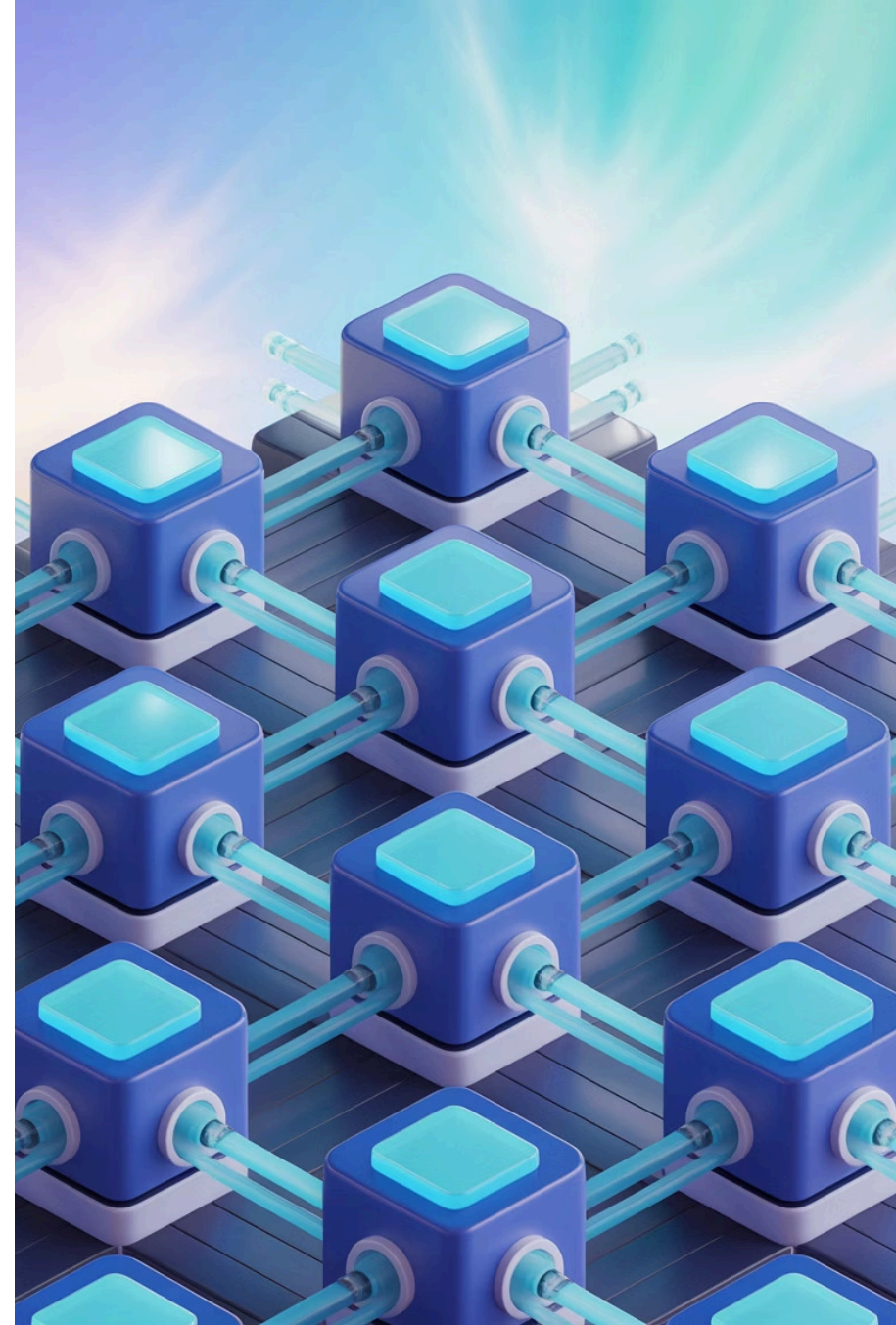


Module 1: Kubernetes Core Architecture & Concepts

Master the foundational building blocks that power cloud-native orchestration at scale





Kubernetes Cluster Components: The Foundation



Control Plane

The intelligent brain of your cluster, managing all state, decisions, and orchestration logic across your entire infrastructure



Worker Nodes

Physical or virtual machines running your containerized workloads as Pods, the fundamental execution units



Essential Agents

kubelet manages pod lifecycle on each node, while **kube-proxy** handles all network routing and service discovery

Pods represent the smallest deployable units, encapsulating one or more tightly coupled containers that share networking and storage resources

Control Plane Responsibilities: The Cluster's Command Center

01

kube-apiserver

Frontend gateway for all cluster interactions, designed to scale horizontally for high availability

02

etcd

Distributed key-value store providing reliable persistence for all cluster state and configuration

03

kube-controller-manager

Runs specialized controllers monitoring and enforcing desired state across resources

04

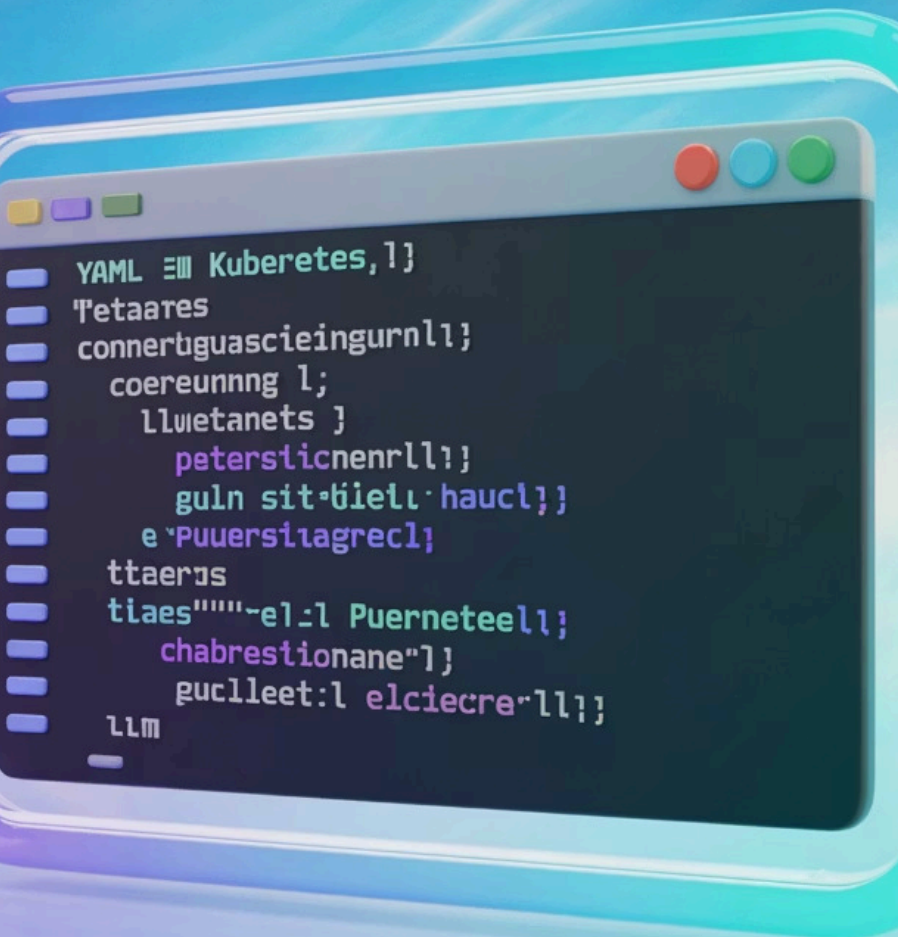
kube-scheduler

Intelligently assigns pods to optimal nodes based on resources, constraints, and policies



The control plane orchestrates cluster health, intelligent scheduling decisions, and maintains state consistency across all worker nodes in real-time

Declarative Configuration Model: Tell Kubernetes What You Want



Define State



Create YAML manifests describing desired Deployments, Services, and resources

Submit to API



Kubernetes API server accepts configs as authoritative source of truth

Automation



No imperative commands needed—lifecycle managed automatically



Example in Action: Define 3 replicas of a pod in your manifest. Kubernetes continuously ensures exactly 3 instances run at all times, automatically recovering from failures.

Desired State Management: Kubernetes' Self-Healing Magic



This reconciliation loop delivers resilient, self-healing application deployments with minimal manual intervention required

Summary: Kubernetes Architecture Empowers Scalable, Reliable Apps

Intelligent Orchestration

Control plane manages cluster-wide decisions, scheduling, and state persistence through etcd

Distributed Execution

Worker nodes run containerized workloads, managed seamlessly by kubelet and kube-proxy agents

Automation & Self-Healing

Declarative configurations and desired state management enable continuous reconciliation and resilience

Production-Ready Foundation

These core architectural concepts provide the power and flexibility for enterprise-scale cloud-native applications

Ready to dive deeper into Kubernetes components, advanced workflows, and real-world deployment patterns!

