(UI, API, CLI)

API Server → Entrypoint to K8s cluster

cluster
detects ^state
changes ← **Controller Manager** → keep tracks whats
happening in the cluster

Scheduler → ensures pod placement

✓ cluster brain
✓ cluster changes
gets stored in etcd ← **etcd** → kubernetes backed up key value store

[Master]

Virtual Network

[kube proxy / container runtime / Kubelet]

[kube proxy / container runtime / kubelet]

[kube proxy / container runtime / kubelet] → forward the request

[Worker]

(1) kubelet interacts both with container & node.

(2) Kubelets starts the pod with a container inside

Kubernetes components

<u>pod</u> :
1. smallest unit of K8S
2. Abstraction over container
3. Usually 1 application per pod

4. Each pod gets its own ip address
5. New ip address on each re-creation

Service :
1. Permanent ip address
2. Lifecycle of pod & service not connected.
3. Service is also a load balancer.
App should be accessible through browser so need to External service.

for Database, create Internal service.

Ingress :
First request goes to Ingress then that forwards the request to the service.

Config Map & Secret

External configuration of the application like DB — URL. goes to Config Map.

Secret contains things like credentials in base64 format

Use it as environment variables.

Volumes

→ K8S doesn't manage data persistence.
→ physical storage attached to K8S cluster

→ can be local / remote storage (external Harddrive)

Deployment —

(1) blueprint for pods
(2) we create deployments instead of pods
(3) Replication

Stateful set

(1) for stateful apps like DB's