

ThoughtWorks®

SERVICES

Rise of the Containers Workshop

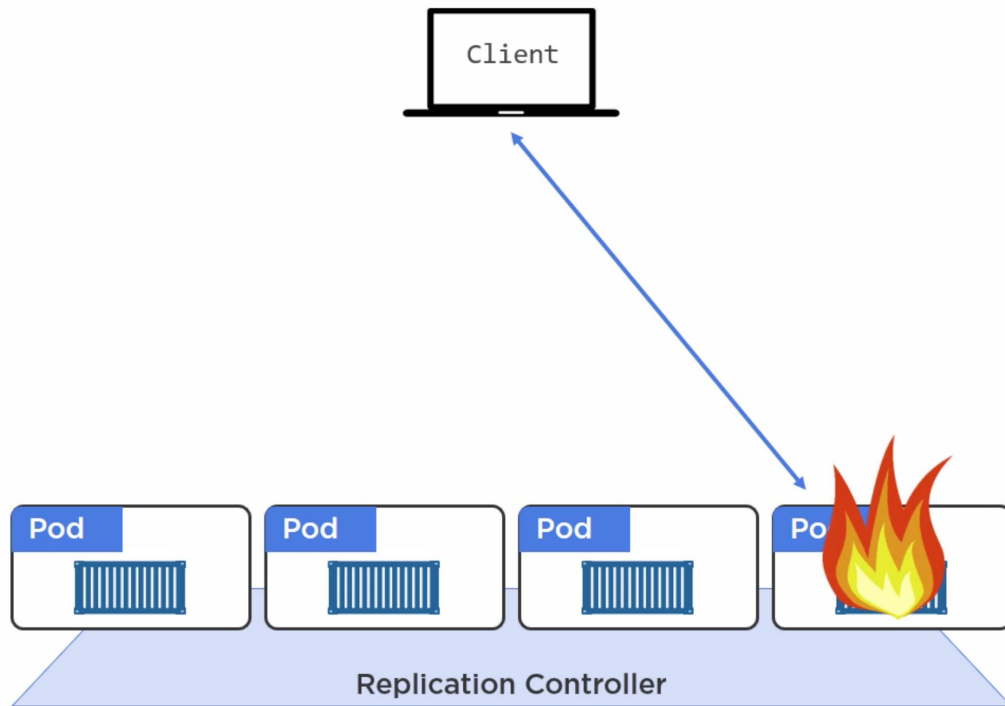


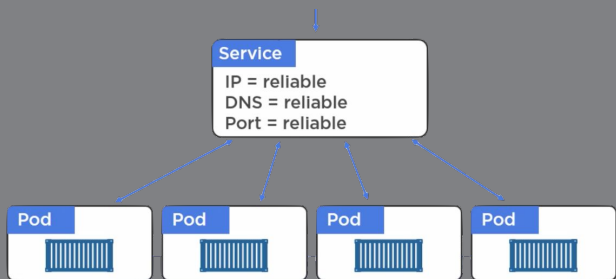


How do we access our app ?

- From outside the cluster
- From inside the cluster

Through Pod IP ?





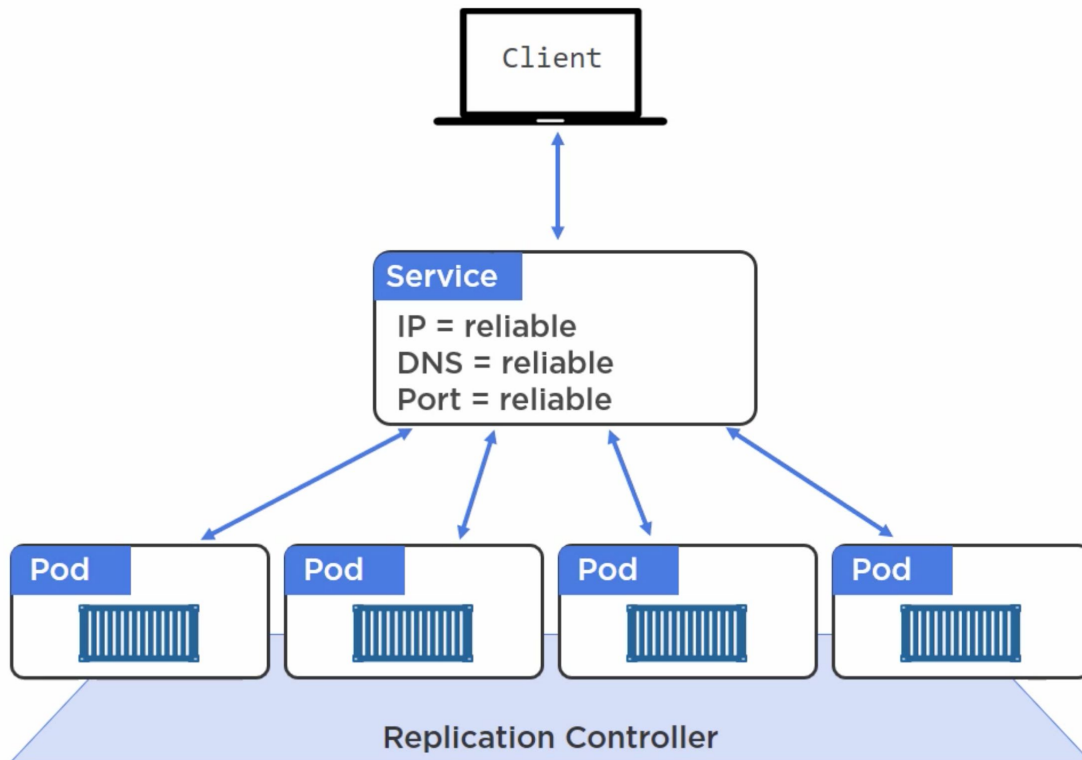
Service

A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service.

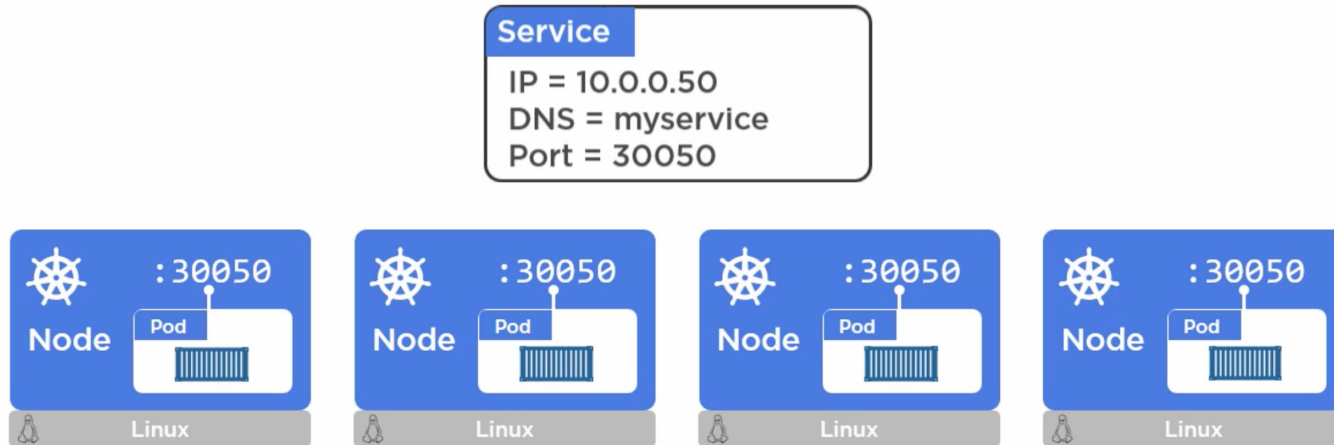
Services

- K8s objects similar to Pods and ReplicaSets
- Abstraction of a logical set of Pods
- Uses labels and selectors to match set of Pods
- Acts as intermediary for Pods to talk to each other

Through Services....

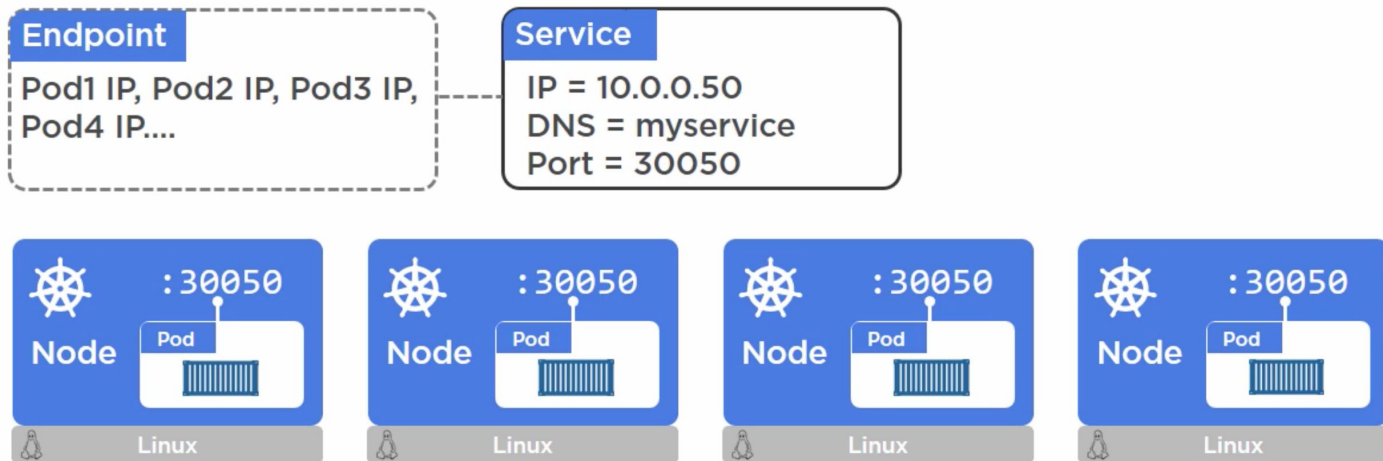


- Every Service gets a virtual IP, port and a DNS which never change
- App is accessible over the same Port on every Node
- Service load balances the request over different pods

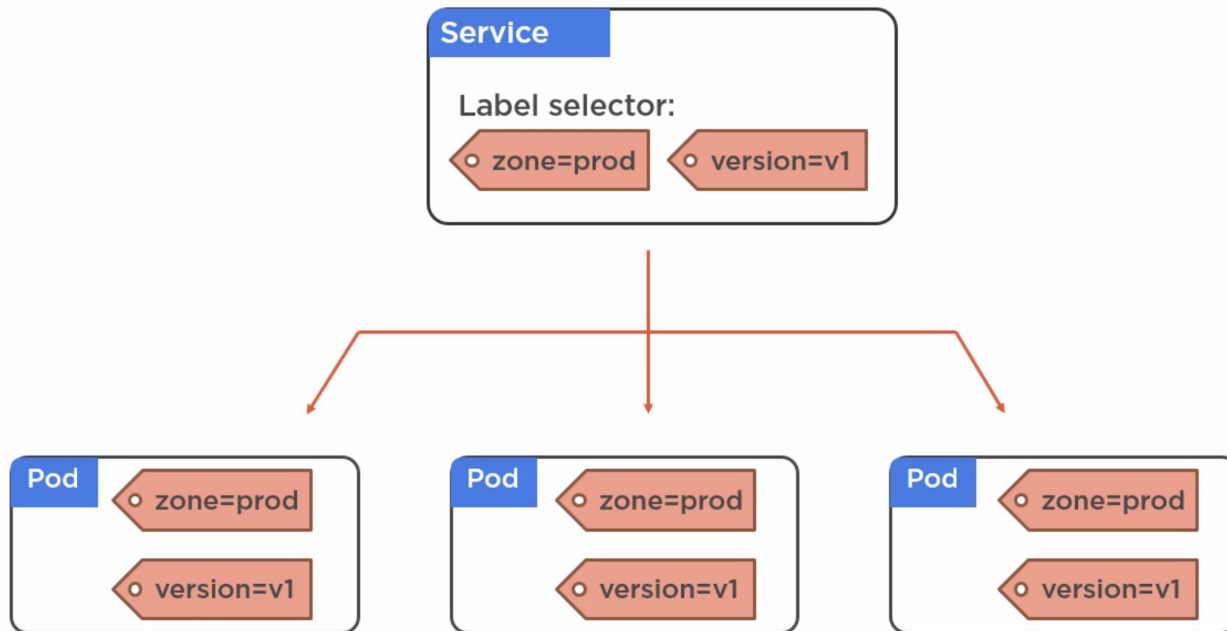


Endpoints

- Each Service is associated with an Endpoint Object
- Contains the list of Pod IP addresses that the service is associated to.
- Keeps getting updated as the pods come and go.



Services and Labels

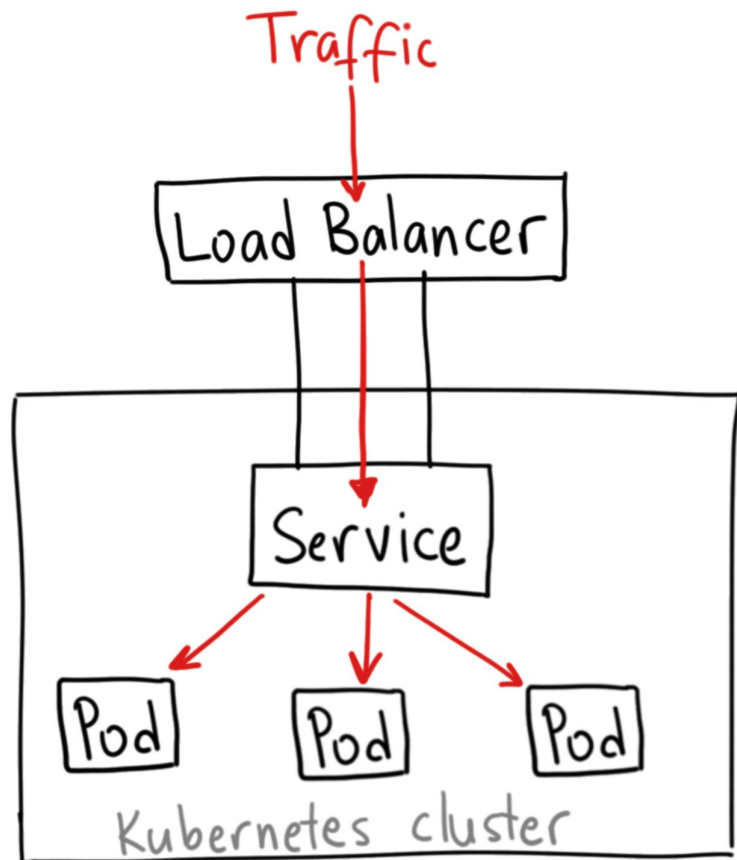


Service Discovery

- Through DNS
- Through Environment variables

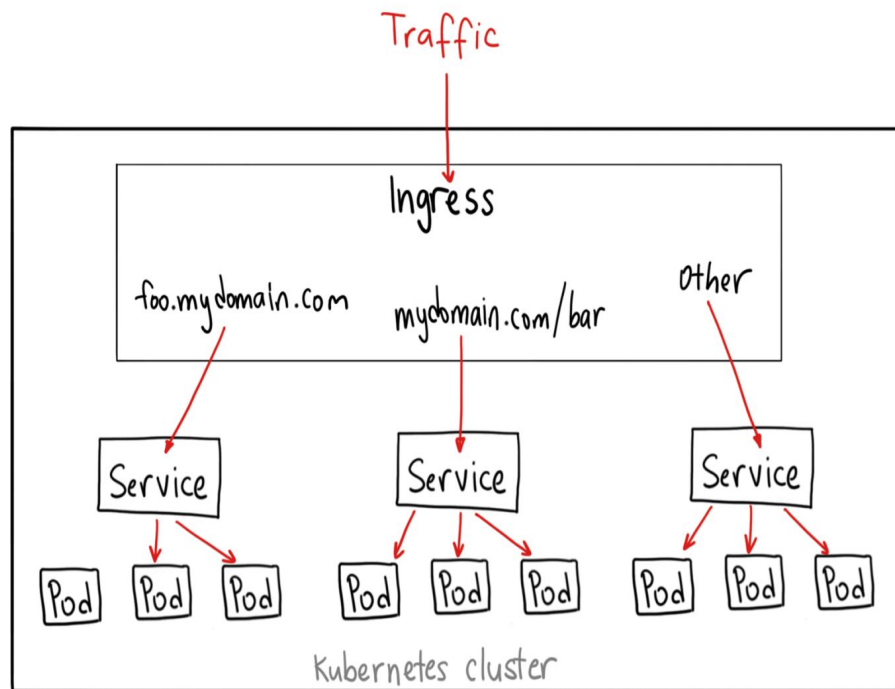
Service Demo

Load Balancer



```
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
  selector:
    app: backend
```

Ingress



```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: hello-ingress
spec:
  rules:
    - host: hello.example.com
      http:
        paths:
          - path: /a
            backend:
              serviceName: backend-a
              servicePort: 80
          - path: /b
            backend:
              serviceName: backend-b
              servicePort: 8080
```

HANDS-ON 1

1. Create a ClusterIp service over already created pods.
2. SSH into minikube and access your app via service IP and clusterPort.
3. Create a NodePort service over the same pods.
4. Access the service via browser.

HANDS-ON 2

1. Create a Pod configuration(YAML) for MongoDB and create a Pod using kubectl create command.
2. Update MetadataService to connect to real MongoDB
3. Now redeploy MetadataService new version to see MongoDB connection working and Pod to Pod to communication channel.

THANK YOU

For questions or suggestions:

Girish Verma

girishv@ThoughtWorks.com

ThoughtWorks®