**Introduction:**

- Orchestration Tool By Google

- Manage Containers in different ddeployment Environments

- Used for managing Microservices

**Features**

- High Availability

- Scalability

- Disaster Recovery (Backup , restore)

**Architecture**

- **Kubelet** (Primary Node Agent) (Master)

- API Server (Communication mechanism)

- Controller Manager (Services Management)

- Scheduler (Pods placement)

- etcd (Key value pairs for backup management)

**- Node (Slave)**

**virtual Network (Layer of communication of**

**slaves and Master)**

**Kubernetes Terms**

1. Pod : Smallest unit which is abstraction of container

- Each pod has internal IP address for communication

- IP changes when pod dies and recreates

- Can die frequently

2. Service : A point of contact b/w D/B and service

3. Ingress : Port Forwarding of request to pods from outside or inside

4. Config Map : A file which contains relevant data for configuring change in

applications (non - confidential data)

5. Secret : Same as 4. but for confidential data encrypted in base64 (not safe)

6. Volume : Physical Storage attached to pod ( can be local or remote)

**Note: Kuberenetes doesn't manage data persistence. It should be taken care by yourself.**

Deployments (Abstraction of pods) :

Replicas are maintained for each pod. (Can be defined)

Load Balancer is present to manage traffic.

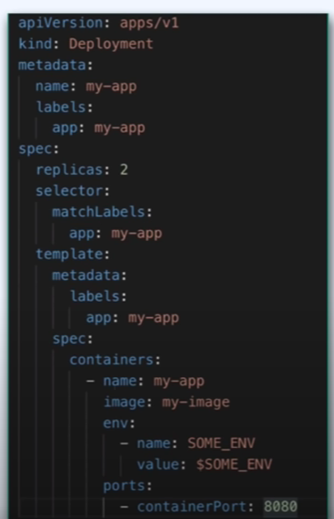
Container -> Pods -> Deployment

7. StatefulSet : For stateful/db apps for managing read and write in database.

(DB's are generally hosted outside of the kubernetes cluster)

**Kubectl : (Command line interface) to manage deploments**

1. Deployment Configuration YAML file :



Main Parts of this file (Deployment/Service) :

1. Metadata : Name and Labels of applications
2. Specifications : Configuration of the component
3. Status : Automatically generated (not in the YAML file) (comes from etcd)

Minikube : Tool to learn kuberenetes

(Run a master/slave on same machine)

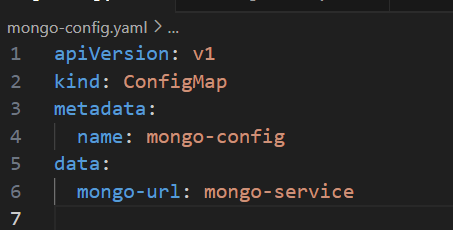
KubeCTL commands:

* kubectl get node (Details of nodes in system ): Note that nodes is the collection of clusters

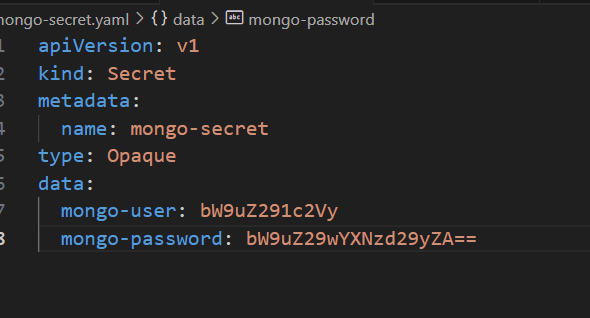
Configuration Files required:

1. ConfigMap : To configure DB endpoint
2. Secret : DB username and password
3. Deployment file for DB with Ingress (Internal)
4. Deployment file for WebApp with Egress (External)

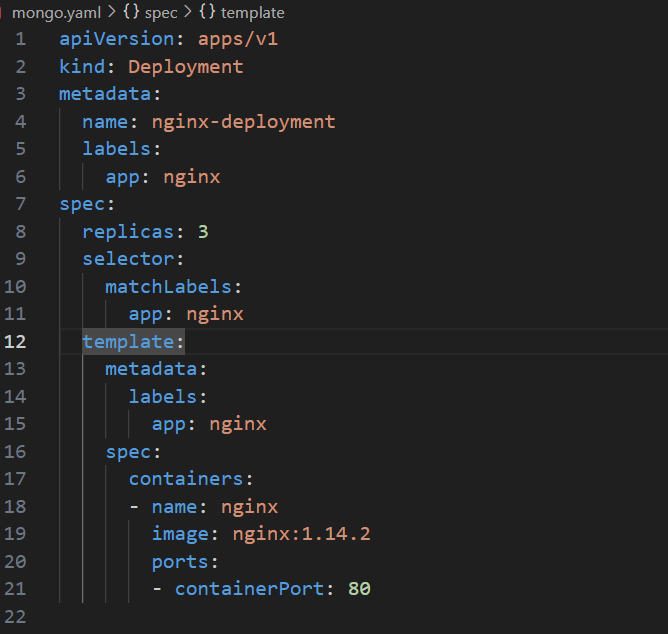
Configmap:

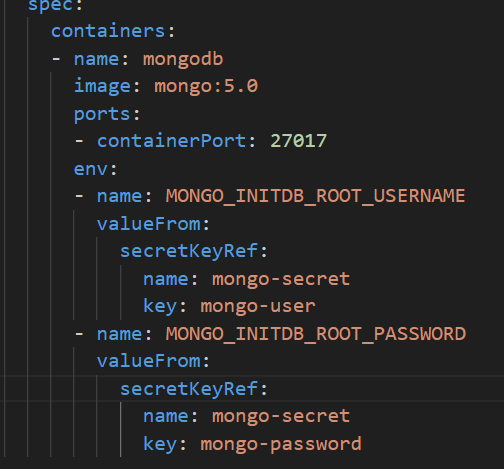


Secret File:



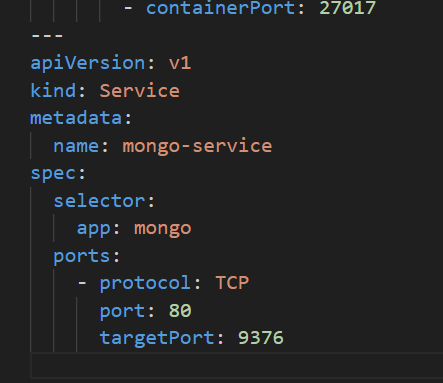
Deployment File





Understanding Deployment File :

1. Template : Configuration of the pod which we will deploy
   1. Spec : Config
   2. Containers : Container inside the pod
   3. Env : Set all environment variables inside a container
   4. Look at syntax above for careful understanding
2. Labels: Key value pairs for identifiers for pods. Identify replicas using labels



Understanding Service File

Name of Service should be same as end point

In Spec, selector , app should match with mongo

Target port should be same as container port in deployment

Port is the port where your service is forwarded.