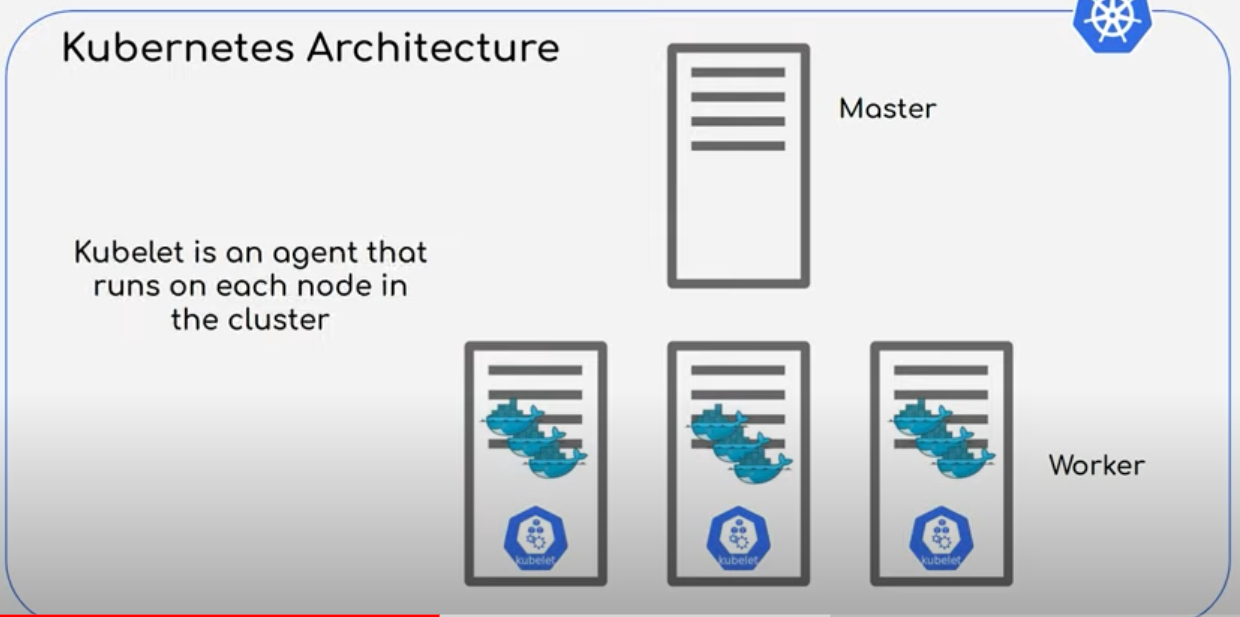
**Kubernetes:**

<https://www.youtube.com/watch?v=lXqqhyCPWC0>

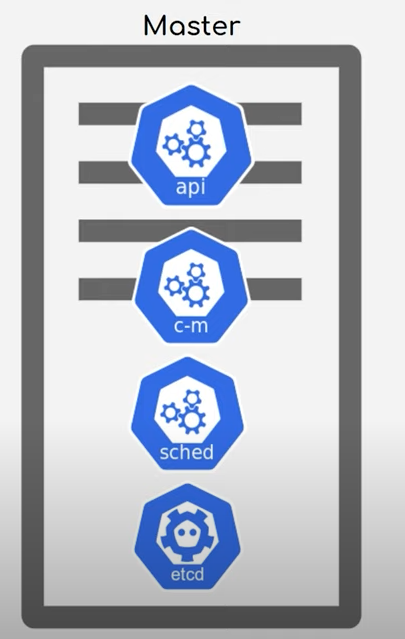


**Modules of Master Nodes:**

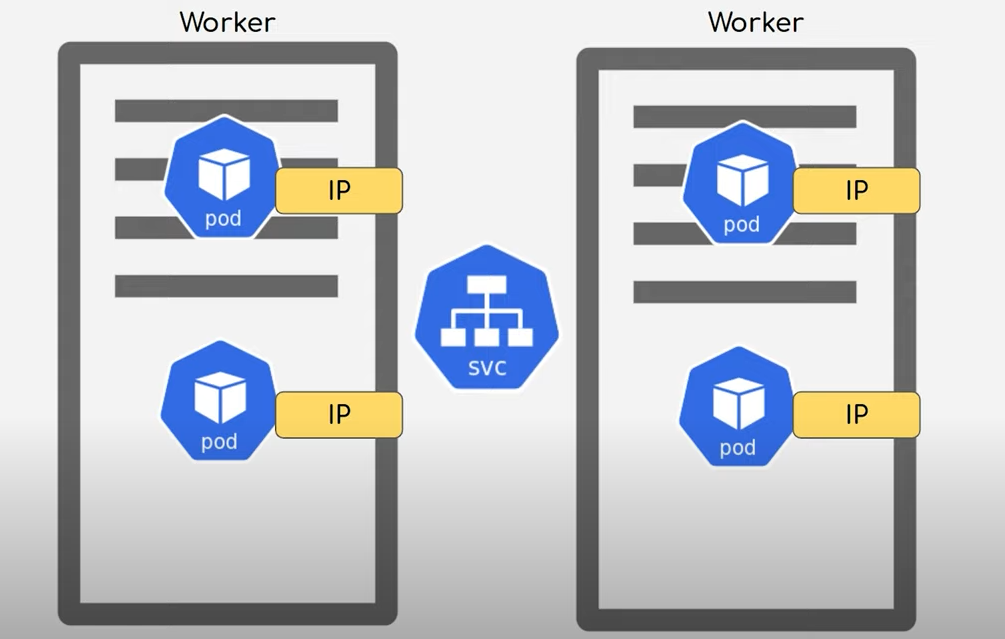
* **API** – Interact with POD/Worker using UI, API , CLI
* **CM – control manager.**

1. Node controller
2. Replication controller
3. End point controller
4. Service account & Token Controller. (namespace)

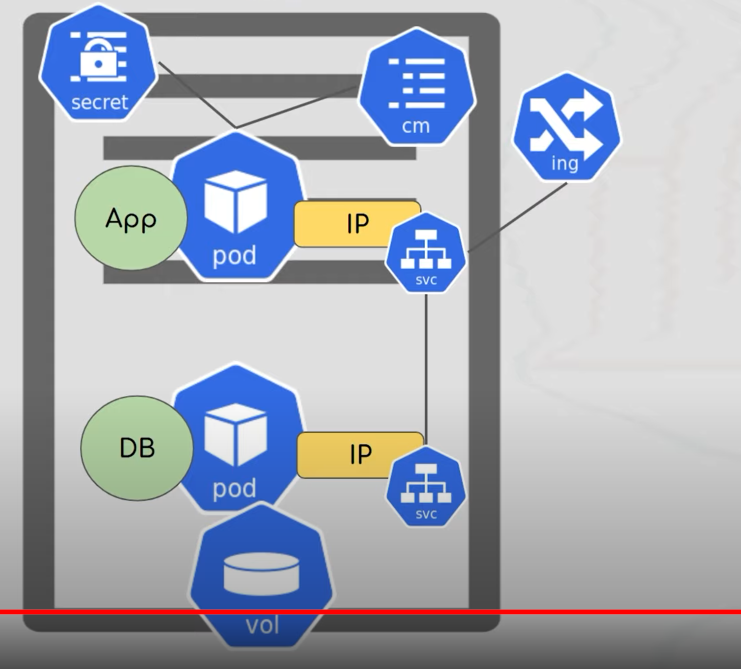
* **Sched** – manage pod placement.
* **Etcd - Key** value pairs to store cluster data.



**POD,cluser IP & Service**



* **Pod –** Abstraction on container. 1 pod one container.
* **Cluster IP -**Internal
* **Service –** when Pod goes down, new IP is substituted, the service will be intact.2
* **Deployments-** Manages entire cluster by yaml file



**Service**

1. **External service -- Ingress Service** , **Node Port,** **Load balancer** .DNS Service.
2. **Internal Service -- Cluster IP.**

**Config Map**

To store application config values

**Secret**

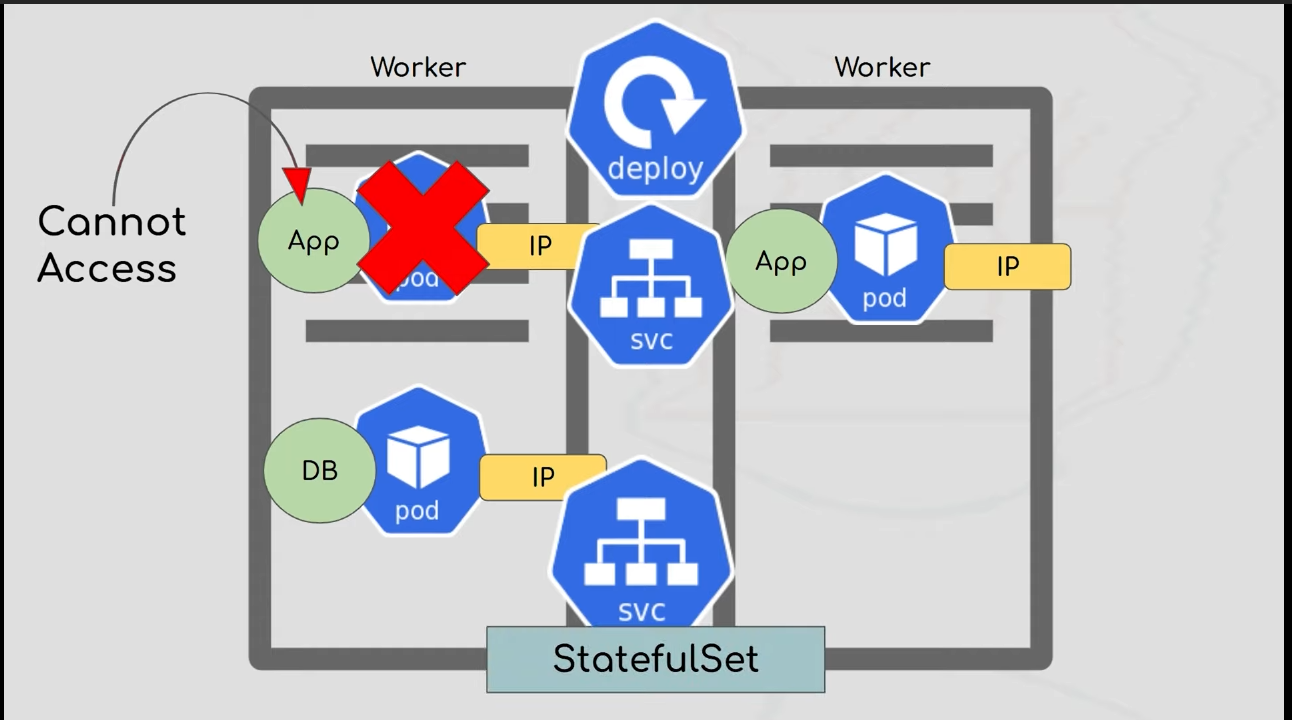
store the Password and other secrets

**Volume**

To store persistence Value.

Local /Remote Volume.

**Deployment**



* **Deployment**

Replica Set - Scalability

Deployment image

* **Stateful Set -** For Database Replica set won’t be useful as the data won’t be persisted. We use stateful set.

**YML:**

* **API Version**
* **Kind**
* **Metadata**
* **Spec**
* **Status -only visible in command prompt**

Text

Description automatically generated

**Service :**

Text

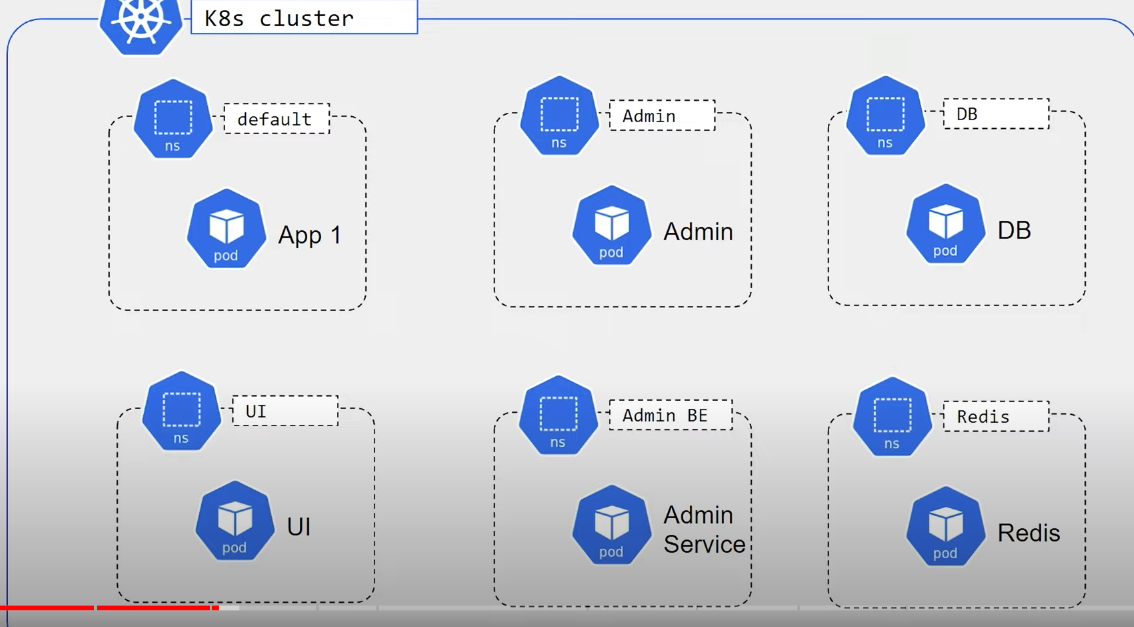
Description automatically generated

**Namespace**

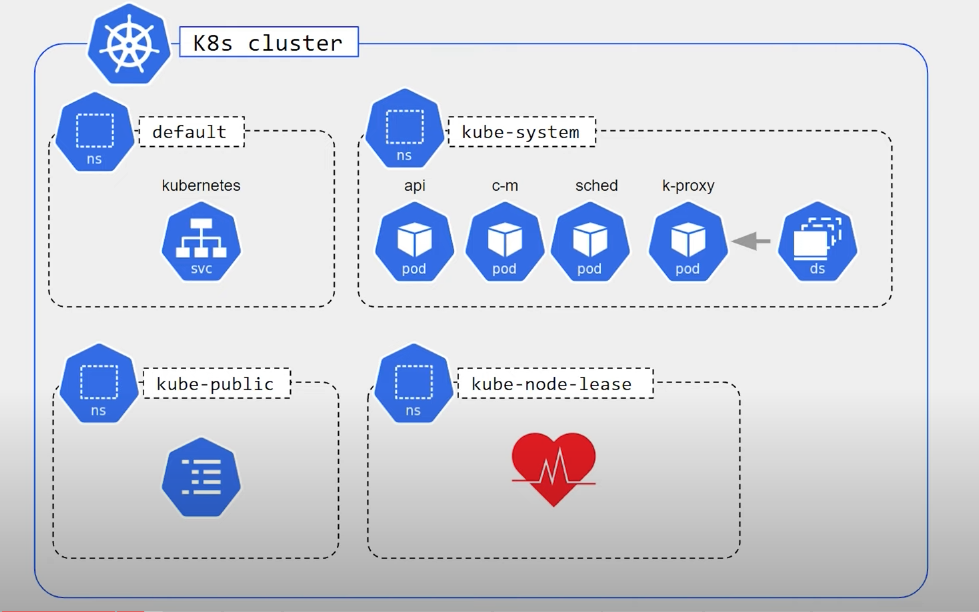
* **Kubectl create namespace <mynamespace>**
* **Kubectl get namespace.**
* **Kubectl apply -f ./deploy.yml –namespace=<mynamespace>**
* **Kubectl get deployment -n <mynamespace>**

**Yml -- Metada**

**Namespace:<my name space>**



**Default cluster -4 namespace available K8s**



**Health Checks:**

Two probes:

1. Liveness probe
2. Readiness Probe

* **Liveness ,**

Text

Description automatically generated

* **Readiness Probe**

Text

Description automatically generated

1. **Command Probe**

Graphical user interface, text, application

Description automatically generated

1. **Http Probe**

A picture containing text

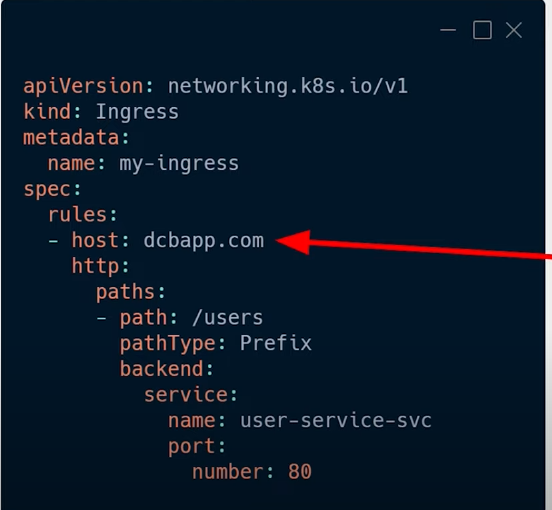
Description automatically generated

1. **TCP Probe**

Text

Description automatically generated with medium confidence

**Ingress Service**



We can define multiple paths in the ingress via rules.

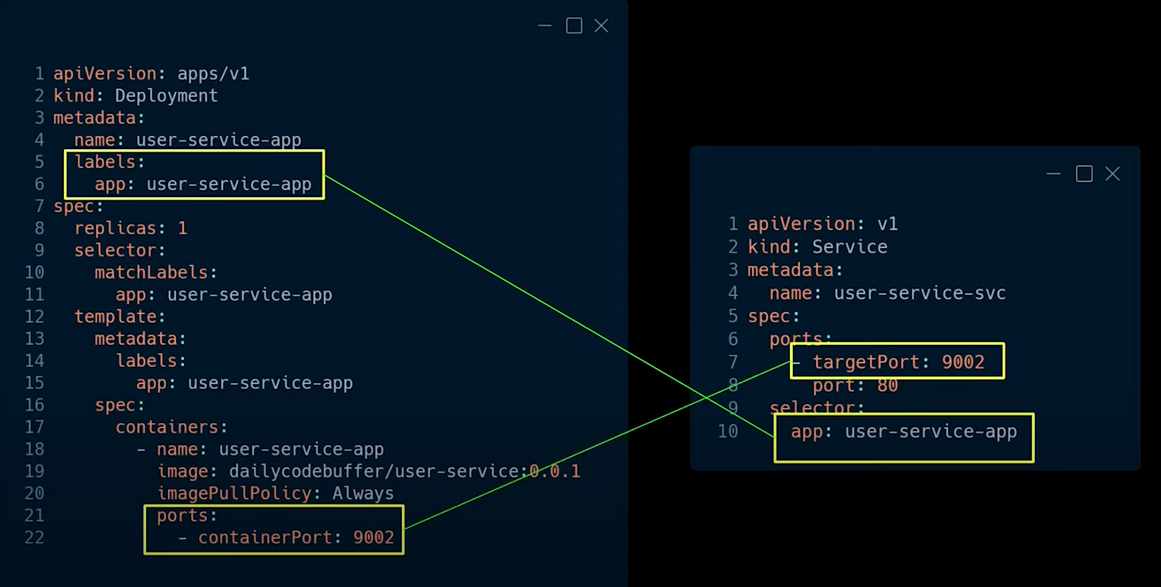
Hostname we have to define in the windows host file.

**>minikube addons enable ingress (nginix)**

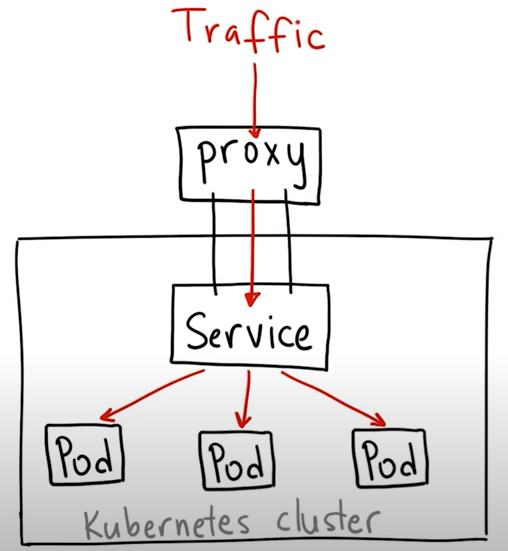
**Statefullset Kubernetes (Not a deployment)**

* **Sticky Identity**
* **Mast slave concept among the Different pods**
* **1 POD will have Read and Write access ,Others will have Read access**
* **Persisted Volume – should be synchronized to POD storage.**
* **We need to headless service.**
* **Should be mainted by user .**

**Kubernetes services:-**



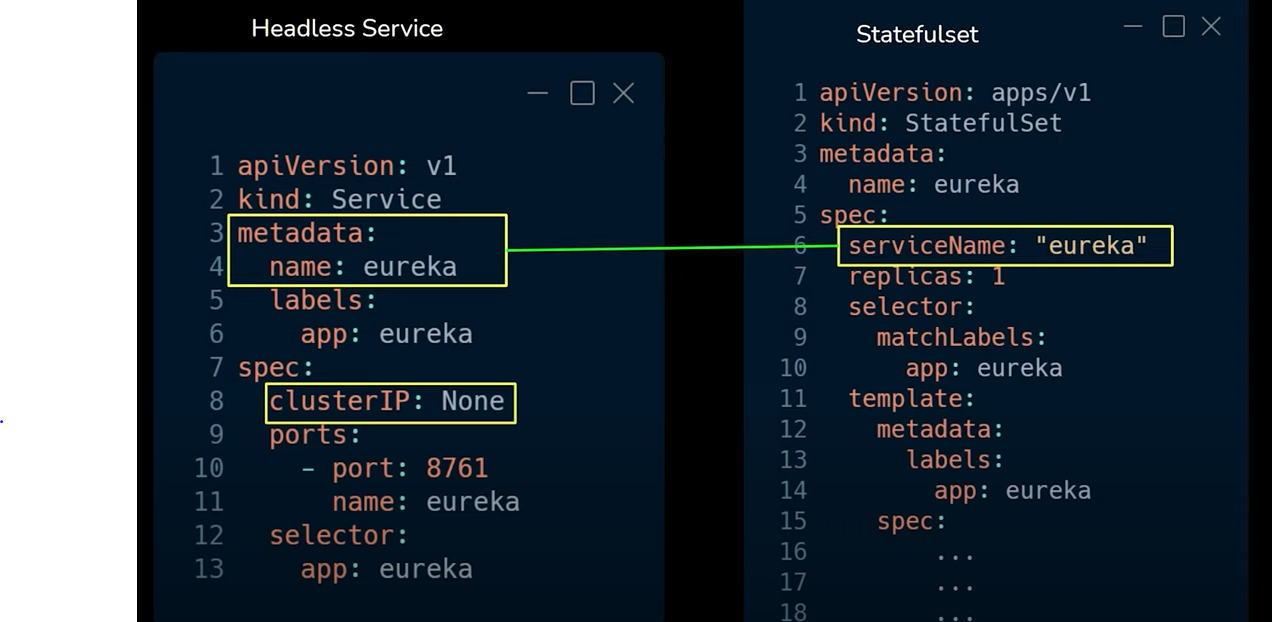
* **Cluster IP Service**



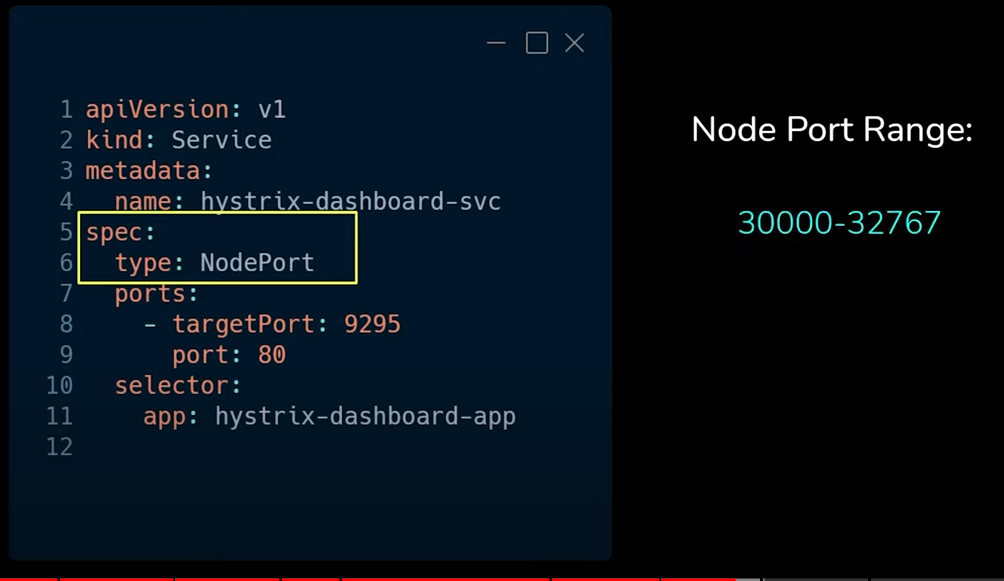
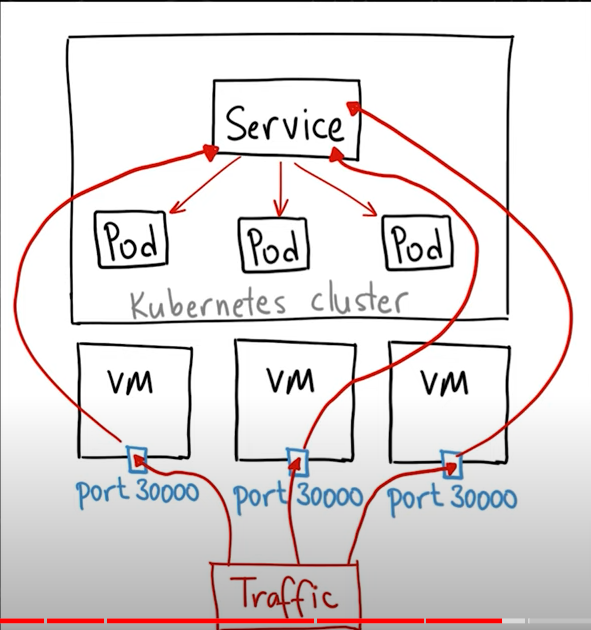
**Cluster IP with ingress controll**



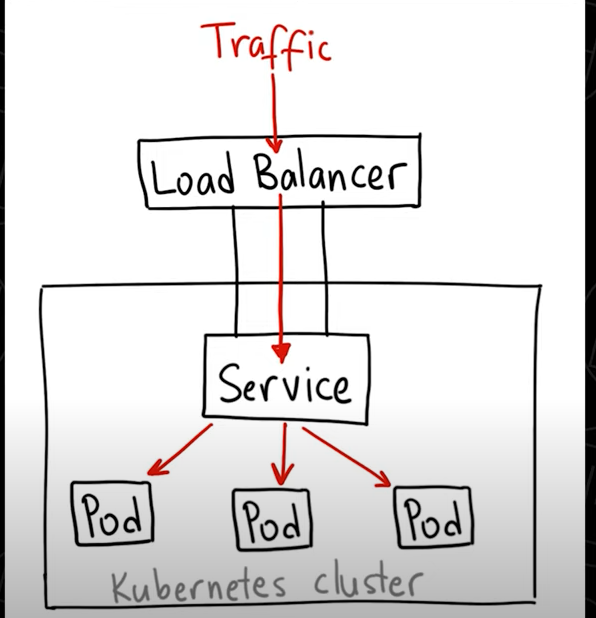
* **Headless Service – Statefull Service, DNS name,Internal Cluster service without IP Address**



* **NodePort Service -same port for all PODS ,External service**

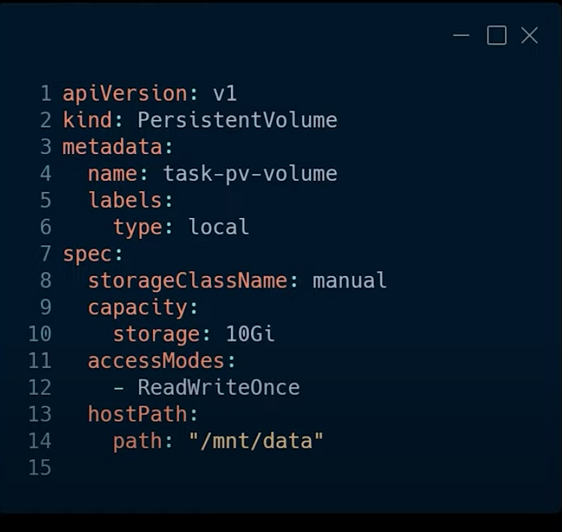
* **Load balancer Service (Single port for All Pods) same as Node Port**



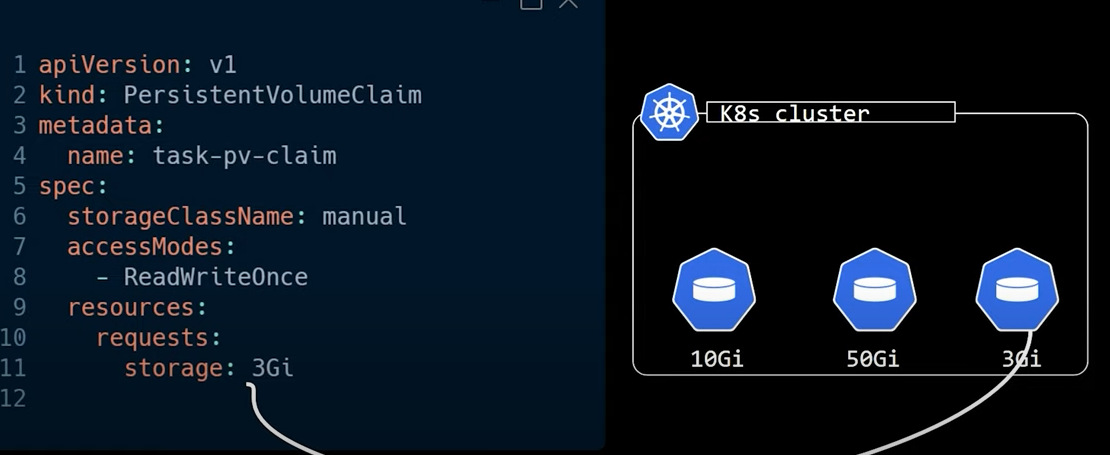


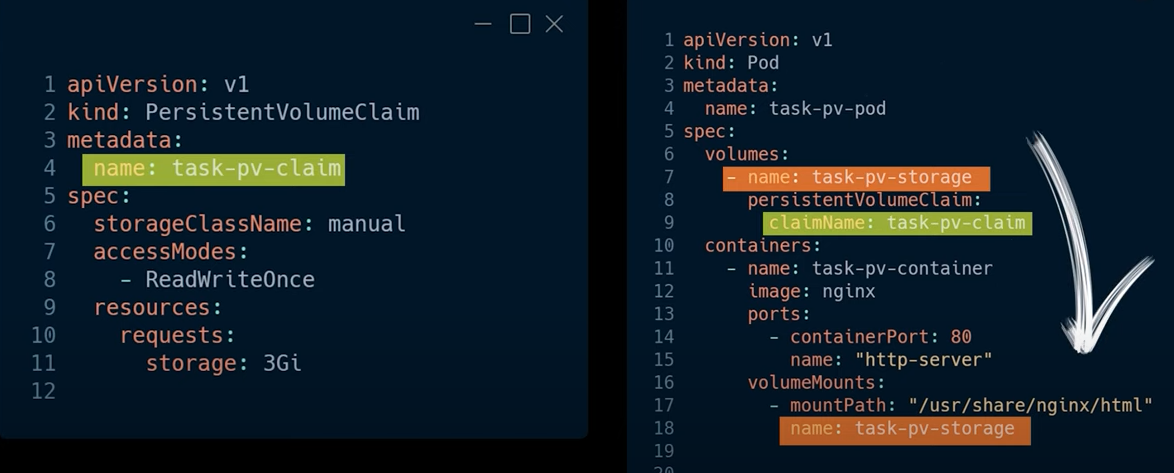
**Volumes:**

* **Should be independent from PODS**
* **Should be available for all nodes**

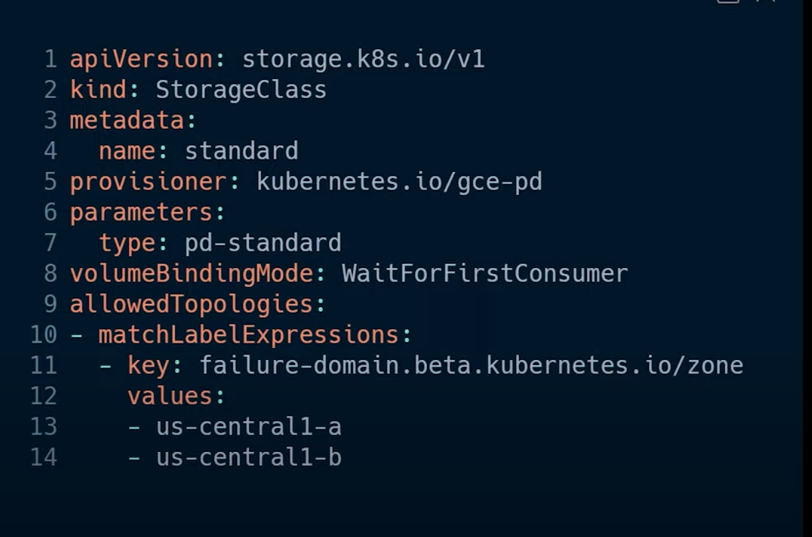


* **Persistence Volume Vs Persistent Volume claim**





* **Storage class**



**Commands:**

Kubectl create deployment nginix-server –image=nginix

1) Kubectl get pod

Kubectl get pod -o wide

2) Kubectl get deployment

Kubectl get replicaset

Kubectl describe pod/deployment

Kubectl get service

kubectl edit deployment

kubectl logs <podname>

kubect exec -it <podname> --bin/bash

kubectl delete deployment <deploymentname>

kubectl apply -f .\<filename.yml>