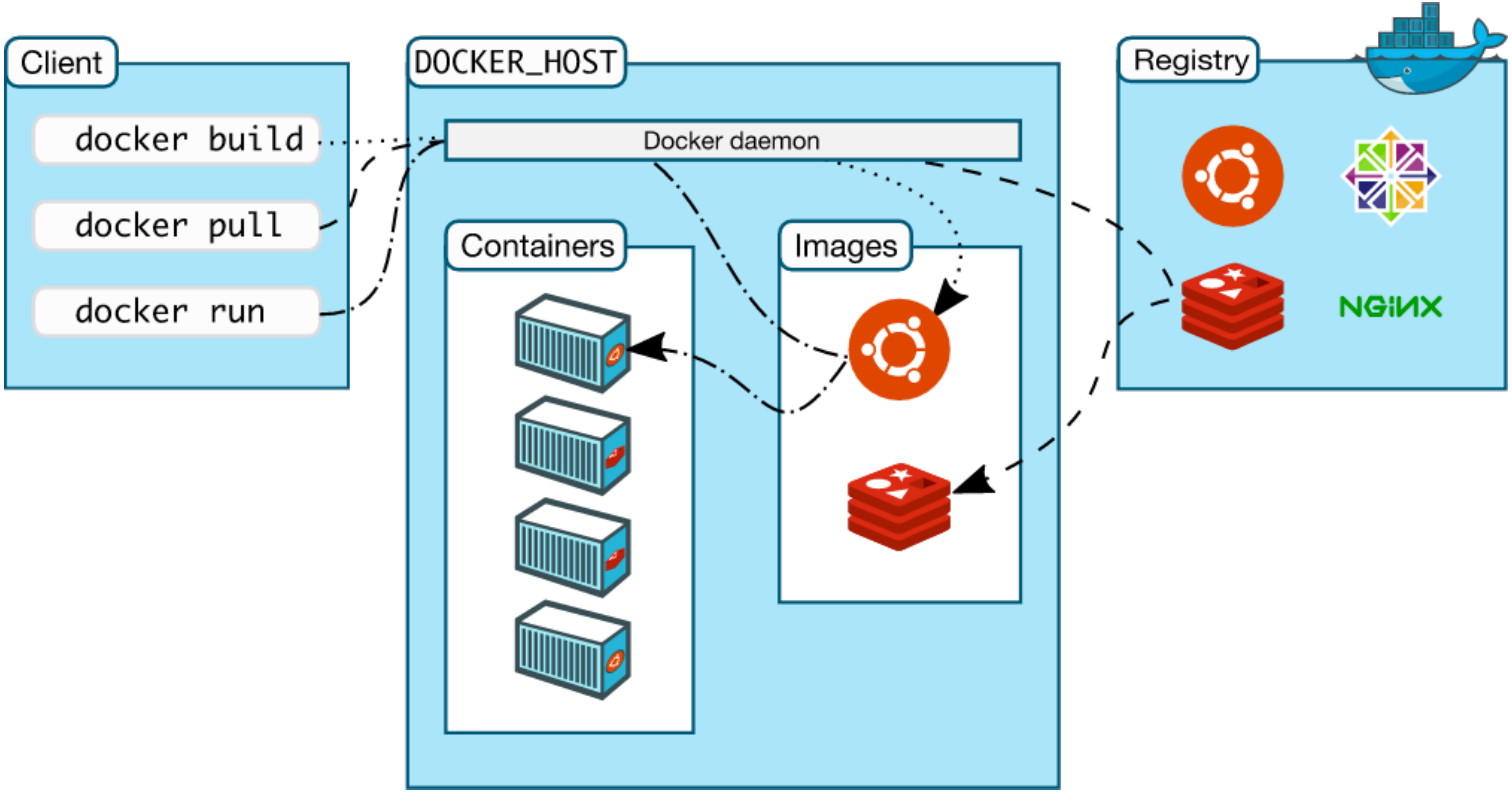
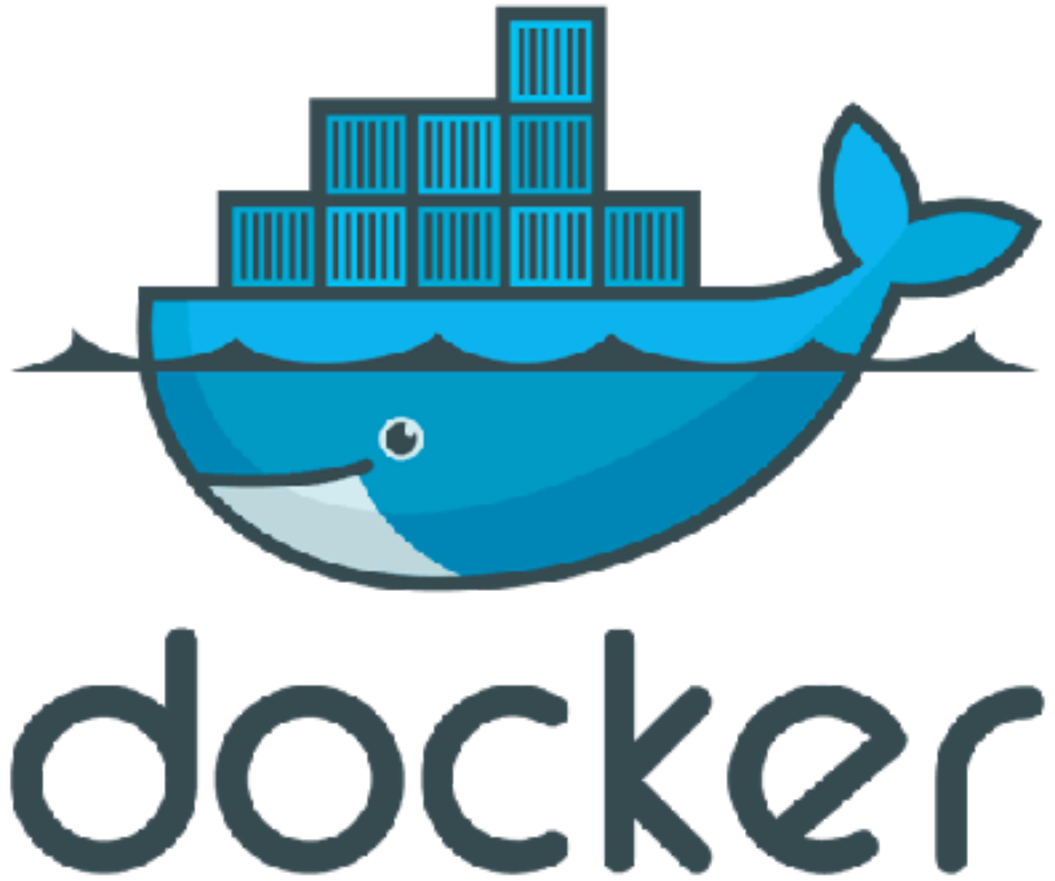
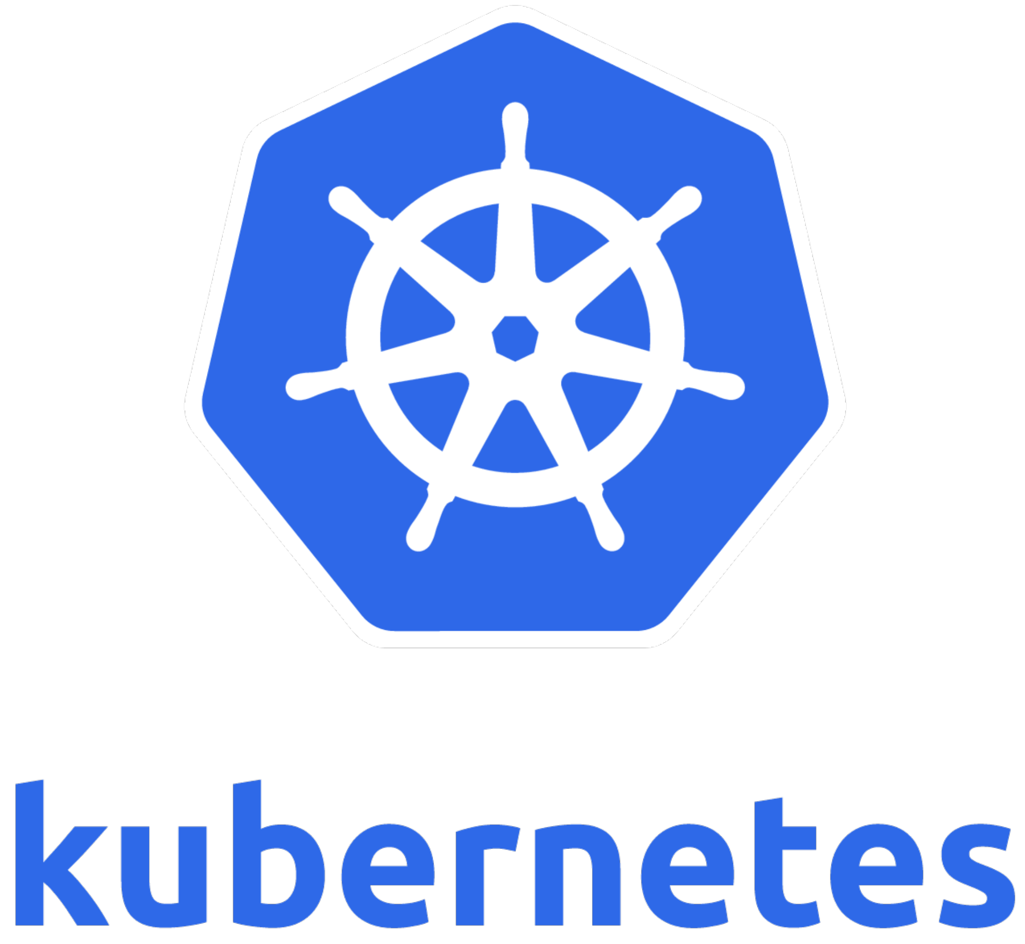
Container Orchestration on Amazon Web Services

Arun Gupta, @arungupta

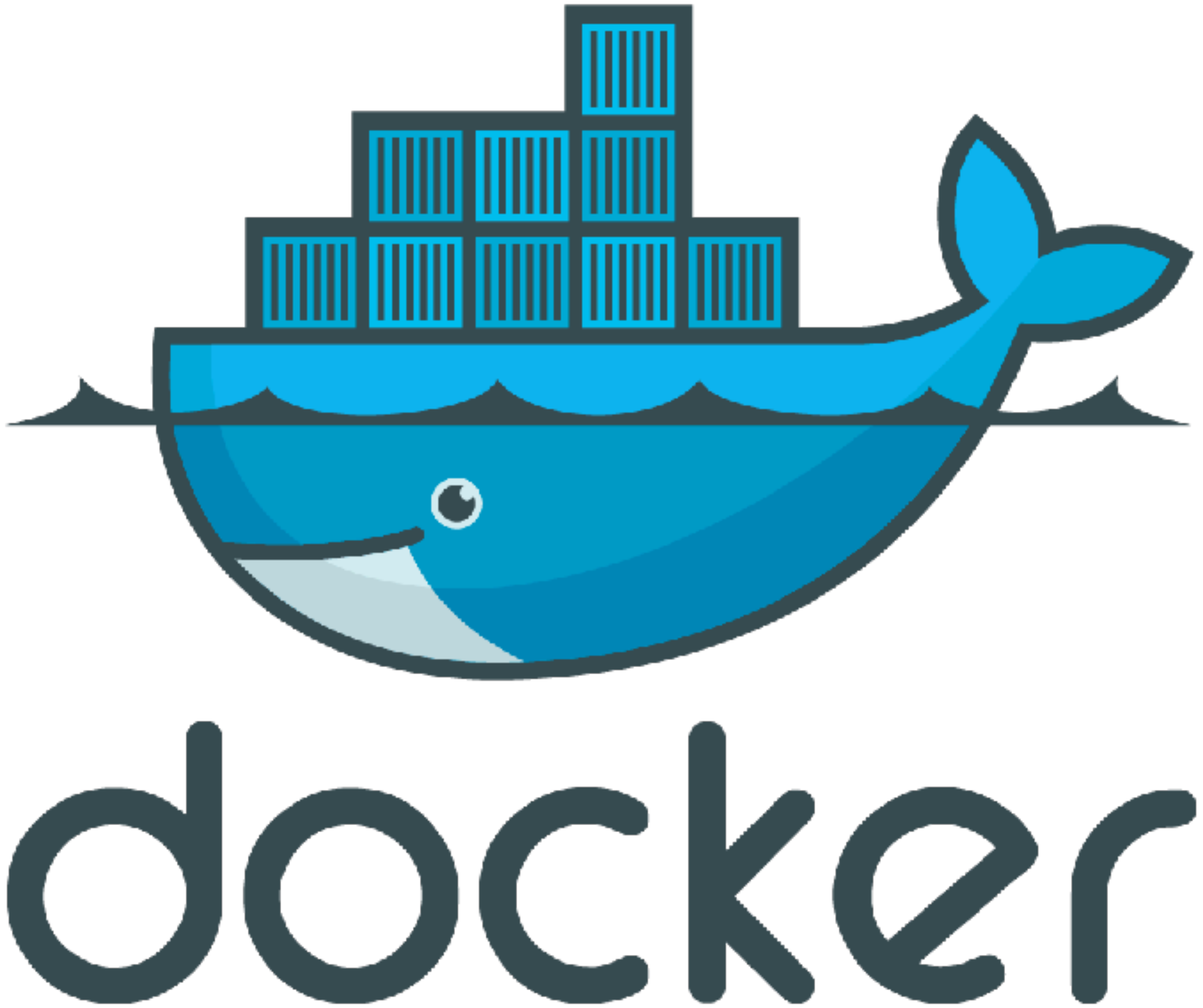
Docker Workflow




docker

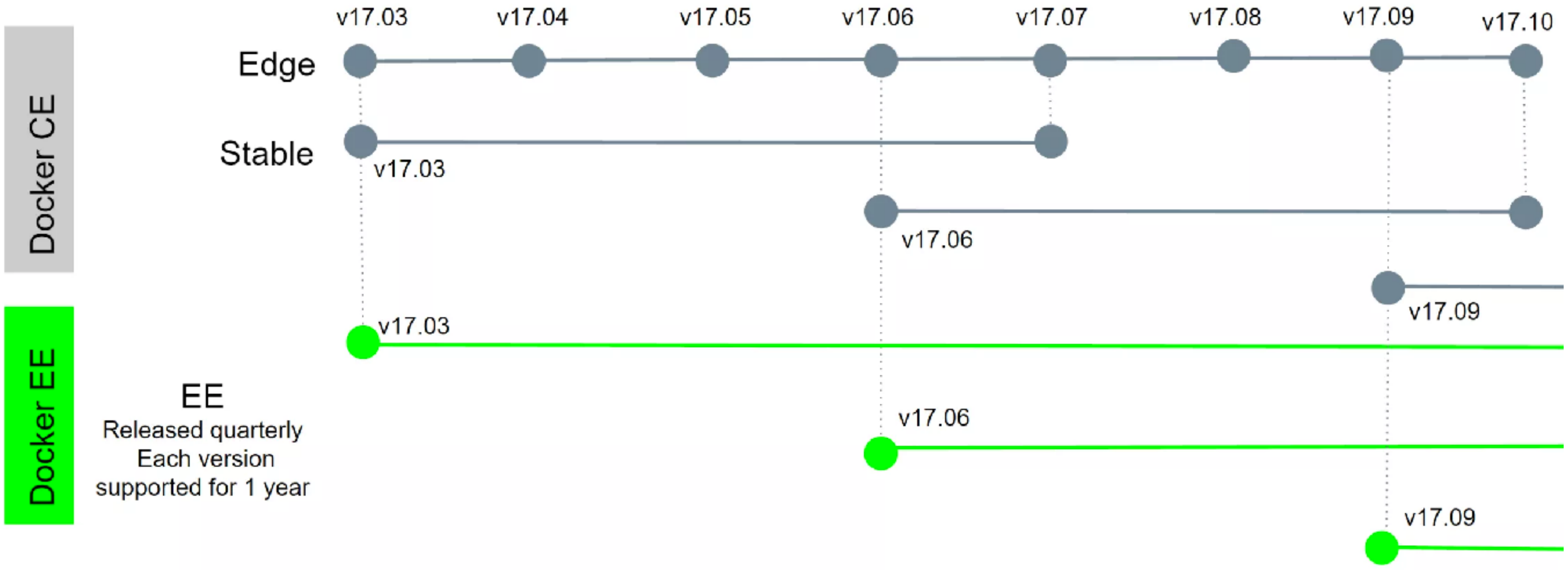

kubernetes

MESOSPHERE


docker

Development using Docker

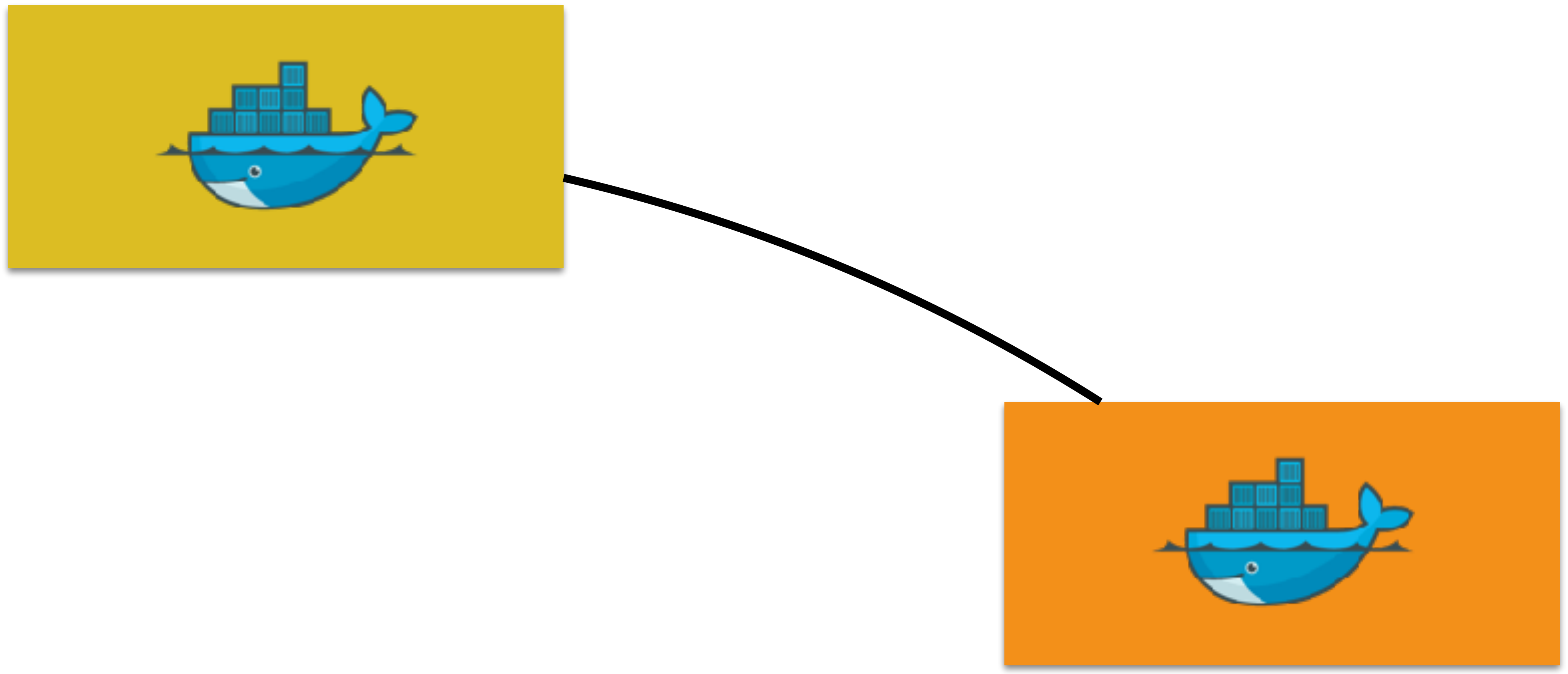
- Docker Community Edition
  - Docker for Mac/Windows/Linux
  - Monthly edge and quarterly stable releases
  - Native desktop or cloud provider experience



Swarm-mode: Initialize

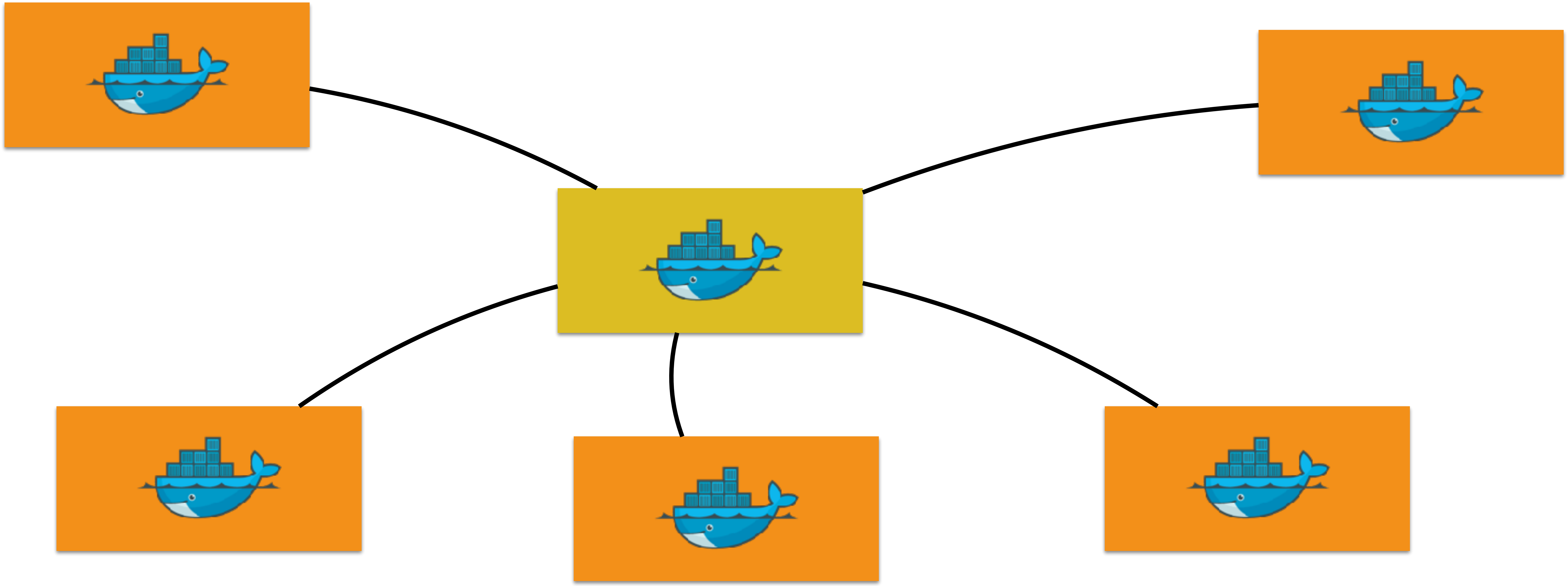
```
docker swarm init --listen-addr <ip>:2377 --secret <SECRET>
```

Swarm-mode: Add Worker



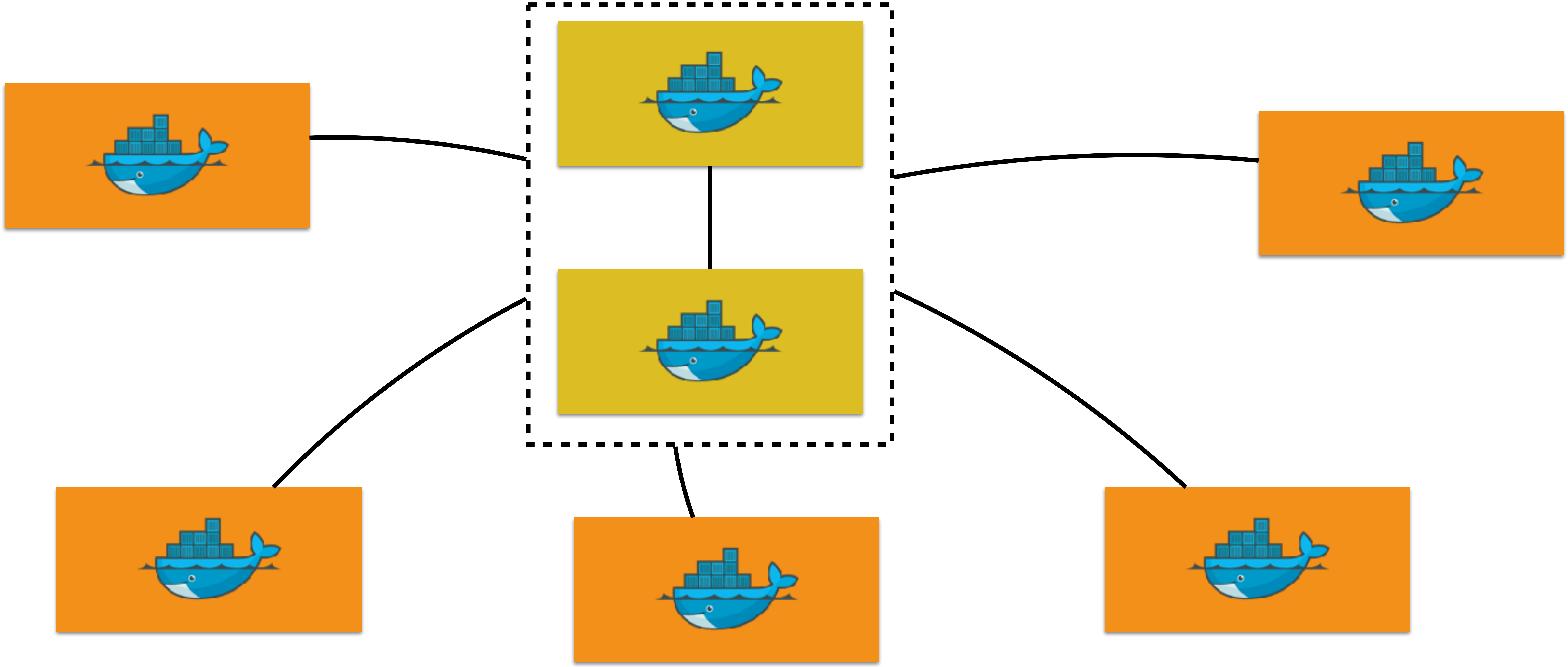
```
docker swarm join --secret <SECRET> <manager>:2377
```

Swarm-mode: Add More Workers



```
docker swarm join --secret <SECRET> <manager>:2377
```

Swarm-mode: Primary/Secondary Master



```
docker swarm join --manager --secret <SECRET> --listen-addr
<master2>:2377 <master1>:2377
```

Docker for AWS

- CloudFormation template
- Integrated with AWS Infrastructure
  - Autoscaling Groups (ASG)
  - Elastic Load Balancer (ELB)
  - Elastic Block Store (EBS)

Deploy Docker
Community Edition (CE)
for AWS (stable)

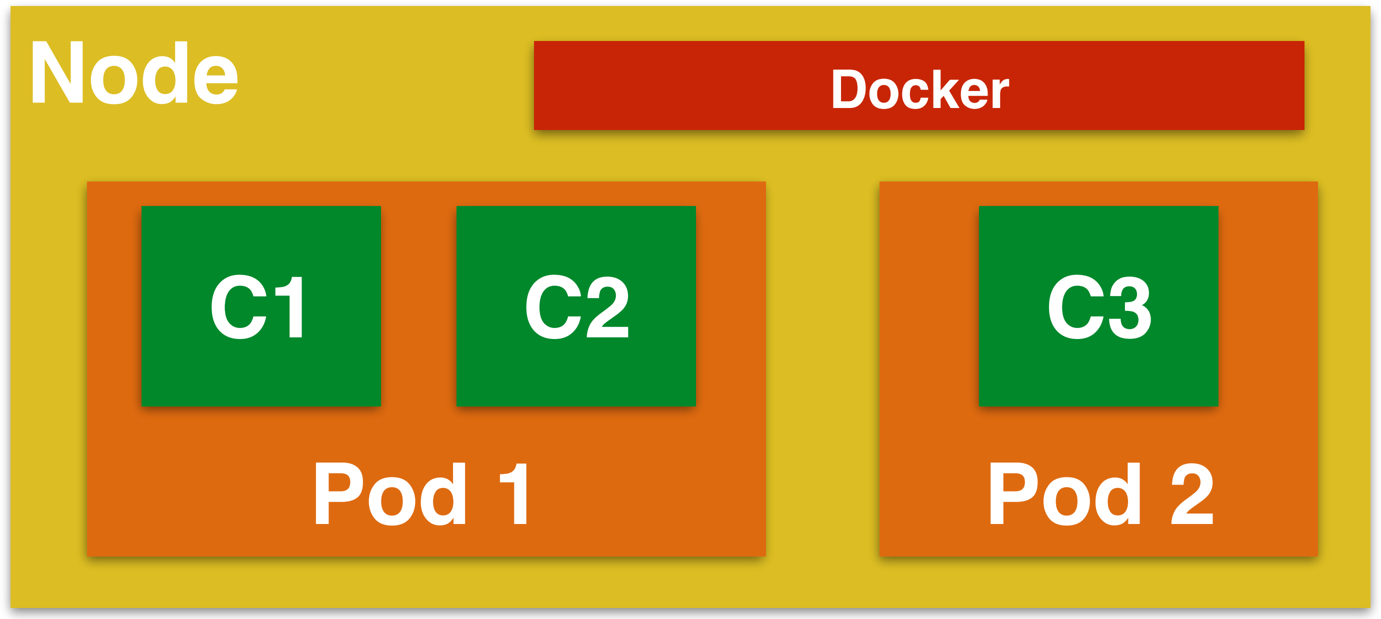
Deploy Docker
Community Edition (CE)
for AWS (edge)

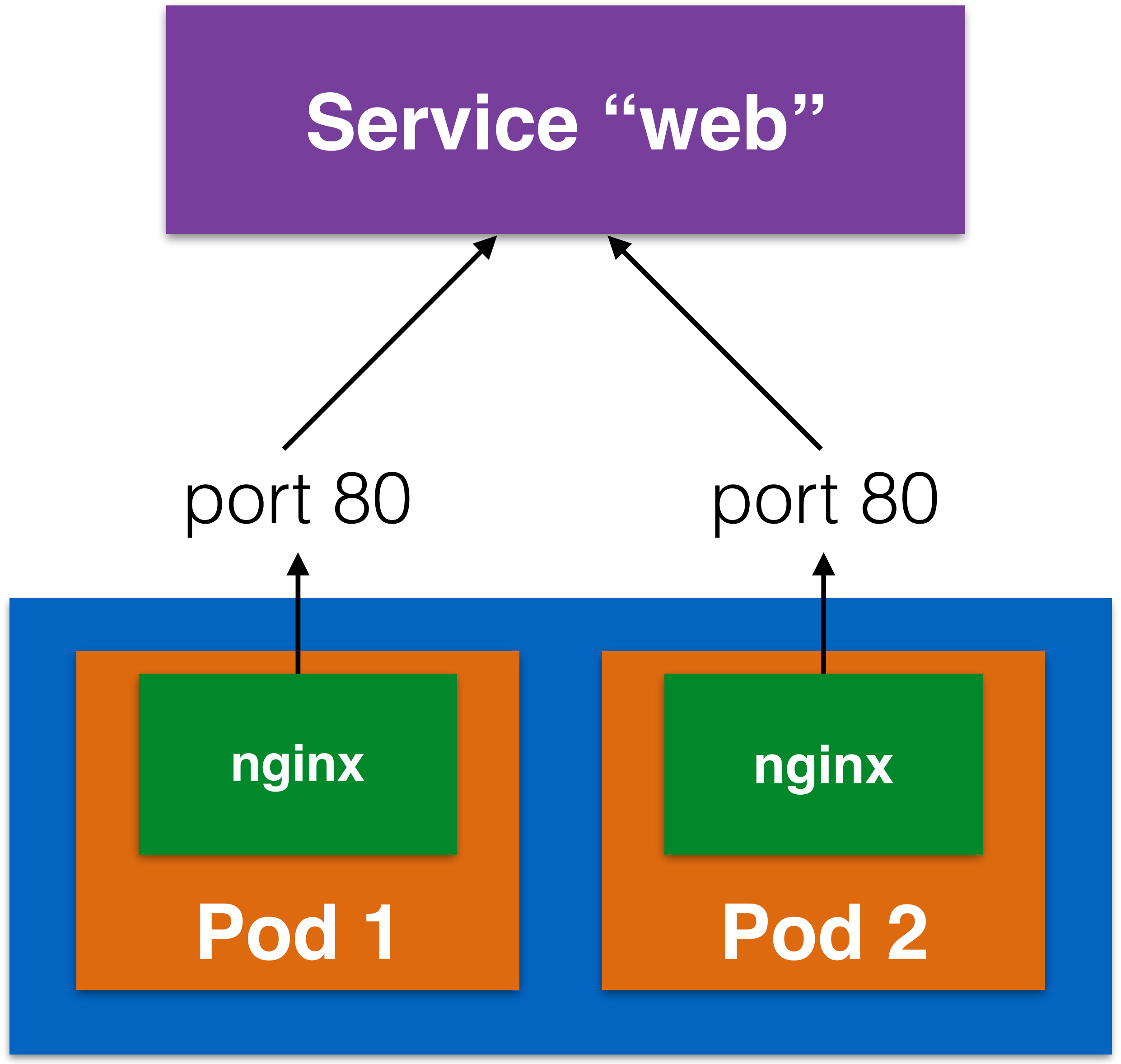
Deploy Docker
Community Edition (CE)
for AWS (edge)
uses your existing VPC

kubernetes

Kubernetes Concepts

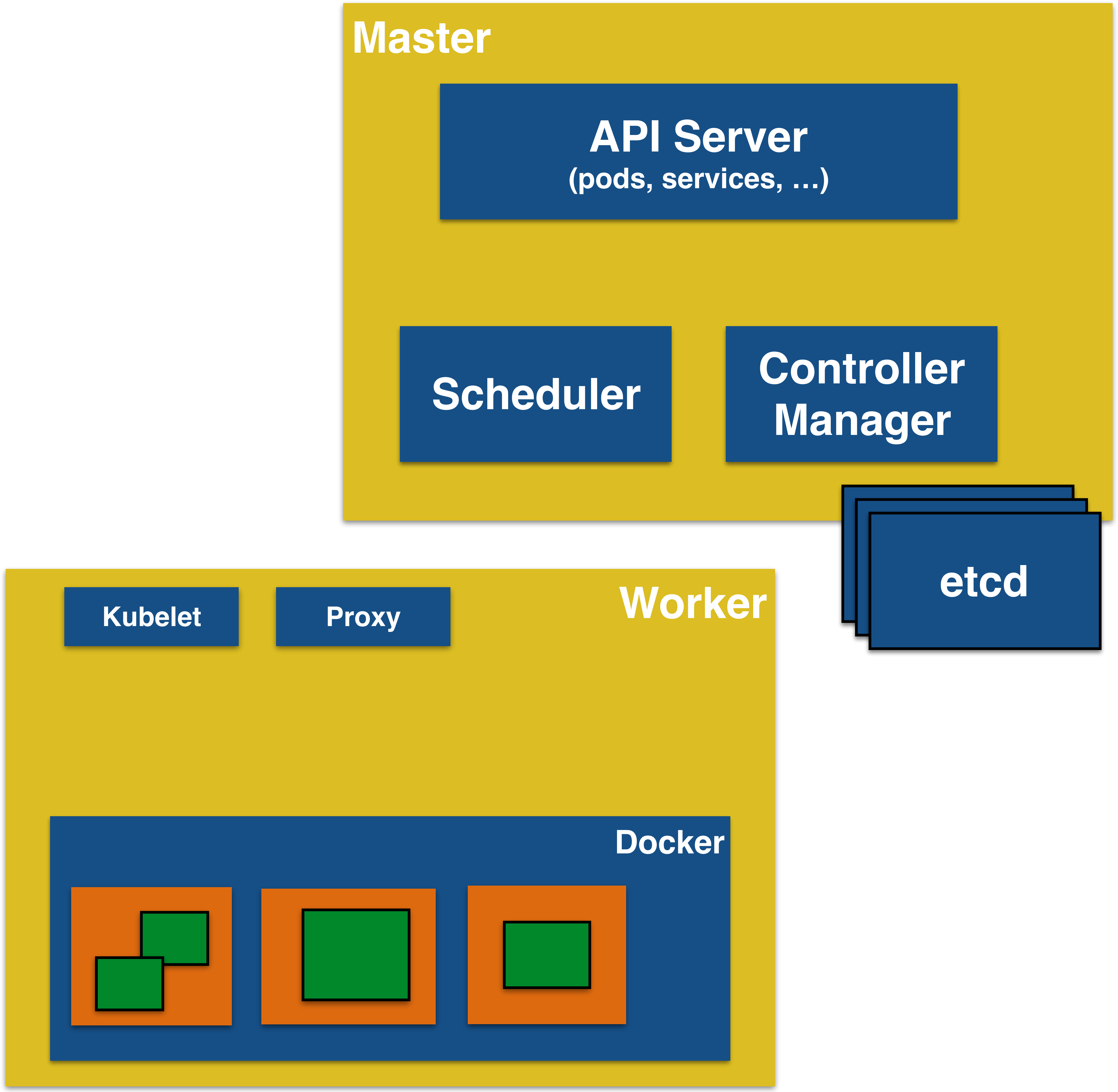
- **Pods**: collocated group of Docker containers that share an IP and storage volume
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to organize and select group of objects
- **Replica Set**: manages the lifecycle of pods and ensures specified number are running

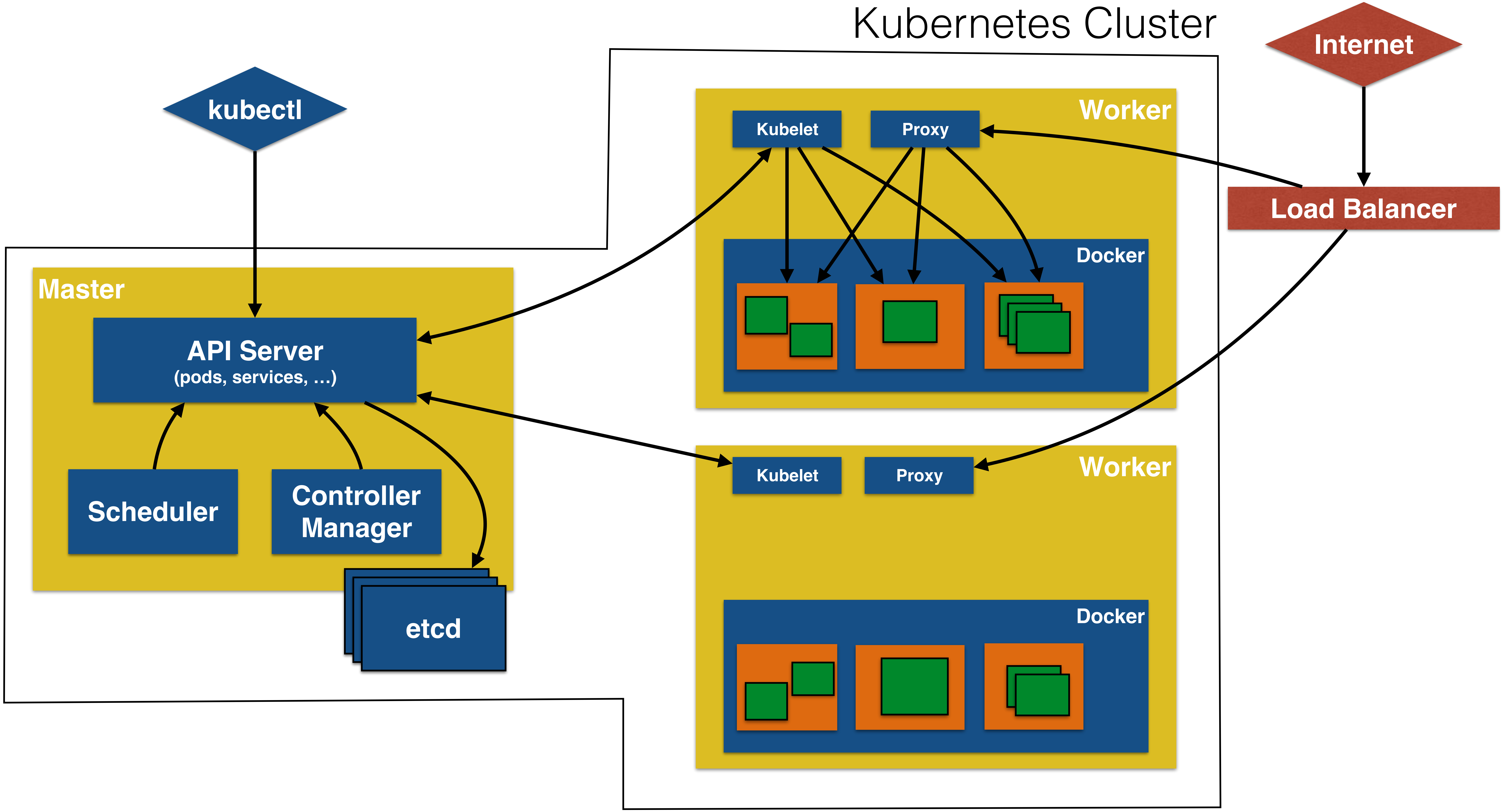




Core Concepts: Kubernetes

- **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
Internet
kubectl
Worker
Kubelet
Proxy
Load Balancer
Docker
Master
API Server
(pods, services, ...)
Scheduler
Controller Manager
Worker
Kubelet
Proxy
Docker
etcd

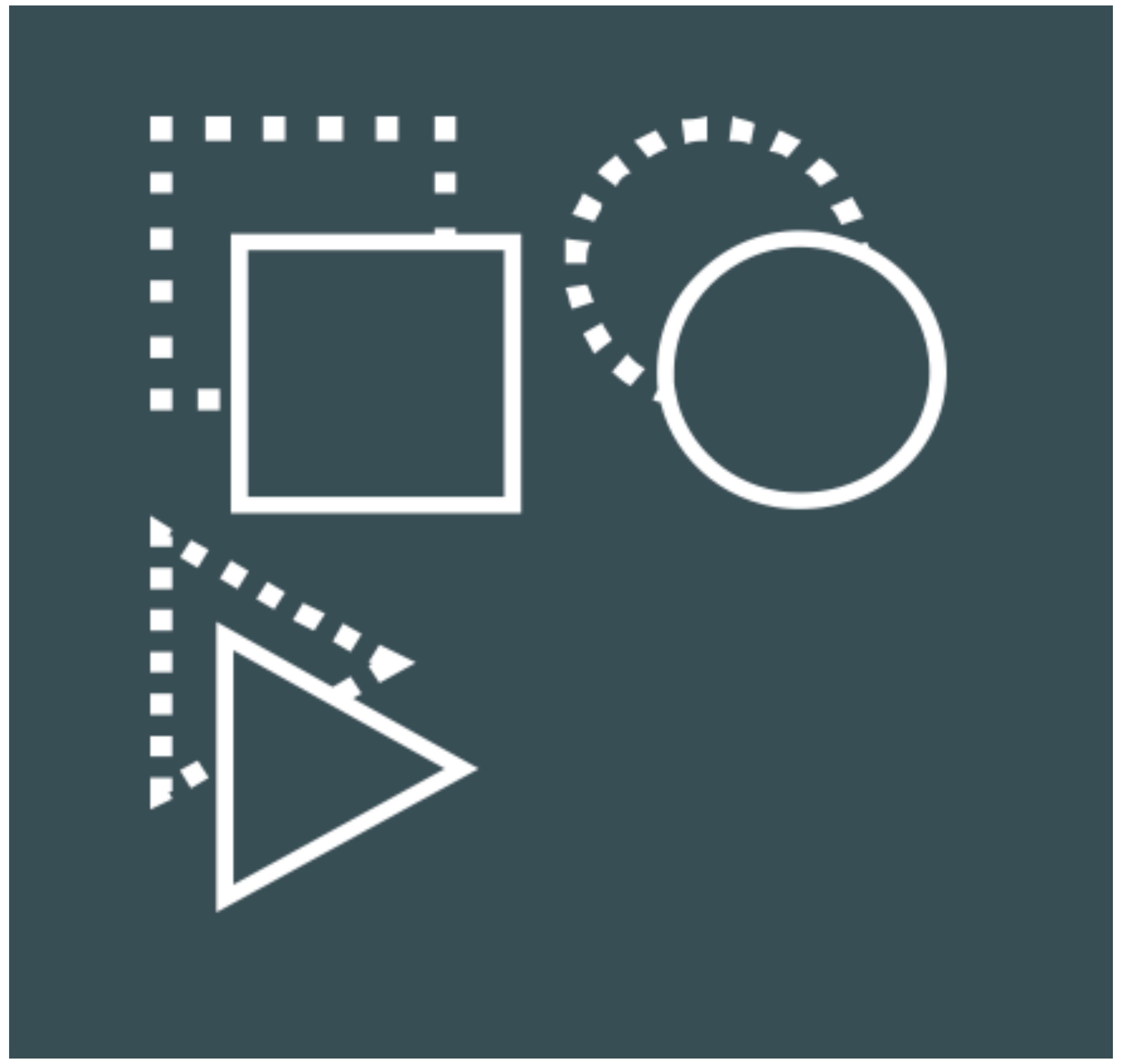
kubectl

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

Kubernetes on AWS

- Single node cluster
  - **minikube**
- Multi-node cluster on AWS
  - **kops**: github.com/kubernetes/kops
  - **kube-aws**: github.com/kubernetes-incubator/kube-aws
  - **Heptio**: github.com/aws-quickstart/quickstart-heptio
- Google Cloud, Azure, Tectonic, OpenShift …

Amazon EC2 Container Service

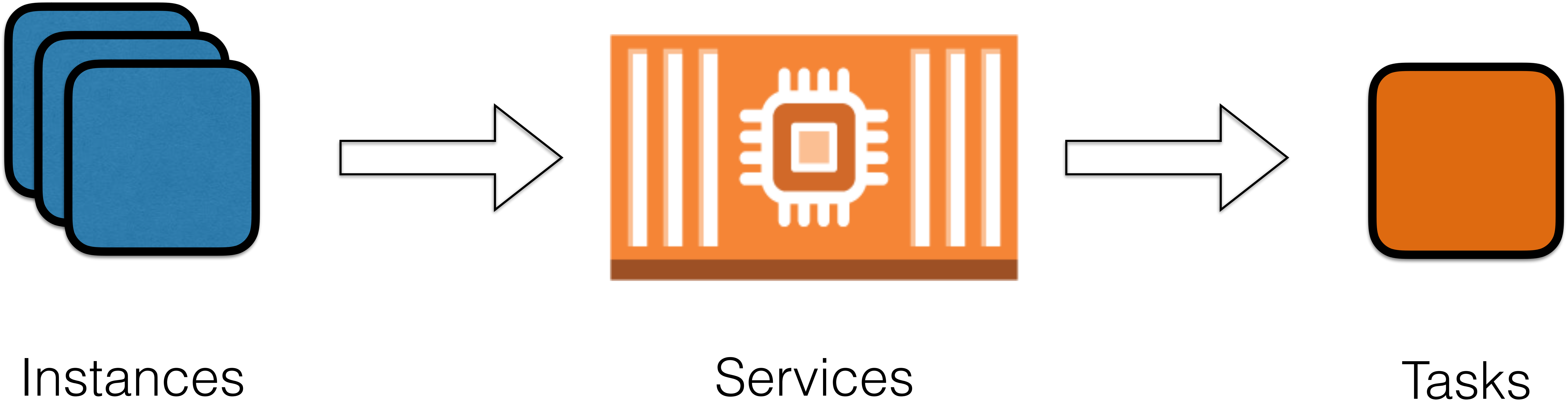


Cluster Management

Container Orchestration

Deep AWS Integration

Mapping to EC2 Workloads



Instances

Services

Tasks

MESOSPHERE

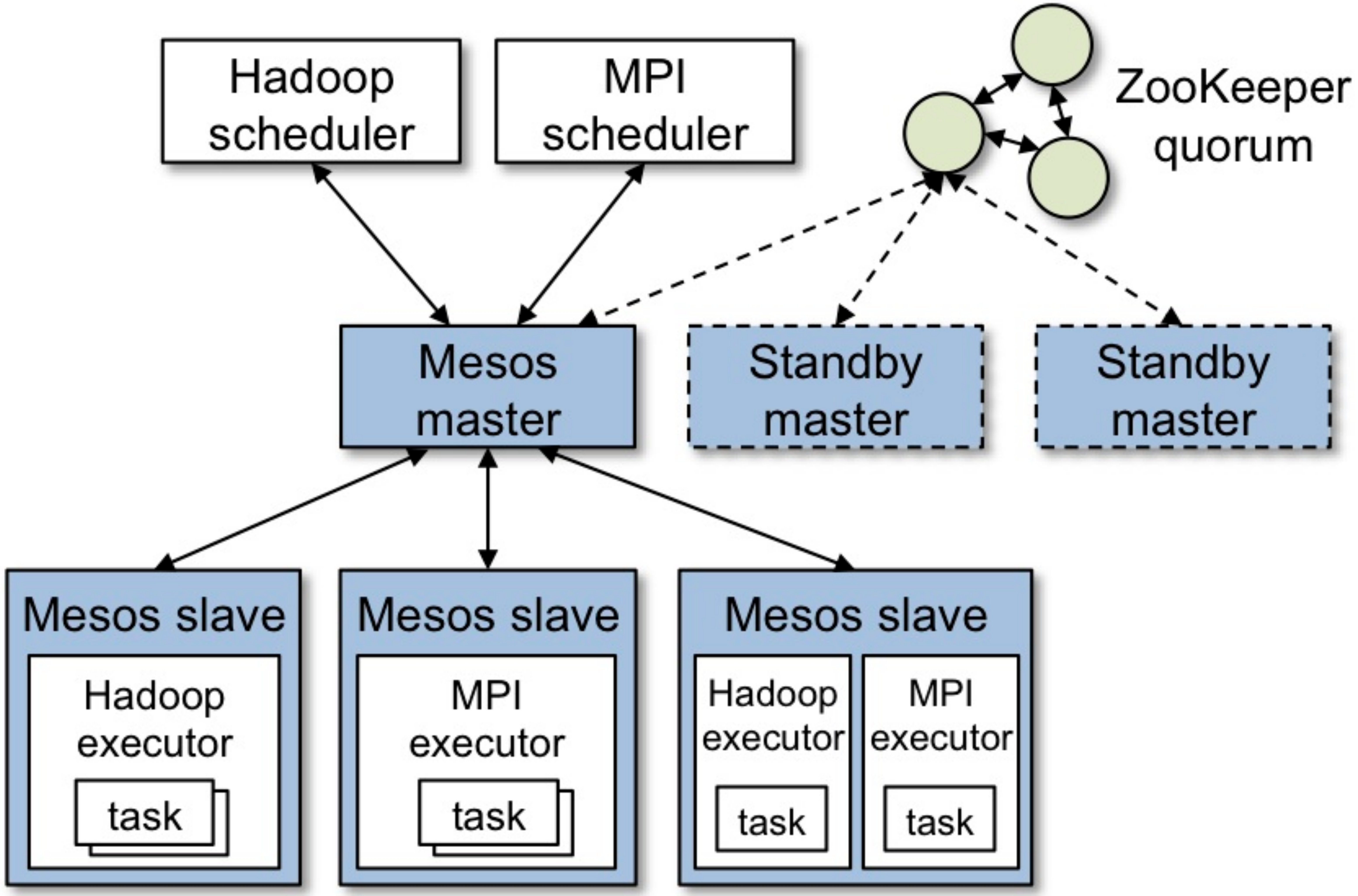
Mesos

- Open source cluster manager
- Developed at 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

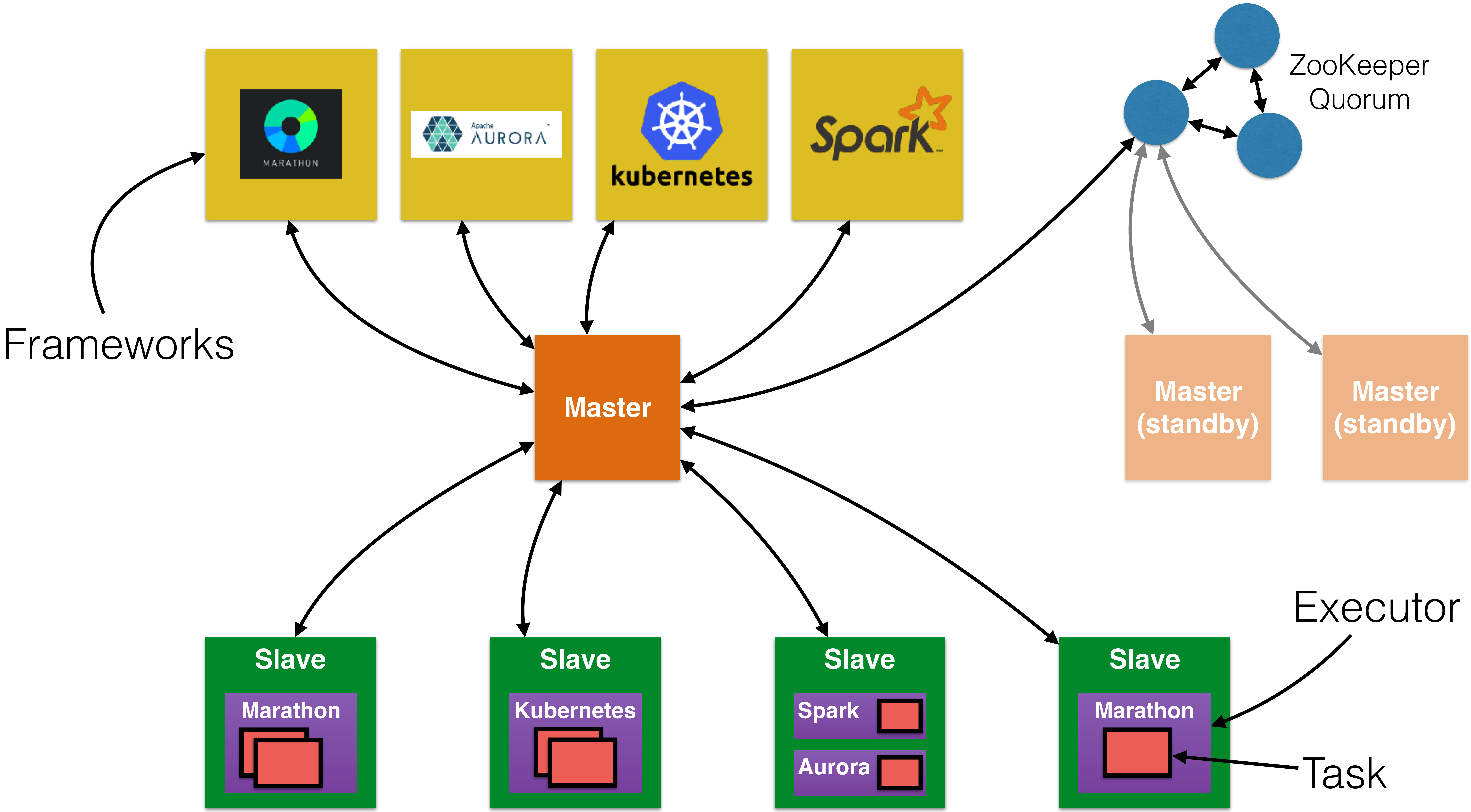
Mesos

- Master Slave architecture
- Fault tolerant
  - Leader election via ZooKeeper
- Multi platform
  - Ubuntu, Mac OS, CentOS

Mesos Architecture



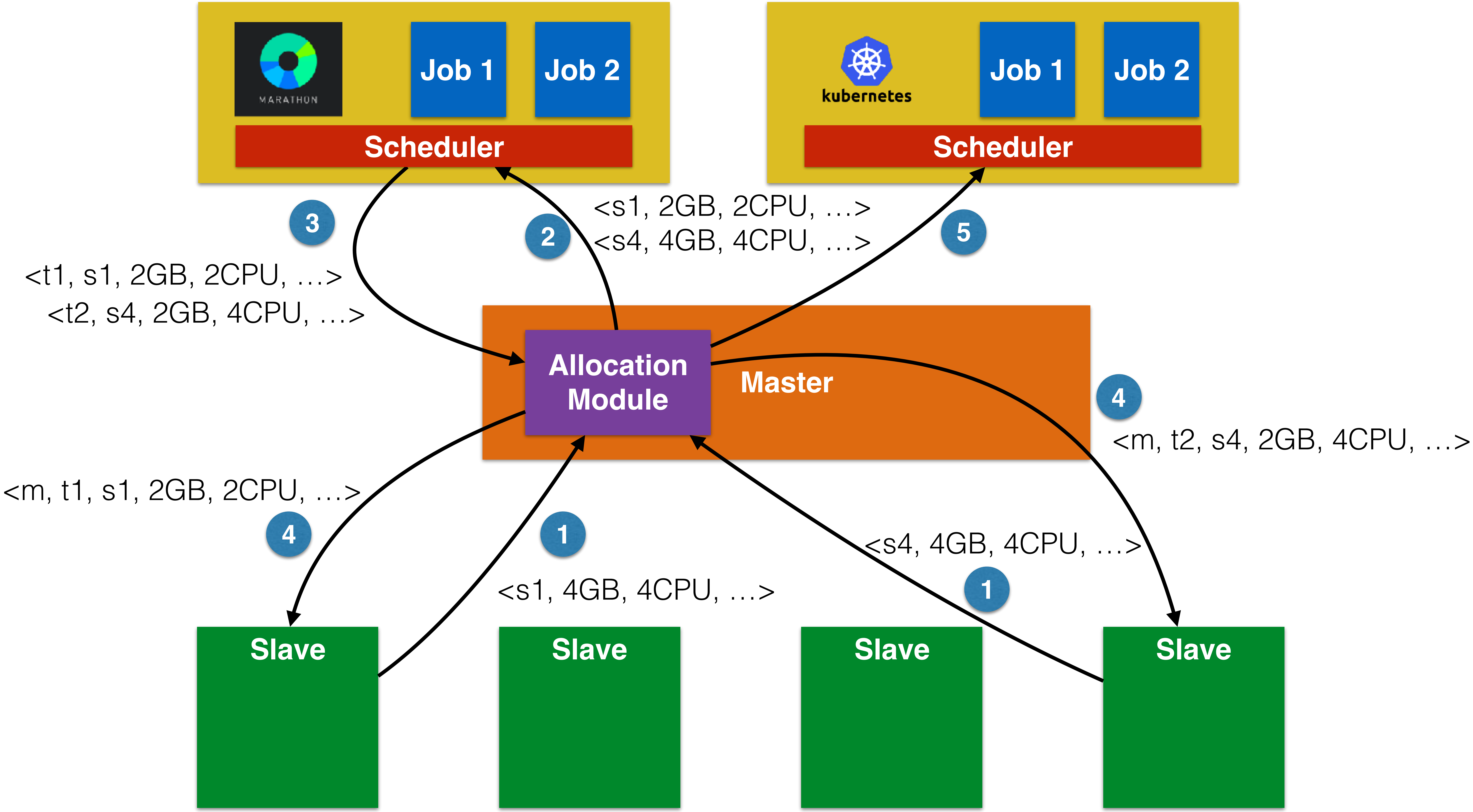
http://mesos.apache.org/documentation/latest/architecture/



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

Frameworks

- Frameworks are targeted at a use case and domain-specific
  - Master node “offers” resources to each framework
  - Framework “accepts” the offer and execute applications
- Framework has “scheduler” and “executor”
  - Scheduler registers with the master for “offer”
  - Executor launched on slave nodes to run the task
    - Passes a description of the task to run



MARATHON

Job 1

Job 2

Scheduler

kubernetes

Job 1

Job 2

Scheduler

3

<t1, s1, 2GB, 2CPU, …>
<t2, s4, 2GB, 4CPU, …>

2

<s1, 2GB, 2CPU, …>
<s4, 4GB, 4CPU, …>

5

Allocation Module

Master

4

<m, t2, s4, 2GB, 4CPU, …>

<m, t1, s1, 2GB, 2CPU, …>

4

1

<s1, 4GB, 4CPU, …>

<s4, 4GB, 4CPU, …>

1

Slave

Slave

Slave

Slave

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

Thanks!

Arun Gupta, @arungupta

github.com/javaee-samples/docker-java/tree/master/slides