

# Demystifying the Nuts & Bolts of Kubernetes Architecture



# Who Am I?

- Principal Systems Development Engineer at DellEMC
- Worked with VMware & CGI
- Docker Captain | Docker Community Leader
- Collabnix Slack – 1800+ Members
- DockerLabs – 500+ tutorials



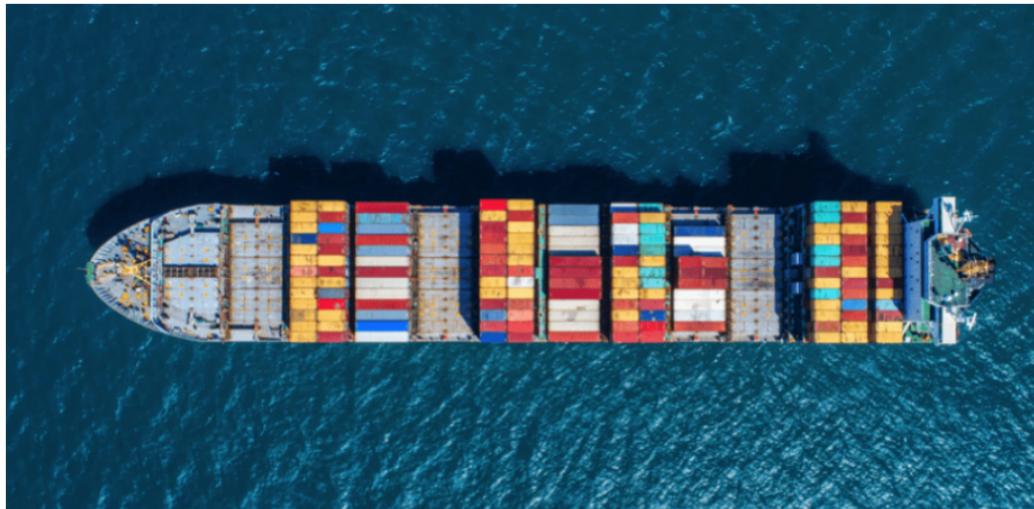
<http://www.collabnix.com>

# Let's start with an analogy..



# A Cargo Ship...

Carries containers across the sea

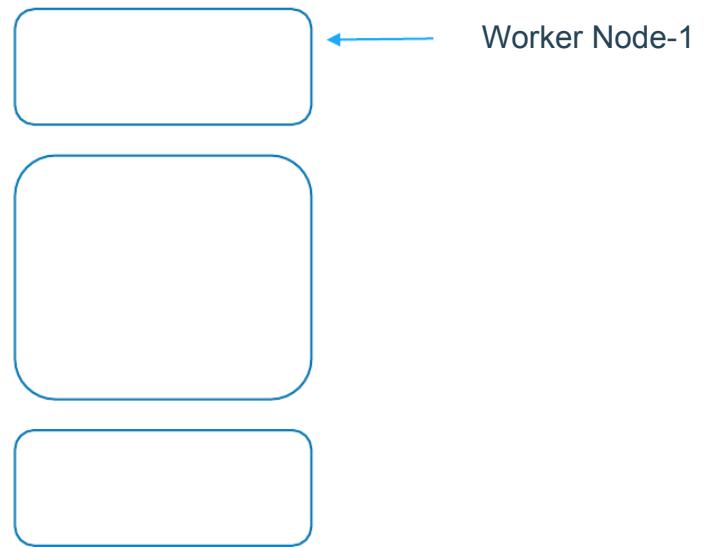


# A Cargo Ship...

Host Application as Containers ~ Worker Nodes



# Overview



# Control Ships..

Managing & Monitoring of the cargo ships

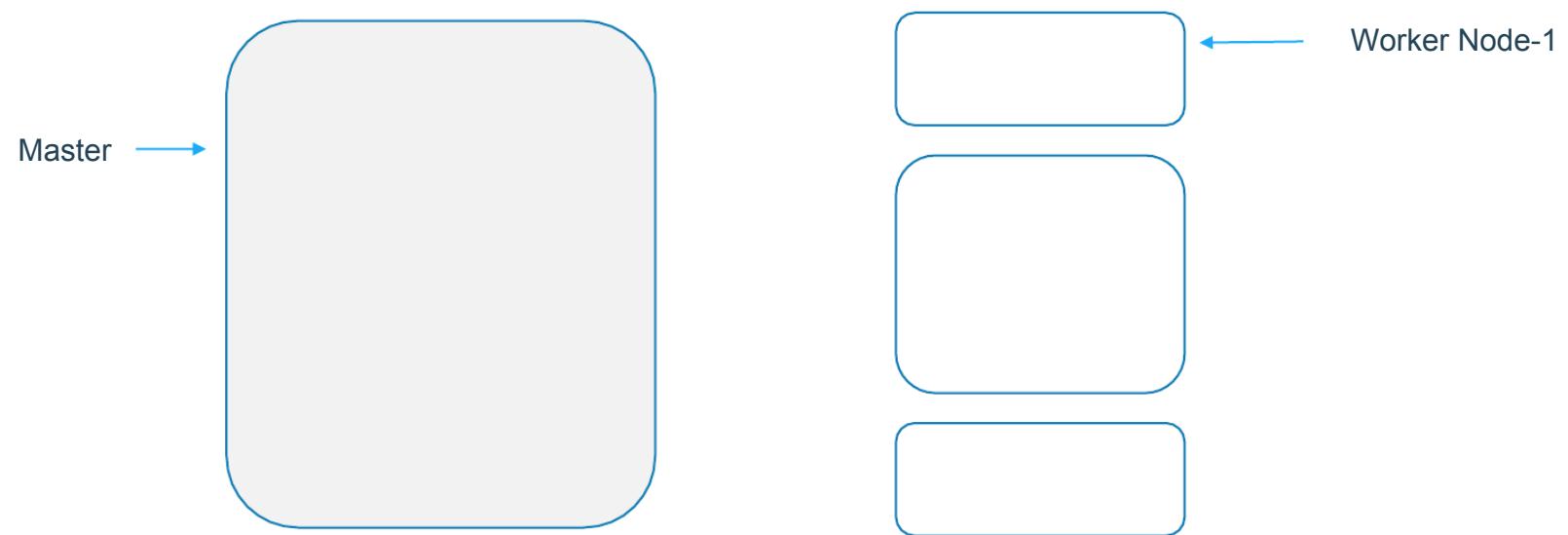


# Control Ships..

Manage, Plan, Schedule, Monitor ~ Master



# Overview



# Let's talk about Master Components..



# Ship Cranes

Identifies the placement of containers

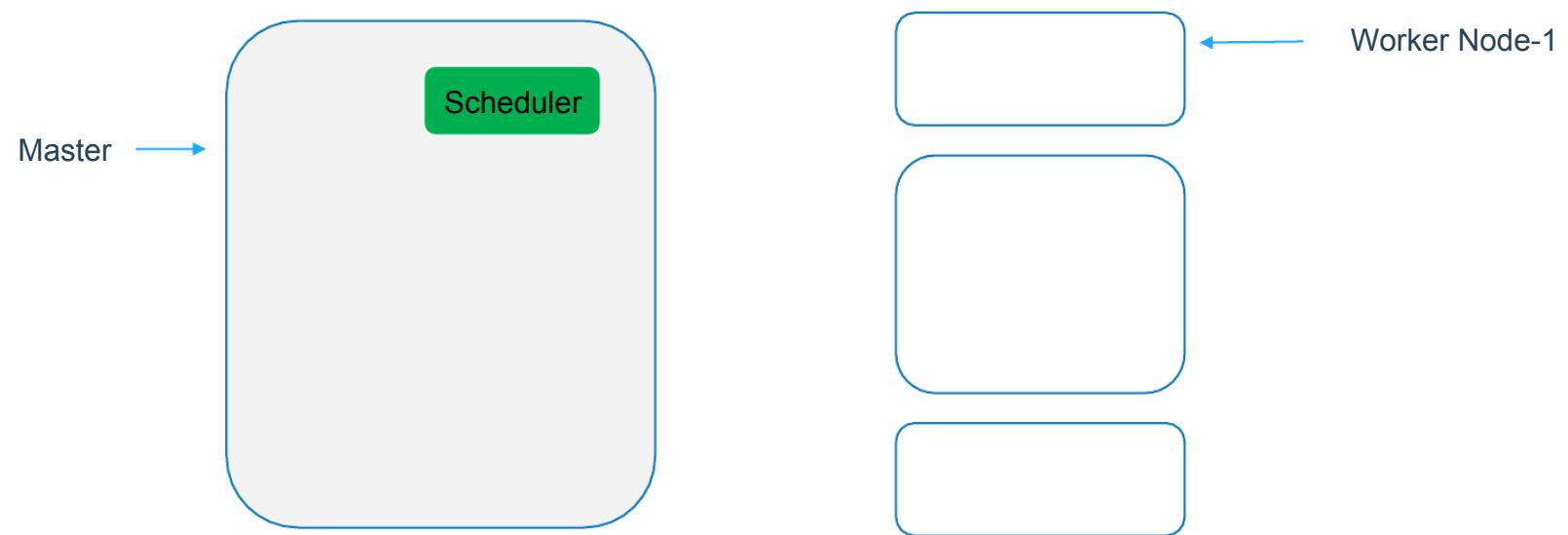


# Ship Cranes

Identifies the right node to place a containers ~ **Kube-Scheduler**



# Overview



# Cargo Ship Profiles

HA database ~ Which containers on which ships? When was it loaded?

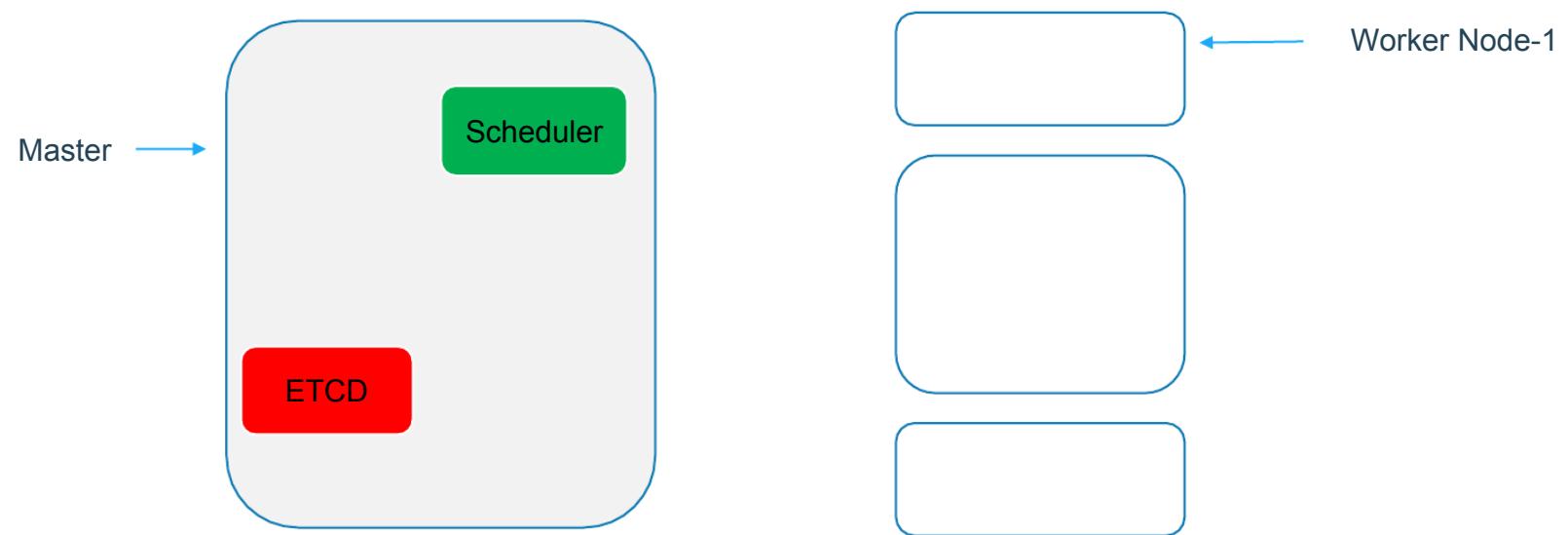


# Cargo Ship Profiles

HA database ~ Which containers on which ships? When was it loaded? ~ **The ETCD Cluster**

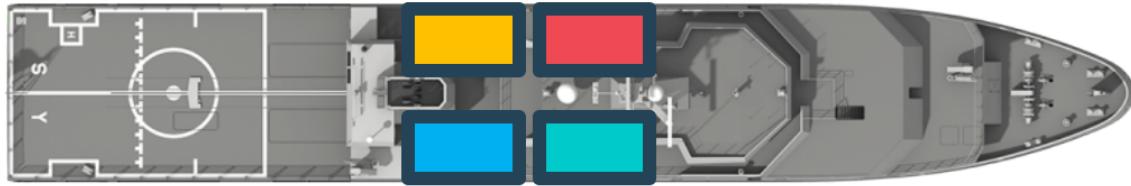


# Overview



# Offices in Dock

- Operation Team Office ~ Ship Handling, Control
- Cargo Team Office ~ verify if containers are damaged, ensure that new containers are rebuilt
- IT & Communication Office – Communication in between various ships

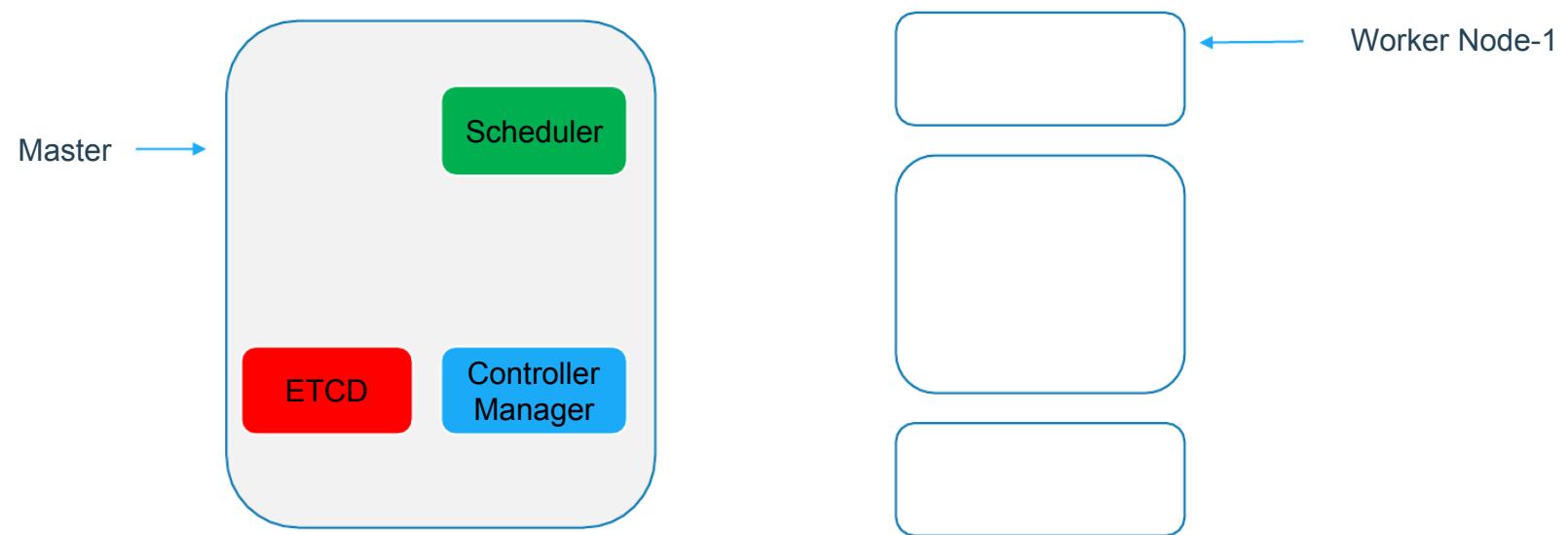


# Controllers

- **Node Controllers** – Takes care of Nodes | Responsible for onboarding new nodes in a cluster | Availability of Nodes
- **Replicas Controller** – Ensures that desired number of containers are running at all times
- **Controller Manager** - Manages all these controllers in place



# Overview

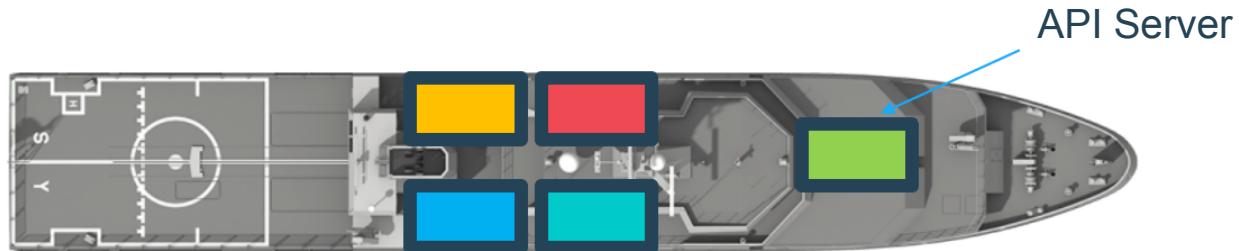


# How does each of these service communicate with each other?

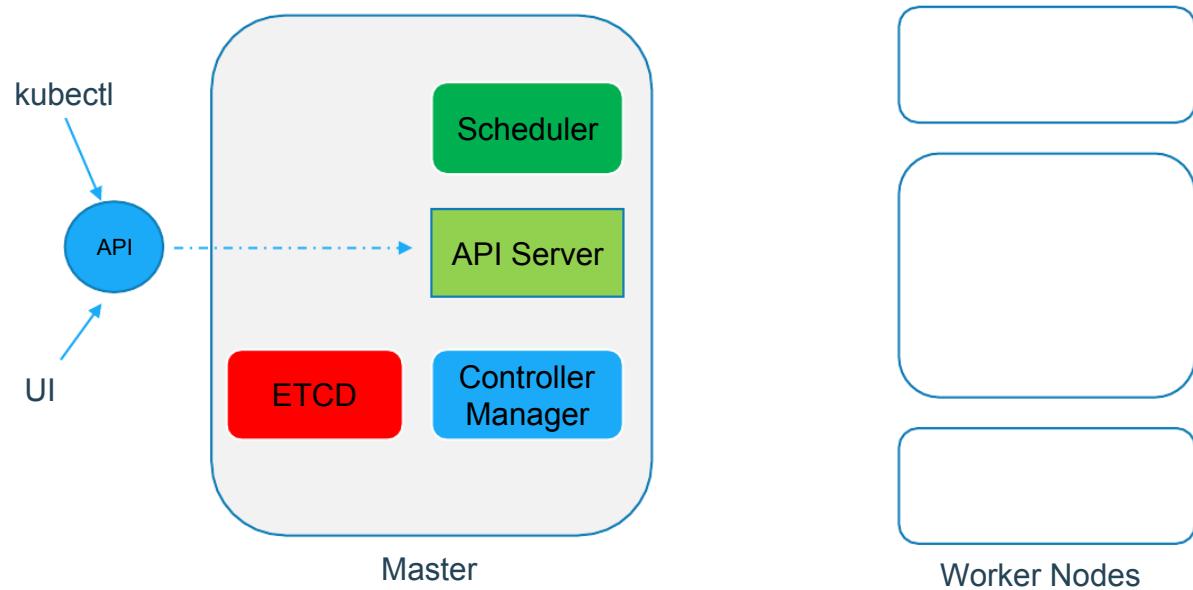


# Kube API Server

- A primary management component of k8s
- Responsible for orchestrating all operations within a cluster
- Exposes K8s API ,used by external users to perform management operation in the cluster and number of controller to monitor the state of the cluster



# Overview



# In nutshell...

## \$kubectl get componentstatus

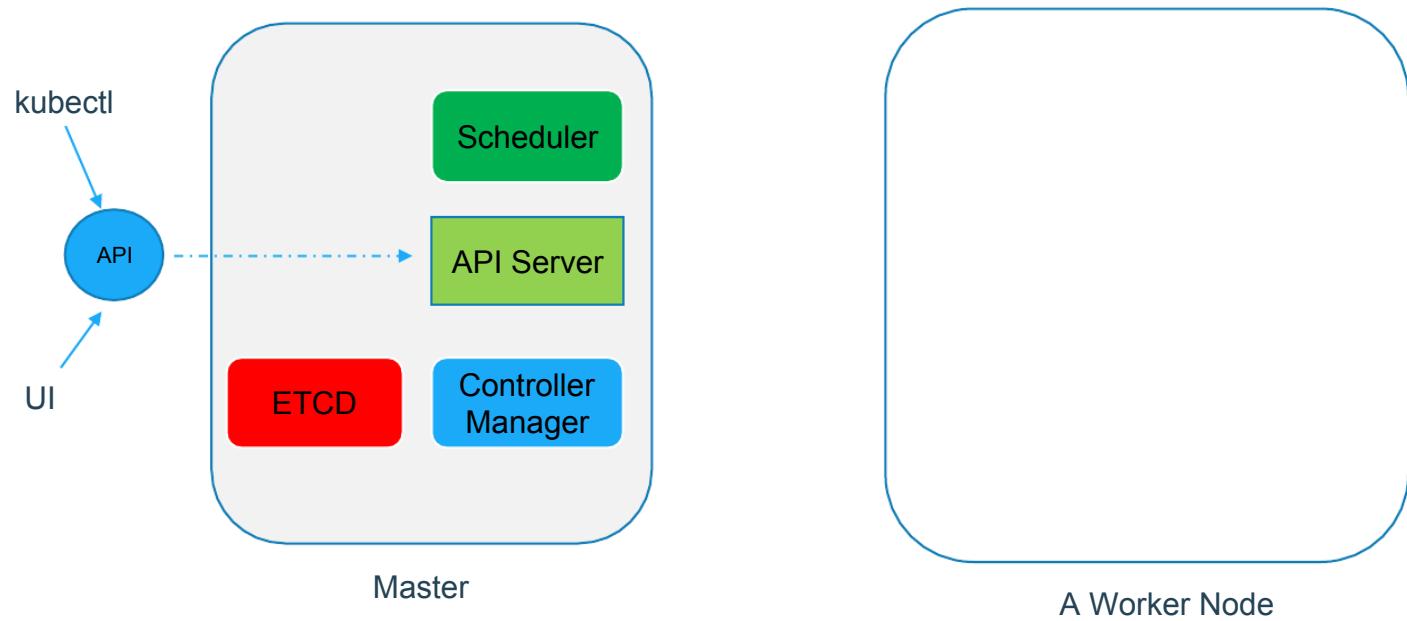
```
[node1 install]$ kubectl get nodes -o wide
NAME    STATUS    ROLES    AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE    KERNEL-VERSION    CONTAINER-
RUNTIME
node1  Ready    master    92s    v1.14.2    192.168.0.18    <none>    CentOS Linux 7 (Core)    4.4.0-141-generic    docker://18.9.6
node2  Ready    <none>    57s    v1.14.2    192.168.0.17    <none>    CentOS Linux 7 (Core)    4.4.0-141-generic    docker://18.9.6
node3  NotReady  <none>    39s    v1.14.2    192.168.0.16    <none>    CentOS Linux 7 (Core)    4.4.0-141-generic    docker://18.9.6
node4  NotReady  <none>    32s    v1.14.2    192.168.0.15    <none>    CentOS Linux 7 (Core)    4.4.0-141-generic    docker://18.9.6
```

```
[node1 install]$ kubectl get componentstatus
NAME    STATUS    MESSAGE    ERROR
scheduler    Healthy    ok
controller-manager    Healthy    ok
etcd-0        Healthy    {"health":"true"}
```

# Let's talk about Worker Components..

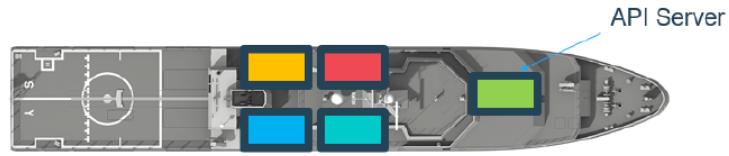


# Overview



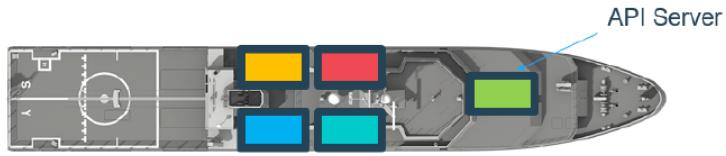
# Captain of the Ship

- Manages all sort of activity on the ship
- Let master ship knows they are interested to join
- Sending reports back to master about the status of the ship
- Sending reports about the status of the containers

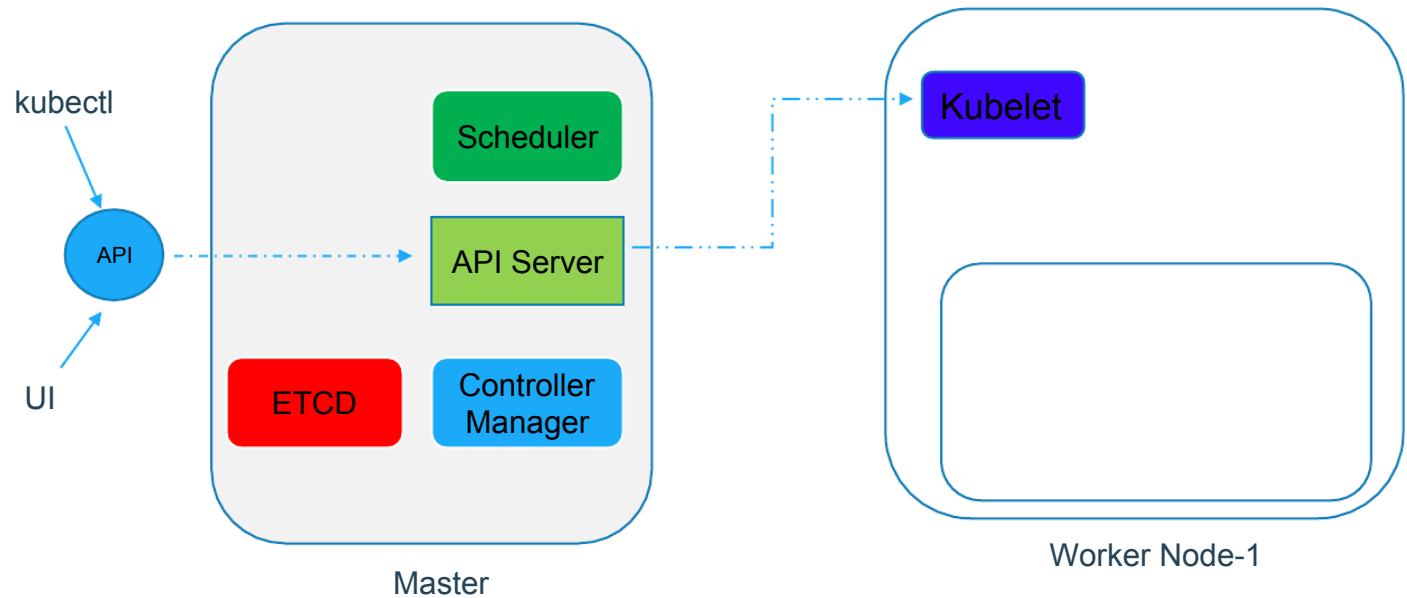


# Captain of the Ship ~ Kubelet

Agent which runs on each nodes of the container

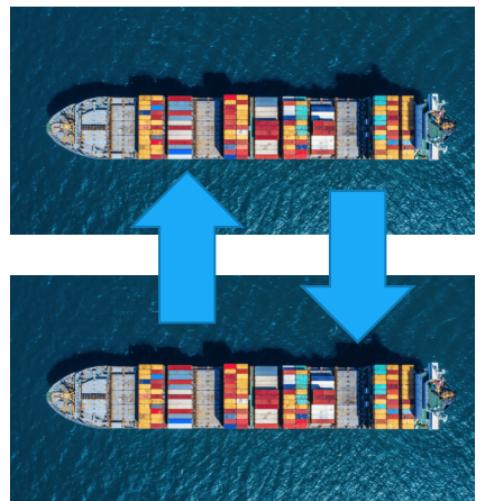


# Overview



# Communication between Cargo Ships

How does two cargo ships communicate with each other?

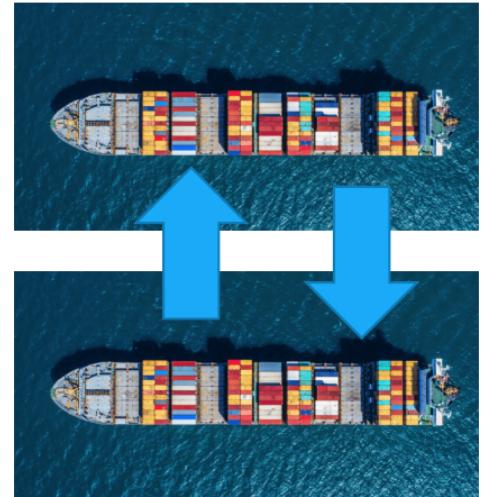


# Kube-proxy Service

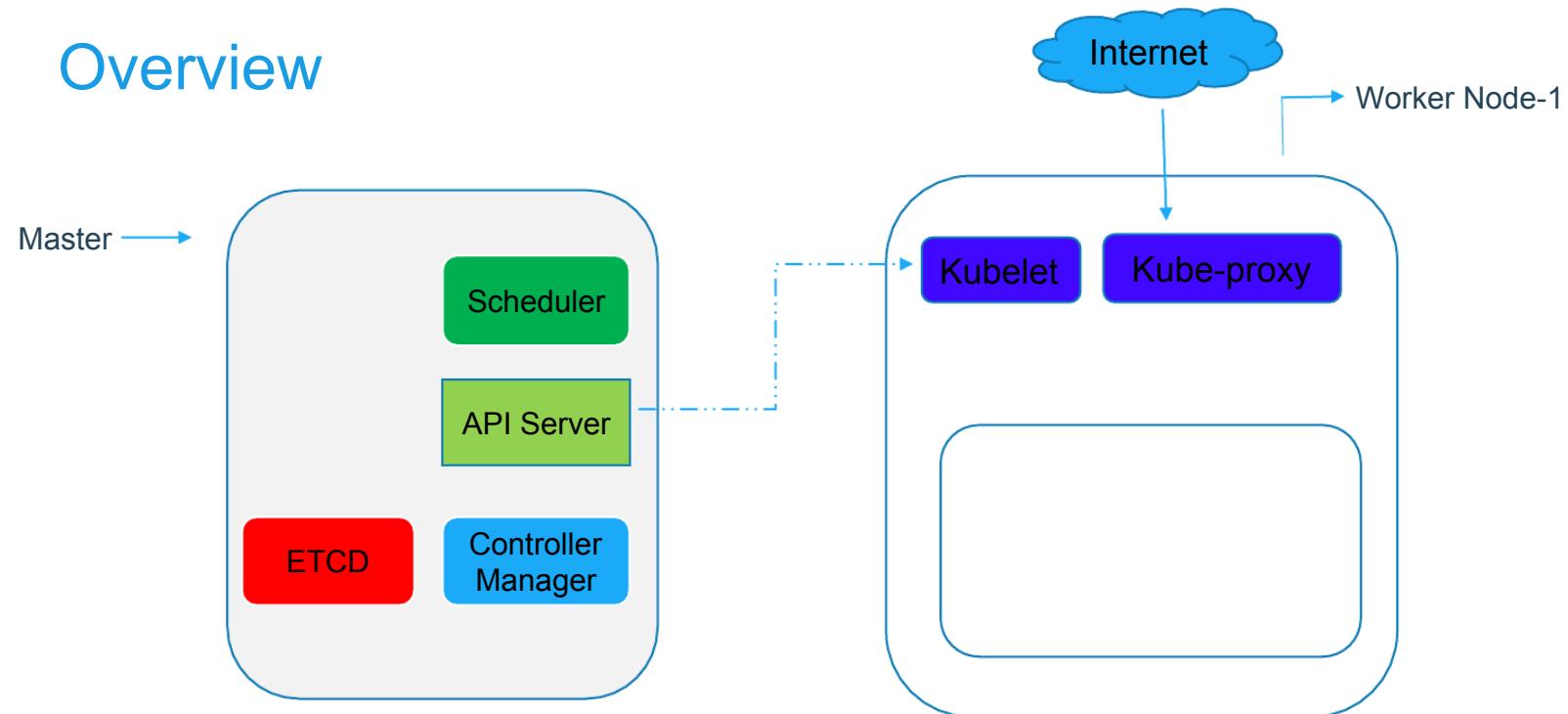
How will web server running on one worker node reach out to DB server on another worker node?

Communication between worker nodes

Kube-proxy



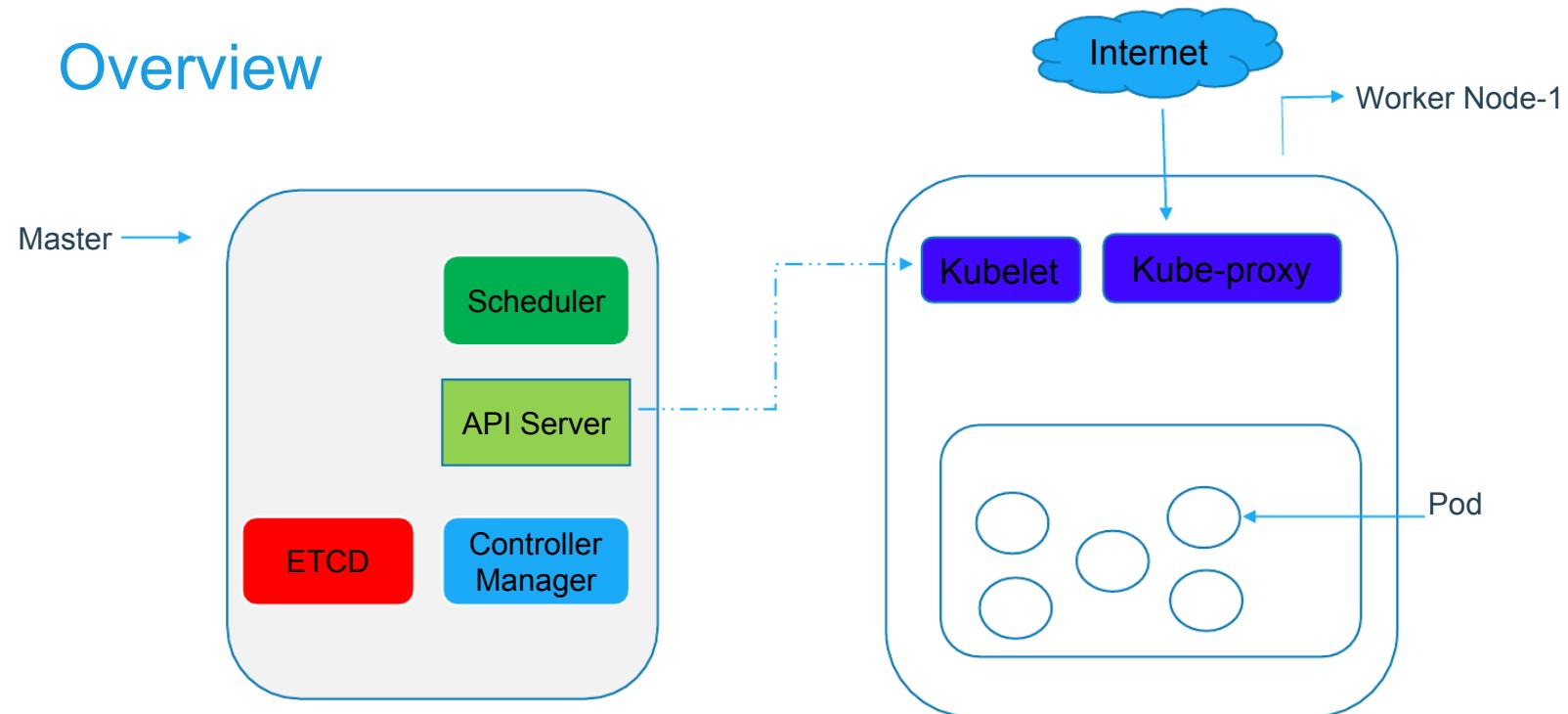
# Overview



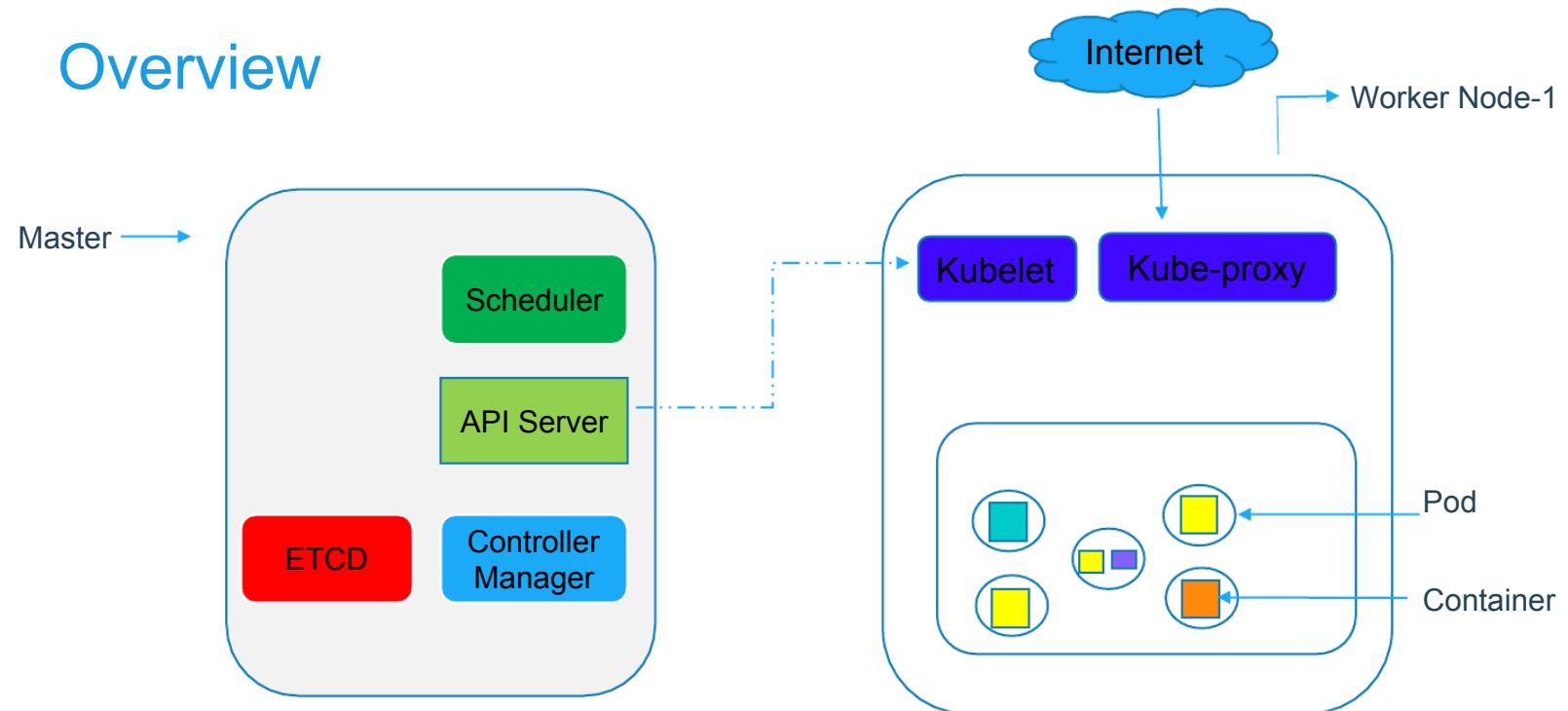
# Let's talk about Pods..



# Overview

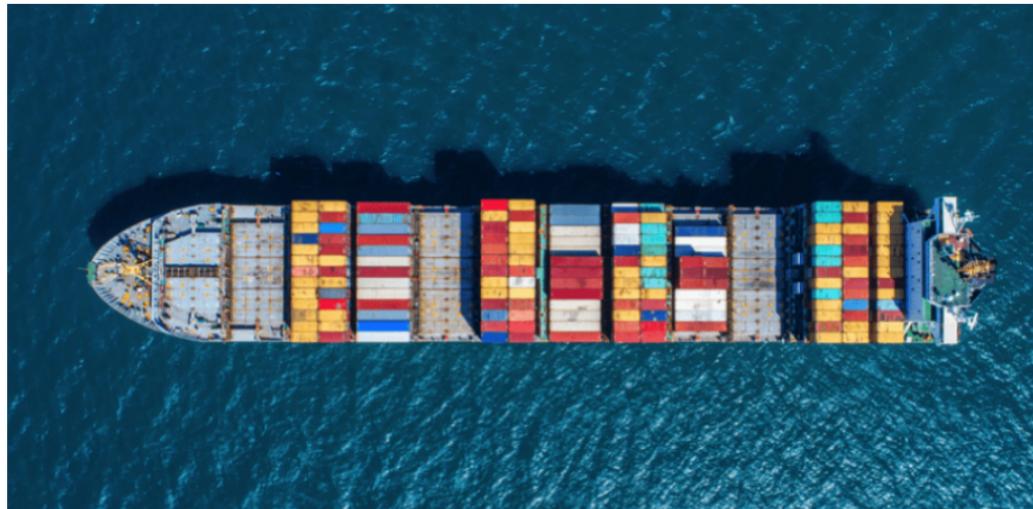


# Overview

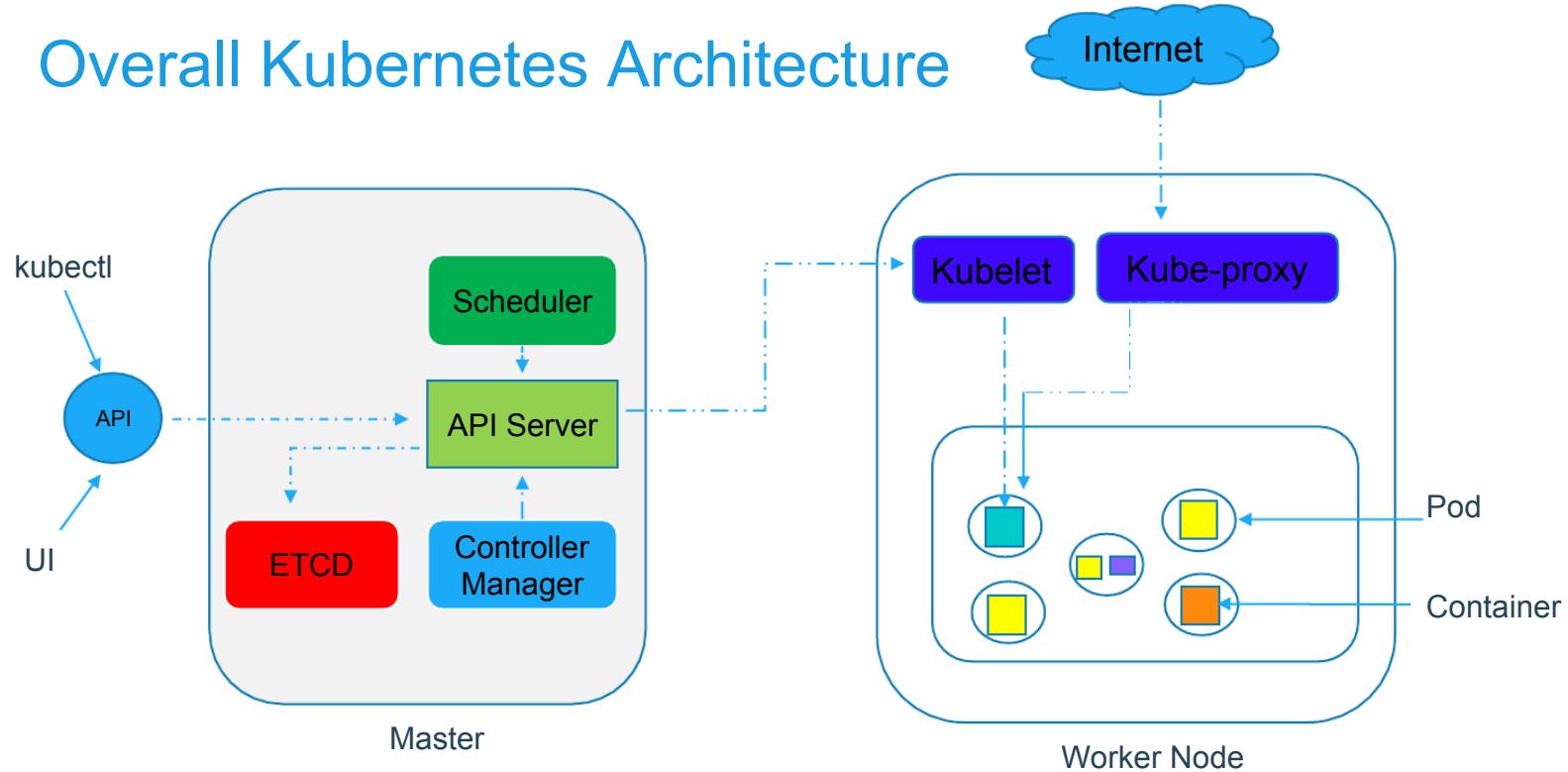


# Docker Containers

A popular Container Runtime



# Overall Kubernetes Architecture



# Demo

- Setting up 5 Node Kubernetes Cluster on PWK