

Day 23: Introduction to Kubernetes

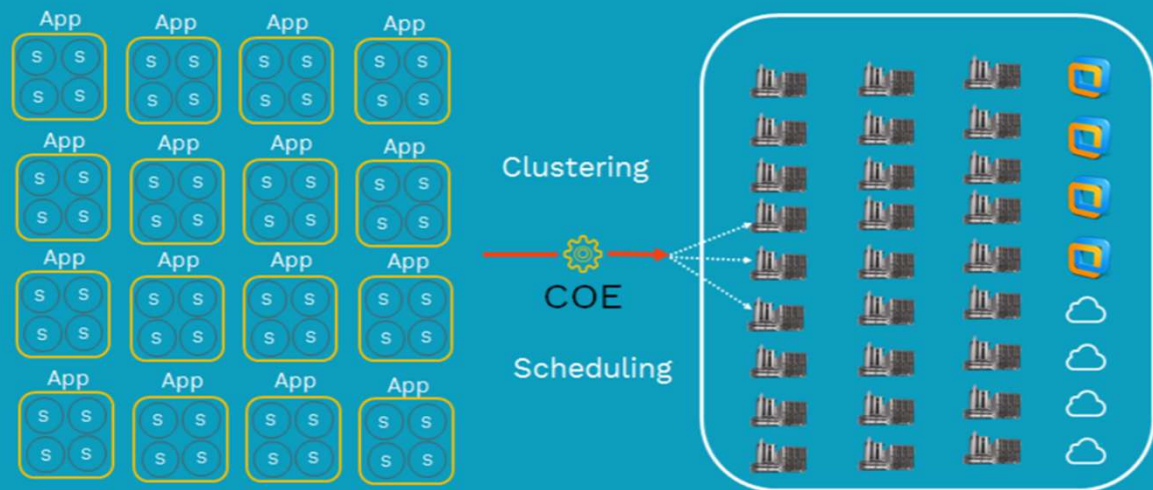
Presented By Mohd Shahid

What is Kubernetes?

- Open-source container orchestration platform.
- Automates deployment, scaling, and operations of application containers.
- Created by Google, donated to CNCF.
- Inspired by Google's internal Borg system.
- Written on GO.

COE Features

- Clustering
- Scheduling
- Scalability
- Load balancing
- Fault tolerance
- Deployment



Why Kubernetes?

- Simplifies container management at scale.
- Automates deployment and self-healing.
- Supports load balancing and rolling updates.
- Works across cloud providers and hybrid environments.

Kubernetes Architecture

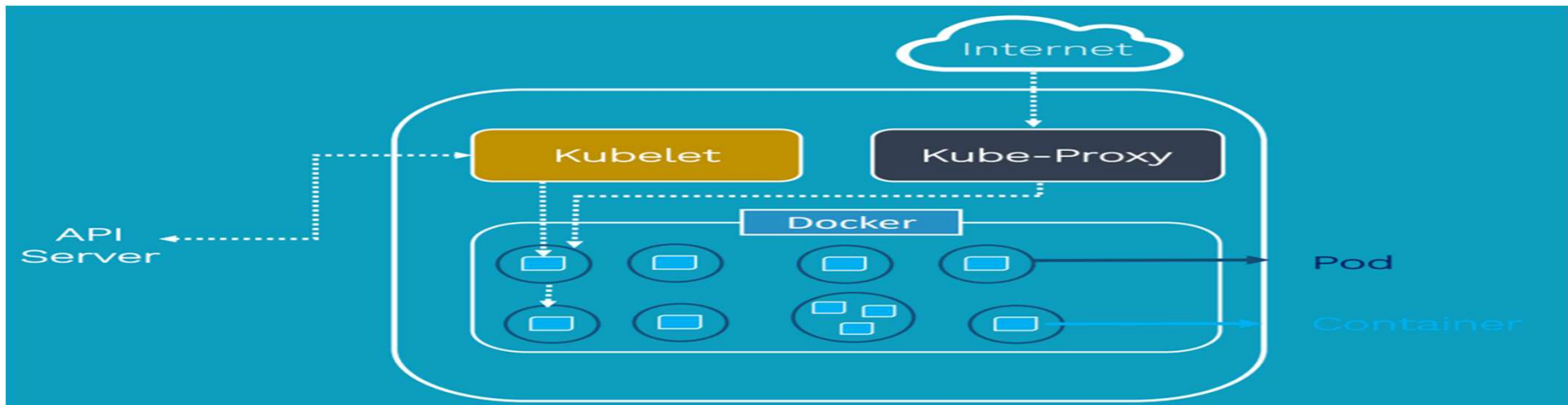
- Follows Master-Worker node architecture.
- Master components manage cluster state and decisions.
- Worker nodes run application containers (Pods).

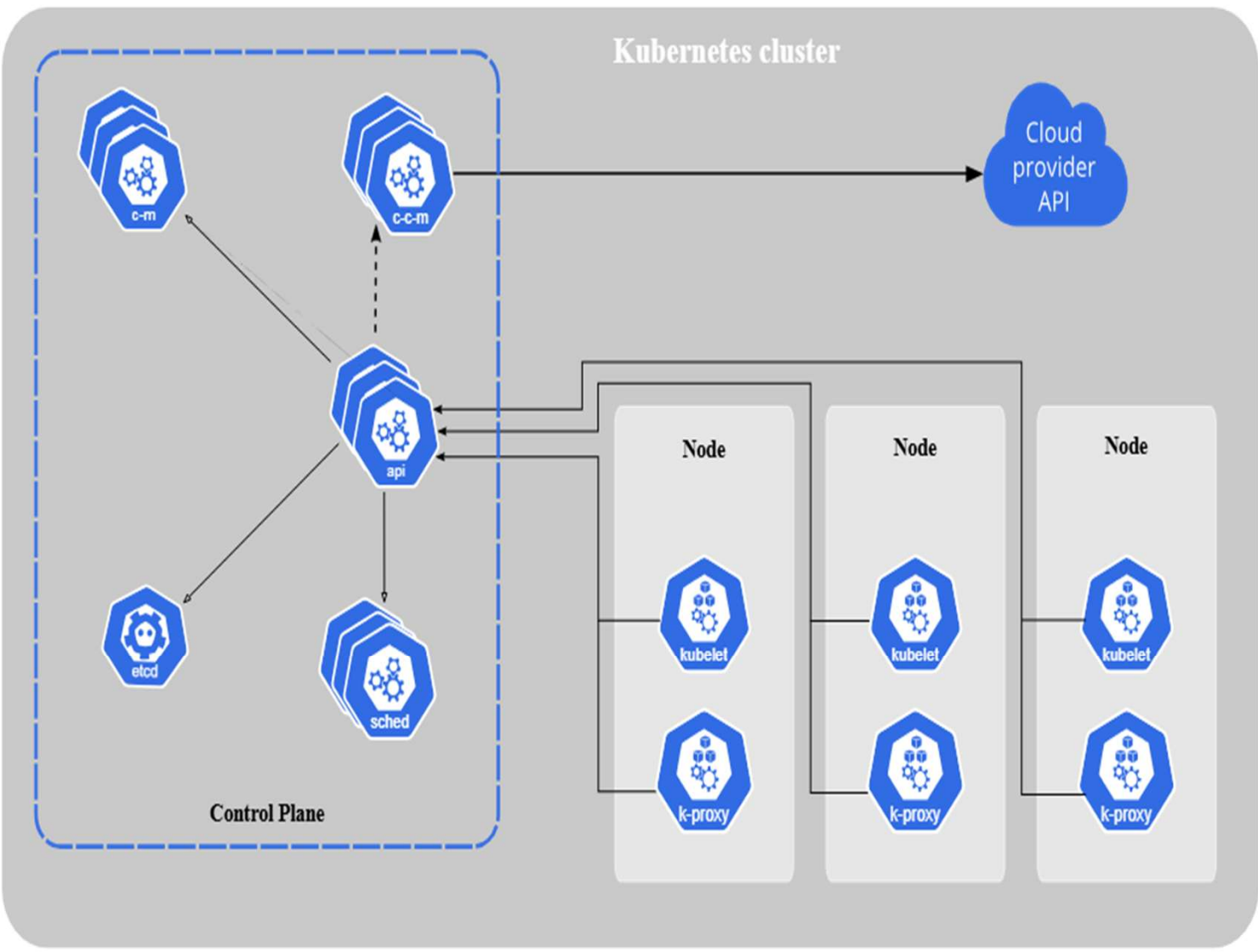
Master Node Components










- API Server: Central management point for the cluster.
- etcd: Distributed key-value store for config/state.
- Scheduler: Assigns pods to appropriate nodes.
- Controller Manager: Ensures desired cluster state.
- Cloud Controller Manager: Cloud-provider integrations.

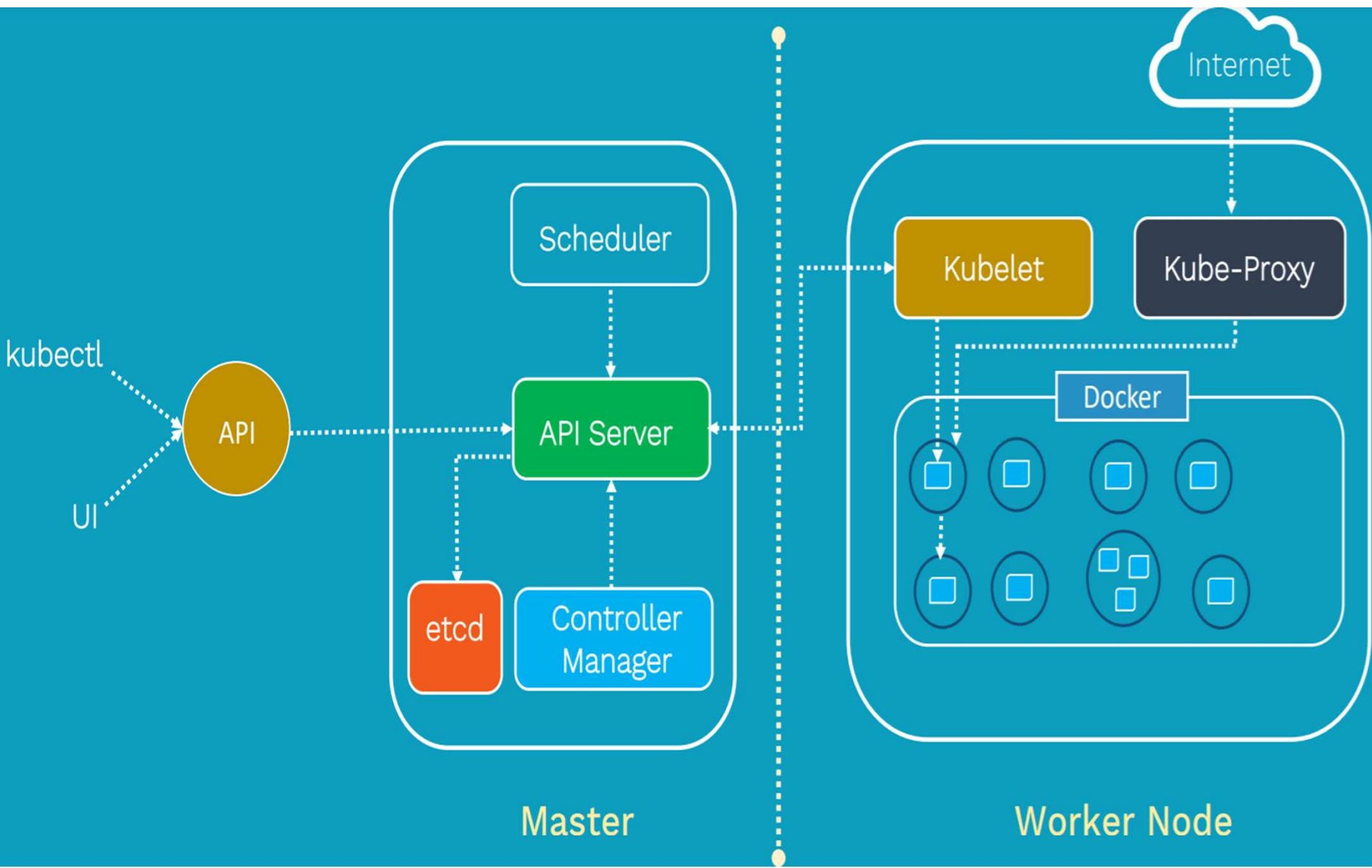
Worker Node Components

- Kubelet: Ensures containers in Pods are running.
- Kube-proxy: Manages network rules/load-balancing.
- Container Runtime: Docker, containerd, etc. run containers.





- API server 
- Cloud controller manager (optional) 
- Controller manager 
- etcd (persistence store) 
- kubelet 
- kube-proxy 
- Scheduler 
- Control plane 
- Node 



Key Concept: Pod

- Smallest deployable unit in Kubernetes.
- Encapsulates one or more containers.
- Each pod gets its own network IP.
- Pods are ephemeral and managed by controllers.

Kubernetes Use Cases

- Run microservices with isolated environments.
- Automate CI/CD and deployments.
- Enable blue/green and canary deployments.
- Auto-healing and auto-scaling infrastructure.

When to Use / Not Use Kubernetes

- Use: Large-scale microservices, auto-scaling needs.
- Use: Hybrid/multi-cloud architectures.
- Not Use: Simple or small team setups.
- Not Use: Limited compute/budget environments.

Summary

- Kubernetes is essential for modern DevOps at scale.
- Abstracts complexity of container orchestration.
- Provides extensible, declarative infrastructure.
- Supports resilience, scalability, and automation.