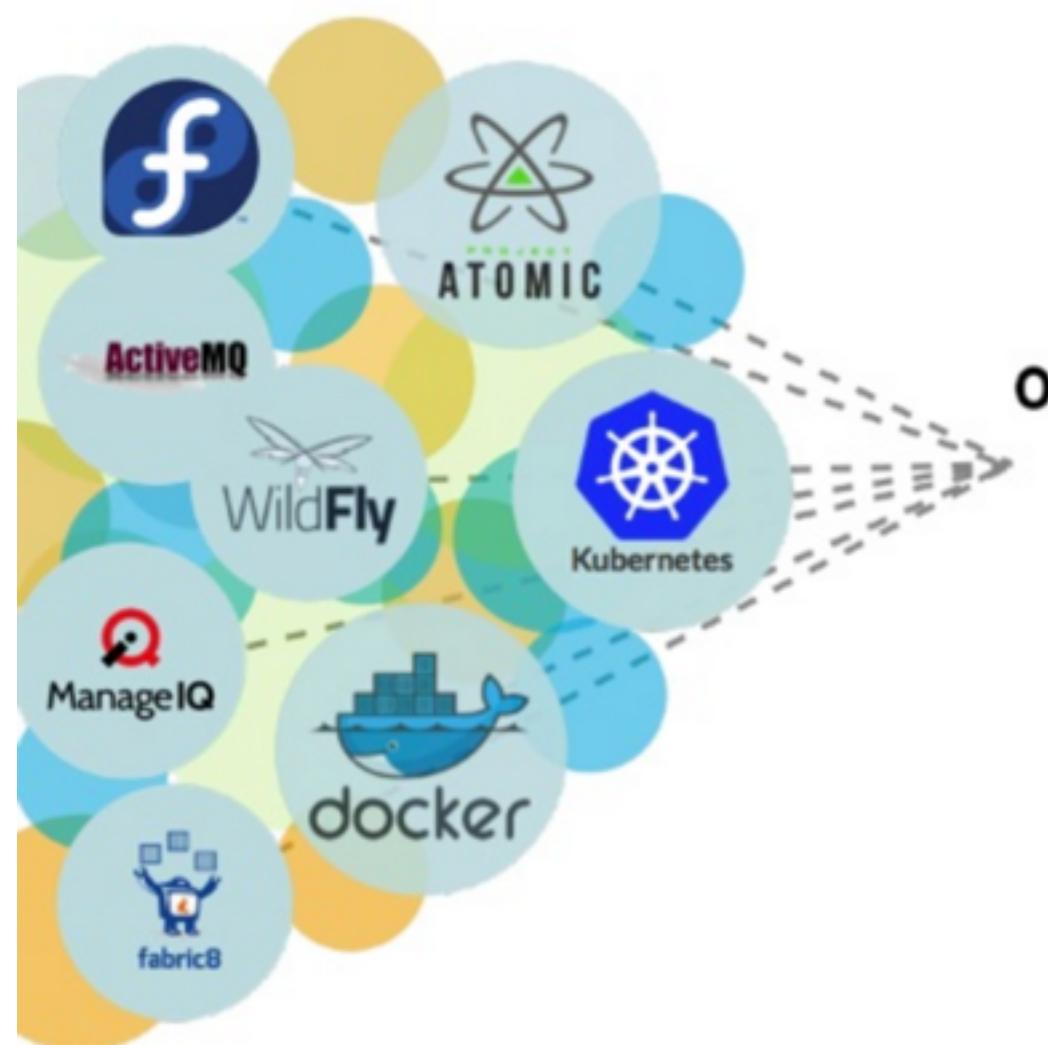
Pet Set

- 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>

Devoxx™



OPENSHIFT
origin





**RED HAT®
OPENSHIFT**
Container Platform

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



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

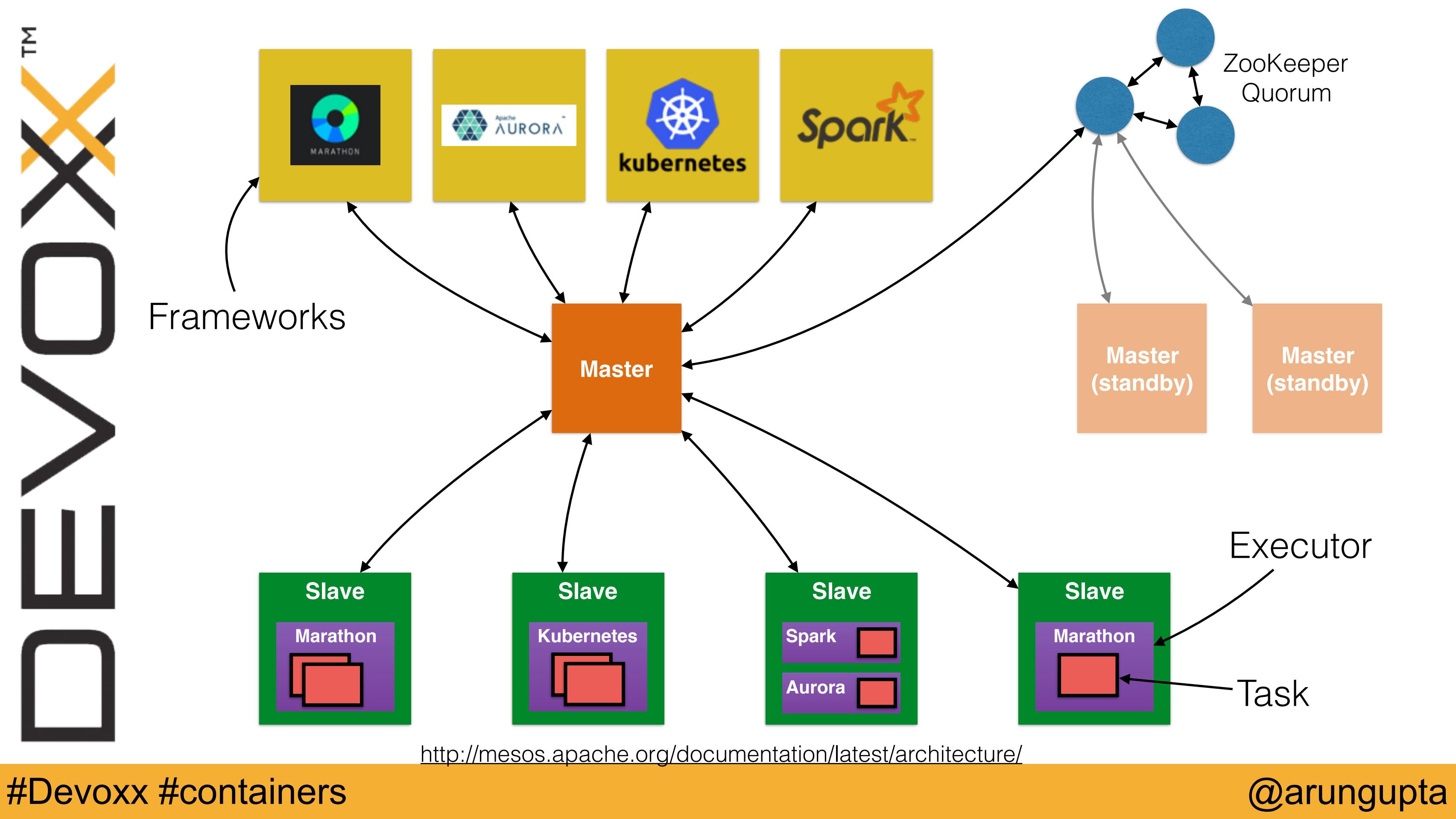
@arungupta



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





DC/OS



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

TM

X
VO
W
H
D
O

#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",
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 }
```

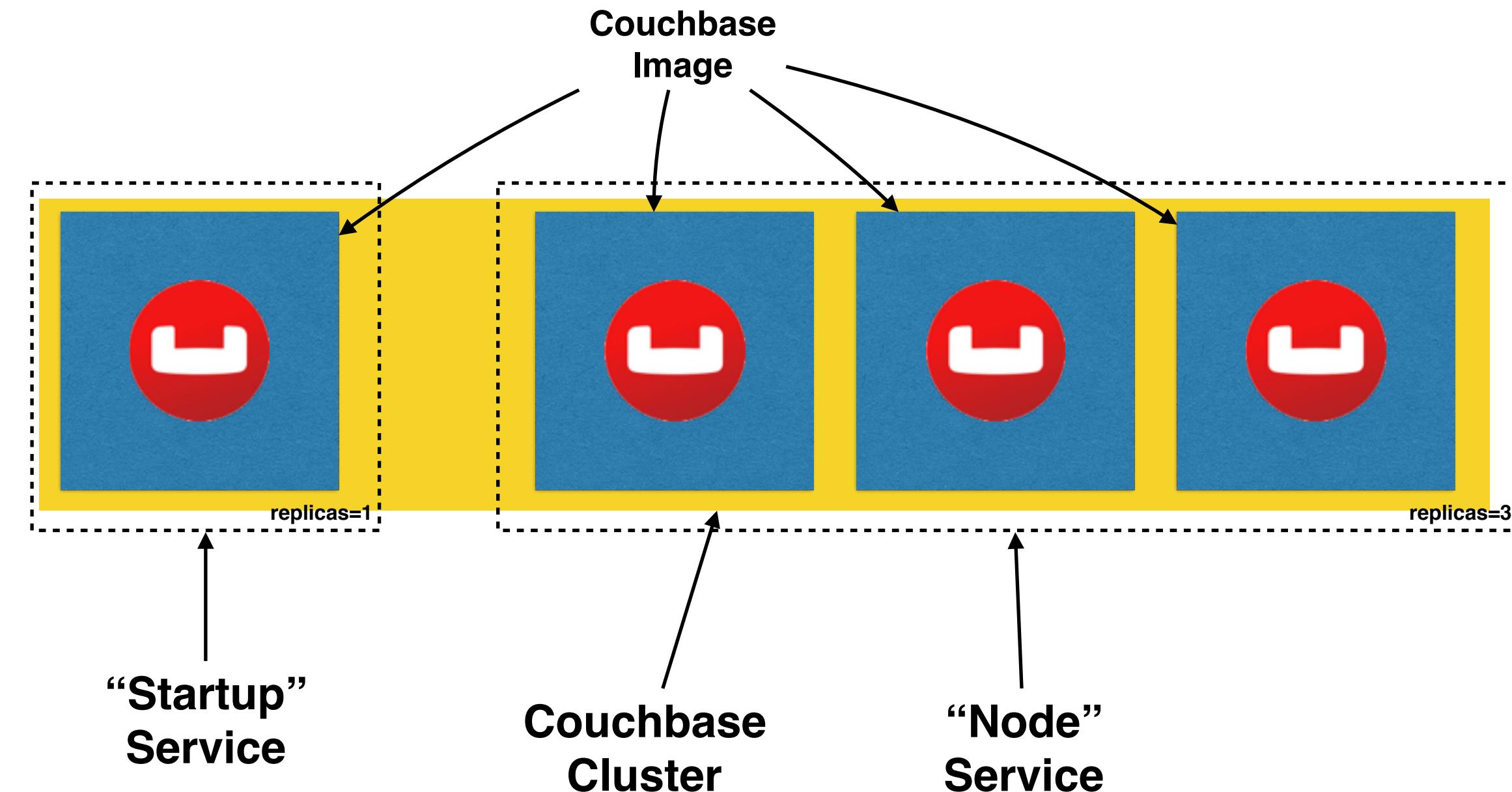
```
32     "labels": {
33         "HAProxy_GROUP": "external",
34         "HAProxy_0_VHOST": "DCOS-PublicSlaveLo-11B276BVFPDAM-950869174.us-west-2.elb.amazonaws.com"
35     },
36     "healthChecks": [
37         {
38             "protocol": "HTTP",
39             "path": "/pools",
40             "ignoreHttp1xx":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

@arungupta

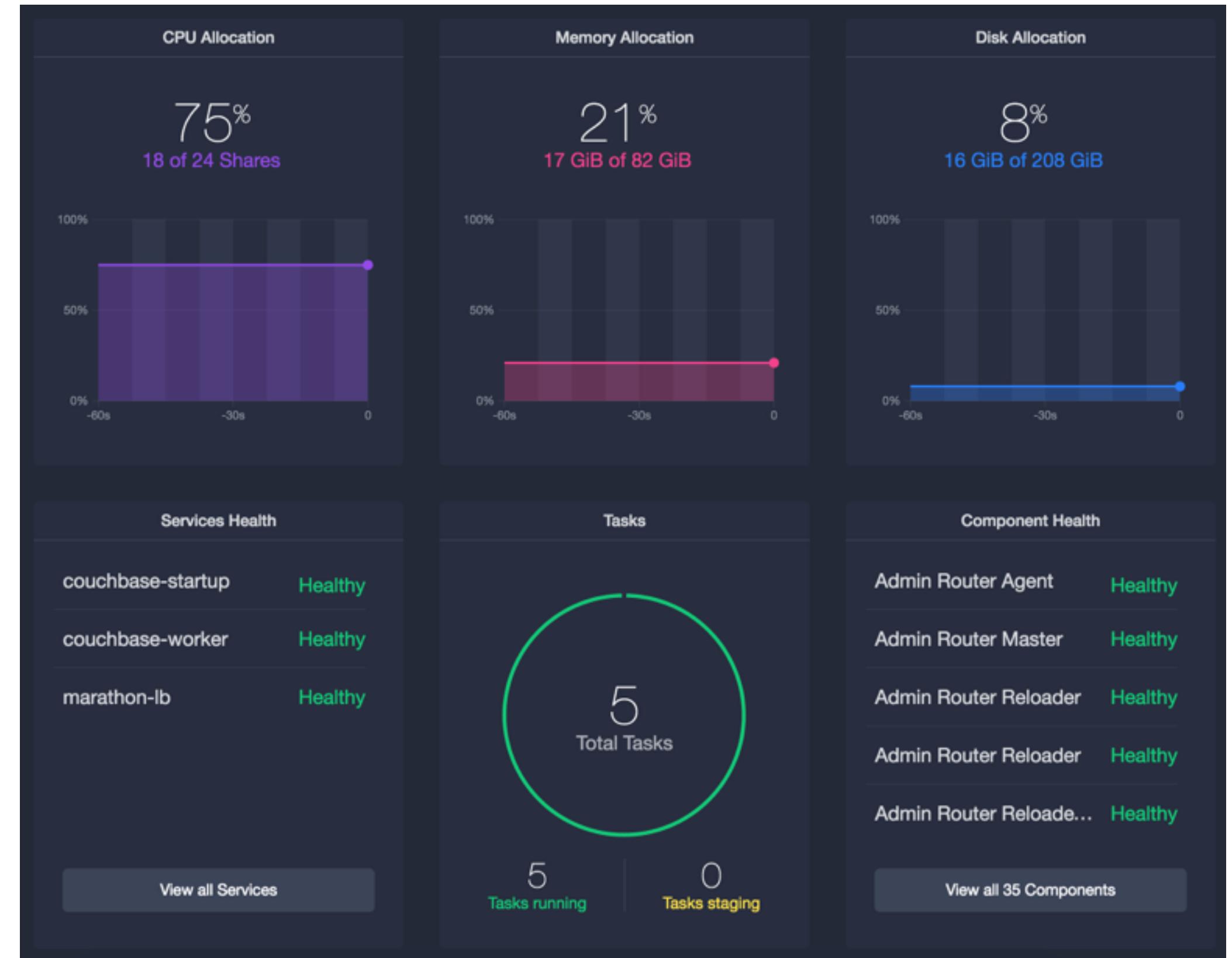


```
1  {
2      "volumes": null,
3      "id": "/couchbase-worker",
4      "cmd": null,
5      "args": null,
6      "user": null,
7      "env": [
8          "TYPE": "WORKER",
9          "COUCHBASE_MASTER": "couchbase-startup.marathon"
10     ],
11     "instances": 1,
12     "cpus": 4,
13     "mem": 4096,
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 }
```

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



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

- 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
- Slides: <https://github.com/javaee-samples/docker-java/blob/master/slides/2016-Devoxx-Migrate-VM-to-Containers.pdf>