使用Pod IP 访问应用的问题：

Service：

主要用于Pod之间的通信，相对于Pod的IP它创建完成以后就是不变的资源。

Namespace级别的隔离。

最常用的service

apiVersion: v1

kind: Service

metadata:

name: my-service

spec:

selector:

app: demo-nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

name: http

Service的类型：

ClusterIP：在集群内部使用的，默认类型

NodePort：在每个宿主机上暴露一个随机端口，30000-32767，--service-node-port-range，集群外部可访问。

LoadBalancer：使用云服务商提供的IP地址。成本太高。

ExternalName：反代到指定的域名上。

没有Selector的service。不会自动创建EndPoints。

192.168.1.100 3306 没有selector的service的名字+端口进行访问到192.168.1.100 3306。

172.16.1.100

ClusterIP+Ingress 域名访问

Ingress：

它是Kubernetes集群中服务的入口，可以提供负载均衡、SSL终止和基于域名的虚拟主机。Treafik、Nginx、HAProxy、Istio。

Ingress 官方文档：<https://kubernetes.io/docs/concepts/services-networking/ingress/>

Ingress-Nginx安装文档：<https://kubernetes.github.io/ingress-nginx/deploy/>

Ingress-Nginx 文档：<https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/>

定义一个Ingress：

apiVersion: networking.k8s.io/v1beta1

kind: Ingress

metadata:

name: simple-fanout-example

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

rules:

- host: foo.bar.com

http:

paths:

- path: /foo

backend:

serviceName: service1

servicePort: 4200

- path: /bar

backend:

serviceName: service2

servicePort: 8080

host: 可选参数，一般都会配置我们自己的域名。

path: 一个路径对应一个serviceName和一个Port

backend: path对应的后端是谁。

foo.bar.com/foo 🡪 service1:4200

Ingress：<https://kubernetes.io/docs/concepts/services-networking/ingress/>