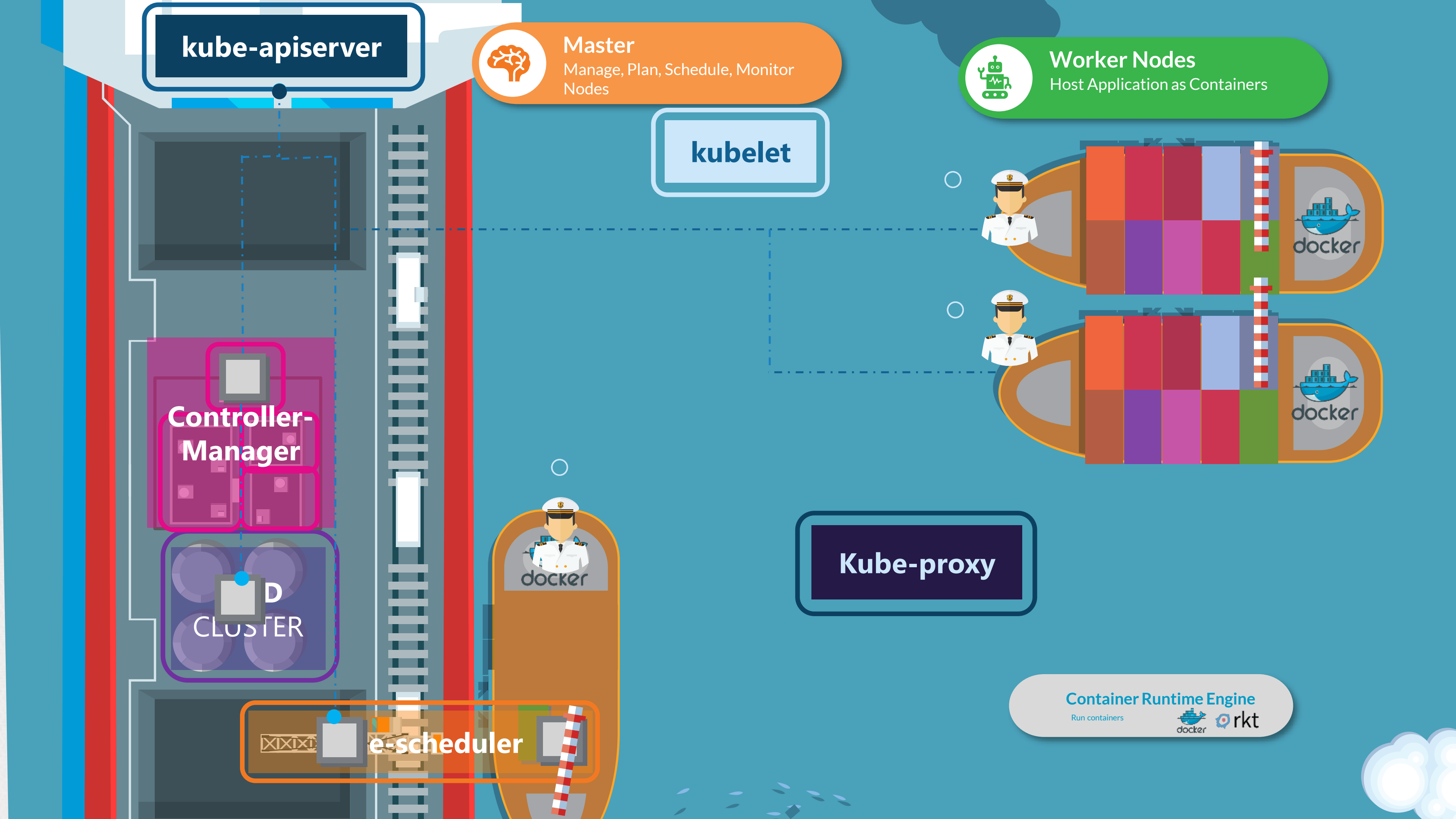
KUBERNETES ARCHITECTURE



kube-apiserver

Master
Manage, Plan, Schedule, Monitor Nodes

Worker Nodes
Host Application as Containers

kubelet

Controller-Manager

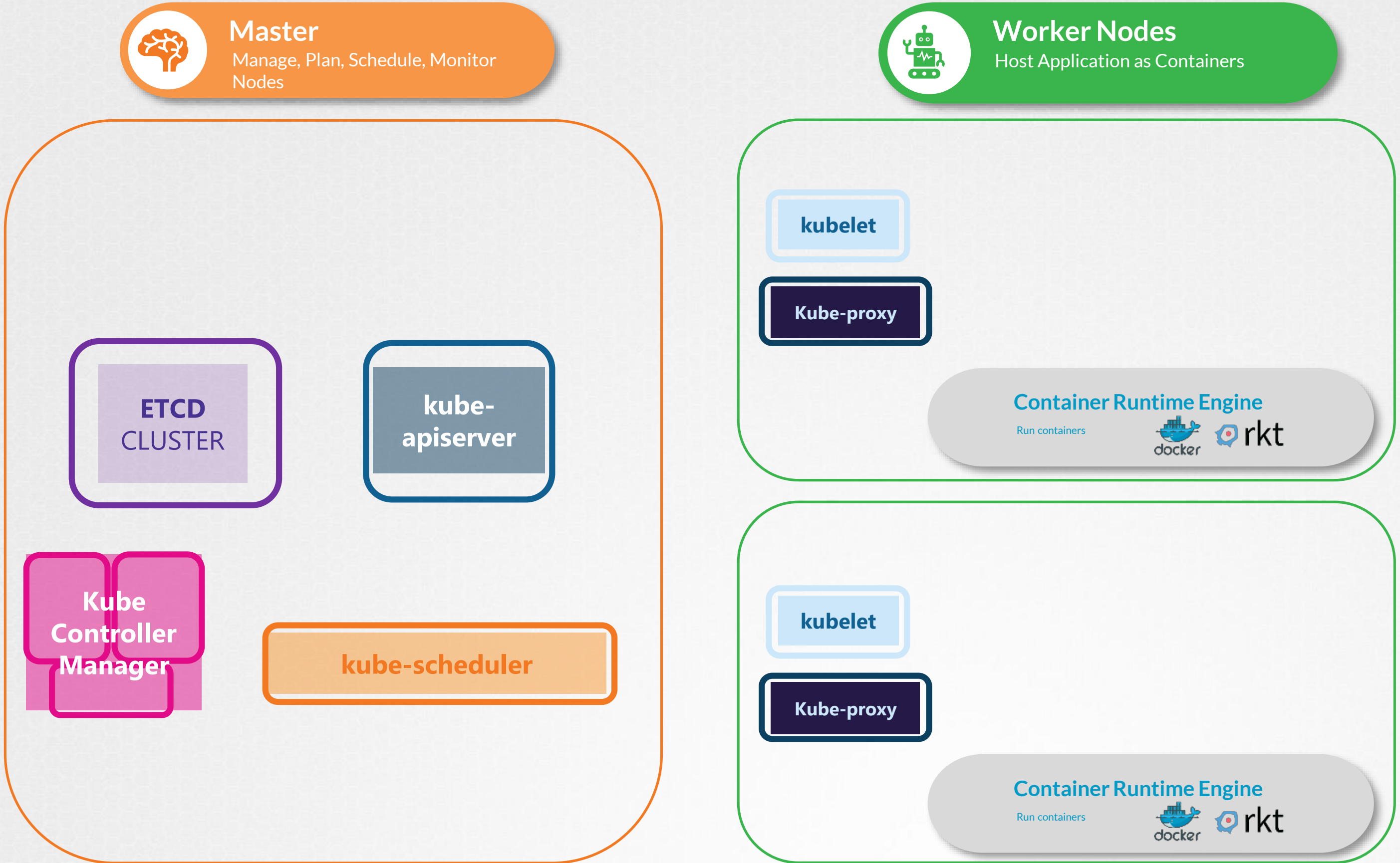
D
CLUSTER

e-scheduler

Kube-proxy

Container Runtime Engine
Run containers

Kubernetes Architecture



Master

Manage, Plan, Schedule, Monitor Nodes

Worker Nodes

Host Application as Containers

11

ETCD

FOR BEGINNERS

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

Operate ETCD

Install from binary directly

3. Run ETCD Service

```
./etcd
```

```
./etcdctl set key1 value1
```

```
./etcdctl get key1
value1
```

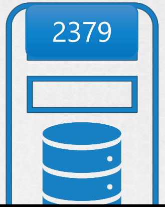
```
./etcdctl
NAME:
   etcdctl - A simple command line client for etcd.

COMMANDS:
     backup          backup an etcd directory
     cluster-health  check the health of the etcd cluster
     mk              make a new key with a given value
     mkdir           make a new directory
     rm              remove a key or a directory
     rmdir           removes the key if it is an empty directory or a key-value pair
     get             retrieve the value of a key
```

KODEKLOUD

ETCD

In Kubernetes

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
```

Important

kube-api server

Kube-api Server

1. Authenticate User
2. Validate Request
3. Retrieve data
4. Update ETCD
5. Scheduler
6. Kubelet

https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/

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
```

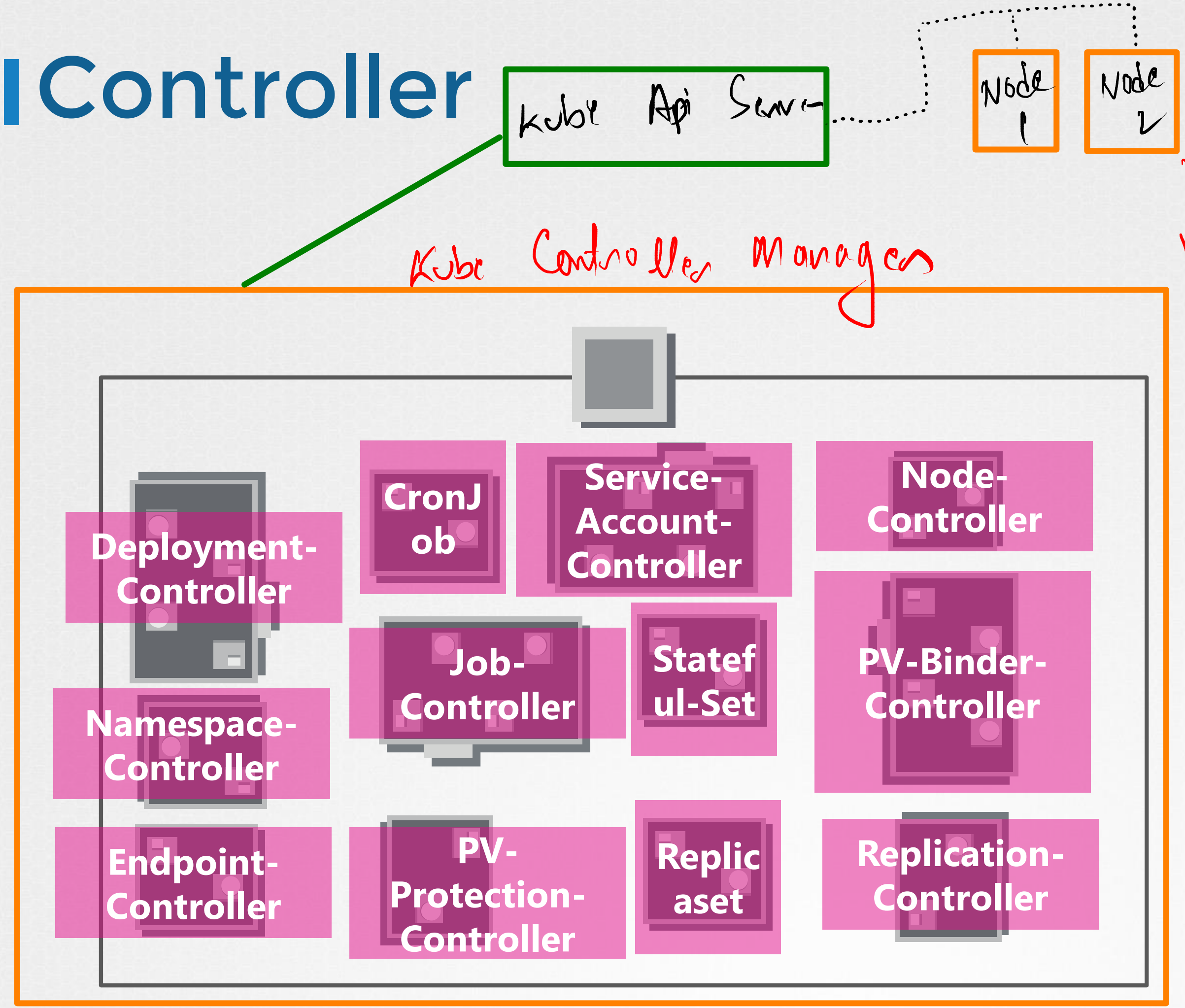
Bind

Link etcd server

Used to verify service account tokens

Kube Controller Manager

Controller



manage desired state of each object

Watch Status

Remediate Situation

After each 5 sec will check status

Node Monitor Period = 5s

Node Monitor Grace Period = 40s

POD Eviction Timeout = 5m

Controller waits for 40s for a response before marking it unhealthy

Time to wait for graceful termination

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
```

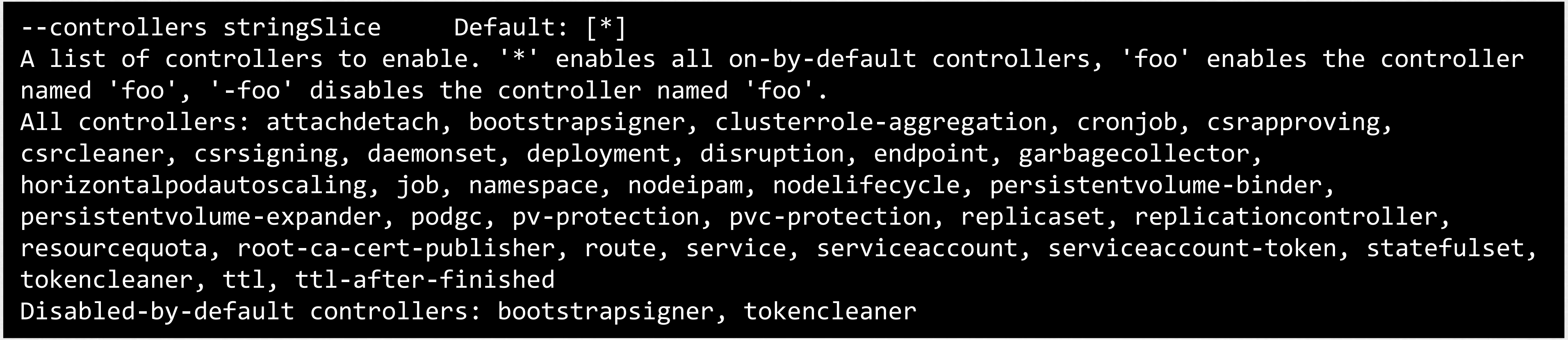
```
N --node-monitor-period=5s
```

```
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



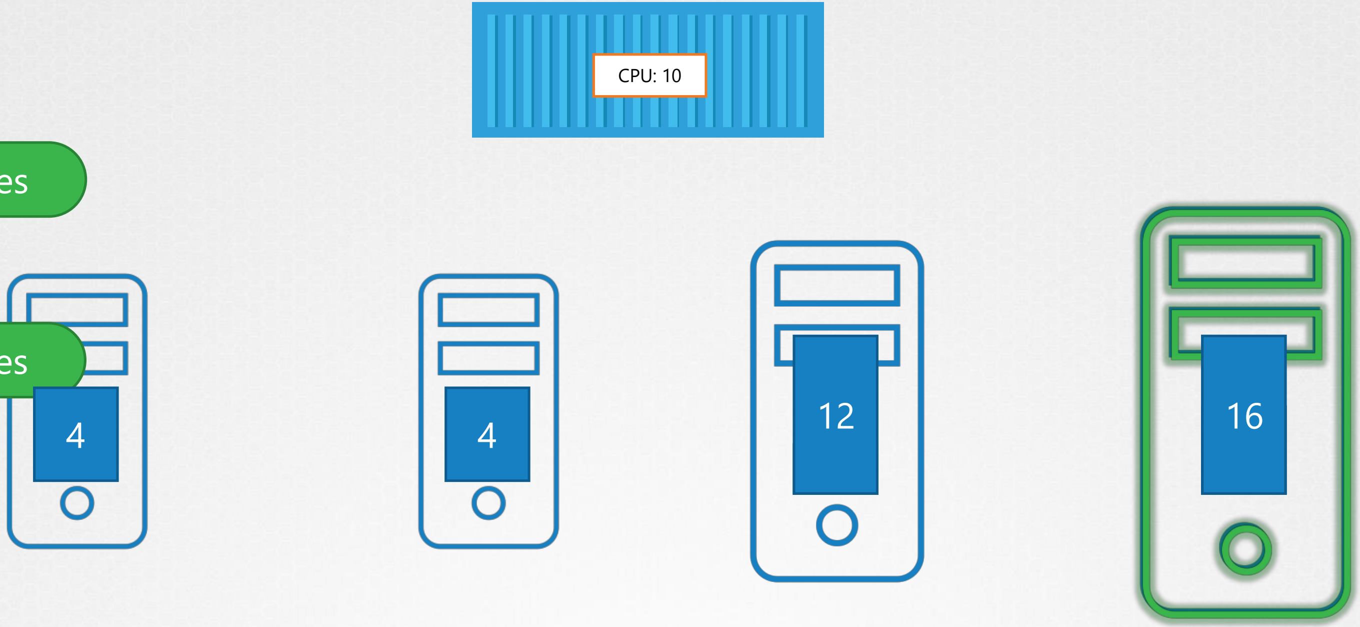
```
--controllers stringSlice     Default: [*]
A list of controllers to enable. '*' enables all on-by-default controllers, 'foo' enables the controller
named 'foo', '-foo' disables the controller named 'foo'.
All controllers: attachdetach, bootstrapsigner, clusterrole-aggregation, cronjob, csrapproving,
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
```

Kube Scheduler

Kube-Scheduler

1. Filter Nodes

2. Rank Nodes



More Later…

- Resource Requirements and Limits ✓
- Taints and Tolerations ✓
- Node Selectors/Affinity ✓

Installing kube-scheduler

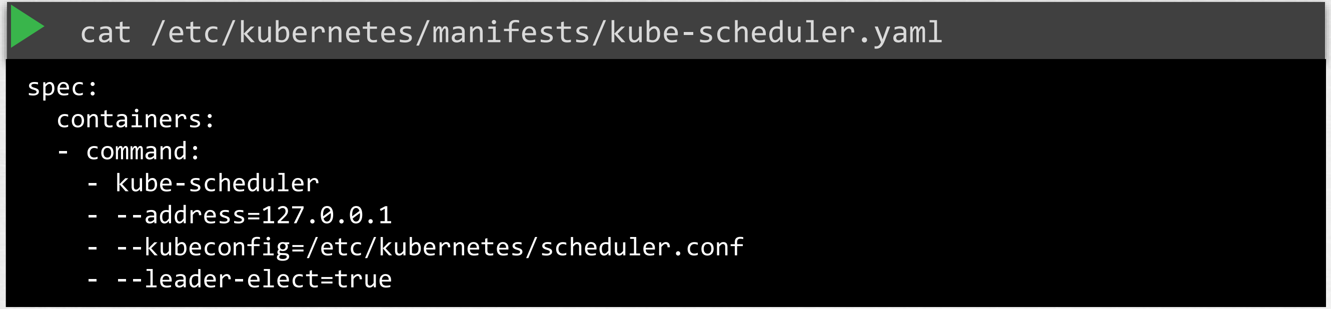
```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kube-scheduler
```

kube-scheduler.service

```
ExecStart=/usr/local/bin/kube-scheduler \\
  --config=/etc/kubernetes/config/kube-scheduler.yaml \\
  --v=2
```

View kube-scheduler options - kubeadm

1

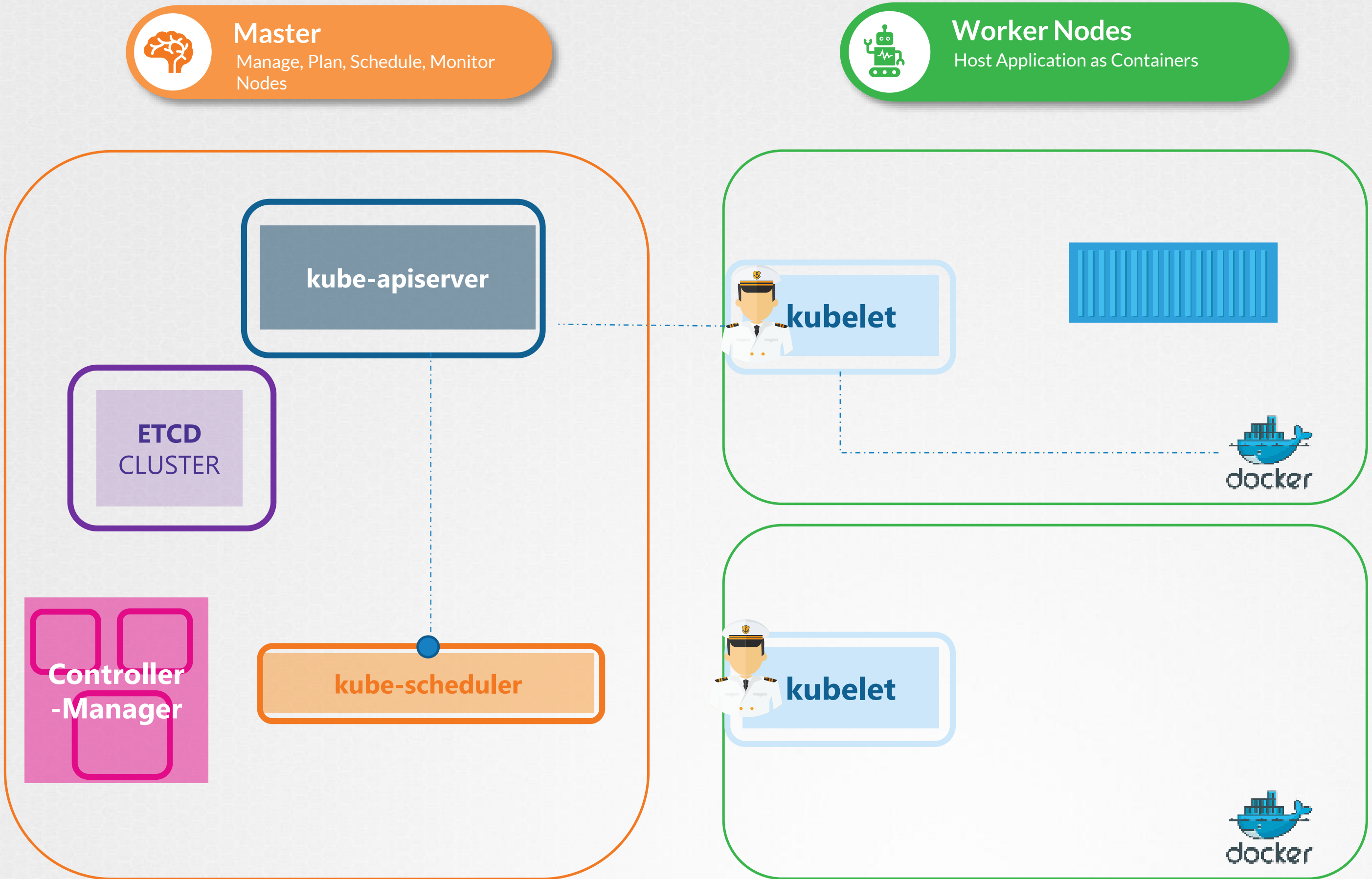


```
▶ cat /etc/kubernetes/manifests/kube-scheduler.yaml
```

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

Kubelet

Kubernetes Architecture



Master

Manage, Plan, Schedule, Monitor Nodes

Worker Nodes

Host Application as Containers

kube-apiserver

ETCD CLUSTER

Controller -Manager

kube-scheduler

kubelet

kubelet

docker

docker

Register Node

Create PODs

Monitor Node & PODs

it also updates status of pods via REST api of kube-apiserver

Installing kubelet

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

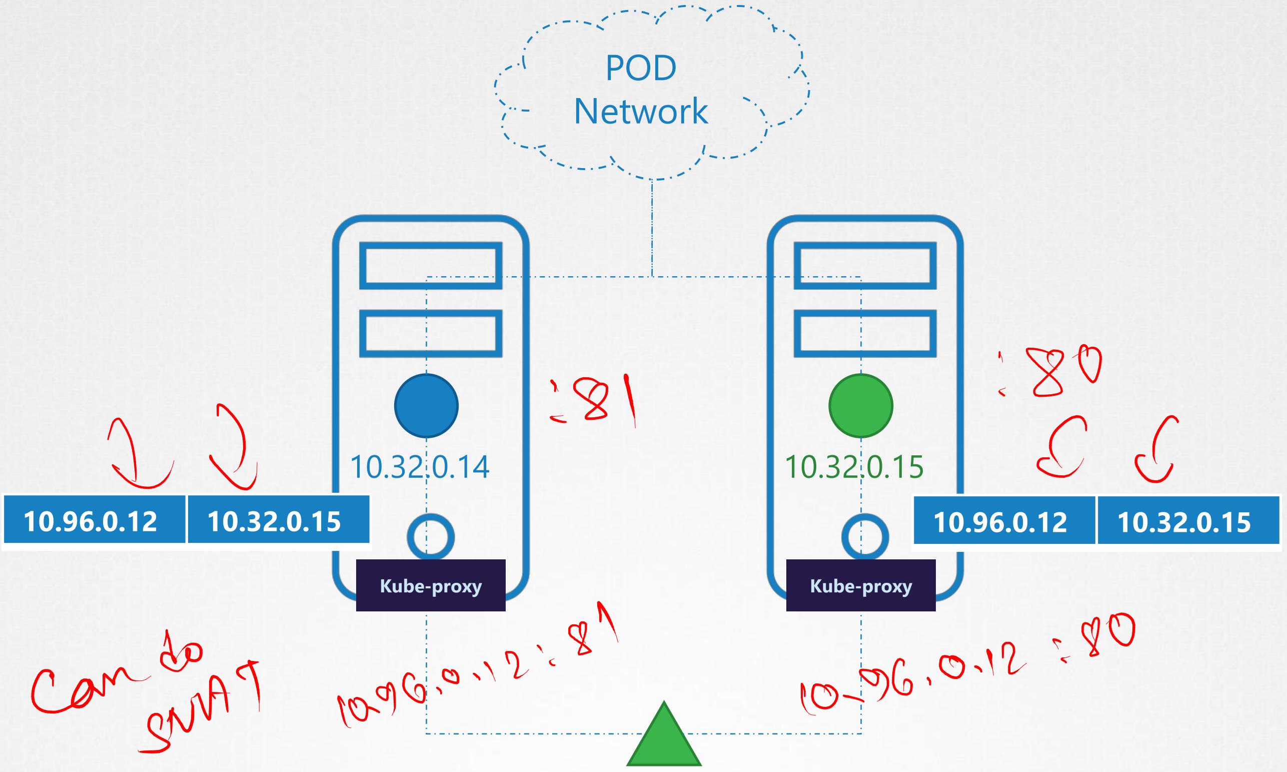
kubelet.service

```
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
```

Kubeadm does not deploy Kubelets

Kube-proxy

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
```

Three ways to see options

① Check in the /etc/kubernetes/manifests for YAML - for daemonsets

② Check in service folder.

/etc/systemd/system/ . . . . .

③ Check process

ps -aux | grep <name>