

Kubernetes Fundamentals

Instructor: Magdy Salem

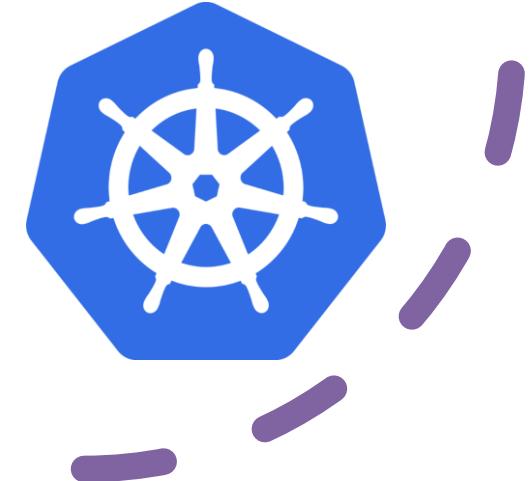
BERNETES

CLOUD OPERATING SYSTEM



Agenda

- Why Kubernetes?
- Deep dive Kubernetes architecture
- Kubernetes CLI (kubectl) introduction
- Installing Minikube vsKIND



Why Kubernetes?

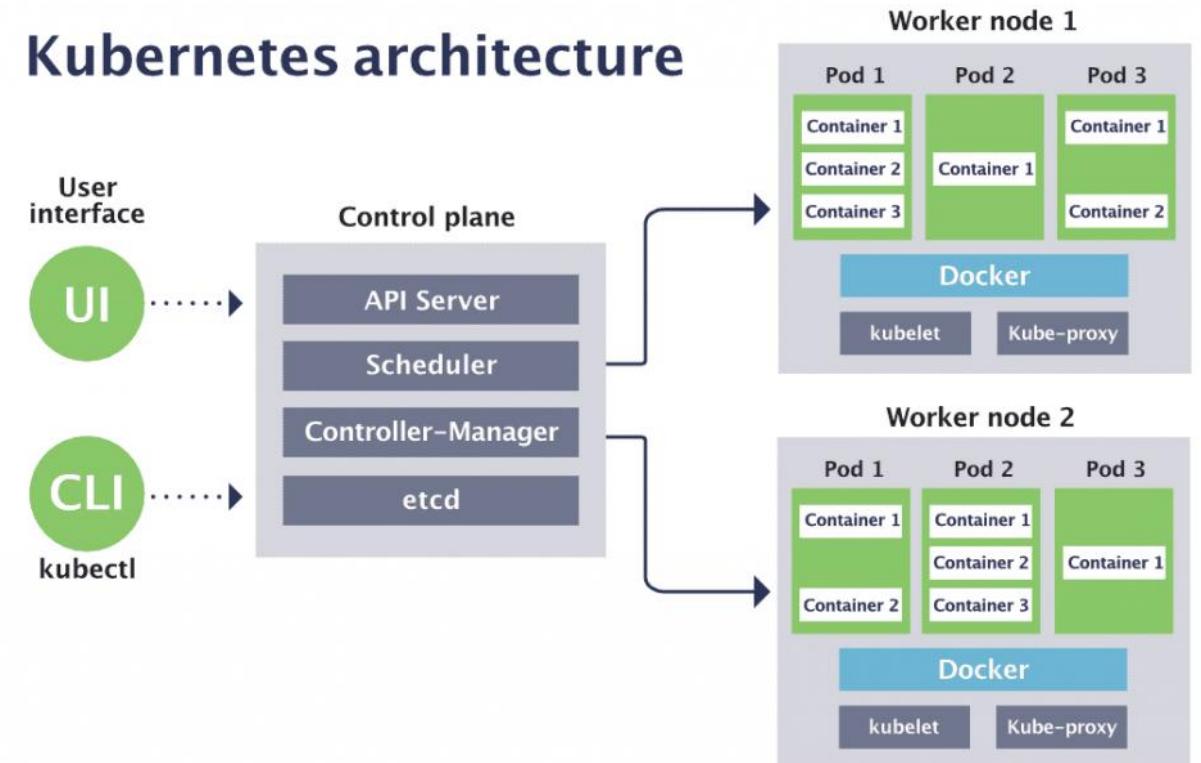
- Manual container orchestration doesn't scale
- Microservices increase complexity
- Kubernetes provides:
 - Automated deployment & scaling
 - Healing & service discovery
 - Config and secret management



Core Concepts & Architecture

- Cluster = control plane + worker nodes
- Pod = smallest deployable unit)
- Deployment = manages desired state of apps
- Service = stable networking endpoint for Pods
- ETCD, Scheduler, Controller Manager, Kubelet

Kubernetes architecture



API Server

- Acts as the front-end for the Kubernetes control plane
- Accepts RESTful calls from kubectl and other clients
- Validates requests and updates the cluster state in ETCD
- Communicates with all other components (scheduler, controller manager)



Scheduler

- Watches for newly created Pods without assigned nodes
- Selects the best available node based on resource requirements and constraints
- Factors: CPU, memory, affinity rules, taints/tolerations, custom policies
- Binds the Pod to a Node (but does not actually launch it)



Controller Manager

- Runs controllers that regulate the state of the cluster
- Examples: ReplicationController, NodeController, JobController, EndpointsController
- Continuously monitors cluster state and performs reconciliation to match the desired state
- Communicates with the API Server to watch resources and submit updates



ETCD

- Consistent and highly-available key-value store
- Stores all cluster configuration data, state, and metadata
- Acts as the single source of truth for Kubernetes
- Used by API Server to read/write cluster information



Kubernetes CLI (kubectl)

- kubectl version
- kubectl cluster-info
- kubectl get nodes
- kubectl get pods
- kubectl apply -f <file.yaml>
- kubectl describe, delete, logs



Installing Minikube

- Minikube = Single-node Kubernetes cluster for local use
- Use VirtualBox, Docker, or Hyperkit as VM driver
- Compatible with kubectl



Installing KIND

- KIND = Kubernetes IN Docker
- Runs K8s clusters in Docker containers
- Good for CI/CD and lightweight testing



Demo



Lab

