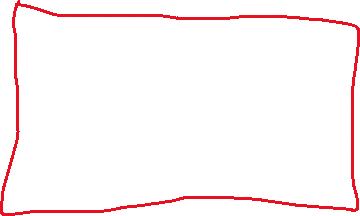
Ingress in Kubernetes clusters :





Application pod < ---- Service < ---- ingress < ---- cloud load balancer < --- http request < - --flow for cloud

Application pod < ---Service < --- ingress < --proxy server < --- http request < - --flow for bare metal

* > before ingress control , Service used to map the external request to the application pod . But when we use ingress it will responsible to map the external request to the service and service wil map to the application pod .

Example :

Service.yaml (internal service)

apiVersion: v1

kind: Service

metadata:

name: myapp-external-service

spec:

selector:

app: myapp

type: LoadBalancer

ports:

-protocol: TCP

port: 80

taregtPort: 8080

nodePort : 35010

ingress.yaml

apiVersion: networking.kis.io/v1beta1

kind : Ingress

metadata:

name: myapp-ingress

spec:

rules:

-host: myapp.com

http:

paths:

backend:

serviceName : myapp-internal-service

servicePort : 8080

host will be hosted as domain name - > <http://host/paths>

how service and ingress will integrate ?

Ans:

* > in ingress we have- > backend - > serviceName (myapp-internal-service) which will integrate or map the service.yaml file by the name .
* > And : Port given in ingress - > port : 80 - > this port will listen the external request and forward to target port : 8080 where pod is running . Can take reference in EKS\_Learning git hub Shambhu

Start work with ingress steps ?

Ans : only ingress.yaml file and service.yaml file does not happen working with ingress . we need to follow some specific steps to work with ingress

Step 1st : ingress controller - > ingress controller is to evaluate all the rules we mentioned in ingress.yaml file , to maange the redirection - > entrypoint to cluster

We have many third party implementation of the ingresss controller - > from Kubernetes (Nginx ingress controller )

What to be taken into consideration before installing ingress controller ?

Ans: Environment on which our cluster is running - > cloud service provider - > cloud load balancer (provided by cloud service provider) due to this cloud load balancer we don’t need to implement loadbalancer by our self so in minimum effort .



Environment : cluster in **bare metal** environment : in this condition only we need to configure our load balancer .

* - > bare metal means in on-premises or on own private server not using cloud ec2 instance to create the server
* - > we need to provide the proxy server soft ware or hard ware as entry point to cluster in bare metal

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80