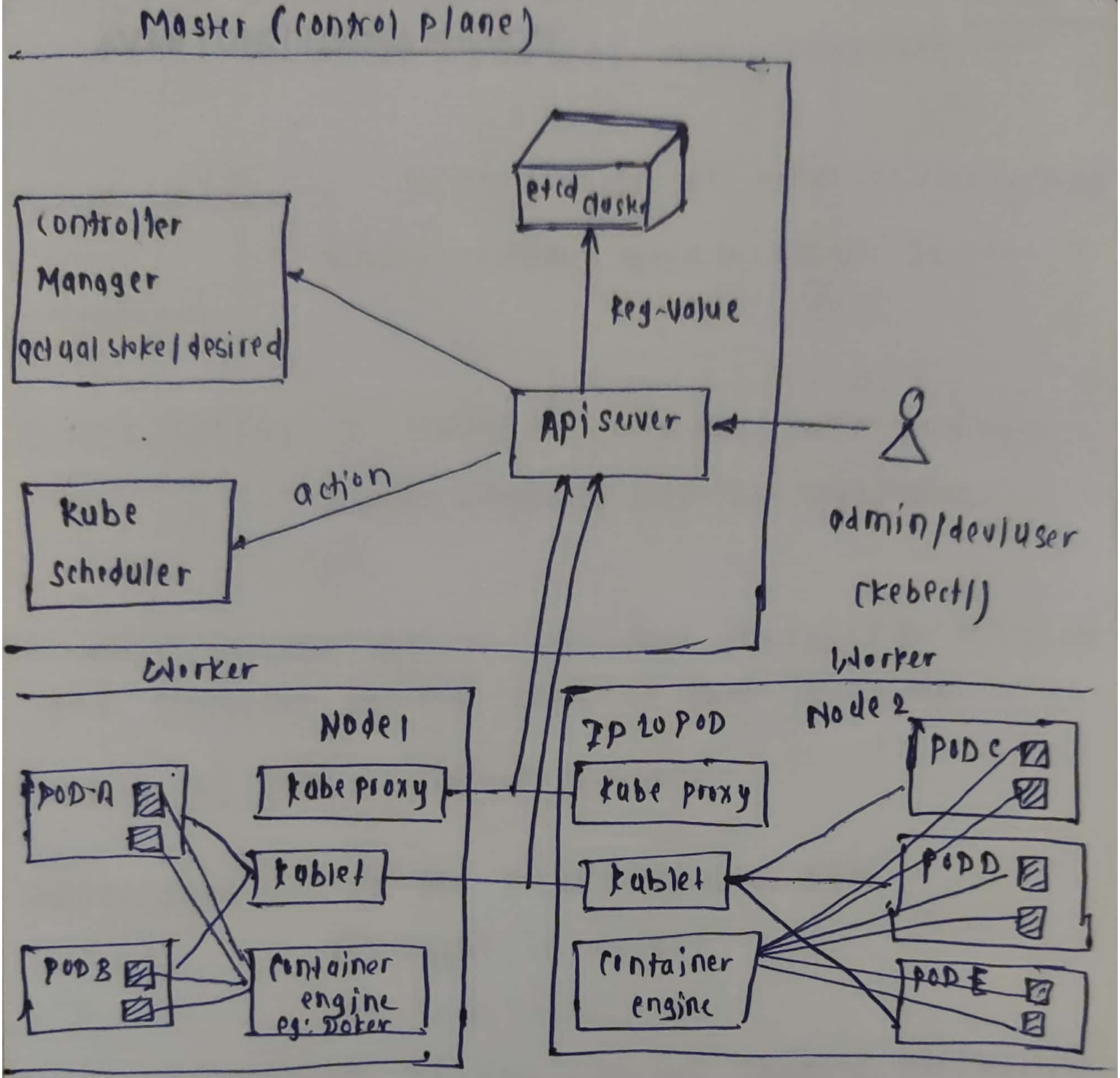

Master (control plane)
etcd clusk
Controller Manager
actual stoke/desired
key-value
Api suver
admin/dev/user
(kebectl)
Kube Scheduler
action
Worker
Node1
POD-A
POD B
kube proxy
kablet
container engine
eg: Doker
Worker
IP to POD
Node 2
POD C
POD D
POD E
kube proxy
kablet
container engine

components of control plane (Master) :-

kube-apiserver   kube-scheduler   controller Manager

etcd

) kube-api-server :- (For all communication)

→ this api-server interacts directly with the user (i-e. we apply yaml or jason manifest to kube-apiserver)

→ This kube-apiserver is meant to scale automatically as per load.

→ kube api-server is front-end of control plane.

2) etcd :-

→ stores metadata and status of cluster.

→ etcd is consistend and high-available store (key-value store)

→ Source of touch for cluster state (info about state of cluster).

etcd has following feature.

① fully replicated :- the entire state is available on every ~~state~~ node in the cluster.

② Secure :- implements automate TLS with optional-client certificate authentication

③ fast :- Benchmarked at 10,000 writes per second.

Kube-scheduler :- (action)

- When users make request for the creation and management of pods, kube scheduler is going to take action on these requests.
- Handles pod creation and management
- Kube scheduler match/assign any node to create and run pods.
- A scheduler watches for newly created pods that have no node assigned for every pod that the scheduler discovers scheduler becomes responsible for finding best node for that pod to run on.
- Scheduler gets the information for hardware configuration files and schedules the pods on nodes accordingly

Controller manager :-

- Make sure actual state of cluster matches to desired state

Two possible choices for ~~controlled~~ manager
① if k8s on cloud, then it will be cloud controll manager.
② if ~~k~~8s on non-cloud, then ~~it~~ will be kube-controller manager.

components of master that runs controller :-

Node controller for checking the cloud provider to etermine if node has been detected in the cloud fter it stops responding

oute controller :- respon~~ding~~sible for setting up network, routes on your cloud.

rvice controller :- responsible for load balancers on your cloud against services of type load blancers.

olume controller :- for creating, attatching, mounting volumes and intracting with the cloud rovider to ochiestrate volume.

Node is going to run 3 important piece of software

① kubelet :- - agent running on node
- Listens to kubernetes master (eg :- pod creation request)
- use port (10255)
- send success fail report to master

Container engine (Docker)
- Works with kublet
- pulling images.
- start, stop containers
- Exposing containers on ports specified in manifest

Kube-proxy

- Assign ip to each pod
- It's required to assign ip adresses to pod (dynamic)
- kube-proxy runs on each node and this makes sure that each pod will get its unique ip adress.

these 3 componanants collectively concists 'nodes'?

POD :- 1. Smallest unit in kubernetes

2. pod is a group of one or more containers that are deployed together on the same host.

- A cluster is a group of nodes
- A clouster has atleast one worker node and master nod
- In kubernet, the control unit is pod not container.
- consist of one or more tighly coupled containers.
- pod runs on node, which is controlled by master.
- kubernets only know about pods (does not know about individual container)
- cannot start containers without a POD
- one pod usually contains one container.

How many types of namespaces?

↳ A way to organize cluster into virtual sub-cluster.

1. Default
2. kube-node-lease
3. kube-public
4. kube-system

they can be helpful when diff team/ Projects share a kubernetes

Kubernetes commands :-

kubectl get pods = All running pods will see

minikube ssh = Logged in to pod

curl ipaddress of pod :

vi pod.yml

kubectl create -f pod.yml

kubectl get pods

kubectl get pods -o wide (everying related to pod will see)

kubectl delete pod nginx

Q. How do you debug a pod?

→ By using kubectl describe pod Nginx = I will get status of everything in pod.

kubectl logs Nginx (Name of the pod)

kubectl get all = list of pods and deployment

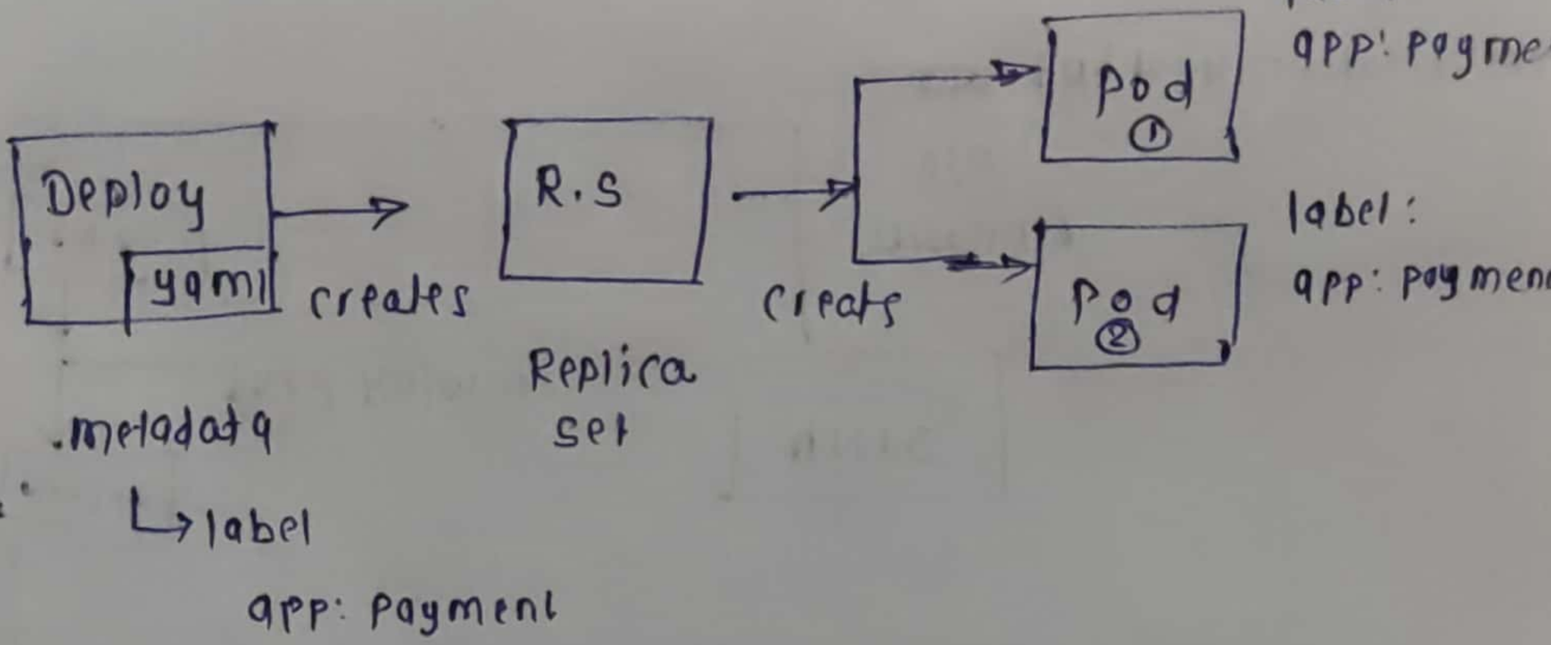
kubectl get all -A = — " — with all namespaces

Deployment.yml → (will creates) replicaset → (will creates) pod

---

service :- ① Load Balancing
② service discovery
↓ labels and selectors
③ Expose to world
→ solving by service

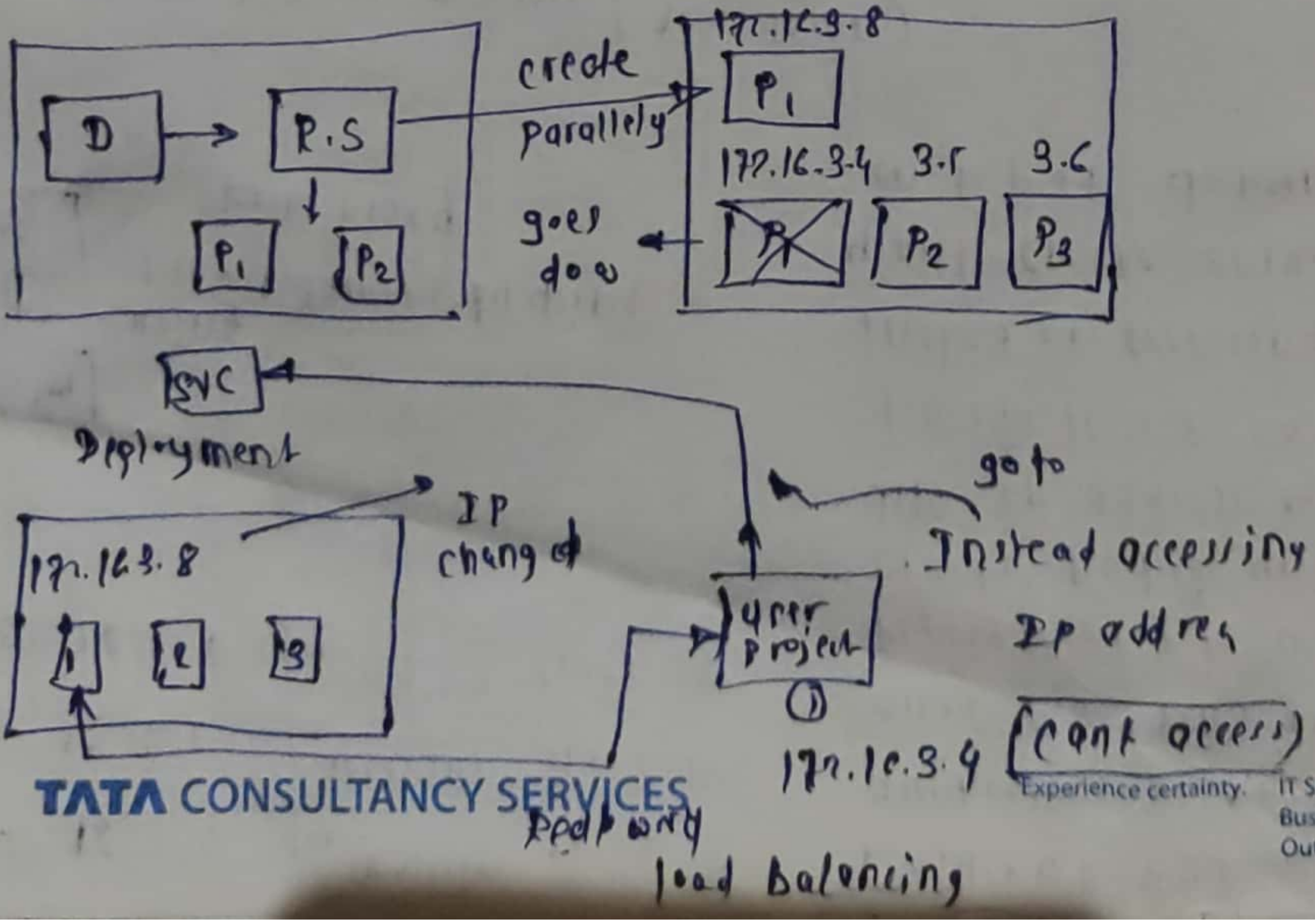
Diagram of kubernetes Deployment and service :



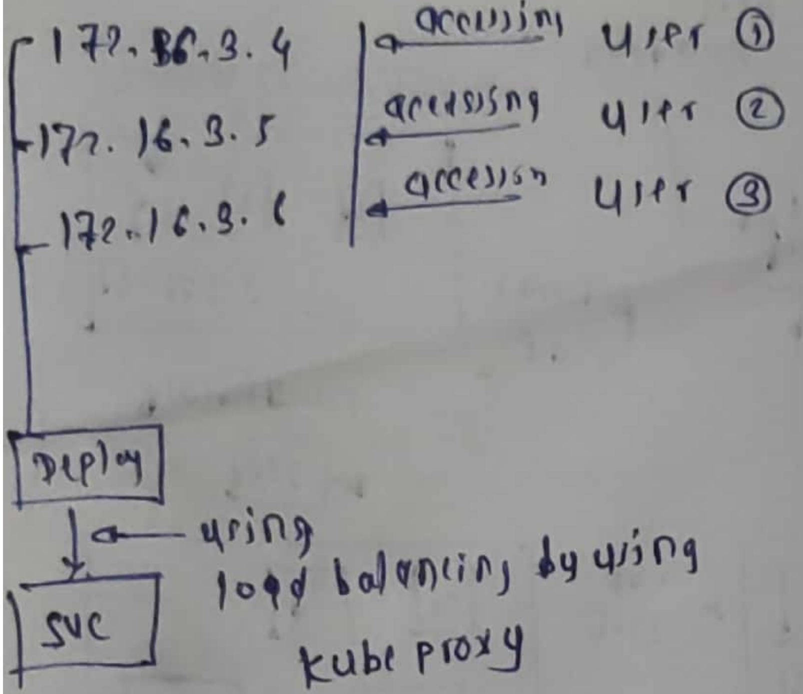
service =

k8s = autohealing

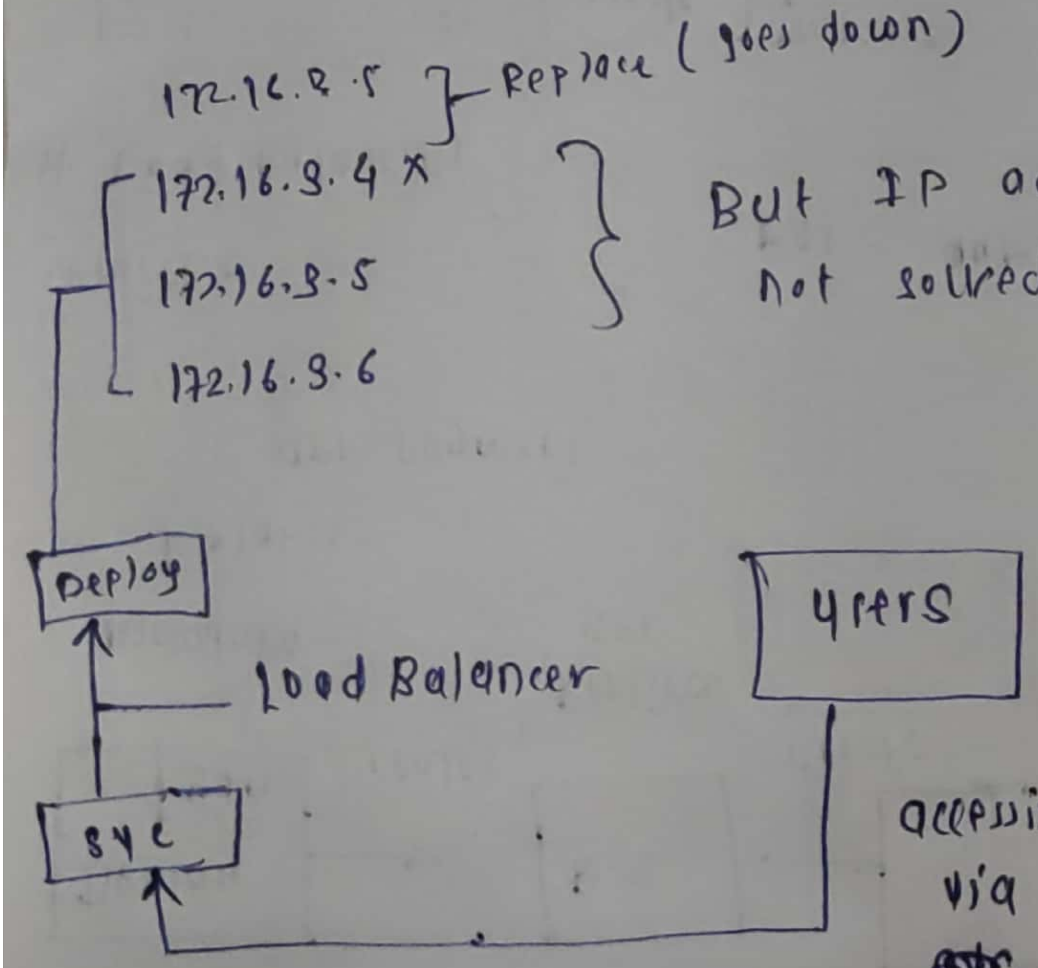
Load Balancing



Kubernetes commands :-



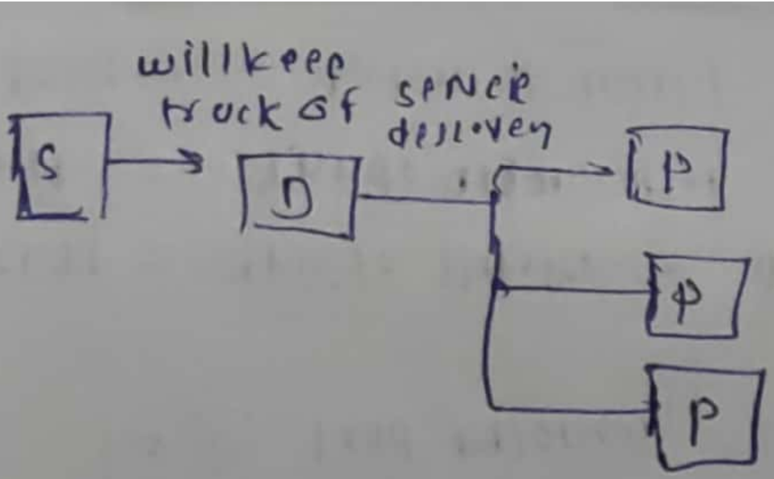
because of k8s autohealing behaviours on of Pod goes down is deployment file will create Pod @ with diff ip address i.e 172.16.3.8 so insted of accessing I.P u will create service on top of depayment



But IP address issue is not solved here

accessing via payment-default-svc

to solve IP address issue there is concept called service descovery



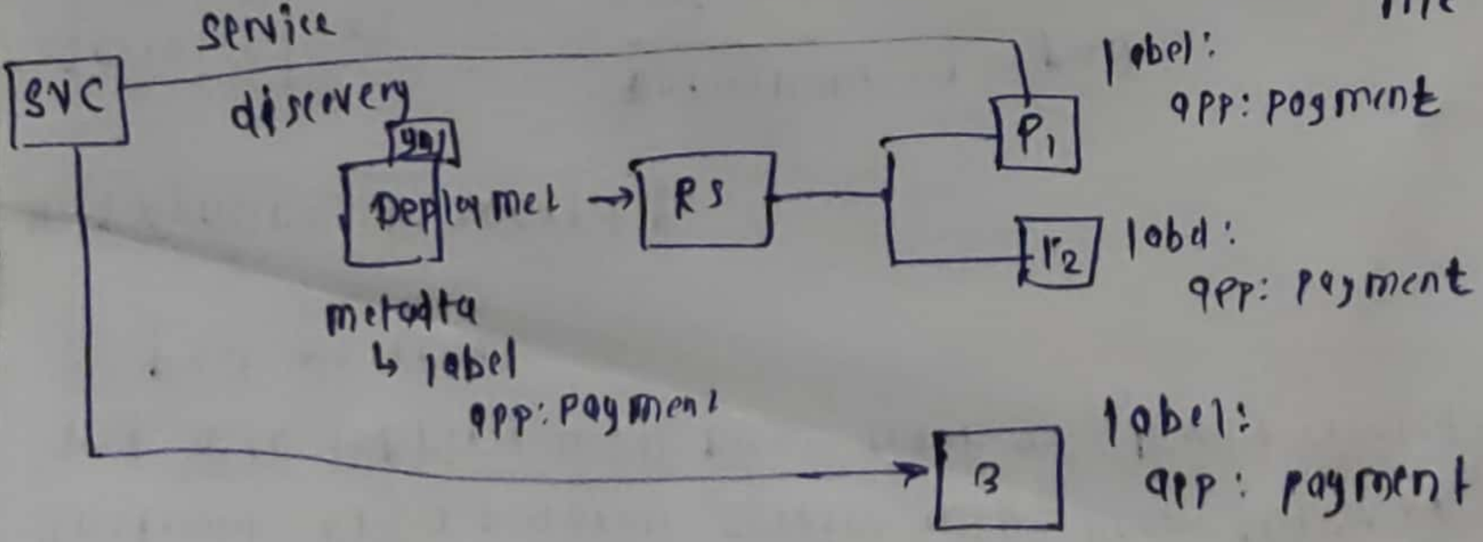
(goes down) change IP adress

IP adress → labels and selectors

Service discovery :-

To fixing I.P adress changéng behavicur of pod (pod goes down then are no of resons behind it) there is a concept called labels and selectors

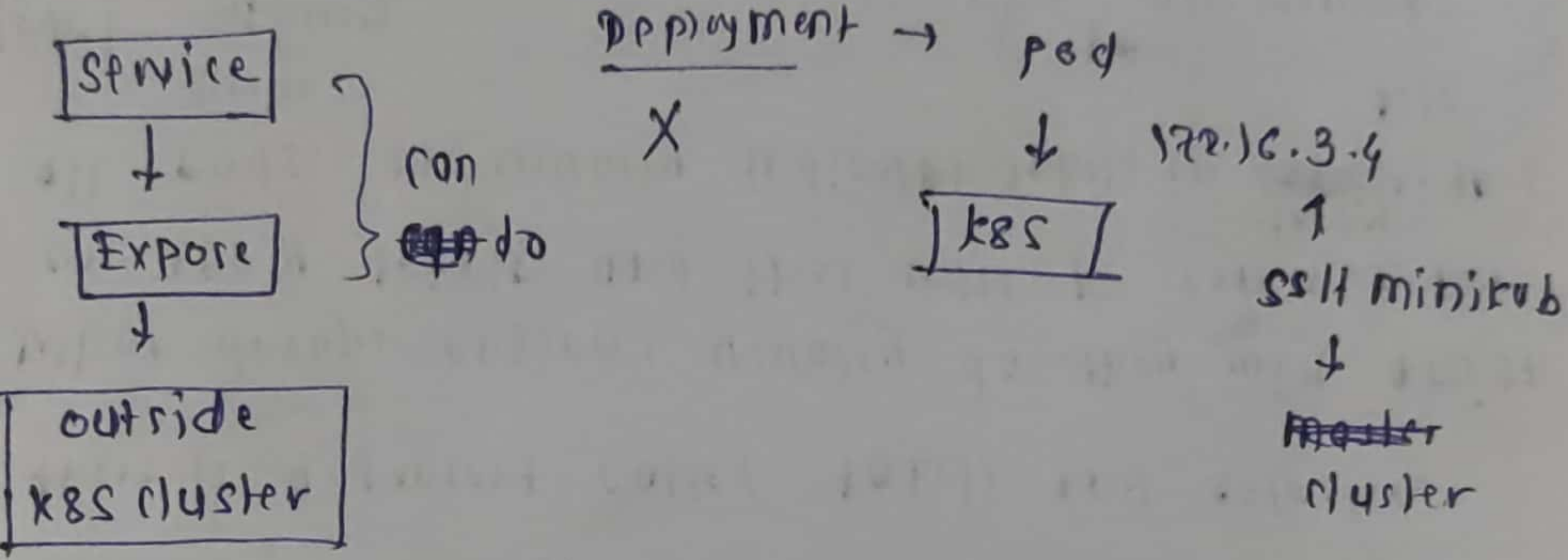
What devops engineer usually do they will assign or apply lebels and that will be common for all pods. We usually indicates labels in deploy.yml file



As of what we learn from service

Instread of I.p adress service will keep track of labe and also lobking will keep track of newly created pod as well.

Exposing to world :-



whenever you are creating service in yaml manifest u can create service in three types

YAML
- ① cluster IP
- ② nodeport mode
- ③ load Balancer

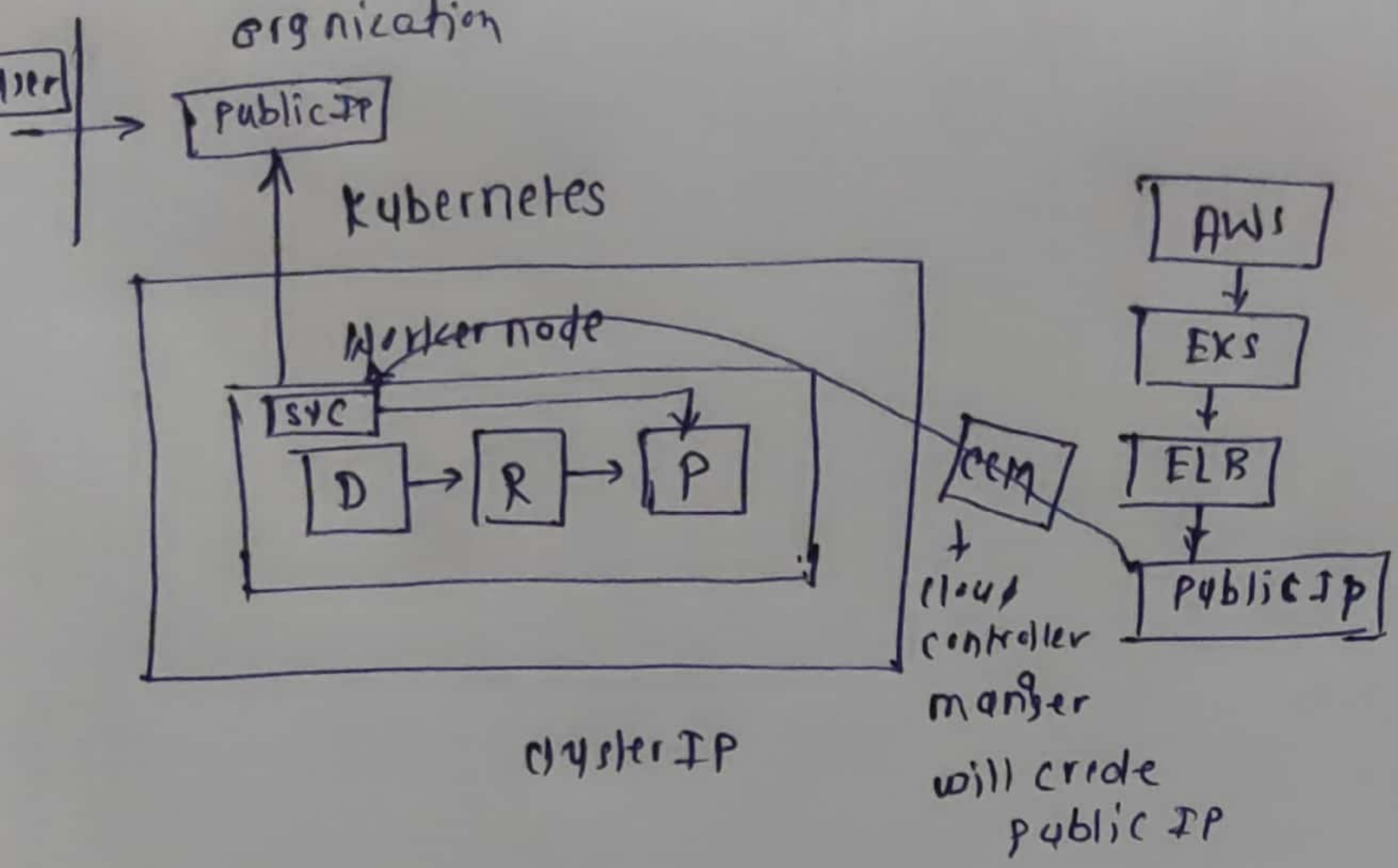
} these are default types

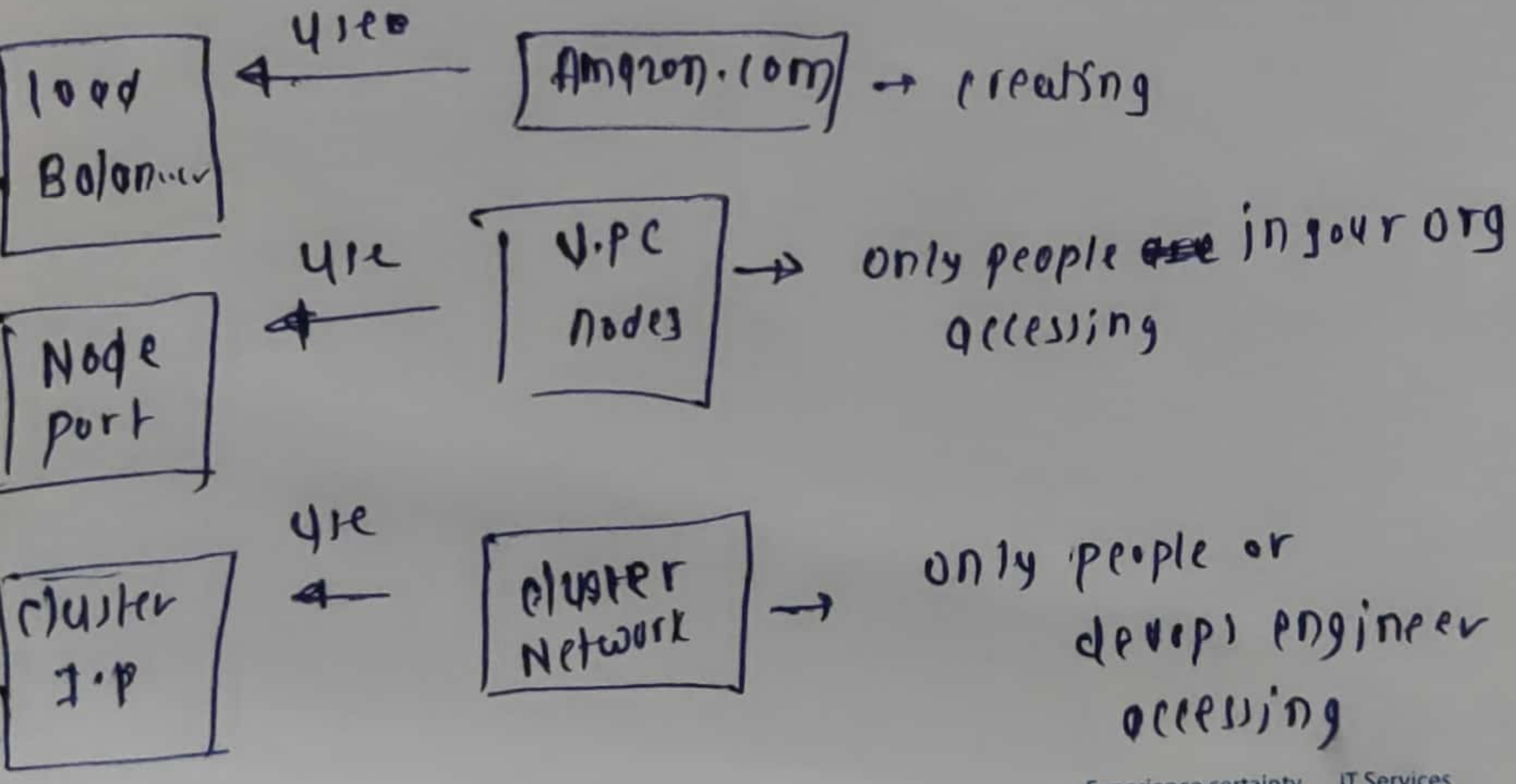
cluster IP = Inside cluster = discovery load balancing

nodeport = Inside organization = Worker node

Load Balancer = External world

EKS → ELB (u will get Elastic load Balanu IP address) → Public IP address


orgnization
user
Public IP
kubernetes
Worker node
SVC
D
R
P
CCM
cloud controller manger will crede public IP
AWS
EKS
ELB
Public IP
cluster IP


load Balancer
uses
Amazon.com
→ creating
Node port
use
V.P.C nodes
→ only people in your org accessing
cluster I.P
use
cluster Network
→ only people or devops engineer accessing