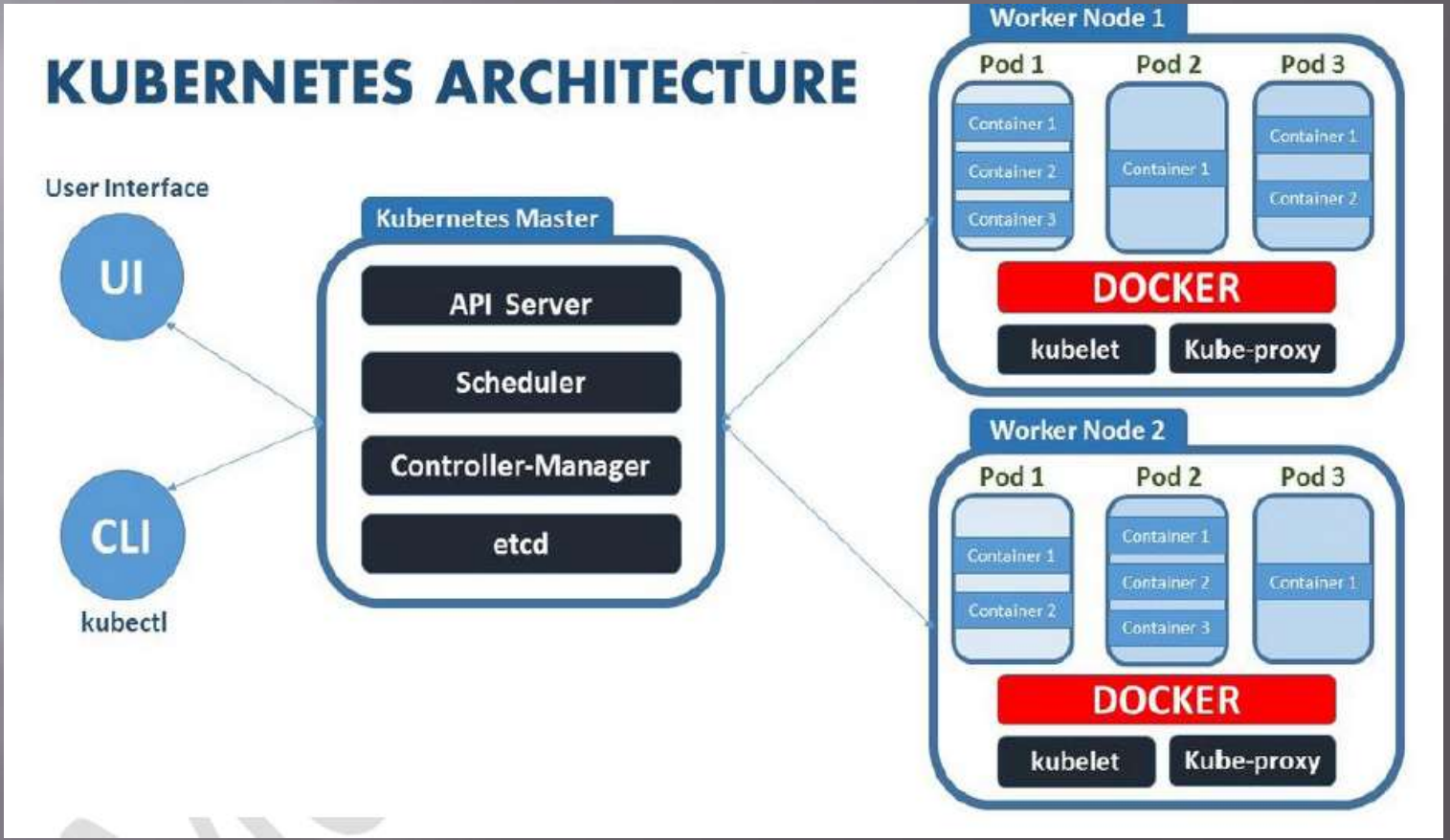
# Kubernets Architecture

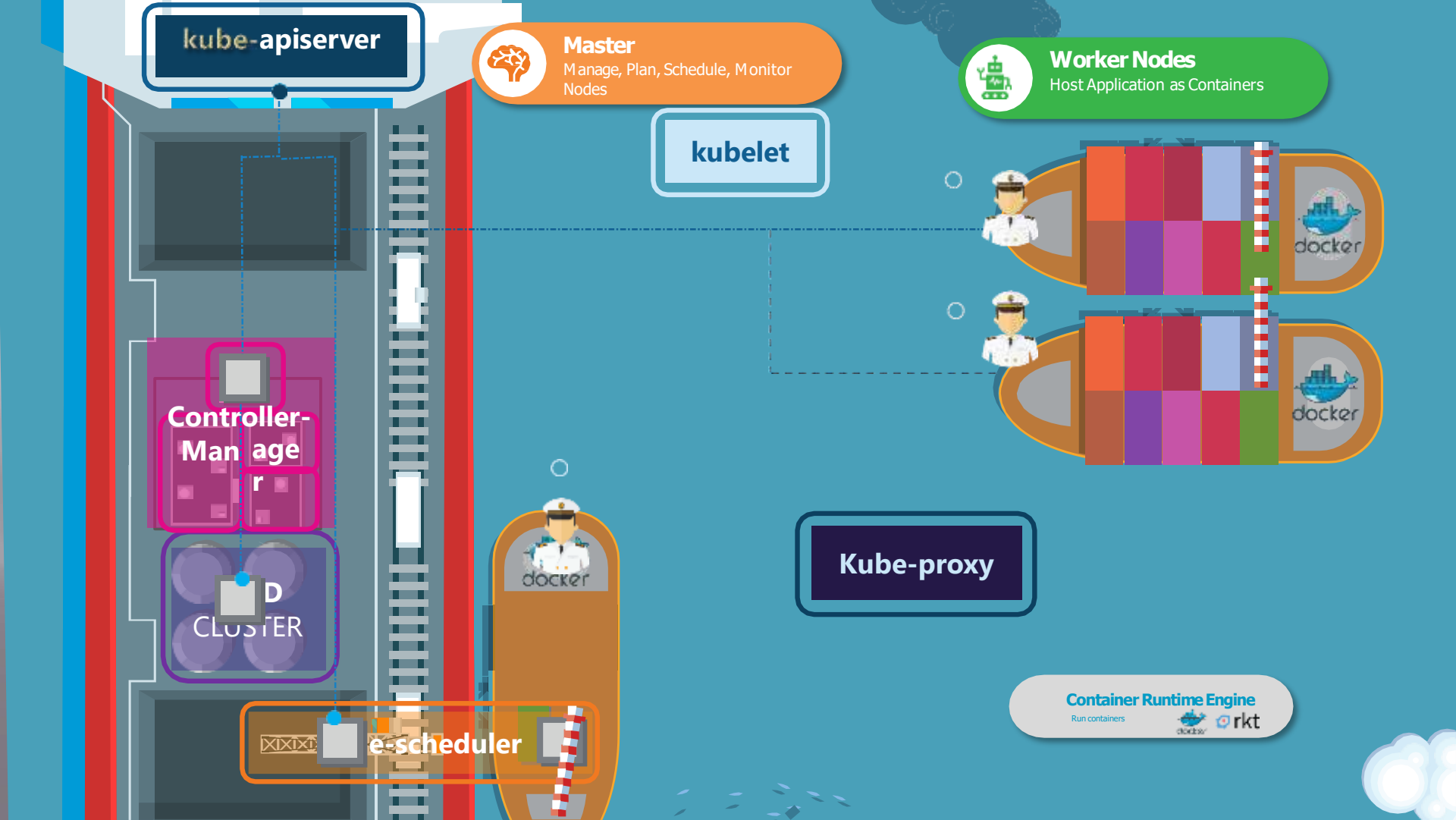
Ravindra Kudache

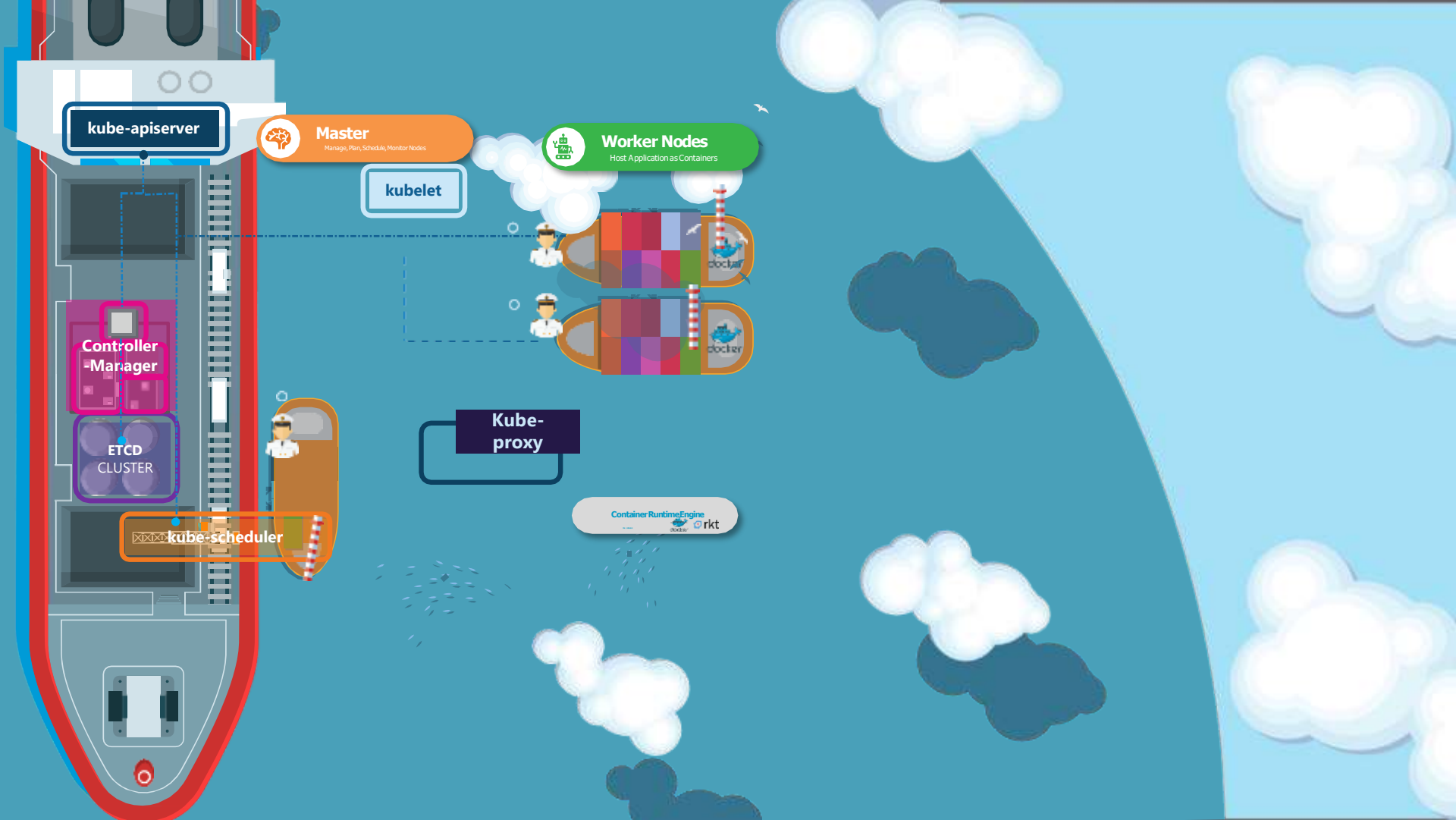

KUBERNETES ARCHITECTURE
User Interface
UI
CLI
kubectl
Kubernetes Master
API Server
Scheduler
Controller-Manager
etcd
Worker Node 1
Pod 1
Container 1
Container 2
Container 3
Pod 2
Container 1
Pod 3
Container 1
Container 2
DOCKER
kubelet
Kube-proxy
Worker Node 2
Pod 1
Container 1
Container 2
Pod 2
Container 1
Container 2
Container 3
Pod 3
Container 1
DOCKER
kubelet
Kube-proxy

# Cluster Architecture

- Kubernetes Architecture
- ETCD For Beginners
- ETCD in Kubernetes
- Kube-API Server
- Controller Managers
- Kube Scheduler
- Kubelet
- Kube Proxy

# KUBERNETES ARCHITECTURE


kube-apiserver
Master
Manage, Plan, Schedule, Monitor Nodes
kubelet
Worker Nodes
Host Application as Containers
docker
docker
Controller-Man age r
D
CLUSTER
docker
e-scheduler
Kube-proxy
Container Runtime Engine
Run containers
rkt


kube-apiserver
Master
Manage, Plan, Schedule, Monitor Nodes
kubelet
Worker Nodes
Host Application as Containers
Controller Manager
ETCD CLUSTER
kube-scheduler
Kube-proxy
Container Runtime Engine
docker
rkt

# Kubernetes Architecture

**Master**
Manage, Plan, Schedule, Monitor Nodes

**Worker Nodes**
Host Application as Containers

**ETCD** CLUSTER

**kube-apiserver**

**Kube Controller Manager**

**kube-scheduler**

**kubelet**

**Kube-proxy**

**Container Runtime Engine**
Run containers

docker rkt

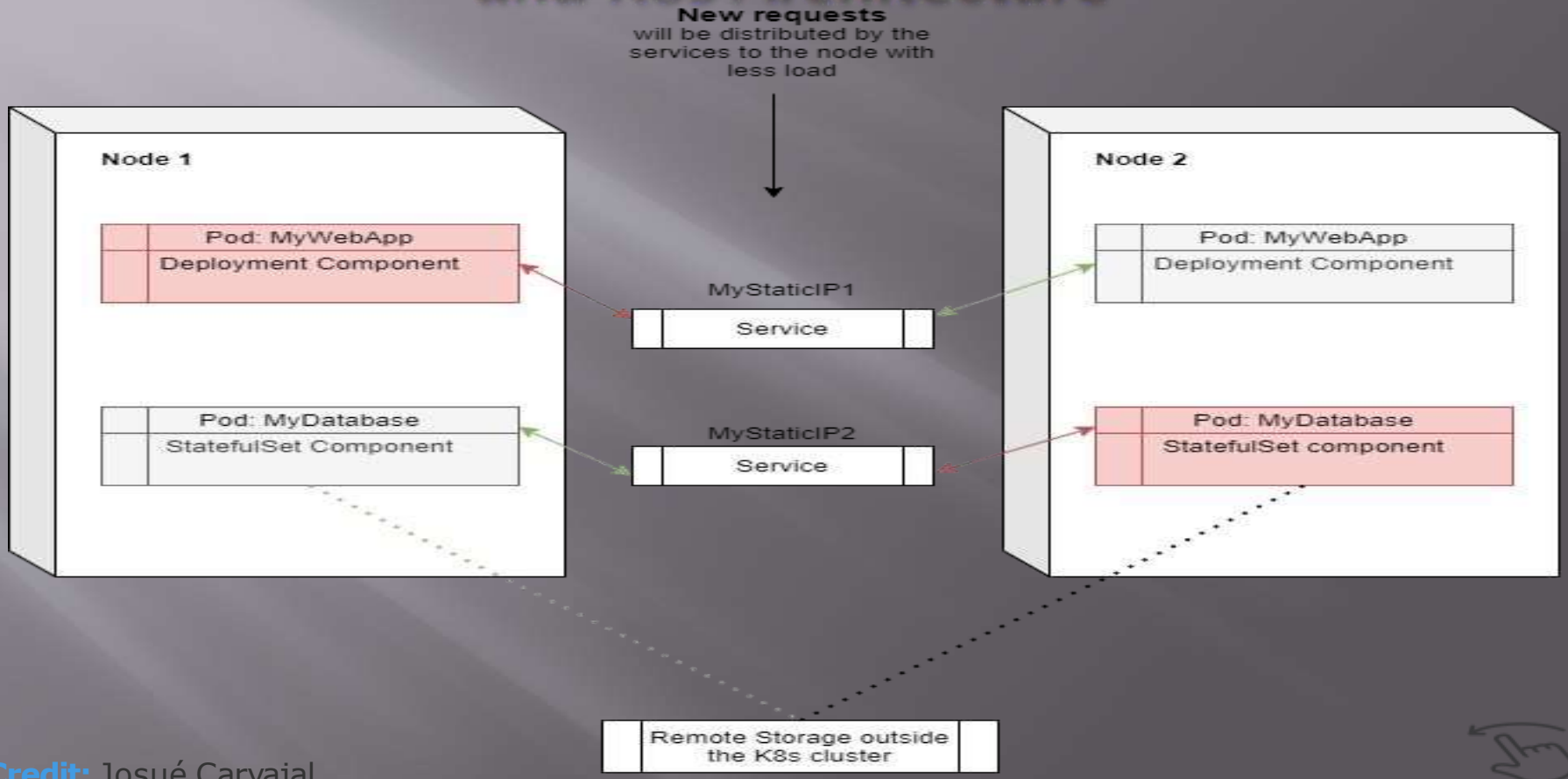
**kubelet**

**Kube-proxy**

**Container Runtime Engine**
Run containers

docker rkt

# Introduction to Kubernetes Components and K8s Architecture




Credit: Josué Carvajal

# Node

A node will contain an **N number of pods**. Usually, it is a physical server, virtual machine, or any server.

# Pod's

A Pod is the smallest unit of Kubernetes, which is an abstraction of what we knew as a container, it will run only one application on each Pod. One important thing to point out is that **each pod has its own IP address and a new IP address will be added per creation** of a pod after it dies, this is usually something that we do not want and can be solved with Kubernetes Service.

# Service

A service allows communication between pods by adding a permanent IP to the pod, this means if the pod dies **it will conserve its IP**. It also acts as a load balancer by locating the less loaded POD and send those requests to it.

# Ingress

The ingress route the traffic into the cluster, this is usually the first service that is reached, **can be used for WEB servers that interact** with the end-user.

# Config Map

It is an external configuration file to our application that we use to configure our Pods therefore all the pods can read this configuration file **without the need of recreating or rebuilding** the pods.

# Config Map — Secrets

The secret is a configuration file that is meant to store sensitive information such as passwords. It uses **base64 encoding format**.

# Volumes

Each time a node is restarted the data on it is deleted, to avoid this Kubernetes has the Volumes component, **which is meant to get persistence**.

# Pod Blueprint

Defines the pod and its replicating mechanisms.

# Deployment

This is how we deploy new pods, it is meant for stateless applications (**applications that do not store data**).

# Stateful

- This is meant for stateful applications **such as databases**, this must be used in this way since it avoids data inconsistency when two replicated nodes sharing a database pod are accessing the same database, how do Kubernetes can ensure there is data consistency, well by using stateful component and making sure who is using the storage at a given point of time.

# ETCD

## FOR BEGINNERS

# Objectives

- What is ETCD?
- What is a Key-Value Store?
- How to get started quickly?
- How to operate ETCD?
- What is a distributed system?
- How ETCD Operates
- RAFT Protocol
- Best practices on number of nodes

ETCD is a distributed reliable key-value store that is Simple, Secure & Fast

# key-value store

Tabular/Relational Databases

| Name | Age | Location | Salary | Grade |
|---|---|---|---|---|
| John Doe | 45 | New York | 5000 | |
| Dave Smith | 34 | New York | 4000 | |
| Aryan Kumar | 10 | New York | | A |
| Lauren Rob | 13 | Bangalore | | C |
| Lily Oliver | 15 | Bangalore | | B |

# key-value store

| Key | Value |
|---|---|
| Name | John Doe |
| Age | 45 |
| Location | New York |
| Salary | 5000 |

| Key | Value |
|---|---|
| Name | Dave Smith |
| Age | 34 |
| Location | New York |
| Salary | 4000 |
| Organization | ACME |

| Key | Value |
|---|---|
| Name | Aryan Kumar |
| Age | 10 |
| Location | New York |
| Grade | A |

| Key | Value |
|---|---|
| Name | Lauren Rob |
| Age | 13 |
| Location | Bangalore |
| Grade | C |

| Key | Value |
|---|---|
| Name | Lily Oliver |
| Age | 15 |
| Location | Bangalore |
| Grade | B |

# key-value store

```
{
  "name": "John Doe",
  "age": 45,
  "location": "New York",
  "salary": 5000
}
```

```
{
  "name": "Dave Smith",
  "age": 34,
  "location": "New York",
  "salary": 4000,
  "organization": "ACME"
}
```

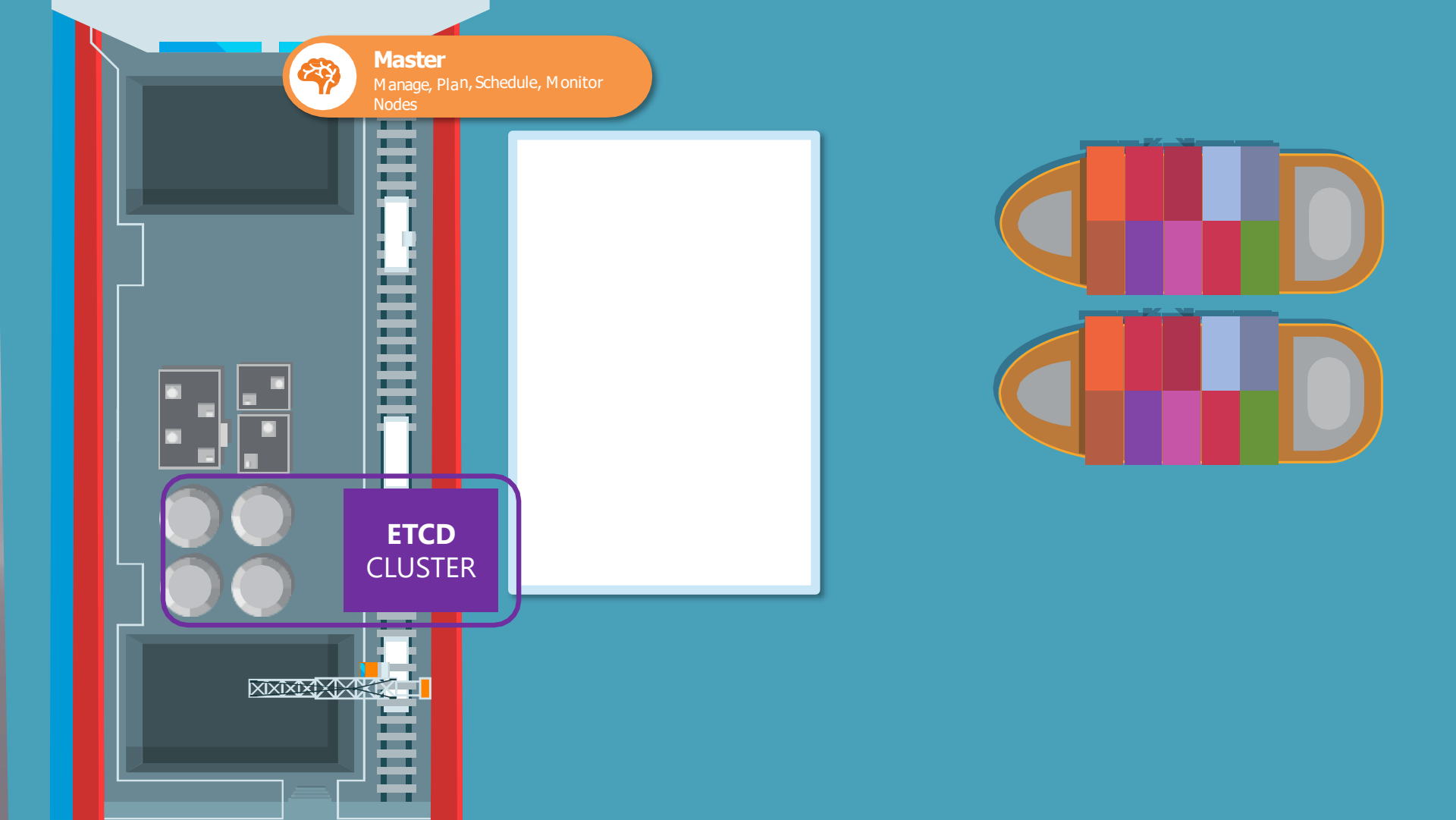
```
{
  "name": "Aryan Kumar",
  "age": 10,
  "location": "New York",
  "Grade": "A"
}
```

```
{
  "name": "Lily Oliver",
  "age": 15,
  "location": "Bangalore",
  "Grade": "B"
}
```

```
{
  "name": "Lauren Rob",
  "age": 13,
  "location": "Bangalore",
  "Grade": "C"
}
```

# ETCD

## In Kubernetes


Master
Manage, Plan, Schedule, Monitor Nodes
ETCD
CLUSTER

# Explore ETCD

```
kubectl exec etcd-master -n kube-system etcdctl get / --prefix -keys-only
/registry/apiregistration.k8s.io/apiservices/v1.
/registry/apiregistration.k8s.io/apiservices/v1.apps
/registry/apiregistration.k8s.io/apiservices/v1.authentication.k8s.io
/registry/apiregistration.k8s.io/apiservices/v1.authorization.k8s.io
/registry/apiregistration.k8s.io/apiservices/v1.autoscaling
/registry/apiregistration.k8s.io/apiservices/v1.batch
/registry/apiregistration.k8s.io/apiservices/v1.networking.k8s.io
/registry/apiregistration.k8s.io/apiservices/v1.rbac.authorization.k8s.io
/registry/apiregistration.k8s.io/apiservices/v1.storage.k8s.io
/registry/apiregistration.k8s.io/apiservices/v1beta1.admissionregistration.k8s.io
```

Registry

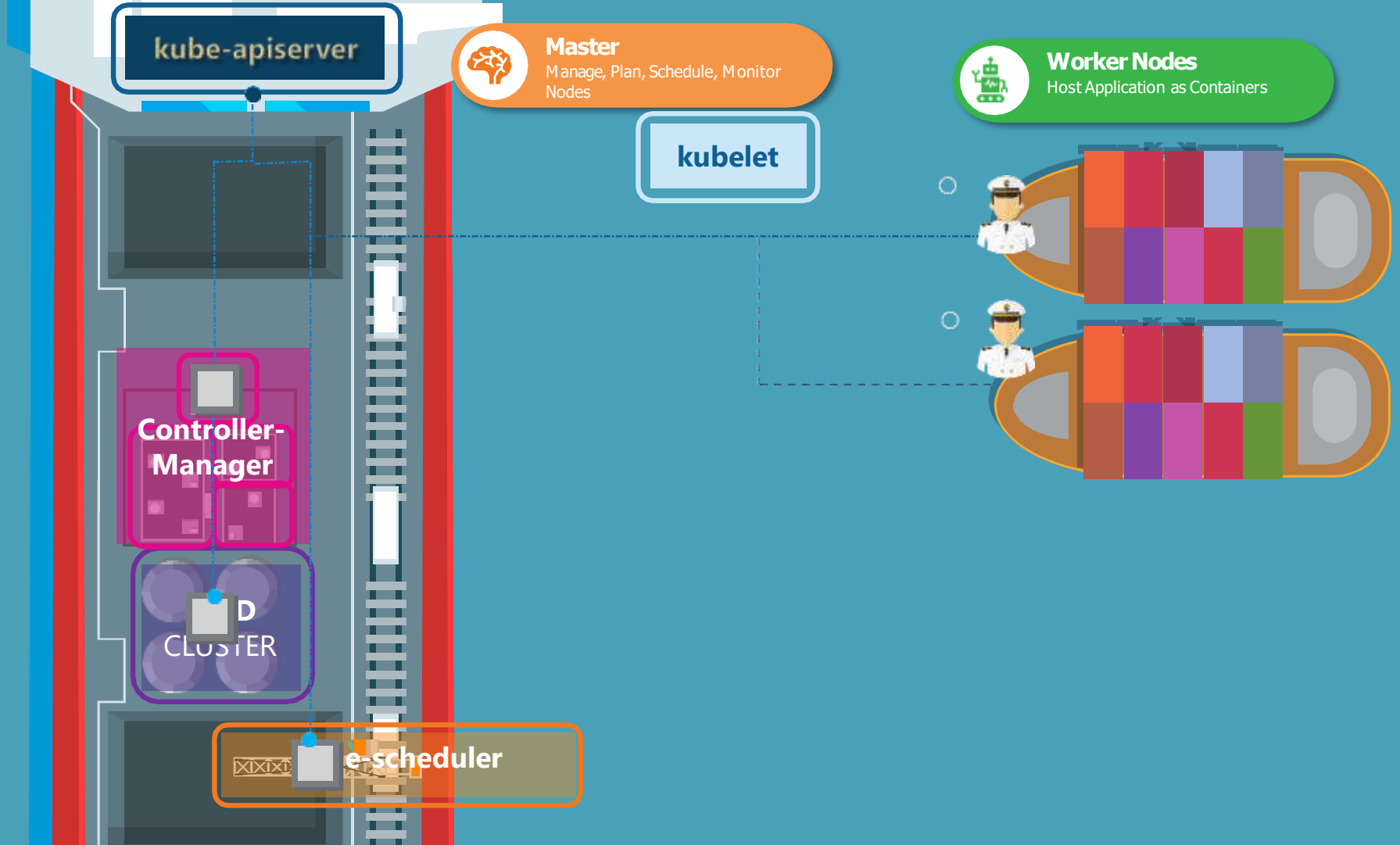
- minions
- pods
- replicasets
- deployments
- roles
- secrets

# ETCD in HA Environment


etcd.service

```
ExecStart=/usr/local/bin/etcd \\
  --name ${ETCD_NAME} \\
  --cert-file=/etc/etcd/kubernetes.pem \\
  --key-file=/etc/etcd/kubernetes-key.pem \\
  --peer-cert-file=/etc/etcd/kubernetes.pem \\
  --peer-key-file=/etc/etcd/kubernetes-key.pem \\
  --trusted-ca-file=/etc/etcd/ca.pem \\
  --peer-trusted-ca-file=/etc/etcd/ca.pem \\
  --peer-client-cert-auth \\
  --client-cert-auth \\
  --initial-advertise-peer-urls https://${INTERNAL_IP}:2380 \\
  --listen-peer-urls https://${INTERNAL_IP}:2380 \\
  --listen-client-urls https://${INTERNAL_IP}:2379,https://127.0.0.1:2379 \\
  --advertise-client-urls https://${INTERNAL_IP}:2379 \\
  --initial-cluster-token etcd-cluster-0 \\
  --initial-cluster controller-0=https://${CONTROLLER0_IP}:2380,controller-1=https://${CONTROLLER1_IP}:2380 \\
  --initial-cluster-state new \\
  --data-dir=/var/lib/etcd
```

# kube-api server


kube-apiserver
Master
Manage, Plan, Schedule, Monitor Nodes
kubelet
Worker Nodes
Host Application as Containers
Controller-Manager
D
CLUSTER
e-scheduler

# Kubernetes Architecture

```
kubectl get nodes

NAME      STATUS    ROLES     AGE       VERSION
master    Ready     master    20m       v1.11.3
node01    Ready     <none>    20m       v1.11.3
```

kube-apiserver

Controller-manager

ETCD CLUSTER

kube-scheduler

**Master**
Manage, Plan, Schedule, Monitor Nodes

**Worker Nodes**
Host Application as Containers

kubelet

Container Runtime Engine
Run containers

docker rkt

**Worker Nodes**
Host Application as Containers

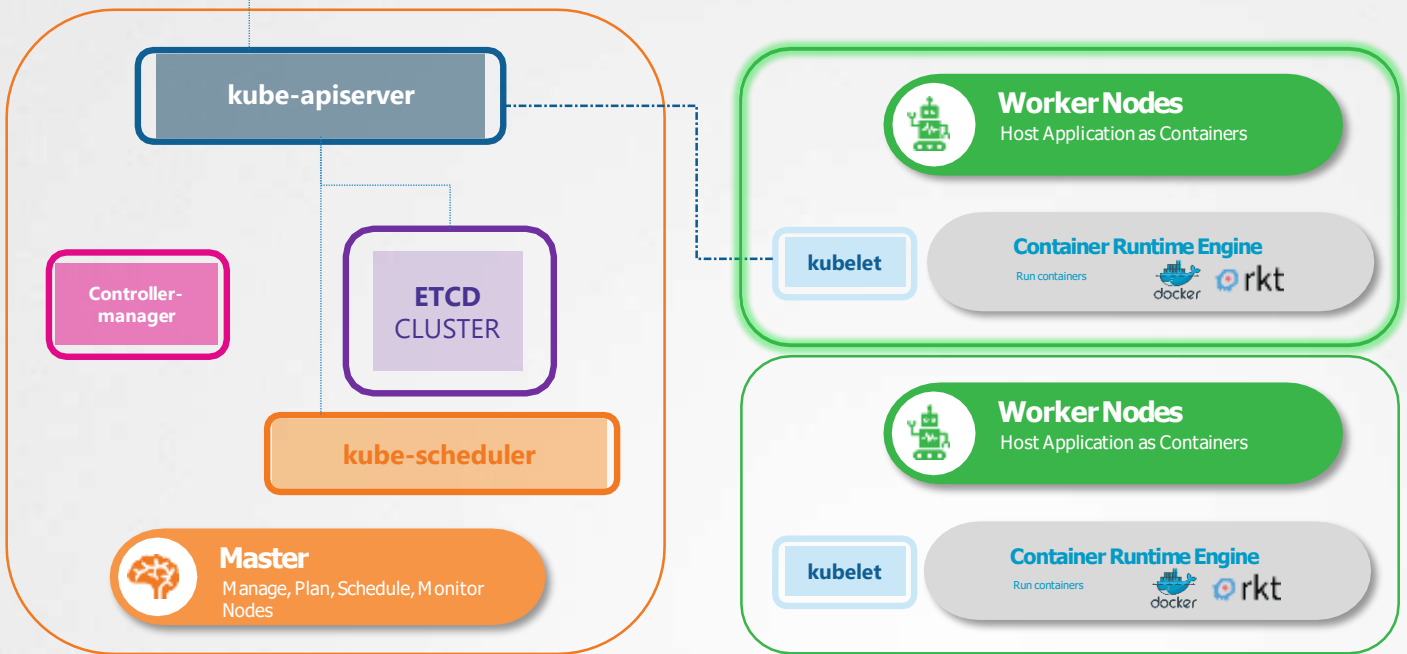
kubelet

Container Runtime Engine
Run containers

docker rkt

# Kubernetes Architecture

```
curl -X POST /api/v1/namespaces/default/pods ...[other]
Pod created!
```



# Installing kube-api server

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kube-apiserver
```

kube-apiserver.service

```
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --authorization-mode=Node,RBAC \\
  --bind-address=0.0.0.0 \\
  --enable-admission-
plugins=Initializers,NamespaceLifecycle,NodeRestriction,LimitRanger,ServiceAccount,DefaultStorageClass,Reso
urceQuota \\
  --enable-swagger-ui=true \\
  --etcd-servers=https://127.0.0.1:2379 \\
  --event-ttl=1h \\
  --experimental-encryption-provider-config=/var/lib/kubernetes/encryption-config.yaml \\
  --runtime-config=api/all \\
  --service-account-key-file=/var/lib/kubernetes/service-account.pem \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --service-node-port-range=30000-32767 \\
  --v=2
```

# View api-server - kubeadm

```
kubectl get pods -n kube-system
```

```
NAMESPACE     NAME                              READY     STATUS    RESTARTS   AGE
kube-system   coredns-78fcdf6894-hwrq9          1/1       Running   0          16m
kube-system   coredns-78fcdf6894-rzhjr          1/1       Running   0          16m
kube-system   etcd-master                       1/1       Running   0          15m
kube-system   kube-apiserver-master             1/1       Running   0          15m
kube-system   kube-controller-manager-master    1/1       Running   0          15m
kube-system   kube-proxy-lzt6f                  1/1       Running   0          16m
kube-system   kube-proxy-zm5qd                  1/1       Running   0          16m
kube-system   kube-scheduler-master             1/1       Running   0          15m
kube-system   weave-net-29z42                   2/2       Running   1          16m
kube-system   weave-net-snmdl                   2/2       Running   1          16m    -
```

# View api-server options - kubeadm

```
cat /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
spec:
  containers:
  - command:
    - kube-apiserver
    - --authorization-mode=Node,RBAC
    - --advertise-address=172.17.0.32
    - --allow-privileged=true
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --disable-admission-plugins=PersistentVolumeLabel
    - --enable-admission-plugins=NodeRestriction
    - --enable-bootstrap-token-auth=true
    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --insecure-port=0
    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
    - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
    - --requestheader-allowed-names=front-proxy-client
    - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
    - --requestheader-extra-headers-prefix=X-Remote-Extra-
    - --requestheader-group-headers=X-Remote-Group
    - --requestheader-username-headers=X-Remote-User
```

# View api-server options

```
cat /etc/systemd/system/kube-apiserver.service
```

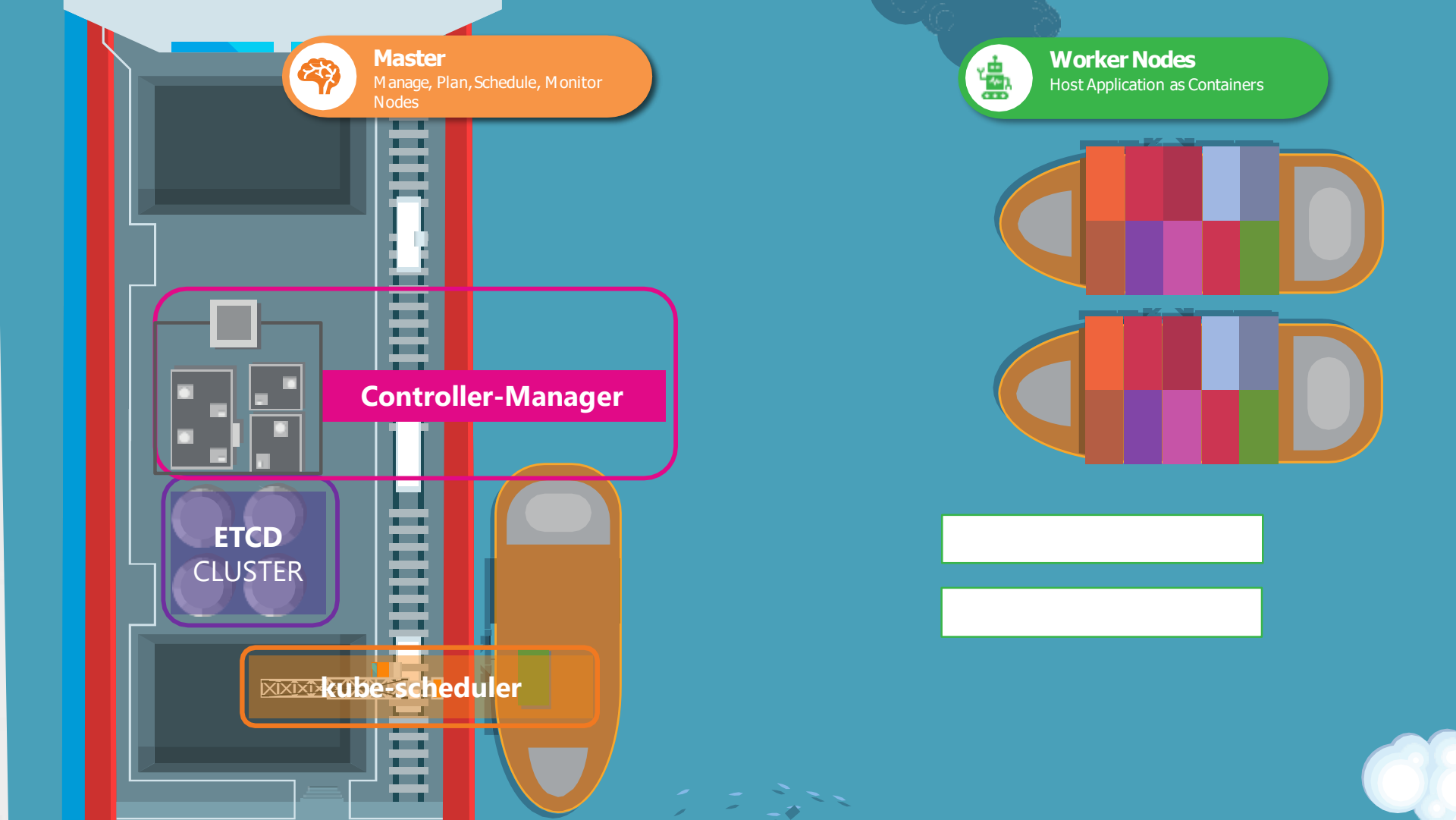
```
[Service]
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --audit-log-maxage=30 \\
  --audit-log-maxbackup=3 \\
  --audit-log-maxsize=100 \\
  --audit-log-path=/var/log/audit.log \\
  --authorization-mode=Node,RBAC \\
  --bind-address=0.0.0.0 \\
  --client-ca-file=/var/lib/kubernetes/ca.pem \\
  --enable-admission-
plugins=Initializers,NamespaceLifecycle,NodeRestriction,LimitRanger,ServiceAccount,Defa
ultStorageClass,ResourceQuota \\
  --enable-swagger-ui=true \\
  --etcd-cafile=/var/lib/kubernetes/ca.pem \\
  --etcd-certfile=/var/lib/kubernetes/kubernetes.pem \\
  --etcd-keyfile=/var/lib/kubernetes/kubernetes-key.pem \\
  --etcd-
servers=https://10.240.0.10:2379,https://10.240.0.11:2379,https://10.240.0.12:2379 \\
  --event-ttl=1h \\
  --experimental-encryption-provider-config=/var/lib/kubernetes/encryption-config.yaml
\\
  --kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \\
  --kubelet-client-certificate=/var/lib/kubernetes/kubernetes.pem \\
```

# View api-server options

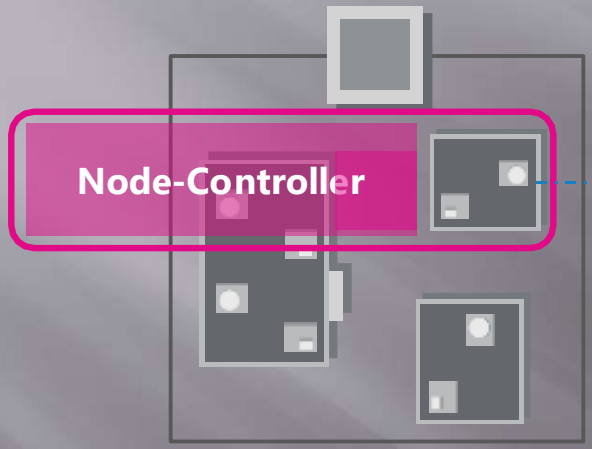
```
ps -aux | grep kube-apiserver
```

```
root      2348  3.3 15.4 399040 315604 ?       Ssl  15:46   1:22 kube-apiserver --authorization-mode=Node,RBAC --advertise-address=172.17.0.32 --allow-privileged=true --client-ca-file=/etc/kubernetes/pki/ca.crt --disable-admission-plugins=PersistentVolumeLabel --enable-admission-plugins=NodeRestriction--enable-bootstrap-token-auth=true --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key --etcd-servers=https://127.0.0.1:2379 --insecure-port=0 --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key--requestheader-allowed-names=front-proxy-client --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt --requestheader-extra-headers-prefix=X-Remote-Extra- --requestheader-group-headers=X-Remote-Group --requestheader-username-headers=X-Remote-User --secure-port=6443 --service-account-key-file=/etc/kubernetes/pki/sa.pub --service-cluster-ip-range=10.96.0.0/12 --tls-cert-file=/etc/kubernetes/pki/apiserver.crt --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
```

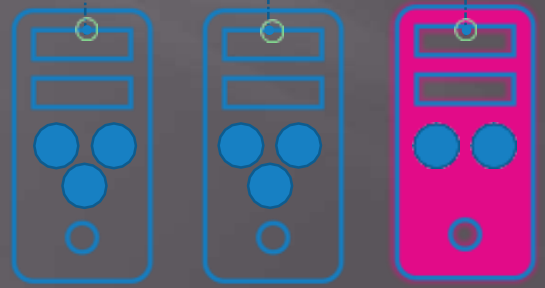
# Kube Controller Manager


Master
Manage, Plan, Schedule, Monitor Nodes
Worker Nodes
Host Application as Containers
Controller-Manager
ETCD
CLUSTER
kube-scheduler

# Controller



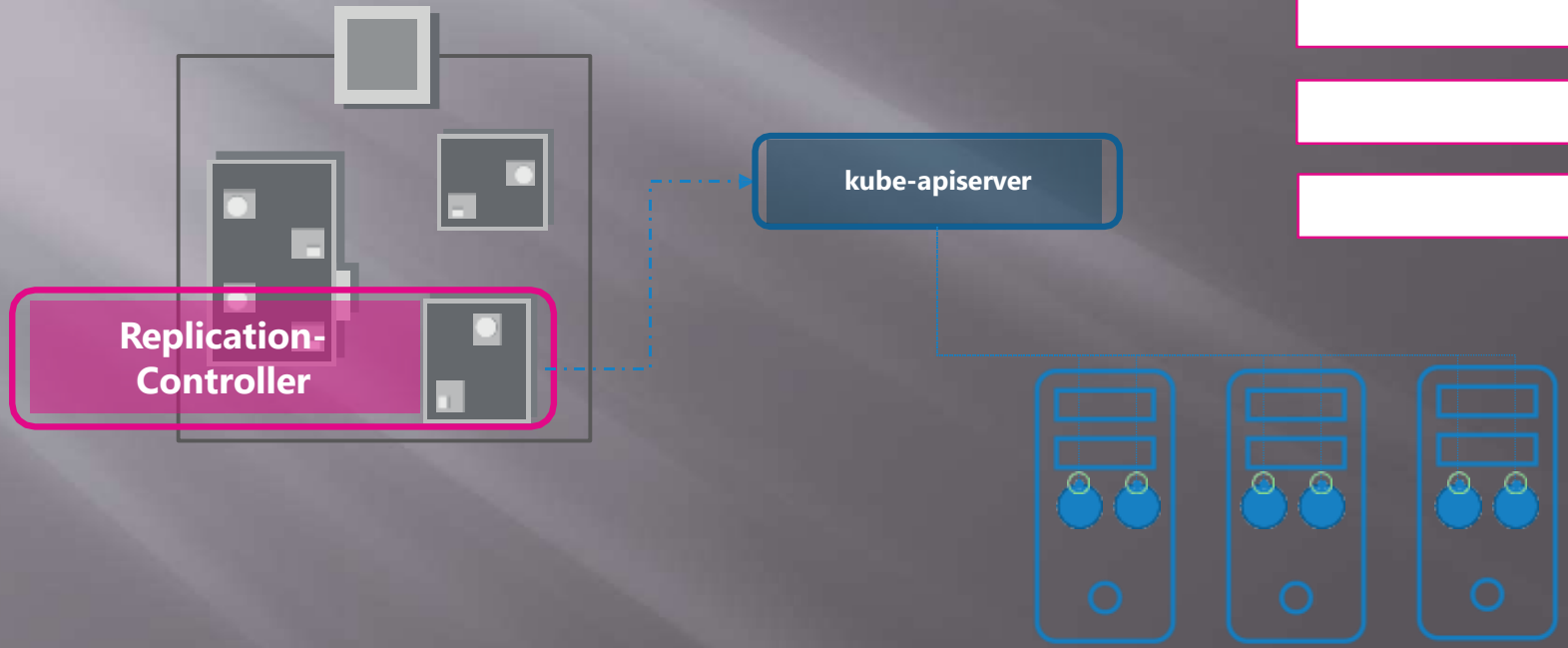

kube-apiserver



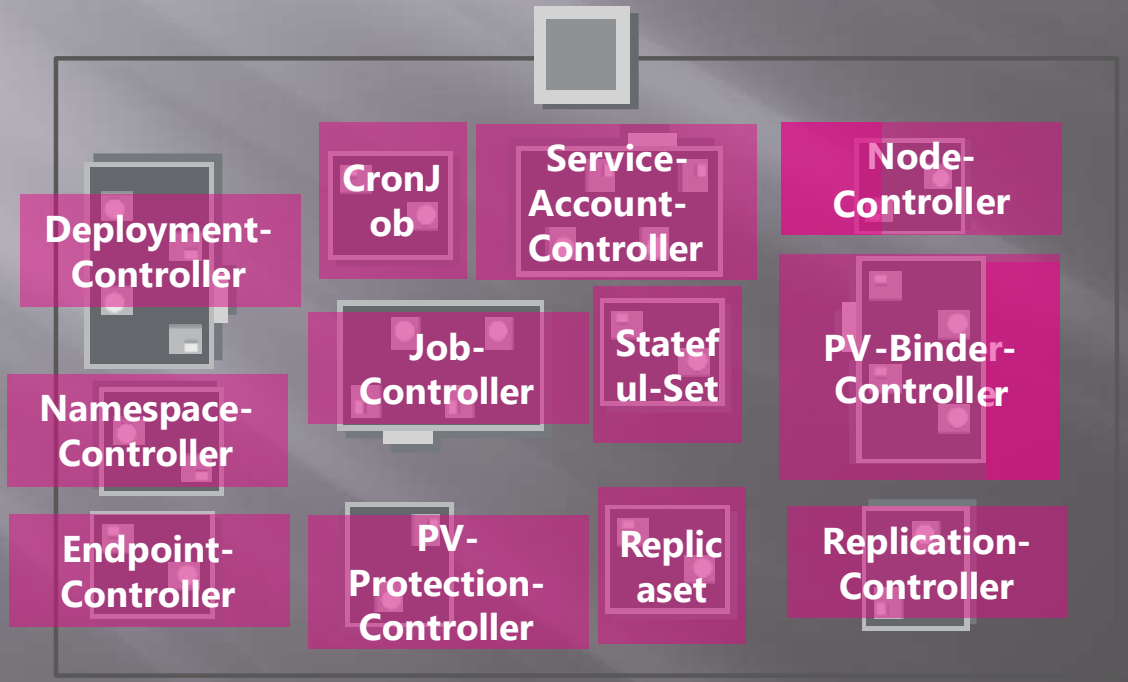
UNREACHABLE

```
kubectl get nodes
NAME       STATUS     ROLES    AGE   VERSION
worker-1   Ready      <none>   8d    v1.13.0
worker-2   NotReady   <none>   8d    v1.13.0
```

# Controller



# Controller


Deployment-Controller
CronJob
Service-Account-Controller
Node-Controller
Namespace-Controller
Job-Controller
Stateful-Set
PV-Binder-Controller
Endpoint-Controller
PV-Protection-Controller
Replicaset
Replication-Controller

# Controller

# Installing kube-controller-manager

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kube-controller-manager
```

kube-controller-manager.service

```
ExecStart=/usr/local/bin/kube-controller-manager \\
  --address=0.0.0.0 \\
  --cluster-cidr=10.200.0.0/16 \\
  --cluster-name=kubernetes \\
  --cluster-signing-cert-file=/var/lib/kubernetes/ca.pem \\
  --cluster-signing-key-file=/var/lib/kubernetes/ca-key.pem \\
  --kubeconfig=/var/lib/kubernetes/kube-controller-manager.kubeconfig \\
  --leader-elect=true \\
  --root-ca-file=/var/lib/kubernetes/ca.pem \\
  --service-account-private-key-file=/var/lib/kubernetes/service-account-key.pem \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --use-service-account-credentials=true \\
  --v=2
```

```
--node-monitor-period=5s
```

```
d=40s
```

```
s
```

```
 --controllers stringSlice     Default: [*]
 A list of controllers to enable. '*' enables all on-by-default controllers, 'foo' enables the controller
 named 'foo', '-foo' disables the controller named 'foo'.
 All controllers: attachdetach, bootstrapsigner, clusterrole-aggregation, cronjob, csrapproving,
 csrcleaner, csrsigning, daemonset, deployment, disruption, endpoint, garbagecollector,
 horizontalpodautoscaling, job, namespace, nodeipam, nodelifecycle, persistentvolume-binder,
 persistentvolume-expander, podgc, pv-protection, pvc-protection, replicaset, replicationcontroller,
```

# Installing kube-controller-manager

```
--controllers stringSlice     Default: [*]
A list of controllers to enable. '*' enables all on-by-default controllers, 'foo' enables the
controller named 'foo', '-foo' disables the controller named 'foo'.
All controllers: attachdetach, bootstrapsigner, clusterrole-aggregation, cronjob, csrapproving,
csrcleaner, csrsigning, daemonset, deployment, disruption, endpoint, garbagecollector,
horizontalpodautoscaling, job, namespace, nodeipam, nodelifecycle, persistentvolume-binder,
persistentvolume-expander, podgc, pv-protection, pvc-protection, replicaset, replicationcontroller,
resourcequota, root-ca-cert-publisher, route, service, serviceaccount, serviceaccount-token,
statefulset, tokencleaner, ttl, ttl-after-finished
Disabled-by-default controllers: bootstrapsigner, tokencleaner
```

# View kube-controller-manager - kubeadm

```
kubectl get pods -n kube-system
```

| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|---|---|---|---|---|---|
| kube-system | coredns-78fcdf6894-hwrq9 | 1/1 | Running | 0 | 16m |
| kube-system | coredns-78fcdf6894-rzhjr | 1/1 | Running | 0 | 16m |
| kube-system | etcd-master | 1/1 | Running | 0 | 15m |
| kube-system | kube-apiserver-master | 1/1 | Running | 0 | 15m |
| kube-system | kube-controller-manager-master | 1/1 | Running | 0 | 15m |
| kube-system | kube-proxy-lzt6f | 1/1 | Running | 0 | 16m |
| kube-system | kube-proxy-zm5qd | 1/1 | Running | 0 | 16m |
| kube-system | kube-scheduler-master | 1/1 | Running | 0 | 15m |
| kube-system | weave-net-29z42 | 2/2 | Running | 1 | 16m |
| kube-system | weave-net-snmdl | 2/2 | Running | 1 | 16m |


- NAMESPACE NAME READY STATUS RESTARTS AGE
- kube-system coredns-78fcdf6894-hwrq9 1/1 Running 0 16m kube-system coredns-78fcdf6894-rzhjr 1/1 Running 0 16m
  kube-system etcd-master 1/1 Running 0 15m kube-system kube-apiserver-master 1/1 Running 0 15m
  kube-system kube-controller-manager-master 1/1 Running 0 15m kube-system kube-proxy-lzt6f 1/1 Running 0 16m
- kube-system kube-proxy-zm5qd 1/1 Running 0 16m kube-system kube-scheduler-master 1/1 Running 0 15m
  kube-system weave-net-29z42 2/2 Running 1 16m
- kube-system weave-net-snmdl 2/2 Running 1 16m -

# View kube-controller-manager options - kubeadm

```
cat /etc/kubernetes/manifests/kube-controller-manager.yaml
```

```
spec:
  containers:
  - command:
    - kube-controller-manager
    - --address=127.0.0.1
    - --cluster-signing-cert-file=/etc/kubernetes/pki/ca.crt
    - --cluster-signing-key-file=/etc/kubernetes/pki/ca.key
    - --controllers=*,bootstrapsigner,tokencleaner
    - --kubeconfig=/etc/kubernetes/controller-manager.conf
    - --leader-elect=true
    - --root-ca-file=/etc/kubernetes/pki/ca.crt
    - --service-account-private-key-file=/etc/kubernetes/pki/sa.key
    - --use-service-account-credentials=true
```

# View controller-manager options

```
cat /etc/systemd/system/kube-controller-manager.service
```

```
[Service]
ExecStart=/usr/local/bin/kube-controller-manager \\
  --address=0.0.0.0 \\
  --cluster-cidr=10.200.0.0/16 \\
  --cluster-name=kubernetes \\
  --cluster-signing-cert-file=/var/lib/kubernetes/ca.pem \\
  --cluster-signing-key-file=/var/lib/kubernetes/ca-key.pem \\
  --kubeconfig=/var/lib/kubernetes/kube-controller-manager.kubeconfig \\
  --leader-elect=true \\
  --root-ca-file=/var/lib/kubernetes/ca.pem \\
  --service-account-private-key-file=/var/lib/kubernetes/service-account-key.pem \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --use-service-account-credentials=true \\
  --v=2
Restart=on-failure
RestartSec=5
```

# View controller-manager options

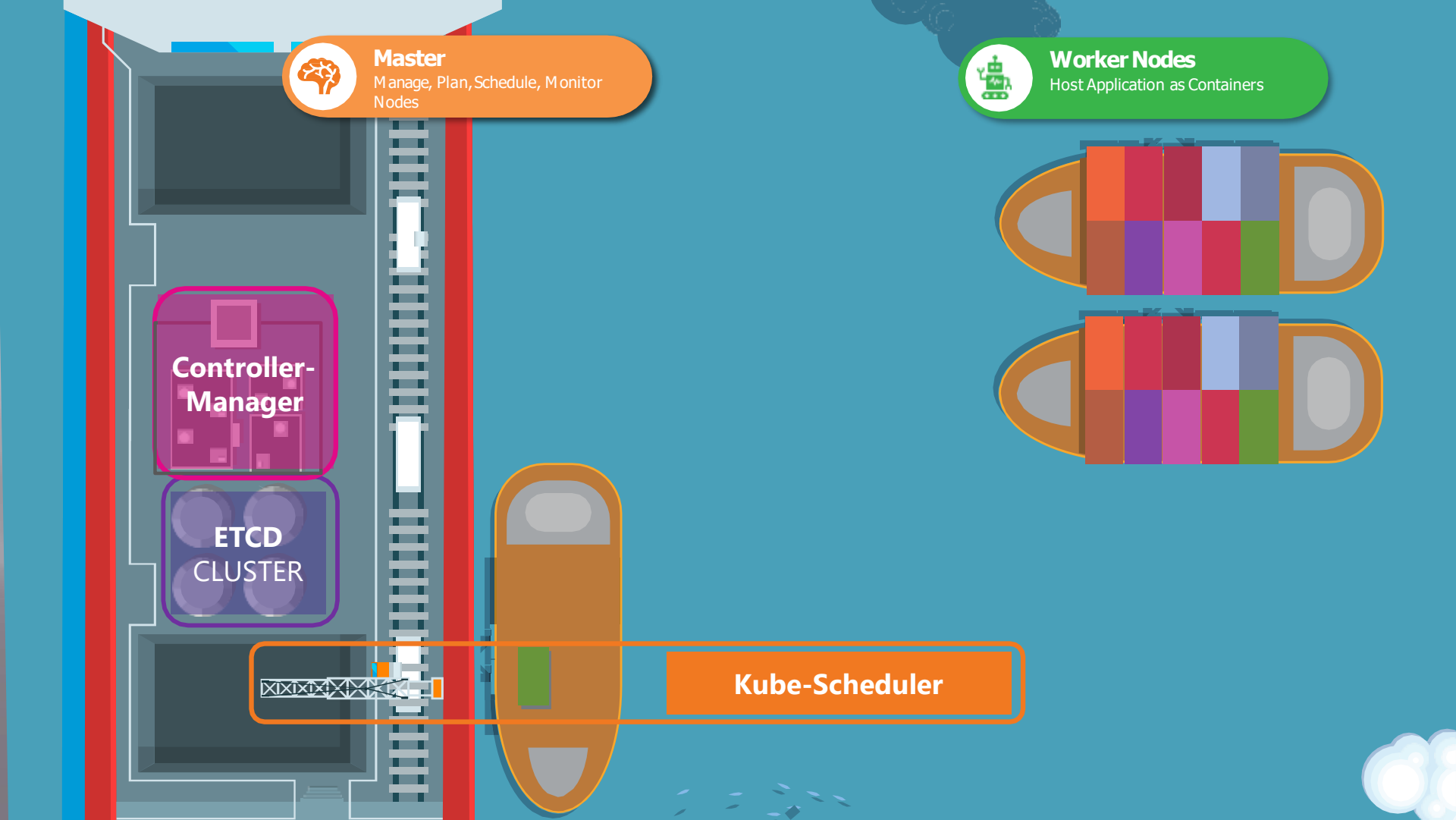
```
ps -aux | grep kube-controller-manager
```

```
root        1994   2.7   5.1 154360 105024 ?           Ssl   06:45    1:25 kube-controller-manager --
address=127.0.0.1 --cluster-signing-cert-file=/etc/kubernetes/pki/ca.crt --cluster-signing-  key-
file=/etc/kubernetes/pki/ca.key --controllers=*,bootstrapsigner,tokencleaner --
kubeconfig=/etc/kubernetes/controller-manager.conf --leader-elect=true --root-ca-
file=/etc/kubernetes/pki/ca.crt --service-account-private-key-file=/etc/kubernetes/pki/sa.key
--use-service-account-credentials=true
```

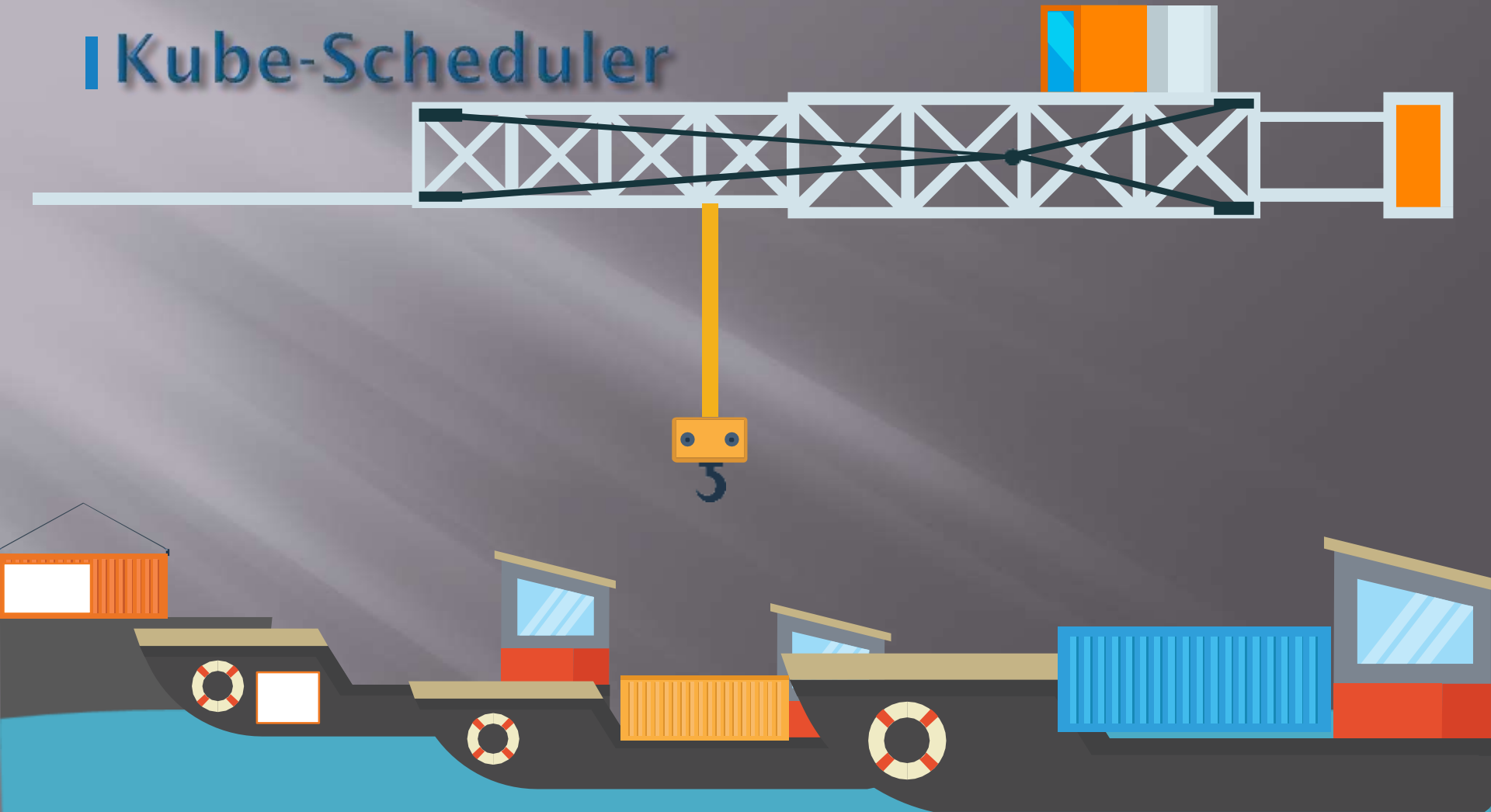
# Course Objectives

- Core Concepts
  - Cluster Architecture
  - API Primitives
  - Services & Other Network Primitives
- Scheduling
- Logging Monitoring
- Application Lifecycle Management
- Cluster Maintenance
- Security
- Storage
- Networking
- Installation, Configuration & Validation
- Troubleshooting

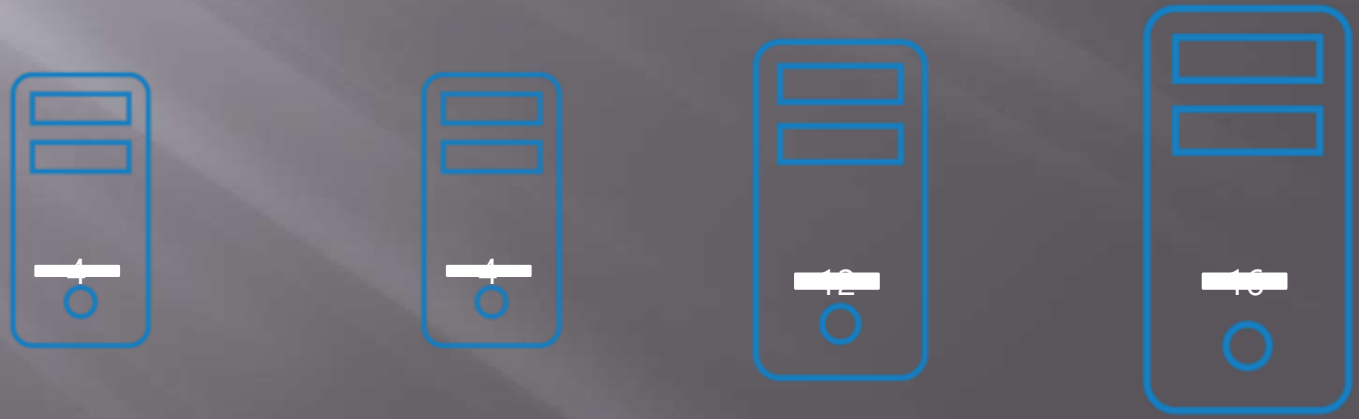
# Kube Scheduler


Master
Manage, Plan, Schedule, Monitor Nodes
Worker Nodes
Host Application as Containers
Controller-Manager
ETCD CLUSTER
Kube-Scheduler

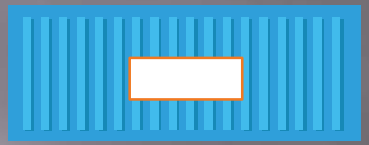
# Kube-Scheduler



# Kube-Scheduler

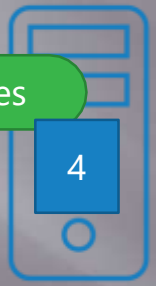


# Kube-Scheduler

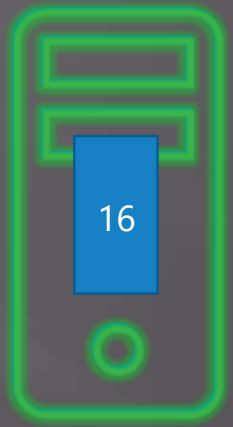


1. Filter Nodes

2. Rank Nodes





# View kube-scheduler options - kubeadm

```
cat /etc/kubernetes/manifests/kube-scheduler.yaml
```

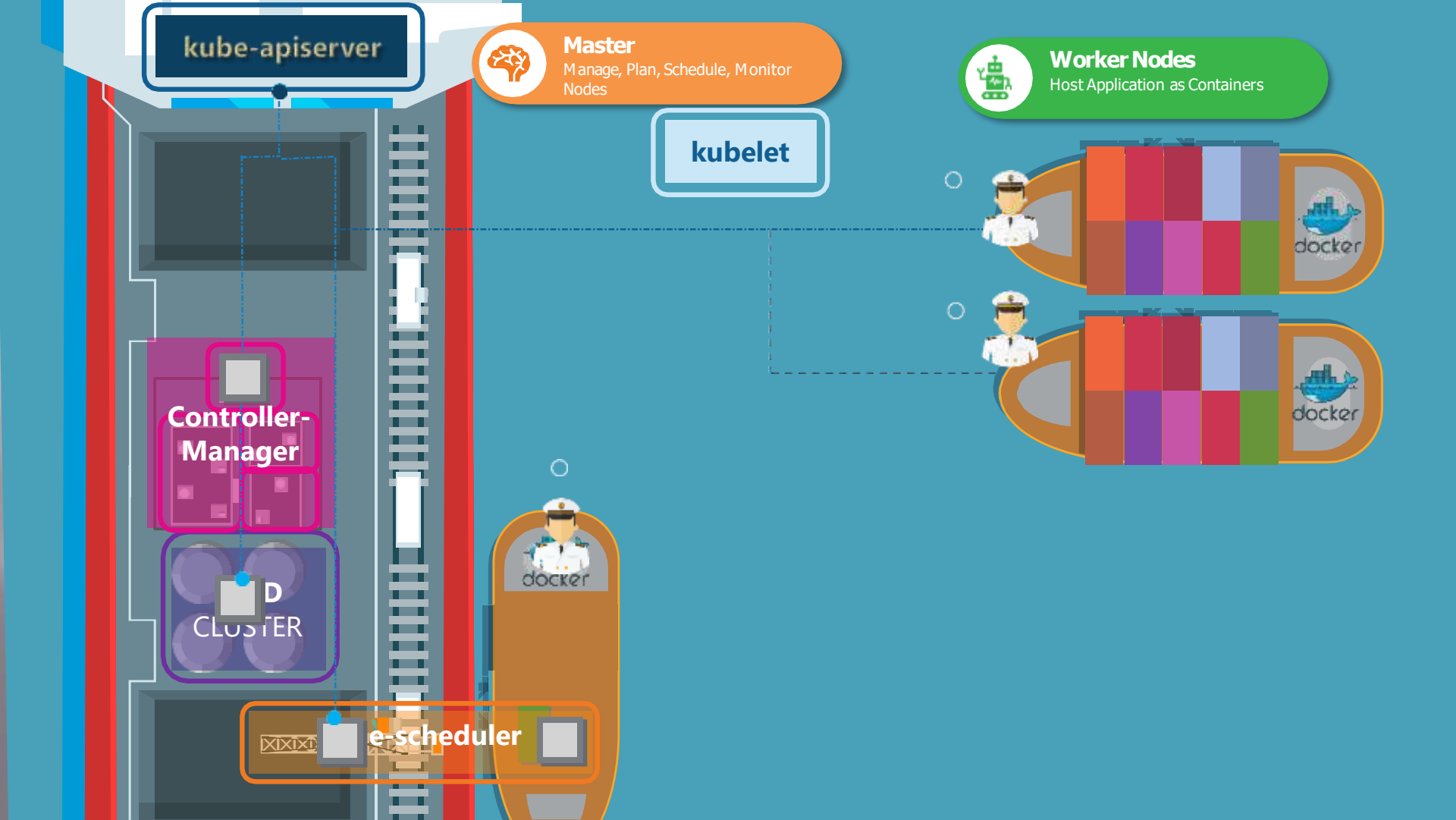
```
spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
```

# View kube-scheduler options

```
ps -aux | grep kube-scheduler
```

```
root      2477  0.8  1.6  48524 34044 ?        Ssl  17:31   0:08 kube-scheduler --
address=127.0.0.1 --kubeconfig=/etc/kubernetes/scheduler.conf --leader-elect=true
```

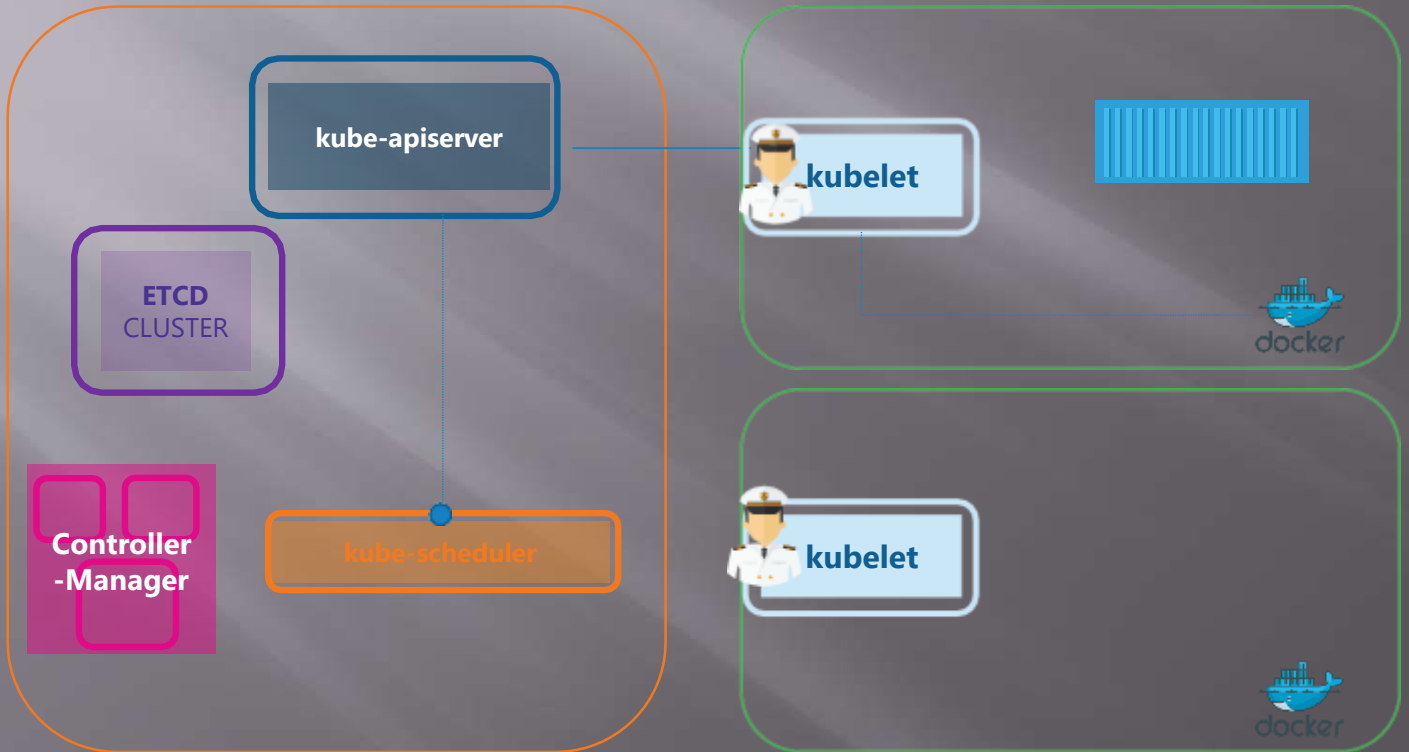
# Kubelet


kube-apiserver
Master
Manage, Plan, Schedule, Monitor Nodes
kubelet
Worker Nodes
Host Application as Containers
docker
docker
Controller-Manager
D
CLUSTER
docker
e-scheduler

# Kubernetes Architecture

**Master**
Manage, Plan, Schedule, Monitor Nodes

**Worker Nodes**
Host Application as Containers



# Installing kubelet

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kubelet
```

kubelet.service

```
ExecStart=/usr/local/bin/kubelet \\
  --config=/var/lib/kubelet/kubelet-config.yaml \\
  --container-runtime=remote \\
  --container-runtime-endpoint=unix:///var/run/containerd/containerd.sock \\
  --image-pull-progress-deadline=2m \\
  --kubeconfig=/var/lib/kubelet/kubeconfig \\
  --network-plugin=cni \\
  --register-node=true \\
  --v=2
```

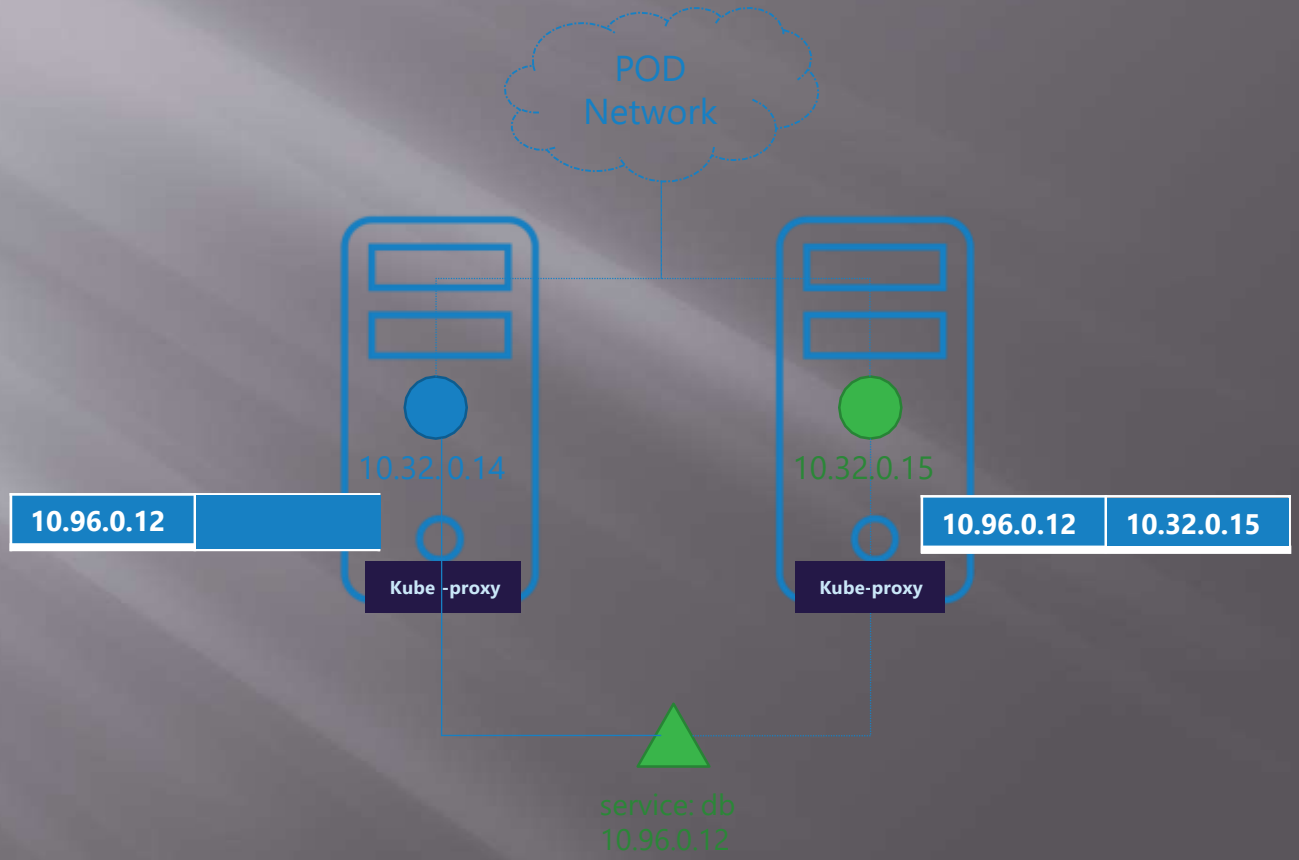
# View kubelet options

```
ps -aux | grep kubelet
```

```
root   2095   1.8   2.4   960676   98788   ?   Ssl   02:32   0:36   /usr/bin/kubelet   --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/lib/kubelet/config.yaml --cgroup-driver=cgroupfs --cni-bin-dir=/opt/cni/bin --cni-conf-dir=/etc/cni/net.d --network-plugin=cni
```

# Kube-proxy

# Kube-proxy


POD
Network
10.32.0.14
10.32.0.15
10.96.0.12
Kube -proxy
10.96.0.12
10.32.0.15
Kube-proxy
service: db
10.96.0.12

# Installing kube-proxy

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kube-proxy
```

kube-proxy.service

```
ExecStart=/usr/local/bin/kube-proxy \\
  --config=/var/lib/kube-proxy/kube-proxy-config.yaml
Restart=on-failure
RestartSec=5
```

# View kube-proxy - kubeadm

```
kubectl get pods -n kube-system
```

```
NAMESPACE     NAME                              READY     STATUS    RESTARTS   AGE
kube-system   coredns-78fcdf6894-hwrq9          1/1       Running   0          16m
kube-system   coredns-78fcdf6894-rzhjr          1/1       Running   0          16m
kube-system   etcd-master                       1/1       Running   0          15m
kube-system   kube-apiserver-master             1/1       Running   0          15m
kube-system   kube-controller-manager-master    1/1       Running   0          15m
kube-system   kube-proxy-lzt6f                  1/1       Running   0          16m
kube-system   kube-proxy-zm5qd                  1/1       Running   0          16m
kube-system   kube-scheduler-master             1/1       Running   0          15m
kube-system   weave-net-29z42                   2/2       Running   1          16m
kube-system   weave-net-snmdl                   2/2       Running   1          16m    -
```

```
kubectl get daemonset -n kube-system
```

```
NAME         DESIRED   CURRENT   READY     UP-TO-DATE   AVAILABLE   NODE SELECTOR                   AGE
kube-proxy   2         2         2         2            2           beta.kubernetes.io/arch=amd64   1h
```

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system coredns-78fcdf6894-hwrq9 1/1 Running 0 16m kube-system coredns-78fcdf6894-rzhjr 1/1 Running 0 16m
kube-system etcd-master 1/1 Running 0 15m kube-system kube-apiserver-master 1/1 Running 0 15m
kube-system kube-controller-manager-master 1/1 Running 0 15m kube-system kube-proxy-lzt6f 1/1 Running 0 16m
kube-system kube-proxy-zm5qd 1/1 Running 0 16m kube-system kube-scheduler-master 1/1 Running 0 15m
kube-system weave-net-29z42 2/2 Running 1 16m

# AKS

| | NOT CHARGED | CHARGED | OPTIONAL / ADDITIONAL |
|---|---|---|---|
| KUBERNETES CONTROL-PLANE | ✓ | | |
| BILLING SUPPORT | ✓ | | |
| BASIC LOAD BALANCER | ✓ | | |
| NODE'S VIRTUAL MACHINES | | ✓ | |
| NODE'S DISKS | | ✓ | |
| BANDWIDTH | | ✓ | |
| STORAGE FOR PERSISTENT VOLUMES | | | ✓ |
| STANDARD LOAD BALANCER | | | ✓ |
| PUBLIC IP ADDRESSES | | | ✓ |

- “There are two ways to be successful with kubernetes:
- One: you’re goon be running certified kubernets distributions everywhere.
- Two: You need to be able to manage all these clusters in a simple, consistent way.”

- -PETER SMAILS, CHIEF MARKETING OFFICER AT RANCER

# Benefits of Managed Kubernets

- Multi-master cluster, including redundant etcd servers;
- Abstracts complexity of distributed systems;
- Makes things easier to run Kubernetes at scale;
- SLA and Support;
- Managed kubernets allows you to focus on end-to-end DevOps and SRE solutions;

# Limitation

- Kubernets Version;
- Predefined Container runtime and Container Network Interface (CNI).
- Predefined OS base image;
- Control Plane limited access and customization;
- Not all features are available in all regions;
- Service Limits;

# AKS Capacity Analysis

- Regions availability
- References
- Products available by region
- https://azure.microsoft.com/en-us/global-infrastructure/services/?products=kubernetes-service
- Restricted Virtual Machine Sizes
- References
- Restricted VM sizes
- https://docs.microsoft.com/en-us/azure/aks/quotas-skus-regions#restricted-vm-sizes
- Resource Group and Tags
- References
- Can I modify tags and other properties of the AKS resources in the node resource group?
- https://docs.microsoft.com/en-us/azure/aks/faq#can-i-modify-tags-and-other-properties-of-the-aks-resources-in-the-node-resource-group
- Node Operating System
- References
- Change kernel parameters
- https://www.devops.buzz/public/aks/reboot-node#change-kernel-parameters
- Kubernetes version
- References
- Supported Kubernetes versions in Azure Kubernetes Service (AKS)
- https://docs.microsoft.com/en-us/azure/aks/supported-kubernetes-versions

# AKS Capacity Analysis

- Network
- References
- Compare network models
- https://docs.microsoft.com/en-us/azure/aks/concepts-network#compare-network-models
- Deep dive – Azure Load Balancer
- https://msandbu.org/deep-dive-azure-load-balancer/
- Resources reservation
- References
- Resource reservations
- https://docs.microsoft.com/en-us/azure/aks/concepts-clusters-workloads#resource-reservations
- Azure subscription limits
- References
- Classic deployment model limits
- https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#classic-deployment-model-limits
- Resource Group limits
- References
- Resource group limits
- https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#resource-group-limits
- Resources not limited to 800 instances per resource group
- https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/resources-without-resource-group-limit

# AKS network architecture

- References
- Network concepts for applications in Azure Kubernetes Service (AKS)
- https://docs.microsoft.com/en-us/azure/aks/concepts-network
- Create a private Azure Kubernetes Service cluster
- https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/aks/private-clusters.md
- Kubenet (basic) vs Azure CNI (advanced)
- References
- Compare network models
- https://docs.microsoft.com/en-us/azure/aks/concepts-network#compare-network-models
- Basic Load Balancer vs Standard Load Balancer
- References
- Deep dive – Azure Load Balancer
- https://msandbu.org/deep-dive-azure-load-balancer/