应用服务网格

用户指南

文档版本 02

发布日期 2023-05-30

华为技术有限公司

商标声明


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。

目 录

1 欢迎使用应用服务网格.....1
2 购买网格.....2
2.1 购买网格.....2
2.2 基础版、社区开源版本对比.....5
3 网格管理.....8
3.1 网格事件.....8
3.2 卸载网格.....9
4 服务管理.....11
4.1 配置诊断.....11
4.2 手动修复项.....13
4.2.1 所有 Pod 是否都配置了 app 和 version 标签.....13
4.2.2 所有 Pod 的 app 和 version 标签是否都相等.....14
4.2.3 所有 Pod 是否都注入了 sidecar.....15
4.3 自动修复项.....16
4.3.1 Service 的端口名称是否符合 istio 规范.....16
4.3.2 Service 的选择器中是否配置了 version 标签.....17
4.3.3 服务是否配置了默认版本的服务路由，路由配置是否正确.....17
5 网关管理.....19
5.1 添加网关.....19
5.2 添加路由.....22
6 灰度发布.....25
6.1 灰度发布概述.....25
6.2 创建灰度任务.....26
6.3 灰度任务基本操作.....30
7 网格配置.....33
7.1 网格配置概述.....33
7.2 sidecar 管理.....33
7.3 istio 资源管理.....37
7.3.1 YAML 方式配置 Istio 资源.....38
7.3.2 YAML 配置资源处理策略.....40
7.3.3 IstioOperator 配置资源处理策略.....41

7.4 升级 41
7.4.1 升级网格 42
7.4.2 1.3 版本特性 43
7.4.3 1.6 版本特性 43
7.4.4 1.8 版本特性 44
7.4.5 1.13 版本特性 44
7.4.6 1.15 版本特性 44
7.4.7 1.18 版本特性 44
7.4.8 1.3 升级 1.8 VirtualService 支持 Delegate 切换 44
7.5 网格扩展 47
8 流量治理 49
8.1 流量治理概述 49
8.2 配置流量策略 50
8.3 更改流量策略 54
9 安全 56
9.1 配置安全策略 56
9.2 JWT 认证原理 57
9.3 在 ASM 中对入口网关进行 JWT 请求认证 59

1 欢迎使用应用服务网格

应用服务网格（Application Service Mesh，简称ASM）是基于开源Istio推出的服务网格平台，它深度、无缝对接了企业级Kubernetes集群服务云容器引擎（CCE），在易用性、可靠性、可视化等方面进行了一系列增强，可为客户提供开箱即用的上手体验。

ASM提供非侵入式的微服务治理解决方案，支持完整的生命周期管理和流量治理，兼容Kubernetes和Istio生态，其功能包括负载均衡、熔断、限流等多种治理能力。ASM内置金丝雀、蓝绿等多种灰度发布流程，提供一站式自动化的发布管理。ASM基于无侵入的监控数据采集，提供实时流量拓扑、调用链等服务性能监控和运行诊断，构建全景的服务运行视图。

更多应用服务网格ASM介绍请参见**产品介绍**。

2 购买网格

2.1 购买网格

ASM支持购买基础版网格，该网格为标准版本网格，提供商用级网格服务。基础版网格的控制面组件安装在用户集群，支持对集群内的服务进行非侵入式的治理、灰度发布、流量监控等管理。兼容Istio 1.8、1.13、1.15和1.18版本，只能够对一个集群进行管理，且最大管理实例数为200。

前提条件

已创建CCE集群，如果未创建，请参照**购买CCE集群**创建。

约束与限制

- 应用服务网格依赖集群CoreDNS的域名解析能力，请确保集群拥有足够资源，且CoreDNS插件运行正常。
- 1.13和1.15版本的istio组件不支持在CentOS和EulerOS2.5操作系统的节点上运行，在创建网格时，请不要指定这些类型的节点为mastre节点。
- 集群启用Istio时，需要开通node节点（计算节点/工作节点）所在安全组的入方向7443端口规则，用于Sidecar自动注入回调。如果您使用CCE创建的默认安全组，此端口会自动开通。如果您自建安全组规则，请手动开通7443端口，以确保Istio自动注入功能正常

操作步骤

步骤1 登录应用服务网格控制台。

步骤2 单击右上角“购买网格”。

步骤3 设置网格参数。

- **网格类型**

  仅支持基础版，该网格为标准版本网格，提供商用级网格服务。

- **网格名称**

  网格的名称，取值必须以小写字母开头，由小写字母、数字、中划线（-）组成，且不能以中划线（-）结尾，长度范围为4~64个字符。

同一账号下网格不可重名，且网格名称创建后不可修改。

- **Istio版本**

  网格支持的Istio版本。

- **启用IPv6**

  是否启用IPv6功能，仅istio1.18及以上版本的基础版网格支持。

- **集群**

  在集群列表中选择集群，或在列表右上角输入集群名称搜索需要的集群。仅可选择当前网格版本支持的集群版本。

- **Istio控制面节点**

  基础版网格的控制面组件安装在用户集群，因此需要选择用于安装Istio控制面的节点。如果需要高可用，建议选择两个或以上不同可用区的节点。

  所选节点会被添加istio:master标签，网格组件会调度到该节点上。

- **设置可观测性配置**

  - 应用指标

    是否启用应用指标，应用指标开启后，可以在网格中构建服务访问指标、应用拓扑、服务健康和服务SLO定义。

  - 访问日志

    是否启用访问日志，访问日志开启后，可以在网格中查询到详细的服务间访问记录，定位每次访问异常。选择”访问日志”后，选择需要对接到的LTS日志组和LTS日志流，访问日志将传输到对应的LTS日志流，并可在“监控中心-访问日志”中查看访问日志。

    说明

    - 访问日志目前仅Istio 1.18及以上版本支持对接到云日志服务（LTS）。若要对接到LTS，请提前在CCE集群插件中心中安装云原生日志采集插件（CCE Log-Agent）。

  - 调用链

    采样率：用于调用链生成的请求占全部请求的采样百分比。

    选择使用服务：即选择调用链上报的服务后端。当选择“第三方Jaeger/Zipkin服务”时需要配置“服务地址”和“服务端口”两个参数，即第三方调用链服务接收请求信息的地址和端口。

    说明

    - 仅Istio 1.15及以上版本支持第三方调用链。
    - 如果您要使用“第三方Jaeger/Zipkin服务”调用链，请先自行完成调用链服务的安装，也可参考《常见问题》->《安装第三方调用链服务》进行安装之后获取服务地址。
    - Jaeger和Zipkin的默认服务端口均为9411，如果安装的时候自定义了服务端口在配置“服务端口”时请填写实际的值。

步骤4 （可选）高级配置。

- **sidecar配置**

  选择命名空间，为命名空间设置标签istio-injection=enabled，其中的Pod在重启后会自动注入istio-proxy sidecar。

如果不进行sidecar配置，可在网格创建成功后在“网格配置 > sidecar管理”中注入sidecar。具体操作请参考**sidecar注入**。

- **是否重启已有服务**

   ：会重启命名空间下已有服务关联的Pod，将会暂时中断业务。只有在重启后，已有服务关联的Pod才会自动注入istio-proxy sidecar。

   ：已有服务关联的Pod不会自动注入istio-proxy sidecar，需要在CCE控制台，手动重启工作负载才会注入sidecar。

- **流量拦截配置**

  说明

  默认情况下，ASM所注入的sidecar会拦截所有入方向和出方向流量，可通过流量拦截配置修改默认的流量拦截规则。

  入方向端口：可用于配置端口级别拦截规则，生效于网格服务的内部端口，以逗号分隔入站端口。

  - 仅拦截指定端口表示访问网格服务指定端口范围内的请求将被重定向到sidecar中。
  - 仅排除指定端口表示访问网格服务的请求，除端口范围外的请求将被重定向到sidecar中。

  出方向端口：可用于配置端口级别拦截规则，生效于网格服务对外访问的流量方向上，以逗号分隔出站端口。

  - 仅拦截指定端口表示网格服务访问指定端口范围内的请求将被重定向到sidecar中。
  - 仅排除指定端口表示网格服务对外访问的请求，除指定端口范围外的流量都将被重定向到sidecar中。

  出方向网段：可用于配置网段级别拦截规则，生效于网格服务对外访问的流量方向上，以逗号分隔 IP ，以 CIDR 形式表示。

  - 仅拦截指定网段表示指定网段范围内的流量将被重定向到sidecar中。
  - 仅排除指定网段表示除指定网段范围外的流量将被重定向到sidecar中。

- **资源标签**

  填写标签键和标签值，最多添加20个标签。

步骤5 设置完成后，在页面右侧配置清单确认网格配置，确认无误后，单击“提交”。

创建网格预计需要1~3分钟，请耐心等待。当网格状态从“安装中”变为“运行中”，表示网格创建成功。

说明

创建网格会自动创建一个otel-collector工作负载。详情请参考**otel-collector工作负载作用**。

启用网格期间会操作如下资源：

- 创建一个Helm应用编排release对象，作为服务网格控制面的资源。
- 开通节点的安全组，允许7443端口的入流量，使其支持对Pod进行自动注入。

----结束

2.2 基础版、社区开源版本对比

大类	功能项	功能点	社区开源版本	基础版
规格	管理规模	支持最大管理实例数	-	200
基础功能	服务发现和服务注册	通过服务中心集群获取服务列表、服务实例状态自动刷新、容器服务自动服务注册，业务无需实现注册逻辑、容器服务自动服务发现，业务无需实现发现订阅逻辑、服务实例副本数动态管理	√	√
	服务多版本	服务创建分版本管理、支持分版本进行监控	-	√
		支持分版本进行服务负载管理	√	√
	服务多端口	支持管理多个端口的服务、支持管理多个端口多协议的服务	√	√
		灰度发布支持多端口	-	√
	服务多形态	支持容器类型的服务后端	√	√
	协议及语言支持	HTTP协议灰度、治理、监控（根据协议特征不同细节不同，参照协议功能矩阵）、gRPC协议灰度、治理、监控（根据协议特征不同细节不同，参照协议功能矩阵）、开发语言无关、开发框架不限定、业务代码无侵入	√	√
	应用网关	支持四层协议对外访问、支持七层协议对外访问、支持入口路径映射、支持网关处TLS终止，支持配置对外证书和密钥	√	√
	负载均衡	支持轮询、随机、最小连接数以及一致性哈希的LB算法，可以基于特定的HTTP Header，或者基于Cookie值	√	√
	故障注入	支持注入指定时延或特定错误的故障，支持配置故障百分比	√	√
	熔断	支持配置最大请求数、每连接最大请求数、最大等待请求数、最大重试次数等七层请求管理；支持配置最大连接数、连接超时时间等四层连接管理；支持异常点检查、故障实例的自动隔离和自动恢复	√	√

大类	功能项	功能点	社区开源版本	基础版
	治理流量类型	支持对服务间内部通信流量进行治理、支持对服务外网访问流量（即Ingress流量）进行治理	√	√
	运行环境支持	支持容器应用治理	√	√
	认证	非侵入的双向TLS认证和通道加密	√	√
	授权	服务访问授权管理	√	√
	灰度发布	支持基于浏览器、操作系统、自定义HTTP Header、Cookie内容等配置灰度分流策略，支持基于URL配置灰度分流策略，支持基于请求参数、流量权重的灰度发布	√	√
		支持金丝雀灰度发布模板	-	√
		支持蓝绿灰度发布模板	-	√
		支持灰度发布过程中服务运行情况的监控以辅助灰度发布决策	-	√
		支持灰度发布过程中服务请求情况的监控以辅助灰度发布决策	-	√
		灰度发布时动态配置服务实例数	-	√
		支持灰度发布过程中动态的流量比例监控	-	√
	应用拓扑	提供应用下服务调用关系的全局拓扑	-	√
		提供拓扑图上各个服务间请求数、异常请求数等重要指标	-	√
		实时应用拓扑查看	-	√
	链路跟踪/调用链	支持非侵入调用链埋点	√	√
	指标监控	提供服务实例CPU、内存、磁盘等运行数据监控，提供服务访问RPS、时延等访问指标的监控，提供对访问指标、异常指标的统计分析，支持对接Prometheus等开源Metric组件，支持通过配置对接不同的Metric后端	√	√
	访问日志	访问日志非侵入采集	√	√

大类	功能项	功能点	社区开源版本	基础版
	安装	支持现有、新建Kubernetes集群按需一键安装启用Istio能力	-	√
	升级	支持控制面平滑升级，不中断应用业务	√	√
		支持数据面平滑升级，不中断应用业务	√	√
	插件管理	支持社区插件按需一键安装，支持Grafana、Prometheus	-	√
		支持社区插件按需一键安装，支持Tracing插件	-	√
		支持社区插件按需一键安装，支持Kiali插件	-	√
		支持社区插件按需一键安装，支持ELK插件	-	√
	代理管理	透明流量拦截、基于Iptables流量拦截、支持代理自动注入、支持Namespace级别和工作负载级别的注入管理	√	√
	代理形态	支持每Pod的Sidecar模式	√	√
	命令行工具	支持使用命令行进行流量策略管理（如istioctl、kubectl）	-	√

3 网格管理

3.1 网格事件

操作场景

应用服务网格支持事件中心，即通过界面提供网格创建、删除；网关创建、删除等重要操作的详细信息。

说明

在ASM 1.0中购买的基础版1.8以及以上版本的网格可以查看网格内事件。

操作步骤

步骤1 登录应用服务网格控制台，按照网格类型搜索基础版网格。

步骤2 单击网格右上角的图标，在右侧弹出的页面查看网格事件。

----结束

3.2 卸载网格

操作场景

当网格不再需要时，可以将其卸载。

约束与限制

- 如果网格有正在运行的灰度发布任务，请先完成灰度发布，再卸载网格。
- 请确保集群中有可用节点，用于运行清理任务，否则将导致卸载失败。

操作步骤

步骤1 登录应用服务网格控制台。

步骤2 单击对应网格下的 图标。

步骤3 在“卸载服务网格”页面，选择是否重启已有服务，并阅读注意事项。

卸载时，默认不会重启已有服务。只有在服务重启后才会去除注入的istio-poxy sidecar，如需重启，请选择“是”，重启服务将会暂时中断您的业务。

说明

建议您选择重启已有服务，如果不重启，会引起如下异常：当前网格卸载后，集群重新启用网格的情况下，网关会访问失败。

- 卸载服务网格将会卸载Istio控制面组件及数据面sidecar。
- 卸载后，应用的对外访问方式将无法继续使用，请将旧的对外访问方式改为用service方式对外发布。

  如需更新对外访问方式，请在CCE控制台“资源管理 > 网络管理 > Service”页面创建服务暴露对外访问方式。
- 卸载时将自动为您清理istio独享节点的相关标签，但不会删除istio-master节点，请到CCE界面删除，避免资源浪费。

  如需查看节点信息，请在CCE控制台“资源管理 > 节点管理”页面中查看。

图 3-1 卸载服务网格

----结束

4 服务管理

4.1 配置诊断

应用服务网格会对管理集群下的所有服务进行诊断，诊断结果为正常的服务，方可进行流量治理、流量监控及灰度发布等操作。

约束与限制

- 如果多个服务对应一个工作负载（Deployment），则不允许将这些服务加入网格进行治理，因为可能出现灰度发布、网关访问等功能异常。
- 如果服务的工作负载使用主机网络模式（Pod配置了hostNetwork: true），则不支持注入sidecar。

服务诊断

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“服务管理”，服务列表中展示了各服务的诊断结果。

如果服务存在异常，请单击“处理”，根据**服务异常修复**进行处理。

图 4-1 服务诊断

步骤3 修复异常项后，您可以单击“重新诊断”对服务进行再次诊断。

----结束

服务异常修复

诊断异常的服务，需要先手动修复异常状态的手动处理项，再一键修复可自动处理项。

步骤1 在诊断状态为异常的服务下单击“处理”，若手动处理项有异常，根据修复指导进行手动修复。

图 4-2 手动处理项

步骤2 手动修复异常状态的手动处理项后，单击“下一步”进入自动修复项页面，单击“一键修复”，自动处理异常状态的检查项。

图 4-3 自动处理项

说明

- 如果自动处理无法修复状态异常的检查项，请根据修复指导进行手动修复。
- 已配置网关或创建灰度发布的服务可能因为Service的端口名称被修改而出现异常，此时不支持进行一键修复。
- 如果服务未在服务列表中展示，请检查对应的工作负载是否存在。

----结束

4.2 手动修复项

4.2.1 所有 Pod 是否都配置了 app 和 version 标签

问题描述

Service关联的所有Pod都必须配置app和version标签。app标签在流量监控中用于流量的跟踪，version标签在灰度发布中用于区分不同版本。如果存在未配置app或version标签的Pod，则报此异常。

修复指导

Pod标签配置在Deployment的spec.template.metadata.labels中，建议配置为：

```
labels:
  app: {serviceName}
  version: v1
```

注意

修改或删除Deployment会触发Pod滚动升级，可能会导致业务短暂中断，请根据业务场景选择适当的时间修改。

步骤1 复制原有工作负载配置，保存为YAML文件。

kubectl get deployment {deploymentName} **-n** {namespace} **-o yaml >** {deploymentName}**-deployment.yaml**

例如：

kubectl get deployment productpage -n default -o yaml > productpage-deployment.yaml

步骤2 修改productpage-deployment.yaml内容，如果没有app和version，需要添加。app的值建议与Service名称一致，version建议为v1。

步骤3 删除原工作负载。

kubectl delete deployment {oldDeploymentName} **-n** {namespace}

步骤4 应用新的工作负载配置。

kubectl apply -f productpage-deployment.yaml

----结束

说明

1.15及以下集群还可以在CCE控制台直接修改Pod的标签，但有可能会导致ReplicaSet残留。

判断是否存在ReplicaSet残留的方法如下：

1. 查询Deployment的ReplicaSet。

   kubectl get replicaset | grep {deploymentName}

2. 找到实例个数大于1的ReplicaSet，如果ReplicaSet个数大于1，可能是修改label导致ReplicaSet残留。需要删除旧配置的ReplicaSet。

   kubectl delete replicaset {replicaSetName} **-n** {namespace}

4.2.2 所有 Pod 的 app 和 version 标签是否都相等

问题描述

Service关联的所有Pod的app和version标签必须都相等。app标签在流量监控中用于流量的跟踪，version标签在灰度发布中用于区分不同版本。如果存在app或version标签不相等的Pod，则报此异常。

修复指导

Pod标签配置在Deployment的spec.template.metadata.labels中，建议配置为：

```
labels:
  app: {serviceName}
  version: v1
```

修改多个Pod标签为相等的操作方法如下：

步骤1 查看Service选择器（spec.selector）配置的标签。

kubectl get svc {serviceName} **-o yaml**

例如，标签是app: ratings和release: istio-bookinfo。

步骤2 根据标签查找Service关联的Pod。

kubectl get pod -n {namespace} **-l app=ratings,release=istio-bookinfo**

说明

{namespace}和Service的namespace一致。

步骤3 根据Pod名称，找到其关联的工作负载。

kubectl get deployment {deploymentName} **-n** {namespace}

说明

- 一般Pod名称格式为{deploymentName}-{随机字符串}-{随机字符串}。
- 如果根据Pod名称未查询到工作负载，可能是因为ReplicaSet有残留，需要将其删除。

  判断是否存在ReplicaSet残留的方法如下：

  1. 查询Deployment的ReplicaSet。

     kubectl get replicaset | grep {deploymentName}

  2. 找到实例个数大于1的ReplicaSet，如果ReplicaSet个数大于1，可能是修改label导致ReplicaSet残留。需要删除旧配置的ReplicaSet。

     kubectl delete replicaset {replicaSetName} **-n** {namespace}

步骤4 请参考**修复指导**修改工作负载中Pod的app和version标签。

----结束

4.2.3 所有 Pod 是否都注入了 sidecar

问题描述

Service管理的所有Pod都必须存在istio-proxy容器，否则报此异常。

修复指导

步骤1 登录ASM控制台，选择服务所在网格。单击左侧导航中的“网格配置”，选择“sidecar管理”页签，检查服务所在命名空间是否已注入sidecar。

- 否，执行**步骤2**。
- 是，执行**步骤3**。

步骤2 注入sidecar。

可参考**sidecar注入**，为命名空间下所有工作负载关联的Pod注入sidecar。或者只为某个工作负载注入sidecar，方法如下：

1. 为工作负载所在命名空间打上istio-injection=enabled标签。

   kubectl label ns <namespace> **istio-injection=enabled**

2. 在CCE控制台为工作负载添加annotations字段。

```
annotations:
  sidecar.istio.io/inject: 'true'
```

```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: httpbin
      version: v1
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: httpbin
        version: v1
      annotations:
        sidecar.istio.io/inject: 'true'
```

您可以单击**Installing the Sidecar**了解更多sidecar注入的知识。

步骤3 如果网格已经开启了命名空间注入，但是Pod未注入sidecar，需要在CCE控制台手动重启Pod。方法如下：

登录CCE控制台，在工作负载所在行，单击操作列的“更多 > 重新部署”。

步骤4 检查工作负载是否配置了主机网络模式。方法如下：

登录CCE控制台，在工作负载所在行，单击操作列的“更多 > 编辑YAML”，查看是否配置了spec.template.spec.hostNetwork: true。如果是，请确认是否可将该字段删除或者配置为false，否则不允许注入sidecar。

```
125 spec:
126   replicas: 1
127   selector:
128     matchLabels:
129       app: nginx
130       version: v1
131   template:
132     metadata:
133       creationTimestamp: null
134       labels:
135         app: nginx
136         version: v1
137     spec:
138       hostNetwork: true
139       containers:
140         - name: container-1
141           image: nginx:alpine
```

步骤5 检查网格实例数量是否已经超出限额。

如果实例总数已经超出 200，那么超出部分的实例将无法注入sidecar。

----结束

4.3 自动修复项

4.3.1 Service 的端口名称是否符合 istio 规范

问题描述

Service端口名称必须包含指定的协议和前缀，按以下格式命名：

```
name: <protocol>[-<suffix>]
```

其中，<protocol>可以是http、tcp、grpc等，Istio根据在端口上定义的协议来提供对应的路由能力。例如“name: http-service0”和“name: tcp”是合法的端口名；而“name: httpforecast”是非法的端口名。

如果未按照以上格式命名，则报此异常。

修复指导

步骤1 登录CCE控制台，单击集群名称进入详情页面。

步骤2 在左侧导航栏选择“资源 > 服务发现”，单击对应服务后的“更多 > 编辑YAML”，查看Service协议，根据支持协议，修改协议，在服务名称前加协议类型，如下图。

```
15 spec:
16   ports:
17     - name: http-ratings
18       protocol: TCP
19       port: 9080
20       targetPort: 9080
```

步骤3 单击“确定”。

----结束

4.3.2 Service 的选择器中是否配置了 version 标签

问题描述

Service的选择器（spec.selector）中不能包含version标签。如果包含，则报此异常。

修复指导

步骤1 登录CCE控制台，单击集群名称进入详情页面。

步骤2 在左侧导航栏选择“资源 > 服务发现”，单击对应服务后的“更多 > 编辑YAML”，查看Service的选择器（spec.selector），删除已配置的version标签。

```
spec:
  ports:
    - name: http-service0
      protocol: TCP
      port: 8000
      targetPort: 80
  selector:
    app: nginx
    version: v1
```

----结束

4.3.3 服务是否配置了默认版本的服务路由，路由配置是否正确

问题描述

Istio在VirtualService和DestinationRule中定义了服务的流量路由规则，所以需要为每个服务配置VirtualService和DestinationRule，需要满足以下的规则：

- VirtualService中必须配置了Service的所有端口。
- VirtualService中的协议类型必须和Service中端口协议类型一致。
- VirtualService和DestinationRule中必须配置了默认的服务版本。

说明

如果检查结果发生改变，可能Service的端口号或端口名称被修改。

修复指导

步骤1 登录ASM控制台，选择服务所在网格，单击左侧导航中的“网格配置”，选择“istio资源管理”页签，在搜索框中选择“istio资源：virtualservices”及服务所属命名空间。

步骤2 确保VirtualService中必须配置了Service的所有端口。


```
spec:
  hosts:
    - reviews
  http:
    - match:
        - gateways:
            - mesh
          port: 9080
      route:
        - destination:
            host: reviews.default.svc.cluster.local
            port:
              number: 9080
            subset: v1
          weight: 50
        - destination:
            host: reviews.default.svc.cluster.local
            port:
              number: 9080
            subset: v2
          weight: 50
```

步骤3 确保VirtualService中的协议类型必须和Service中端口协议类型一致。

图 4-4 VirtualService 的协议类型

```
34 spec:
35   hosts:
36     - ratings
37   http:
38     - route:
39         - destination:
40             host: ratings
41             port:
42               number: 9080
43             subset: v1
```

图 4-5 Service 的端口协议类型

```
13 spec:
14   ports:
15     - name: http-ratings
16       protocol: TCP
17       port: 9080
18       targetPort: 9080
19   selector:
20     app: ratings
```

----结束

5 网关管理

5.1 添加网关

服务网关在微服务实践中可以做到统一接入、流量管控、安全防护、业务隔离等功能。

前提条件

服务网关使用弹性负载均衡服务（ELB）的负载均衡器提供网络访问，因此在添加网关前，请提前创建负载均衡。

创建负载均衡时，需要确保所属VPC与集群的VPC一致，详情请参见**创建共享型负载均衡器**。

操作步骤

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网关管理”，单击“添加网关”。

步骤3 配置网关参数。

- **网关名称**

  请输入网关的名称。由小写字母、数字和中划线（-）组成，且必须以小写字母开头，小写字母或数字结尾，长度范围为4~59个字符。

- **集群选择**

  选择网关所属的集群。

- **访问方式**
  - 协议版本：支持选择IPV4和双栈两种，已开启ipv6的网格方可配置该参数。
  - 服务网关使用弹性负载均衡服务（ELB）的负载均衡实例提供网络访问，支持共享型实例规格，实例的网络类型支持IPv4公网和IPv4私网。

- **访问入口**
  - 对外协议

    请根据业务的协议类型选择。支持HTTP、GRPC、TCP、TLS及HTTPS五种协议类型的选择。

- 对外端口

  开放在负载均衡服务地址的端口，可任意指定。

- 外部访问地址

  系统自动填充负载均衡实例的IP地址，作为服务访问入口地址，您也可以将其修改为负载均衡实例关联的域名。

- TLS终止

  对外协议为HTTPS时，TLS终止为开启状态，且不可关闭。

  对外协议为TLS时，可选择开启/关闭TLS终止。开启TLS终止时需要绑定证书，以支持TLS数据传输加密认证；关闭TLS终止时，网关将直接转发加密的TLS数据。

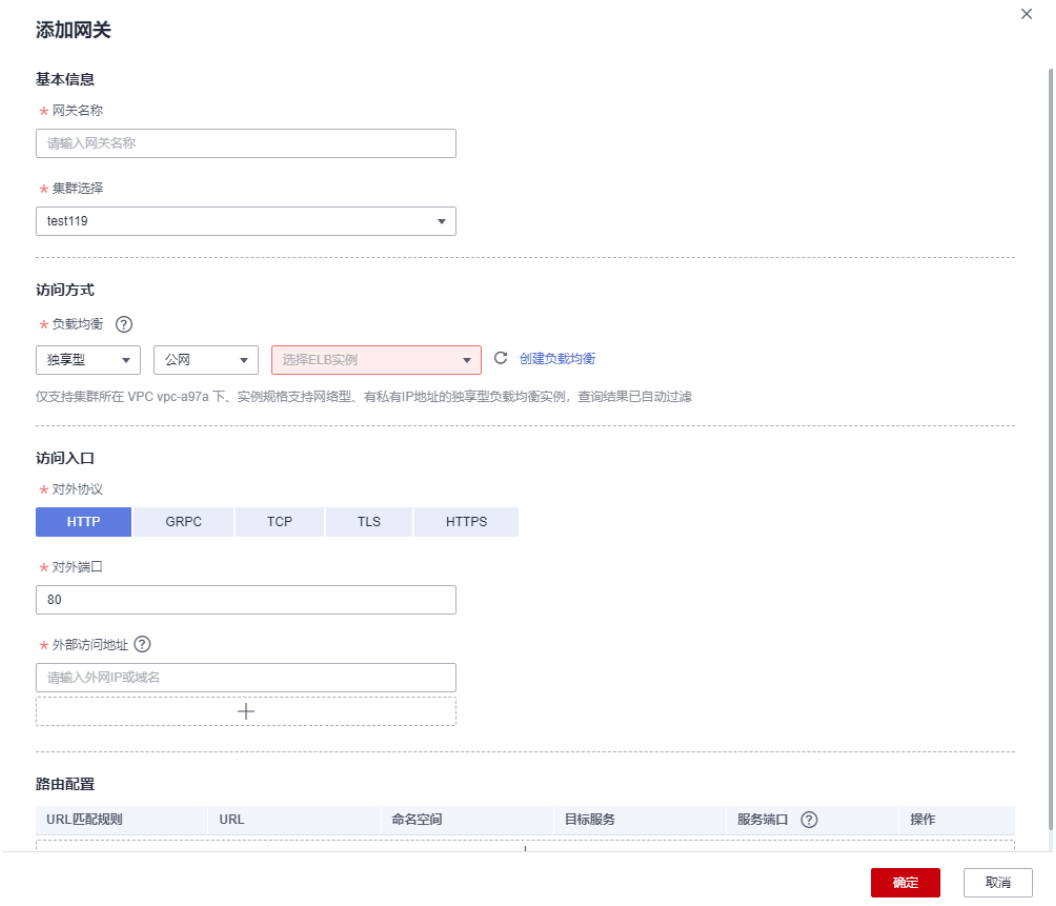
- 密钥证书

  - 配置TLS协议并开启TLS时，需要绑定证书，以支持TLS数据传输加密认证。
  - 配置HTTPS协议时，需要绑定密钥证书。

- TLS最低版本/TLS最高版本

  配置TLS协议并开启TLS终止，或者配置HTTPS协议时，提供TLS最低版本/TLS最高版本的选择。

图 5-1 添加网关



步骤4 （可选）配置路由参数。

请求的访问地址与转发规则匹配（转发规则由外部访问地址+URL组成）时，此请求将被转发到对应的目标服务处理。单击┼图标，弹出“添加路由”对话框。

- **URL匹配规则**
  - 前缀匹配：例如映射URL为/healthz，只要符合此前缀的URL均可访问。例如/healthz/v1、/healthz/v2。
  - 完全匹配：只有完全匹配上才能生效。例如映射URL为/healthz，则必须为此URL才能访问。
- **URL**

  服务支持的映射URL，例如/example。
- **命名空间**

  服务网关所在的命名空间。
- **目标服务**

  添加网关的服务，直接在下拉框中选择。目标服务会根据对应的网关协议进行过滤，过滤规则请参见**添加路由时，为什么选不到对应的服务？**。

  配置诊断失败的服务无法选择，需要先根据**手动修复项**或**自动修复项**进行修复。
- **访问端口**

  仅显示匹配对外协议的端口。
- **重写**

  （对外协议为HTTP时可配置）

  重写HTTP URI和Host/Authority头，于转发前执行。默认关闭。开启后，需要配置如下参数：
  - URI：使用此值重写URI的路径（或前缀），如果原始URI是基于前缀匹配，那么将替换相应匹配的前缀。
  - Host/Authority头：使用此值重写HTTP的Host/Authority头。

图 5-2 添加路由

步骤5 配置完成后，单击“确定”。

网关添加完成后，可前往“服务管理”页面，获取服务外网访问地址。

图 5-3 服务外网访问地址

----结束

5.2 添加路由

操作场景

您可以给已创建好的网关添加多个路由，配置多个转发策略。

操作步骤

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网关管理”，在需要添加路由的网关所在行，单击操作列的“添加路由”，配置如下参数。

- **URL匹配规则**
  - 前缀匹配：例如映射URL为/healthz，只要符合此前缀的URL均可访问。例如/healthz/v1、/healthz/v2。
  - 完全匹配：只有完全匹配上才能生效。例如映射URL为/healthz，则必须为此URL才能访问。
- **URL**

  服务支持的映射URL，例如/example。

  说明

  同一网关下的URL配置不能相同。
- **命名空间**

  服务网关所在的命名空间。
- **目标服务**

  添加网关的服务，直接在下拉框中选择。目标服务会根据对应的网关协议进行过滤，过滤规则请参见。

  配置诊断失败的服务无法选择，需要先根据**手动修复项**或**自动修复项**进行修复。
- **访问端口**

  仅显示匹配对外协议的端口。
- **重写**

  （对外协议为HTTP时可配置）

  重写HTTP的URI和Host/Authority头，于转发前执行。默认关闭。开启后，需要配置如下参数：
  - URI：使用此值重写URI的路径（或前缀），如果原始URI是基于前缀匹配，那么将替换相应匹配的前缀。
  - Host/Authority头：使用此值重写HTTP的Host/Authority头。

图 5-4 添加路由

步骤3 配置完成后，单击“确定”。

----结束

6 灰度发布

6.1 灰度发布概述

应用程序升级面临最大挑战是新旧业务切换，将软件从测试的最后阶段带到生产环境，同时要保证系统不间断提供服务。如果直接将某版本上线发布给全部用户，一旦遇到线上事故（或BUG），对用户的影响极大，解决问题周期较长，甚至有时不得不回滚到前一版本，严重影响了用户体验。

长期以来，业务升级逐渐形成了几个发布策略：金丝雀发布、蓝绿发布、A/B测试、滚动升级以及分批暂停发布，尽可能避免因发布导致的流量丢失或服务不可用问题。ASM当前支持金丝雀发布和蓝绿发布两种发布方式。

金丝雀发布

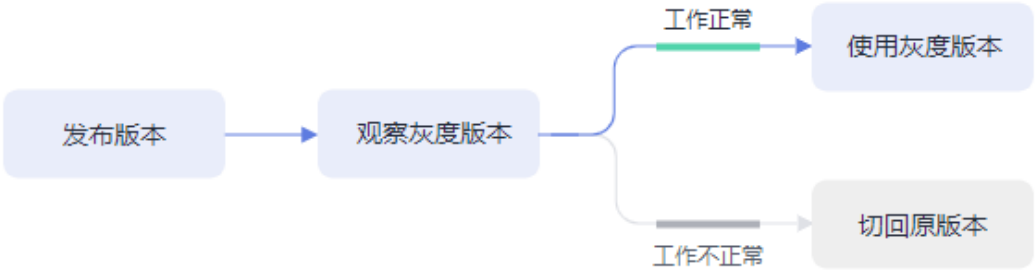
又称灰度发布，是版本升级平滑过渡的一种方式，当版本升级时，使部分用户使用新版本，其他用户继续使用老版本，待新版本稳定后，逐步扩大范围把所有用户流量都迁移到新版本上面来。这样可以最大限度地控制新版本发布带来的业务风险，降低故障带来的影响面，同时支持快速回滚。

图 6-1 金丝雀发布流程

蓝绿发布

蓝绿发布提供了一种零宕机的部署方式，是一种以可预测的方式发布应用的技术，目的是减少发布过程中服务停止的时间。在保留老版本的同时部署新版本，将两个版本同时在线，新版本和老版本相互热备，通过切换路由权重的方式（非0即100）实现应用的不同版本上线或者下线，如果有问题可以快速地回滚到老版本。

图 6-2 蓝绿发布流程



6.2 创建灰度任务

基本概念

- 灰度版本

  一个服务仅支持发布一个灰度版本，可以对灰度版本配置相应的灰度策略。

- 灰度策略

  当您需要在生产环境发布一个新的待上线版本时，您可以选择添加一个灰度版本，并配置相应的灰度策略，将原有的生产环境的默认版本的流量引流一部分至待上线版本。经过评估稳定后，可以将此灰度版本接管所有流量，下线原来的版本，从而接管原有的生产环境的版本上的流量。

创建灰度发布

步骤1 登录应用服务网格控制台，使用以下任意一种方式进入创建灰度任务页面。

- （快捷方式）在网格右上方，单击图标。
- （快捷方式）在网格中心位置，单击“创建灰度任务”。
- 在网格详情页创建。
  a. 单击网格名称，进入网格详情页，单击左侧导航栏的“灰度发布”。
  b. 如果当前不存在发布中的灰度任务，请在金丝雀发布或蓝绿发布中单击“立即发布”；如果当前存在发布中的灰度任务，请单击右上角“灰度发布”。

步骤2 配置灰度发布基本信息。

- **灰度类型**

  选择创建灰度发布的类型，可根据实际需求选择金丝雀发布和蓝绿发布，两者的区别可参考**灰度发布概述**。

- **灰度任务名称**

  自定义灰度任务的名称。输入长度范围为4到63个字符，包含小写英文字母、数字和中划线（-），并以小写英文字母开头，小写英文字母或数字结尾。

- **命名空间**

  服务所在的命名空间。

- **灰度发布服务**

  在下拉列表中选择待发布的服务。正在进行灰度任务的服务不可再进行选择，列表中已自动过滤。

- **工作负载**
  选择服务所属的工作负载。
- **版本号**
  当前服务版本号，版本号不支持修改。

图 6-3 灰度发布基本信息

步骤3 部署灰度版本信息。

- **部署集群**
  灰度发布服务所属的集群。
- **版本号**
  输入服务的灰度版本号。
- **实例数量**
  灰度版本的实例数量。灰度版本可以有一个或多个实例，用户可根据实际需求进行修改。每个灰度版本的实例都由相同的容器部署而成。
- **镜像名称**
  默认为该服务的镜像。
- **镜像版本**
  请选择灰度版本的镜像版本。
- **自定义镜像**
  用户可通过自定义镜像自行配置本地或第三方镜像地址，灰度发布镜像将采用所配置镜像。注意需确保自定义镜像地址有效，镜像可拉取。

图 6-4 灰度版本信息

步骤4 单击“发布”，灰度版本开始创建。

请确保灰度版本的实例状态正常，且启动进度为100%时，再开始下一步进行流量策略的配置。发布之后进入观察灰度状态页面，可查看Pod监控，包括启动日志和性能监控信息。

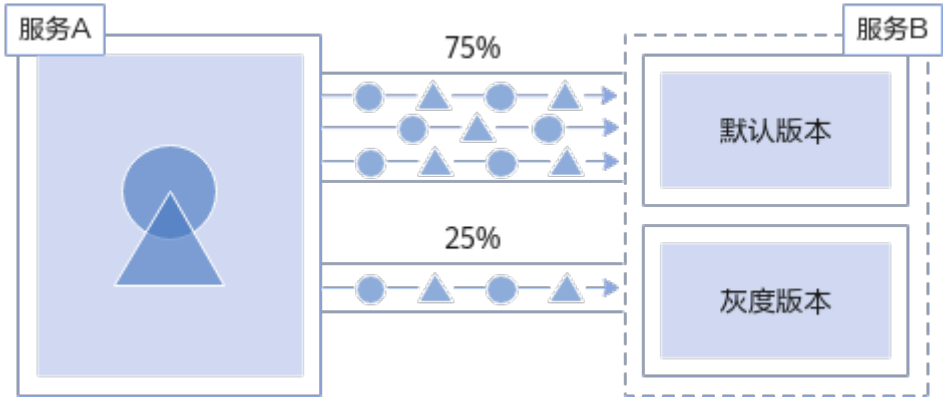
步骤5 （仅金丝雀发布涉及）单击“配置流量策略”，进行流量策略配置。

策略类型：分为“基于流量比例”和“基于请求内容”两种类型，通过页签选择确定。

- **基于流量比例**

  根据流量比例配置规则，从默认版本中切分指定比例的流量到灰度版本。例如75%的流量走默认版本，25%的流量走灰度版本。实际应用时，可根据需求将灰度版本的流量配比逐步增大并进行策略下发，来观测灰度版本的表现情况。

  图 6-5 基于流量比例

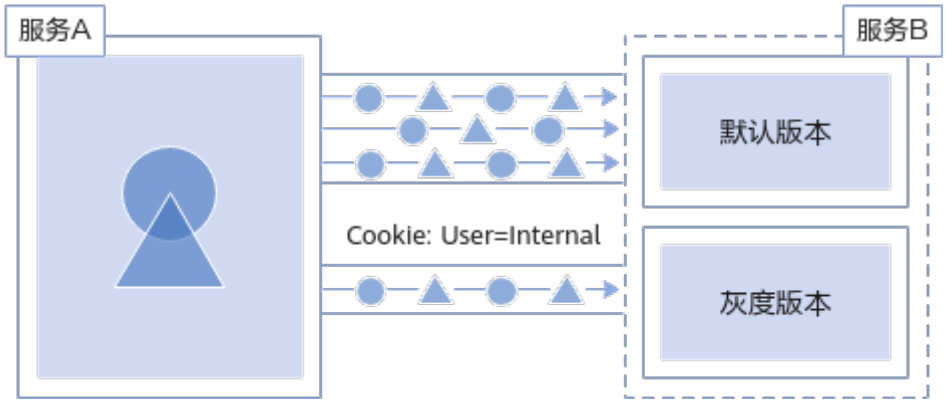
  

  流量配比：可以为默认版本与灰度版本设置流量配比，系统将根据输入的流量配比来确定流量在两个版本间分发的比重。

- **基于请求内容**

  目前支持基于Cookie内容、自定义Header、Query、操作系统和浏览器的规则约束，只有满足规则约束的流量才可访问到灰度版本。例如，仅Cookie满足“User=Internal”的HTTP请求才能转发到灰度版本，其余请求仍然由默认版本接收。

  图 6-6 基于请求内容

  - Cookie内容

    正则匹配：此处需要您使用正则表达式来匹配相应的规则。
  - 自定义Header
    - 完全匹配：只有完全匹配上才能生效。例如：设置Header的Key=User，Vaule=Internal，那么仅当Header中包含User且值为Internal的请求才由灰度版本响应。
    - 正则匹配：此处需要您使用正则表达式来匹配相应的规则。

      可以自定义请求头的key和value，value支持完全匹配和正则匹配。
  - Query
    - 完全匹配：只有完全匹配上才能生效。例如：设置Query的Key=User，Vaule=Internal，那么仅当Query中包含User且值为Internal的请求才由灰度版本响应。
    - 正则匹配：此处需要您使用正则表达式来匹配相应的规则。

      可以自定义Query的key和value，value支持完全匹配和正则匹配。
  - 允许访问的操作系统：请选择允许访问的操作系统，包括iOS、Android、Windows、macOS。
  - 允许访问的浏览器：请选择允许访问的浏览器，包括Chrome、IE。
  - 流量管理Yaml信息：根据所设置的参数自动生成规则YAML。

  说明

  基于请求内容流量策略只对直接访问的入口服务有效。如果希望对所有服务有效，需要业务代码对HTTP请求的Header信息进行传播。

  例如：如果您基于reviews服务，配置了基于请求内容的灰度发布策略，通过访问productpage服务的界面，是无法看到效果的。

  原因是，您的客户端访问productpage服务携带了HTTP请求的Header信息，而productpage服务请求reviews服务时，将这些Header信息丢失了，从而失去了基于请求内容的灰度发布效果。

步骤6 设置完成后，单击“策略下发”。

灰度策略的生效需要几秒时间，您可以查看服务的流量监控，以及对原始版本及灰度版本的健康监控。

----结束

6.3 灰度任务基本操作

使用说明

对灰度版本的相关操作，其原理是修改Istio的DestinationRule和VirtualService两个资源的配置信息。修改完成后，需要等待10秒左右，新的策略规则才会生效。

修改灰度版本的流量策略

修改基于流量比例的策略

选择基于流量比例的策略时，一般会逐步加大灰度版本的流量配比，这样可以避免直接切换带来的业务风险。修改流量配比的方法如下：

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击金丝雀发布任务的名称。

步骤3 在“配置流量策略”页面，重新输入灰度版本的流量配比。

假设将灰度版本流量配比调整至x，那么原版本的流量配比自动调整为100-x。

步骤4 单击“策略下发”。

----结束

修改基于请求内容的策略

目前支持基于Cookie内容、自定义Header、Query、操作系统和浏览器的规则约束，只有满足规则约束的流量才可访问到灰度版本。实际应用时，可能会多次修改规则，从而充分验证灰度版本运行效果。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击金丝雀发布任务的名称。

步骤3 在“配置流量策略”页面，重新配置Cookie内容、自定义Header、Query、允许访问的操作系统或允许访问的浏览器。

步骤4 单击“策略下发”。

----结束

切换灰度策略类型

您可以在“基于请求内容”和“基于流量比例”的策略之间切换。策略完成切换后，之前配置的规则将全部失效，所有的流量会根据配置的新策略重新分配。

须知

只有状态为“运行中”的任务才支持流量策略变更，版本发布完成后（即新版本完全接管旧版本流量，且旧版本已下线），将不支持重新配置流量策略。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击金丝雀发布任务的名称。

步骤3 在“配置流量策略”页面，切换策略类型。

步骤4 单击“策略下发”。

----结束

全流量接管

执行原版本或灰度版本后的“全流量接管”，原版本或灰度版本将接管全部流量。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击灰度发布任务的名称。

步骤3 在“监测与处理”页面，单击版本后的“全流量接管”。

图 6-7 全流量接管

步骤4 在弹出的“全流量接管”窗口单击“确定”，此版本即可接管全部流量。

----结束

结束灰度任务

当新创建的灰度版本接管全部流量后，您可以选择结束灰度任务。结束灰度任务将下线原版本，其中包含的工作负载和Istio相关配置资源会全部删除。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击灰度发布任务的名称。

步骤3 在“监测与处理”页面，单击灰度版本后的“全流量接管”。

步骤4 单击右下角“结束灰度任务”。

步骤5 在弹出的“结束灰度任务”窗口单击“确定”。

结束的灰度任务可以前往“已结束灰度任务”页签查看，状态显示为“发布成功”。

----结束

取消灰度任务

当原版本接管全部流量后，您可以选择取消灰度任务。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击灰度发布任务的名称。

步骤3 在“监测与处理”页面，单击原版本后的“全流量接管”。

步骤4 单击右下角“取消灰度任务”。也可以在灰度任务列表，单击任务右上角的图标。

步骤5 在弹出的“取消灰度任务”窗口单击“确定”。

取消的灰度任务可以前往“已结束灰度任务”页签查看，状态显示为“发布取消”。

----结束

查看已结束灰度任务

取消的灰度任务，以及结束的灰度任务均可在“已结束灰度任务”页签查看。

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“灰度发布”，单击“已结束灰度任务”页签。

您可以查看：发布任务名称、发布结果、服务、发布时间，还可以删除已结束的灰度任务。

----结束

7 网格配置

7.1 网格配置概述

网格配置提供了集群管理、sidecar管理、istio资源管理以及升级能力。

Istio控制面组件负责向数据面组件注入sidecar，管理数据面sidecar行为，下发策略配置，搜集监控数据等。其中，sidecar是指运行在业务Pod中，与业务容器协同工作，负责业务Pod的路由转发，监控数据采集，流量规则配置等功能。

“网格配置”中各个页签的功能如下：

- “基本信息”页签：可查看网格名称、网格ID、网格状态、网格类型、当前版本、计费模式、创建时间以及已启用网格的集群。
- “sidecar管理”页签：支持查看所有注入了sidecar的工作负载信息，还可以进行sidecar注入、配置sidecar资源限制等操作。详情请参见**sidecar管理**。
- “istio资源管理”页签：展示所有istio资源（如VirtualService、DestinationRule），还支持以YAML或JSON格式创建新的istio资源，或对现有istio资源进行修改。详情请参见**istio资源管理**。
- “升级”页签：提供网格版本升级能力。详情请参见**升级网格**。

7.2 sidecar 管理

sidecar管理中支持查看所有注入了sidecar的工作负载信息，还可以进行sidecar注入、配置sidecar资源限制等操作。

sidecar 注入

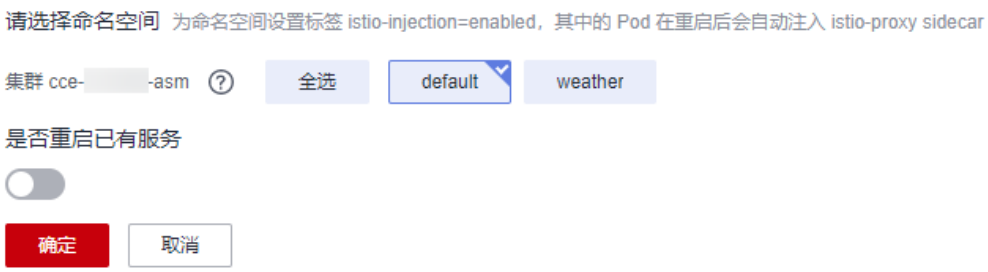
可展示当前已注入sidecar的命名空间及所属集群。如果还未做过注入操作，或者需要为更多命名空间注入sidecar，请参考以下操作：

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网格配置”，单击“sidecar管理”页签。

步骤3 单击“sidecar管理”，选择命名空间，判断是否重启已有服务，单击“确定”。

图 7-1 注入 sidecar



- 选择命名空间：选择一个或多个命名空间，系统将为命名空间设置标签istio-injection=enabled。
- 是否重启已有服务：

   ：会重启命名空间下已有服务关联的Pod，将会暂时中断业务。只有在重启后，已有服务关联的Pod才会自动注入istio-proxy sidecar。

   ：已有服务关联的Pod不会自动注入istio-proxy sidecar，需要在CCE控制台，手动重启工作负载才会注入sidecar。是否重启已有服务只会影响已有服务，只要为命名空间设置了istio-injection=enabled标签，后面新建的服务实例都会自动注入sidecar。

说明

- 若界面提示“以下集群未开放命名空间注入修改操作”，需要通过kubectl命令行开放，具体操作请参见**如何为集群开放命名空间注入？**。
- 为集群的命名空间开启sidecar注入后，该命名空间下所有工作负载关联的Pod将自动注入sidecar。如果某些工作负载不希望注入sidecar，可参考**某些工作负载不注入sidecar，该如何配置？**进行配置。
- 对于1.8.4-r1及之前（所有1.3、1.6）版本的网格，建议确认工作负载是否包含注解：sidecar.istio.io/inject: 'true'。如未包含，请在spec.template.metadata.annotations字段下添加：

```
annotations:
  sidecar.istio.io/inject: 'true'
```

----结束

查看工作负载详情

列表中展示了该网格所管理的集群下所有已创建服务的工作负载，支持查看负载的名称、所属集群、服务，以及负载的sidecar信息，包括sidecar名称、sidecar版本、状态、CPU使用率、内存使用率等。操作方法如下：

步骤1 在列表右上角搜索框，选择集群、命名空间，并输入工作负载名称搜索指定工作负载，查看负载的相关信息。

步骤2 单击工作负载前的 图标，查看负载的sidecar信息。

如果提示工作负载中无sidecar，是因为该负载所属命名空间还未注入sidecar，参考**sidecar注入**进行注入。

----结束

配置 sidecar 资源限制

支持为sidecar（即istio-proxy容器）配置CPU和内存的资源上下限。同一个节点上部署的工作负载，对于未设置资源上下限的工作负载，如果其异常资源泄露会导致其他工作负载分配不到资源而异常。未设置资源上下限的工作负载，工作负载监控信息也会不准确。

默认的sidecar资源上下限为：

- CPU（Core）：最小 0.1，最大 2
- MEM（MiB）：最小 128，最大 1024

如需更改，请参考以下操作：

步骤1 单击工作负载操作列的“sidecar资源限制”，也可以勾选多个工作负载，在列表左上角单击“sidecar资源限制”进行批量配置。

图 7-2 sidecar 资源限制

- CPU最小值：也称CPU请求，表示容器使用的最小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 ≥ 容器CPU请求数时，才允许将容器调度到该节点。
- CPU最大值：也称CPU限制，表示容器能使用的CPU最大值。
- MEM最小值：也称内存请求，表示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 ≥ 容器内存请求数时，才允许将容器调度到该节点。
- MEM最大值：也称内存限制，表示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

为 sidecar 资源配置内存告警

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。因此，建议用户为所有sidecar资源配置内存告警，当告警被触发后，及时扩容对应sidecar的内存最大值，可降低业务影响范围。

步骤1 登录应用运维管理 AOM控制台。

步骤2 在左侧导航栏选择“告警 > 告警规则”，单击右上角“添加规则”。

步骤3 设置告警规则。

- 规则名称：输入规则名称，例如：istio-proxy。
- 规则类型：选择“阈值规则”。
- 监控对象：单击“选择资源对象”，选择“按指标维度添加”，指标名称选择“云服务指标 > CCE > 容器 > 物理内存使用率”。

  指标维度选择“集群ID”，选择开启告警的集群ID；选择“容器名称”，容器名称选择“istio-proxy”，单击“确定”。

  图 7-3 选择监控对象

- 告警条件：设置统计周期、连续周期、阈值条件等触发条件参数，如下图所示，请根据具体需求设置。

  图 7-4 设置告警条件

- 告警方式：选择“直接告警”。

步骤4 单击“立即创建”。

创建后在规则列表中可以看到如下一行，表示创建成功。

图 7-5 规则列表

步骤5 在告警列表中可以查看最新的活动告警信息及对应负载的详情。

当告警被触发后，活动告警列表会新增一条或多条记录。单击告警名称，在“告警对象”页签查看deploymentName、nameSpace等参数，以确定告警源自哪个sidecar资源。业务面sidecar资源限制请参考**配置sidecar资源限制**修改，istio-ingressgateway、istio-egressgateway、istio-eastwestgateway等控制面sidecar资源限制需要在CCE控制台“工作负载”页面通过升级方式来修改。

图 7-6 告警详情

----结束

7.3 istio 资源管理

7.3.1 YAML 方式配置 Istio 资源

网格中服务关联的Istio资源（如VirtualService、DestinationRule）如需修改，可以在“istio资源管理”中以YAML或JSON格式进行编辑。同时，还支持创建新的istio资源。

⚠ 注意

使用YAML方式创建和修改Istio资源可能导致与控制台配置不兼容，控制台相关功能将不再开放使用。如果您确定后续将只通过YAML方式维护Istio资源，则参考本章节编辑或新建Istio资源，否则请不要在此处操作。

编辑已有 istio 资源

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网格配置”，单击“istio资源管理”页签。

步骤3 在搜索框中选择istio资源类型（如“istio资源：virtualservices”），以及资源所属命名空间。

图 7-7 筛选 istio 资源

步骤4 单击操作列的“编辑”，在右侧页面修改相关配置，默认勾选底部提示信息，即控制台相关功能将不再开放使用，单击“确定”保存。

说明

编辑Istio资源后，具体哪些控制台功能不可用，与Istio资源类型有关。详细说明请参见**YAML配置资源处理策略**。

支持以YAML或JSON格式显示，同时可以将配置文件下载到本地。

----结束

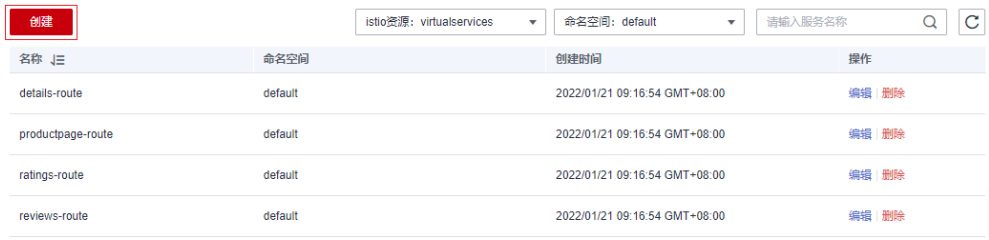
创建新的 istio 资源

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网格配置”，单击“istio资源管理”页签。

步骤3 单击列表左上方的“创建”。

图 7-8 创建 istio 资源



步骤4 在右侧页面直接输入内容，或者单击“导入文件”，将本地编辑好的YAML或JSON文件上传上来。

步骤5 确认文件内容无误，默认勾选底部提示信息，即控制台相关功能将不再开放使用，单击“确定”保存。

说明

创建Istio资源后，具体哪些控制台功能不可用，与Istio资源类型有关，详细说明请参见**YAML配置资源处理策略**。

----结束

Istio 资源说明

表 7-1 Istio 资源说明

资源类型	说明
AuthorizationPolicy	用于配置授权规则。
DestinationRule	定义路由的目标服务和流量策略。VirtualService和DestinationRule是流量控制最关键的两个资源，DestinationRule定义了网格中某个Service对外提供服务的策略及规则，包括负载均衡策略、异常点监测、熔断控制、访问连接池等。
EnvoyFilter	EnvoyFilter为服务网格控制面提供更强大的扩展能力，使Envoy中Filter Chain具备自定义配置的能力。
Gateway	Gateway定义了所有HTTP/TCP流量进入网格或者从网格中出站的统一入口和出口，它描述了一组对外公开的端口、协议、负载均衡以及SNI配置。
PeerAuthentication	Istio的认证策略包含PeerAuthentication和RequestAuthentication，PeerAuthentication策略用于配置服务通信的mTLS模式。
RequestAuthentication	Istio的认证策略包含PeerAuthentication和RequestAuthentication，RequestAuthentication策略用于配置服务的请求身份验证方法。
ServiceEntry	用于注册外部服务到网格内，并对其流量进行管理。
Sidecar	对Sidecar代理进行整体设置。

资源类型	说明
VirtualService	用于网格内路由的设置。VirtualService和DestinationRule是流量控制最关键的两个资源，在VirtualService中定义了一组路由规则，当流量进入时，逐个规则进行匹配，直到匹配成功后将流量转发给指定的路由地址。
WorkloadEntry	用来将虚拟机（VM）或者裸金属（Bare Metal）进行抽象，使其在网格化后作为与Kubernetes中的Pod同等重要的负载，具备流量管理、安全管理、可视化等能力。

7.3.2 YAML 配置资源处理策略

通过控制台和“Istio资源管理”中的YAML方式均可以创建Istio资源，为了避免两个入口的配置相冲突，建议您：

- 控制台创建的资源，在控制台维护
- YAML创建的资源，YAML方式维护

如果控制台创建的资源通过YAML方式修改，将会导致控制台对应功能不可用（例如，YAML配置VirtualService资源后，控制台对应服务的流量治理、灰度发布将不可用）。

处理策略

在“网格配置 > Istio资源管理”页面编辑或创建服务关联的Istio资源时，页面底部会默认勾选“控制台相关功能不开放使用”，当保持勾选状态并单击“确定”后，配置生效，但该服务的控制台部分功能将不再开放使用，后续将只能通过YAML方式维护。

⚠ 注意

如果不勾选“控制台相关功能不开放使用”，可能会导致修改的配置与部分控制台功能冲突，请谨慎选择。

表 7-2 勾选“控制台相关功能不开放使用”对服务的控制台功能的影响

现象	可能原因
在“服务管理”页面，目标服务的“安全”按钮置灰，不可选择	通过YAML方式修改了AuthorizationPolicy资源
在“服务管理”页面，目标服务的“流量治理”按钮置灰，不可选择	通过YAML方式修改了DestinationRule或VirtualService资源
在“服务管理”页面，目标服务的“发布版本”按钮置灰，不可选择	通过YAML方式修改了DestinationRule或VirtualService资源
在“灰度发布”页面创建灰度发布任务时，目标服务置灰，不可选择	通过YAML方式修改了DestinationRule或VirtualService资源

现象	可能原因
在“网关管理”页面，目标网关的添加路由按钮置灰，不可选择	通过YAML方式修改了Gateway资源
在“网关管理”页面添加路由时，目标服务置灰，不可选择	通过YAML方式修改了DestinationRule或VirtualService资源

7.3.3 IstioOperator 配置资源处理策略

Istio采用Istio Operator安装的场景下，有时需要更新被Istio Operator管理的组件（包括istiod、istio-ingressgateway、istio-egressgateway）的工作负载，例如：升级网格版本、扩容istio-ingressgateway实例数等。更新这些工作负载可在CCE控制台“工作负载”页面修改。

处理策略

为了避免多个入口的配置相冲突，以及确保Istio各工作负载持续稳定运行，ASM 1.8.6及以上版本采取如下策略：

- 定义工作负载的关键运行配置和非关键运行配置

  表 7-3 各资源类型下的关键运行配置

工作负载	资源类型	配置项	配置项描述
istiod istio-ingressgateway istio-egressgateway	Deployment	spec.replicas	实例数
		spec.strategy	升级策略
		spec.template.spec.nodeSelector	调度策略
		spec.template.spec.affinity	调度策略
		spec.template.spec.tolerations	调度策略
		spec.template.spec.containers.resources	资源请求和限制

- Istio Operator默认保持当前集群中工作负载的关键运行配置不做更新，仅支持非关键运行配置更新。
- 若需要对关键运行配置进行修改，建议用户通过CCE控制台“工作负载”页面修改，若用户有特定需求，可通过工单进行咨询。

7.4 升级

7.4.1 升级网格

操作场景

用户可以将低版本的网格升级到高版本，以获取更优质的体验。基础版网格支持金丝雀升级，企业版本支持补丁原地升级。

升级影响

- 网格升级将自动重新注入新版本数据面代理，过程中会滚动重启服务Pod，可能造成短暂服务实例中断。
- 升级期间请勿进行灰度发布、流量规则配置等操作。

升级路径

网格类型	源版本	目标版本	升级方式
基础版	1.8.4-r1	1.8.6-r2	补丁更新
	1.8.4-r2	1.8.6-r2	补丁更新
	1.8.4-r3	1.8.6-r2	补丁更新
	1.8.4-r4	1.8.6-r2	补丁更新
	1.8.4-r5	1.8.6-r2	补丁更新

说明

各大版本特性请参见1.3版本特性、1.6版本特性和1.8版本特性。

操作步骤

步骤1 登录应用服务网格控制台，确认网格是否需要升级版本。判断方法如下：

- 列表上方是否提示可升级版本的网格。

- 网格名称右侧是否存在“可升级”提示。

若存在可升级版本的网格，单击该网格名称，进入网格详情页面。

步骤2 在左侧导航栏选择“网格配置”，单击“升级”页签。

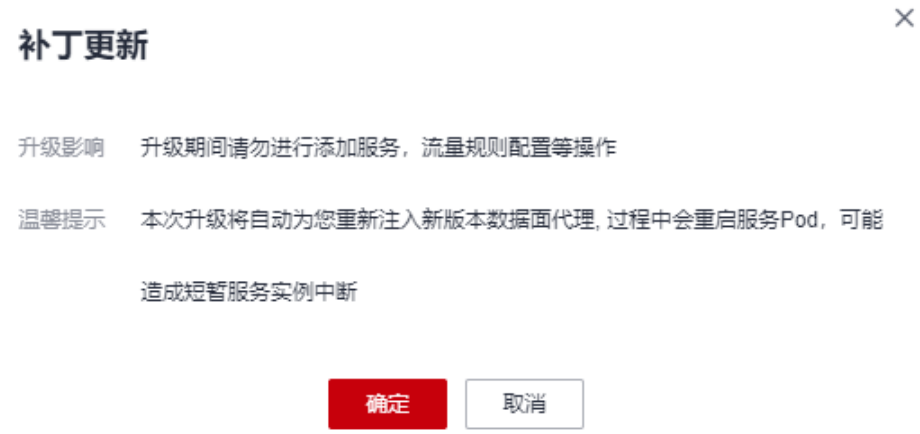
步骤3 根据升级路径选择合适的升级方式完成网格升级。

- **基础版本升级**

单击“版本升级”，系统自动完成升级诊断，检查结果全部成功后，单击“升级”。

- **企业版补丁更新**
  单击“补丁更新”，在弹出的提示框中单击“确定”。

  图 7-9 补丁更新



----结束

7.4.2 1.3 版本特性

- 基于Mixer的Metric、访问日志和调用链
- 支持SDS，证书轮换无需重启Pod
- 支持Sidecar API
- 控制面性能优化
- 华为私有版本Virtual Service Delegation上线，提前为大客户提供解耦Gateway流量配置
- 支持v1.13、v1.15、v1.17和v1.19 CCE集群版本

详细内容请参阅：**https://istio.io/latest/news/releases/1.3.x/announcing-1.3/change-notes/**

7.4.3 1.6 版本特性

- 控制面组件合一，简化控制面安装和运维
- 社区正式版本Virtual Service Delegation更新（华为贡献特性，API和华为1.3先发版本完全相同）
- Workload Entry方便对非Kubernetes负载进行定义和管理
- SDS默认启用
- 支持基于数据面，通过Telemetry V2的非Mixer方式的监控数据采集
- 支持多端口服务基于端口粒度的服务治理
- 支持v1.15和v1.17 CCE集群版本

详细内容请参阅：**https://istio.io/latest/news/releases/1.6.x/announcing-1.6/change-notes/**

7.4.4 1.8 版本特性

- Mixer组件正式下线，访问日志、调用链和监控均基于数据面采集
- EnvoyFilter增强，支持更多灵活的Insert操作
- 启用基于AuthorizationPolicy的新的授权策略
- 支持v1.15、v1.17、v1.19和v1.21 CCE集群版本

详细内容请参阅：**https://istio.io/latest/news/releases/1.8.x/announcing-1.8/change-notes/**

7.4.5 1.13 版本特性

- 支持Istio 1.13.9版本
- 支持v1.21、v1.23 CCE Turbo集群版本
- 支持v1.21、v1.23 CCE集群版本
- 支持通过ProxyConfig API配置Istio sidecar proxy
- 从Istio 1.10版本开始，Sidecar不再重定向流量到 loopback interface（lo），而是将流量重定向到应用的eth0。若您的业务pod监听到了lo，需要改成监听到eth0或者全零监听，否则升级后将导致服务无法访问。**了解更多**

详细内容请参阅：**https://istio.io/latest/news/releases/1.13.x/announcing-1.13/change-notes/**

7.4.6 1.15 版本特性

- 支持Istio 1.15.7版本
- 支持v1.21、v1.23、v1.25、v1.27 CCE Turbo集群版本
- 支持v1.21、v1.23、v1.25、v1.27 CCE集群版本
- 修复了 CVE-2023-44487、CVE-2023-39325、CVE-2023-27487 等安全漏洞

详细内容请参阅：**https://istio.io/latest/news/releases/1.15.x/announcing-1.15.7/**

7.4.7 1.18 版本特性

- 支持Istio 1.18.5版本
- 支持v1.25、v1.27 、v1.28 CCE Turbo集群版本
- 支持v1.25、v1.27、v1.28 CCE集群版本
- 支持 Kubernetes Gateway API

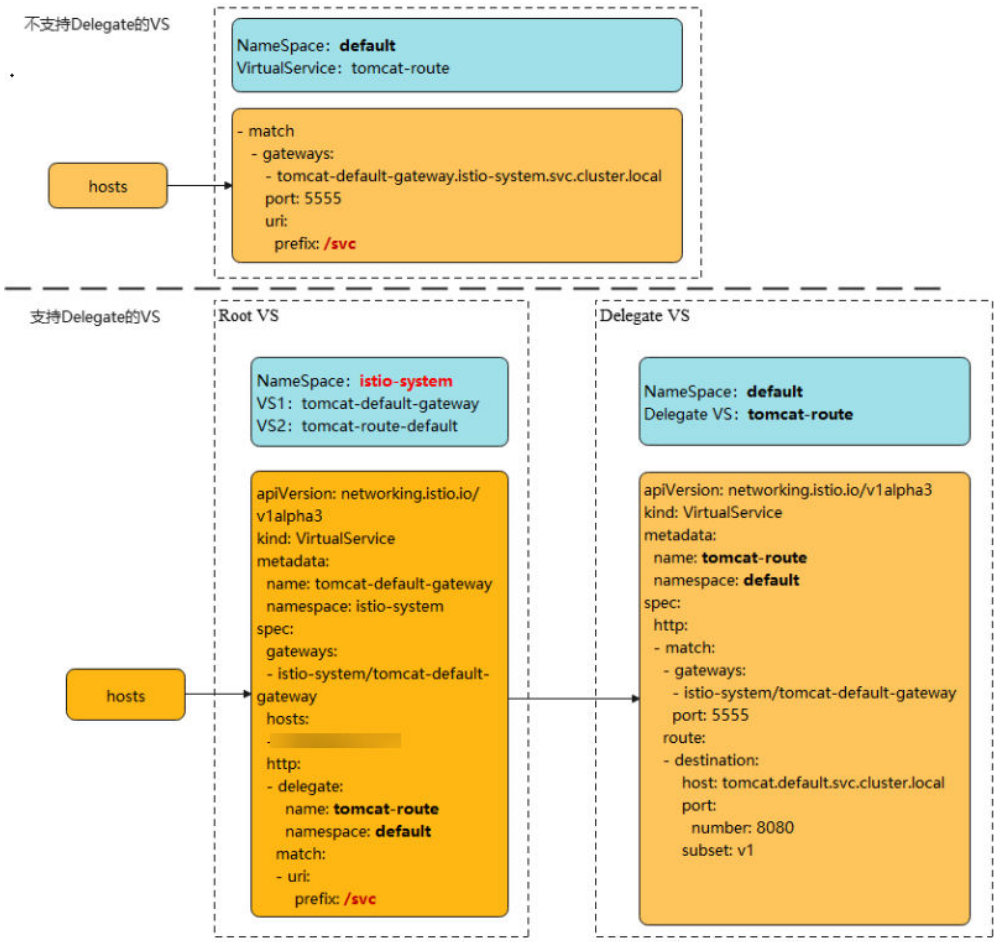
详细内容请参阅：**https://istio.io/latest/news/releases/1.18.x/**

7.4.8 1.3 升级 1.8 VirtualService 支持 Delegate 切换

操作场景

1.8版本的网格默认支持VirtualService的Delegate功能，同时ASM控制台界面也仅支持delegate格式的VirtualService，升级版本并不会对用户的VirtualService进行修改，在

升级后用户将无法在页面对路由进行维护，因此用户需要根据本文指导对应用VirtualService进行修改。



说明

对于Delegate的介绍可以参考istio社区的说明：

https://istio.io/latest/docs/reference/config/networking/virtual-service/#Delegate

约束与限制

- 只有在route和redirect为空时才能设置Delegate。
- ASM只支持一级Delegate，多级Delegate不会生效。
- Delegate VirtualService的HTTPMatchRequest必须是root VirtualService的子集，否则会产生冲突。
- Delegate特性只对HTTP/gRPC协议有效，其他协议无需修改。

操作步骤

修改将分两种情况，下面以加入网格的Tomcat服务为例。

一、若升级前服务未添加网关，则升级后无需修改。

二、若升级前服务添加了网关，则升级后进行如下修改：

1. 为网格所在集群配置kubectl命令，参考CCE控制台集群详情页的指导进行配置。
2. 在istio-system命名空间下创建两个VirtualService YAML文件。

   文件名：**tomcat**-default-gateway.yaml

   其中，

   - tomcat：为修改的服务名
   - tomcat-default-gateway：为该VirtualService名，格式为{服务名}-default-gateway
   - tomcat-route：为修改VirtualService的名字
   - 100.85.219.117：为ELB的IP

   ```
   apiVersion: networking.istio.io/v1beta1
   kind: VirtualService
   metadata:
     name: tomcat-default-gateway
     namespace: istio-system
   spec:
     gateways:
     - istio-system/tomcat-default-gateway
     hosts:
     - 100.85.219.117
     http:
     - delegate:
         name: tomcat-route
         namespace: default
       match:
       - uri:
           prefix: /test
   ```

   文件名：**tomcat**-route-default.yaml

   其中，

   - tomcat：为修改的服务名
   - tomcat-route-default：为该VirtualService名，格式为{服务名}-route-default
   - tomcat-route：为修改VirtualService的名字

   ```
   apiVersion: networking.istio.io/v1beta1
   kind: VirtualService
   metadata:
     name: tomcat-route-default
     namespace: istio-system
   spec:
     hosts:
     - tomcat.default.svc.cluster.local
     http:
     - delegate:
         name: tomcat-route
         namespace: default
       match:
       - uri:
           prefix: /
   ```

   使用如下命令创建VirtualService。

   kubectl create -f tomcat-route-default.yaml

   kubectl create -f tomcat-default-gateway.yaml

3. **kubectl -n{namespace} get vs**获取到服务的VirtualService，执行**kubectl -n{namespace} edit vs tomcat-route**修改如下：

   a. 删除spec.gateways、spec.hosts和spec.http.match.uri

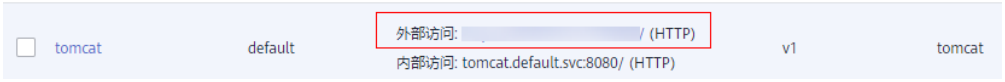
   b. tomcat-default-gateway.istio-system.svc.cluster.local替换成istio-system/tomcat-default-gateway

c. 修改apiVersion: networking.istio.io/v1alpha3为apiVersion: networking.istio.io/v1beta1

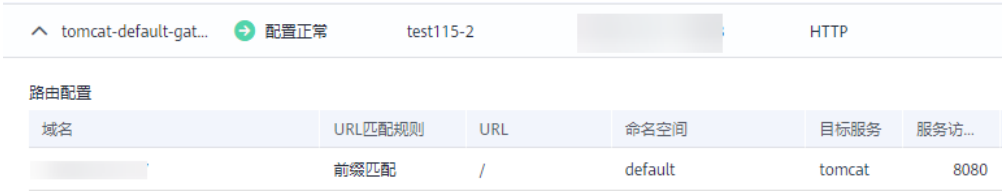
d. destination.host:格式为{服务名}.{namespace}.svc.cluster.local

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: tomcat-route
  namespace: default
spec:
  gateways:
  - tomcat-default-gateway.istio-system.svc.cluster.local
  - mesh
  hosts:
  - tomcat
  - 100.85.219.117  # spec.gateways、spec.hosts需要删除
  http:
  - match:
    - gateways:
      - istio-system/tomcat-default-gateway
      port: 5555
      uri:
        prefix: /test  # spec.http.match.uri需要删除
    route:
    - destination:
        host: tomcat.default.svc.cluster.local
        port:
          number: 8080
        subset: v1
  - match:
    - gateways:
      - mesh
      port: 8080
    route:
    - destination:
        host: tomcat.default.svc.cluster.local
        port:
          number: 8080
        subset: v1
```

4. 升级完成后在服务列表页面，单击外部访问URL，检查访问是否正常。



5. 在服务网关页面，检查服务网关路由是否显示正常。



7.5 网格扩展

可观测性配置展示了当前应用服务网格的应用指标、访问日志、调用链配置；并可启用应用指标和访问日志功能。

说明

调用链只能在创建网格的时候设置启用，网格扩展页面不支持设置启用。

约束与限制

目前访问日志仅Istio 1.18及以上版本支持对接到云日志服务（LTS）。若要启用访问日志，请提前在CCE集群插件中心中安装云原生日志采集插件（CCE Log-Agent）。

启用应用指标

步骤1 登录应用服务网格ASM控制台。

步骤2 单击应用服务网格名称，进入应用服务网格详情页。

步骤3 单击左侧导航栏“网格配置”，选择“网格扩展”页签。

步骤4 单击应用指标 的“立即开启”按钮，选择AOM实例，单击“确定”即可。

----结束

启用访问日志

步骤1 登录应用服务网格ASM控制台。

步骤2 单击应用服务网格名称，进入应用服务网格详情页。

步骤3 单击左侧导航栏“网格配置”，选择“网格扩展”页签。

步骤4 单击访问日志 的“立即开启”按钮，选择需要使用的日志组，日志流，单击“确定”即可。

----结束

8 流量治理

8.1 流量治理概述

流量治理是Istio的核心功能，其目标是提供非侵入的流量治理能力，用户仅仅关注自己的业务逻辑，无需关注服务访问管理。流量治理要解决的问题类似如下：

- 动态修改服务间访问的负载均衡策略，比如配置一致性哈希将流量转发到特定的服务实例上；
- 同一个服务有两个版本在线，将一部分流量切到某个版本上；
- 服务保护，如限制并发连接数、限制请求数、隔离有故障的服务实例等；
- 动态修改服务中的内容，或者模拟一个服务运行故障等。

应用服务网格ASM当前支持重试、超时、连接池、熔断、负载均衡、HTTP头域、故障注入等流量治理能力，可满足大多数业务场景的治理需求。

表 8-1 常用的网格功能和其执行位置

网格功能	治理位置	
	服务发起方	服务提供方
路由管理	Y	N
负载均衡	Y	N
调用链分析	Y	Y
服务认证	Y	Y
可观测性数据	Y	Y
重试	Y	N
重写	Y	N
重定向	Y	N
授权	N	Y

故障注入	Y	N
超时	Y	N
连接池	Y	N
熔断	Y	N
HTTP头域	Y	N

约束与限制

配置诊断失败的服务不能进行流量治理，请先参考**手动修复项**或**自动修复项**修复异常。

8.2 配置流量策略

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“服务管理”，在列表右上方选择服务所在命名空间。

步骤3 选择一个服务，单击操作列的“流量治理”，在右侧页面进行重试、超时、连接池、熔断、负载均衡、HTTP头域、故障注入策略的配置。

重试

服务访问失败自动重试，提高总体访问成功率和质量。

选择“重试”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-2 重试参数说明

参数名称	参数说明	取值范围
重试次数	单个请求允许的重试次数，间隔默认为25ms，实际的重试次数也取决于配置的超时时间和重试超时时间	1-2147483647
重试超时时间（s）	单个请求的超时时间，包括初次请求和重试请求，默认值与配置的超时时间相同	0.001-2592000
重试条件	配置重试的条件，详情请参照**retry policies**和**gRPC retry policies**。	-

超时

服务访问超时自动处理，快速失败，避免资源锁定和请求卡顿。

选择“超时”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-3 超时参数说明

参数名称	参数说明	取值范围
超时时间（s）	HTTP请求超时时间	0.001-2592000

连接池

目的是断开超过阈值的连接和请求，保护目标服务。连接池设置应用于上游服务的每个实例，可以应用在TCP级别和HTTP级别。更多信息请参照**Circuit breaker**。

选择“连接池”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-4 TCP 设置参数说明

参数名称	参数说明	取值范围
最大连接数	到目标服务的HTTP/TCP连接的最大数量，默认2^{32}-1	1-2147483647
最大无响应次数	在确定连接已失效前，要发送的保活探测的最大数量。默认使用操作系统级别配置（除非被覆盖，Linux默认为9）	1-2147483647
健康检查间隔（s）	保活探测之间的持续时间。默认使用操作系统级别配置（除非被覆盖，Linux默认为75秒）	0.001-2592000
连接超时时间（s）	TCP连接超时时间，默认值为10秒	0.001-2592000
最短空闲时间（s）	在开始发送保活探测前，连接需要处于空闲状态的持续时间。默认是使用操作系统级别配置（除非被覆盖，Linux默认为7200秒，即2小时）	0.001-2592000

表 8-5 HTTP 设置参数说明

参数名称	参数说明	取值范围
最大请求数	转发到单个服务实例的最大请求数，默认2^{32}-1	1-2147483647
最大等待请求数	转发到目标服务的最大待处理的HTTP请求数，默认2^{32}-1	1-2147483647

参数名称	参数说明	取值范围
连接最大空闲时间（s）	上游服务连接的空闲超时。如果一定时间内没有活跃的请求，达到空闲时间后，连接将关闭，如果未设置，默认值为1小时	0.001-2592000
最大重试次数	一定时间内，服务所有实例的最大重试次数，默认为2^{32}-1	1-2147483647
每连接最大请求数	每个连接到后端的最大请求数。将此参数设置为1将禁用保活，默认0，表示“无限”，最多2^{29}	1-536870912

熔断

自动隔离不健康的实例，提高服务整体访问成功率。

熔断器跟踪服务实例的访问状态，通过跟踪一段时间内服务实例的访问请求判定其健康状况，将不健康的实例从连接池中隔离，从而提高服务总体的访问成功率，适用于HTTP和TCP服务。对于HTTP服务，连续返回5xx错误的实例认定为不健康。对于TCP服务，连接超时或者连接失败的实例认定为不健康。详情请参照**Outlier detection**。

选择“熔断”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-6 熔断参数说明

参数名称	参数说明	取值范围
连接错误数	检查周期内连续错误的次数，连续错误次数超过该阈值将会被隔离，此功能默认为5	1-2147483647
基础隔离时间（s）	满足熔断条件的实例被隔离的基础隔离时间。服务实例实际隔离时间等于基础隔离时间 x 隔离次数，必须>=0.001s，默认值为30秒	0.001-2592000
检查周期（s）	实例异常检查的间隔，根据在该时段内异常次数是否超过阈值决定是否触发隔离，必须>=0.001s，默认值为10秒	0.001-2592000
最大隔离实例比例（%）	被隔离的服务实例最大百分比，默认为10%	1-100

负载均衡

为目标服务配置满足业务要求的负载均衡策略，控制选择后端服务实例。

选择“负载均衡”页签，单击“立即配置”，在弹出对话框中，根据实际需求选择负载均衡算法。

- 轮询调度：网格使用的默认负载均衡算法，实例池中的每个实例轮流获得请求。
- 最少连接：随机选取两个健康的实例，再从所选取的两个实例中选择一个连接数较少的实例。
- 随机调度：从所有健康的实例中，随机选取一个。
- 一致性哈希：包含四种类型，如表8-7所述。

表 8-7 一致性哈希算法类型

类型	说明
基于HTTP的Header	需要输入HTTP头域的键名，根据HTTP请求中的请求头名称来计算哈希值，相同的值会转发到同一实例中。
基于Cookie	需要输入Cookie的键名，根据HTTP请求中的cookie的键名来计算哈希值，相同的值会转发到同一实例中。
基于源IP	根据HTTP请求中的源IP地址来计算哈希值，相同的值会转发到同一实例中。
基于query参数	需要输入query参数的键名，根据HTTP请求中的query参数名称来计算哈希值，相同的值会转发到同一实例中。

HTTP头域

灵活增加、修改和删除指定HTTP头域，非侵入方式管理请求内容。

选择“HTTP头域”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-8 将 HTTP 请求转发到目标服务之前，对 Headers 的操作

参数名称	参数说明
添加请求的Headers	添加新的请求Headers参数，需要设置key和value。还可以单击⊕图标，添加更多同类型参数。
修改请求的Headers	修改已有的请求Headers参数，需要设置key和value。还可以单击⊕图标，添加更多同类型参数。
移除请求的Headers	删除已有的请求Headers参数，需要设置key。还可以单击⊕图标，添加更多同类型参数。

表 8-9 将 HTTP 响应回复给客户端前，对 Headers 的操作

参数名称	参数说明
添加响应的Headers	添加新的响应Headers参数，需要设置key和value。还可以单击⊕图标，添加更多同类型参数。

参数名称	参数说明
修改响应的Headers	修改已有的响应Headers参数，需要设置key和value。还可以单击⊕图标，添加更多同类型参数。
移除响应的Headers	删除已有的响应Headers参数，需要设置key。还可以单击⊕图标，添加更多同类型参数。

故障注入

故障注入是一种有效的测试方法，它能够将错误引入系统，以确保系统能够承受错误的并从错误中恢复。

选择“故障注入”页签，单击“立即配置”，在弹出对话框中，根据实际需求配置如下参数：

表 8-10 故障注入参数说明

参数名称	参数说明	取值范围
故障类型	包括延时故障和中断故障。 ● 延时故障：对通往组件的请求有延迟。 ● 中断故障：会中断该组件的服务并返回预设状态码。	延时故障、中断故障
延时（s）	故障类型选择“延时故障”时需要设置。 在转发请求之前添加的固定延迟。	0.001-2592000
HTTP状态码	故障类型选择“中断故障”时需要设置。 终止故障时返回的HTTP状态码，默认返回500。	200-599
故障百分比（%）	注入延时或中断故障的请求百分比。	1-100

----结束

8.3 更改流量策略

操作场景

流量策略设置完成后，支持更改流量策略，如将负载均衡的算法由“轮询调度”转为“随机调度”。

操作步骤

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“服务管理”，选择需要更改流量策略的服务，单击操作列的“流量治理”，在右侧页面进行流量策略更改。

步骤3 在“负载均衡”页签中，单击“立即配置”，在弹出的“负载均衡”页面更改算法为“随机调度”，单击“确定”。

图 8-1 修改负载均衡算法

----结束

9 安全

9.1 配置安全策略

ASM提供的安全功能主要分为访问授权、对端认证和JWT认证，以确保服务间通信的可靠性。

操作步骤

步骤1 登录应用服务网格控制台，单击服务网格的名称，进入网格详情页面。

步骤2 在左侧导航栏选择“服务管理”，在列表右上方选择服务所在命名空间。

步骤3 选择一个服务，单击操作列的“安全”，在右侧页面进行访问授权和对端认证配置。

访问授权

用来实现对网格中服务的访问控制功能，即判断一个请求是否允许发送到当前的服务。

选择“访问授权”，单击“立即配置”，在弹出对话框中，单击 

，选择指定命名空间下的一个或多个服务进行访问授权设置。

对端认证

Istio通过客户端和服务端的PEP（Policy Enforcement Points）隧道实现服务实例之间的通信，对端认证定义了流量如何通过隧道（或者不通过隧道）传输到当前服务的实例。已经注入sidecar的服务实例之间，默认通过隧道进行通信，流量会自动进行TLS加密。

选择“对端认证”页签，单击“立即配置”，在弹出对话框中选择认证策略类型。

表 9-1 参数说明

参数名称	参数说明
默认模式（UNSET）	如果父作用域配置了对端认证策略，服务将继承父作用域的配置。

参数名称	参数说明
宽容模式（PERMISSIVE）	请求可以不通过隧道进行传输，既可以是明文，也可以是TLS加密的密文。默认情况下，网格配置了宽容模式（PERMISSIVE）的对端认证策略。
严格模式（STRICT）	流量只能通过隧道进行传输，因为请求必须是TLS加密的密文，且必须带有客户端证书。

JWT认证

在服务网格中配置JWT（JSON Web Token）请求授权，可以实现来源认证。在接收用户请求时，该配置用于认证请求头信息中的Access Token是否可信，并授权给来源合法的请求。

说明

仅支持为HTTP协议的服务配置JWT认证。

选择“JWT认证”页签，单击“立即配置”，在弹出对话框中配置如下参数：

- 发行者：JWT的颁发者。
- 令牌受众：设置哪些服务可以使用JWT Token访问目标服务，多个受众用“,”隔开，若为空表示对访问的服务不受限制。
- jwks：JWT规则集。

JWT认证的原理及应用示例请参见**JWT认证原理**和**在ASM中对入口网关进行JWT请求认证**。

----结束

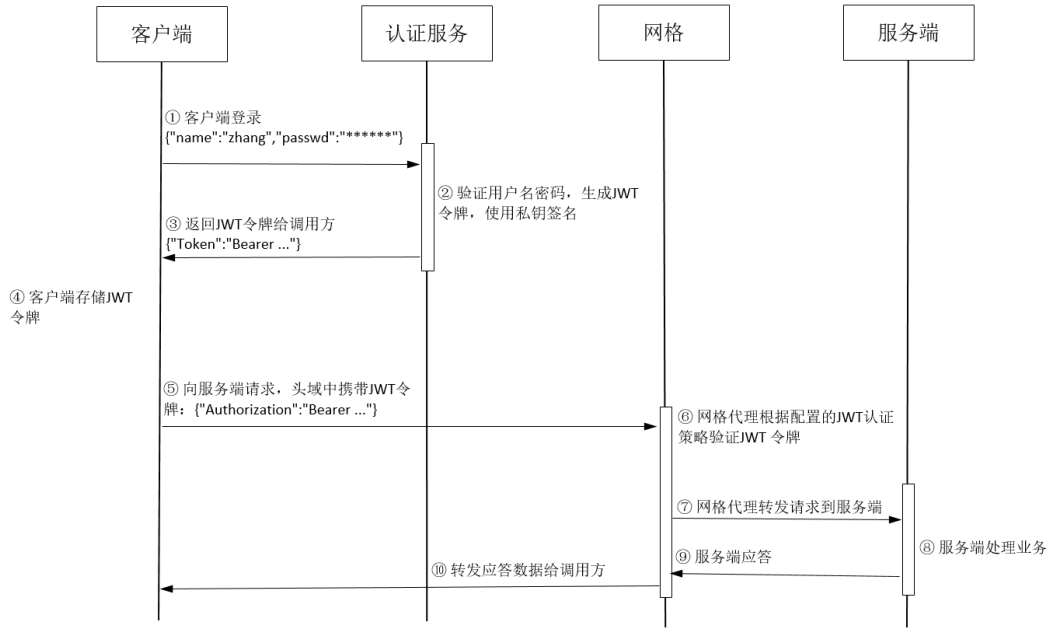
9.2 JWT 认证原理

JWT 认证原理

JWT（Json Web Token）是一种服务端向客户端发放令牌的认证方式。客户端用户名密码登录时，服务端会生成一个令牌返回给客户端；客户端随后在向服务端请求时只需携带这个令牌，服务端通过校验令牌来验证是否是来自合法的客户端，进而决定是否向客户端返回应答。从机制可以看到，这种基于请求中携带令牌来维护认证的客户端连接的方式解决了早期服务端存储会话的各种有状态问题。

在Istio使用中，JWT令牌生成由特定的认证服务提供，令牌验证由网格执行，彻底解耦用户业务中的认证逻辑，使应用程序专注于自身业务。基于Istio的JWT完整机制如**图9-1**所示。

图 9-1 Istio JWT 认证流程



① 客户端连接认证服务，提供用户名和密码；

② 认证服务验证用户名和密码，生成JWT令牌，包括用户标识和过期时间等信息，并使用认证服务的私钥签名；

③ 认证服务向客户端返回生成的JWT令牌；

④ 客户端将收到的JWT令牌存储在本端，供后续请求时使用；

⑤ 客户端在向其他服务发起请求时携带JWT令牌，无需再提供用户名、密码等信息；

⑥ 网格数据面代理拦截到流量，使用配置的公钥验证JWT令牌；

⑦ 验证通过后，网格代理将请求转发给服务端；

⑧ 服务端处理请求；

⑨ 服务端返回应答数据给网格代理；

⑩ 网格数据面代理转发应答数据给调用方。

在这个过程中，重点是第六步，原来服务端的JWT认证功能卸载到了网格代理上。网格数据面从控制面配置的认证策略中获取验证JWT令牌的公钥，可以是jwks（JSON Web Key Set）上配置的公钥，也可以是从jwksUri配置的公钥地址获取到的公钥。获得公钥后，网格代理使用该公钥对认证服务私钥签名的令牌进行验证，并解开令牌中的iss，验证是否匹配认证策略中的签发者信息。验证通过的请求发送给应用程序，验证不通过则直接拒绝，不会发送给应用程序。

JWT 结构

JWT是一个包含了特定声明的Json结构。从前面介绍的JWT认证流程第六步知道，只要验证这个Json结构本身，即可以确认请求身份，不需要查询后端服务。下面解析JWT结构从而了解如何携带这些认证信息。

JWT包含三部分：头部Header、负载Payload和签名Signature。

- 头部Header

  头部描述JWT的元数据，包括算法alg和类别typ等信息。alg描述签名算法，这样接收者可以根据对应的算法来验证签名，默认是如下所示的HS256，表示 HMAC-SHA256；typ表示令牌类型，设置为JWT，表示这是一个JWT类型的令牌。

  ```
  {
    "alg": "HS256",
    "typ": "JWT"
  }
  ```

- 负载Payload

  存放令牌的主体内容，由认证服务AuthN生成相关信息并放到令牌的负载中。重要属性包括：

  - iss：令牌发行者 issuer
  - aud：令牌受众 audience

  在JWT验证时，会校验发行者、受众信息和令牌负载中的发行者iss、受众audience是否匹配。JWT的内容本身不是加密的，所有拿到令牌的服务都可以看到令牌负载Payload中的内容，因此建议Payload里不要存放私密的信息。

- 签名Signature

  签名字段是对头部和负载的签名，确保只有特定合法的认证服务才可以发行令牌。实际使用中一般是把头部和负载分别执行Base64转换成字符串，然后使用认证服务的密钥对拼接的字符串进行签名，签名算法正是前面介绍的头域中定义的算法。

一个完整的JWT示例如下，对于头部Header和负载Payload进行签名得到Signature。

```
# Header:
{
  "alg": "RS512",
  "typ": "JWT"
}

# Payload
{
  "iss": "weather@cloudnative-istio",
  "audience": "weather@cloudnative-istio"
}

# Signature
RSASHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload)
  )
```

以上结构最终输出的令牌如下，可以看到“.”分割的三个字符串分别对应JWT结构的头部、负载和签名三部分。

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjQ2ODU5ODk3MDAsInZlciI6IjIuMCIsImlhdCI6MTUzMjM4OTcwMCwiaXNzIjoid2VhdGhlckBjbG91ZG5hdGl2ZS1pc3Rpby5ib29rIiwic3ViIjoid2VhdGhlckBjbG91ZG5hdGl2ZS1pc3Rpby5ib29rIn0.SEp-8qiMwI45BuBgQPH-wTHvOYxcE_jPI0wqOxEpauw
```

9.3 在 ASM 中对入口网关进行 JWT 请求认证

本文介绍如何在ASM中对入口网关进行JWT请求认证，确保用户在通过入口网关访问服务时，必须带有可信赖的Access Token。

准备工作

1. 已创建一个1.8.6或1.13版本网格。

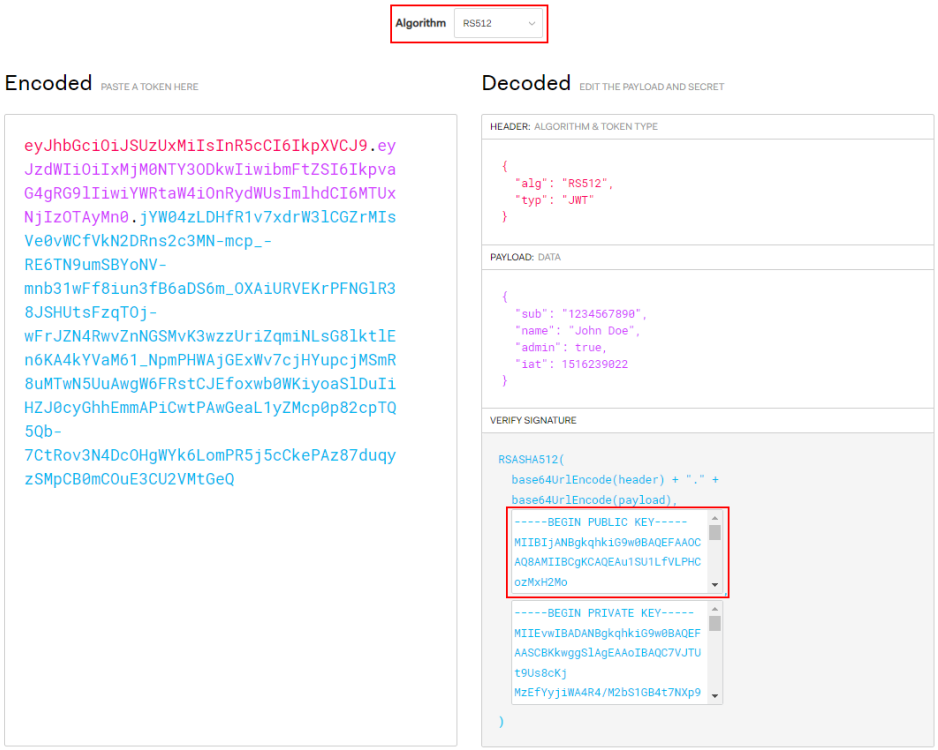
2. 网格中有诊断通过的httpbin服务，镜像为httpbin，端口协议为http，端口号为80。
3. 网格中已为httpbin服务创建可访问的网关。

创建 JWT 认证

步骤1 创建JWK。

1. 访问**JWT工具网站**，选择“Algorithm”为“RS512”，获取公钥（PUBLIC KEY）。

图 9-2 生成公钥



2. 在**JWK to PEM Convertor online**工具中选中“PEM-to-JWK (RSA Only)”，输入上一步获取的公钥，单击“submit”，将公钥转换为JWK。

图 9-3 将公钥转换为 JWK

{"kty":"RSA","e":"AQAB","kid":"a78641b9-d81e-4241-
b35a-71726c3fa053","n":"u1SU1LfVLPHCozMxH2Mo4lgOEePzNm0tRgeLezV6ffAt0gunVTLw7onLRnrq0
_IzW7yWR7QkrmBL7jTKEn5u-qKhbwKfBstIs-
bMY2Zkp18gnTxKLxoS2tFczGkPLPgizskuemMghRniWaoLcyehkd3qqGElvW_VDL5AaWTg0nLVkjRo9z-40
RQzuVaE8AkAFmxZzow3x-
VJYKdjykkJ0iT9wCS0DRTXu269V264Vf_3jvredZiKRkgwlL9xNAwxXFg0x_XFw005UWVRIkdgcKWTjpBP2d
PwVZ4WWC-9aGVd-Gyn1o0CLelf4rEjGoXbAAEgAqeGUxrcIlbjXfbcmw"}

步骤2 创建JWT认证。

1. 登录应用服务网格控制台，单击网格名称，进入网格详情页面。
2. 在左侧导航栏选择“服务管理”，在列表右上方选择服务所在命名空间。
3. 选择httpbin服务，单击操作列的“安全”，在右侧页面选择“JWT认证”页签，单击“立即配置”，在弹出的“JWT认证”页面填写如下信息：
   - 发行者：JWT的颁发者，本文设置为“test”。
   - 令牌受众：JWT受众列表，设置哪些服务可以使用JWT Token访问目标服务，本文设置为“ASM”。
   - jwks：设置JWT信息，本文设置为{"keys": [**步骤1**创建的JWK]}，例如**步骤1**创建的JWK为{"kty":"RSA","e":"AQAB","kid":"a78641b9-d81e-4241-b35a-71726c3****"}，则jwks为{"keys": [{"kty":"RSA","e":"AQAB","kid":"a78641b9-d81e-4241-b35a-71726c3****"}]}。

图 9-4 创建 JWT 认证

4. 单击“确定”完成创建。

----结束

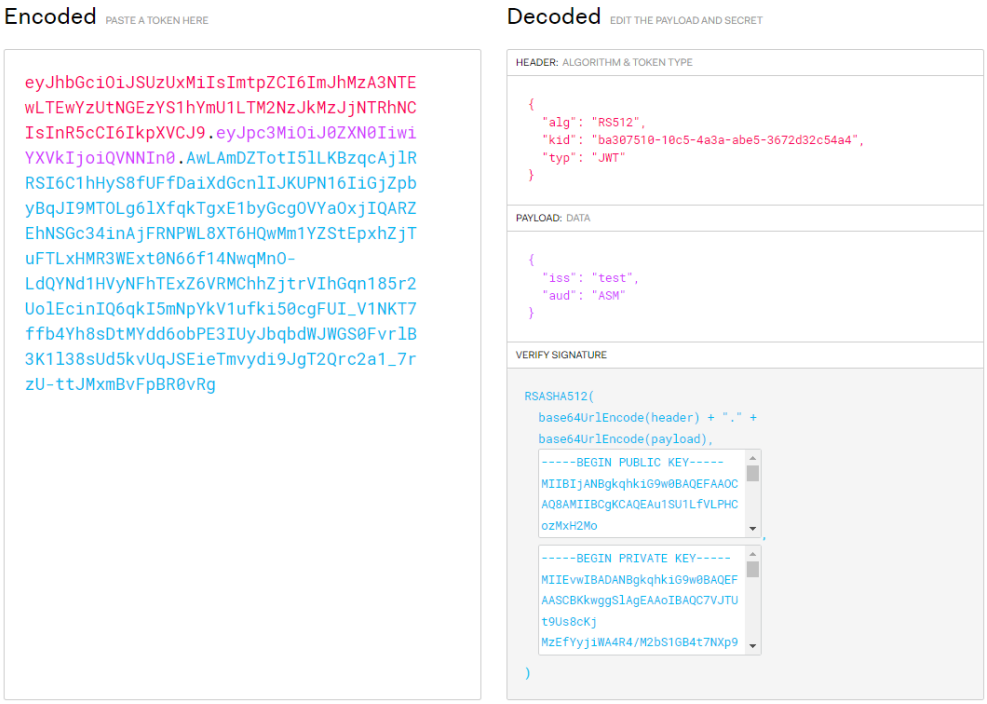
验证 JWT 认证是否生效

步骤1 使用**JWT工具**将JWT请求信息编码成JWT Token。

在“Decoded”区域输入以下JWT请求信息，在“Encode”区域将看到自动转换后的JWT Token。

- HEADER：设置alg为“RS512”，输入**步骤1**创建的JWK中的kid，设置type为“JWT”。
- PAYLOAD：设置iss为“test”，aud为“ASM”，确保与**步骤2**中配置的发行者、令牌受众保持一致。
- VERIFY SIGNATURE：与**步骤1.1**中的公钥保持一致。

图 9-5 创建 JWT Token



步骤2 通过入口网关访问httpbin服务。

1. 执行以下命令，带**步骤1**创建的JWT Token访问服务。

   TOKEN=*步骤1创建的JWT Token*

   curl -I -H "Authorization: Bearer $TOKEN" http://*{httpbin服务的外部访问地址}*/

   预期输出：

   ```
   HTTP/1.1 200 OK
   server: istio-envoy
   date: Wed, 21 Sep 2022 03:11:48 GMT
   ```

2. 执行以下命令，带无效的JWT Token访问服务。

   curl -I -H "Authorization: Bearer invalidToken" http://*{httpbin服务的外部访问地址}*/

   预期输出：

   ```
   HTTP/1.1 401 Unauthorized
   www-authenticate: Bearer realm="http://***.***.***.***:***/", error="invalid_token"
   content-length: 145
   content-type: text/plain
   date: Wed, 21 Sep 2022 03:12:54 GMT
   server: istio-envoy
   x-envoy-upstream-service-time: 19
   ```

3. 修改**步骤2**中创建的JWT认证，将令牌受众置空（表示对访问的服务不受限制），然后执行以下命令，带**步骤1**创建的JWT Token访问服务。

   curl -I -H "Authorization: Bearer $TOKEN" http://*{httpbin服务的外部访问地址}*/

   预期输出：

   ```
   HTTP/1.1 200 OK
   server: istio-envoy
   date: Wed, 21 Sep 2022 03:20:07 GMT
   ```


4. 执行以下命令，不带JWT Token访问服务。

   curl -I http://*{httpbin服务的外部访问地址}/*

   预期输出：

   ```
   HTTP/1.1 403 Forbidden
   content-length: 85
   content-type: text/plain
   date: Wed, 21 Sep 2022 03:29:31 GMT
   server: istio-envoy
   x-envoy-upstream-service-time: 6
   ```

根据以上结果，可以看到带有正确的JWT Token的请求访问服务成功，带有错误的JWT Token或者不带JWT Token的请求访问服务失败，说明请求身份认证生效。

----结束

