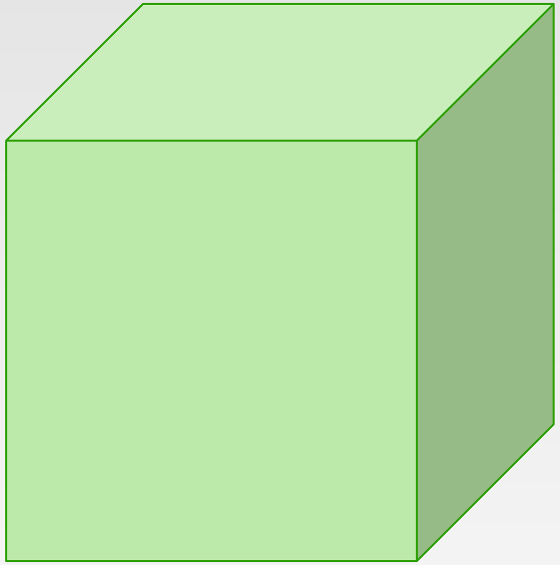
BATCH :	B107 AWS-DevOps
LESSON :	**Kubernetes**
DATE :	19.05.2023
SUBJECT :	**Intro**

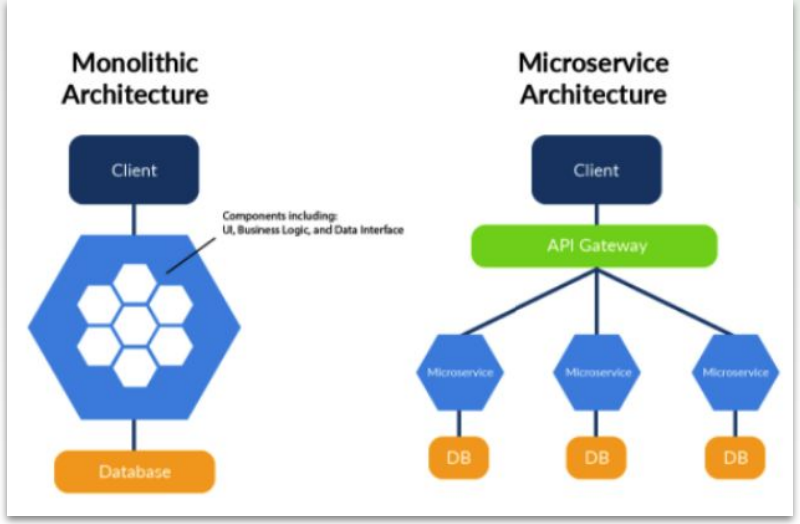
ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ

Kubernetes

Monolith vs Microservices



- Microservice applications are scalable, flexible and more reliable.
- Features are independent.
- Each service is deployed individually. Continuous deployment is easier.
- Complex



- Monolith applications are simple to develop, test and deploy.
- Difficult to scale.
- Features depend on each other.
- Not cost-efficient in the long run.
- Entire codebase is deployed. Continuous deployment is difficult.
- Difficult to maintain, update.
- Change in a framework will affect the entire application.

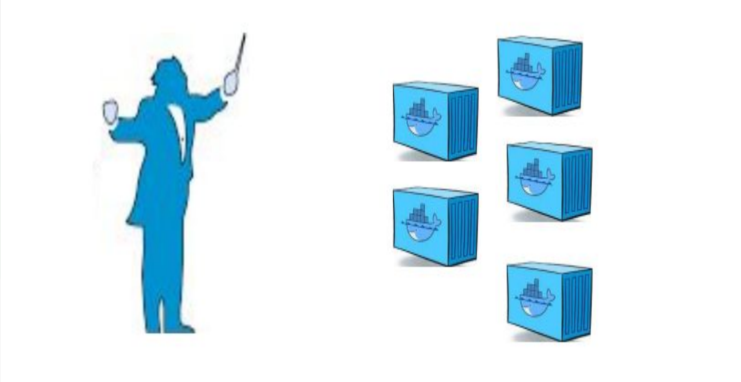
Orchestration

- Microservice applications and containers grow in demand.
- Containers are great, but when you get lots of them running, at some point, you need them all working together in harmony to solve business problems.
- Tools to manage, scale, and maintain containerized applications are called orchestrators, and the most common example of this is Kubernetes.

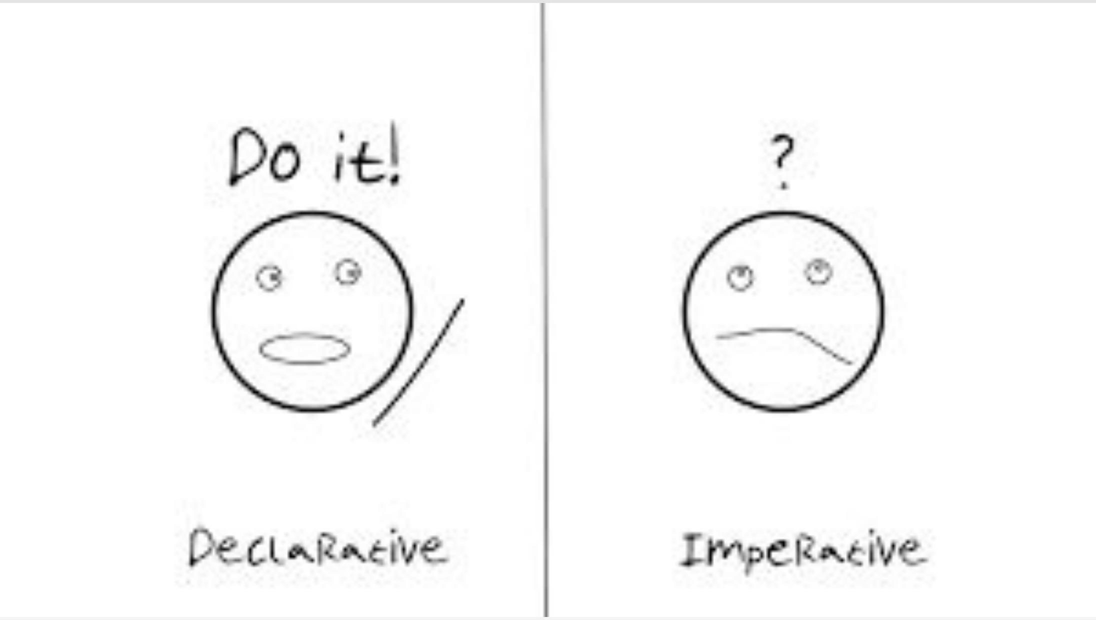
Orchestration

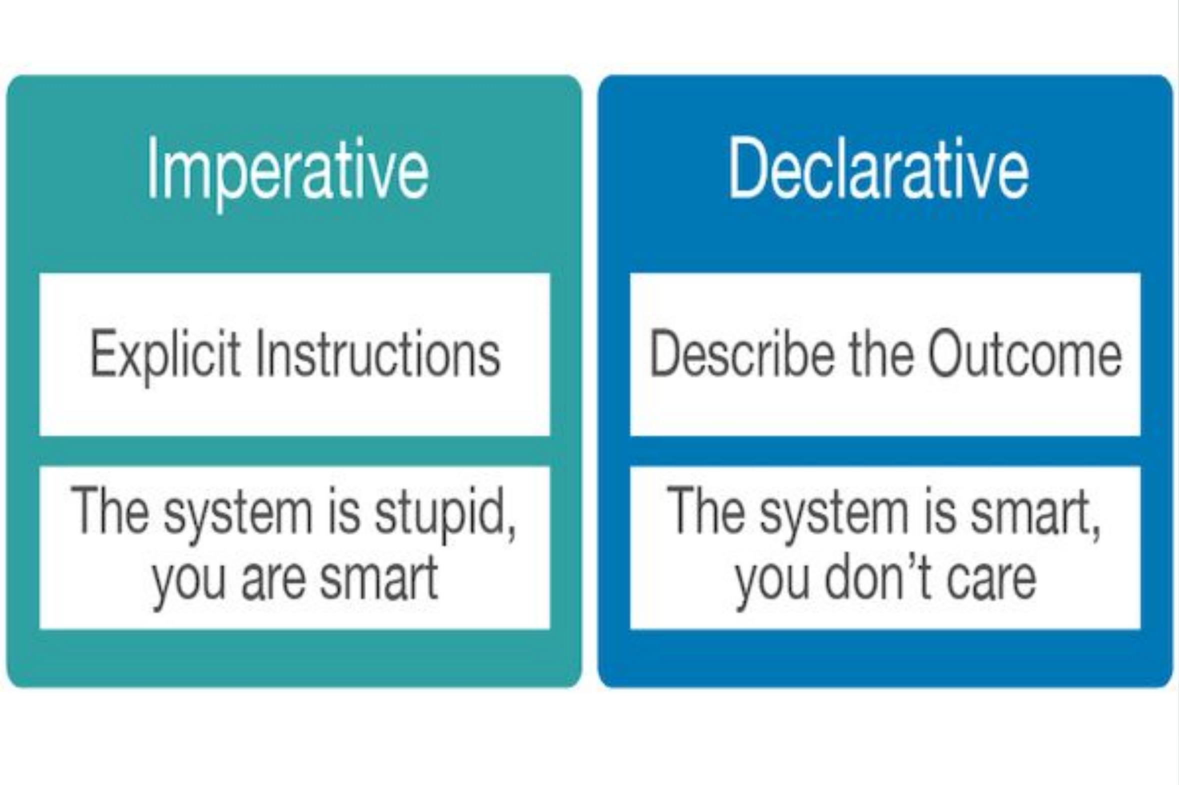
Container orchestration is used to automate the following tasks at scale:

- Provisioning and deployments of containers
- Availability of containers
- Load balancing, traffic routing and service discovery of containers
- Health monitoring of containers
- Securing the interactions between containers.
- Configuring and scheduling of containers
- Allocation of resources between containers



Declarative vs Imperative





Imperative	Declarative
Explicit Instructions	Describe the Outcome
The system is stupid, you are smart	The system is smart, you don't care

What Does “Kubernetes” Mean?

Greek word for “pilot” or
“Helmsman of a ship”
or K8s

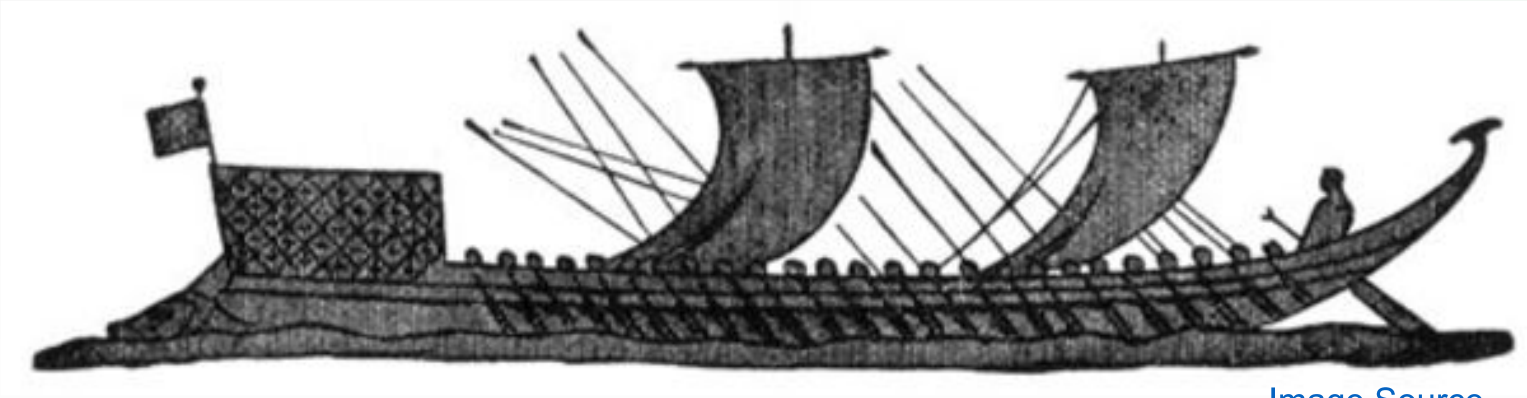


Image Source

What is Kubernetes?

- Kubernetes is Open Source Orchestration system for Containerized Applications.
- Kubernetes is a platform that **eliminates the manual processes** involved in **deploying containerized applications.**
- **PODs house containers.**
- Kubernetes is used to manage the State of Containers.
    - Start Containers on Specific Nodes
    - Restart Containers when gets Killed
    - Move containers from one Node to Another

Why you need Kubernetes?

- Containers are a perfect way to get the applications packaged and run. In a production environment, you should manage the containers that run the applications and ensure no downtime.

- Kubernetes supplies you with:

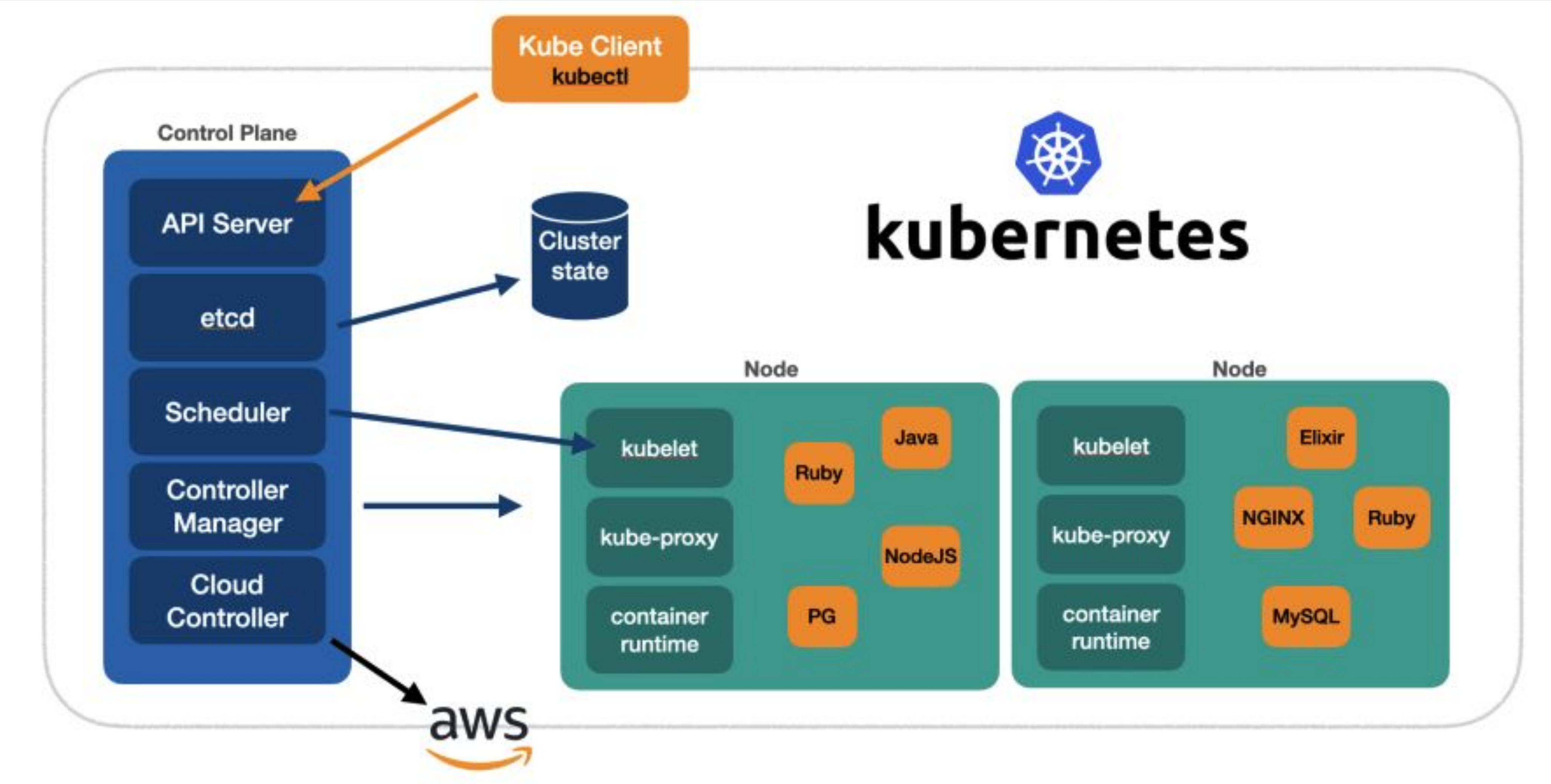
  Service discovery and load balancing
  Storage orchestration
  Automated rollouts and rollbacks
  Automatic bin packing
  Self-healing
  Secret and configuration management

Who manages Kubernetes?

CLOUD NATIVE
COMPUTING FOUNDATION

The CNCF is a child entity of the Linux Foundation and operates as a vendor neutral governance group.

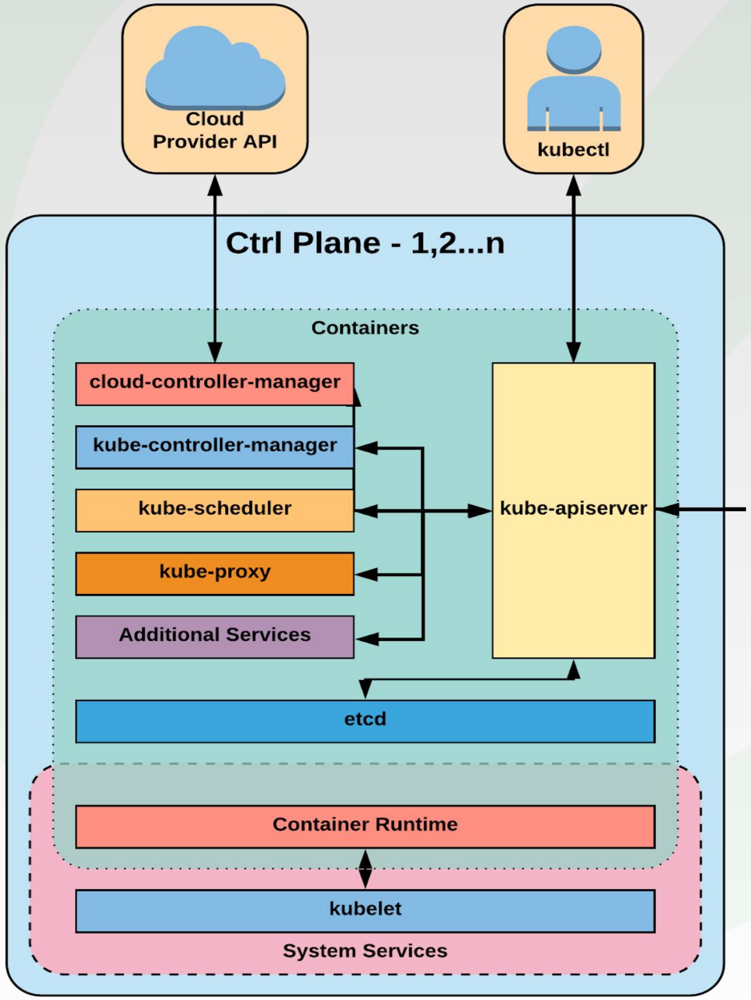
Architecture Overview


Kube Client
kubectl
Control Plane
API Server
etcd
Scheduler
Controller Manager
Cloud Controller
Cluster state
kubernetes
Node
kubelet
kube-proxy
container runtime
Ruby
Java
NodeJS
PG
Node
kubelet
kube-proxy
container runtime
Elixir
NGINX
Ruby
MySQL
aws

Control Plane Components

Control Plane Components

- kube-apiserver
- etcd
- kube-controller-manager
- kube-scheduler



kube-apiserver

- Provides a forward facing REST interface into the kubernetes control plane and datastore.
- All clients and other applications interact with kubernetes strictly through the API Server.
- Acts as the gatekeeper to the cluster by handling authentication and authorization, request validation, mutation, and admission control in addition to being the front-end to the backing datastore.
- **Center of communication**

etcd

- etcd acts as the cluster datastore.
- Provides strong, consistent and highly available key-value store for persisting cluster state.
- Stores objects and config information.
- **Memory of Kubernetes Cluster**

etcd

Uses “*Raft Consensus*” among a quorum of systems to create a fault-tolerant consistent “*view*” of the cluster.

https://raft.github.io/

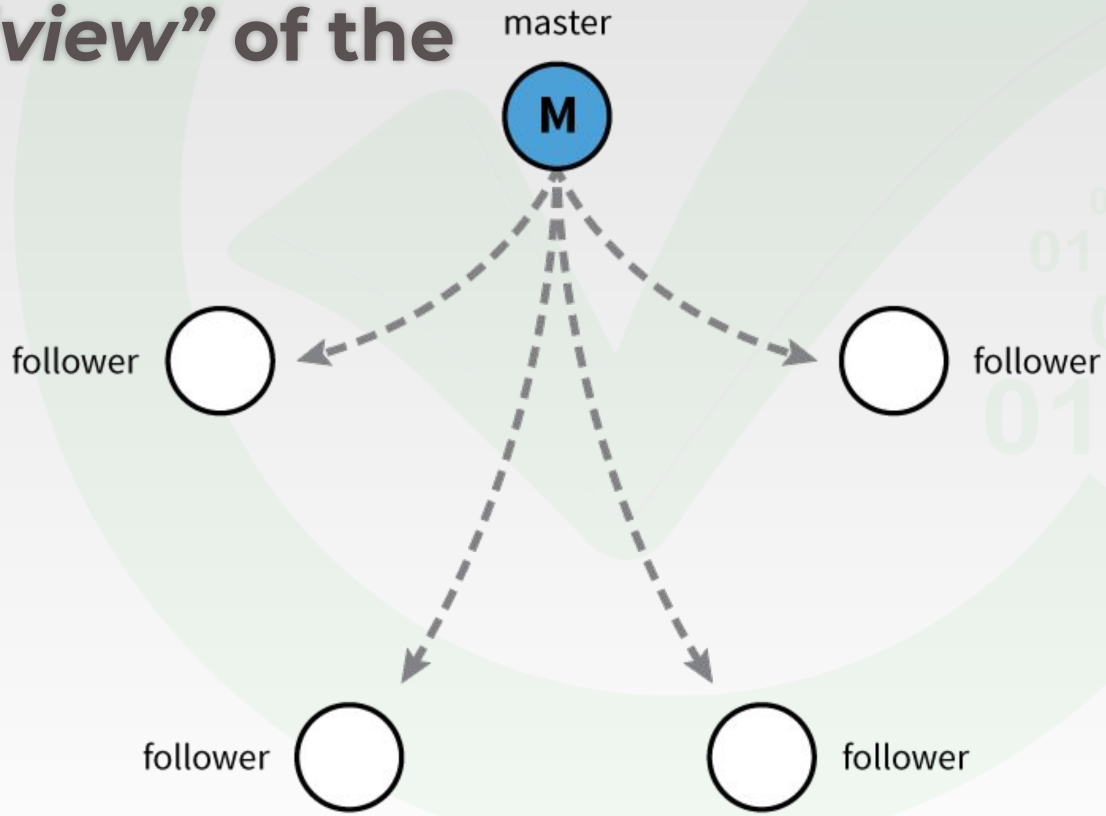


Image Source

kube-controller-manager

- Serves as the primary daemon that manages all core control processes.
- Monitors the cluster state via the apiserver and **steers the cluster towards the desired state**.
- List of core controllers:
  - Node
  - Job
  - ServiceAccount
  - ...

cloud-controller-manager

- Daemon that provides cloud-provider specific controllers.
- If K8s is run locally, the cluster has no cloud controller manager.
- The controllers include Node, Route, Service.

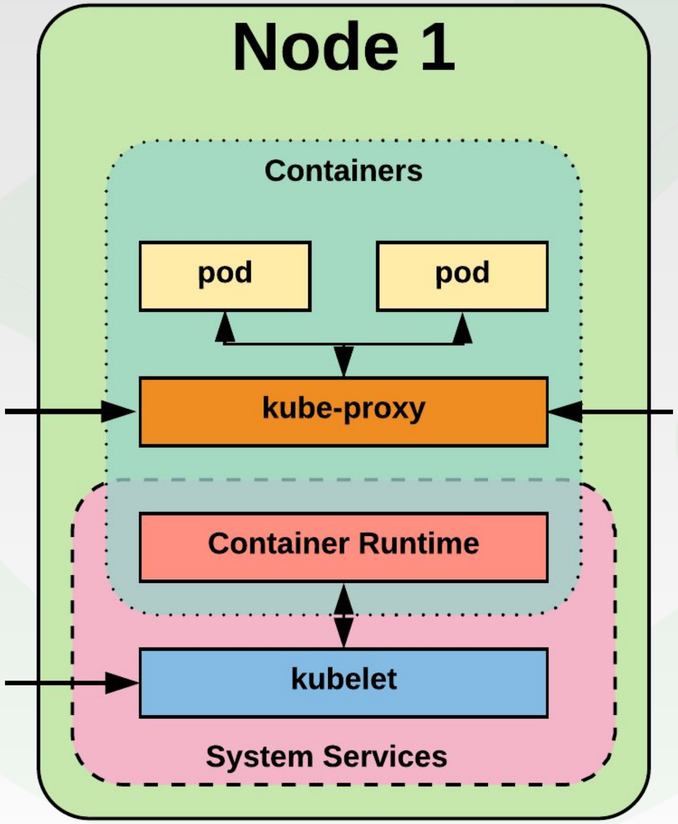
kube-scheduler

- Evaluates workload requirements and attempts to place it on a matching resource.
- Watches for newly created Pods with no assigned node, and **selects a node for them to run on**.
- Default scheduler uses bin packing.
- Workload Requirements for scheduling decisions can include: general hardware requirements, affinity/anti-affinity, labels, and other various custom resource requirements.

Node Components

Node Components

- kubelet
- kube-proxy
- Container Runtime Engine



kubelet

- Acts as the node agent responsible for **managing the lifecycle of every pod on its host**.
- It **makes sure that containers are running in a Pod.**
- The kubelet doesn't manage containers which were not created by Kubernetes.
- The kubelet works in terms of a PodSpec. A PodSpec is a YAML or JSON object that describes a pod.

kube-proxy

- **Manages the network** rules on each node.
- Performs **connection forwarding or load balancing** for Kubernetes cluster services.
- Available Proxy Modes:
  - Userspace
  - iptables
  - ipvs (default if supported)

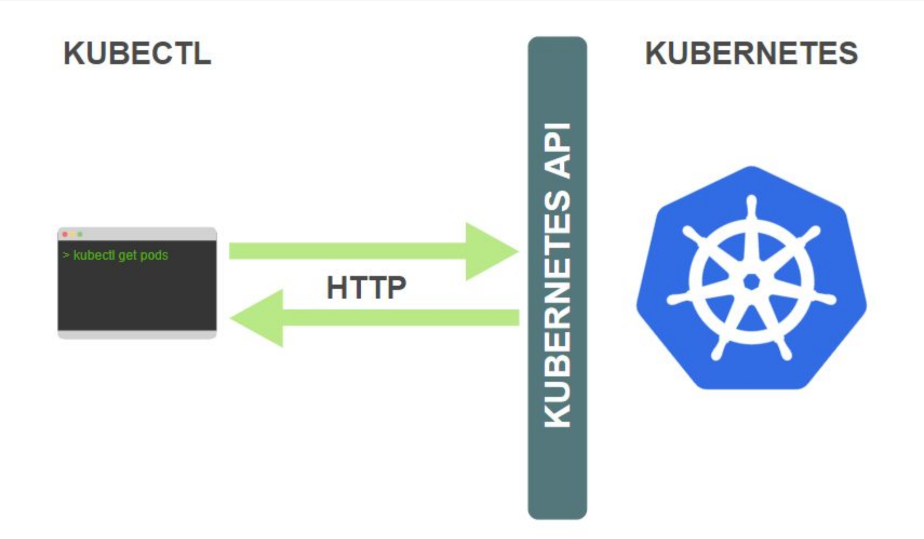
Container Runtime Engine

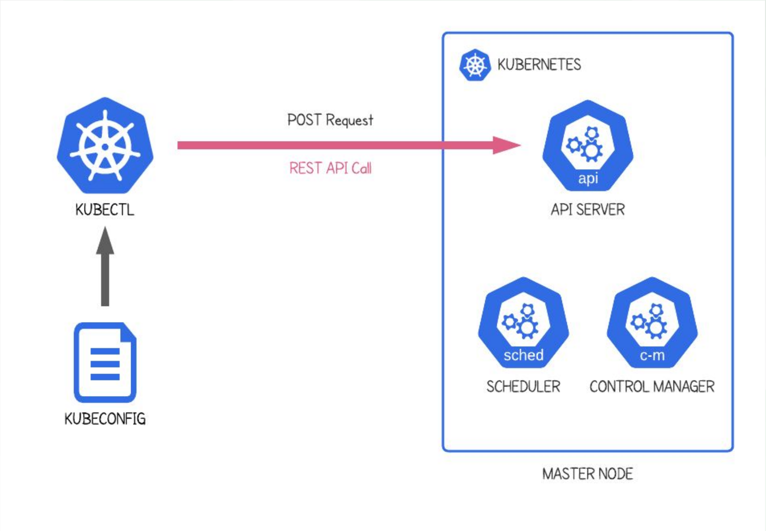
- A container runtime is a **CRI (Container Runtime Interface)** compatible application that executes and manages containers.
    - Containerd
    - Docker
    - Cri-o
    - Rkt
    - Kata (formerly clear and hyper)
    - Virtlet (VM CRI compatible runtime)

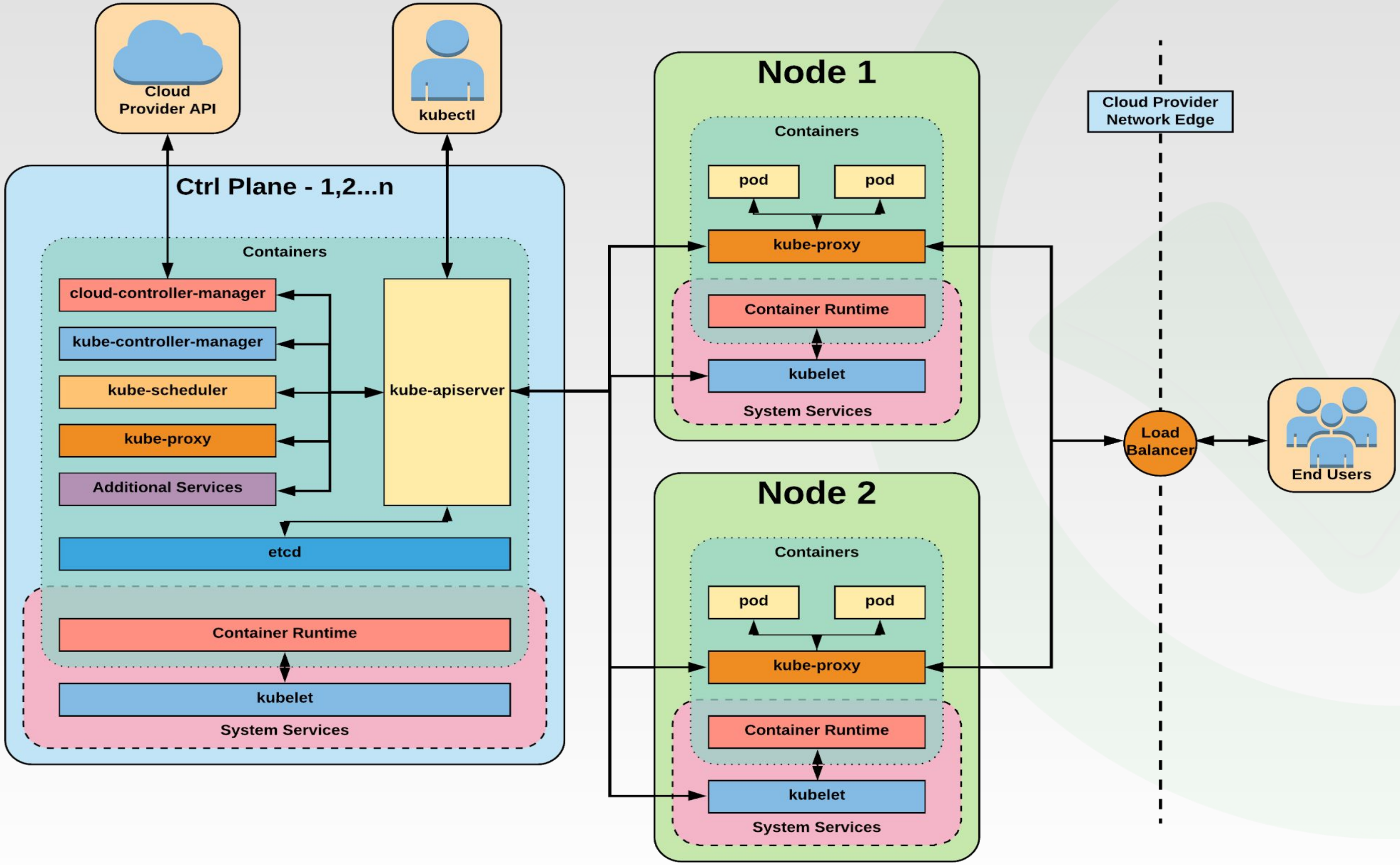
kubectl

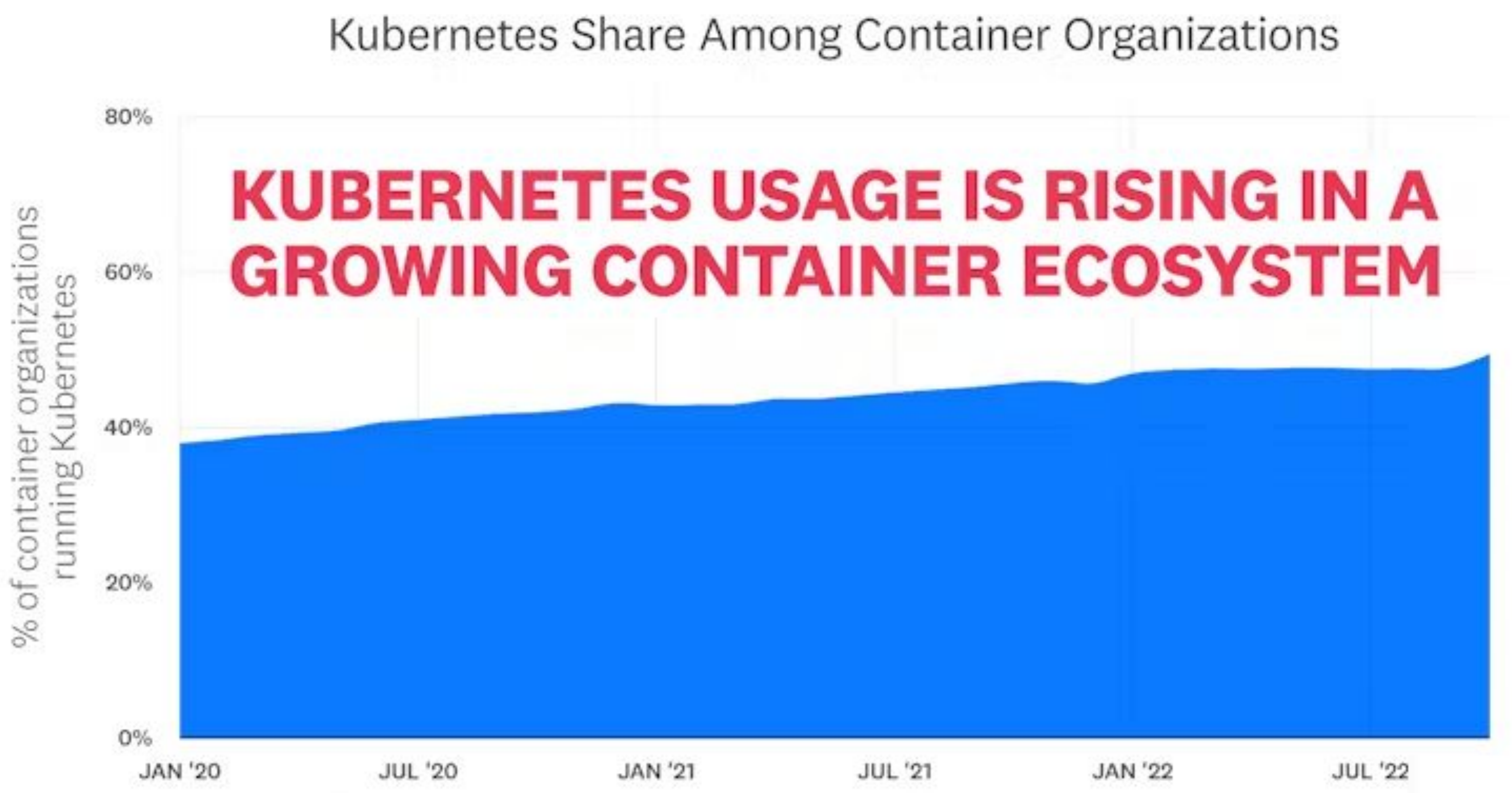
Kubectl is (almost) the only tool we'll need to talk to Kubernetes

- It is a rich CLI tool around the Kubernetes API
- Everything you can do with kubectl, you can do directly with the API
- Kubectl can be pronounced "Cube C T L", "Cube cuttle", "Cube cuddle






Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users



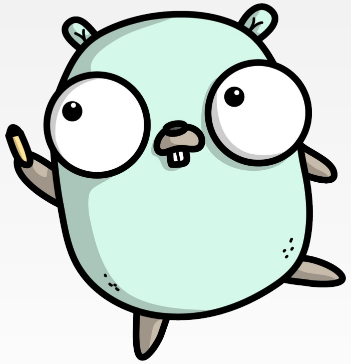
KUBERNETES USAGE IS RISING IN A GROWING CONTAINER ECOSYSTEM

The definition of a container organization includes organizations running Kubernetes Amazon Elastic Container Service, serverless container technologies, and more.

Source: Datadog

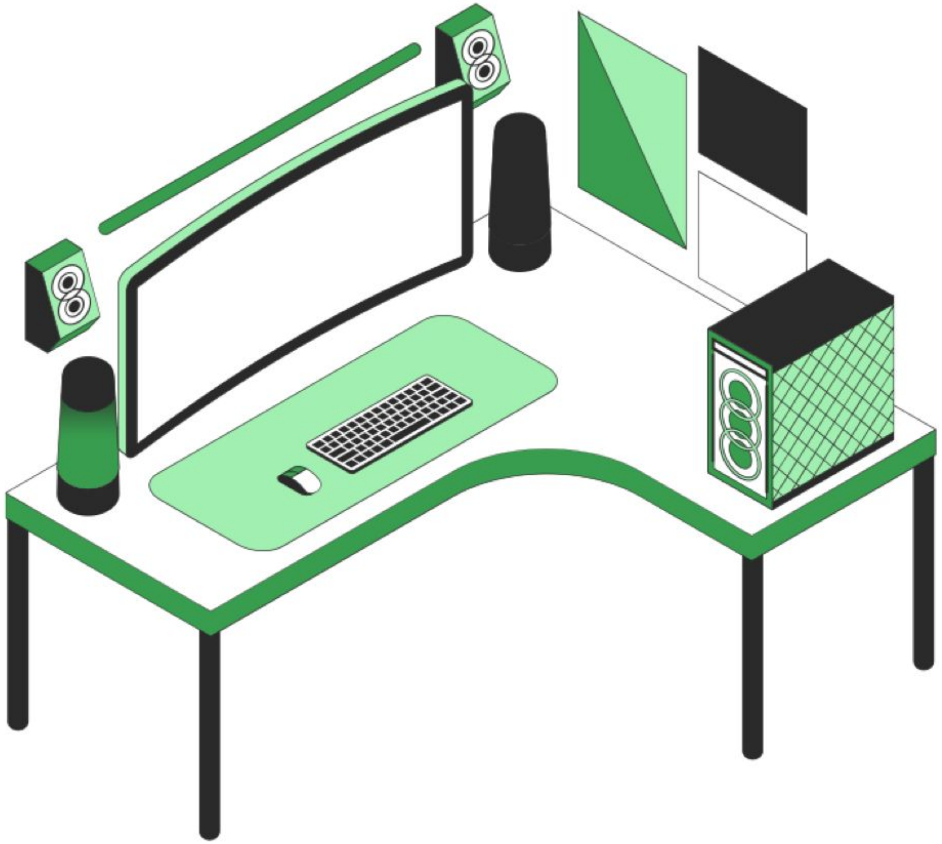
How to install and use K8s

- kind: https://kind.sigs.k8s.io/
- minikube: https://minikube.sigs.k8s.io/
- katacoda: https://killercoda.com/playgrounds
- play with kubernetes: https://training.play-with-kubernetes.com/
- kubernetes on Docker Desktop
- kubeadm
- kops



Summary and Tips

- k8s stands for ..
- k8s CLI is ..
- memory of k8s is ..
- communications center of k8s is ..
- k8s master node is ..
- k8s node components are ..
- k8s control plane components are ..
- pod is ..



Do you have any questions?

Send it to us! We hope you learned something new.