

应用服务网格

最佳实践

文档版本 01
发布日期 2025-03-18



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目 录

1 数据面 sidecar 升级不中断业务.....	1
2 面向 Dubbo 协议的服务治理.....	4
2.1 简介.....	4
2.2 服务发现模型.....	4
2.3 SDK 适配方式.....	5
2.3.1 PASSTHROUGH 方案.....	5
2.3.2 静态目标服务.....	6
3 网关访问保留源 IP.....	7
4 如何搭建 IPv4/IPv6 双栈网络.....	14
5 AOM 页面查询应用服务网格详细指标.....	17
6 ASM 用户委托权限收缩指南.....	18
7 Istio-ingressgateway 高可用性配置指导.....	19
7.1 网关工作负载.....	19
7.1.1 滚动升级策略.....	19
7.1.2 Pod 优雅删除.....	20
7.1.2.1 Istio-proxy 容器终止排出时间.....	20
7.1.2.2 Pod 缩容时间窗.....	21
7.1.3 调度策略.....	21
7.1.3.1 污点和容忍策略.....	22
7.1.3.2 负载亲和调度策略.....	23
7.1.3.3 节点亲和调度策略.....	26
7.1.4 容器资源申请和限制.....	28
7.2 网关 Service.....	29
7.2.1 服务亲和.....	29
7.2.2 负载均衡器健康检查.....	30

1 数据面 sidecar 升级不中断业务

应用服务网格作为集群网络管理的重要工具，为网格内的服务提供流量治理与流量监控的能力。sidecar作为应用服务网格数据面的重要组件，需要依赖服务业务pod的更新来实现sidecar的升级和重新注入。

本章节主要介绍如何实现数据面sidecar升级过程中，不中断服务的业务流量。

配置服务实例数

为保证您的服务在sidecar升级的过程中不中断业务流量，首先确保您的服务实例数大于等于2，升级策略为滚动升级（rollingUpdate）。

相关滚动升级策略如下，供参考：

kubectl get deploy nginx -n *namespace_name* -oyaml | grep strategy -a10

```
spec:
  minReadySeconds: 2
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
      version: v1
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
```

配置项说明：

- 服务实例数：deployment.spec.replicas >= 2
- 升级策略：deployment.spec.strategy.type == RollingUpdate
- 滚动升级最小存活实例数：deployment.spec.replicas - deployment.spec.strategy.maxUnavailable > 0

添加 readiness 探针

添加readiness探针，可以保证您的新实例pod在真正准备就绪时，才开始接管业务流量。这就避免了在新的实例pod未启动时，接管业务流量造成的访问不通问题。

相关配置如下：

```
kubectl get deploy nginx -n namespace_name -oyaml | grep readinessProbe -a10
```

```
readinessProbe:
  failureThreshold: 3
  httpGet:
    path: /
    port: 80
    scheme: HTTP
  initialDelaySeconds: 1
  periodSeconds: 10
  successThreshold: 1
  timeoutSeconds: 1
```

配置项说明：

readiness探针配置项：deployment.spec.template.spec.containers[i].readinessProbe

其中，包括探针检查初始时间，检查间隔，超时时间等配置。

设置服务就绪时间

服务就绪时间，minReadySeconds：用于标识pod的ready时间至少保持多长时间，才会认为服务是运行中。

相关配置如下：

```
kubectl get deploy nginx -n namespace_name -oyaml | grep minReadySeconds -a1
```

```
spec:
  minReadySeconds: 2
  progressDeadlineSeconds: 600
```

配置项说明：

服务就绪时间：deployment.spec.minReadySeconds，可根据您业务的实际情况确定。

配置优雅关闭时间

terminationGracePeriodSeconds，优雅关闭时间。在滚动升级过程中，首先会移除旧的服务实例pod的endpoint，并将实例pod的状态置为Terminating，这时K8S会发送SIGTERM信号给pod实例，并等待优雅关闭时间后，将pod强制终止。您可以利用这段时间，处理未完成的请求：

```
kubectl get deploy nginx -n namespace_name -oyaml | grep terminationGracePeriodSeconds -a1
```

```
securityContext: {}
terminationGracePeriodSeconds: 30
tolerations:
```

配置项说明：

优雅关闭时间：deployment.spec.template.spec.terminationGracePeriodSeconds，默认值为30s，可根据业务诉求适当调整。

配置 preStop

preStop会在实例pod中止前调用，可以用来实现业务pod的优雅关闭。此处需要根据业务诉求来进行相应的配置，以Nginx为例。

kubectl get deploy nginx -n namespace_name -oyaml | grep lifec -a10

```
lifecycle:
  preStop:
    exec:
      command: ["/bin/sh", "-c", "nginx -s quit; sleep 10"]
```

在“lifecycle”下的“preStop”中，定义了一个命令**nginx -s quit; sleep 10**，这个命令首先会发送一个优雅关闭信号给Nginx进程，然后暂停10秒。这样，在Pod终止之前，Nginx会有足够的时间完成正在处理的请求并优雅地关闭。

需要注意的是，sleep 10中的10秒是一个示例值，您可以根据实际需要和应用程序的性能来调整这个值，关键是为Nginx提供足够的时间来优雅地关闭。

类似地，您可以选择运行自定义命令或自定义脚本，来优雅关闭您的服务进程。

2 面向 Dubbo 协议的服务治理

2.1 简介

Dubbo作为一种特有协议，需要有如下支持：

- 网格服务数据面Envoy支持对Dubbo协议的解析和流量管理。
- 网格控制面支持对Dubbo治理规则的配置。支持灰度发布、负载均衡、访问授权等服务管理。

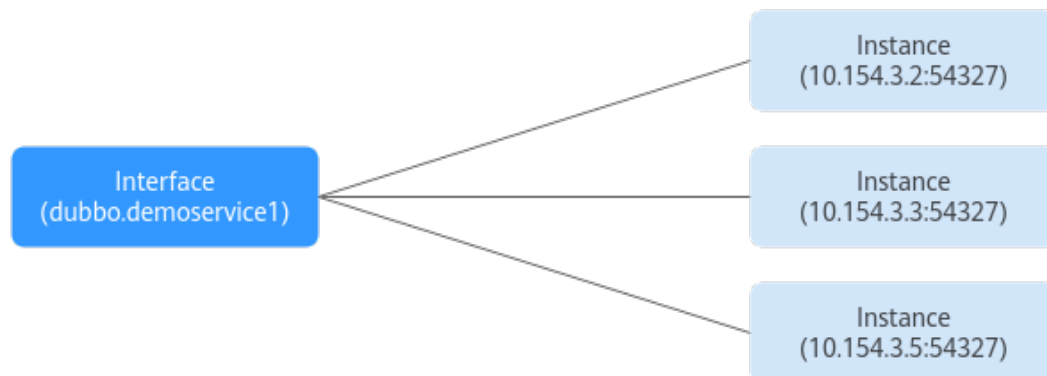
另外，Dubbo的服务发现模型和Kubernetes、Spring Cloud等服务发现模型不一致，需要额外的处理。

2.2 服务发现模型

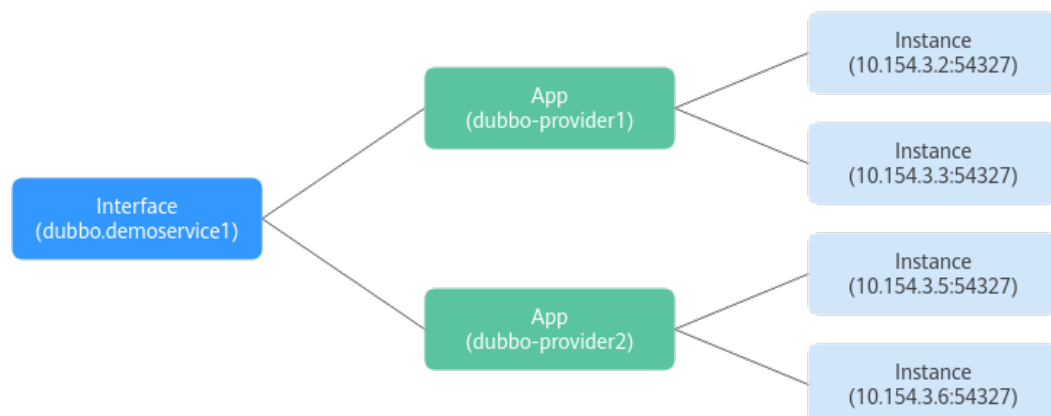
Dubbo现有模型存在的问题（来自Dubbo社区2.7.4总结）：

- 在微服务架构中，注册中心管理的对象是应用（服务），而非对外的服务接口，不过目前Dubbo的注册中心管理的对象是Dubbo服务接口，与Spring Cloud或Cloud Native注册方式背道而驰。
- 一个Dubbo应用（服务）允许注册N个Dubbo服务接口，当N越大时，注册中心的负载越重。

Dubbo现有服务模型：根据Dubbo接口查找服务实例。



Dubbo Cloud Native服务发现模型，将原来Interface一级的服务发现拆分成两级，基于App找实例地址。

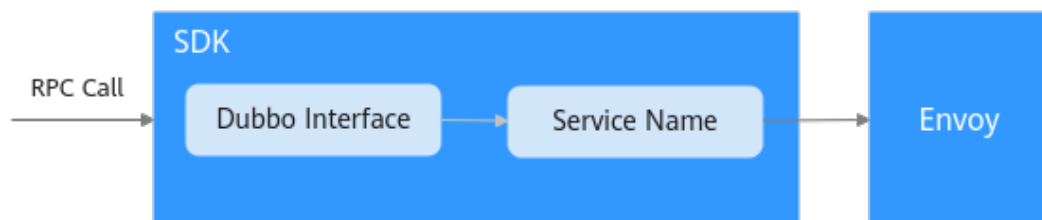


2.3 SDK 适配方式

2.3.1 PASSTHROUGH 方案

方案介绍

SDK中客户端使用Interface调用目标服务时，修改原有服务发现逻辑。将原有通过Interface查找服务实例，修改为通过接口查找服务名，直接对服务名发起访问。



详细说明

对于Dubbo协议的不同版本会有不同：

- 2.7.4+版本：2.7.4以上的Cloud Native版本中重构过与Kubernetes一致的服务发现模型，直接可以从Interface关联到服务信息。
- 2.7.3及之前版本：Dubbo社区版本未提供Interface到Service的二级关系，需要SDK根据自己的实际使用方式来维护Interface到服务的映射关系。如可以使用在服务注册时通过扩展信息的方式提供服务名等信息。

客户实际使用中可以根据自己的SDK使用情况选择灵活的处理方式。对于老版本的SDK可以基于现有的服务注册和服务发现流程，做如下处理：

1. 在注册信息中扩展Service的定义。在服务部署时会通过环境变量将服务的元数据信息注入到SDK中，包括appname、namespace，分别表示部署的服务名、部署的命名空间。
2. 在服务启动时向注册中心注册Dubbo接口和Kubernetes服务名和命名空间的关系。

3. 在客户端发起访问时，根据原有服务发现的流程根据Interface查询到服务的元数据，并用对应的服务信息组装RPC请求。建议在Dubbo请求中使用Attachment扩展字段存储appname、namespace信息。

2.3.2 静态目标服务

方案介绍

通过 **dubbo:reference** 在Dubbo服务的服务消费者中对引用的服务提供者进行配置。使用选项url定义点对点直连服务提供者地址，绕过注册中心，直接调用目标服务。

详细说明

如果原Dubbo服务中使用的是xml配置文件，则只需要修改配置文件即可。

```
<?xml version="1.0" encoding="UTF-8"?>
<beans>
  <!-- 提供哪些接口给消费者调用 -->
  <dubbo:reference id="helloService" interface="com.dubbo.service.HelloService" url = "dubbo://
helloService:20880" />
</beans>
```

如果服务中使用注解的方式来定义引用的目标服务，则需要修改代码中对目标服务的注解。

```
@Reference(url = "dubbo://helloService:20880")
HelloService helloService;
```

3 网关访问保留源 IP

⚠ 注意

设置成节点亲和在某些场景下可能导致无法访问ELB，具体情况请查看 [集群内部无法使用ELB地址访问负载](#)。

操作场景

服务通过网关访问时，默认情况下，目标容器中看到的不是客户端的源IP，如果需要保留源IP，请参考本节指导操作。

配置方法

请在CCE控制台“服务发现”页面，istio-system命名空间下，更新服务所关联的网关服务，将服务亲和改成“节点级别”。前提是已开启ELB的获取客户端IP功能（当前为默认开启）。



externalTrafficPolicy：表示此Service是否希望将外部流量路由到节点本地或集群范围的端点。有两个可用选项：Cluster（默认）和Local。Cluster隐藏了客户端IP，可能导致第二跳到另一个节点，但具有良好的整体负载分布。Local保留客户端源IP并避免LoadBalancer和NodePort类型服务的第二跳，但存在潜在的不均衡流量传播风险。

验证方式

结合httpbin镜像在“x-forward-for”字段中可以看到源IP，httpbin是一个HTTP Request & Response Service，可以向他发送请求，他将会按照指定的规则将请求返回。httpbin镜像可在SWR中搜索。使用httpbin镜像进行验证时请确保集群已开通网络。

1. 登录ASM应用服务网格控制台，选择一个可用的测试网格并单击进入。
2. 选择左侧“网格配置”查看其关联的集群。



3. 单击集群名称进入集群详情页，单击“工作负载-无状态工作负载”，单击右上角“创建工作负载”按钮。
4. 配置工作负载的信息。

基本信息

- 负载类型：选择无状态工作负载Deployment。
- 负载名称：命名工作负载为httpbin。
- 命名空间：选择工作负载的命名空间，默认为default。
- 其余参数使用默认值。



容器配置

- 基本信息：
 - 容器名称：为容器自定义一个名称
 - 镜像名称：单击后方“选择镜像”，在右上角搜索框搜索“httpbin”镜像并选择，单击“确认”。
 - 镜像版本：选择一个镜像版本
 - 其余参数使用默认值。



服务配置

- 服务（Service）是用来解决Pod访问问题的。每个Service有一个固定IP地址，Service将访问流量转发给Pod，而且Service可以给这些Pod做负载均衡。
- 单击服务配置参数下面的“+”进入创建服务页面。
- Service名称：Service名称为工作负载名称。

- 访问类型：选择集群内访问。
- 端口设置：
 - 协议：TCP
 - 容器端口：80（以实际访问端口为准）
 - 服务端口：80（以实际访问端口为准）

创建服务

Service名称

httpbin

访问类型

集群内访问

通过集群的内部IP暴露服务，只能在集群内部访问

节点访问

通过每个节点上的IP和静态端口（NodePort）暴露服务

负载均衡

通过ELB负载均衡对外部提供服务，高可用，超高性能，稳定安全

DNAT网关

通过NAT网关暴露集群节点访问类型服务，支持多个节点共享使用弹性IP

集群外访问推荐选择负载均衡访问类型

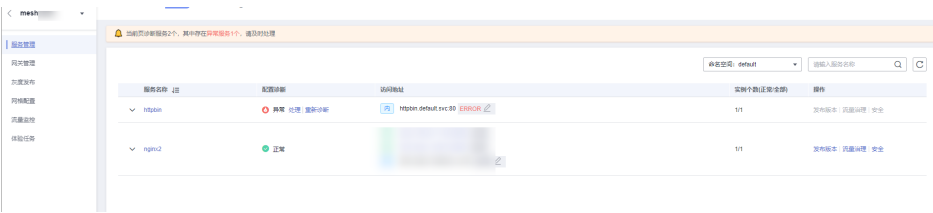
端口配置

协议	容器端口	服务端口	操作
TCP	80	80	删除

- 5. 单击右下角“确定”完成服务创建。
- 6. 单击右下角“创建工作负载”完成工作负载创建。
- 7. 在集群详情页选择左侧“服务发现”页签，可在服务列表中查看到所创建的httpbin服务。



- 8. 返回ASM应用服务网格，选择左侧“服务管理”页签，在服务管理中可查看到httpbin的配置诊断显示为异常。



- 9. 单击此服务配置诊断中的“处理”按钮，按照弹出“配置诊断”页面对应的修复指导进行修复。

配置诊断

手动处理项

2 可自动处理项

检查项	检查结果	操作
Service 的端口名称是否符合istio规范	失败 访问端口 80 协议 —	修复指导
Service的选择器中是否配置了version标签	失败	修复指导
服务是否配置了默认版本的服务路由，路由配置是否正确	失败	修复指导

重新诊断

一键修复

以“Service的端口名称是否符合Istio规范”为例，选择协议为“http”，单击“一键修复”。

配置诊断

手动处理项

2 可自动处理项

检查项	检查结果	操作
Service 的端口名称是否符合istio规范	成功 访问端口 80 协议 http	修复指导
Service的选择器中是否配置了version标签	成功	修复指导
服务是否配置了默认版本的服务路由，路由配置是否正确	成功	修复指导

重新诊断

一键修复

10. 选择左侧“网关管理”页签，单击右上角“添加网关”，在弹出“添加网关”页面输入配置信息。

- 配置信息
- 网关名称：httpbin。
 - 集群选择：网格所关联使用的集群。
 - 负载均衡：选择公网、选择一个elb公网。
 - 对外协议：选择HTTP。
 - 对外端口：自定义端口。
 - 外部访问地址：负载均衡中所选elb公网地址。

添加网关

基本信息

* 网关名称

httpbin

* 集群选择

cluster1

访问方式

* 负载均衡 ?

公网

elb-1198()

创建负载均衡

仅支持集群所在 VPC vpc-e5b9 下的负载均衡实例，查询结果已自动过滤

访问入口

* 对外协议

HTTP

GRPC

TCP

TLS

HTTPS

* 对外端口

80

* 外部访问地址 ?

- 单击“路由配置”下方的“+”在弹出“添加路由”页面添加路由。
- URL：选择完全匹配并输入映射。

▪ 命名空间：服务所在命名空间。

▪ 目标服务：默认设置。

添加路由

★ URL

完全匹配

/get

★ 命名空间

default

★ 目标服务

httpbin

80

当前服务已根据网关协议自动过滤。[了解更多](#)

重写

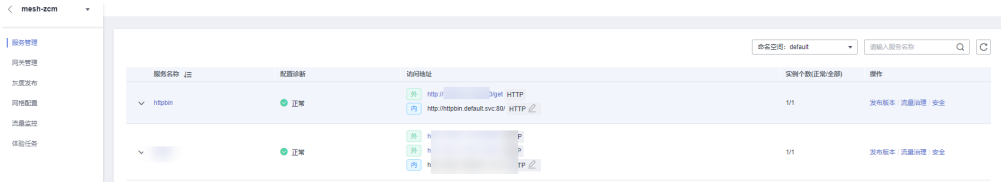
重写HTTP URI和Host/Authority头，于转发前执行

确定

取消

配置完成后单击“确定”。

11. 单击“确定”完成网关添加。
12. 选择左侧“服务管理”页签，可以在“访问地址”查看到所创建路由的外部访问地址。



13. 在映射的外部访问地址后加上“?show_env=1”，访问添加字段后的外部访问地址，例如：http://xxx.xxx.xxx:80/get?show_env=1。可以在“x-forward-for”字段中查看网关获取的IP为容器段IP。
- ...
- ...
- ...
- x-forward-for: xxxx

14. 返回集群详情页，选择左侧导航栏“服务发现”，更改服务所关联的网关服务的配置。方法如下：
- 下拉上方“命名空间”列表选择“istio-system”。



展开服务后方“更多”选项，单击“更新”，在弹出“更新服务”页面将“服务亲和”更改为“节点级别”，勾选“我已阅读《负载均衡使用须知》”，单击“确定”。

更新服务

Service名称httpbin-svc

访问类型

集群内访问ClusterIP

节点访问NodePort

负载均衡LoadBalancer

服务亲和

集群级别

节点级别

命名空间

istio-system

选择器

键=值

确认添加引用负载标签

app = istio-ingressgateway

istio = ingressgateway

服务通过选择器与负载（标签）关联

负载均衡器

elb-1198

负载均衡配置：分配策略：加权轮询算法；会话保持类型：不启用；健康检查：不启用；

我已阅读《负载均衡使用须知》

端口配置

协议	容器端口	服务端口	操作
TCP	1025	80	删除

+

如果当前服务被关联到路由，则更新服务的端口信息后需要手动更新路由转发策略配置信息

注解

键=值

确认添加使用指南

asm.huaweicloud.com/post = ["kin...]

asm.huaweicloud.com/updateTime...

service.protai.kubernetes.io/type =...

15. 返回13中访问的外部地址并刷新，若设置之后“x-forward-for”字段中显示的网关获取IP的结果为本机源IP，则完成验证。

```
...
...
...
x-forward-for: xxxx
```

4 如何搭建 IPv4/IPv6 双栈网格

当前CCE支持创建IPv4/IPv6双栈集群，在双栈集群基础上支持开启IPv4/IPv6网格。启用双栈网格后，服务拥有 IPv4地址和IPv6地址，通过这两个地址都可以进行服务间的访问。本教程将指引您搭建一个IPv4/IPv6双栈的网格，使网格内的服务可以通过IPv6地址互访。添加双栈网关后，可以为IPv6终端的客户提供对外访问。

使用场景

- 如果您的服务需要使用IPv6地址进行服务间的互访和流量治理，您需要使用IPv6双栈。
- 如果您要为使用IPv6终端的客户提供对外访问，您需要使用IPv6双栈网格来创建网关。

约束与限制

- 支持启用双栈网格的约束。

网格类型	Istio版本	支持启用的集群类型	集群网络类型	其他说明
基础版	1.18及以上	CCE Turbo 集群	云原生网络2.0	集群需要已启用IPv6，请参考 通过CCE搭建IPv4/IPv6双栈集群

- 支持创建IPv4/IPv6网关的约束。

网格类型	Istio版本	支持启用的ELB类型	ELB规格	其他说明
基础版	1.18及以上	独享型ELB	4层网络型	ELB需要配置IPv6地址

- 仅istio1.18及以上版本的基础版网格支持启用IPv6功能。

- 您的网格开启IPv4/IPv6双栈后不支持关闭，未开启双栈的网格也不支持创建完成之后开启双栈。
- 您低版本的网格升级到1.18及以上时，不支持升级后开启IPv4/IPv6双栈。

创建 IPv6 网格

步骤1 登录ASM控制台，购买一个ASM网格，网格参数填写如下：

- 网格类型：选择“基础版”。
- 网格名称：输入网格名称。
- Istio版本：选择"1.18"或以上版本。
- 启用IPv6：打开启用开关，开启后将过滤满足条件的CCE集群。

网格名称

请输入网格名称

同一账户下网格不可重名。创建后不可修改

Istio版本

1.8

1.13

1.15

1.18

启用 IPv6



[如何搭建 IPv4/IPv6 双栈网格](#)

其他配置参数根据实际情况填写。

步骤2 创建完成后，单击网格名称进入，在“网格配置-基本信息”中可以看到“IPv6双栈 已开启”字样。

网格状态	✔ 运行中
计费模式	免费
IPv6 双栈	已开启

----结束

添加 IPv4/IPv6 双栈网关

步骤1 登录ASM控制台，在网格列表页面单击已经开启IPv6双栈的网格名称，单击“网关管理-添加网关”。网关参数如下填写

- 访问方式：选择“DualStack”。
- 负载均衡：选择独享型，选中的独享型ELB需要有IPv6地址。

其他配置参数根据实际情况填写。

说明

当前IPv4/IPv6双栈网关添加IPv6外部访问地址时，只支持添加域名，通过域名访问，不支持添加IPv6地址访问。

----结束

验证方式

1. 客户端通过配置域名解析，解析域名到网关的IPv6地址，可以通过域名访问实现IPv6客户端访问。

```
[root@k8s ~]# curl -v http://test.com:8678
* Trying 2407:c080:11f0:5fa:4ede:f73d:a9e9:c3c2:8678...
* Connected to test.com (2407:c080:11f0:5fa:4ede:f73d:a9e9:c3c2) port 8678 (#0)
```

2. 查看ingressgateway日志有对应IPv6请求日志信息。

```
[{"user_agent":"curl/7.79.1","upstream_transport_failure_reason":null,"cluster_id":"972f35ec-8388-11ee-9bad-0255ac108086","mesh_id":"59bac675-94d9-11ee-8776-0255ac10809a","downstream_local_address":"[2407:c080:11f0:601:7d2f:d21b:504d:a96c]:1027","upstream_local_address":"[2407:c080:11f0:601:7d2f:d21b:504d:a96c]:51094","requested_server_name":null,"upstream_service_time":3,"request_id":"805241c0-f1b5-49f3-8ebc-d040d1b7ae5c","connection_termination_details":null,"upstream_host":"[2407:c080:11f0:601:4260:f09f:ebef:8c5d]:8080","path":"/","pod_namespace":"istio-system","response_flags":"","duration":3,"forwarded_for":"[2407:c080:11f0:5fa:4b12:f0c9:6655:deaf],"authority":"test.com:8678","bytes_received":0,"route_name":"tomcat-http-v6.default.svc.cluster.local:8082","response_code_details":{"via_upstream","response_code":200,"protocol":"HTTP/1.1"},"upstream_cluster":"outbound|8082|v1|tomcat-http-v6.default.svc.cluster.local","downstream_remote_address":"[2407:c080:11f0:5fa:4b12:f0c9:6655:deaf]:53380","metadata":{"@type":"GET","start_time":"2023-12-09T08:24:23.704Z","bytes_sent":11230,"pod_name":"istio-ingressgateway-1-16-5-r1-5dbfd5d48b-cpr12"}
```

5 AOM 页面查询应用服务网格详细指标

服务网格开启应用指标并接入华为云AOM后，可以在AOM页面查询网格的详细指标，具体步骤如下：

步骤1 登录AOM页面，进入“指标浏览”页面。

步骤2 选择“指标源”，选择想要查询的网格上报指标的Prometheus实例。



步骤3 在页面上查询上报至AOM的指标。

- 全量指标查询

通过输入具体指标名称，即可查阅istio网格相关指标。

同时支持通过增加条件过滤相关指标信息。

- 按普罗语句查询

通过输入promql语句查询网格指标。

----结束

6 ASM 用户委托权限收缩指南

背景介绍

应用服务网格权限管理是通过IAM统一身份认证里的委托实现的，而2024年7月前已进行授权的用户可能存在委托权限过大的问题，鉴于安全考虑，建议通过下文指导进行委托的权限收缩。

操作步骤

- 步骤1** 登录[IAM控制台](#)。
- 步骤2** 在左侧菜单栏单击委托，在搜索框中，搜索asm_admin_trust,单击搜索出来的委托名称。
- 步骤3** 单击进入“授权记录”页签，删除所有权限。
- 步骤4** 单击“授权”，搜索“CCE Administrator”策略，选择该策略，单击“下一步”。选择授权范围方案为“指定区域项目资源”，选择将要使用ASM服务的region，单击“确定”。
- 步骤5** 单击“授权”，搜索“Tenant Guest”策略，选择该策略，单击下一步”。选择授权范围方案为“指定区域项目资源”，选择将要使用ASM服务的region，单击“确定”。

----结束

说明

步骤3请删除所有权限再重新授权，否则可能出现异常。

7 Istio-ingressgateway 高可用性配置指导

7.1 网关工作负载

7.1.1 滚动升级策略

- 步骤1** 进入CCE Console页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。
- 步骤2** 通过YAML配置如下参数：

```
spec:
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 10%
```

也可以通过单击步骤1页面待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：

高级配置

升级策略

策略策略

标签与注解

DNS配置

升级方式

滚动升级

分批升级

逐步用新版本实例替换旧版本实例。升级过程中，业务会同时均衡分布到新老实例上，因此业务不会中断。

最大无效实例数

0

个

最大 surge

10

%

每次滚动升级允许的最大无效实例数。如果等于实例数有断链风险（最小存活实例数 = 实例数 - 最大无效实例数）

每次滚动升级允许超出所需规模的最大实例数

参数说明：

参数	说明
最大无效实例数（maxUnavailable）	与spec.replicas相比，可以有多少个Pod失效，也就是删除的个数或比例，建议值是0个。 比如spec.replicas为3，那升级过程中就至少有3个可用的Pod存在。

参数	说明
最大浪涌（maxSurge）	与spec.replicas相比，可以有多少个Pod存在，建议值是10%。 比如spec.replicas为3，那升级过程中就不能超过4个Pod存在，即按10%（1个）的步长升级，实际升级过程中会换算成数字，且换算会向上取整。这个值也可以直接设置成个数。

----结束

说明

- 仅配置升级策略参数maxUnavailable及maxSurge，不会触发工作负载滚动升级，即Pod不会发生重启。新配置值在工作负载的下次滚动升级中生效。
- 在CCE Console中执行工作负载的重新部署，实际上就是完成一次工作负载的滚动升级流程，只是工作负载的版本没有发生变化。

7.1.2 Pod 优雅删除

7.1.2.1 Istio-proxy 容器终止排出时间

步骤1 进入[CCE Console](#)页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。

```
394 * spec:
395   replicas: 1
396 * selector:
397 *   matchLabels:
398     app: istio-ingressgateway
399     istio: ingressgateway
400 * template:
401 *   metadata:
402     creationTimestamp: null
403 *   labels:
404     annotations:
405       proxy.istio.io/config: |
406         terminationDrainDuration: 300s
407     redeploy-timestamp: 'true'
408     sidecar.istio.io/inject: 'false'
```

步骤2 通过YAML配置如下参数：

```
spec:
  template:
    metadata:
      annotations:
        proxy.istio.io/config: |
          terminationDrainDuration: 300s
```

参数说明：

参数	说明
终止排出时间 （terminationDrainDuration）	Istio-proxy容器终止等待时间，Istio-ingressgateway场景下建议值是300秒。

----结束

说明

配置该参数会触发工作负载滚动升级，即Pod会立即发生重启。老Pod删除时按旧的配置值（5s）生效。

7.1.2.2 Pod 缩容时间窗

步骤1 进入CCE Console页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。

步骤2 通过YAML配置如下参数：

```
spec:
  template:
    spec:
      terminationGracePeriodSeconds: 301
```

参数说明：

参数	说明
缩容时间窗（terminationGracePeriodSeconds）	优雅删除时间，建议值是301秒（大于Istio-proxy容器终止排出时间），删除Pod时发送SIGTERM终止信号，然后等待容器中的应用程序终止执行，如果在terminationGracePeriodSeconds时间内未能终止，则发送SIGKILL的系统信号强行终止。

也可以通过单击步骤1页面待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：



----结束

说明

- 配置该参数会触发工作负载滚动升级，即Pod会立即发生重启。老Pod删除时按旧的配置值（30s）生效。
- 针对注入了Sidecar的业务负载，需要取Istio-proxy容器终止排出时间和业务容器优雅退出时间中的较大值。

7.1.3 调度策略

7.1.3.1 污点和容忍策略

- 步骤1 进入CCE Console页面，单击“集群名称-节点管理-节点”，单击更多列的|“编辑YAML”。
- 步骤2 通过YAML配置如下参数：

```
spec:
  taints:
  - key: istio
    value: ingressgateway
    effect: NoExecute
```

也可通过步骤1节点页面更多列的“污点管理”按钮配置。



- 步骤3 进入CCE Console页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。
- 步骤4 通过YAML配置如下参数：

```
spec:
  template:
    spec:
      tolerations:
      - key: istio
        operator: Equal
        value: ingressgateway
        effect: NoExecute
```

也可以通过单击步骤3页面待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：



容忍策略设置参数说明：

参数名	参数描述
污点键	节点的污点键。

参数名	参数描述
操作符	- Equal: 设置此操作符表示准确匹配指定污点键（必填）和污点值的节点。如果不填写污点值，则表示可以与所有污点键相同的污点匹配。 - Exists: 设置此操作符表示匹配存在指定污点键的节点，此时容忍度不能指定污点值。若不填写污点键则可以容忍全部污点。
污点值	操作符为Equal时需要填写污点值。
污点策略	- 全部: 表示匹配所有污点效果。 - NoSchedule: 表示匹配污点效果为NoSchedule的污点。 - PreferNoSchedule: 表示匹配污点效果为PreferNoSchedule的污点。 - NoExecute: 表示匹配污点效果为NoExecute的污点。
容忍时间窗	即tolerationSeconds参数，当污点策略为NoExecute时支持配置。在容忍时间窗内，Pod还会在拥有污点的节点上运行，超出时间后会被驱逐。

----结束

说明

配置该参数会触发工作负载滚动升级，即Pod会立即发生重启。

7.1.3.2 负载亲和调度策略

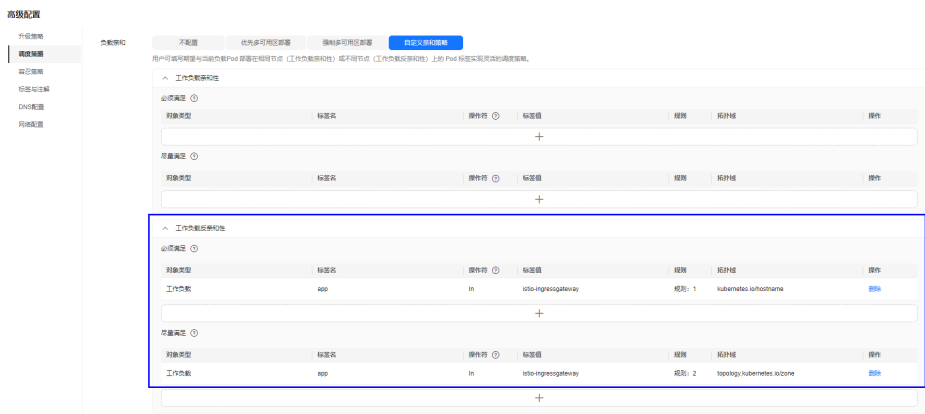
步骤1 进入[CCE Console](#)页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。

步骤2 通过YAML配置如下参数：

```
spec:
  template:
spec:
  affinity:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchExpressions:
              - key: app
                operator: In
                values:
                  - istio-ingressgateway
          namespaces:
            - istio-system
        topologyKey: kubernetes.io/hostname
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          podAffinityTerm:
            labelSelector:
              matchExpressions:
                - key: app
                  operator: In
                  values:
                    - istio-ingressgateway
            namespaces:
```

```
- istio-system
topologyKey: topology.kubernetes.io/zone
```

也可以通过单击[步骤1](#)页面待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：



建议值说明：

- 负载反亲和硬约束：Pod调度到不同的节点上。
- 负载反亲和软约束：Pod优先调度到不同可用区的节点上。

负载亲和参数说明：

策略	规则类型	说明
工作负载亲和性	必须满足	即硬约束，设置必须满足的条件，对应YAML定义中的requiredDuringSchedulingIgnoredDuringExecution字段。 通过标签筛选需要亲和的Pod，如果满足筛选条件的Pod已经运行在拓扑域中的某个节点上，调度器会将本次创建的Pod强制调度到该拓扑域。  说明： 添加多条亲和性规则时，即设置多个标签筛选需要亲和的Pod，则本次创建的Pod必须要同时亲和所有满足标签筛选的Pod，即所有满足标签筛选的Pod要处于同一拓扑域中才可以调度。
	尽量满足	即软约束，设置尽量满足的条件，对应YAML定义中的preferredDuringSchedulingIgnoredDuringExecution字段。 通过标签筛选需要亲和的Pod，如果满足筛选条件的Pod已经运行在拓扑域中的某个节点上，调度器会将本次创建的Pod优先调度到该拓扑域。  说明： 添加多条亲和性规则时，即设置多个标签筛选需要亲和的Pod，则本次创建的Pod会尽量同时亲和多个满足标签筛选的Pod。但即使所有Pod都不满足标签筛选条件，也会选择一个拓扑域进行调度。

策略	规则类型	说明
工作负载亲和性和性	必须满足	即硬约束，设置必须满足的条件，对应YAML定义中的requiredDuringSchedulingIgnoredDuringExecution字段。 通过标签筛选需要反亲和的一个或多个Pod，如果满足筛选条件的Pod已经运行在拓扑域中的某个节点上，调度器不会将本次创建的Pod调度到该拓扑域。  说明： 添加多条反亲和性规则时，即设置多个标签筛选需要反亲和的Pod，则本次创建的Pod必须要同时反亲和所有满足标签筛选的Pod，即所有满足标签筛选的Pod所处的拓扑域都不会被调度。
	尽量满足	即软约束，设置尽量满足的条件，对应YAML定义中的preferredDuringSchedulingIgnoredDuringExecution字段。 通过标签筛选需要反亲和的一个或多个Pod，如果满足筛选条件的Pod已经运行在拓扑域中的某个节点上，调度器会将本次创建的Pod优先调度到其他拓扑域。  说明： 添加多条反亲和性规则时，即设置多个标签筛选需要反亲和的Pod，则本次创建的Pod会尽量同时反亲和多个满足标签筛选的Pod。但即使每个拓扑域都存在需要反亲和的Pod，也会选择一个拓扑域进行调度。

负载亲和/反亲和调度策略参数说明：

参数名	参数描述
权重	仅支持在“尽量满足”策略中添加。权重的取值范围为1-100，调度器在进行调度时会将该权重加到其他优先级函数的评分上，最终将Pod调度到总分最大的节点上。
命名空间	指定调度策略生效的命名空间。
拓扑域	拓扑域（topologyKey）通过节点的标签先圈定调度的节点范围，例如标签指定为kubernetes.io/hostname，则根据标签值不同（标签值为节点名称）区分范围，不同名称的节点为不同的拓扑域，此时一个拓扑域中仅包含一个节点；如果指定标签为kubernetes.io/os，则根据标签值不同（标签值为节点的操作系统类型）来区分，不同操作系统的节点为不同的拓扑域，此时一个拓扑域中可能包含多个节点。 根据拓扑域确定节点范围后，然后再选择策略定义的内容（通过标签名、操作符、标签值确定）进行调度，调度时最小单位为拓扑域。例如，某个拓扑域中的一个节点满足负载亲和性规则，则该拓扑域中的节点均可以被调度。

参数名	参数描述
标签名	设置工作负载亲和/反亲和性时，填写需要匹配的工作负载标签。 该标签可以使用系统默认的标签，也可以使用自定义标签。
操作符	可以设置四种匹配关系（In、NotIn、Exists、DoesNotExist）。 • In：亲和/反亲和对象的标签在标签值列表（values字段）中。 • NotIn：亲和/反亲和对象的标签不在标签值列表（values字段）中。 • Exists：亲和/反亲和对象存在指定标签名。 • DoesNotExist：亲和/反亲和对象不存在指定标签名。
标签值	设置工作负载亲和/反亲和性时，填写工作负载标签对应的标签值。

----结束

📖 说明

配置负载亲和参数会触发工作负载滚动升级，即Pod会立即发生重启。

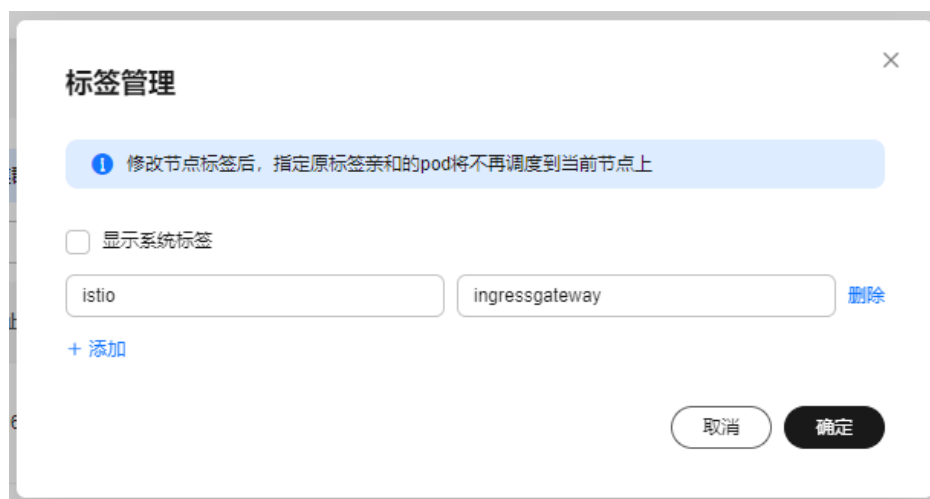
7.1.3.3 节点亲和调度策略

步骤1 进入[CCE Console](#)页面，单击“集群名称-节点管理-节点”，单击更多列的“编辑YAML”。

步骤2 通过YAML配置如下参数：

```
metadata:
  labels:
    istio: ingressgateway
```

也可在[步骤1节点页面](#)单击操作列的“标签管理”设置。



步骤3 进入[CCE Console](#)页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。

步骤4 通过YAML配置如下参数：

```
spec:
  template:
```

```
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: istio
                operator: In
                values:
                  - ingressgateway
```

也可以通过单击[步骤3页面](#)待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：



建议值说明：

节点亲和硬约束：Pod调度到包含istio:ingressgateway标签的节点上；

节点亲和参数说明：

参数名	参数描述
必须满足	即硬约束，设置必须要满足的条件，对应requiredDuringSchedulingIgnoredDuringExecution。 添加多条“必须满足”规则时，只需要满足一条规则就会进行调度。
尽量满足	即软约束，设置尽量满足的条件，对应preferredDuringSchedulingIgnoredDuringExecution。 添加多条“尽量满足”规则时，满足其中一条或者都不满足也会进行调度。

节点亲和性调度策略参数说明：

参数名	参数描述
标签名	设置节点亲和性时，填写需要匹配的节点标签。 该标签可以使用系统默认的标签，也可以使用自定义标签。

参数名	参数描述
操作符	可以设置六种匹配关系（In、NotIn、Exists、DoesNotExist、Gt、Lt）。 • In：亲和/反亲和对象的标签在标签值列表（values字段）中。 • NotIn：亲和/反亲和对象的标签不在标签值列表（values字段）中。 • Exists：亲和/反亲和对象存在指定标签名。 • DoesNotExist：亲和/反亲和对象不存在指定标签名。 • Gt：仅在节点亲和性中设置，调度节点的标签值大于列表值（字符串比较）。 • Lt：仅在节点亲和性中设置，调度节点的标签值小于列表值（字符串比较）。
标签值	设置节点亲和性时，填写节点标签对应的标签值。

----结束

📖 说明

配置节点亲和参数会触发工作负载滚动升级，即Pod会立即发生重启。

7.1.4 容器资源申请和限制

步骤1 进入[CCE Console](#)页面，单击“集群名称--工作负载”，单击待升级工作负载更多列的“编辑YAML”。

步骤2 通过YAML配置如下参数：

```
spec:
  template:
    spec:
      containers:
      - name: istio-proxy
        resources:
          limits:
            cpu: '2'
            memory: 4Gi
          requests:
            cpu: '2'
            memory: 4Gi
```

也可以通过单击[步骤1](#)页面待升级工作负载的操作列的“升级”按钮来配置，对应参数如下：



建议值说明：

- 网关对CPU敏感，建议申请和限制比例为1:1。避免申请值过小而限制值过大，容易导致节点超分严重。

- 2U4G仅为示例，实际配置值请根据压测结果自行调整。

CPU配额参数说明：

参数	说明
申请	容器使用的最小CPU需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配CPU总量 $\geq$ 容器CPU申请数时，才允许将容器调度到该节点。
限制	容器能使用的CPU最大值。

内存配额参数说明：

参数	说明
申请	容器使用的最小内存需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配内存总量 $\geq$ 容器内存申请数时，才允许将容器调度到该节点。
限制	容器能使用的内存最大值。当内存使用率超出设置的内存限制值时，该实例可能会被重启进而影响工作负载的正常使用。

----结束

说明

配置容器资源请求和限制会触发工作负载滚动升级，即Pod会立即发生重启。

7.2 网关 Service

7.2.1 服务亲和

注意

设置成节点亲和在某些场景下可能导致无法访问ELB，具体情况请参考 [集群内部无法使用ELB地址访问负载](#)处理。

步骤1 进入[CCE Console](#)页面，单击“集群名称--服务--服务”，单击要操作服务更多列的“编辑YAML”。

步骤2 通过YAML配置如下参数：

```
spec:
  type: LoadBalancer
  externalTrafficPolicy: Local
  allocateLoadBalancerNodePorts: true
```

也可在[步骤1服务页面](#)单击更多操作列的“更新”在更新服务页面进行设置。

更新服务

Service名称

test-svc

访问类型

 负载均衡

通过ELB负载均衡对外部提供服务，高可用，超高性能，稳定安全

服务亲和

集群级别

节点级别

?

社区在节点级别服务亲和的场景下存在访问限制。详情请参考 [集群内无法访问Service的说明](#)

命名空间

istio-system

选择器

键

=

值

确认添加

引用负载标签

app = istio-ingressgateway

istio = ingressgateway

服务通过选择器与负载（标签）关联，可将流量导向携带选择器标签的负载 Pod

负载均衡器

网络型规格

应用型规格

elb-zxx

负载均衡配置：分配策略：加权轮询算法；会话保持类型：不启用；[编辑](#)

☐ 我已阅读 [《负载均衡使用须知》](#)

参数说明：

参数	说明
服务亲和 (externalTrafficPolicy)	有如下两个取值： 集群级别（ cluster ）：请求会在集群内转发，从任意节点IP+服务端口都能访问到后端工作负载。服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源IP。 节点级别（ local ）：请求只会转发给本节点上的Pod，如果节点没有Pod的话请求会挂起。 当使用CCE Turbo集群 + 独享型ELB实例时，支持ELB直通Pod，使部署在容器中的业务时延降低、性能无损耗；此时取值必须为集群级别（ cluster ）。 其他场景从ELB过来的流量会先访问到节点，然后通过Service转发到Pod，建议值为节点级别（ local ）。流量最终转发给本节点上的Pod，而不会负载均衡到其他节点上的Pod，避免出现因跨节点网络故障而导致请求异常。

----结束

7.2.2 负载均衡器健康检查

- 步骤1 进入CCE Console页面，单击“集群名称--服务--服务”，单击要操作服务更多列的“编辑YAML”。
- 步骤2 通过YAML配置如下参数：

```
metadata:
  annotations:
```

```
kubernetes.io/elb.health-check-flag: 'on'
kubernetes.io/elb.health-check-option: '{"protocol":"TCP","delay":"5","timeout":"10","max_retries":"3"}'
```

参数说明：

参数	说明
kubernetes.io/elb.health-check-flag	是否开启ELB健康检查功能，建议值'on'。需同时填写kubernetes.io/elb.health-check-option字段。
kubernetes.io/elb.health-check-option	protocol：健康检查的协议。 delay：健康检查间隔（秒）。 timeout：健康检查的超时时间（秒）。 max_retries：健康检查的最大重试次数。

也可在[步骤1服务页面](#)单击操作列的“更新”在更新服务页面进行设置。

更新服务

Service名称

test-svc

访问类型

 负载均衡

通过ELB负载均衡对外部提供服务，高可用，超高性能，稳定安全

服务亲和

集群级别

节点级别

社区在节点级别服务亲和的场景下存在访问限制。详情请参考 [集群内无法访问Service的说明](#)

命名空间

istio-system

选择器

键

=

值

确认添加

[引用负载标签](#)

app = istio-ingressgateway

istio = ingressgateway

服务通过选择器与负载（标签）关联，可将流量导向携带选择器标签的负载 Pod

负载均衡器

网络型规格

应用型规格

elb-zxx

负载均衡配置：分配策略：加权轮询算法；会话保持类型：不启用；[编辑](#)

☐ 我已阅读《负载均衡使用须知》

健康检查

不启用

全局检查

自定义检查

协议：TCP | 检查周期（秒）：5 | 超时时间（秒）：10 | 最大重试次数：3

----结束