kubernetes

APURVA BHANDARI

AGENDA

- Kubernetes Vs Docker Swarm
- Kubernetes Architecture
- Pod Lifecycle
- Kubernetes Dashboard
- Demo

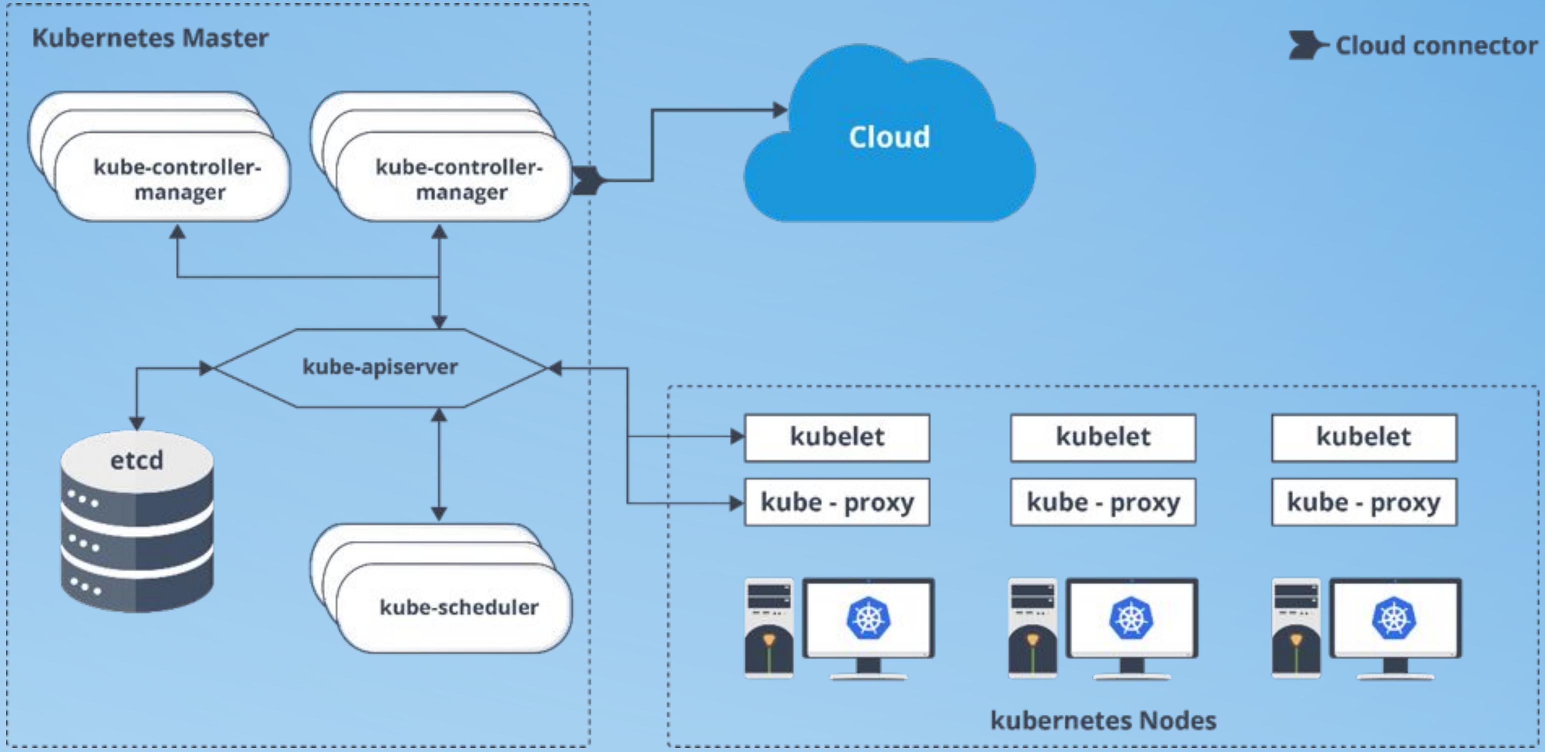
KUBERNETES VS DOCKER SWARM

Kubernetes	Docker Swarm
• Installation is complicated.	• Installation is easy.
• GUI available.	• No GUI.
• Auto scaling.	• No Auto scaling.
• Manual intervention needed for load balancing traffic between diff containers in diff pods.	• Auto load balancing of traffic between container and cluster.
• Rolling updates and automatic rollback.	• No Automatic rollback.
• Share volumes with other containers in same pod.	• Share volumes with any container.

WHY KUBERNETES

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

KUBERNETES ARCHITECTURE



kubernetes

MANAGED K8S VS SELF-MANAGED K8S

Managed k8s

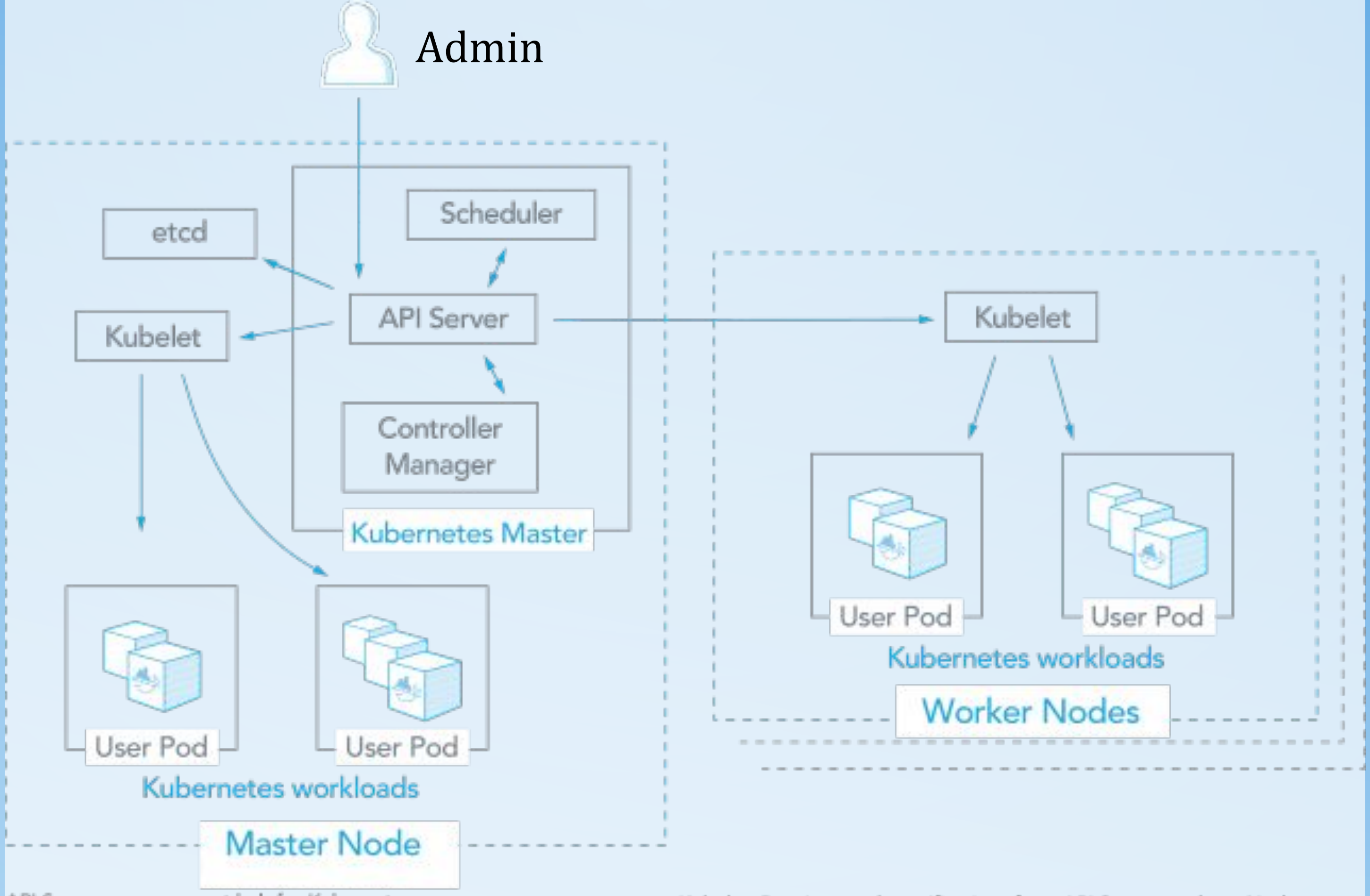
- Services offering from Cloud provider (i.e. EKS, AKS, GKE, OKE, etc)
- No Access to Control Plane.
- Interact with cluster but you do not manage master.

Self-managed k8s

- On Premises/ Bare Metal Deployment.
- Have access to Control Plane
- Full control over the all the components.

kubernetes

ARCHITECTURE OVERVIEW DEEP DIVE

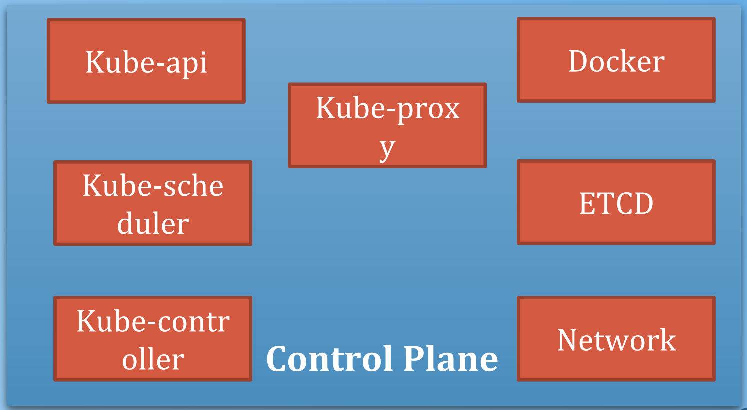


API Server: management hub for Kubernetes
Scheduler: places a workload on the appropriate Node
Controller Manager: scales workloads up/down
etcd: stores configuration data which can be accessed by API Server

Kubelet: Receives pod specifications from API Server, updates Nodes
Master Node: places workloads on Nodes
Worker Nodes: receives requests from Master Nodes and dispatches them
User Pod: a group of containers with shared resources

CONTROL PLANE COMPONENTS

- Kube-apiserver
- ETCD
- Kube-scheduler
- Kube-controller manager



Kube-api

Kube-proxy

Docker

Kube-scheduler

ETCD

Kube-controller

Control Plane

Network

KUBE-APISERVER

- Kube-apiserver is Primary Management Component.
- The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.
- The Kubernetes API server validates and configures data for the api objects which include pods, services, replicationcontrollers, and others.
- It is responsible for Authenticate User, Validate Request, Retrieve data, Update ETCD, Schedular, kubelet.

kubernetes

ETCD

- ETCD act as cluster database.
- It stores data in the form of Key-Value pair.
- It stores object and config (Nodes, Pods, Configs, Secrets, Accounts, Role, RoleBindings).

KUBE-CONTROLLER

- Component on the master that runs controllers.
- Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.
- These controllers include:

  Node Controller

  Replication Controller

  Endpoints Controller

  Service Account & Token Controllers

kubernetes

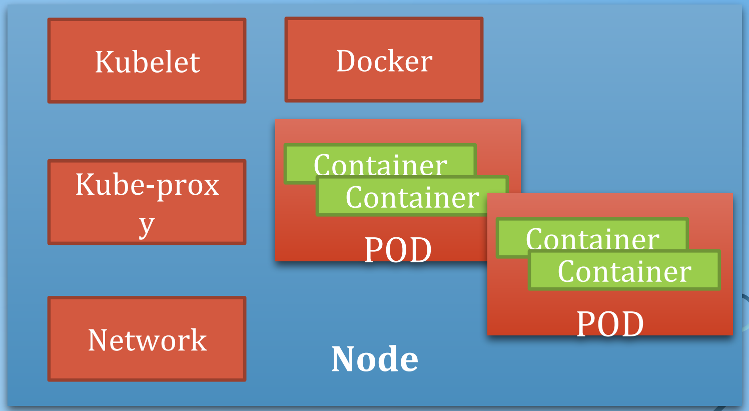
KUBE-SCHEDULER

- Component on the master that watches newly created pods that have no node assigned, and selects a node for them to run on.
- Factors taken into account for scheduling decisions include individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference and deadlines.

kubernetes

NODE COMPONENTS

- Container Runtime
- Kubelet
- Kube-proxy



kubernetes

CONTAINER RUN TIME INTERFACE (CRI)

- To run containers in Pods, Kubernetes uses a container runtime.
- At the lowest layers of a Kubernetes node is the software that, among other things, starts and stops containers.

Docker

CRI-O

Containerd

RKT

KUBELET

- An agent that runs on each node in the cluster. It makes sure that containers are running in a pod.
- The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn’t manage containers which were not created by Kubernetes.

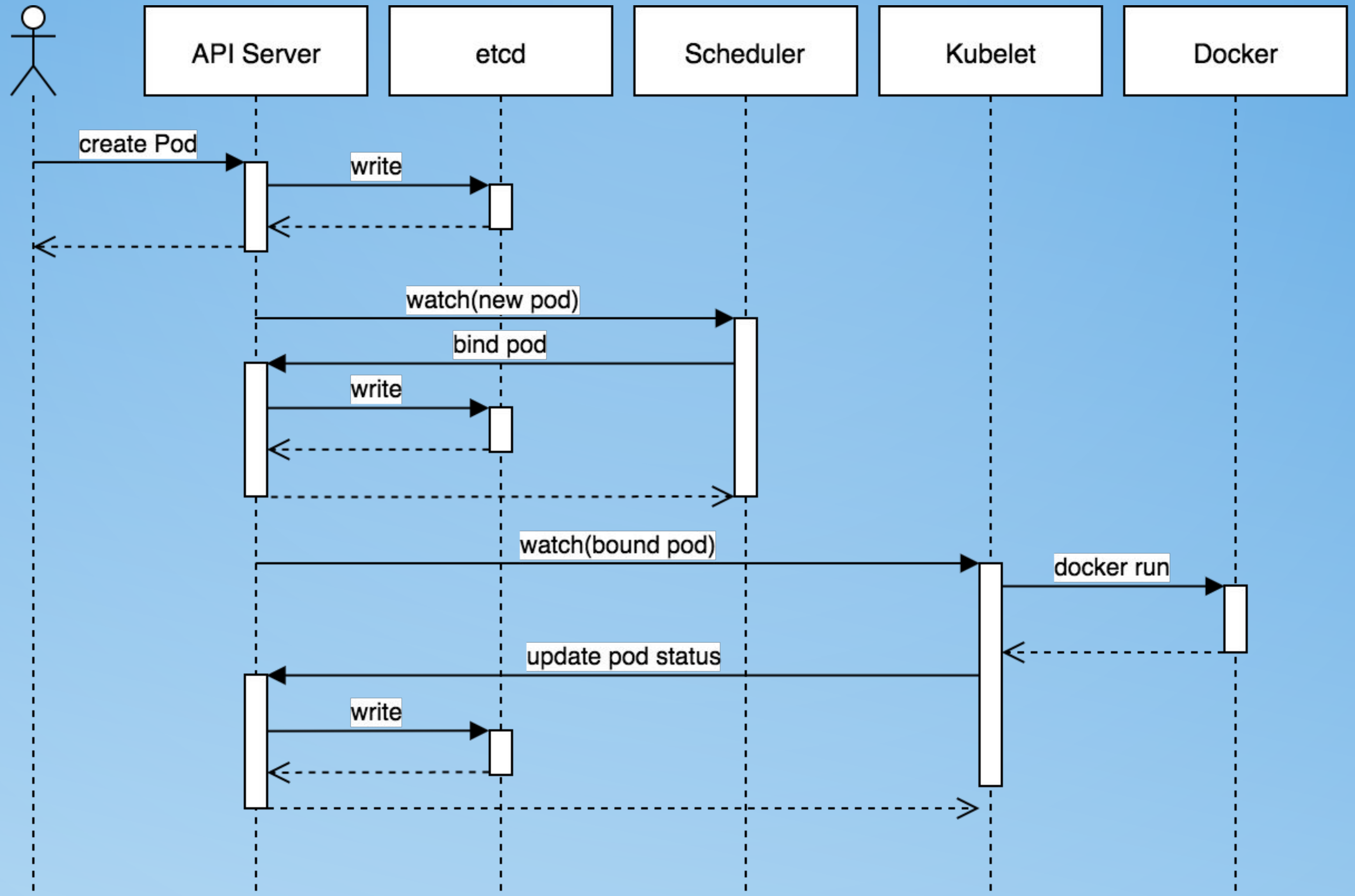
kubernetes

KUBE-PROXY

- kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.
- kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.
- kube-proxy uses the operating system packet filtering layer if there is one and it’s available. Otherwise, kube-proxy forwards the traffic itself.

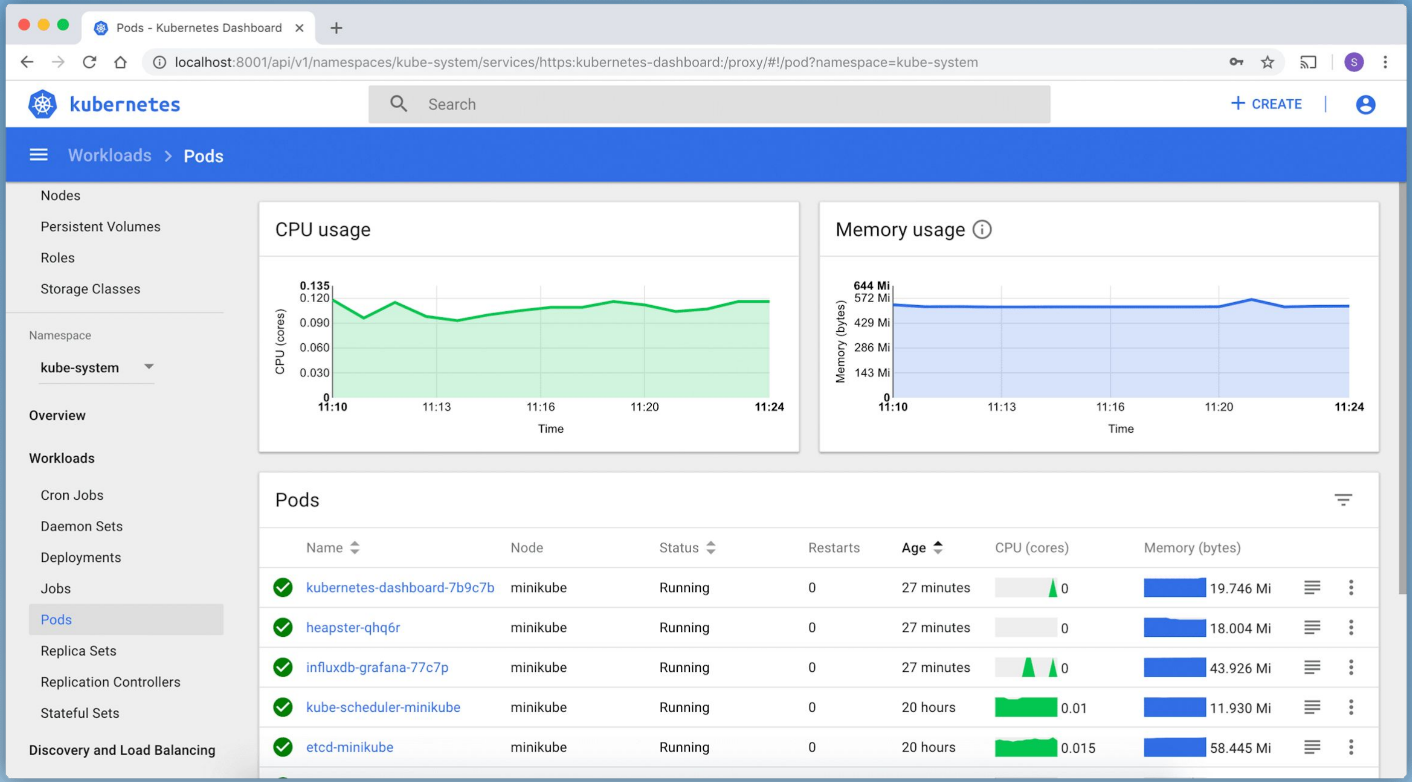
kubernetes

POD LIFECYCLE



kubernetes

KUBERNETES DASHBOARD



Pods

	Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
✓	kubernetes-dashboard-7b9c7b	minikube	Running	0	27 minutes	0	19.746 Mi
✓	heapster-qhq6r	minikube	Running	0	27 minutes	0	18.004 Mi
✓	influxdb-grafana-77c7p	minikube	Running	0	27 minutes	0	43.926 Mi
✓	kube-scheduler-minikube	minikube	Running	0	20 hours	0.01	11.930 Mi
✓	etcd-minikube	minikube	Running	0	20 hours	0.015	58.445 Mi

DEMO APP DEPLOYMENT

- Simple nginx App Deployment on Kubernetes.
- Kubernetes Dashboard and GUI for Deployment.

THANKS

$whoami

Apurva Bhandari, CKAD

SRE @ Vuclip, Pune.

Docker & Kubernetes Enthusiast

Email: apurvbhandari@gmail.com
GitHub: https://github.com/apurvabhandari/Kubernetes

LinkedIn: https://www.linkedin.com/in/apurvabhandari-linux