WALCHAND COLLEGE OF ENGINEERING, SANGLI

WALCHAND LINUX USERS' GROUP

WLUG
Community | Knowledge | Share

METAMORPHOSIS

Navigating the DevOps Wave

LIMITED SEATS

EXCITING PRIZES

15 & 16 FEB

Let's Dock It!
SESSION 1

DockerVerse
SESSION 2

REGISTER AT

Sailing to K8s
SESSION 3

Pod Power
SESSION 4

WARGAMES

POWERED BY
IT'S FOSS

CONTACT WITH US

₹349 | ₹299 each

Main & Mini CCF

94220 17108
82084 32514

Mr. Yash Patil	Dr. M. A. Shah	Dr. R. R. Rathod	Dr. A. J. Umbarkar	Dr. A. R. Surve	Dr. M. M. Khot	Dr. U. A. Dabade
President WLUG	HoD Computer Science and Engineering	HoD Information Technology	Staff Advisor WLUG	Staff Advisor WLUG	Associate Dean WCE	I/C Director WCE

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CONTENTS

- Distributed systems
- Microservices & REST APIs
- What is Kubernetes?
- Architecture of K8S
- Docker vs Kubernetes
- Kubernetes Installation

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker


PEOPLE
KUBERNETES
....
STORAGE
NETWORKING
DOCKER
YAML
LINUX

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DISTRIBUTED SYSTEMS

Collection of independent computers which appear to be working as single coherent system

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CHARACTERISTICS

- Resource sharing
- Scalability
- Parallelism
- Fault tolerance

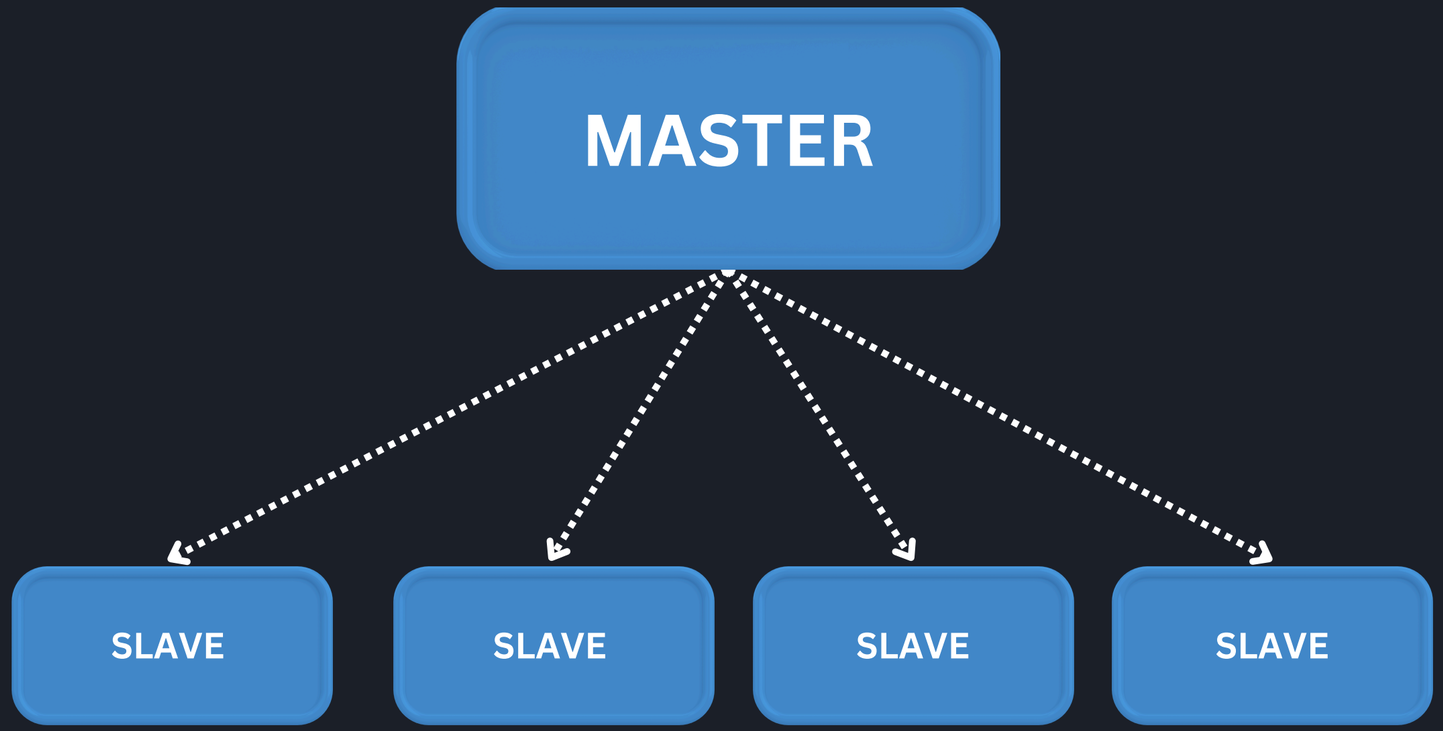
WLUG
COMMUNITY | KNOWLEDGE | SHARE

DISTRIBUTED SYSTEMS

Blade Servers

WLUG
COMMUNITY | KNOWLEDGE | SHARE

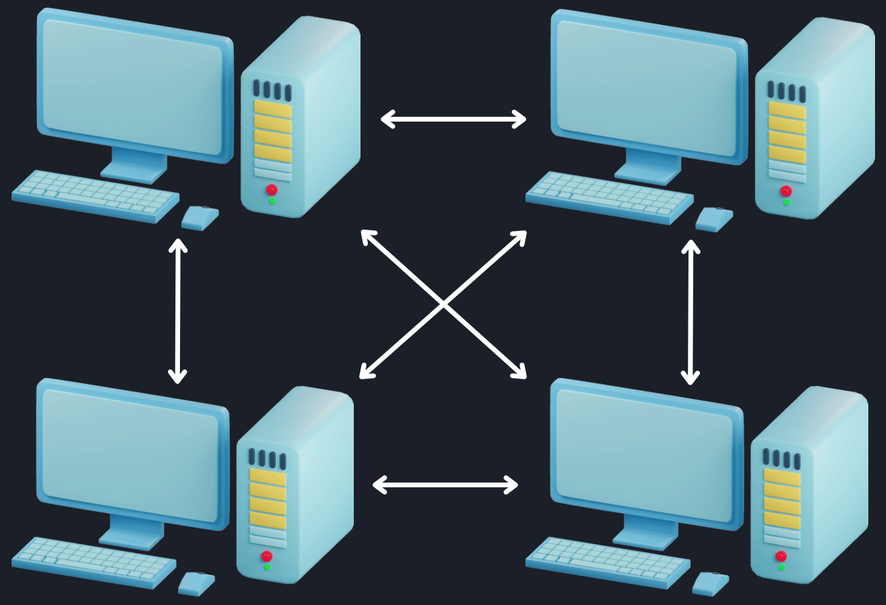
DISTRIBUTED ALGORITHMS



Master Slave Architecture

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DISTRIBUTED ALGORITHMS



Peer-to-Peer Architecture

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MICROSERVICES

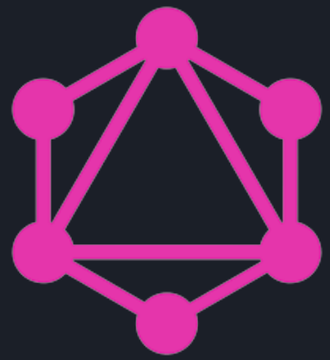
- Implementation of distributed systems
- An application built up of different parts
- Use API for communication
- Broken into microservices
- Decentralized managemnet
- Example: Various Restaurant services

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WEB API

A standard protocol for communication between two devices over a network

REST



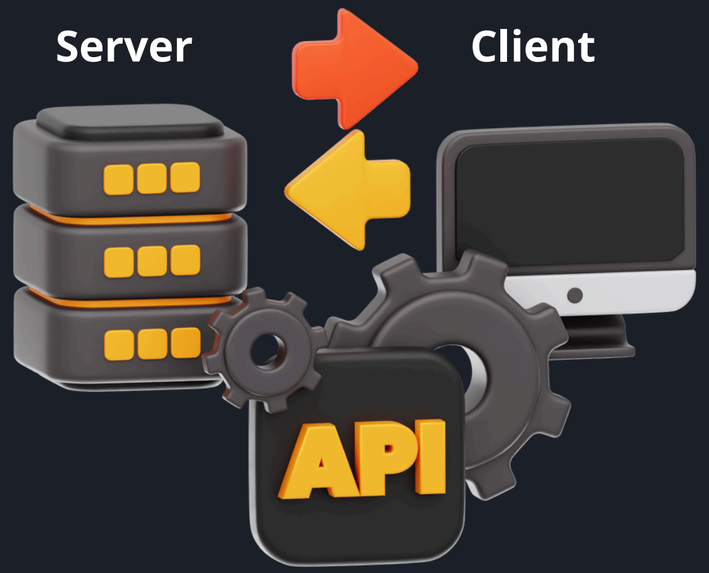
GraphQL

Sockets

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CLIENT-SERVER ARCHITECTURE

- The party requesting specific service is client
- The party serving the request and making a response is server
- Back and forth communication



WLUG
COMMUNITY | KNOWLEDGE | SHARE

CLIENT-SERVER ARCHITECTURE

Database Server

Login Server

User

WLUG
COMMUNITY | KNOWLEDGE | SHARE

BENEFITS OF MICROSERVICES

WLUG
COMMUNITY | KNOWLEDGE | SHARE

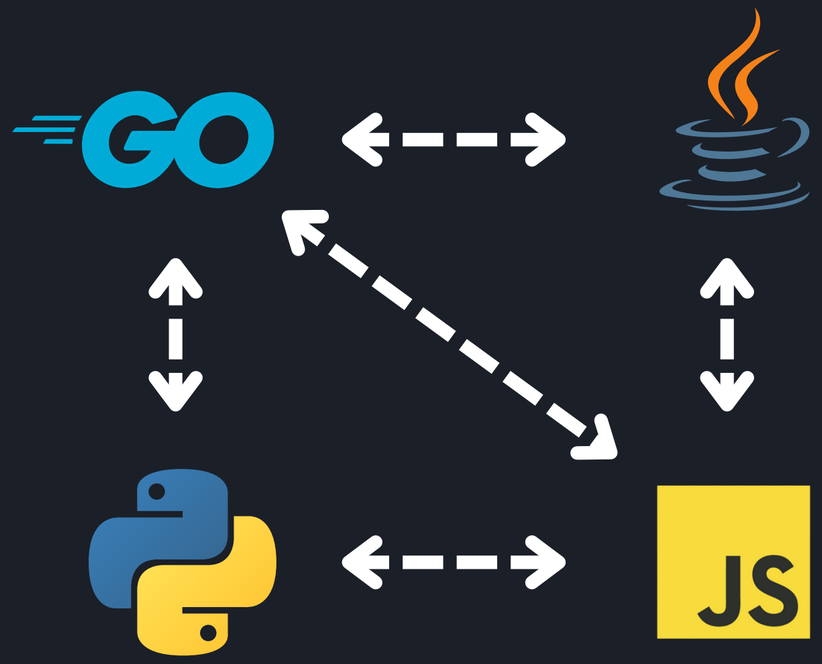
SCALING

Easily scale up and allocate available resources across any service

WLUG
COMMUNITY | KNOWLEDGE | SHARE

FLEXIBILITY

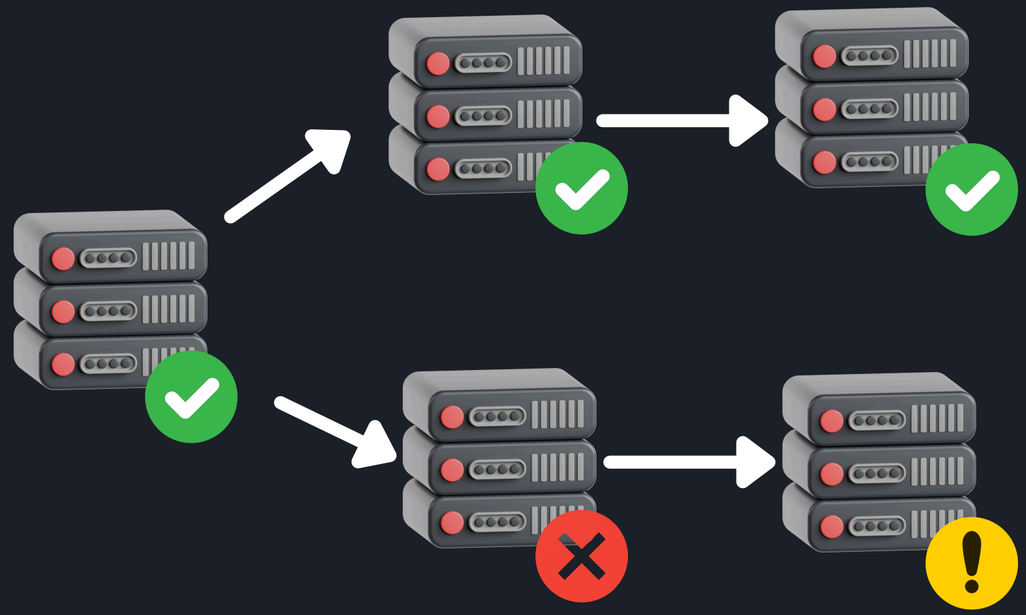
Independent development, testing and deployment of individual services



WLUG
COMMUNITY | KNOWLEDGE | SHARE

FAULT ISOLATION

Service failure only affects dependent services, while others remain unaffected



WLUG
COMMUNITY | KNOWLEDGE | SHARE

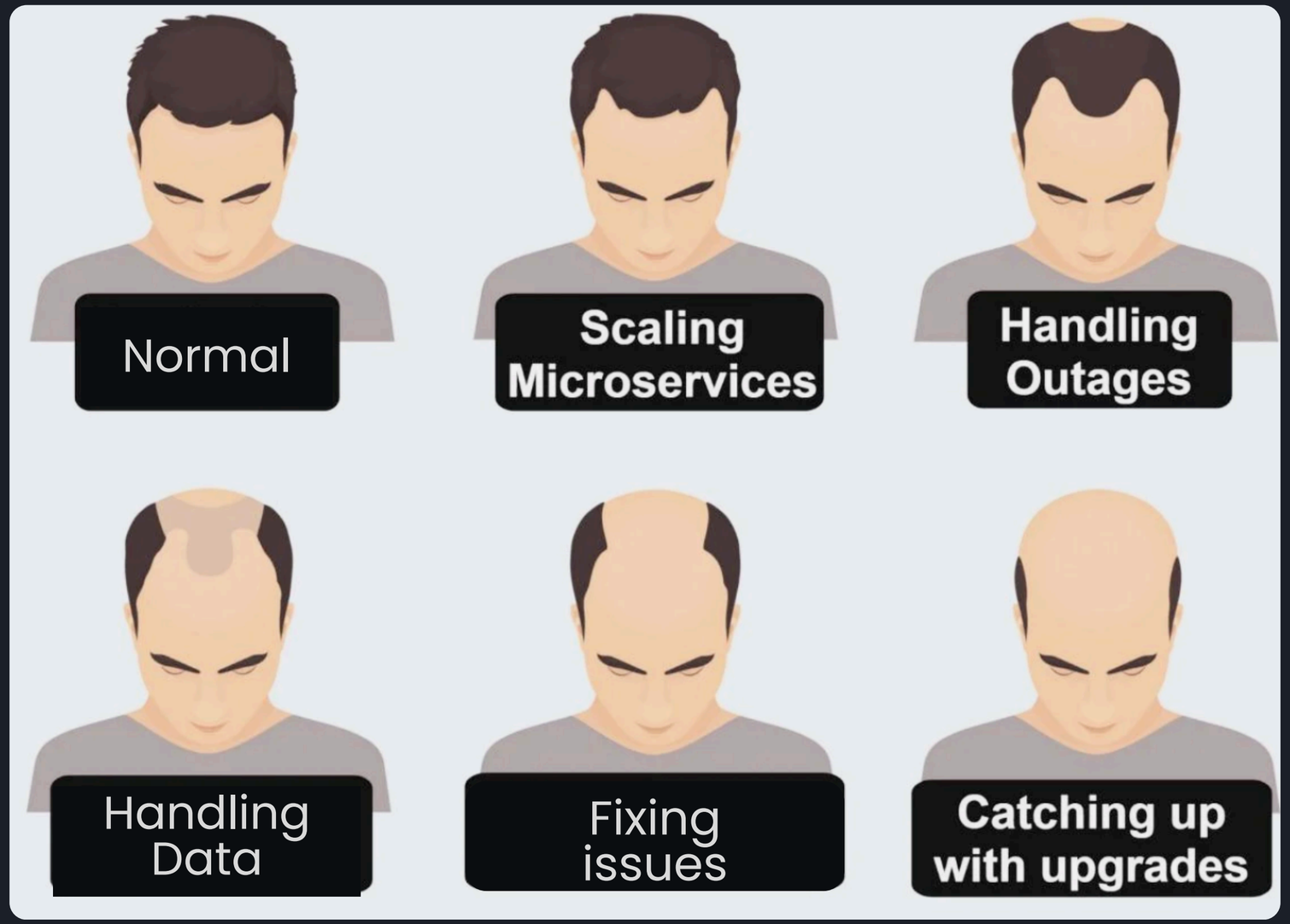
FASTER DEVELOPMENT AND DEPLOYMENT

Services can be developed, tested and deployed parallel

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CHALLENGES

- Distributing work between same service is difficult
- Implementing fault tolerance and manually tracking services is time consuming
- Updating and deploying new service to integrate with current can be tricky

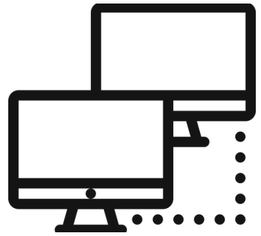

Normal
Scaling Microservices
Handling Outages
Handling Data
Fixing issues
Catching up with upgrades

WLUG
COMMUNITY | KNOWLEDGE | SHARE

NEED OF A MANAGER

Kubernetes acts as a comprehensive management tool
It ensures the following:

Automated Operations



Load Balancing

Resource Optimization

WLUG
COMMUNITY | KNOWLEDGE | SHARE

KUBERNETES

- Open Source container orchestration software
- Originally developed by Google in 2014
- Now maintained by the Cloud Native Computing Foundation (CNCF)
- Written in the GO programming language

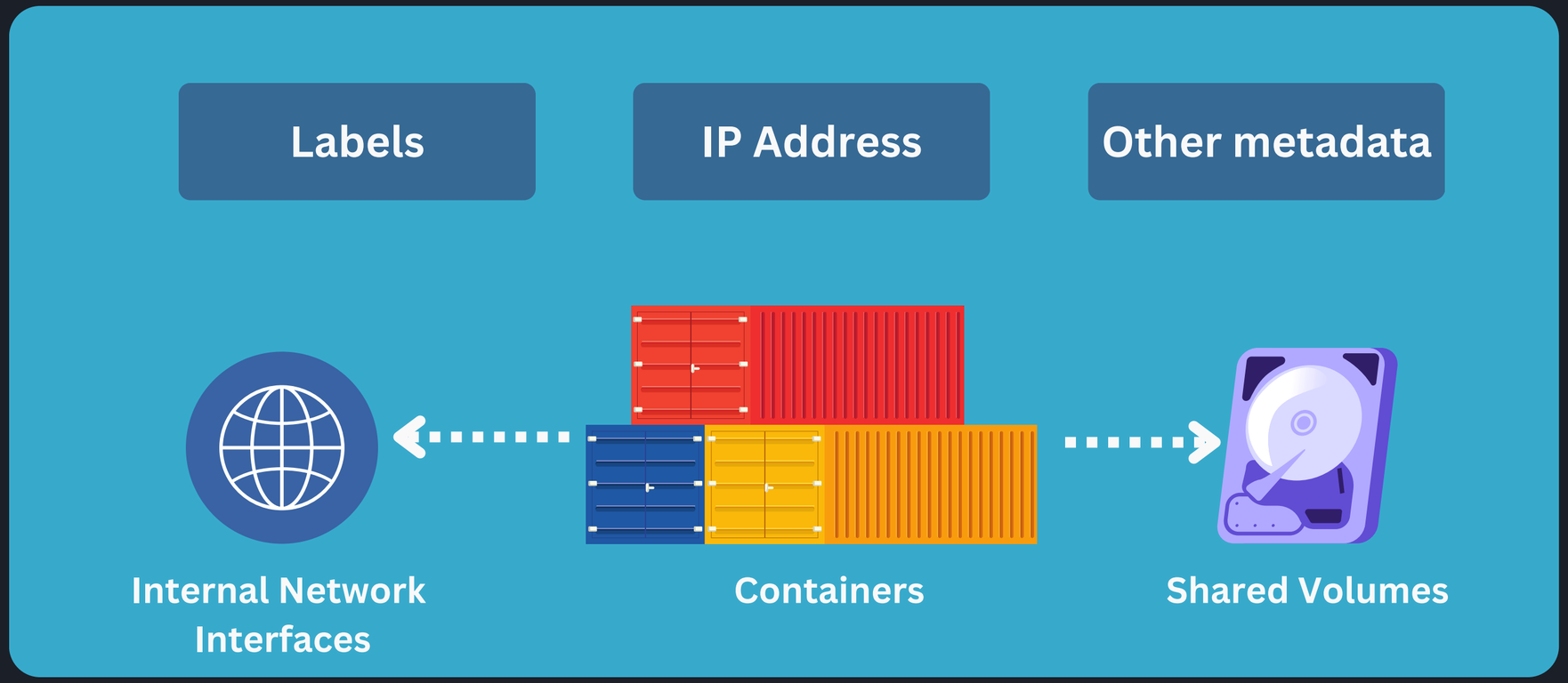
WLUG
COMMUNITY | KNOWLEDGE | SHARE

PODS

- Smallest, most basic deployable unit in Kubernetes
- Represents a single instance of a running process in your application
- Can contain one or more tightly coupled containers
- Containers within a pod share the same network namespace and storage volumes

WLUG
COMMUNITY | KNOWLEDGE | SHARE

PODS



A Pod

WLUG
COMMUNITY | KNOWLEDGE | SHARE

NODES

- Actual computer that run applications which are the pods
- Each node contains the necessary components to manage pods
- A node can be a virtual or physical machine
- Includes a container runtime (like Docker), kubelet and kube-proxy

WLUG
COMMUNITY | KNOWLEDGE | SHARE

NODES

Container Runtime

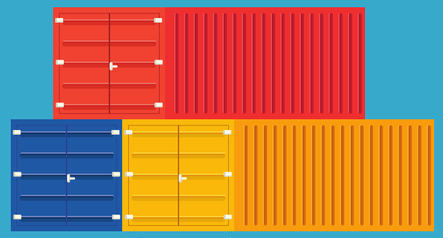
Kubelet

Kube Proxy

POD

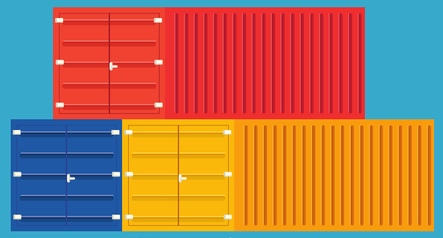
Containers

POD



Containers

POD



Containers

A Kubernetes Node

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CLUSTER

- Set of Nodes managed as a single entity by Kubernetes
- Consists of a control plane that manages worker nodes
- Allows seamless orchestration of applications
- High availability and fault tolerance

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CLUSTER

CONTROL PLANE NODE

WORKER NODE

Components

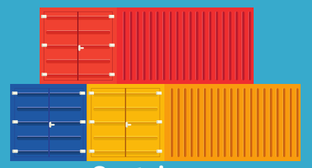
POD

Containers

WORKER NODE

Components

POD



Containers

WORKER NODE

Components

POD

Containers

WORKER NODE

Components

POD

Containers

WLUG
COMMUNITY | KNOWLEDGE | SHARE

USE CASES

- Cloud-Native Apps
- Multi-Cloud Environments
- Continuous Updates
- Scalable Services

When the person with Kubernetes experience is brought into a room...

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHY KUBES?

Kubernetes is like a smart manager, automating tasks to keep everything running smoothly

WLUG
COMMUNITY | KNOWLEDGE | SHARE

PROPERTIES

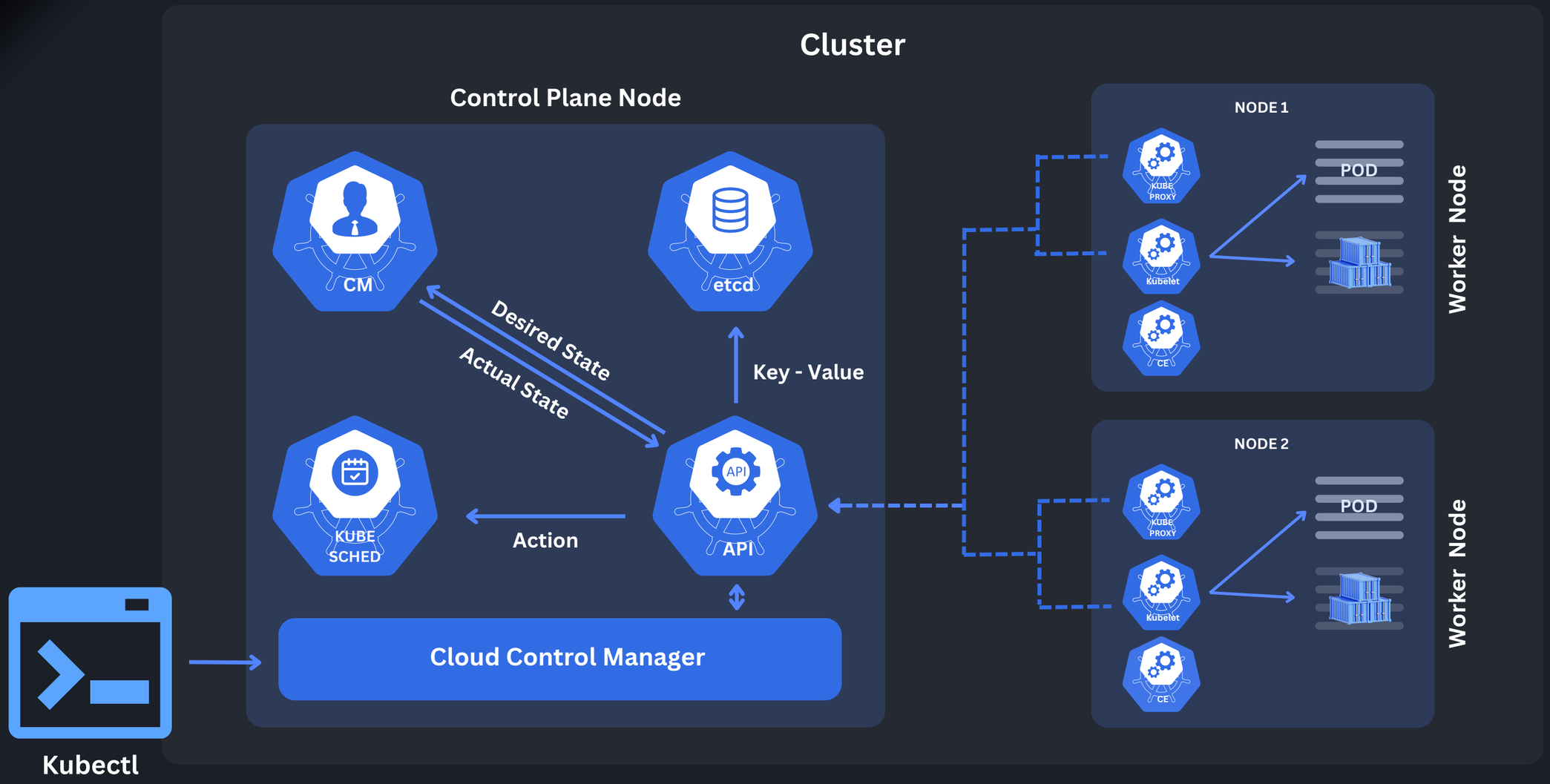
- Self-Healing
- Automated Updates
- Load Balancing
- Efficient Use of Resources

WLUG
COMMUNITY | KNOWLEDGE | SHARE

K8S

ARCHITECTURE

- Master-Slave Architecture
- Control Plane Components
- Worker Node Components



WLUG
COMMUNITY | KNOWLEDGE | SHARE

HEADQUARTERS

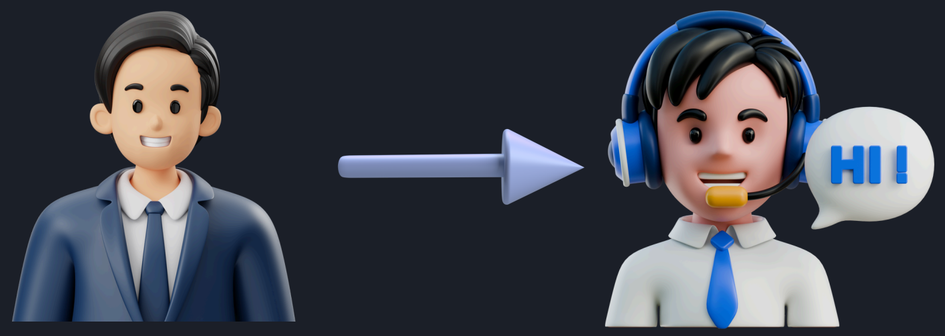
Branch Office 1

Branch Office 2

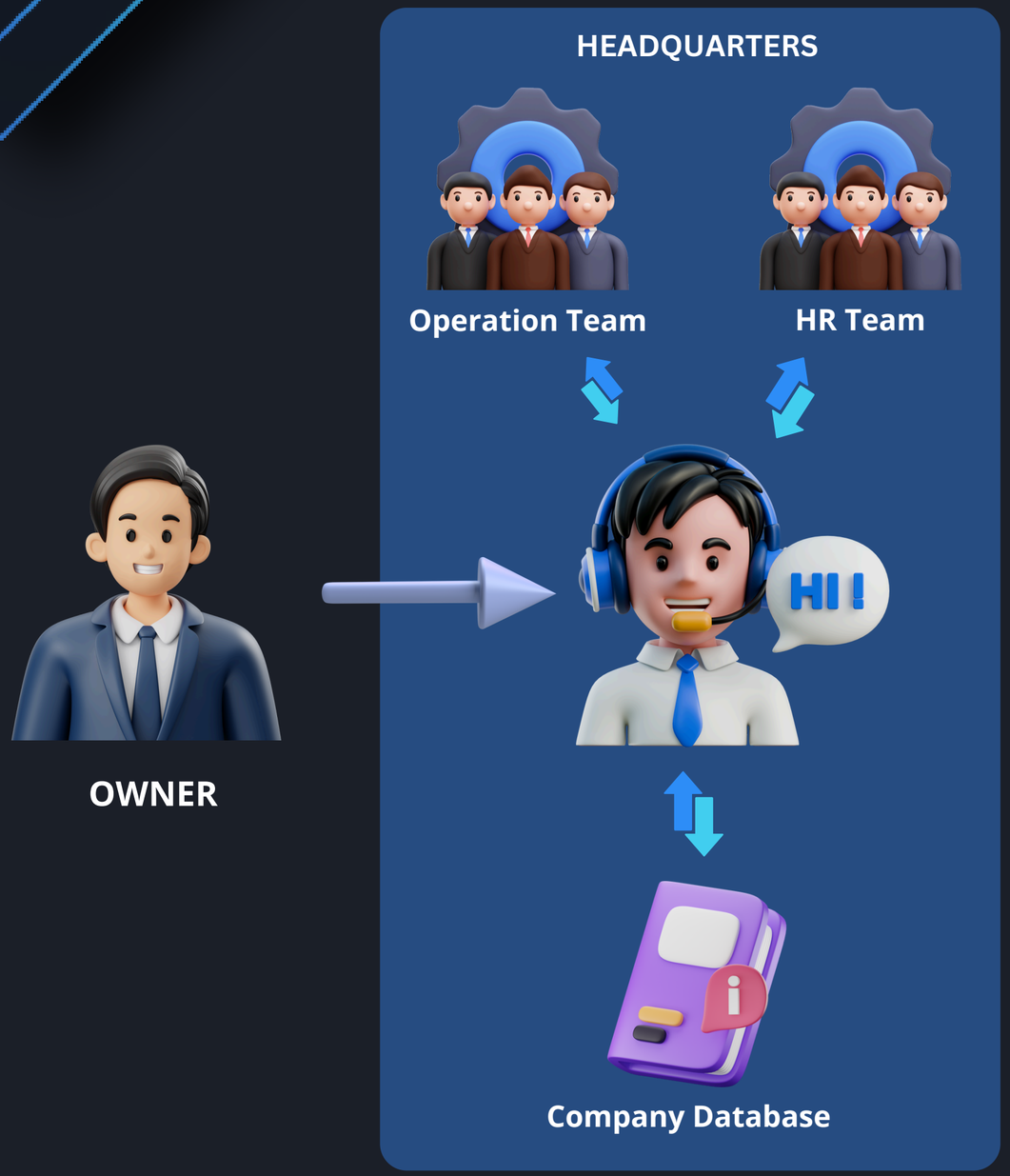
Branch Office 3

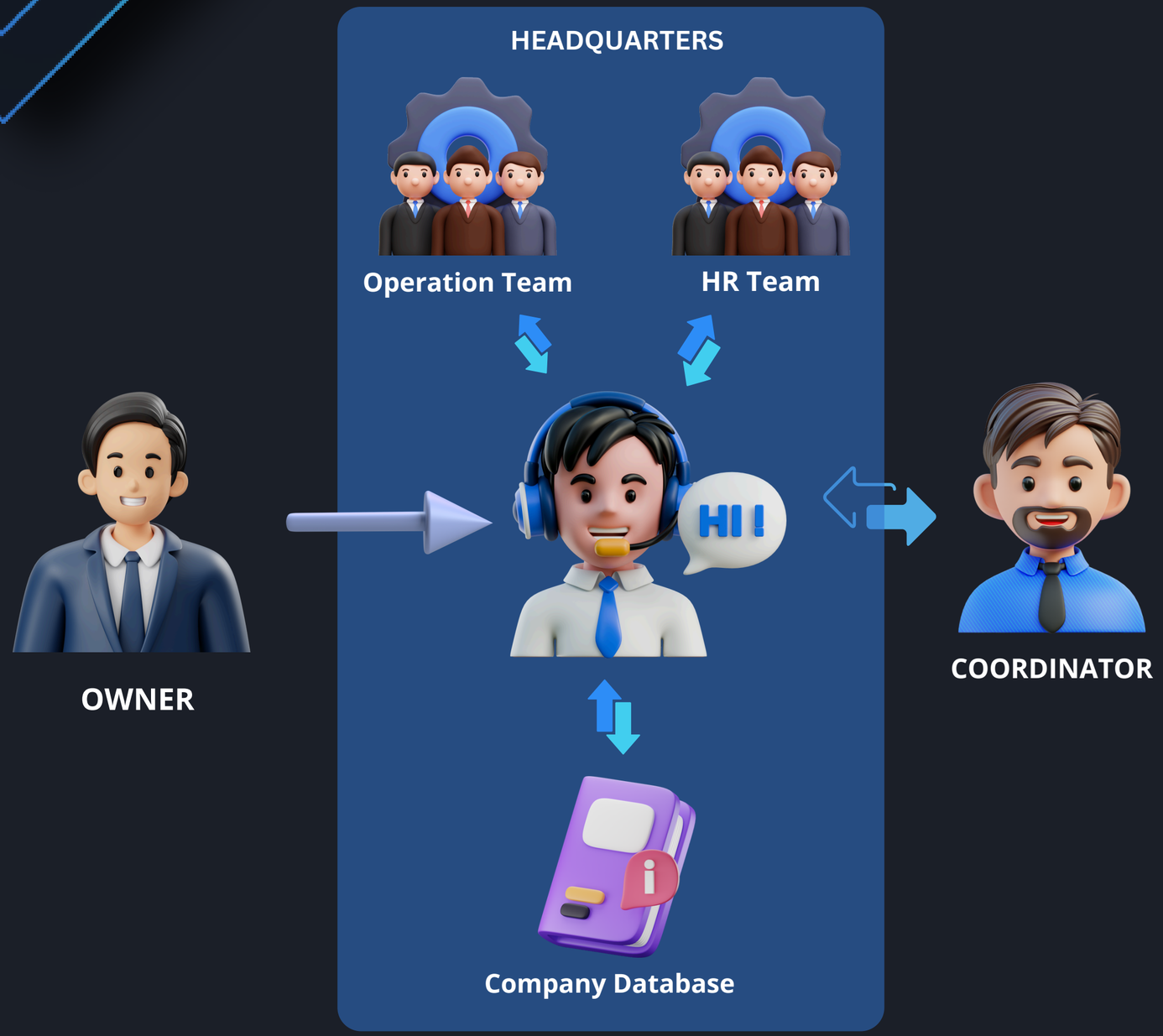
OWNER

WLUG
COMMUNITY | KNOWLEDGE | SHARE



OWNER


HEADQUARTERS
Operation Team
HR Team
HI !
OWNER
Company Database


HEADQUARTERS
Operation Team
HR Team
HI !
OWNER
COORDINATOR
Company Database

WLUG
COMMUNITY | KNOWLEDGE | SHARE

HEADQUARTERS

Operation Team

HR Team

OWNER

Company Database

COORDINATOR

Branch Office 1

Manager

Branch Office 2

Manager

Branch Office 3

Manager

CONTROL PLANE NODE

- Brain of a Kubernetes clusters
- Manages the overall cluster and worker nodes
- Api Server, ETCD, Controller Manager, Kube Scheduler
- Manages scheduling, scaling and maintaining the application's health

WORKER NODES

- Executes containerized application and manages Pods
- Handles tasks assigned by the Control Plane Node
- Kubelet, KubeProxy, Container Engine
- User workloads run on Worker Nodes

Manages

Control Plane Node

Worker Nodes

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Cluster

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Cluster

?

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Cluster

Control Plane Node

CONTROL PLANE
NODE

API SERVER

Entry Point to K8S cluster

- Handles communication between the user and the control plane
- Entry/Access point to K8S cluster
- Authenticates, validates and routes requests within the cluster

CONTROL PLANE
NODE

CONTROLLER MANAGER

Keeps track of actual and desired state

- Runs a control loop to monitor the current state of the cluster
- Compares the current state with the desired state and makes changes to align the two
- Maintains cluster's health and performance

CONTROL PLANE
NODE

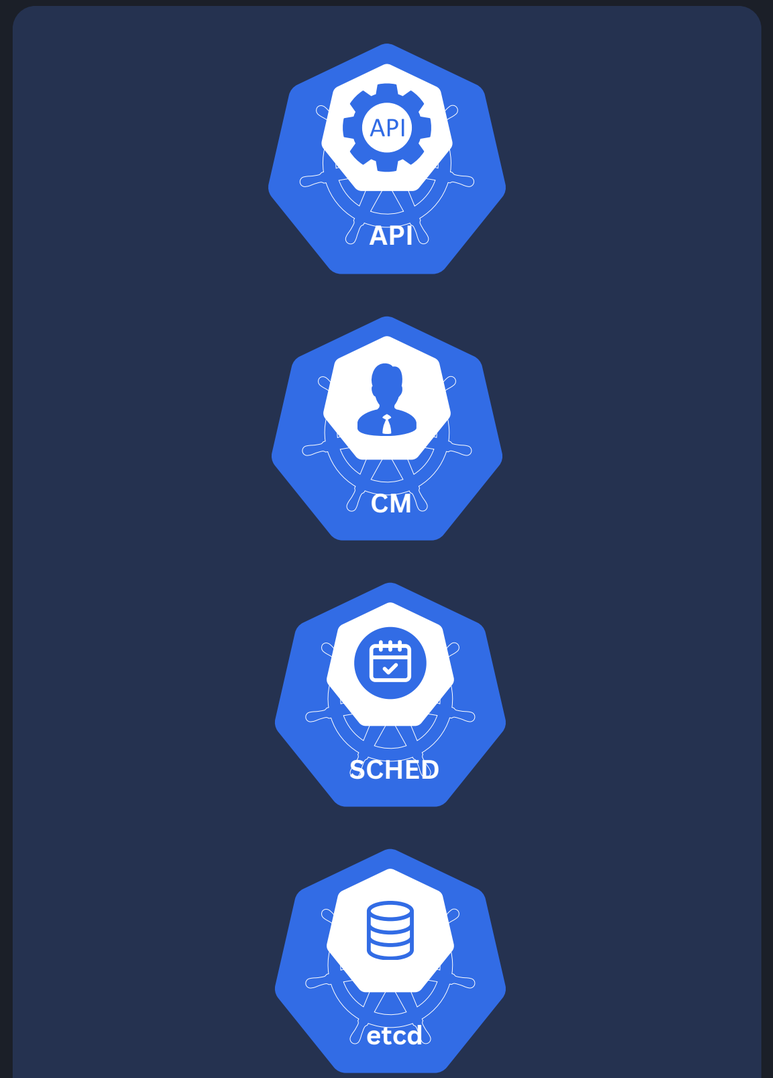
KUBE SCHEDULER
performs action based on request

- Finds the "best" machine (node) to deploy the pod
- Prefers machines where the already-running pods are consuming the least resources
- Ensures pods are spread out across nodes, avoiding resource hotspots and improving cluster balance

CONTROL PLANE
NODE

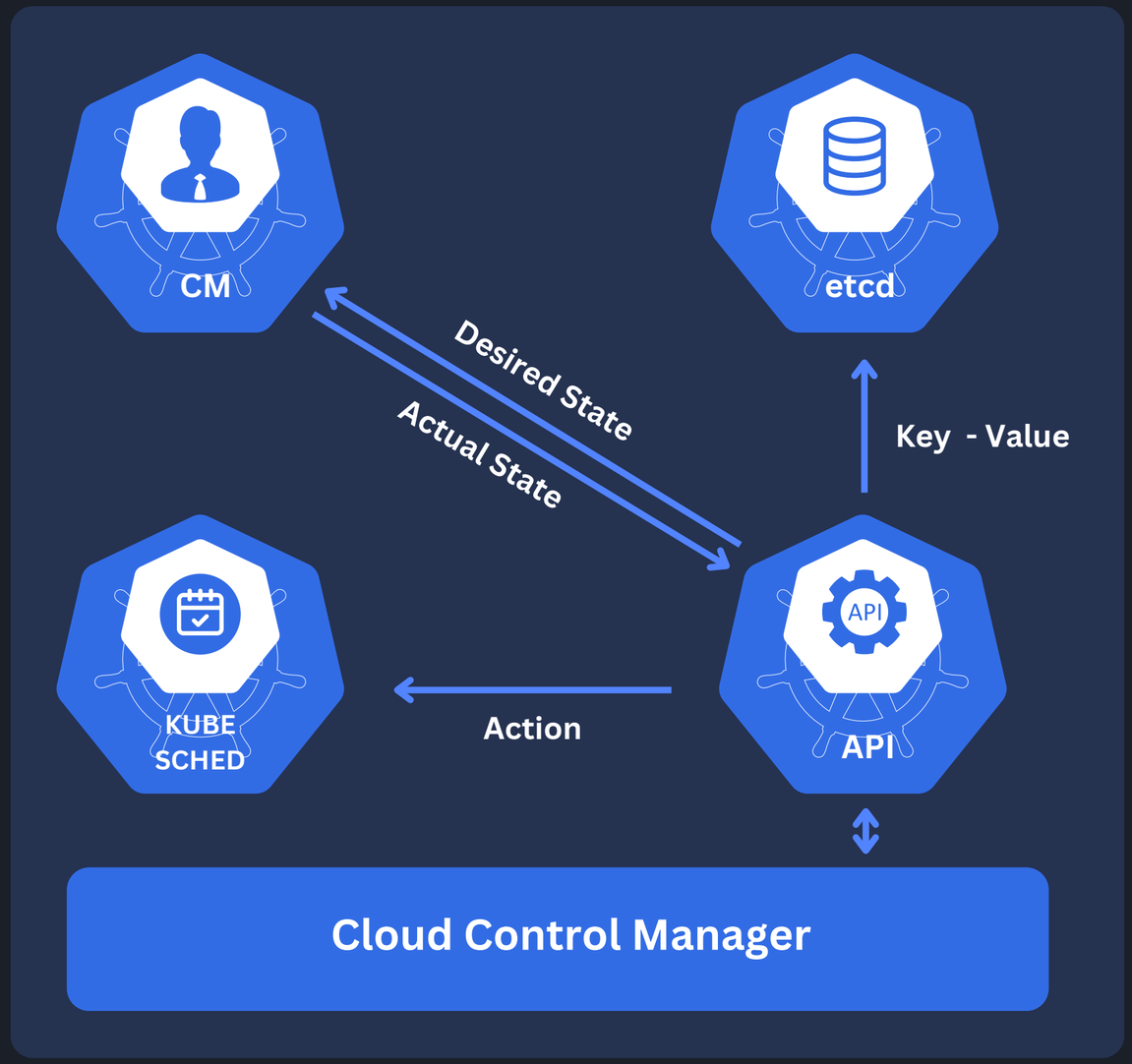
etcd
Kubernetes storage

- Distributed key-value store
- Stores metadata, status of cluster and configuration data
- Consistent and highly-available in nature

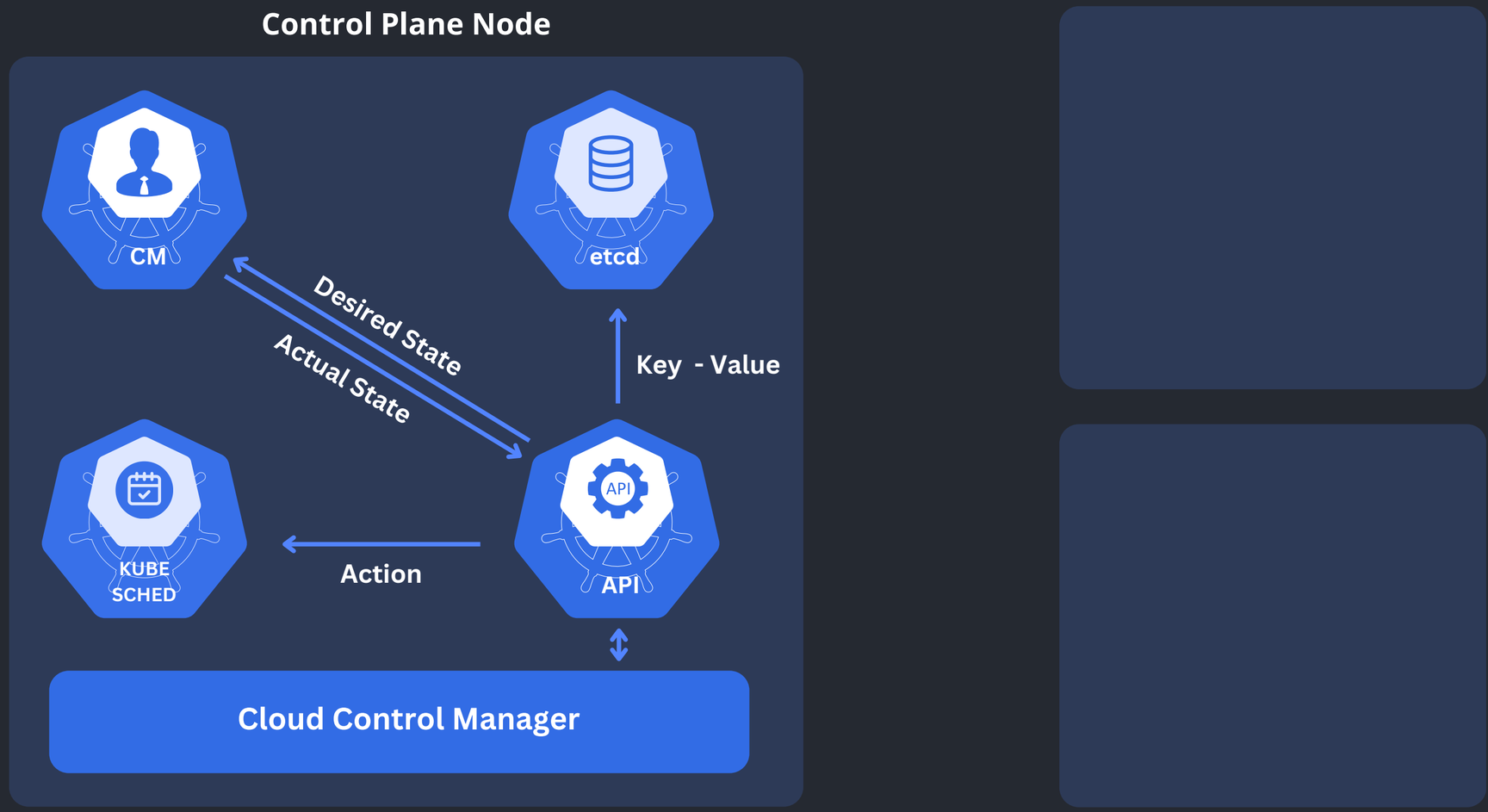


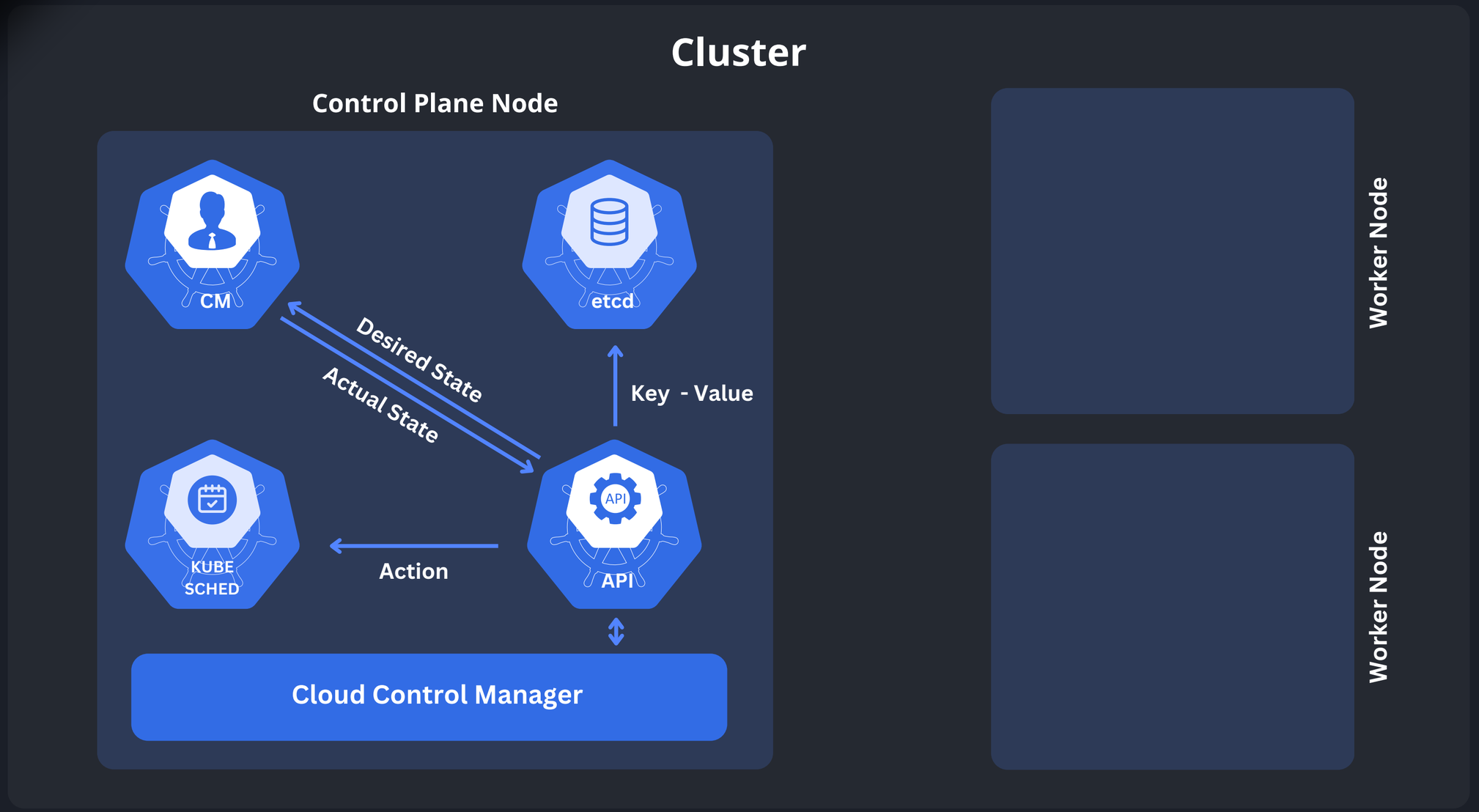
CONTROL PLANE
NODE

Control Plane Node



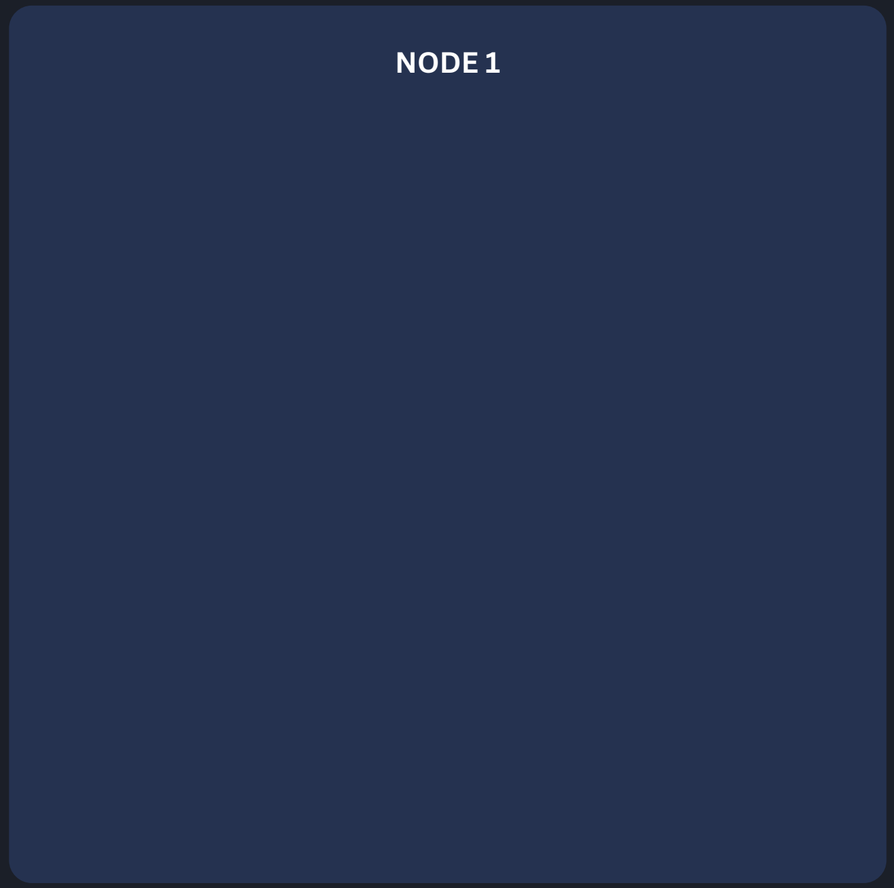
Cluster


Control Plane Node
CM
etcd
Desired State
Actual State
Key - Value
KUBE
SCHED
Action
API
API
Cloud Control Manager
2
2


Cluster
Control Plane Node
CM
etcd
Desired State
Actual State
Key - Value
KUBE
SCHED
Action
API
API
Cloud Control Manager
Worker Node
Worker Node

NODE

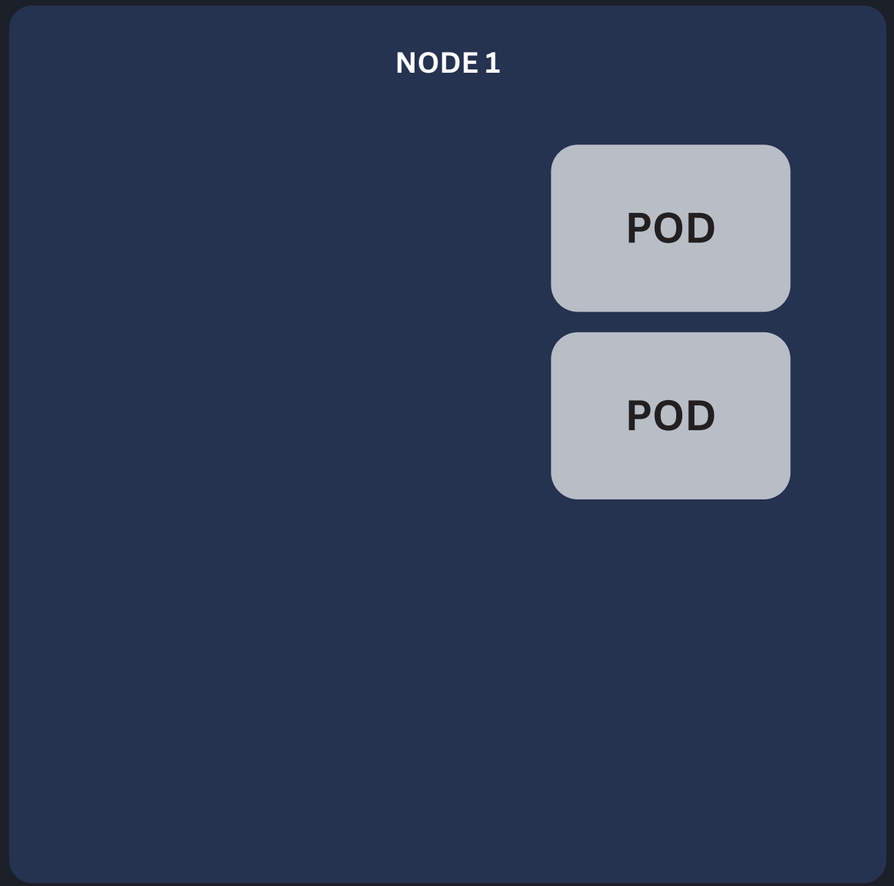
- Single machine
  (Virtual / Physical)
- Runs in K8S cluster and hosts POD
- Provides CPU, memory and storage to Pods
- Flexible in nature



WLUG
COMMUNITY | KNOWLEDGE | SHARE

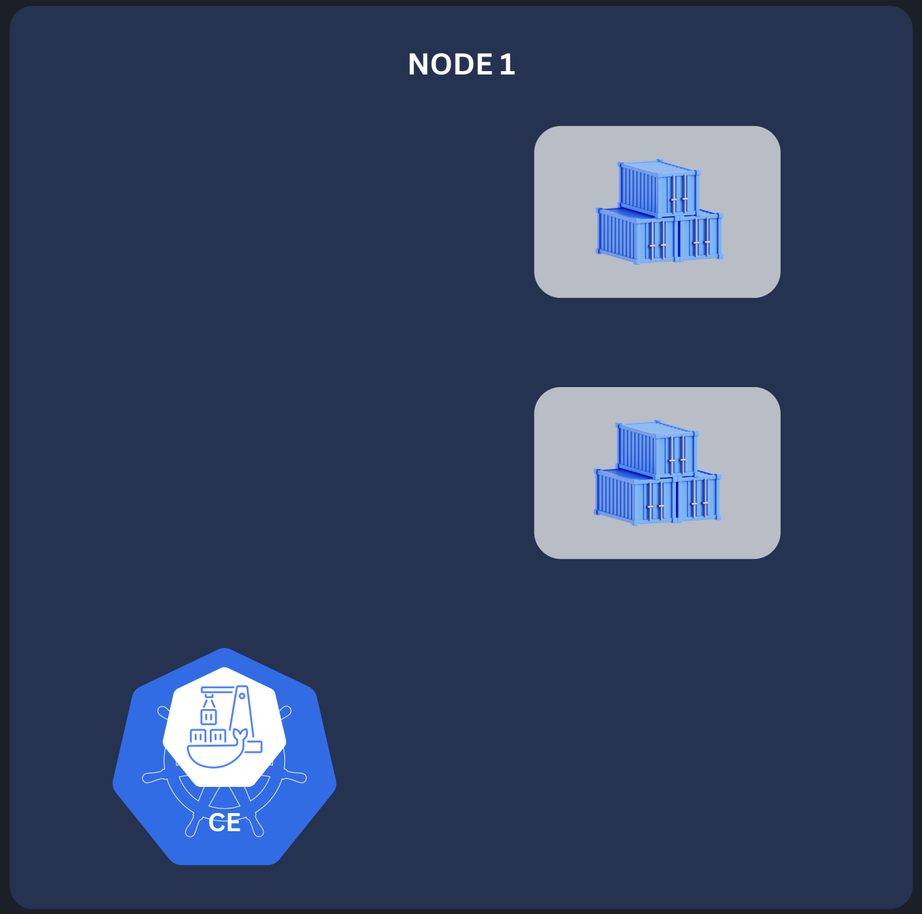
PODS

- Smallest and deployable unit
- Abstraction over container
- Single instance of a running process
- Each Pod gets its own IP address
- Pods are ephemeral



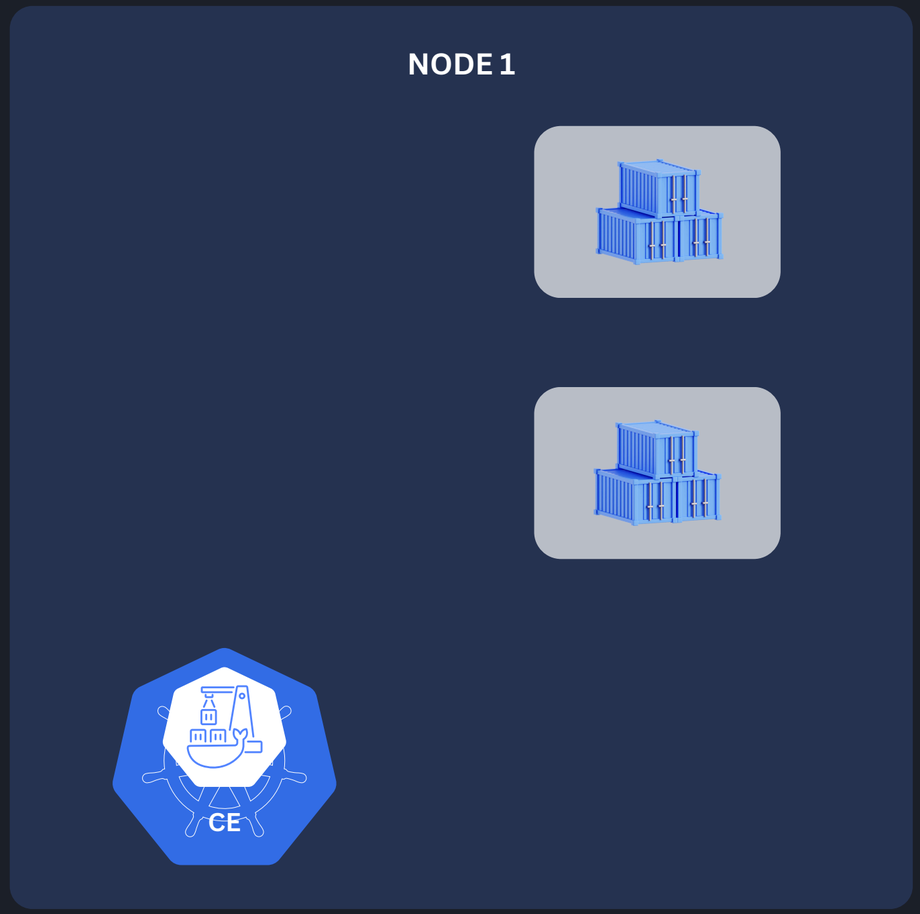
CONTAINER

- Pod usually runs 1 application
- It may contain multiple containers
- Each container has a specific role (e.g., app code, logging, monitoring).
- Containers share the Pod’s resources
- They communicate via localhost



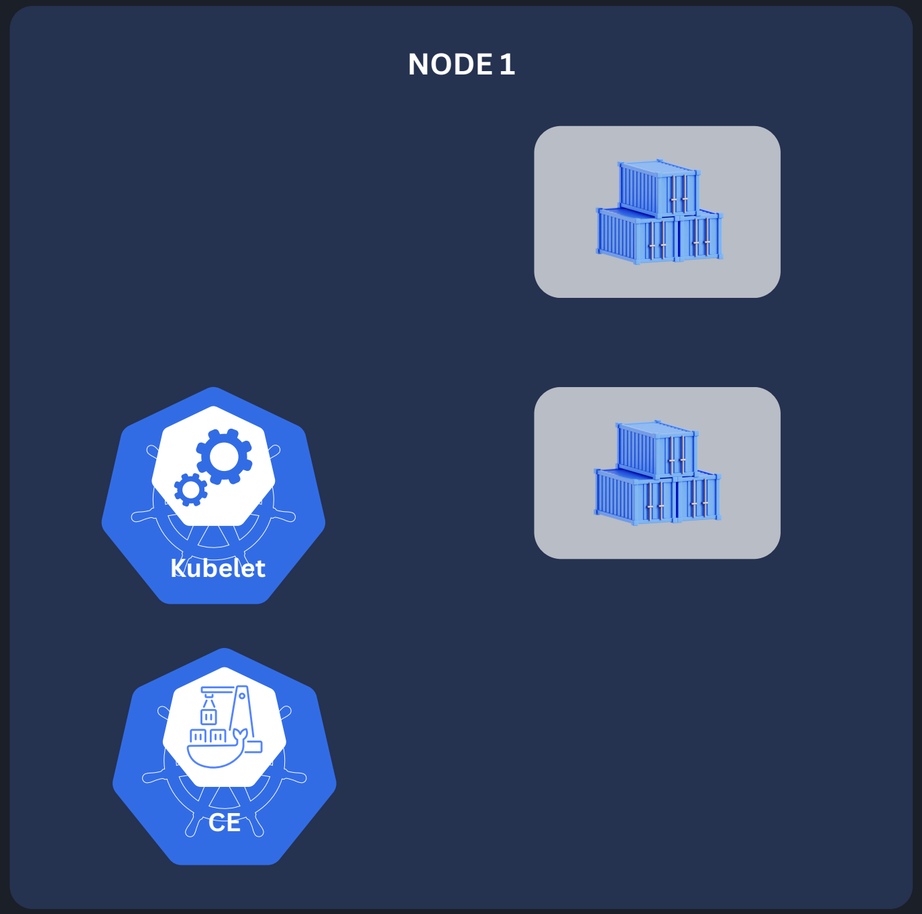
CONTAINER ENGINE

- Software responsible for running the container on each node
- Pull images from registry
- Stop - Start container
- Docker, containerd, CRI-O (common runtimes in Kubernetes)



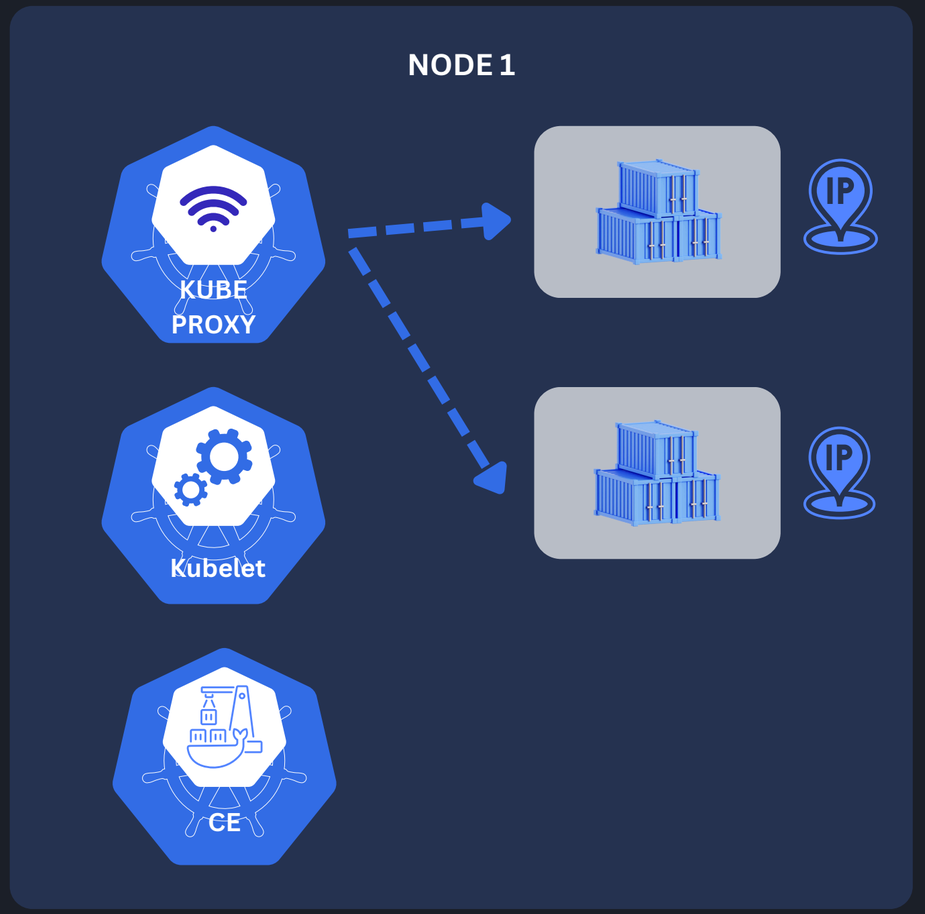
KUBELET

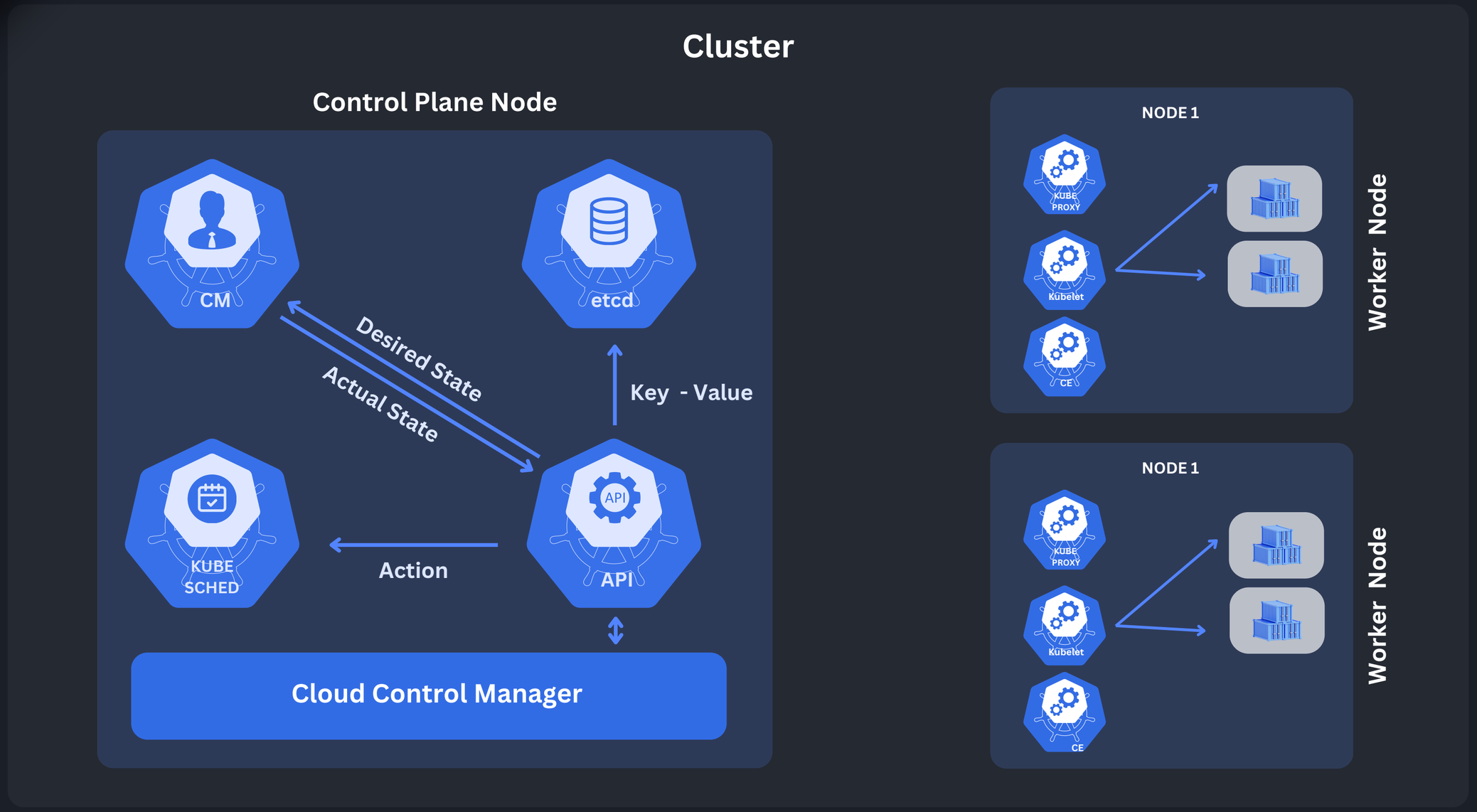
- Node agent running on node
- Communicates with API server
- Listen to commands from API server
- Sends success / fail report to API server

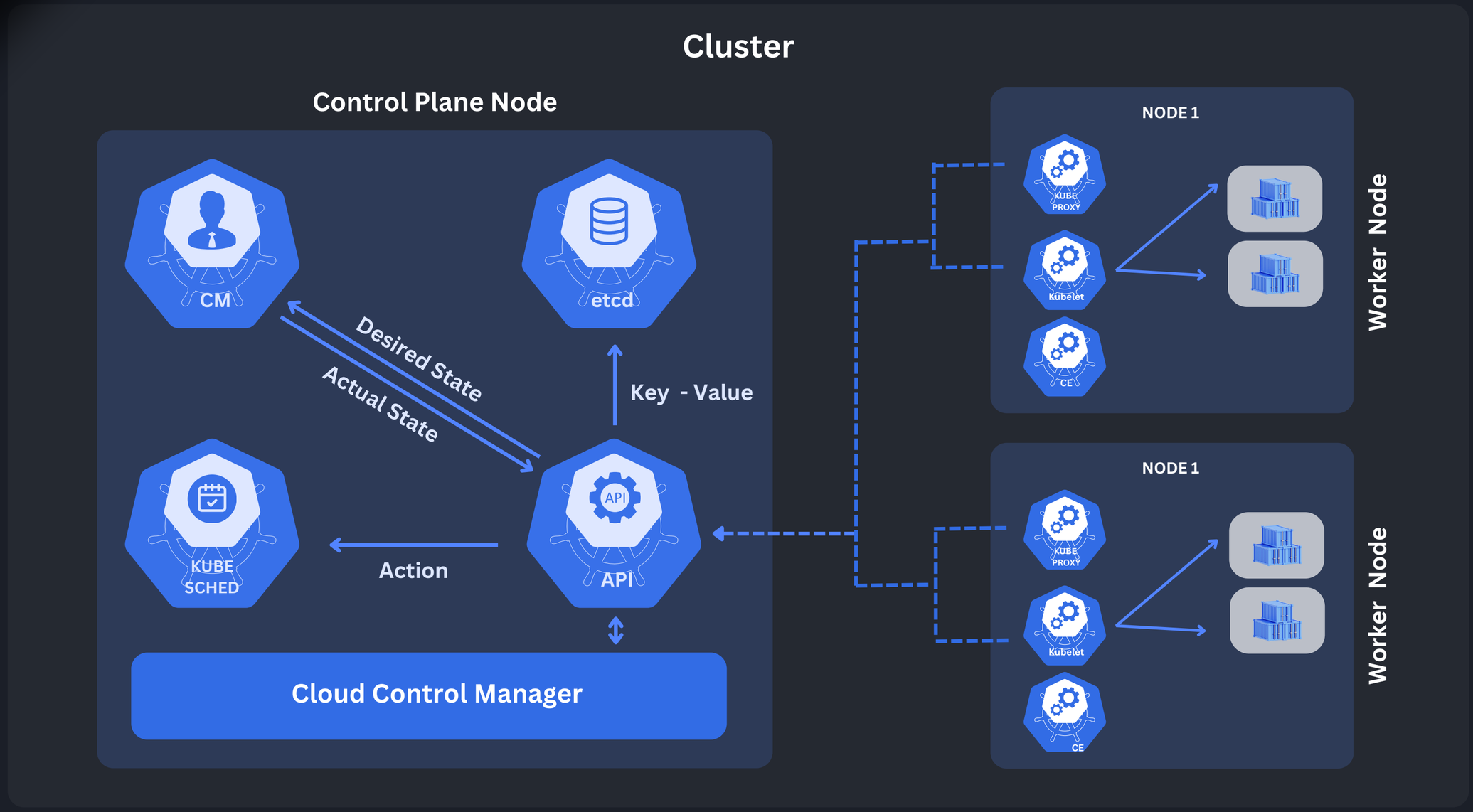


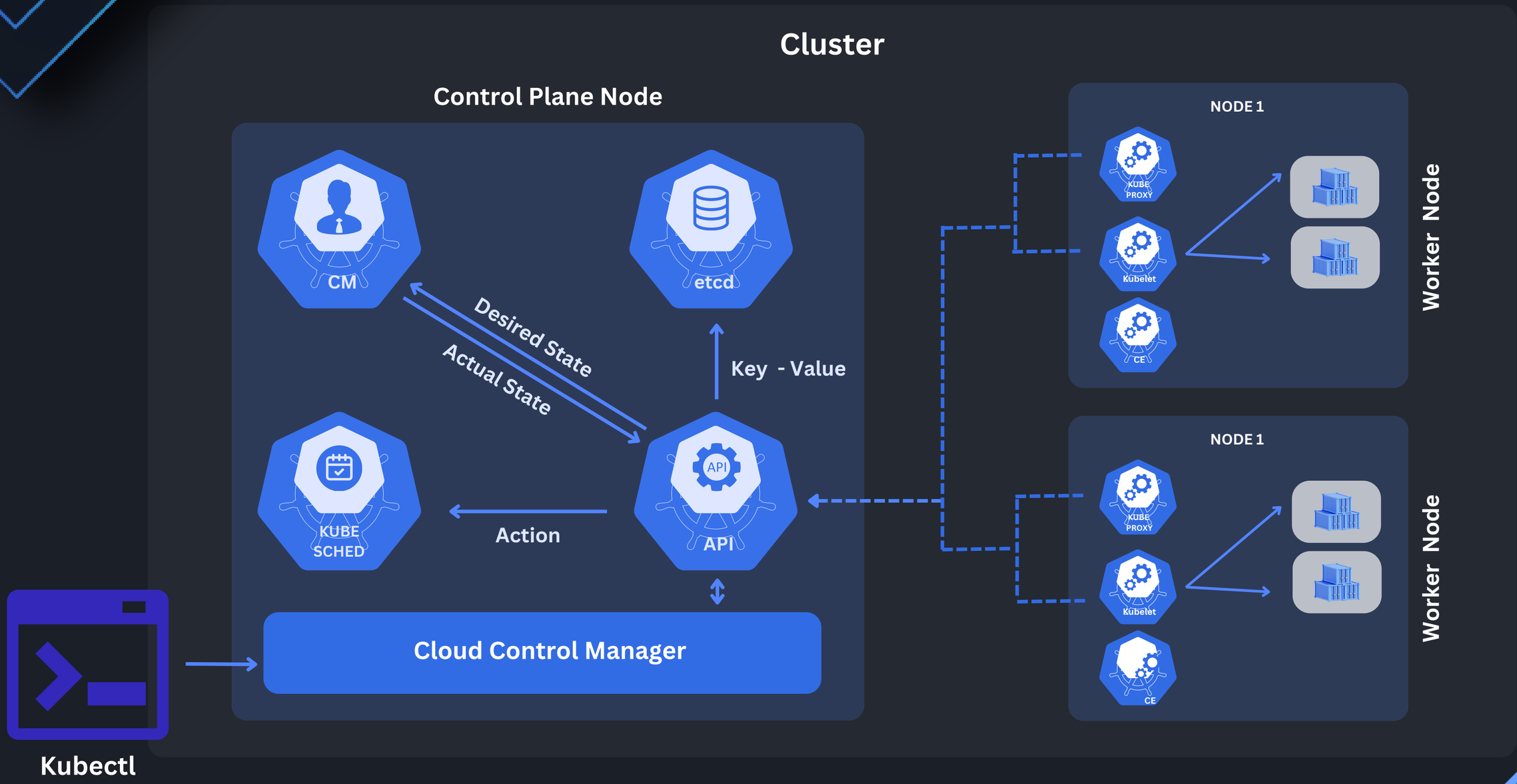
KUBE-PROXY

- Manages networking and communication for the pods
- Responsible for managing network rules
- Routing and load balancing traffic between services and pods






Cluster
Control Plane Node
CM
etcd
Desired State
Actual State
Key - Value
KUBE SCHED
API
API
Action
Cloud Control Manager
NODE 1
KUBE PROXY
Kubelet
CE
Worker Node
NODE 1
KUBE PROXY
Kubelet
CE
Worker Node



K8S FEATURES

- Automated rollouts and rollbacks
- Service discovery and load balancing
- Storage orchestration
- Self-healing

WLUG
COMMUNITY | KNOWLEDGE | SHARE

K8S FEATURES

- Secret and configuration management
- IPv4/IPv6 dual-stack
- Batch execution
- Horizontal scaling

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER VS K8S

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER

- Tool to deploy, create and run application

KUBERNETES

- Container Orchestration tool

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER

- Provides a consistant environment for application using container

KUBERNETES

- Manages, Scales and Operates containers across multiple hosts

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER

- Docker does not support auto-scaling

KUBERNETES

- Supports auto-scaling of pods in a cluster

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER

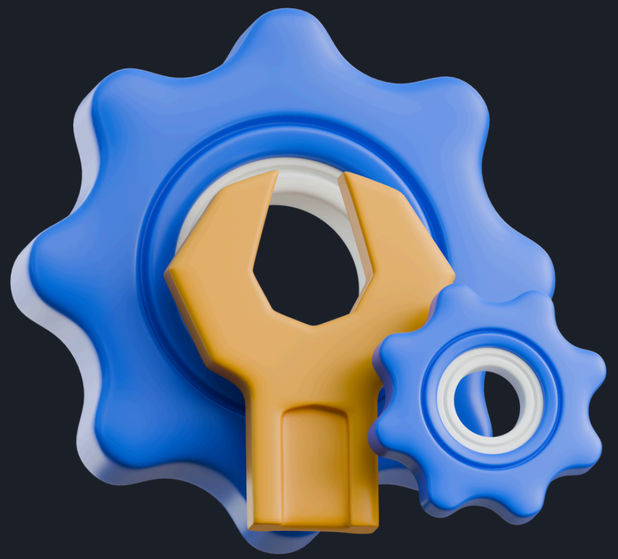
- Manual setup or third-party tools for service discovery

KUBERNETES

- Built-in DNS-based service discovery (KubeProxy)

WLUG
COMMUNITY | KNOWLEDGE | SHARE

K8S TOOLS



WLUG
COMMUNITY | KNOWLEDGE | SHARE

MINIKUBE

- Sets up a local Kubernetes cluster
- Lightweight, single-node cluster
- Creates a virtual machine for Kubernetes
- Supports the latest kubernetes release
- Used to learn and development

NEED OF MINIKUBE

- To Practice & Learn Kubernetes without needing cloud resources
- Local Development and Testing
- Zero Cost
- Easy to set up

KUBECTL

- Command-line tool
- Used to control, inspect and manage Kubernetes clusters
- Creates communication bridge between K8S cluster’s control plane using K8S API

KUBECTL

FEATURES:

- Declarative control: Define what you want and kubectl makes it happen
- Portability: Same commands work on any cluster (GKE, EKS, Minikube)

WORKING:

- Communicates with the Kubernetes API server to execute commands
- Uses YAML/JSON config files to define resources

WLUG
COMMUNITY | KNOWLEDGE | SHARE

INSTALLATION

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MINIKUBE

```
sudo apt update
```

```
minikube version
```

WLUG
COMMUNITY | KNOWLEDGE | SHARE

KUBECTL

```
sudo apt update
```

```
kubectl version --client
```

WLUG
COMMUNITY | KNOWLEDGE | SHARE

COMMAND THE FLEET

```
minikube start
```

```
minikube status
```

```
kubectl get nodes
```