Services ----🡪 k8s object allow communication between different component inside app and outside app.

Services types ---🡪

1. Node Port
2. Cluster IP
3. Load Balancer service

Each pod has dynamic private ip address so we cannot directly access app running inside pod using pod ip so solution is using node port service.

node port service --🡪 k8s object that listens on port on node and if there is any traffic on this port forward it to port that app listen on (target port).

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

Each service has its own ip address.

How to connect service to specific pod ?

We need on spec section 🡪 selector that has labels similar to labels defined on pod on metadata section.

How to create Nodeport service ?

apiVersion: v1

kind: Service

metadata:

name: webapp-service

namespace: default

spec:

ports:

- nodePort: 30080. #this port on node

port: 8080 # this is service port

targetPort: 8080 #this port on pod

selector:

name: simple-webapp

type: NodePort

Clustetr ip service -------🡪

Create virtual ip allow servies inside cluster to communicate with each other using this ip.

A screenshot of a computer

Description automatically generated

Group related services together and provide single end point for other services to access this service.

Access services behind endpoint is randomly.

How to create cluster-ip service ?

apiVersion: v1

kind: Service

metadata:

name: cluster-ip-service

spec:

type: ClusertIP

ports:

* port: #service port

targetPort: #port on pod

selector:

#labels on pod definition

LB-service -----🡪 this service for cloud provider

Single url to access app not ip:port

A computer screen shot of a diagram

Description automatically generated

How to create lb-service?

apiVersion: v1

kind: Service

metadata:

name: service-x

spec:

type: loadBalancer

selector:

#labels on pod definition

ports:

* nodePort: #on node

port: # on service

targetPort: # port on pod

kodekloud/simple-webapp:red ------🡪 image to create deployment

replicas =4. Port on 8080