Kubernetes Architecture

- We are gonna look at two types of nodes that k8s operate on one is master and another is a slave and what roles those 2 have.
- And gonna go through how k8s does what it does and how the cluster is self managed and healing and automated.

- Node Processes
  - Worker machine in k8s cluster
    - One of the main components of k8s architecture are its worker servers or nodes and each node will have multiple pods with containers running in that node and the k8s does it using 3 process that must be installed on very node that are used to schedule and manage those pods
    - Nodes are the cluster servers that actually do the work that's why sometimes also called worker nodes
      - The first process that needs run on every node is the container runtime (in our ex: docker)
      - Application pods have container running inside container runtime that needs to be installed on every node
      - But the process that actually schedules those pods and the containers in underneath is kubelet which is a process of k8s itself unlike container runtime that has interface with both container runtime and the machine the node itself
        - Kubelet interacts with both - the container and node
        - Kubelet starts the pod with a container inside and then assigning resources from that node to that container like CPU, RAM, storage resources
    - Usually these clusters are made of multiple nodes which also must have container runtime and kubelet services installed
    - The third process which is responsible for forwarding request from services to pods is tube proxy which also must be installed
  - How do you interact with this cluster like
    - How to:
      - Schedule pod?
      - Monitor?
      - re-schedule/re-start pod?
      - Join a new Node?
    - The answer is - all this managing processes are done by "Master Nodes"

- Master Process

- Master servers or master node have completely different processes run inside, and there are 4 processes that run on every master node that controls cluster state and the worker nodes as well
  - The first service is a Api Server,
    - It’s like a cluster gateway where client interact from cli or dashboard or from api, it will get the initial request, like if you want to add an application or update or query or want to do anything.
    - Acts as a gatekeeper for authentication, some requests - validates request - forward it to other processes like - docker etc
    - One entry point to the cluster
  - Scheduler
    - To schedule new pod
    - If you request to schedule a pod then the api server will validate and handed over to scheduler
    - And will decide where to put the pod based on the resource calculations
    - Kubelet will get the request from the scheduler and executes that request
    - Scheduler -> kubelet
  - Contriller manager
    - Detects cluster state changes and will help in rescheduling
    - Like crashing, destroying a pod
    - Controller manger -> scheduler -> kubelet
  - etcd
    - Key value store of a cluster state
    - Think of it as cluster brain
    - Cluster changes get stored in the key value store
    - Which helps communicate master process with work/node processes
    - Stored etcd value should be reliable so as a common practice master process is also relplicated and api server is load balanced and etcd store forms a distributed storage across all the master nodes basically replicas