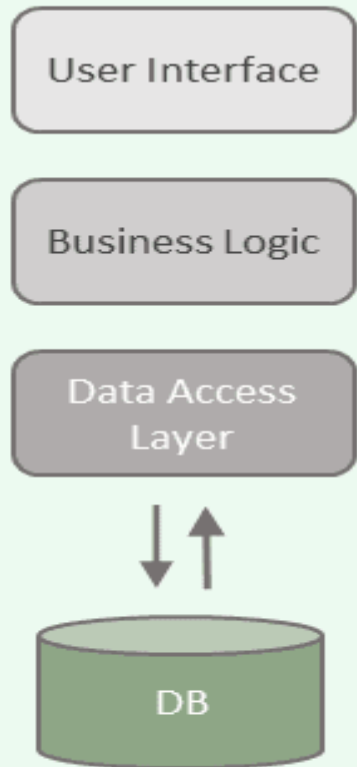
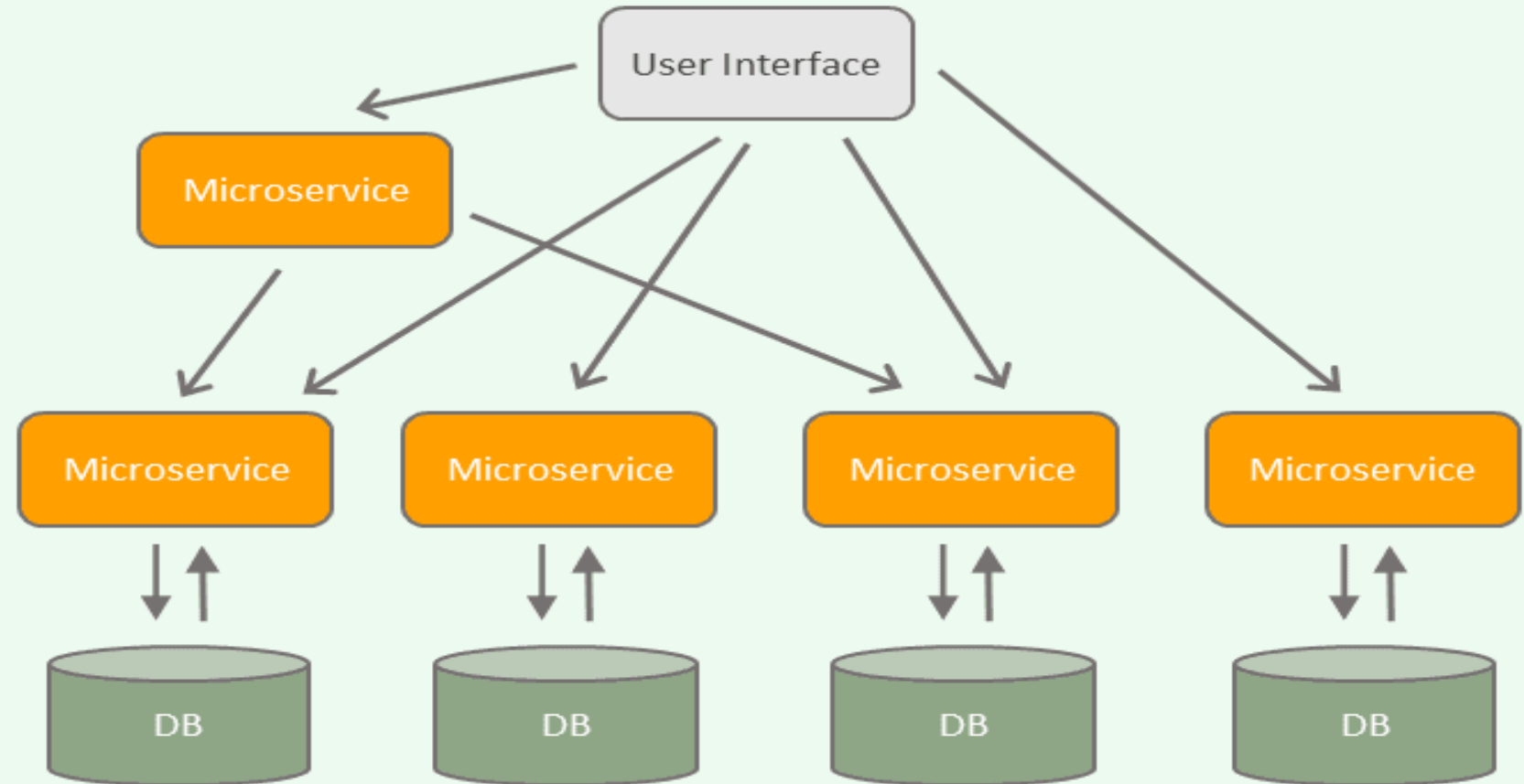


MONOLITHIC ARCHITECTURE



MICROSERVICES ARCHITECTURE



Container Orchestration

Containers Limitation?

High Availability?

Overlay Network?

Versioning of Application – Rollout, Rollback?

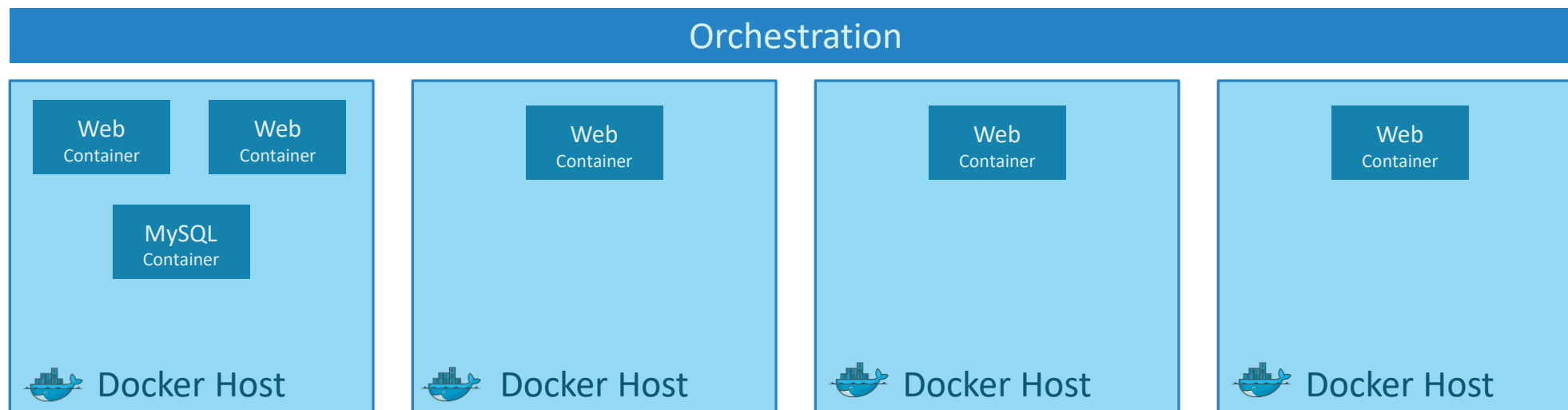
Scaling?

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

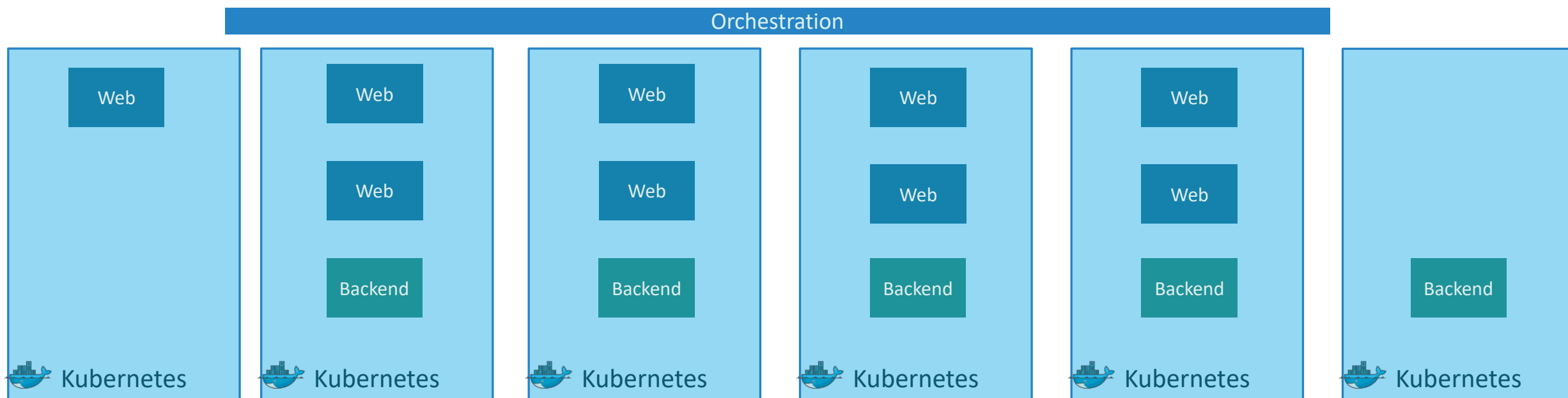
Load balancing

Auto Scaling

Application Health checks

Rolling updates

Kubernetes Advantage



And that is **kubernetes**..

Setup

Raman



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

Manages, Plans, Schedules, Monitors Nodes

kubelet



Worker Nodes

Host Application as Containers

Controller-
Manager

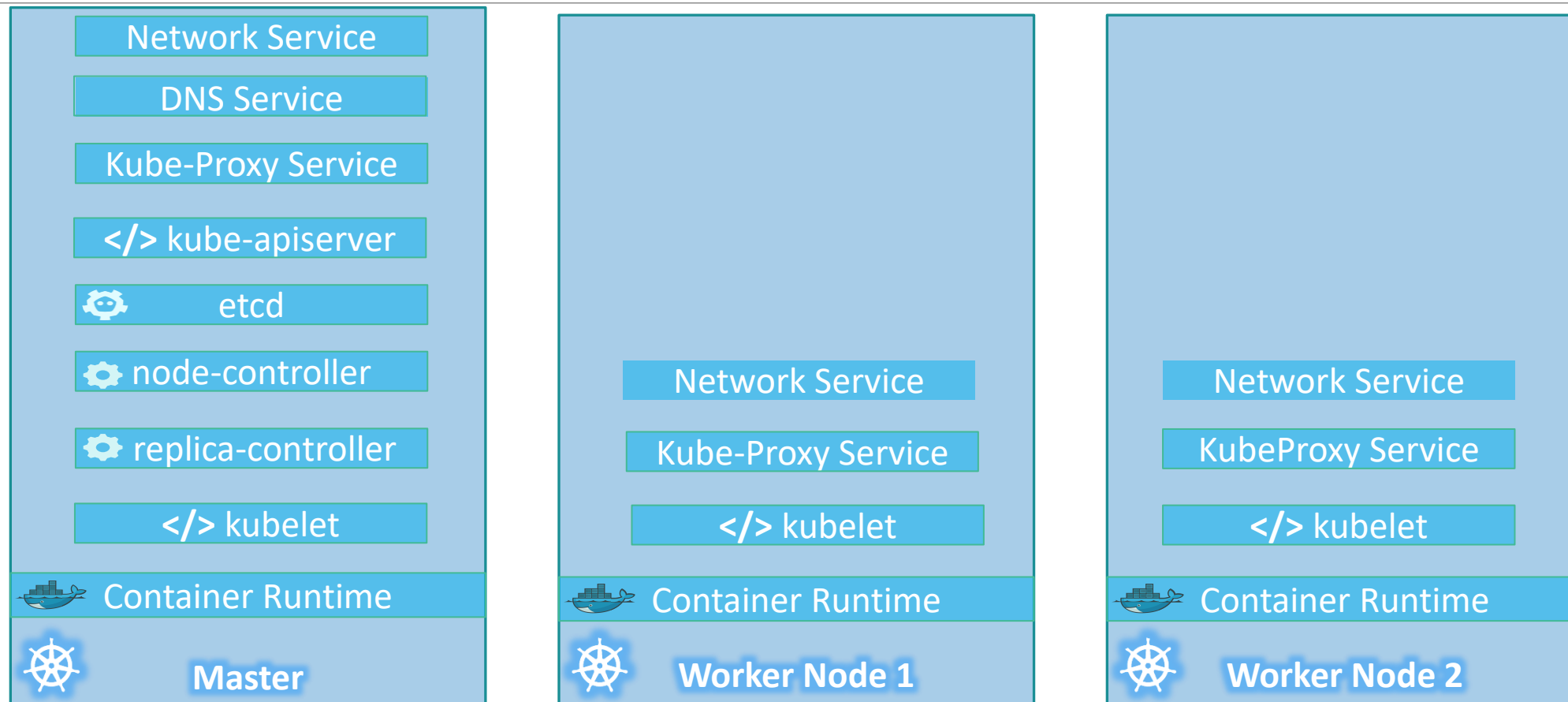
ETCD
CLUSTER

kube-scheduler

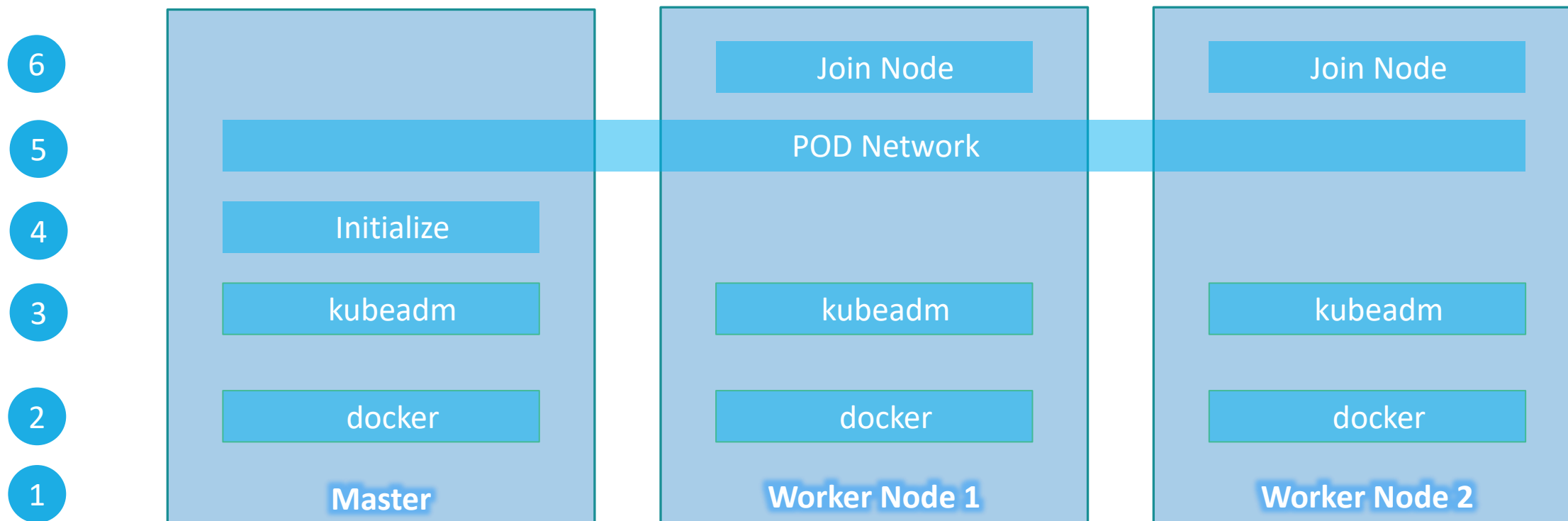
Kube-
proxy

Container Runtime Engine
docker rkt

kubeadm



Steps



POD

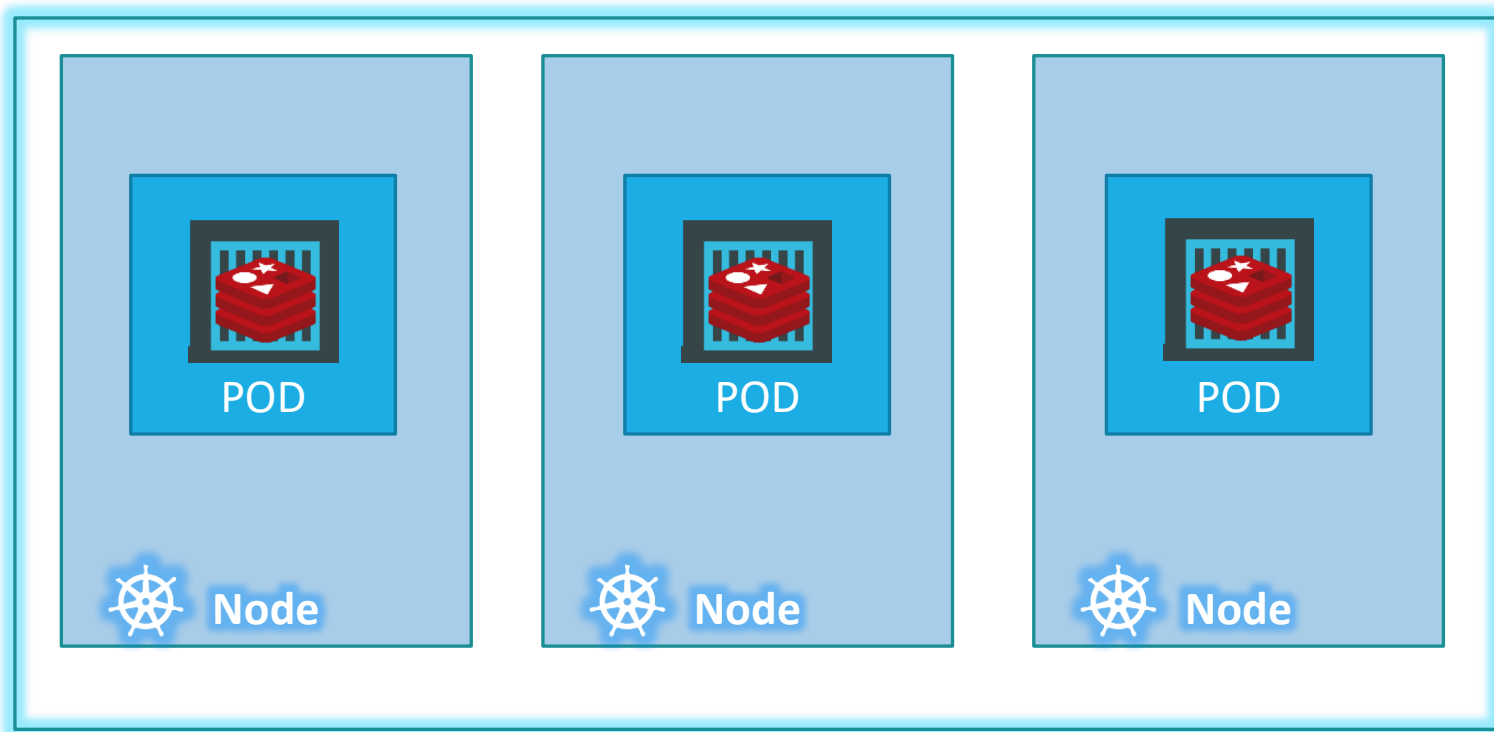
Raman

Assumptions

Docker Image

Kubernetes Cluster

POD



POD

