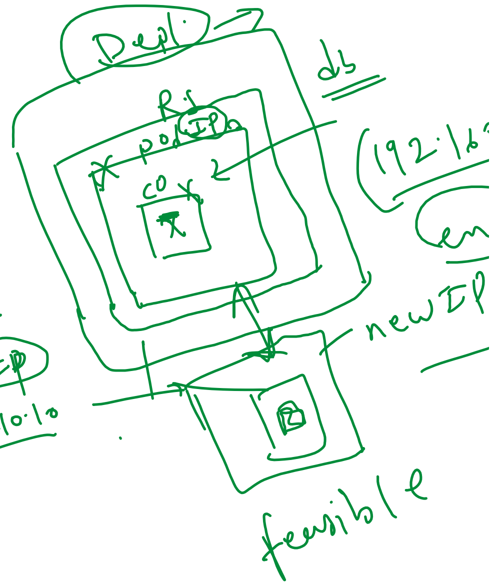
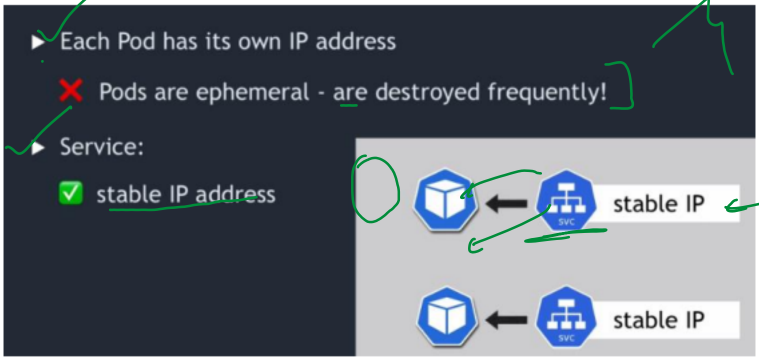
Accessing the applications - Services

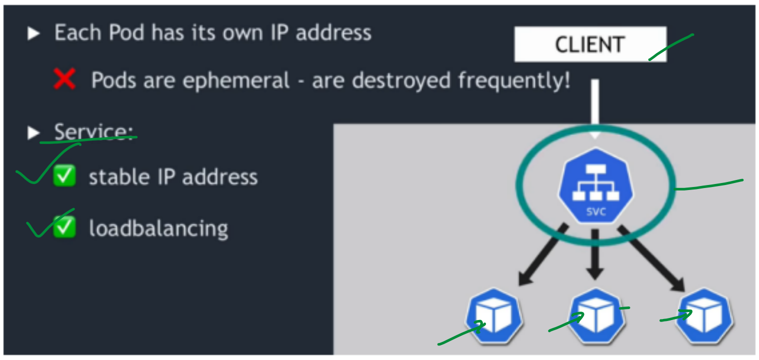
10:22 PM

pod R.C R.S Dep Vol. Live K8S K8S → Object docker shall





- Each Pod has its own IP address
  - ❌ Pods are ephemeral - are destroyed frequently!
- Service:
  - ✅ stable IP address



- Each Pod has its own IP address
  - ❌ Pods are ephemeral - are destroyed frequently!
- Service:
  - ✅ stable IP address
  - ✅ loadbalancing

Services

- Service – an abstraction which defines a logical set of Pods and a policy by which to access them
- The service maps an incoming port to a target port
- The pods targeted are defined by the selector → Endpoints
- We can have services without selector → no Endpoints object is created automatically
- iptables proxies depends on working readiness probes
- Service discovery:
  - Environment variables – are created when the pod is created → requires ordering (the service should be defined first)
  - DNS – optional cluster add-on. No ordering is required.

-p 8080 : 80

Service types

K8S →

- ClusterIP: Exposes the service on a cluster-internal IP – only reachable from within the cluster. Default
- NodePort: Exposes the service on each Node's IP at a static port. The service will be reachable from outside the cluster using NodeIP:NodePort
- LoadBalancer: Exposes the service externally using a cloud provider's load balancer.
- ExternalName: Maps the service to the contents of the externalName field, by returning a CNAME record with its value.

192.168.10.50 : 9090
x.x.10.55 9091
.56 9080
Se

Working with Services

- Expose the ports of a deployment/RC
  - kubectl expose deployment nginx --port=80 --type=NodePort
- Create services from file:

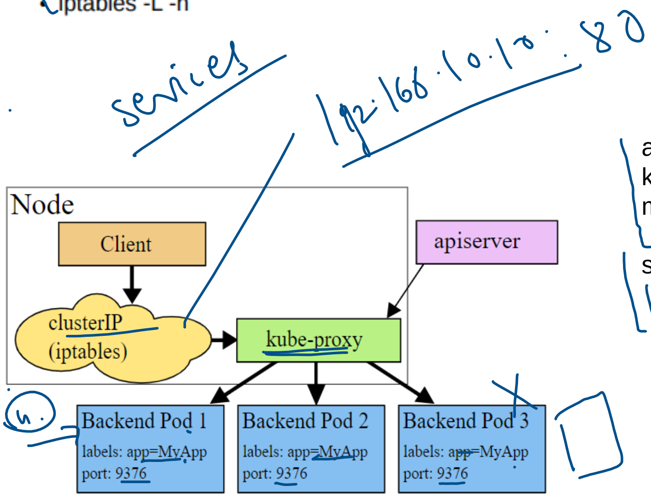
```
kind: Service
apiVersion: v1
metadata:
 name: my-service
spec:
 selector:
  app: MyApp
 ports:
  - protocol: TCP
   port: 80
   targetPort: 80

kubectl create -f svc1.yaml
```

Working with Services

kubect get apiResource

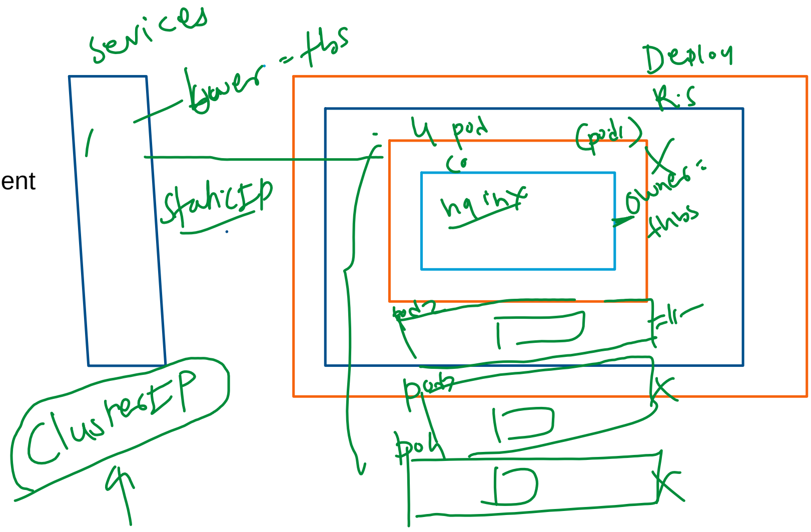
- Get service information:
  - kubectl get svc
  - kubectl describe svc
- Check service discovery
  - kubectl exec -ti busybox env
  - kubectl exec -ti busybox nslookup nginx
- Check the iptables rules on the nodes
  - iptables -t nat -L -n
  - iptables -L -n



```
apiVersion: v1
kind: Service
metadata:
 name: my-service
spec:
 selector:
  app: MyApp
 ports:
 - protocol: TCP
  port: 80
  targetPort: 9376
```

Replica

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: my-deployment
spec:
 replicas: 2
 strategy:
  type: Recreate
 selector:
  matchLabels:
   app: my-app
 template:
   metadata:
    labels:
     app: my-app
     env: prod
   spec:
    containers:
    - name: my-deployment-container
```



```
image: nginx
```