**Kubernetes:**

* **Open source container orchestration tool, developed by google**
* **Helps manage containerized apps in different deployment environments (local, virtual, cloud, hybrid)**

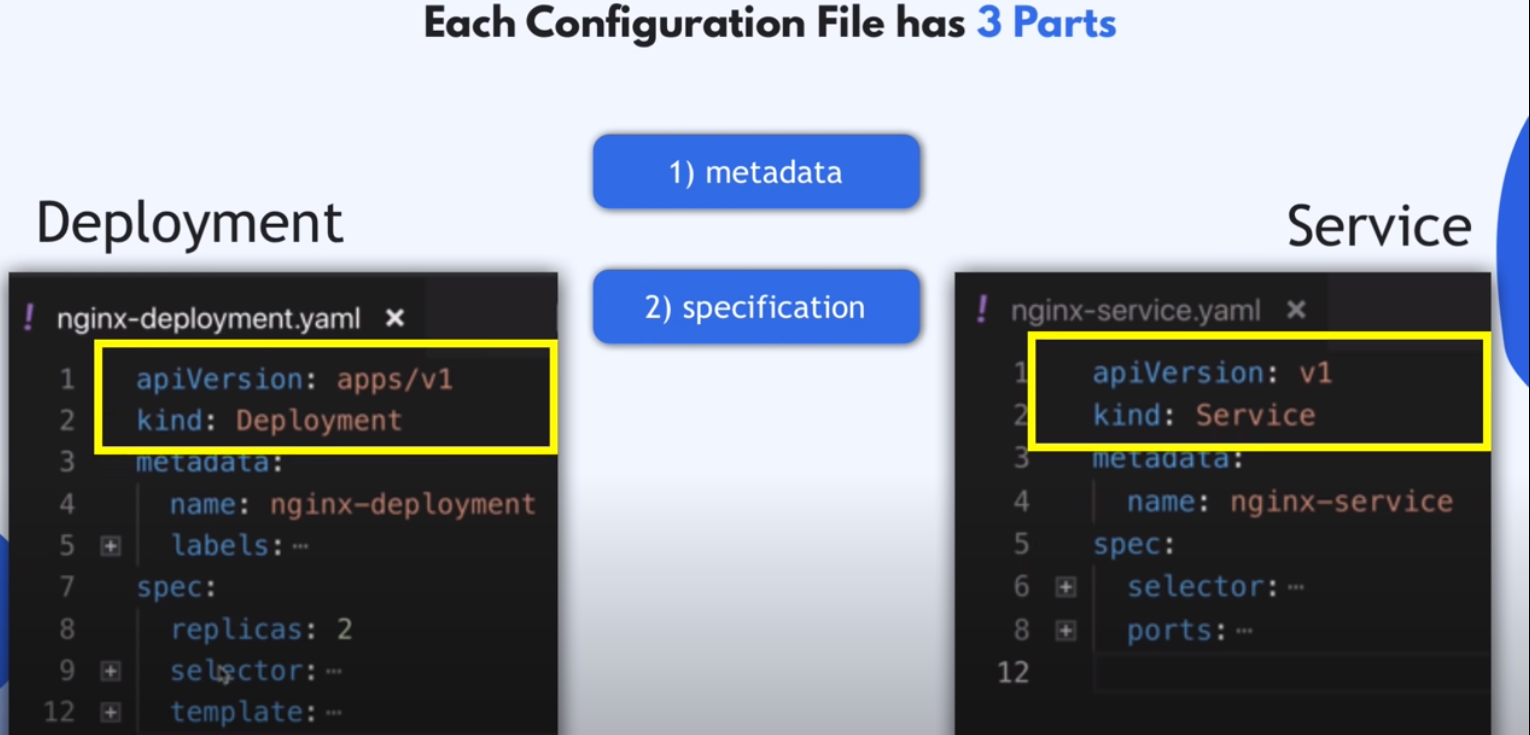
**What it solves:**

* **Trend from monolith to microservices**
* **Increased usage of containers**
* **Demand for a prorper way of managing those hundreds of containers**

**Kubernetes Architecture:**

* **Master Node:**
  + **Brain of kubernetes cluster**
  + **Controls the entire system & has components like,**
    - **Api server - accepts & process requests from users & other components (k8s)**
    - **Etcd - distributed key-value store, stores cluster’s configuration data**
    - **Scheduler - decides where to run containers (ensures pods placement)**
    - **Controller manager - ensures desired & current state of cluster matches (tracks cluster)**
* **Worker Nodes (minions):**
  + **Executes the containers**
  + **Each node runs several services like,**
    - **Kubelet - communicates with master node & manages containers on the node**
    - **Container runtime - s/w responsible for running containers (docker, containerd)**
    - **Kube-proxy - manages n/w communication b/w containers on different nodes**
* **Pods:**
  + **Smallest deployable units in kubernetes**
  + **Abstraction over container**
  + **Usually 1 application per pod**
  + **Each pod gets its own IP address**
  + **Groups of one/more containers that are scheduled together on same worker node**
  + **Containers within a pod share same n/w namespace & communicate via localhost**
* **ReplicaSets/Deployments:**
  + **Ensures specified no.of identical pod replicas are running at all times.**
  + D**B can’t be replicated via deployement**
  + **Used to scale applications**
  + **Deployment is for stateless apps**
  + **For stateful apps use statefulSet (it is difficult to deply stateful apps)**
  + **Deployment = a template for creating pods**
* **Services:**
  + **Enable communication & load balancing b/w pods**
  + **Provide a stable IP address & DNS name for pods, even if pods are replaced/rescheduled**
* **Volumes:**
  + **Enable data persistence for containers**
  + **Allows data to be stored & shared between pods**
  + **It can be local/remote**
* **ConfigMaps (non-confidential data) & Secrets (confidential data):**
  + **Manages configuration data & sensitive information (passwords), which is consumed by pods**

**Kubernetes Configuration:**

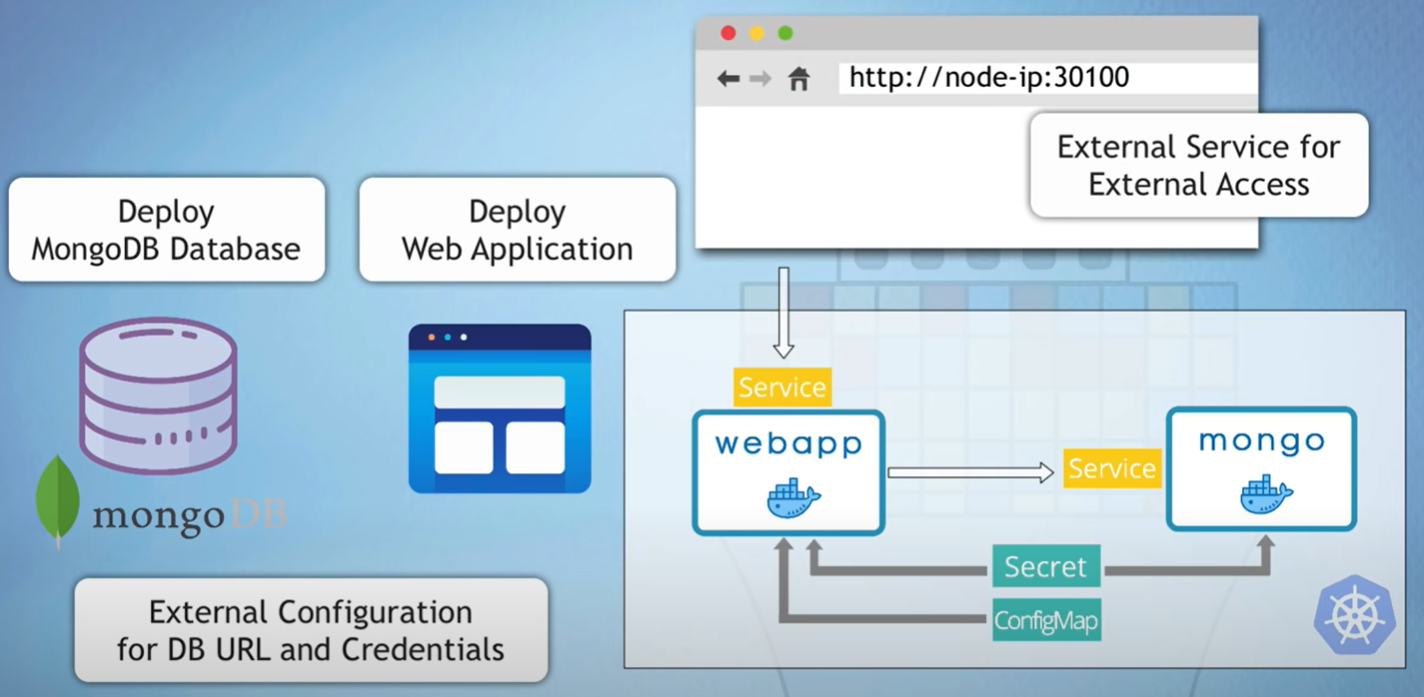
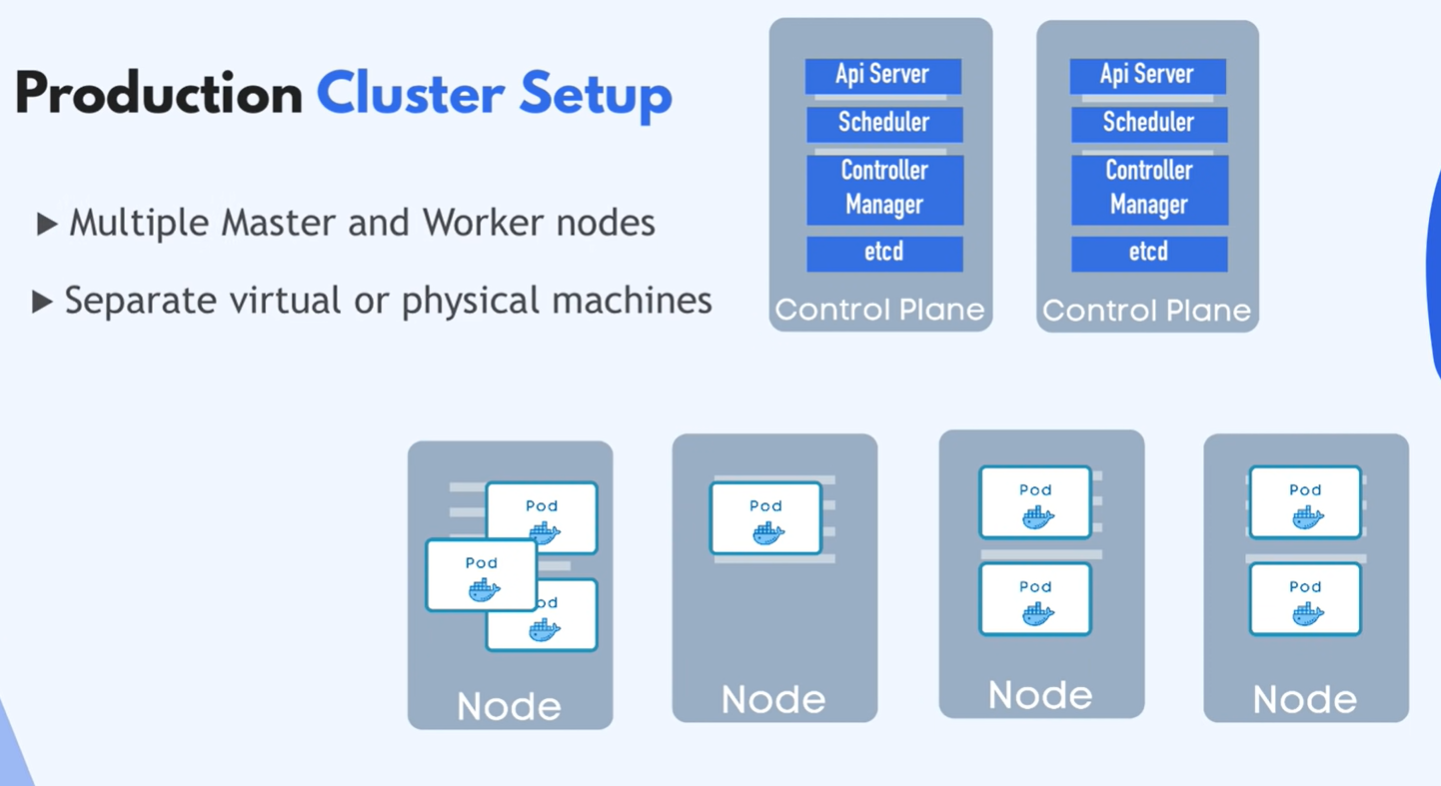
* **Kubernetes client is available as CLI (kubectl) & UI**
* **Parts of k8s config file,**
  + **API version & kind**
  + **Metadata**
  + **Specification**
  + **Status**
* ****

**Minikube:**

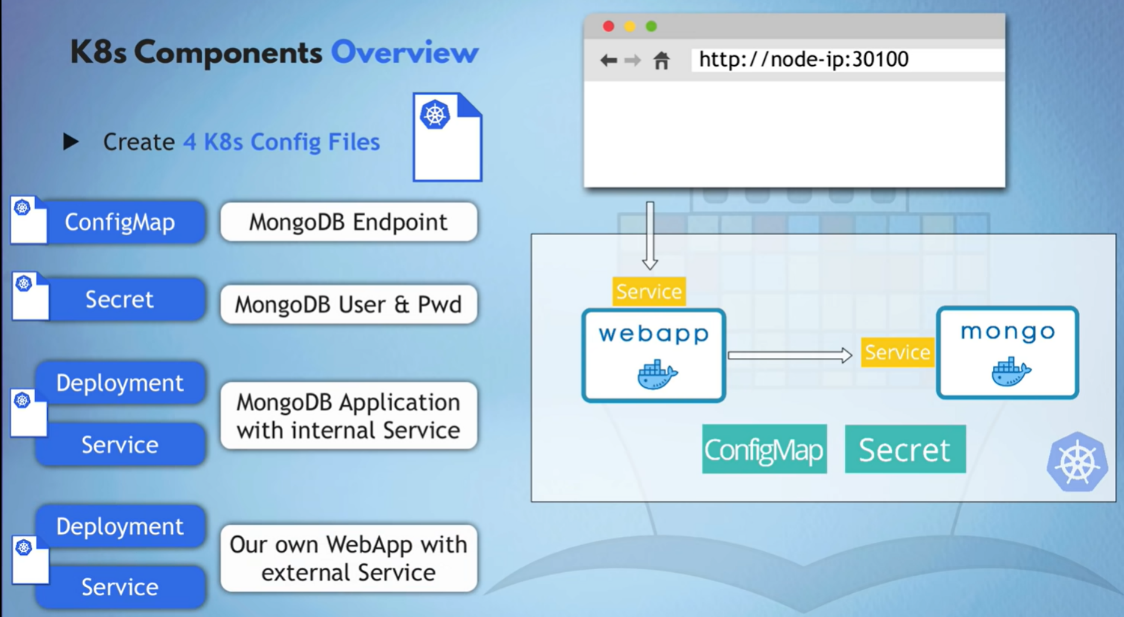
* **It is a tool that enables developers to run a single-node kubernetes cluster on their local machines.**
* **It allows us to experience & experiment with kubernetes without the need for a full-fledged production cluster.**
* **Docker is preinstalled, we can directly run it**
  + **Minikube runs as docker container**
  + **Docker inside minikube to run out application containers**
* **Only for startup/deleting the cluster**

**Kubectl:**

* **It is a CLI ued to interact with k8s clusters.**
* **It is main control tool for managing & deploying apps on kubernetes**
* **Provides a way to create/update/delete k8s resources, inspects cluster’s current state & interact with running apps**
* **It is for configuring minikube cluster**

****

**Config Files:**

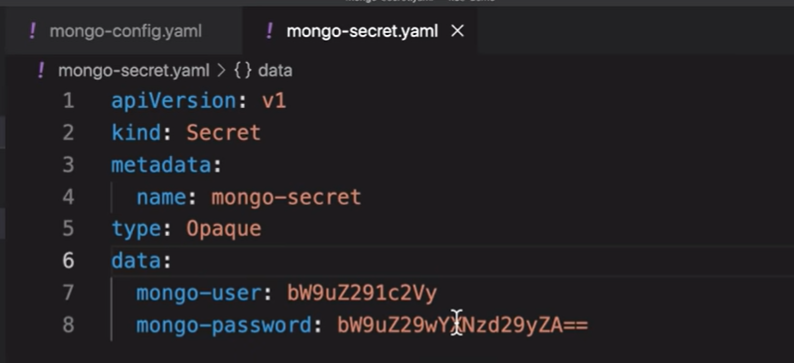
****

**ConfigMap file:**

****

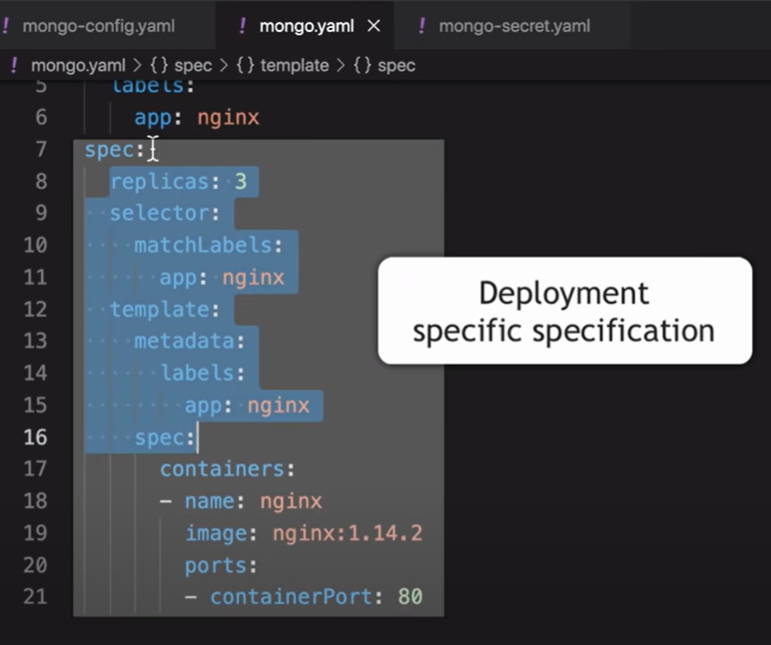
**Secret file:**

* + **We can’t simply give plain password/username. So, convert the plain text to base64 format using following commands,**
    - **echo -n mongouser |base64**

****

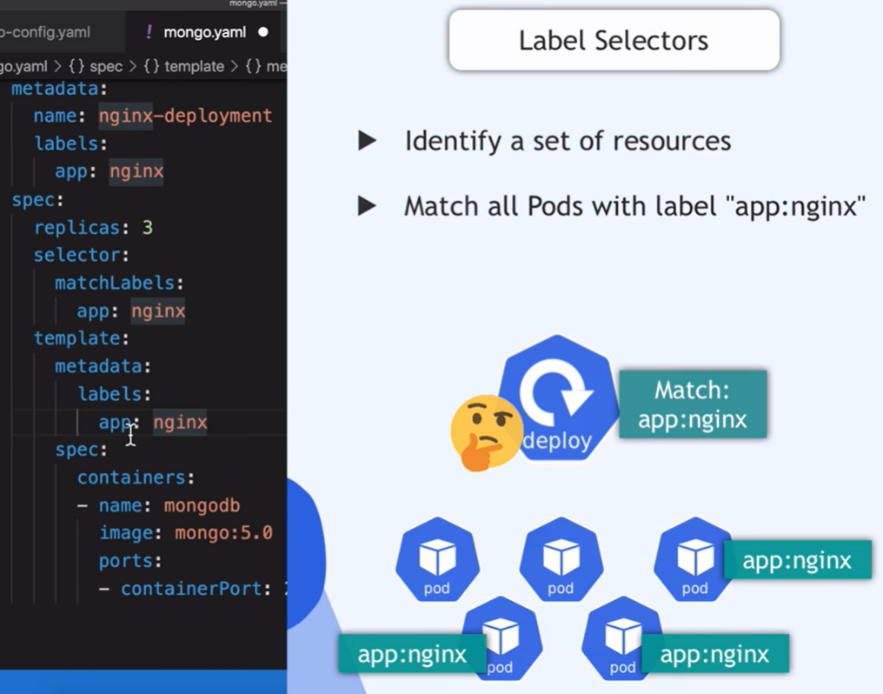
**Deployment & service file:**

* + **Deployment & Service in 1 file, because they belong together**
  + **It is complex than remaining config files**

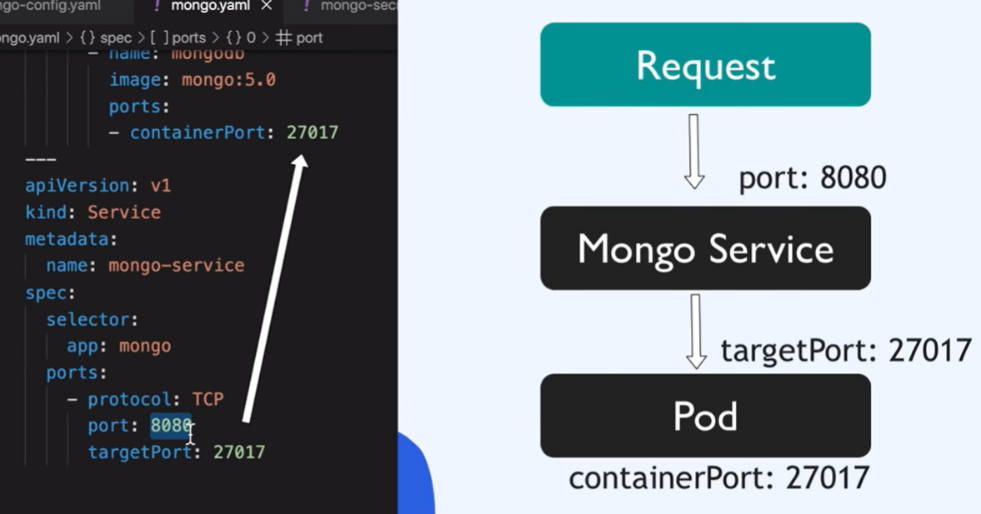
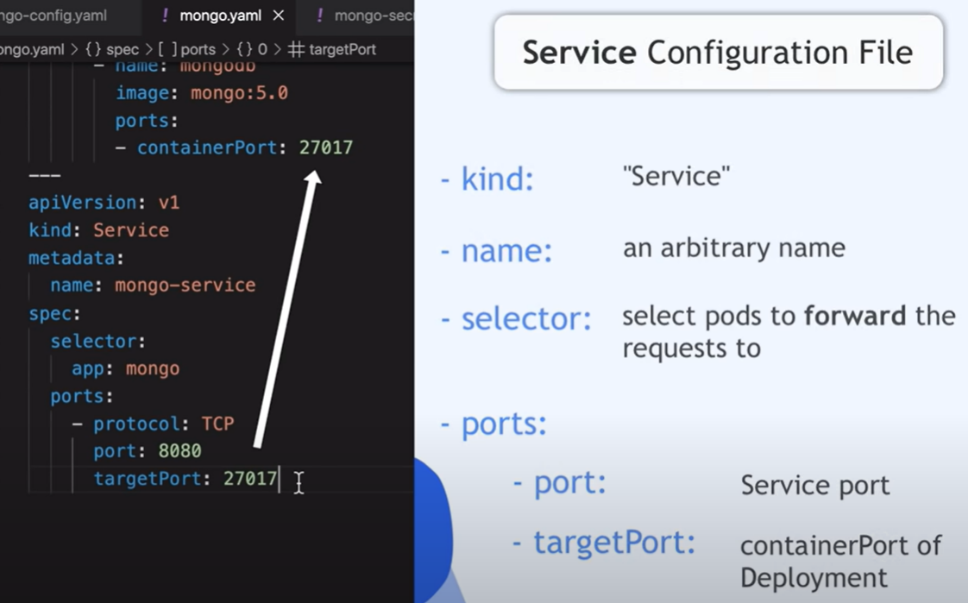
** A screenshot of a computer

Description automatically generated**

**Label & lebel selectors:**

** **

**Service Files:**

** **

**Commands:**

**minikube start --driver docker - starts minikube**

**minikube status - to see status of minikube cluster**

**kubectl get node - lists all nodes in cluster**

**kubectl get pod**

**Notes:**

* **In production always maintain backup of master node, because if we lost it then we can’t access its worker nodes.**
* **DB can’t be replicated via deployement**
* **Store k8s config file with our code**