IIT Kharagpur | IIT Madras | IIT Goa | IIT Palakkad

Applied Accelerated Artificial Intelligence

NVIDIA

DeepOps: Deep Dive into Kubernetes with deployment of various AI based Services

Session I – Kubernetes

Satyadhyan Chickerur, PhD

Professor

School of Computer Science and Engineering

KLE Technological University

NVIDIA DLI Ambassador/Instructor

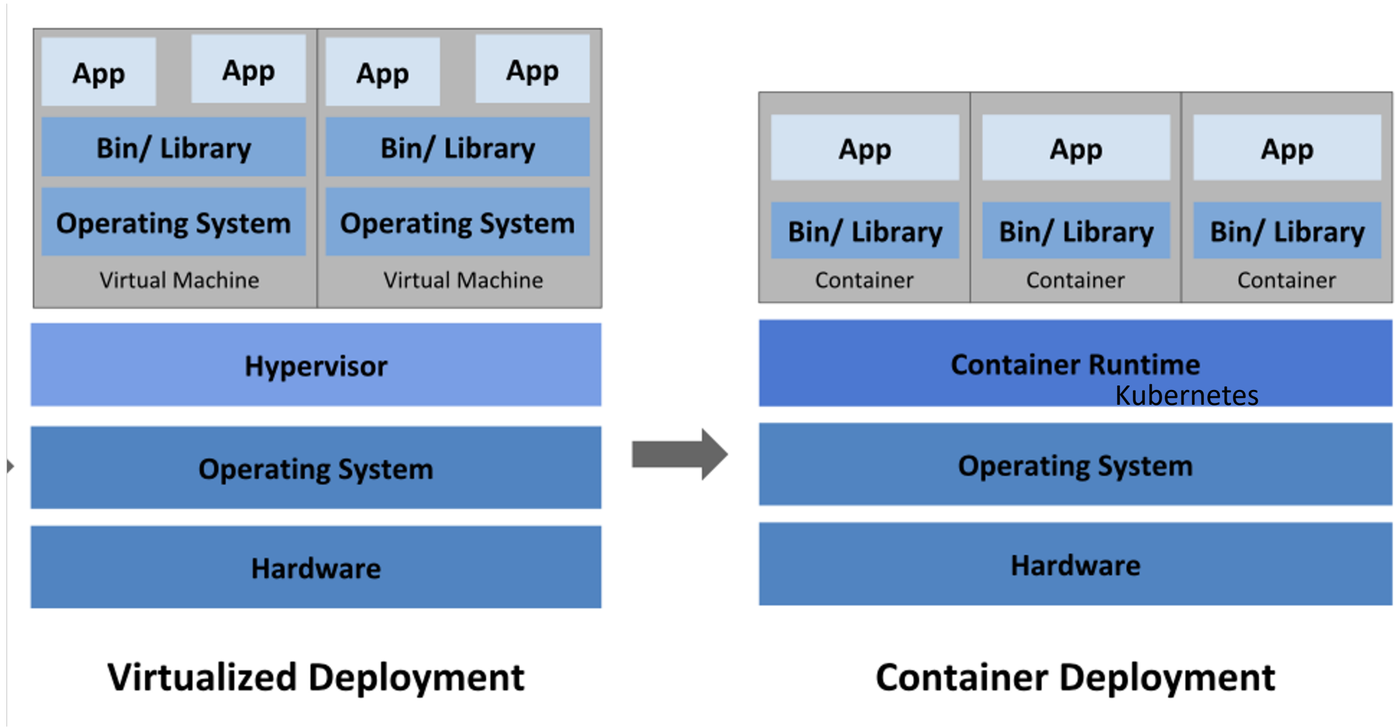
National Supercomputing Mission

Centre for Development of Advanced Computing

Agenda

- Need for Kubernetes
- Demo –
- Deployments Demo – Load balancing

Need for Kubernetes ?



- Containers are a good way to bundle applications.
- Need to manage the containers that run the applications and ensure that there is no downtime.
- if a container goes down, another container needs to start.
- Wouldn't it be easier/useful if this behavior was handled by a system?
- Kubernetes

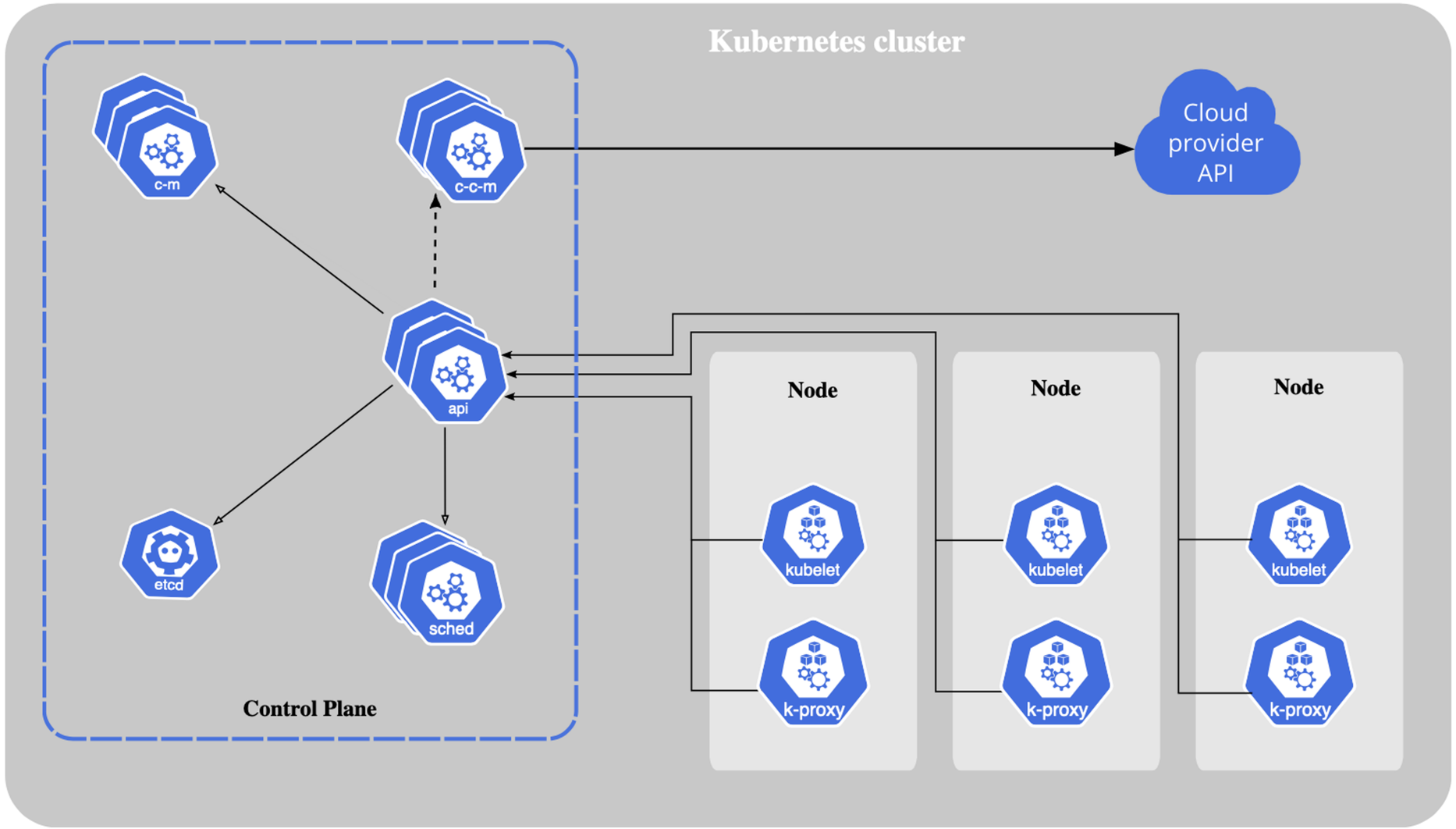
Source: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Kubernetes

- Framework to run distributed systems resiliently.
- It takes care of scaling and failover for our applications.
- Provides deployment patterns.

- Service discovery and load balancing
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Self-healing
- Secret and configuration management

Kubernetes Components



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

kube-proxy

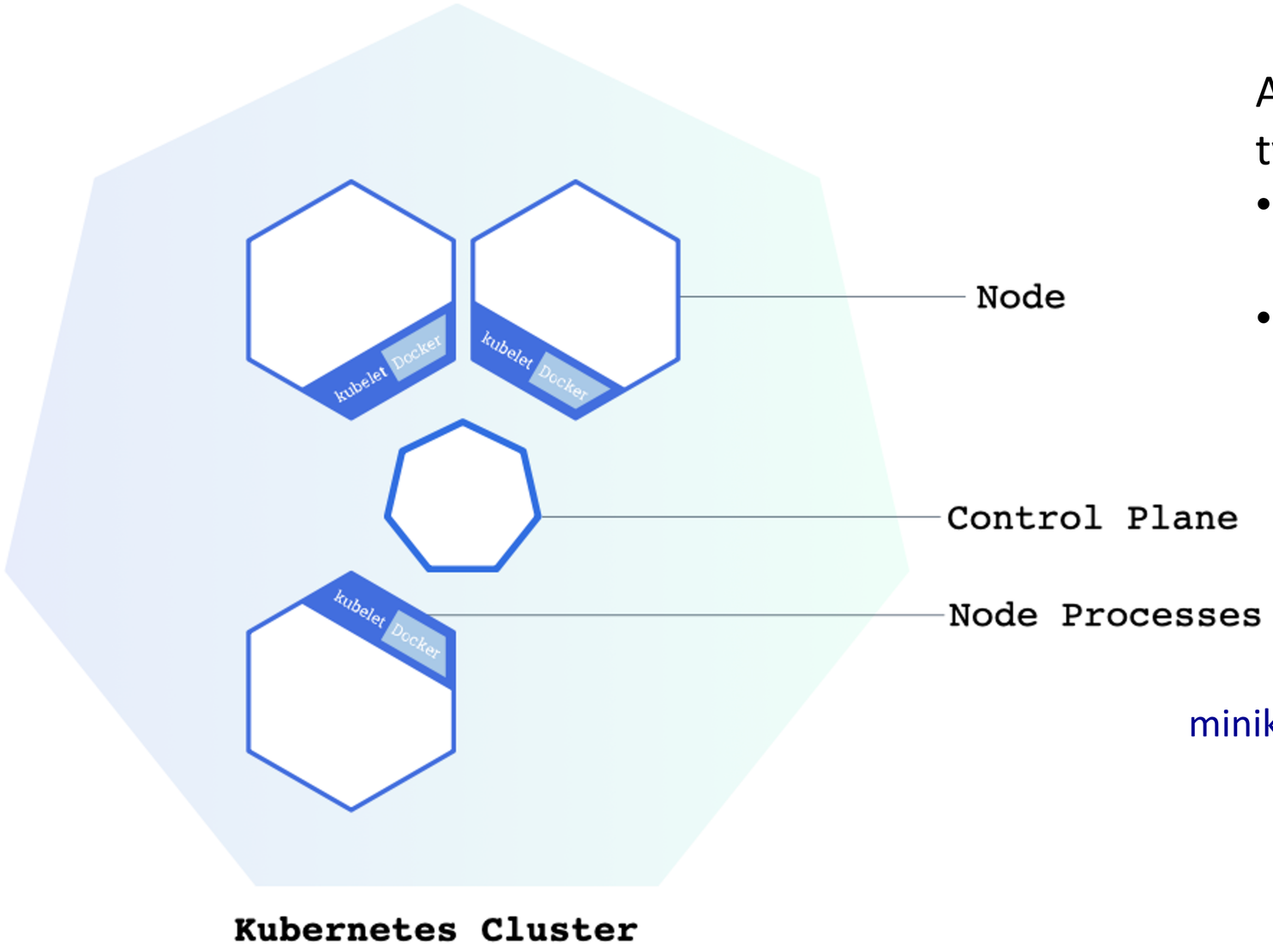
Scheduler

Control plane

Node

Source: https://kubernetes.io/docs/concepts/overview/components/

Cluster



A Kubernetes cluster consists of two types of resources:

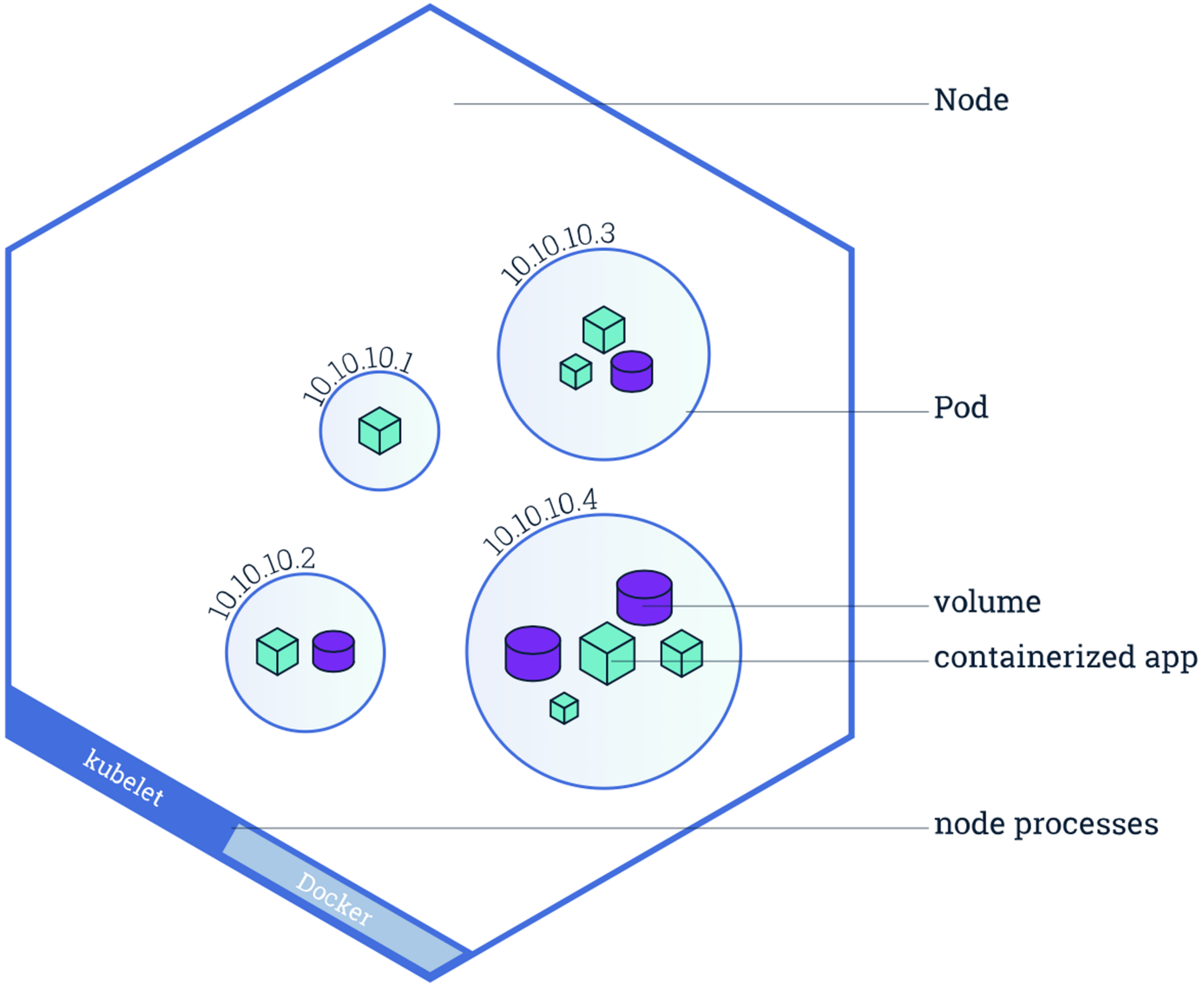
- The **Control Plane** coordinates the cluster
- **Nodes** are the workers that run applications

minikube start --nodes 3

minikube start --nodes 3 --driver=docker

Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/

Nodes



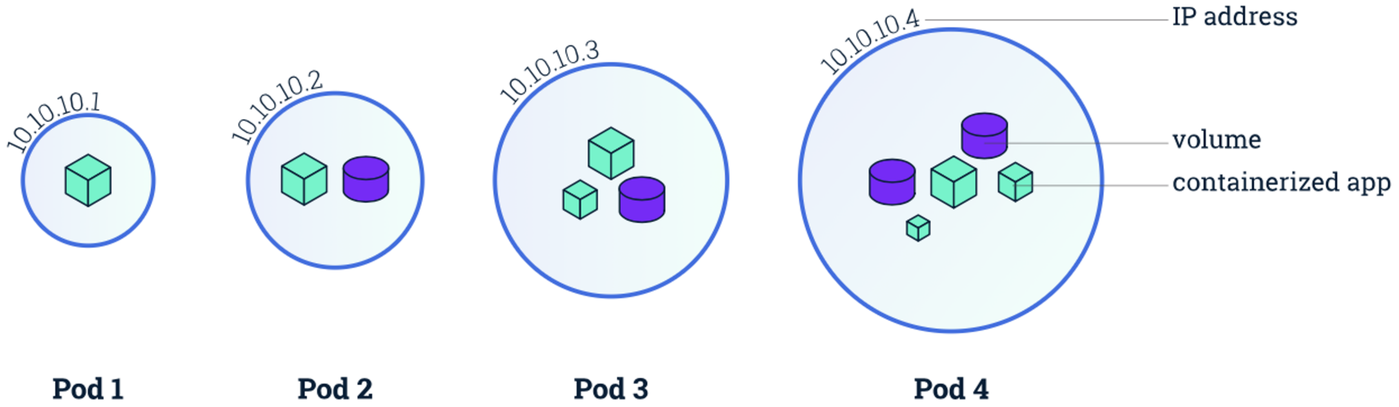
- **Nodes** are the workers that run applications.
- The components on a node include **the kubelet, a container runtime, and the kube-proxy.**

kubectl get nodes

docker ps

Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/

Pods



- A pod is **the smallest execution unit in Kubernetes.**
- A Pod represents a single instance of a running process in your cluster.
- Pods are ephemeral.

kubectl create deployment demo --image=nginx --replicas=6

kubectl get pods -o wide

Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/

Demo of Pod crash and Load Balancing

- Now we can simulate a crash on one of the nodes by stopping one of the worker nodes, since the docker containers have it's node name as the container name

```
docker stop minikube-m03
```

```
kubectl get nodes
```

- with **kubectl get nodes** we will be able to see that the cluster realizes that the node it not longer in **Ready** state (This will take few minutes)

```
kubectl get pods -o wide
```

Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/

Thank You