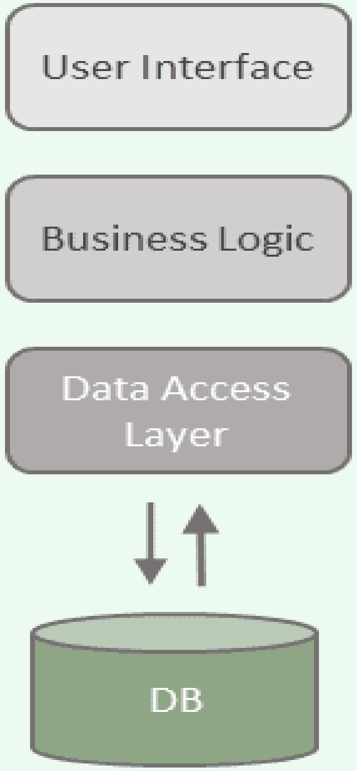
MONOLITHIC ARCHITECTURE



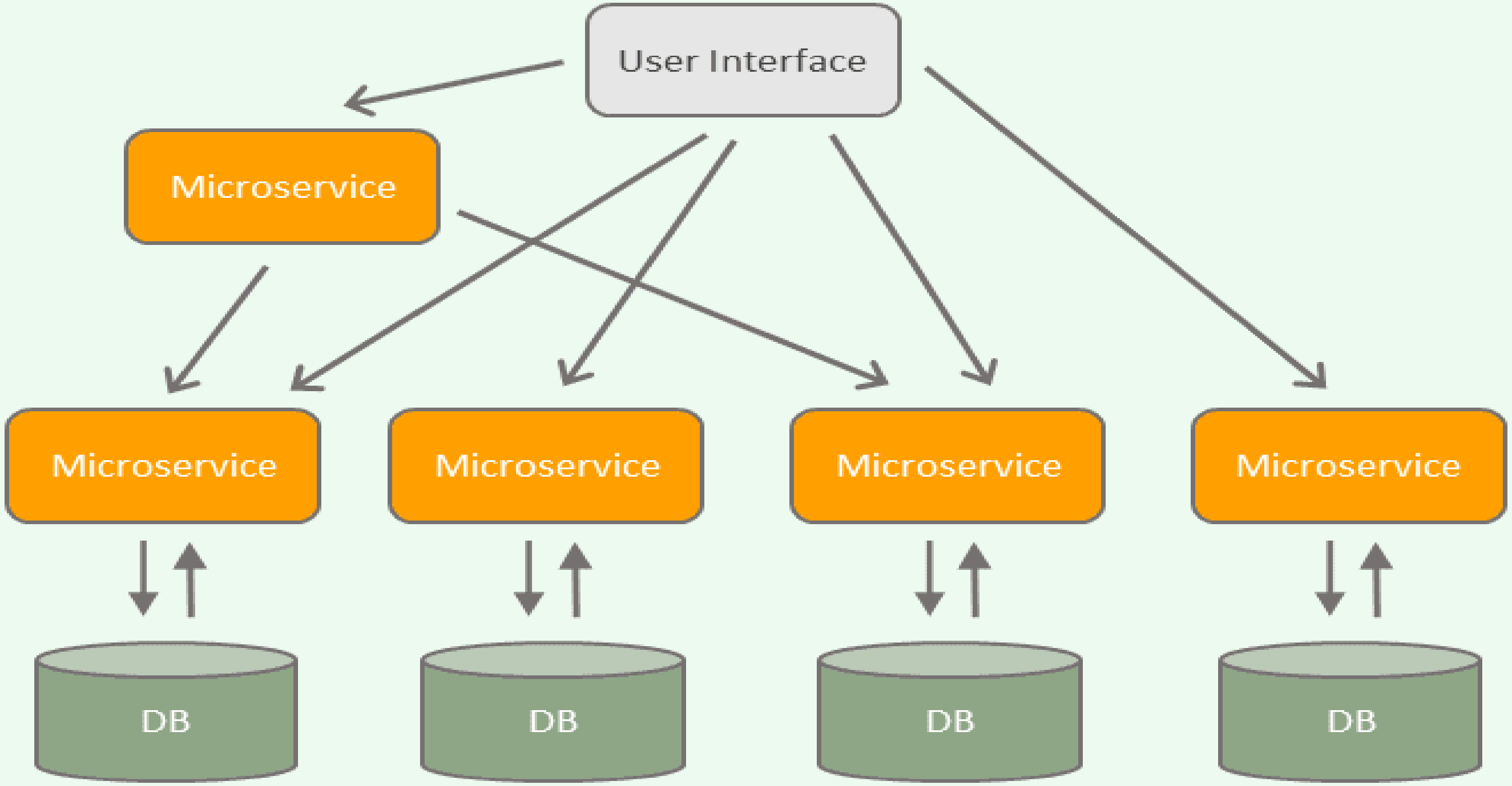
User Interface

Business Logic

Data Access Layer

DB

MICROSERVICES ARCHITECTURE



User Interface

Microservice

Microservice

Microservice

Microservice

Microservice

DB

DB

DB

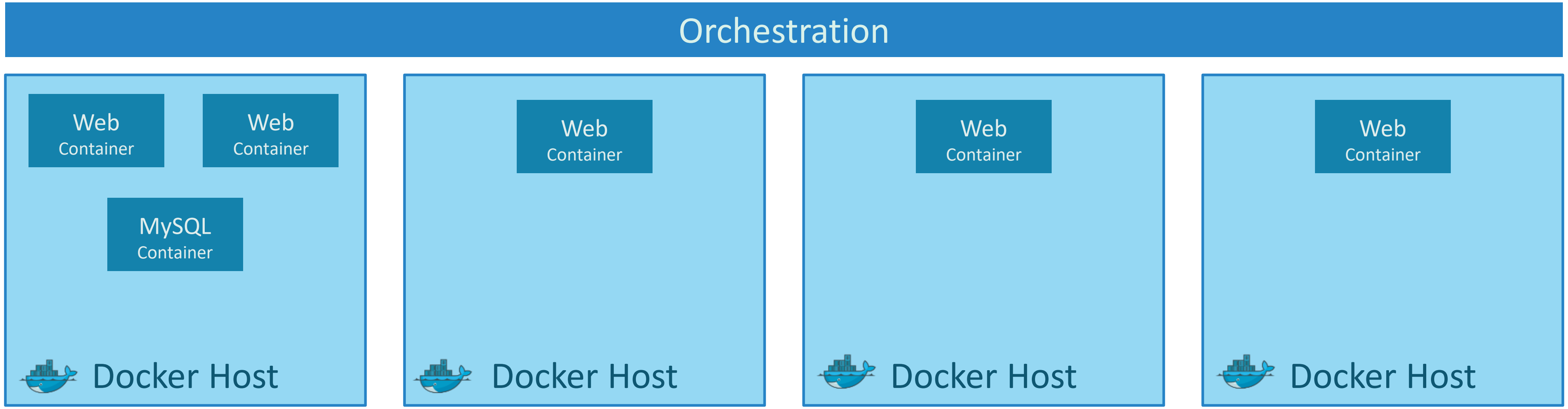
DB

Container Orchestration

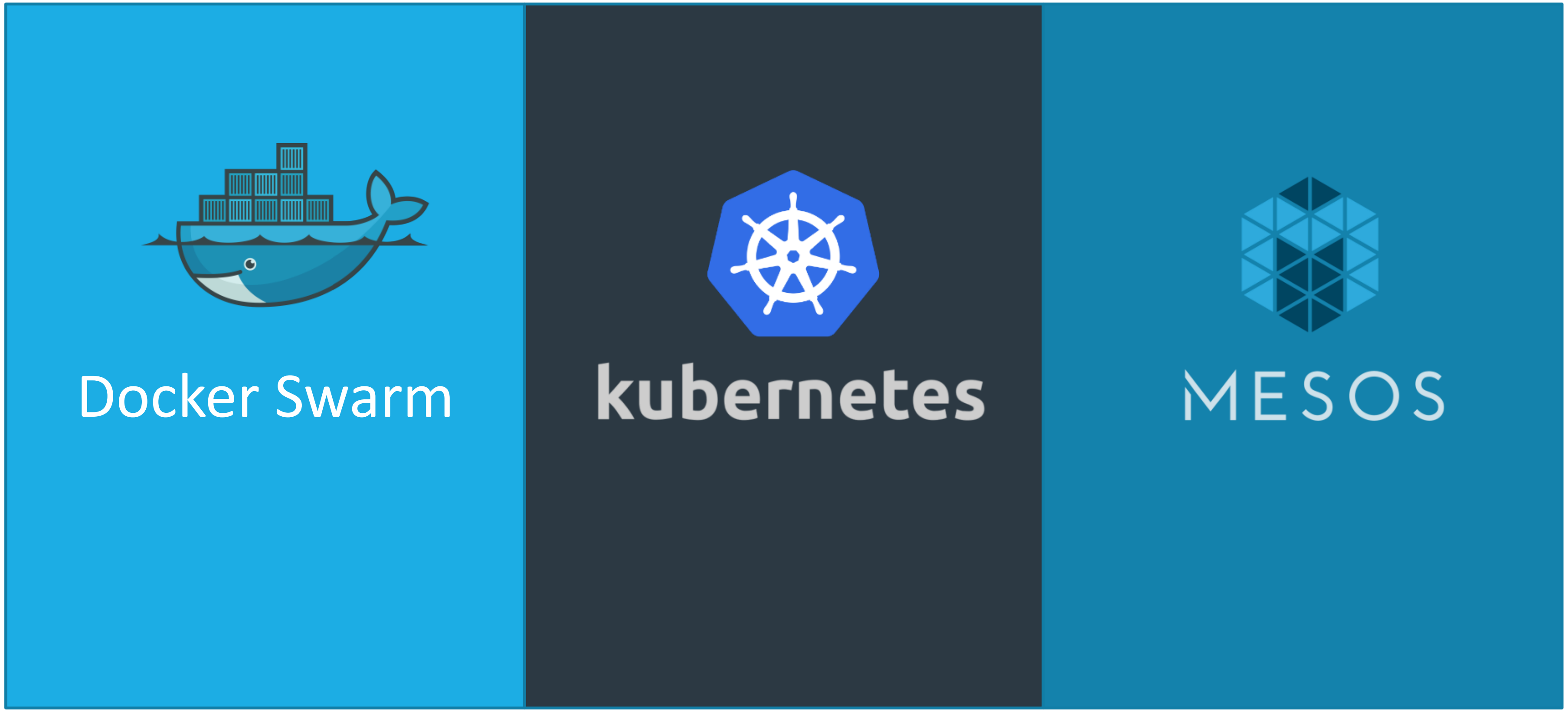
Containers Limitation?

High Availability?
Overlay Network?
Versioning of Application – Rollout, Rollback?
Scaling?
Autoscaling?
Monitoring?
Dependency between containers?

Container orchestration



Orchestration Technologies



Docker Swarm

kubernetes

MESOS

What is Kubernetes?

The Kubernetes project was started by Google in 2014.

Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale.

Kubernetes can run on a range of platforms, from your laptop, to VMs on a cloud provider, to rack of bare metal servers.

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.

portable: with all public, private, hybrid, community cloud

self-healing: auto-placement, auto-restart, auto-replication, auto-scaling

Why Kubernetes

Kubernetes can schedule and run application containers on clusters of physical or virtual machines.

host-centric infrastructure to a **container-centric** infrastructure.

Orchestrator

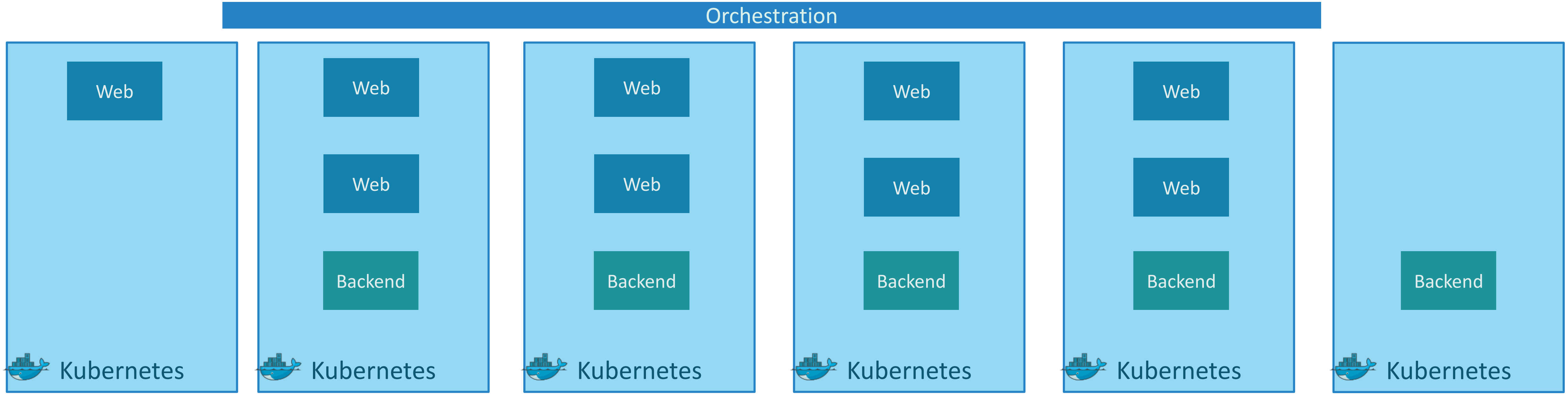
Load balancing

Auto Scaling

Application Health checks

Rolling updates

Kubernetes Advantage



Orchestration

Web	Web	Web	Web	Web	
	Web	Web	Web	Web	
	Backend	Backend	Backend	Backend	Backend
Kubernetes	Kubernetes	Kubernetes	Kubernetes	Kubernetes	Kubernetes

And that is kubernetes..

Setup

Minikube

Kubeadm

Google Cloud Platform

Amazon Web Services

play-with-k8s.com

Setup Kubernetes

Setup - kubeadm

Kubernetes Cluster

A Kubernetes cluster consists of two types of resources:

Master: Which coordinates with the cluster

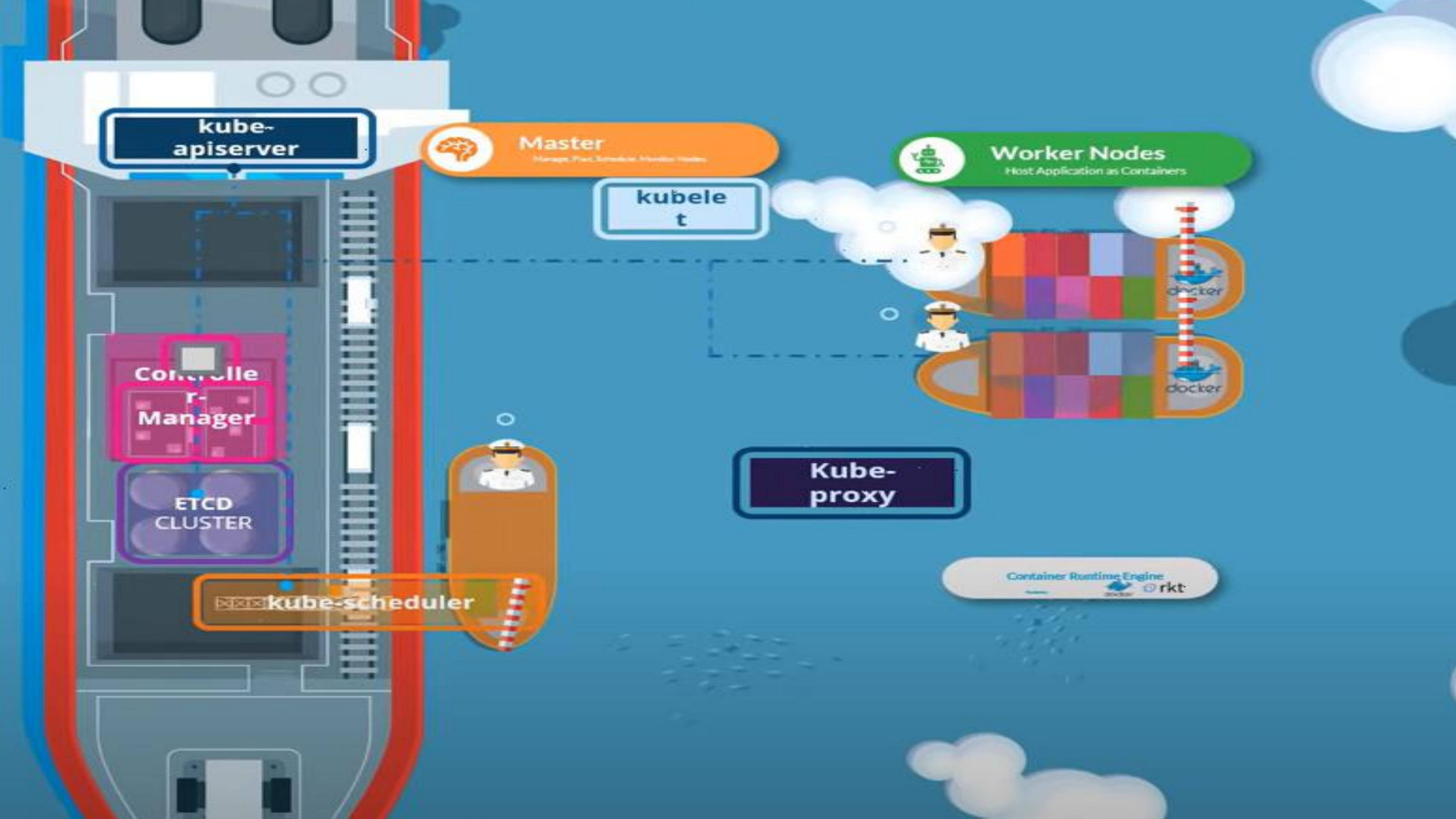
The Master is responsible for managing the cluster. The master coordinates all activities in your cluster, such as scheduling applications, maintaining applications' desired state, scaling applications, and rolling out new updates.

Nodes: Are the workers that run application

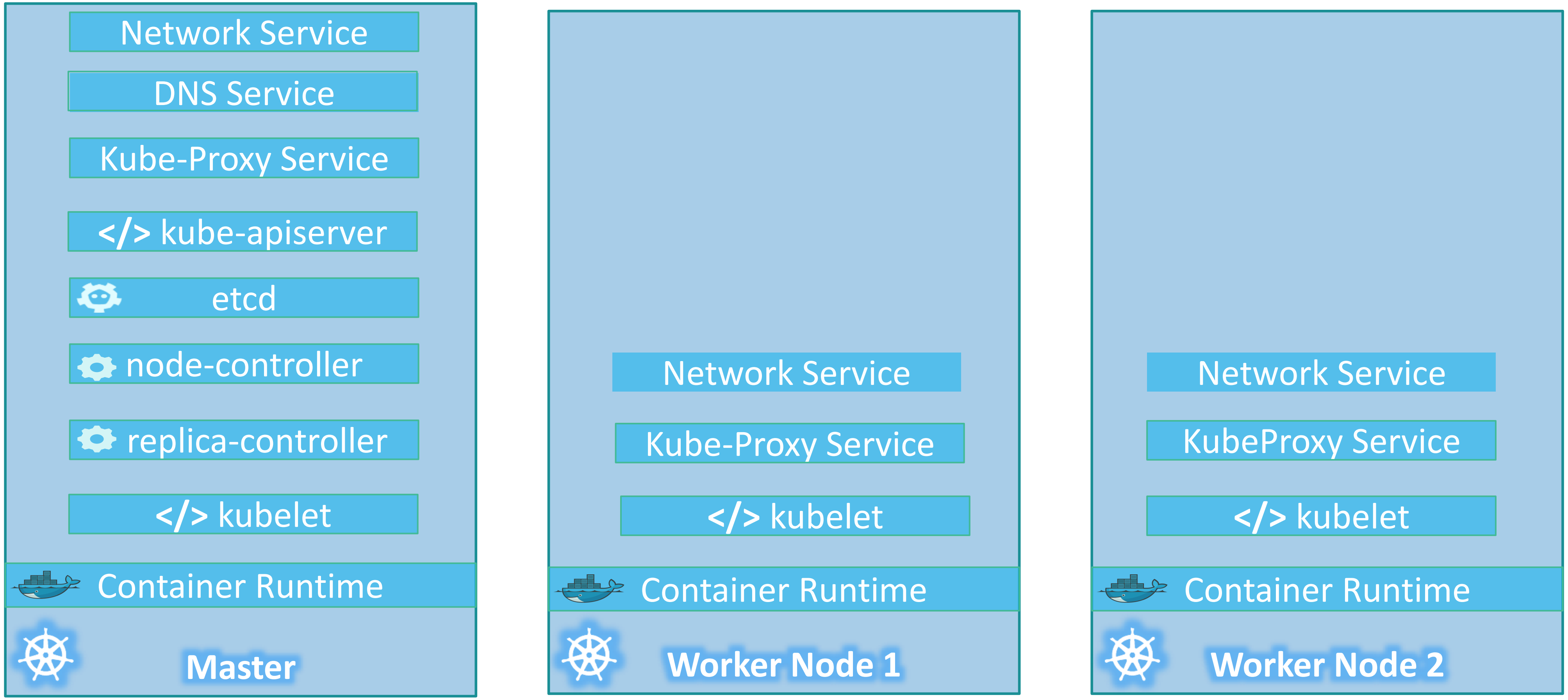
A node is a VM or a physical computer that serves as a worker machine in a Kubernetes cluster.

Masters manage the cluster and the nodes are used to host the running applications.

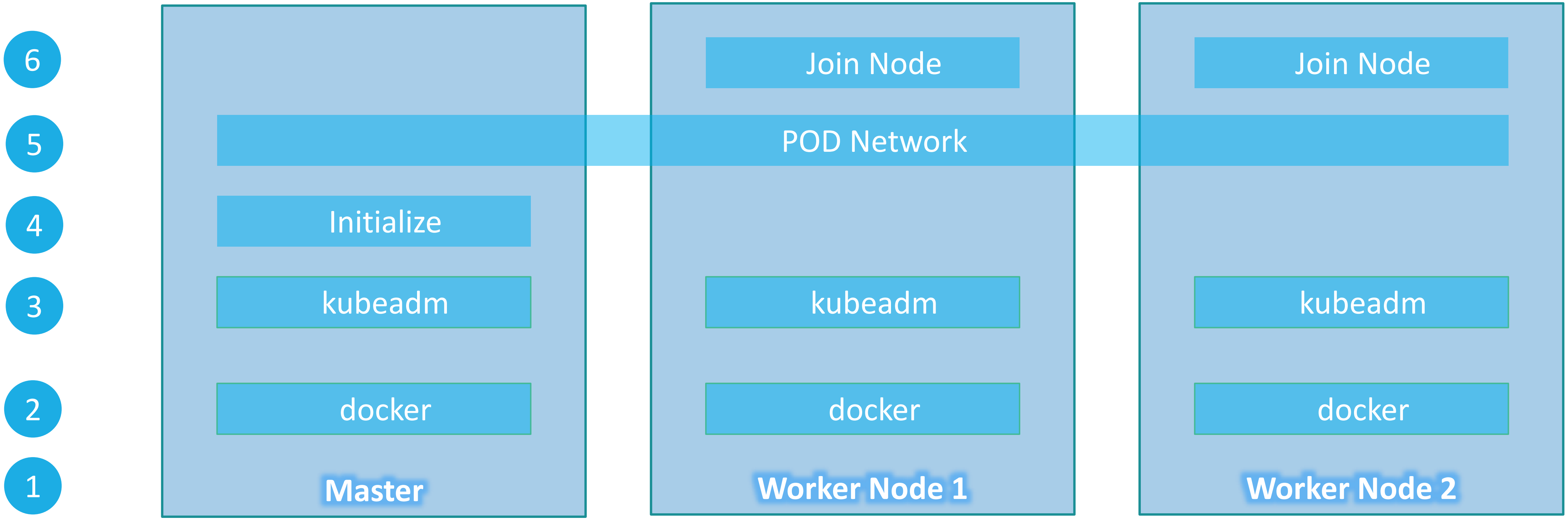
The nodes communicate with the master using the Kubernetes API, which the master exposes.


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

kubeadm


Network Service
DNS Service
Kube-Proxy Service
</> kube-apiserver
etcd
node-controller
replica-controller
</> kubelet
Container Runtime
Master
Network Service
Kube-Proxy Service
</> kubelet
Container Runtime
Worker Node 1
Network Service
KubeProxy Service
</> kubelet
Container Runtime
Worker Node 2

Steps


6
5
4
3
2
1
Join Node
Join Node
POD Network
Initialize
kubeadm
kubeadm
kubeadm
docker
docker
docker
Master
Worker Node 1
Worker Node 2

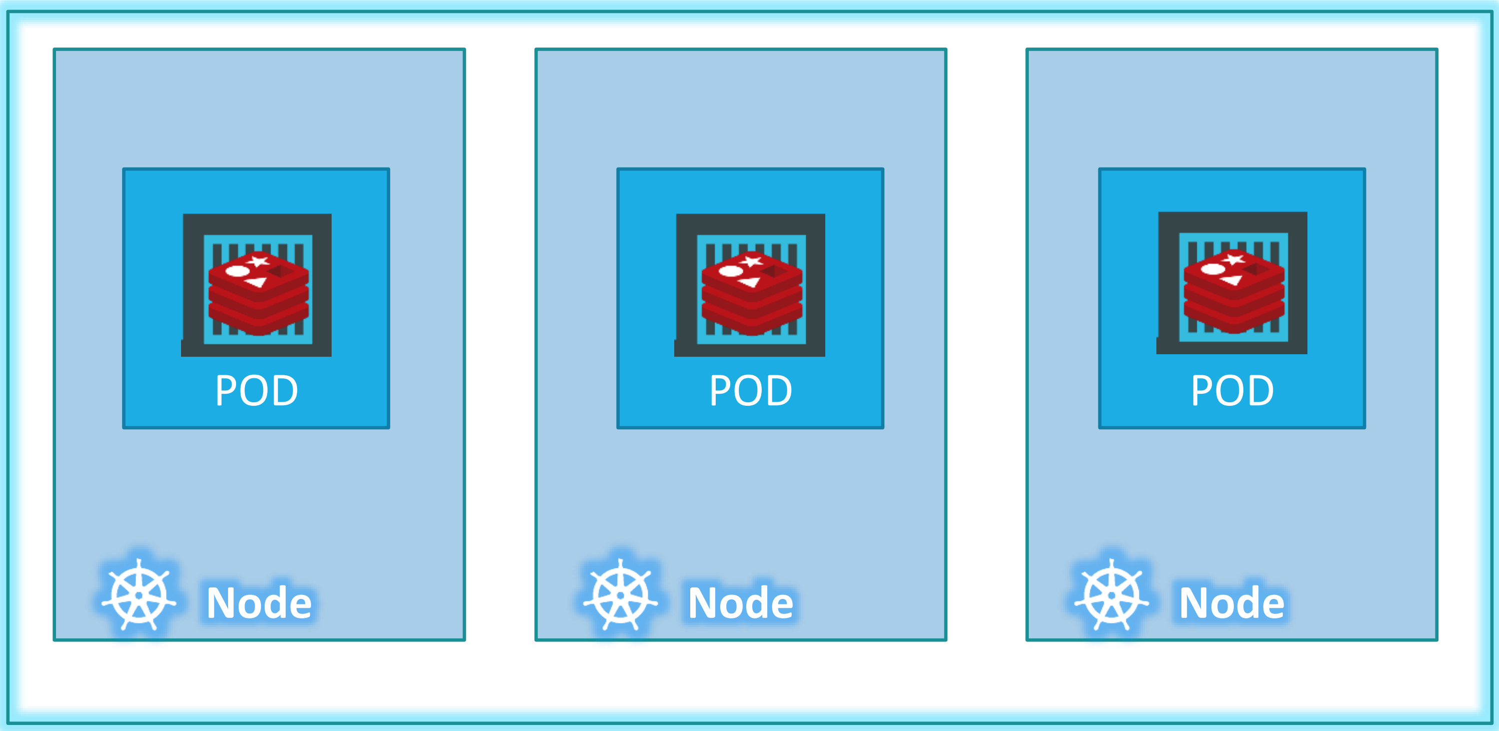
POD

Assumptions

Docker Image

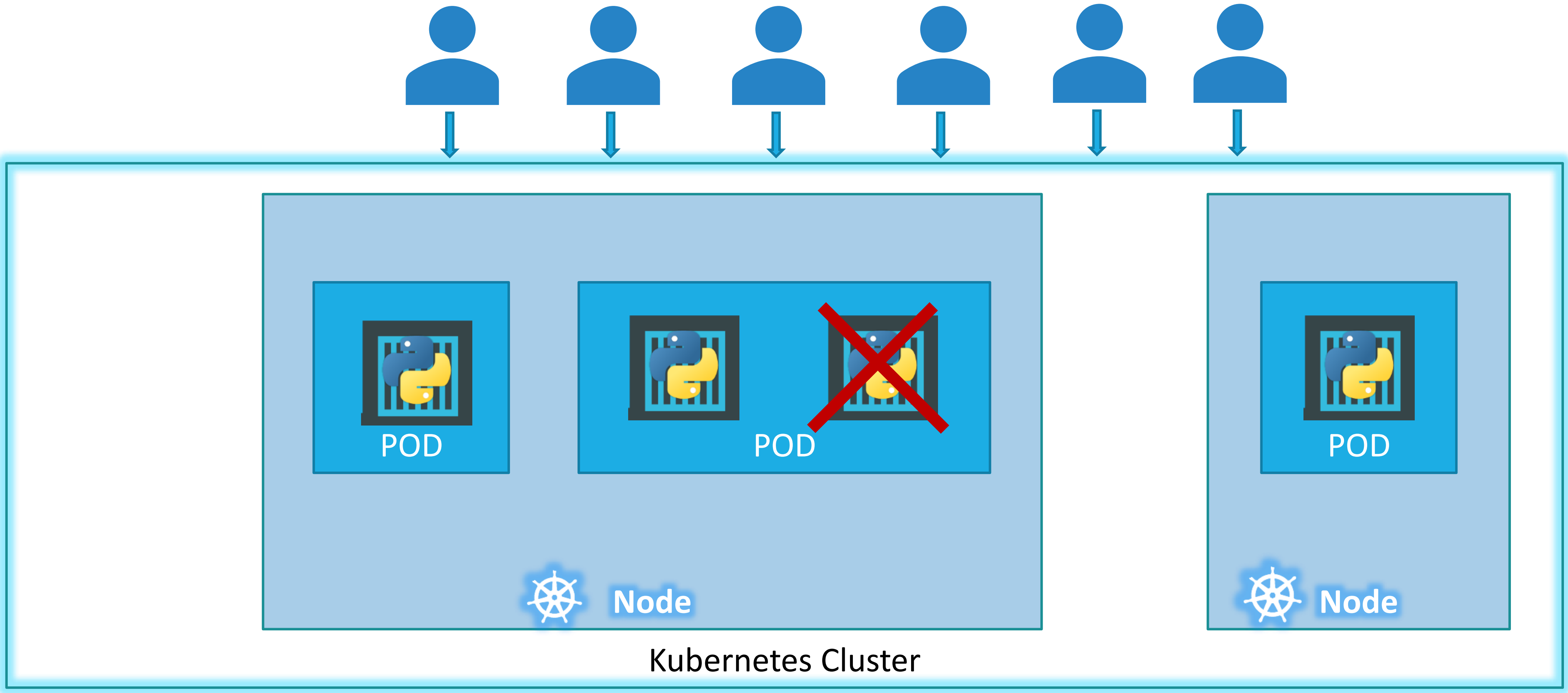
Kubernetes Cluster

POD

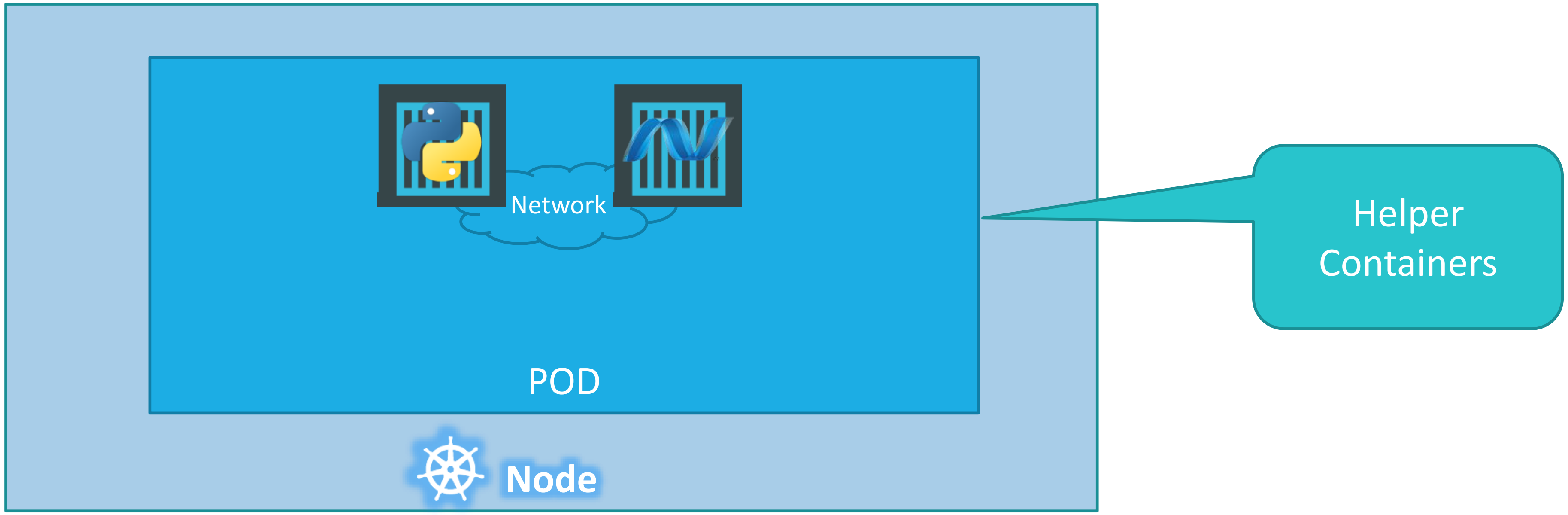


TechLanders
Delivering the future

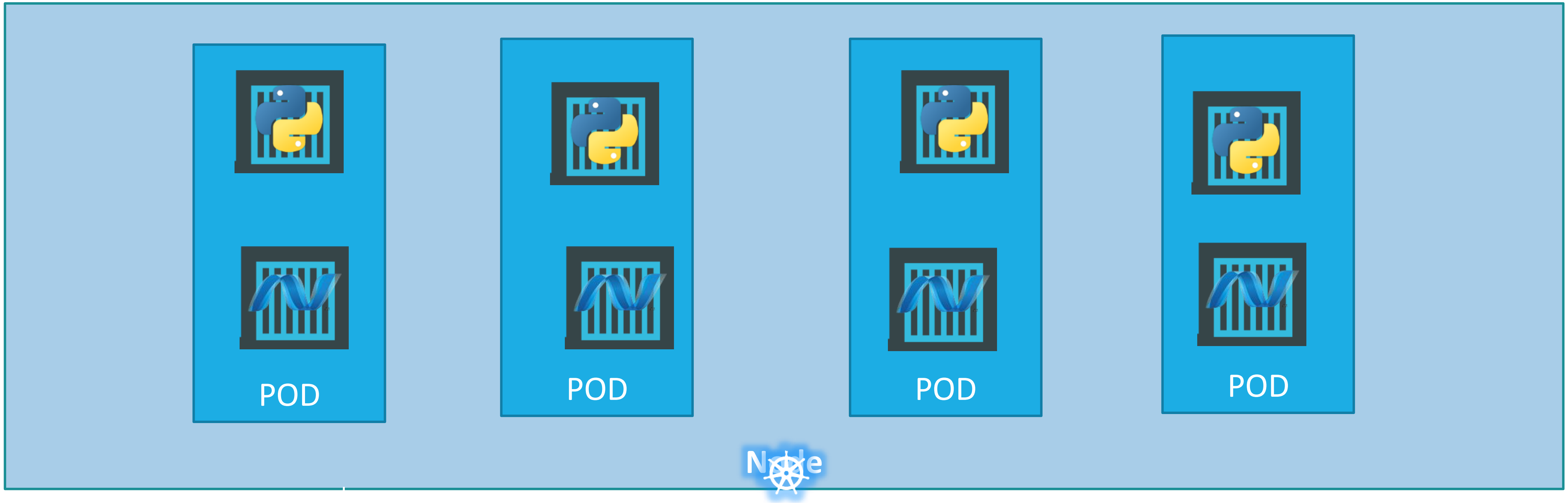
POD



Multi-Container PODs



PODs Again!

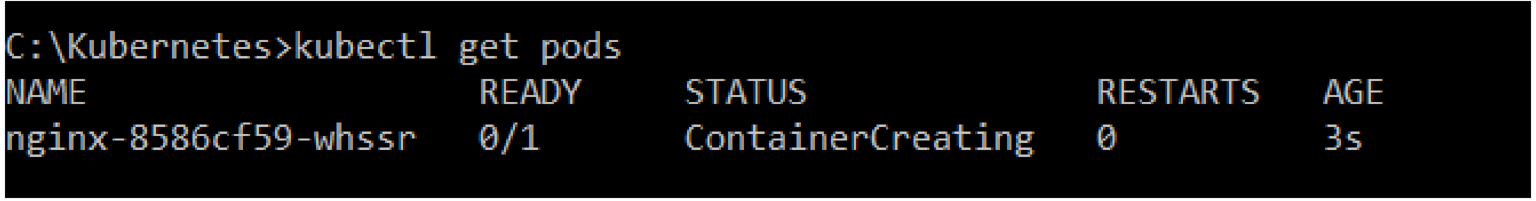


Note: I am avoiding networking and load balancing details to keep explanation simple.

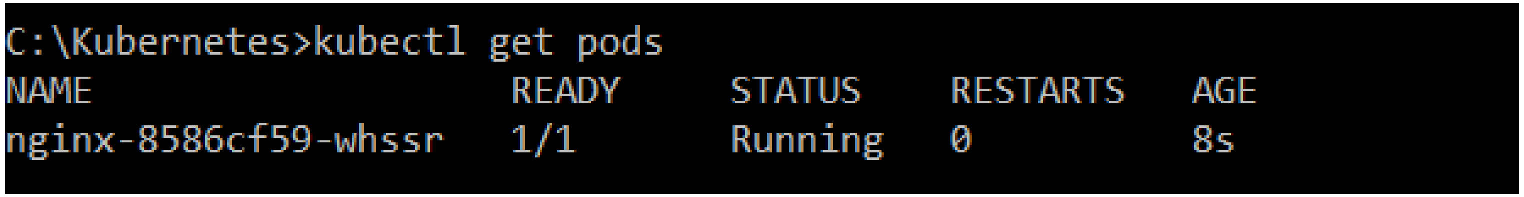
kubectl

```
kubectl run nginx --image nginx
```

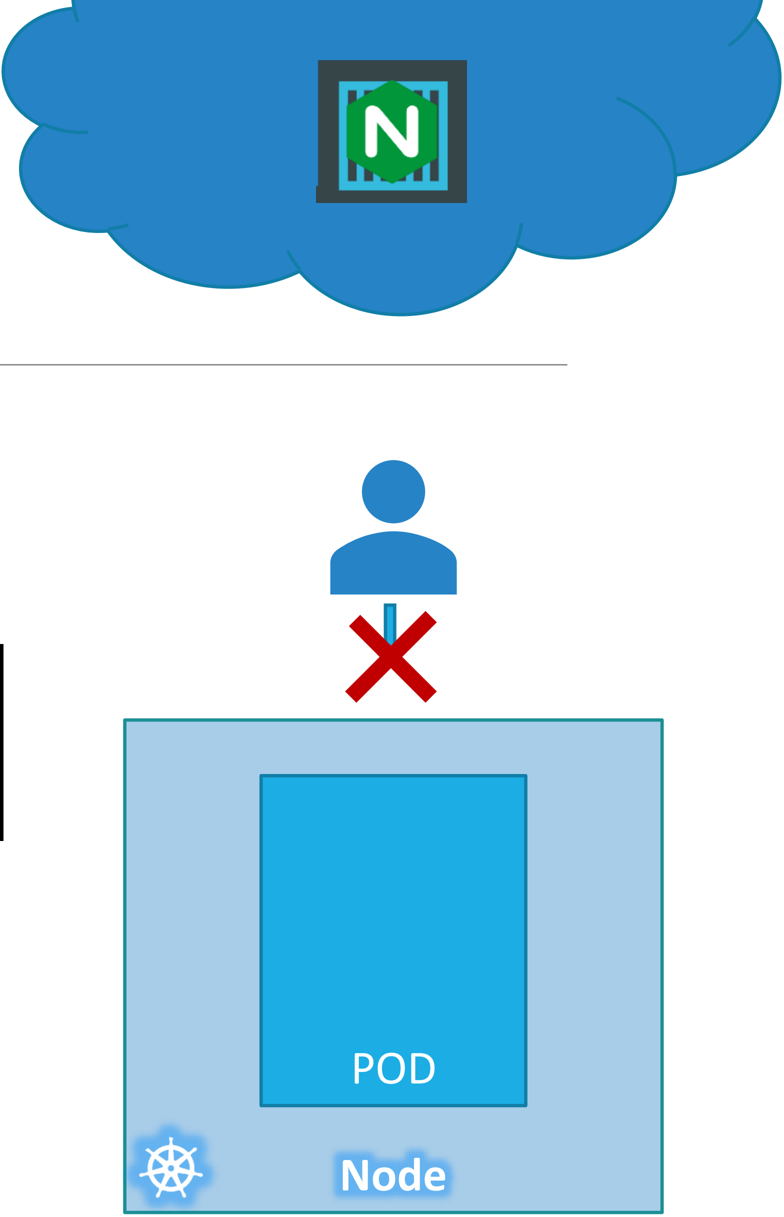
```
kubectl get pods
```



```
C:\Kubernetes>kubectl get pods
NAME                    READY     STATUS              RESTARTS   AGE
nginx-8586cf59-whssr    0/1       ContainerCreating   0          3s
```



```
C:\Kubernetes>kubectl get pods
NAME                    READY     STATUS    RESTARTS   AGE
nginx-8586cf59-whssr    1/1       Running   0          8s
```

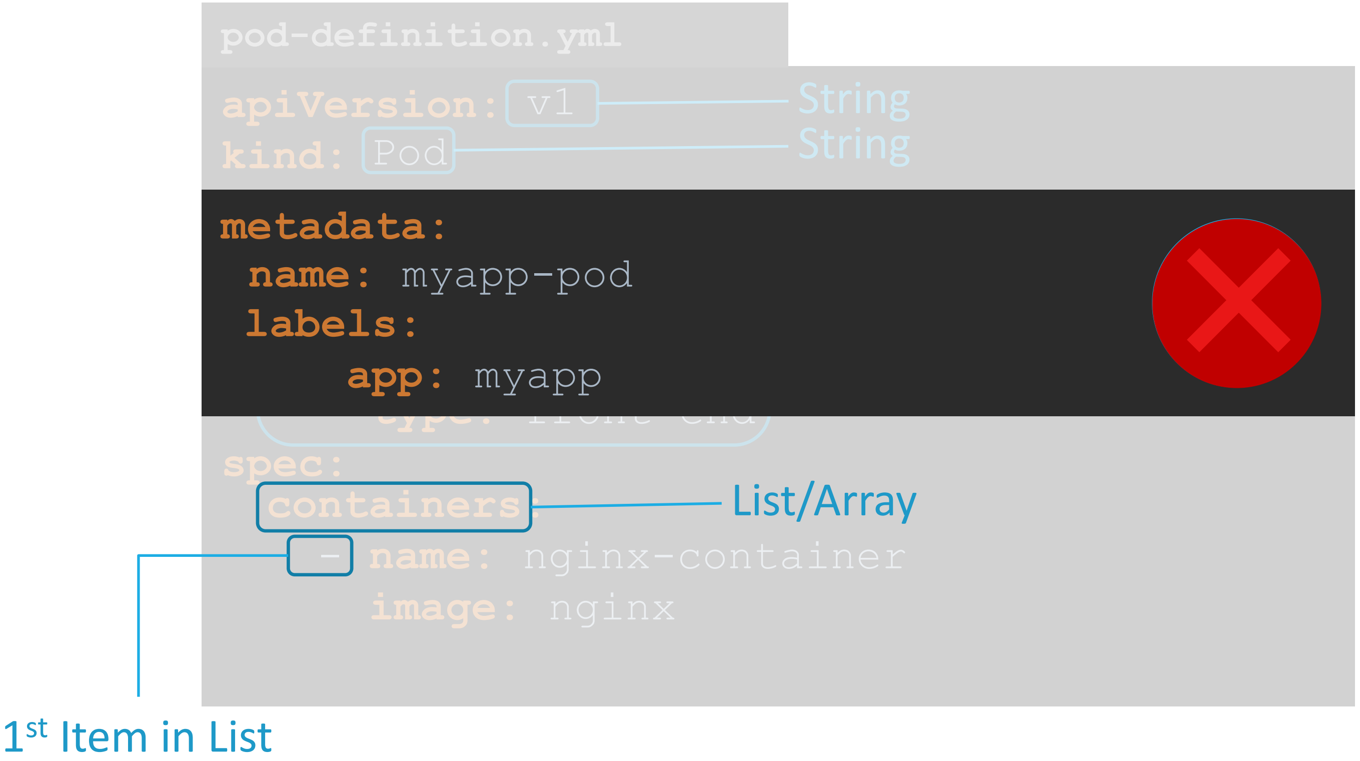


YAML Introduction

POD

With YAML

YAML in Kubernetes

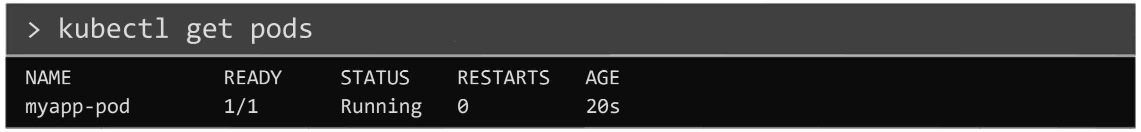


```
pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
 name: myapp-pod
 labels:
     app: myapp
     type: front-end
spec:
  containers
   - name: nginx-container
     image: nginx
```

```
kubectl create -f pod-definition.yml
```

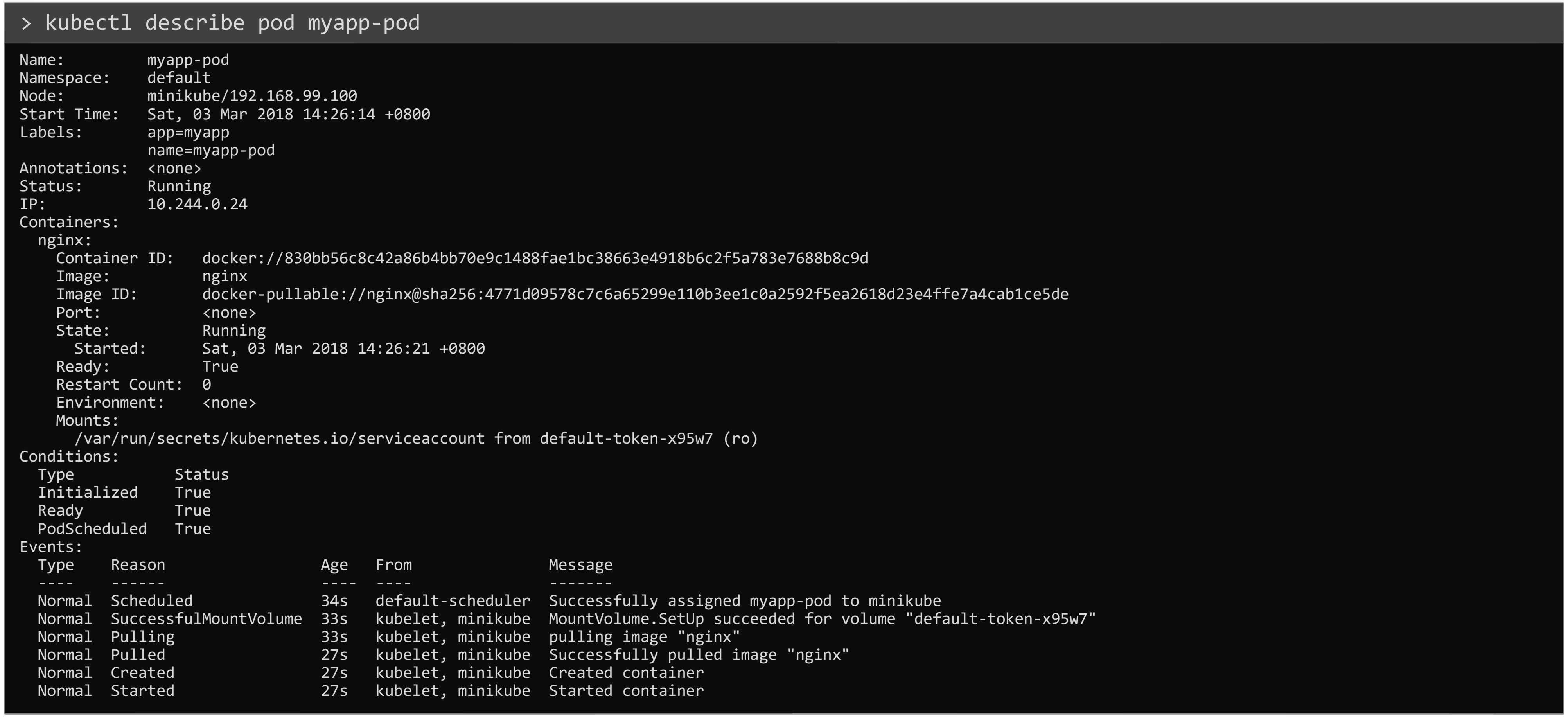
Kind	Version
POD	v1
Service	v1
ReplicaSet	apps/v1
Deployment	apps/v1

Commands



```
> kubectl get pods
```

```
NAME             READY     STATUS    RESTARTS   AGE
myapp-pod        1/1       Running   0          20s
```



```
> kubectl describe pod myapp-pod
```

```
Name:         myapp-pod
Namespace:    default
Node:         minikube/192.168.99.100
Start Time:   Sat, 03 Mar 2018 14:26:14 +0800
Labels:       app=myapp
              name=myapp-pod
Annotations:  <none>
Status:       Running
IP:           10.244.0.24
Containers:
  nginx:
    Container ID:   docker://830bb56c8c42a86b4bb70e9c1488fae1bc38663e4918b6c2f5a783e7688b8c9d
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:4771d09578c7c6a65299e110b3ee1c0a2592f5ea2618d23e4ffe7a4cab1ce5de
    Port:           <none>
    State:          Running
      Started:      Sat, 03 Mar 2018 14:26:21 +0800
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-x95w7 (ro)
Conditions:
  Type           Status
  Initialized    True
  Ready          True
  PodScheduled   True
Events:
  Type    Reason                 Age   From               Message
  ----    ------                 ----  ----               -------
  Normal  Scheduled              34s   default-scheduler  Successfully assigned myapp-pod to minikube
  Normal  SuccessfulMountVolume  33s   kubelet, minikube  MountVolume.SetUp succeeded for volume "default-token-x95w7"
  Normal  Pulling                33s   kubelet, minikube  pulling image "nginx"
  Normal  Pulled                 27s   kubelet, minikube  Successfully pulled image "nginx"
  Normal  Created                27s   kubelet, minikube  Created container
  Normal  Started                27s   kubelet, minikube  Started container
```