把我们的 httpserver 服务以 Istio Ingress Gateway 的形式发布出来。以下是你需要考虑的几点：

* 如何实现安全保证；
* 七层路由规则；
* 考虑 open tracing 的接入。

1. 用HTTPS实现安全保证

kubectl create ns securesvc

kubectl label ns securesvc istio-injection=enabled

kubectl create -f httpserver.yaml -n securesvc

创建secret、crt及key

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -subj '/O=edwin Inc./CN=\*.edwin.io' -keyout edwin.io.key -out edwin.io.crt

kubectl create -n istio-system secret tls edwin-credential --key=edwin.io.key --cert=edwin.io.crt

部署istio的virtual service和gateway

kubectl apply -f virsrv.yaml -n securesvc

文本

描述已自动生成

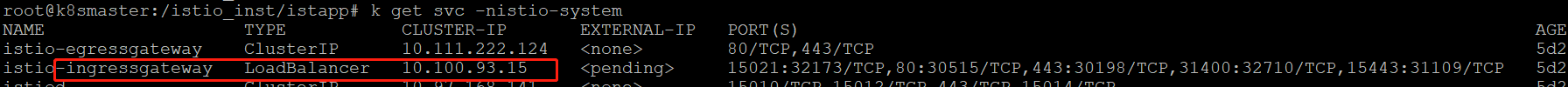
kubectl apply -f gateway.yaml -n securesvc

文本

描述已自动生成

Istio的ingress svc的地址

istio-ingressgateway LoadBalancer 10.100.93.15



然后curl 测试https

curl --resolve http-basic.edwin.io:443:$INGRESS\_IP https://http-basic.edwin.io/healthz -v -k

文本

描述已自动生成

curl https://http-basic.edwin.io:443/-v -k

电脑的屏幕截图

描述已自动生成

1. 七层路由规则的测试

基于uri 地址的路由规则

apiVersion: networking.istio.io/v1beta1

kind: VirtualService

metadata:

name: httpsserver

spec:

gateways:

- httpsserver

hosts:

- http-basic.edwin.io

http:

- match:

- port: 443

- uri:

exact: "/hc/"

rewrite:

uri: "/healthz/"

route:

- destination:

host: http-basic.securesvc.svc.cluster.local

port:

number: 80

- match:

- port: 443

- uri:

exact: "/root/"

rewrite:

uri: "/"

route:

- destination:

host: http-basic.securesvc.svc.cluster.local

port:

number: 80

代码片段

文本

描述已自动生成

配置curl参数：

电脑的屏幕截图

描述已自动生成

访问不通过的地址

curl <https://http-basic.edwin.io:443/hc> -v -k

curl https://http-basic.edwin.io:443/root/ -v -k

图形用户界面, 网站

描述已自动生成

1. Istio tracking

docker build -t istio\_track0 -f Dockerfile .

docker build -t istio\_track0 -f Dockerfile .

docker tag 1b2e23d2a082 edwindocker/istio\_track0

docker push edwindocker/istio\_track0

按照上述方法制作其他的两个service1、service2的镜像

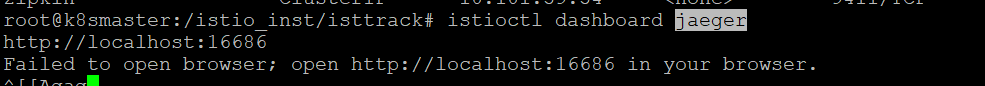
图形用户界面

描述已自动生成

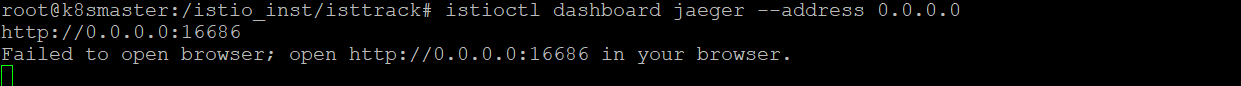
文本

描述已自动生成

启动jaeger



启动jaeger



访问：<http://192.168.30.188:16686/>

图形用户界面, 应用程序

描述已自动生成