⚓

Kubernetes

> 💡 **Is a Container Orchestrator tool**

- High availability or no downtime
- Scalability or high performance
- Disaster recovery, backup and restore

Ayuda a hacer Docker mucho mejor.

Master Node

It’s the more important Node. It’s necessary a backup.

Manage:

- **Kubernetes API Server** [it’s a container]
- **Controller Manager** → keeps track of whats happening in the cluster
- **Scheduler** → ensures Pods placement
- **etcd** → Kubernetes backing store

Nodes [worker nodes]

Each one have a Kubelet proccess

It could have different quantity of docker containers

Here is where the apps are running

Virtual Network → Creates one unified machine

▼ Kubernetes Components

Node

A simple server, virtual or physical machine

Pod

Smallest unit in Kubernetes

Abstraction over container

Creates a running environment that manage many containers

Usually 1 App per Pod

> 💡 The idea is that you only interact with the Kubernetes layer

Each pod has an internal IP to communicate.

Pods are ephemeral, usually die; the problem is that the ip changes every time the pod is created

Service & Ingress

Permanent IP address, even if the POD dies the IP still the same

The service is also a load balancer

Lifecycle of Pod and Service not connected

Types:

- **External services** → open the communication from external sources (it's not practical, only for tests)
- **Internal Service** → you specify when create it

Ingress → instead of service the request goes first to ingress and it does the forwarding then to the service

ConfigMap [CM] & Secret

ConfigMap → is a external configuration of your application

> 💡 **CM is for non-confidential data only**

Secret → is like a ConfigMap but the difference is that it's used to store secret data, it's stored in base 64 encoded format. This encoding doesn't make it automatically secure, the secrete components are meant to be encrypted using third-party tools

> 💡 You just connect the CM and the secrets to the Pod

Data Storage → Volumes

Basically attaches a physical storage on a HDD to the pod and that storage could be either on a local machine or on a remote storage [out of the cluster]

> 💡 **Kubernetes doesn't manage data persistence**

Deployment & StateFulSet

Deployment defines **blueprint** for Pods

💡 Is a template for creating Pods

Specify how many replicas you want to have

Abstraction of Pods

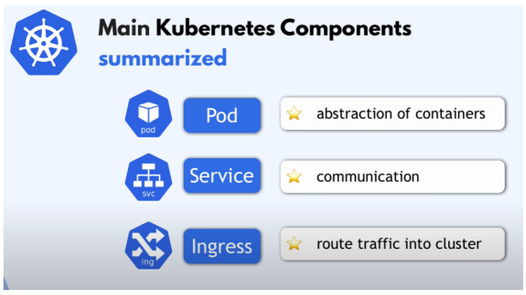
If one Pod die the service will forward the request another one

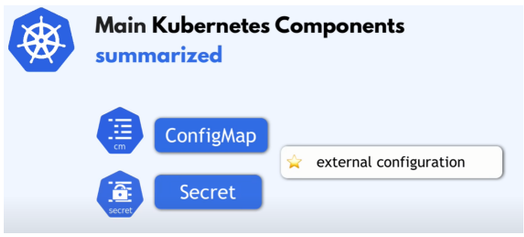
DB can’t be replicated via Deployment!

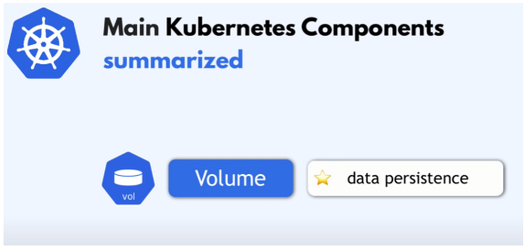
Statefulset is for stateful apps like MySQL, elastic, mongoDB

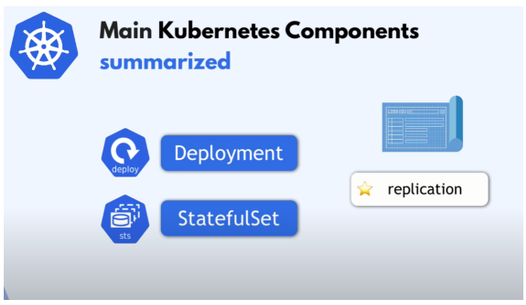
💡 **Deployment** → for stateLESS Apps
StatefulSet → for stateFUL Apps or Databases

💡 Is a common practice to host database applications outside of the Kubernetes cluster and just have the deployments or stateless applications that replicate and scale with no problem inside









▼ Configuration

Deployment

```
apiversion: apps/v1
 kind: Deployment
 metadata:
   name: my-app
   labels:
     app: my-app
 spec:
   replicas: 2
```

```
    selector:
     matchLabels:
        app: my-app
   template:
     metadata:
        labels:
          app: my-app
      spec:
        containers:
         - name: my-app
            image: my-image
            env:
              - name: SOME_ENV
                value: $SOME_ENV
           ports:
                containerport: 8080
```

3 Parts of a K8s Configuration File

1→ Metadata

2→ Specification

3→ Status: is automatically generated and added by Kubernetes

Deployment

```
apiversion: apps/v1
 kind: Deployment
 metadata:
   name: nginx-deployment
   labels:-
 spec:
   replicas: 2
   selector:-
   temp late:-
```

Service

```
apiversion: v1
 kind: Service
metadata:
   name: nginx-service
 spec:
  selector: -
  ports: -
```

Attributes of “spec” are specific to the kind

YAML Configuration Files

- Is human friendly
- **Syntax→** strict indentation [Code Editors have plugins for YAML syntax validation]
- Store the config file with the code or own git repository

Minikube

Is one node cluster where the master processes and the worker processes both run on one node and this node will have a docker container runtime pre-installed so you will be able to run the containers or the pods with the containers on this node

Kubectl

Is a command line tool for Kunernetes cluster

With cube cdl you can basically do anything in kubernetes that you want

- Submits commands to the api server to create components
- Delete components
- etc