应用服务网格

用户使用指南

天翼云科技有限公司

目 录

1 产品概述

1.1 产品定义

什么是应用服务网格

应用服务网格 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，功能包括负载均衡、熔断、限流等多种治理能力。并内置金丝雀、蓝绿灰度发布流程，提供一站式自动化的发布管理。

什么是 Istio

Istio 是一个提供连接、保护、控制以及观测功能的开放平台，通过提供完整的非侵入式的微服务治理解决方案，能够很好的解决云原生服务的网络连接以及安全管理等服务治理问题。

随着微服务的大量应用，其构成的分布式应用架构在运维、调试和安全管理等维度变得更加复杂，开发者需要面临更大的挑战，如：服务发现、负载均衡、故障恢复、指标收集和监控，以及金丝雀发布、蓝绿发布、限流、访问控制、端到端认证等。

在较高的层面上，Istio 有助于降低这些部署的复杂性，并减轻开发团队的压力。它是一个完全开源的服务网格，可以透明地分层到现有的分布式应用程序上。它也是一个平台，包括允许集成到任何日志记录平台、遥测或策略系统的 API。Istio 的多样化功能使您能够成功高效地运行分布式微服务架构，并提供保护、连接和监控微服务的统一方法。

服务网格

服务网格（Service Mesh）通常用于描述构成应用程序的微服务网络以及应用之间的交互。它的需求包括服务发现、负载均衡、故障恢复、指标收集和监控以及更加复杂的运维需求，例如蓝绿发布、金丝雀发布、限流、访问控制和端到端认证等。

为什么要使用 Istio

Istio 提供了一个完整的解决方案，通过为整个服务网格提供行为洞察和操作控制来满足微服务应用程序的多样化需求。

Kubernetes 提供了部署、升级和有限的运行流量管理能力，但并不具备熔断、限流降级等能力。Istio 是基于 Kubernetes 构建的开放平台，它很好的补齐了 Kubernetes 在微服务治理上的诸多能力。

想要让服务支持 Istio，只需要在您的环境中部署一个特殊的 Sidecar 代理，使用 Istio 控制平面功能配置和管理代理，拦截微服务之间的所有网络通信：

- 实现 HTTP、gRPC、WebSocket 和 TCP 流量的自动负载均衡。
- 通过丰富的路由规则、重试、故障转移和故障注入，可以对流量行为进行细粒度控制。
- 对出入集群入口和出口中所有流量自动度量指标、日志记录和追踪。
- 通过强大的基于身份的验证和授权，在集群中实现安全的服务间通信。

Istio 旨在实现可扩展性，满足多种部署需求。

1.2 产品功能

灰度发布

- 基于请求内容灰度规则：支持基于请求内容灰度规则，可以配置 Header、Cookie 等多种请求信息。
- 基于流量比例灰度规则：支持基于流量比例灰度规则，根据权重比例分配流量。
- 金丝雀灰度流程：提供向导方式引导用户完成金丝雀灰度流程，包括灰度版本上线、观察灰度版本运行、配置灰度规则、观测访问情况、切分流量等。
- 蓝绿灰度流程：提供向导方式引导用户完成蓝绿灰度流程，包括灰度版本上线、观察灰度版本运行、观测访问情况、版本切换等。

流量治理

- 七层连接池管理：支持界面基于拓扑配置，配置最大等待 HTTP 请求数、最大请求数、每个连接的最大请求数、最大重试次数。
- 四层连接池管理：支持界面基于拓扑配置，配置 TCP 的最大连接数、连接超时等。
- 熔断：支持界面基于拓扑配置服务熔断规则，包括实例被驱逐前的连续错误次数、驱逐间隔时长、最小驱逐时间、最大驱逐比例等。
- 重试：支持配置 HTTP 重试次数等进行 HTTP 重试。
- 重定向：支持配置 HTTP 重定向到一个指定的目标地址（后台配置）。
- 重写：支持配置 HTTP 重写一个目标的地址（后台配置）。
- 流量镜像：支持将流量实时镜像到另外一个目标地址上（后台配置）。
- 请求超时：支持配置 HTTP 超时时间（后台配置）。
- 降级：不支持传统微服务的降级语义。
- 负载均衡：支持界面基于拓扑配置随机、轮询、最小连接数等多种负载均衡策略。
- 会话保持：支持界面配置会话保持规则。
- 故障注入：支持配置错误和延时的故障。

安全

- 透明双向认证：支持界面基于拓扑配置服务间的双向认证。
- 细粒度访问授权：支持界面基于拓扑配置服务间的访问授权（后台 API 可以配置 Namespace 级别授权，授权可以给一个特定的接口）。

可观察性

- 应用访问拓扑：支持网格应用访问拓扑，体现服务间依赖。
- 服务运行监控：支持服务访问信息，包括服务和服务各个版本的 QPS 和延时等指标。
- 访问日志：支持收集和检索服务的访问日志。

网格数据面服务框架

- Spring Cloud：支持 Spring Cloud SDK 开发的服务在网格上统一管理。
- Dubbo：支持 Dubbo SDK 开发的服务在网格上统一管理。

兼容性和扩展

- 社区版本兼容：API 完全兼容 Istio。
- 社区插件支持：支持 Tracing、Prometheus、Kiali、Grafana。

1.3 产品优势

简单易用

无需修改任何服务代码，也无需手动安装代理，只需开启应用服务网格功能，即可实现丰富的无侵入服务治理能力。

内置金丝雀、蓝绿灰度发布流程

- 灰度版本一键部署，流量切换一键生效。
- 灰度策略可配置，支持流量比例、请求内容（Cookie、OS、浏览器等）、源 IP 等。
- 一站式健康、性能、流量监控，实现灰度发布过程量化、智能化、可视化。

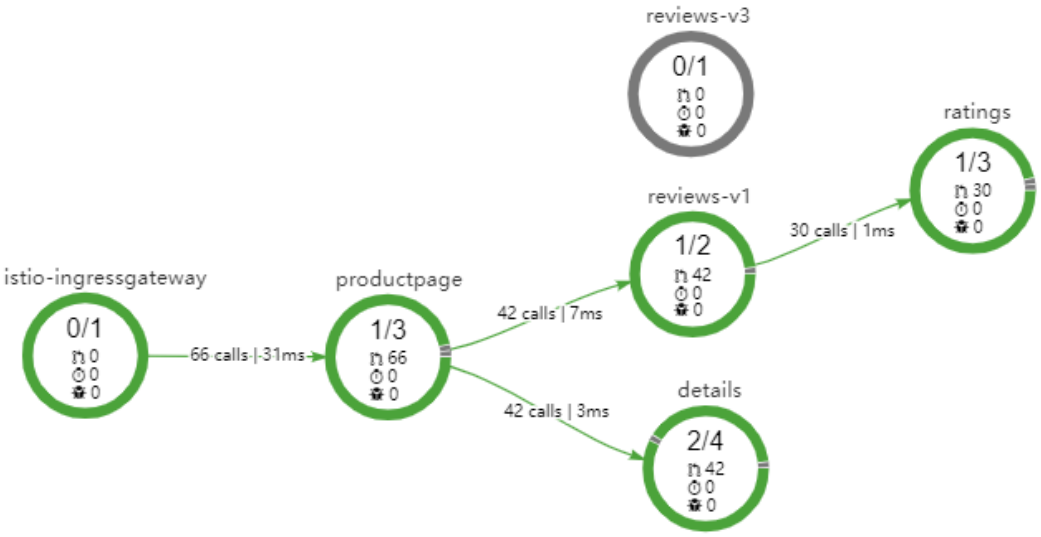
策略化的智能路由与弹性流量管理

支持对服务配置负载均衡、服务路由、故障注入、熔断容错等治理规则，并结合一站式治理系统，提供实时的、可视化的微服务流量管理；无侵入智能流量治理，应用无需任何改造，即可进行动态的智能路由和弹性流量管理。

- 权重、内容、TCP/IP 等路由规则，实现应用灵活灰度发布。
- HTTP 会话保持，满足业务处理持续性诉求。
- 限流、熔断，实现服务间链路稳定、可靠。
- 网络长连接管理降低资源损耗，提升网络吞吐量。
- 服务安全认证：认证、鉴权、审计等，提供服务安全保障基石。

图形化应用全景拓扑，流量治理可视化

应用服务网格提供了可视化的流量监控，链路健康状态、异常响应、超长响应时延、流量状态信息拓扑等一目了然。



结合应用运维管理 AOM、应用性能管理 APM 服务，提供了详细的微服务级流量监控、异常响应流量报告以及调用链信息，实现更快速、更准确的问题定位。

性能增强，可靠性增强

控制面和数据面在社区版本基础上进行可靠性加固和性能优化。

多基础设施

提供免运维的托管控制面，提供全局统一的服务治理，灰度、安全和服务运行监控能力，并支持对支持容器和 VM 等多种基础设施的统一服务发现和管理。

协议扩展

社区通用的 HTTP、gRPC、TCP、TLS 外扩展对 Dubbo 协议的支持。

传统 SDK 集成

提供 Spring Cloud、Dubbo 等传统微服务 SDK 的集成解决方案，传统的微服务 SDK 开发的业务代码无需大的代码修改即可迁移到云原生的容器和网格运行环境上来。

1.4 应用场景

1.4.1 服务灰度发布

适用场景

通常产品优化迭代的方式，是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。

灰度发布是版本升级平滑过渡的一种方式，当版本升级时，使部分用户使用高版本，其他用户继续使用低版本，待高版本稳定后，逐步扩大范围把所有用户流量都迁移到高版本上面来。

价值

应用服务网格为应用治理提供多种灰度发布功能，在初始灰度的时候可以发现、调整问题，以保证其影响度和整体系统的稳定，稳定高效地推动企业应用的迭代升级。

优势

- **内置灰度流程：**基于细粒度的分流规则，在 ASM 中内置了多种典型的灰度发布流程，提供一个灰度发布的向导，方便用户便捷的进行灰度发布实践。在一个服务版本正常工作，正常处理流量的同时，用户可以创建一个新的灰度版本。当灰度版本启动成功后，引导用户配置灰度规则来切分流量。
- **灵活的灰度策略：**灰度规则可以是基于权重的按比例切分流量，也可以根据服务访问的内容来将特定内容的请求发往灰度版本，对于常用的 HTTP 协议，如请求中的 OS、浏览器、Cookie 和 Header 信息等，在配置了灰度规则后，可以实时的观察到多个线上版本的运行和访问信息，从而在向导中一键式完成版本选择，将所有流量都切换到最终选定的版本上。
- **自动化灰度：**用户也可以将以上灰度发布流程进行自动化配置，从而实现无人值守的灰度发布。即只要灰度版本运行成功就会自动触发预制的灰度策略给灰度版本分配流量，在流量运行特定的时间后，根据预制规则自动的切换流量完成灰度发布。
- **应用灰度：**除了服务粒度的灰度发布外，用户也可以对多个服务组成的应用，在入口服务处配置灰度规则，对一组服务使用同样的分流策略，进行灰度发布。

1.4.2 服务流量管理

适用场景

流量治理是一个非常宽泛的话题，例如：

- 动态修改服务间访问的负载均衡策略，如根据某个请求特征做会话保持。
- 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。
- 对服务进行保护，例如限制并发连接数、限制请求数、隔离故障服务实例等。

- 动态修改服务中的内容，或者模拟一个服务运行故障等。

价值

在 Istio 中实现这些服务治理功能时无须修改任何应用的代码。

应用服务网格 ASM 基于 Istio 可以为管理的服务提供非侵入的流量治理能力。根据服务的协议，提供策略化、场景化的网络连接管理。在应用拓扑上对选定服务的选定端口，根据需要配置各种不同的治理规则。

优势

- **负载均衡：**配置各种负载均衡策略，如随机、轮询、最大连接数等，还可以配置会话保持将流量发到特定的服务实例上。
- **连接池：**通过连接池管理，可以对四层协议配置 TCP 的最大连接数、连接超时等连接配置，对七层协议配置最大等待 HTTP 请求数、最大请求数、每个连接的最大请求数、最大重试次数等，从而防止一个服务的失败级联影响到整个应用。
- **熔断：**通过熔断配置实例被驱逐前的连续错误次数、驱逐间隔时长、最小驱逐时间、最大驱逐比例等参数，从而定期考察被访问的服务实例的工作情况，如果连续出现访问异常，则将服务实例标记为异常并进行隔离，在一段时间内不为其分配流量。过一段时间后，被隔离的服务实例会再次被解除隔离，尝试处理请求，如果还不正常，则被隔离更长的时间。从而实现异常服务实例的故障隔离和自动故障恢复。
- **故障注入：**通过对选定的服务注入中断的故障、延时故障来构造故障场景，从而无需修改代码即可进行故障测试。

1.4.3 端到端的透明安全

适用场景

众所周知，将传统的单体应用拆分为一个个微服务固然带来了各种好处，包括更好的灵活性、可伸缩性、重用性，但微服务也同样面临着特殊的安全需求，如下所示：

- 为了抵御中间人攻击，需要用到流量加密。
- 为了提供灵活的服务访问控制，需要用到 TLS 和细粒度访问策略。
- 为了决定哪些人在哪些时间可以做哪些事，需要用到审计工具。

面对这些需求应用服务网格提供全面的安全解决方案，包括身份验证策略，透明的 TLS 加密以及授权和审计工具。

价值

- **默认的安全性：**无需修改即可保证应用程序代码和架构的安全性。
- **纵深防御：**与现有的安全系统结合并提供多层防御。
- **零信任网络：**在不受信任的网络上构建安全解决方案。

优势

- **非侵入安全：**应用服务网格是以一种安全基础设施的方式向用户提供透明的安全能力，让不涉及安全问题的代码安全运行，让不太懂安全的人可以开发和运维安全的服务，不用修改业务代码就能提供服务访问安全。应用服务网格提供了一个透明的分布式安全层，并提供了底层安全的通信通道，管理服务通信的认证、授权和加密，提供 Pod 到 Pod、服务到服务的通信安全。开发人员在这个安全基础设施层上只需专注于应用程序级别的安全性。
- **细粒度授权：**在认证的基础上，就可以进行服务间的访问授权管理，可以控制某个服务，或者服务的一个特定接口进行授权管理。如只开放给特定的一个 Namespace 下的服务，或者开放给某个特定的一个服务。源服务和目标服务可以在不同的集群，甚至源服务的不同实例在不同的集群，目标服务的不同实例在不同的集群。

1.4.4 服务运行监控

适用场景

运营容器化的基础设施带来了一系列新的挑战。我们需要增强容器、评估 API 端点的性能以及识别出基础设施中的有害部分。Istio 服务网格可在不修改代码的情况下实现 API 增强，并且不会带来服务延迟。

价值

应用服务网格为网格内的所有服务通信生成详细的遥测，这种遥测技术提供了服务行为的可观察性，允许运营商对其应用程序进行故障排除、维护和优化，而不会给服务开发人员带来任何额外负担。通过应用服务网格，运营商可以全面了解被监控的服务如何与其他服务以及组件本身进行交互。

优势

- **非侵入监控数据采集：**在复杂应用的场景下，服务间的访问拓扑，监控等都是我们对服务整体运行状况进行管理，服务访问异常时进行定位定界的必要手段。服务网格技术的一项重要能力就是以应用非侵入的方式提供这些监控数据的采集，用户只需关注自己的业务开发，无需额外关注监控数据的生成。
- **灵活的服务运行管理：**在拓扑图上通过服务的访问数据，可以直观的观察到服务的健康状况，服务间的依赖情况。并且可以对关心的服务进行下钻，从服务级别下钻到服务版本级别，还可以进一步下钻到服务实例级别。通过实例级别的拓扑可以观察到配置了熔断规则后，网格如何隔离故障实例，使其逐渐接收不到流量。并且可以在故障实例正常时，如何进行实例的故障恢复，自动给恢复的实例重新分配流量。

1.5 约束与限制

集群版本限制

启用应用服务网格前，您需要创建或已有一个可用集群，并确保集群版本为 v1.15、v1.17、v1.19。

1.6 基本概念

工作负载

工作负载即 Kubernetes 对一组 Pod 的抽象模型，用于描述业务的运行载体，包括 Deployment、Statefulset、Job、Deamonset 等。

- 无状态工作负载（即 kubernetes 中的“Deployments”）：Pod 之间完全独立、功能相同，具有弹性伸缩、滚动升级等特性。如：nginx、wordpress。
- 有状态工作负载（即 kubernetes 中的“StatefulSets”）：Pod 之间不完全独立，具有稳定的持久化存储和网络标示，以及有序的部署、收缩和删除等特性。如：mysql-HA、etcd。

实例（Pod）

Pod 是 Kubernetes 部署应用或服务的最小的基本单位。一个 Pod 封装多个应用容器（也可以只有一个容器）、存储资源、一个独立的网络 IP 以及管理控制容器运行方式的策略选项。

金丝雀发布

又称灰度发布，是迭代的软件产品在生产环境安全上线的一种重要手段。在生产环境上引一部分实际流量对一个新版本进行测试，测试新版本的性能和表现，在保证系统整体稳定运行的前提下，尽早发现新版本在实际环境上的问题。

蓝绿发布

蓝绿发布提供了一种零宕机的部署方式。不停老版本，部署新版本进行测试，确认运行正常后，将流量切到新版本，然后老版本同时也升级到新版本。始终有两个版本同时在线，有问题可以快速切换。

流量治理

应用流量治理提供可视化云原生应用的网络状态监控，并实现在线的网络连接和安全策略的管理和配置，当前支持 HTTP、TCP 流量下的路由规则、负载均衡、会话保持、连接池管理、RBAC 等能力。

连接池管理

配置 TCP 和 HTTP 的连接和请求池相关阈值，保护目标服务，避免对服务的过载访问。

熔断

配置快速响应和隔离服务访问故障，防止网络和服务调用故障级联发生，限制故障影响范围，防止故障蔓延导致系统整体性能下降或者雪崩。

1.7 规格推荐

ingressgateway 实例资源消耗参考

每个 ingressgateway 实例的资源消耗与连接类型、连接数量、QPS 有关，可以参考以下数据：

表 长连接内存消耗

连接数量	内存消耗（MB）
1	0.055
1000	55
10000	550

表 短连接 CPU 和内存消耗

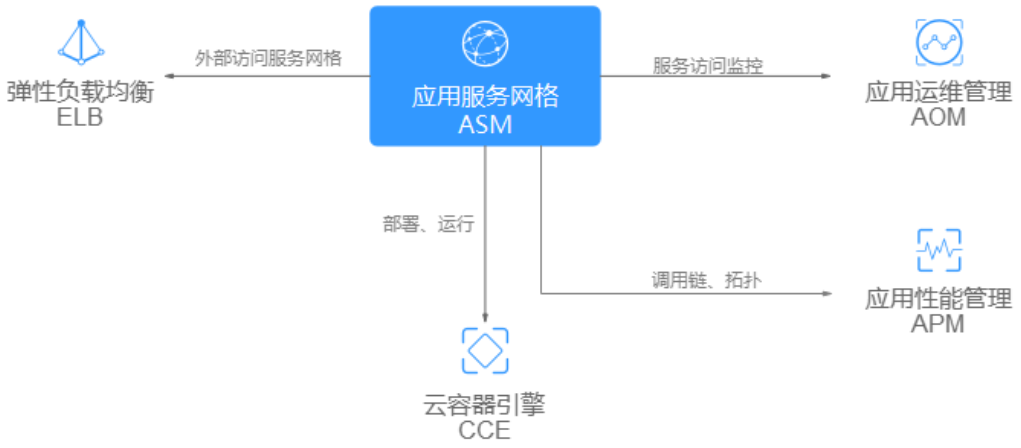
QPS	CPU 消耗（m）	内存消耗（MB）
100	30	100
1000	300	100
10000	3000	150

以上数据仅供参考，具体的资源消耗和实际的业务模型有关，以实际测试结果为准。

1.8 与其它云服务的关系

应用服务网格与周边服务的依赖关系如下图所示。

应用服务网格与其他云服务关系



云容器引擎 CCE

云容器引擎（Cloud Container Engine）提供高可靠高性能的企业级容器应用管理服务，支持 Kubernetes 社区原生应用和工具，简化云上自动化容器运行环境搭建。

您可以通过 ASM 部署运行在云容器引擎上。

弹性负载均衡 ELB

弹性负载均衡（ Elastic Load Balance，ELB）将访问流量自动分发到多台云服务器，扩展应用系统对外的服务能力，实现更高水平的应用容错。

您可通过弹性负载均衡从外部访问 ASM。

应用运维管理 AOM

应用运维管理 AOM 为您提供一站式立体运维平台，实时监控应用、资源运行状态，通过数十种指标、告警与日志关联分析，快速锁定问题根源，保障业务顺畅运行。

您可使用应用运维管理，对 ASM 中的服务和资源进行实时监控，对指标、告警和日志进行关联分析。

应用性能管理 APM

应用性能管理服务（Application Performance Management，简称 APM）是实时监控并管理云应用性能和故障的云服务，提供专业的分布式应用性能分析能力，可以帮助运维人员快速解决应用在分布式架构下的问题定位和性能瓶颈等难题，为用户体验保驾护航。

您可使用应用性能管理，对 ASM 中运行的服务进行全链路拓扑管理和分布式调用链追踪，方便您快速进行故障定位和根因分析。

2 快速入门

2.1 Bookinfo 应用的灰度发布实践

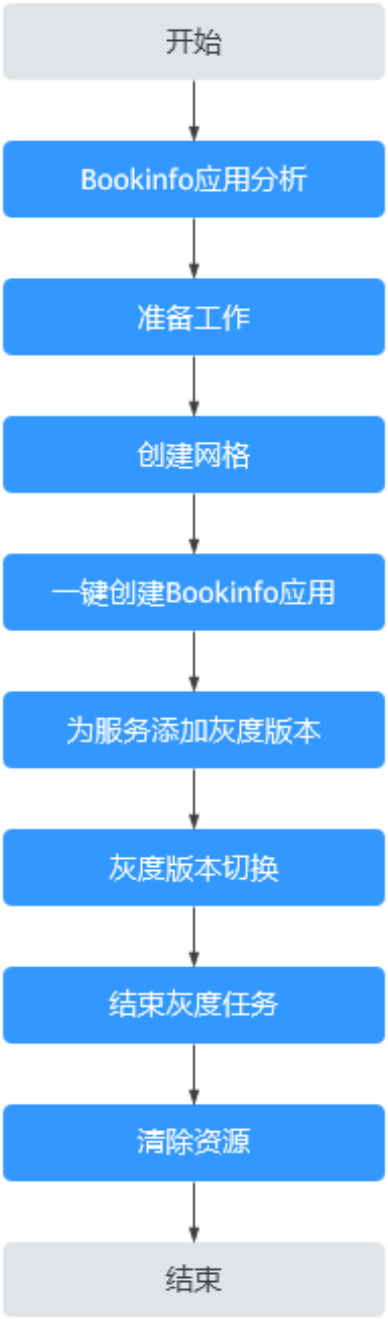
应用服务网格是基于开源 Istio 的服务网格平台，它深度、无缝对接了企业级 Kubernetes 集群服务云容器引擎（CCE），在易用性、可靠性、可视化等方面进行了一系列增强，可为客户提供开箱即用的上手体验。

入门指引

灰度发布是迭代软件产品在生产环境安全上线的一种重要手段。本教程以 Bookinfo 应用为例，向您讲解服务网格基于 Istio 提供的服务治理能力。

Bookinfo 应用灰度发布流程包含以下步骤：

图 Bookinfo 应用灰度发布流程



Bookinfo 应用分析

Bookinfo 是一个模仿在线书店的应用，页面上会显示一本书籍的描述，书籍的细节（如页数），以及关于书籍的一些评论。

Bookinfo 应用由四个单独的服务构成，几个服务是由不同的语言编写的。这些服务对应用服务网格 ASM 并无依赖，但是构成了一个有代表性的服务网格的例子，即由多个服务、多个语言构成，且 reviews 服务具有多个版本。这四个服务的说明如下：

- productpage：会调用 details 和 reviews 两个服务，用来生成页面。
- details：包含了书籍的信息。
- reviews：包含了书籍相关的评论，同时会调用 ratings 服务。
- ratings：包含了由书籍评价组成的评级信息。

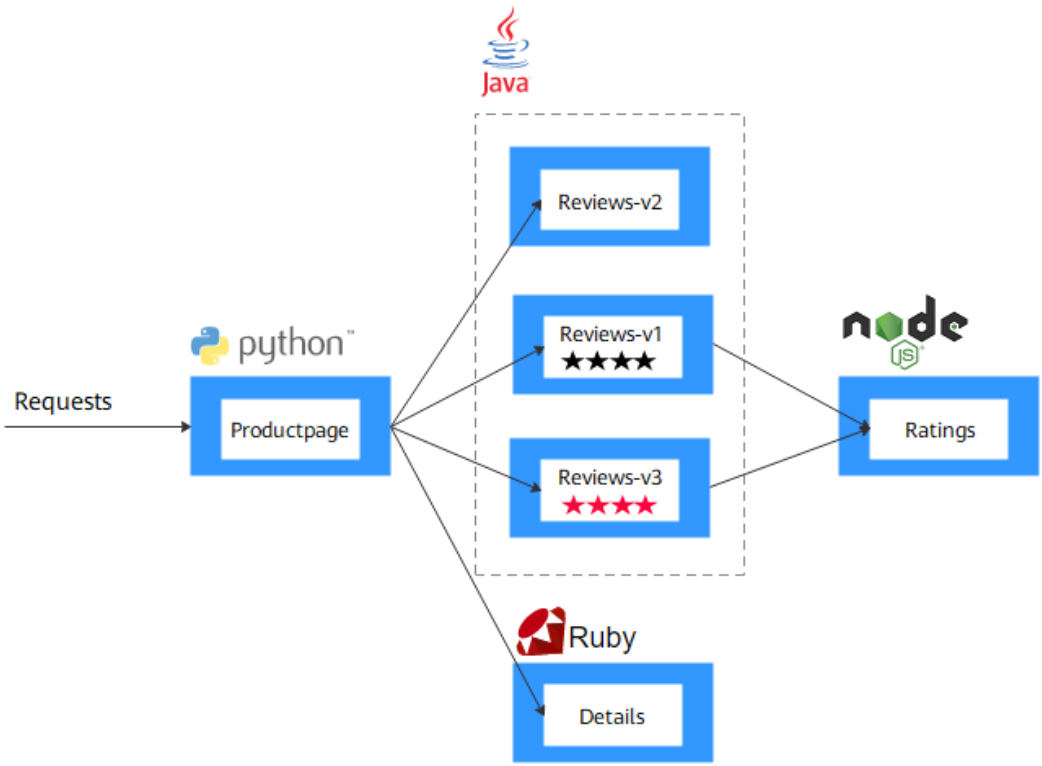
其中，reviews 服务有 3 个版本：

- v1（1.5.1）版本会调用 ratings 服务，并使用 1 到 5 个黑色星形图标来显示评分信息。
- v2（1.5.0）版本不会调用 ratings 服务。
- v3（1.5.2）版本会调用 ratings 服务，并使用 1 到 5 个红色星形图标来显示评分信息。

说明

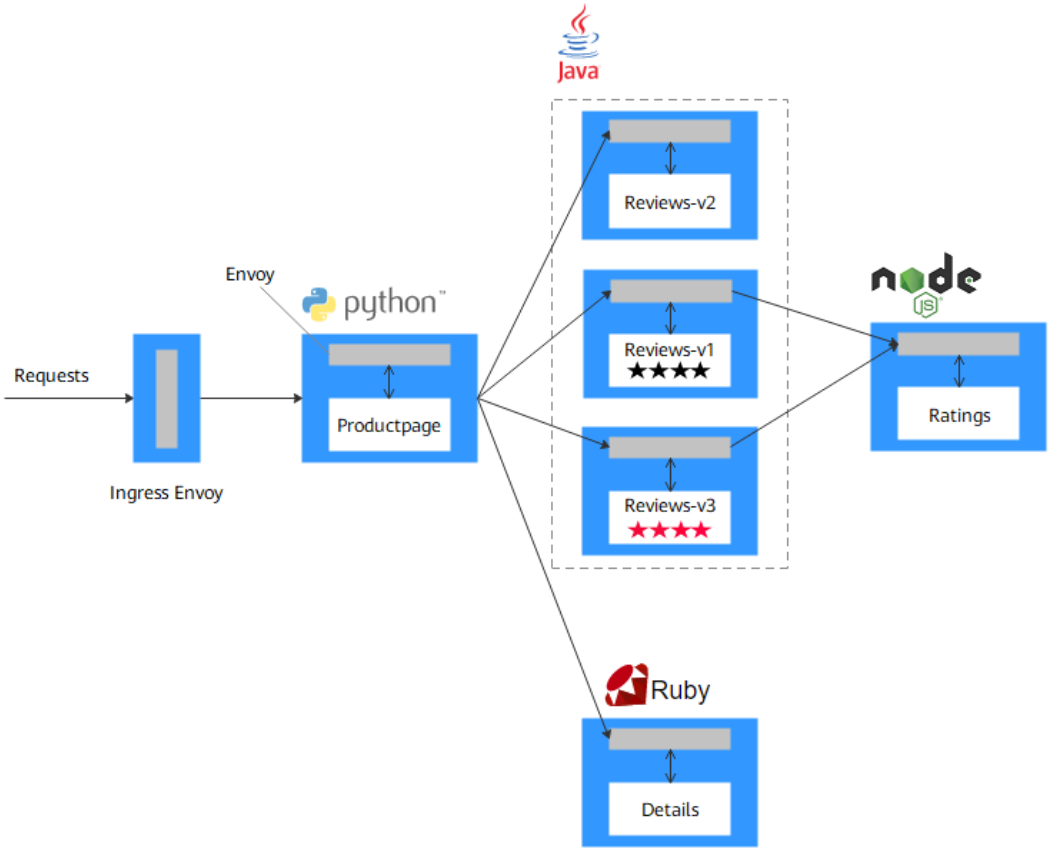
为了直观的展示灰度版本之间流量切换情况，本教程以 reviews 服务的 v1 版本（黑星形）、v3 版本（红星形）为例进行说明。

图 Bookinfo 应用的端到端架构



在 ASM 中运行 Bookinfo 应用，无需对应用自身做出任何改变，只需简单的在 ASM 环境中对服务进行配置和运行，即把 Envoy Sidecar 注入到每个服务之中。最终的部署结果如下图所示。

图 Envoy Sidecar 注入之后的 Bookinfo 应用



所有的服务都和 Envoy Sidecar 集成在一起，被集成服务的所有出入流量都被 Sidecar 所劫持，这样就可以利用 ASM 为应用提供服务路由、遥测数据收集以及策略实施等功能。

准备工作

在开始之前，您需要完成如下的准备工作。

步骤 1 创建虚拟私有云和子网。

虚拟私有云（Virtual Private Cloud，简称 VPC）提供一个隔离的、用户自主配置和管理的虚拟网络环境，可以提升资源的安全性，简化用户的网络部署。

1. 登录虚拟私有云 VPC 控制台。
2. 单击右上角“创建虚拟私有云”。
3. 根据界面提示完成参数填写，单击“立即创建”。

步骤 2 创建密钥对。

新建一个密钥对，用于远程登录节点时的身份认证。

4. 登录弹性云服务器 ECS 控制台。
5. 选择左侧导航中的“密钥对”，单击右上角“创建密钥对”。

6. 输入密钥对名称后，单击“确定”。
7. 您的浏览器会提示您下载或自动下载私钥文件。文件名是您为密钥对指定的名称，文件扩展名为“.pem”。请将私钥文件保存在安全位置。然后在系统弹出的提示框中单击“确定”。

说明

为保证安全，私钥只能下载一次，请妥善保管，否则将无法登录节点。

步骤 3 创建负载均衡。

弹性负载均衡将作为服务网格对外访问入口，被服务网格管理的应用流量，均从此实例进入并分发到后端服务。

8. 登录弹性负载均衡 ELB 控制台。
9. 单击右上角的“创建弹性负载均衡”。
10. 所属 VPC、子网请选择步骤 1 中创建的虚拟私有云和子网，其他参数根据界面提示填写，单击“立即创建”。

步骤 4 创建集群。

11. 登录云容器引擎 CCE 控制台。
12. 选择左侧导航中的“资源管理 > 集群管理”，单击右上角“创建 CCE 集群”。
13. 在“服务选型”页面设置如下参数，其余参数均采用默认值。
    - 集群名称：用户自行输入，此处设置为“cce-asm”。
    - 虚拟私有云、所在子网：选择步骤 1 中创建的虚拟私有云和子网。
14. 单击“下一步：创建节点”，配置添加节点的参数。除节点规格和登录方式外，其余参数保持默认。
    - 节点规格：vCPUs 为 4 核，内存为 8GB。

说明

此规格为部署 Bookinfo 应用所需的最小资源。

如果在创建网格中创建的网格版本为 1.3.0，则此处的节点规格需要选择 **8 核 16GB**。为了保证 sidecar 的稳定性，1.3.0 版本网格默认每个 sidecar 的 CPU 限制为 1 核，内存限制为 512M，因此需要更多资源。

    - 登录方式：选择步骤 2 中创建的密钥对，用于远程登录节点时的身份认证。
15. 单击“下一步：安装插件”，在“安装插件”步骤中选择要安装的插件。

    “系统资源插件”为必装插件，“高级功能插件”可根据实际需求进行选择性安装。
16. 单击“下一步：配置确认”，阅读使用说明并勾选“我已知晓上述限制”，确认所设置的服务选型参数、规格等信息。
17. 确认订单无误后，单击“提交”，集群开始创建。

    集群创建预计需要 6-10 分钟，您可以单击“返回集群管理”进行其他操作或单击“查看集群事件列表”后查看集群详情。

----结束

创建网格

步骤 1 登录应用服务网格 ASM 控制台。

步骤 2 单击右上角“创建网格”。

步骤 3 设置如下参数，其余参数均采用默认值。

- **网格类型**

  默认为基础版。

- **网格名称**

  设置网格的名称。

- **Istio 版本**

  网格支持的 Istio 版本。

- **集群**

  选择步骤 4 中创建的集群。

- **Istio 控制面节点**

  如果需要高可用，建议选择两个或以上不同可用区的节点。

步骤 4 设置完成后，在右侧的配置清单中确认网格配置，单击“提交”。

创建时间预计需要 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。

----结束

一键创建 Bookinfo 应用

为集群开启应用服务网格功能后，可以通过“体验任务”创建一个 Bookinfo 应用 Demo，具体操作如下：

步骤 1 登录应用服务网格 ASM 控制台。

步骤 2 单击网格名称，进入详情页面。

步骤 3 选择左侧导航中的“体验任务”，单击 Bookinfo 任务中的“安装”。

步骤 4 在右侧页面设置 Bookinfo 应用所在的集群，在“负载均衡”中选择与所选集群处于同一 VPC 和子网的负载均衡实例，并设置一个对外端口，单击“安装”。

图 安装 Bookinfo

步骤 5 等待 Bookinfo 应用创建完成。创建完成后单击网格名称，进入“服务管理”页面，配置诊断栏将显示为“正常”，Bookinfo 应用包含 productpage、details、reviews、ratings 四个服务。

图 服务列表

----结束

为服务添加灰度版本

本步骤将为 Bookinfo 应用的“reviews”服务添加新的灰度版本，并配置相应的灰度策略，将原有生产环境的默认版本的流量引流一部分到新版本中。

下面将以为“reviews”服务添加一个 v3 新版本，且 v3 新版本接收 Bookinfo 应用的30%流量为例进行配置。

部署灰度版本

步骤 1 在左侧导航中选择“灰度发布”，在金丝雀发布下，单击“立即发布”。

步骤 2 配置灰度发布基本信息。

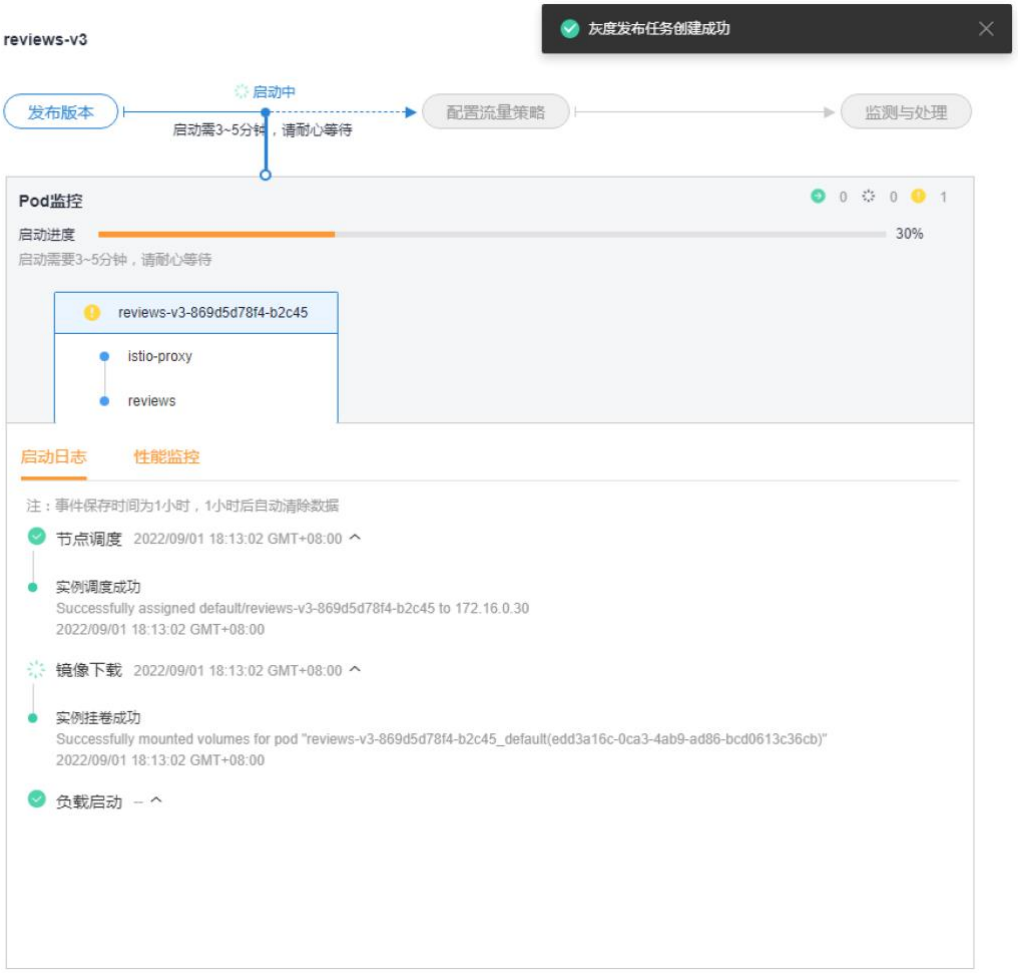
- 灰度任务名称：用户自定义，此处设置为 reviews-v3。
- 命名空间：选择服务所在命名空间。
- 灰度发布服务：在下拉框中选择 reviews。
- 工作负载：选择服务所属的工作负载。

步骤 3 配置灰度版本信息。

- 部署集群：选择服务所属的集群。
- 版本号：配置为 v3。
- 实例数量：使用默认。
- 实例配置：镜像版本选择 1.5.2，其他参数保持默认。

步骤 4 单击“发布”，待启动进度为 100%，表明灰度版本部署成功。

图 查看启动进度



----结束

配置流量策略

为灰度版本设置流量策略，灰度版本会根据配置的流量配比引流老版本中的部分或全部流量。

步骤 5 灰度版本部署成功后，单击“配置流量策略”。

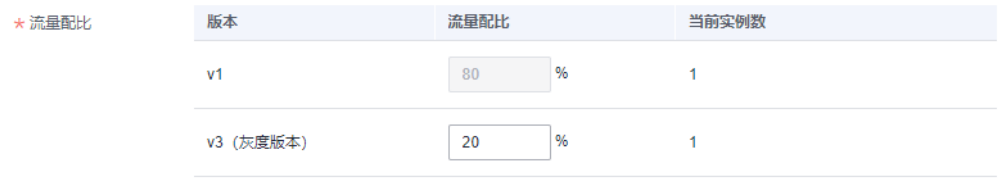
步骤 6 设置流量策略。

策略类型分为“基于流量比例”和“基于请求内容”，通过页签选择确定。

- 基于流量比例：根据流量比例配置规则，将从原版本中切分指定比例的流量到灰度版本。例如 80%的流量走原版本，20%的流量走灰度版本。
- 基于请求内容：根据请求内容配置规则，只有请求内容中满足特定条件的流量会切分到灰度版本上。例如只有在 Windows 操作系统上的用户可以访问灰度版本。

以“基于流量比例”为例，且 v3 版本流量配比为 20%。

图 流量策略



单击“策略下发”，灰度策略的生效需要几秒的时间。

步骤 7 在“服务列表”页面，单击 productpage 服务中的“访问地址”。不断刷新页面，页面在 v1 和 v3 版本之间来回切换。

图 v1 版本页面

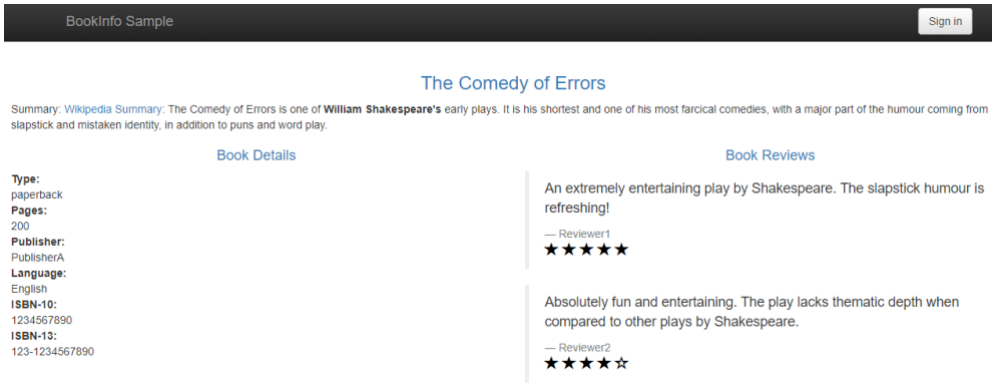
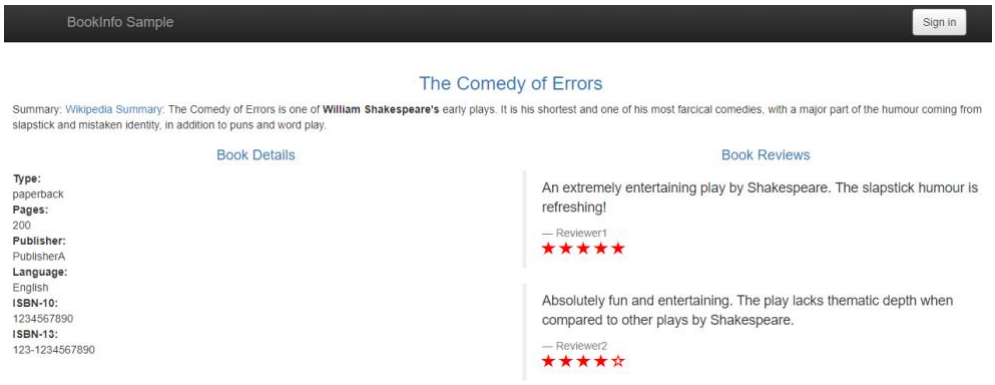


图 v3 版本页面



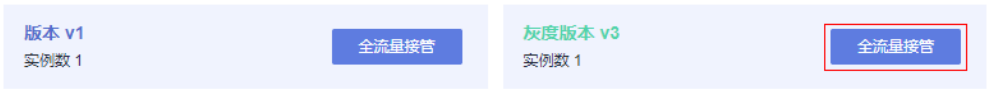
----结束

灰度版本切换

检查 v3 版本的资源数与 v1 版本是否相匹配，确认其能承接 v1 的所有流量后，即可将 v1 流量全部切换到 v3。

步骤 1 在“监测与处理”页面，单击 v3 版本后的“全流量接管”。

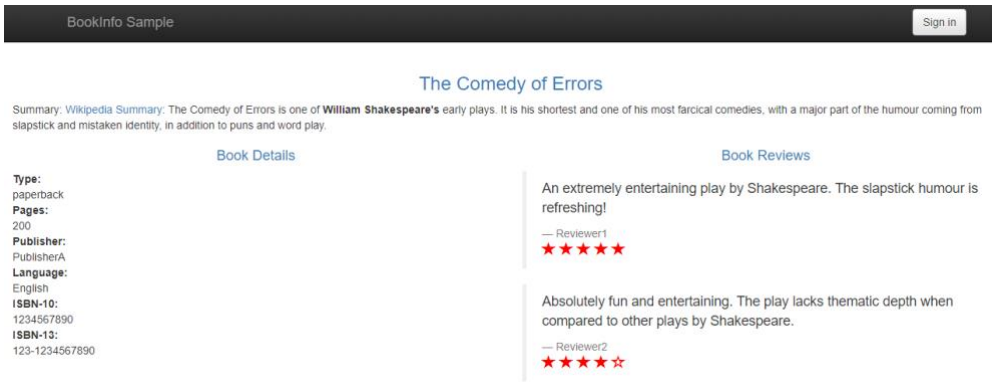
图 全流量接管



步骤 2 单击“确定”。

页面右上角会提示全流量接管成功。在 Bookinfo 应用页面，不断刷新页面，页面仅显示 v3 版本信息，即星形图标全部为红色。

图 v3 版本页面



----结束

结束灰度任务

v3 承接 v1 版本所有流量后，即可结束灰度任务，释放 v1 版本的资源。

步骤 1 在“监测与处理”页面，单击“结束灰度任务”。

步骤 2 单击“确定”，结束灰度任务将下线原版本，并删除灰度任务。

图 结束灰度任务

下线服务版本，会将包含的工作负载和 Istio 相关配置资源全部删除。

----结束

清除资源

到此本 Demo 已全部操作完成，为了避免资源的浪费，请及时删除应用和节点。

步骤 1 选择左侧导航中的“体验任务”，单击 Bookinfo 任务中的“卸载”。

步骤 2 单击“确定”。卸载 Bookinfo 体验任务，会自动删除 productpage、details、reviews、ratings 服务及相关资源。

图 卸载体验任务

说明

卸载体验任务后，已完成灰度发布的服务，其灰度版本对应的负载需要手动在 CCE 控制台删除。

----结束

2.2 为集群开通 Istio

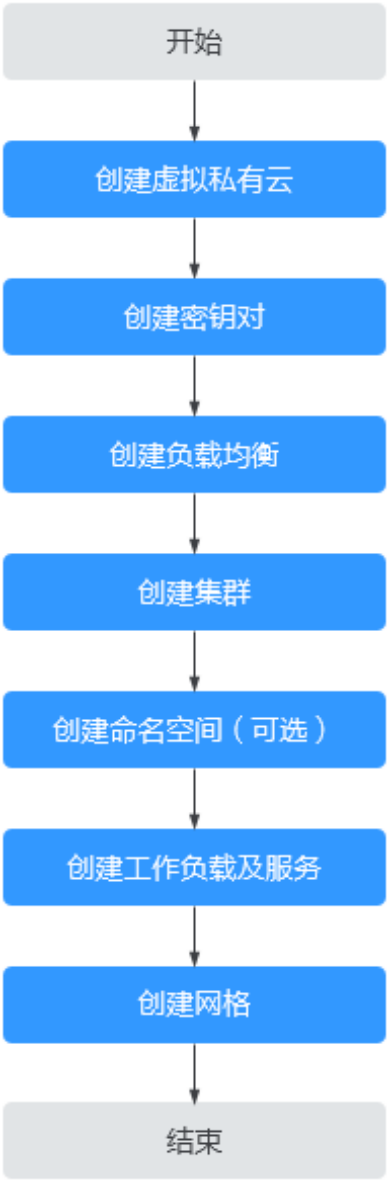
2.2.1 入门概述

应用服务网格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。

流程说明

为集群开通 Istio 的流程包含以下步骤：

图 为集群开通 Istio 的流程图



2.2.2 准备工作

为集群开通 Istio 之前，您需要完成如下的准备工作。

创建虚拟私有云

虚拟私有云（Virtual Private Cloud，简称 VPC）提供一个隔离的、用户自主配置和管理的虚拟网络环境，可以提升资源的安全性，简化用户的网络部署。

步骤 1 登录虚拟私有云 VPC 控制台。

步骤 2 单击右上角“创建虚拟私有云”。

步骤 3 根据界面提示完成参数填写，单击“立即创建”。

----结束

创建密钥对

新建一个密钥对，用于远程登录节点时的身份认证。

步骤 1 登录弹性云服务器 ECS 控制台。

步骤 2 选择左侧导航中的“密钥对”，单击右上角“创建密钥对”。

步骤 3 输入密钥对名称后，单击“确定”。

步骤 4 您的浏览器会提示您下载或自动下载私钥文件。文件名是您为密钥对指定的名称，文件扩展名为“.pem”。请将私钥文件保存在安全位置。然后在系统弹出的提示框中单击“确定”。

说明

为保证安全，私钥只能下载一次，请妥善保管，否则将无法登录节点。

----结束

创建负载均衡

弹性负载均衡将作为服务网格对外访问入口，被服务网格管理的应用流量，均从此实例进入并分发到后端服务。

步骤 1 登录弹性负载均衡 ELB 控制台。

步骤 2 单击右上角的“创建弹性负载均衡”。

步骤 3 所属 VPC、子网请选择创建虚拟私有云中创建的虚拟私有云和子网，其他参数根据界面提示填写，单击“立即创建”。

----结束

创建集群

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“资源管理 > 集群管理”，单击右上角“创建 CCE 集群”。

步骤 3 在“服务选型”页面设置如下参数，其余参数均采用默认值。

- 集群名称：用户自行输入，此处设置为“cluster-test”。
- 虚拟私有云、所在子网：选择创建虚拟私有云中创建的虚拟私有云和子网。

步骤 4 单击“下一步：创建节点”，配置添加节点的参数。除节点规格和登录方式外，其余参数保持默认。

- 节点规格：vCPUs 为 4 核，内存为 8GB。
- 登录方式：选择创建密钥对中创建的密钥对，用于远程登录节点时的身份认证。

步骤 5 单击“下一步：安装插件”，在“安装插件”步骤中选择要安装的插件。

“系统资源插件”为必装插件，“高级功能插件”可根据实际需求进行选择性安装。

步骤 6 单击“下一步：配置确认”，阅读使用说明并勾选“我已知晓上述限制”，确认所设置的服务选型参数、规格等信息。

步骤 7 确认订单无误后，单击“提交”，集群开始创建。

集群创建预计需要 6-10 分钟，您可以单击“返回集群管理”进行其他操作或单击“查看集群事件列表”后查看集群详情。

----结束

创建命名空间（可选）

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“资源管理 > 命名空间”，单击右上角“创建命名空间”。

步骤 3 输入命名空间的名称，并选择已创建的集群。

步骤 4 单击“确定”。

----结束

创建工作负载及服务

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“工作负载 > 无状态负载 Deployment”，单击右上角“创建无状态工作负载”。

步骤 3 参考《云容器引擎 用户指南》中的指导，创建工作负载和服务。

----结束

2.2.3 创建网格

ASM 支持创建基础版网格，提供商用级网格服务。

操作步骤

步骤 1 登录应用服务网格 ASM 控制台，单击右上角“创建网格”。

步骤 2 设置如下参数，其余参数均采用默认值。

- **网格类型**
  默认为基础版。
- **网格名称**
  设置网格的名称。
- **Istio 版本**
  网格支持的 Istio 版本。

- **集群**

  选择创建集群中创建的集群。

- **Istio 控制面节点**

  如果需要高可用，建议选择两个或以上不同可用区的节点。

步骤 3 设置完成后，在右侧的配置清单中确认网格配置，单击“提交”。

创建时间预计需要 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。

----结束

2.3 配置式应用灰度发布

2.3.1 入门概述

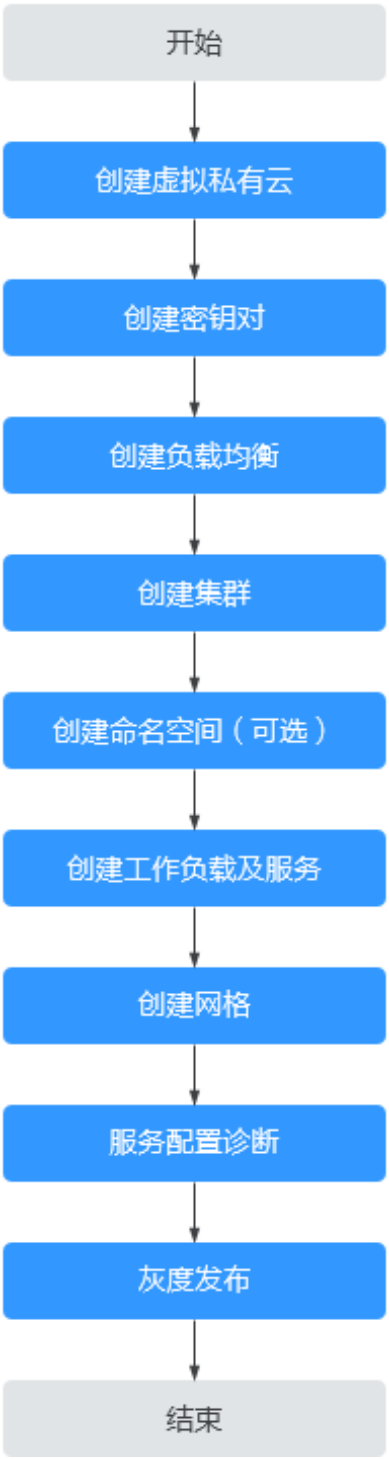
灰度发布是版本升级平滑过渡的一种方式，当版本升级时，使部分用户使用高版本，其他用户继续使用低版本，待高版本稳定后，逐步扩大范围把所有用户流量都迁移到高版本上面来。

本章将从创建虚拟私有云开始，到创建一个灰度版本，讲解如何实现一次灰度发布。

流程说明

实现灰度发布的流程包含以下步骤：

图 实现灰度发布的流程图



2.3.2 准备工作

应用灰度发布之前，您需要完成如下的准备工作。

创建虚拟私有云

虚拟私有云（Virtual Private Cloud，简称 VPC）提供一个隔离的、用户自主配置和管理的虚拟网络环境，可以提升资源的安全性，简化用户的网络部署。

步骤 1 登录虚拟私有云 VPC 控制台。

步骤 2 单击右上角“创建虚拟私有云”。

步骤 3 根据界面提示完成参数填写，单击“立即创建”。

----结束

创建密钥对

新建一个密钥对，用于远程登录节点时的身份认证。

步骤 1 登录弹性云服务器 ECS 控制台。

步骤 2 选择左侧导航中的“密钥对”，单击右上角“创建密钥对”。

步骤 3 输入密钥对名称后，单击“确定”。

步骤 4 您的浏览器会提示您下载或自动下载私钥文件。文件名是您为密钥对指定的名称，文件扩展名为“.pem”。请将私钥文件保存在安全位置。然后在系统弹出的提示框中单击“确定”。

说明

为保证安全，私钥只能下载一次，请妥善保管，否则将无法登录节点。

----结束

创建负载均衡

弹性负载均衡将作为服务网格对外访问入口，被服务网格管理的应用流量，均从此实例进入并分发到后端服务。

步骤 1 登录弹性负载均衡 ELB 控制台。

步骤 2 单击右上角的“创建弹性负载均衡”。

步骤 3 所属 VPC、子网请选择创建虚拟私有云中创建的虚拟私有云和子网，其他参数根据界面提示填写，单击“立即创建”。

----结束

创建集群

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“资源管理 > 集群管理”，单击右上角“创建 CCE 集群”。

步骤 3 在“服务选型”页面设置如下参数，其余参数均采用默认值。

- 集群名称：用户自行输入，此处设置为“cluster-test”。

- 虚拟私有云、所在子网：选择创建虚拟私有云中创建的虚拟私有云和子网。

步骤 4 单击“下一步：创建节点”，配置添加节点的参数。除节点规格和登录方式外，其余参数保持默认。

- 节点规格：vCPUs 为 4 核，内存为 8GB。
- 登录方式：选择创建密钥对中创建的密钥对，用于远程登录节点时的身份认证。

步骤 5 单击“下一步：安装插件”，在“安装插件”步骤中选择要安装的插件。

“系统资源插件”为必装插件，“高级功能插件”可根据实际需求进行选择性安装。

步骤 6 单击“下一步：配置确认”，阅读使用说明并勾选“我已知晓上述限制”，确认所设置的服务选型参数、规格等信息。

步骤 7 确认订单无误后，单击“提交”，集群开始创建。

集群创建预计需要 6-10 分钟，您可以单击“返回集群管理”进行其他操作或单击“查看集群事件列表”后查看集群详情。

----结束

创建命名空间（可选）

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“资源管理 > 命名空间”，单击右上角“创建命名空间”。

步骤 3 输入命名空间的名称，并选择已创建的集群。

步骤 4 单击“确定”。

----结束

创建工作负载及服务

步骤 1 登录云容器引擎 CCE 控制台。

步骤 2 选择左侧导航中的“工作负载 > 无状态负载 Deployment”，单击右上角“创建无状态工作负载”。

步骤 3 参考《云容器引擎 用户指南》中的指导，创建工作负载和服务。

----结束

创建网格

步骤 1 登录应用服务网格 ASM 控制台，单击右上角“创建网格”。

步骤 2 设置网格名称为“asmtest”，选择在创建集群中创建的名称为“cluster-test”的集群，并选择安装 Istio 控制面的节点，建议选择两个或以上不同可用区的节点。

步骤 3 在“sidecar 配置”中勾选名称为“default”的命名空间，选择重启已有服务。

步骤 4 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。

----结束

服务配置诊断

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才能进行灰度发布。

步骤 1 登录应用服务网格 ASM 控制台，单击名称为“asmtest”的网格，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“命名空间：default”并查看“servicetest”的配置诊断状态。

步骤 3 如果配置诊断为异常，单击“处理”，根据修复指导的操作修复，直到服务的配置诊断状态为正常。

----结束

2.3.3 灰度发布

为服务添加灰度版本

步骤 1 登录应用服务网格 ASM 控制台，单击“asmtest”网格内的  。

步骤 2 创建灰度任务名称为“test”的灰度任务，并配置基本信息及灰度版本信息，灰度发布服务选择创建工作负载及服务中创建的名称为“servicetest”的服务，工作负载会自动关联“deptest”，单击“发布”。

图 创建灰度任务

说明

如果服务“servicetest”无法选择，需要确认服务是否异常，修复后才能选择。

步骤 3 单击“配置流量策略”，选择“基于流量比例”策略类型，为 v2（灰度版本）配置“流量配比”为 80%。

图 配置流量策略

步骤 4 单击“策略下发”。

灰度策略的生效需要几秒的时间，您可以在监测灰度运行状态页面，观察灰度版本的运行状态。

----结束

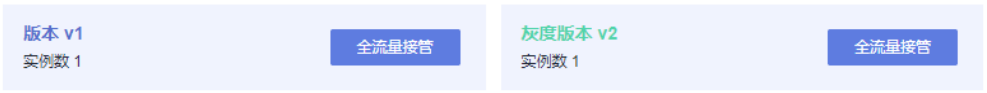
灰度版本切换

检查 v2 版本的资源数与 v1 版本是否相匹配，确认其能承接 v1 的所有流量后，即可将 v1 流量全部切换到 v2。

步骤 1 在“灰度发布”页面，单击灰度任务名为“test”后的“监测与处理”。

步骤 2 单击灰度版本 v2 后的“全流量接管”。

图 全流量接管



步骤 3 在弹出的“全流量接管”窗口单击“确定”，页面右上角会提示全流量接管成功。

----结束

将原版本下线

v2 承接 v1 所有流量后，即可删除 v1 版本，释放 v1 版本的资源。

步骤 1 在“灰度发布”页面，单击灰度任务名为“test”后的“监测与处理”。

步骤 2 在“监测与处理”页面中，当 v2 版本中的流量占比为 100%时，表明 v2 已承接 v1 所有流量。单击“结束灰度任务”。

步骤 3 确认无误后，单击“确定”。

结束灰度任务将下线 v1 版本，并删除 test 灰度任务。

----结束

3 创建网格

ASM 支持创建基础版网格，提供商用级网格服务。

前提条件

已创建 CCE 集群，并确保集群版本为 v1.15、v1.17 或 v1.19。如果未创建，请先创建 CCE 集群。

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。
- 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“创建网格”。

步骤 3 设置网格参数。

- **网格类型**

  默认为基础版网格。

- **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。

  同一帐号下网格不可重名，且网格名称创建后不可修改。

- **Istio 版本**

  网格支持的 Istio 版本。

- **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。

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。

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

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

4 网格管理

4.1 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。

- 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。

----结束

5 服务管理

5.1 配置诊断

服务诊断

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。

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 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，需要先手动处理异常状态的检查项，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”，根据修复指导进行手动修复。

图 手动处理项

步骤 2 手动处理异常状态的检查项后，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 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。

----结束

5.2 手动修复项

5.2.1 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}

5.2.2 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：
- 查询 Deployment 的 ReplicaSet。

  kubectl get replicaset | grep {deploymentName}

1. 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

5.2.3 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，“所有 Pod 是否都注入了 sidecar”检查结果会失败。

修复指导

请检查命名空间是否配置了自动注入 sidecar，如果配置后重启 Pod 仍然无法注入，请检查实例数量是否超过了套餐限制。

- 如果网格未开启命名空间注入，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：
  a. 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**

  b. 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。
- 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，或者检查网格实例数量是否已经超出套餐配额。

5.3 自动修复项

5.3.1 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“Service 的端口名称是否符合 istio 规范”检查结果会失败。

修复指导

步骤 1 登录 CCE 控制台。

步骤 2 在左侧导航栏选择“资源管理 > 网络管理”，在“Service”页签中按照集群名称和命名空间搜索服务，单击对应服务后的“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

5.3.2 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。

步骤 2 在左侧导航栏选择“资源管理 > 网络管理”，在“Service”页签中按照集群名称和命名空间搜索服务，单击对应服务后的“编辑 YAML”，查看 Service 的选择器（spec.selector），删除已配置的 version 标签。

```
36 spec:
37   ports:
38     - name: http-service0
39       protocol: TCP
40       port: 8000
41       targetPort: 80
42   selector:
43     app: nginx
44     version: v1
```

----结束

5.3.3 Service 是否配置了 app 标签，值是否正确

问题描述

Service 的标签（metadata.labels）中，需要配置 app 标签，并和关联的 Deployment 中（spec.template.metadata.labels）Pod 的 app 标签一致。如果未配置或者值不一致，会报此异常。

说明

app 标签在流量监控中用于流量的跟踪。

修复指导

步骤 1 登录 CCE 控制台。

步骤 2 在左侧导航栏选择“工作负载 > 无状态负载 Deployment”，单击对应负载后的“更多 > 编辑 YAML”，查看 Deployment 中 Pod 的 app 标签（spec.template.metadata.labels）。

```
117 spec:
118   replicas: 1
119   selector:
120     matchLabels:
121       app: nginx
122       version: v1
123   template:
124     metadata:
125       creationTimestamp: null
126       labels:
127         app: nginx
128         version: v1
```

步骤 3 在左侧导航栏选择“资源管理 > 网络管理”，单击对应服务后的“编辑 YAML”，查看 Service 的 app 标签（metadata.labels）。

```
metadata:
  name: nginx
  namespace: default
  selfLink: /api/v1/namespaces/default/services/nginx
  uid: b2f3bf23-9ef1-4101-882a-f4eb151a0da9
  resourceVersion: '1478205'
  creationTimestamp: '2021-12-15T11:56:28Z'
  labels:
    app: nginx
  managedFields:
```

步骤 4 修改 Service 的 app 标签与 Pod 的 app 标签值一致。

----结束

5.3.4 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5-1 VirtualService 的协议类型

```
spec:
  hosts:
    - ratings
  http:
    - route:
        - destination:
            host: ratings
            port:
              number: 9080
            subset: v1
```

图5-2 Service 的端口协议类型

```
spec:
  ports:
    - name: http-ratings
      protocol: TCP
      port: 9080
      targetPort: 9080
  selector:
    app: ratings
```

----结束

6 网关管理

6.1 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。

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。

- **负载均衡配置**

  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，支持共享型和独享型规格，且支持公网和私网。

  负载均衡实例需与当前集群处于相同 VPC。

- **监听器配置**

  服务网关为负载均衡器配置监听器，监听器对负载均衡器上的请求进行监听，并分发流量。

  - **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。

- **TLS 终止**

  配置 TLS 协议时，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 终止时，提供 TLS 最低版本/TLS 最高版本的选择。

图 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（转发规则由域名+URL 组成，域名置空时，默认为 ELB IP 地址）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。

- **域名**

  请填写组件对外发布域名。不填时访问地址默认为负载均衡实例 IP 地址。如果您开启了 TLS 终止，则必须填写证书内认证域名，以完成 SNI 域名校验。

- **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**

  重写 HTTP URI 和 Host/Authority 头，于转发前执行。默认关闭。

图 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 服务外网访问地址

----结束

6.2 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。

- **域名**

  请填写组件对外发布域名。不填时访问地址默认为负载均衡实例 IP 地址。如果您开启了 TLS 终止，则必须填写证书内认证域名，以完成 SNI 域名校验。

- **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9.3.4 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5.2 手动修复项或5.3 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**

  重写 HTTP URI 和 Host/Authority 头，于转发前执行。默认关闭。

步骤 3 配置完成后，单击“确定”。

----结束

7 灰度发布

7.1 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 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 蓝绿发布流程

7.2 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7.1 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 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。

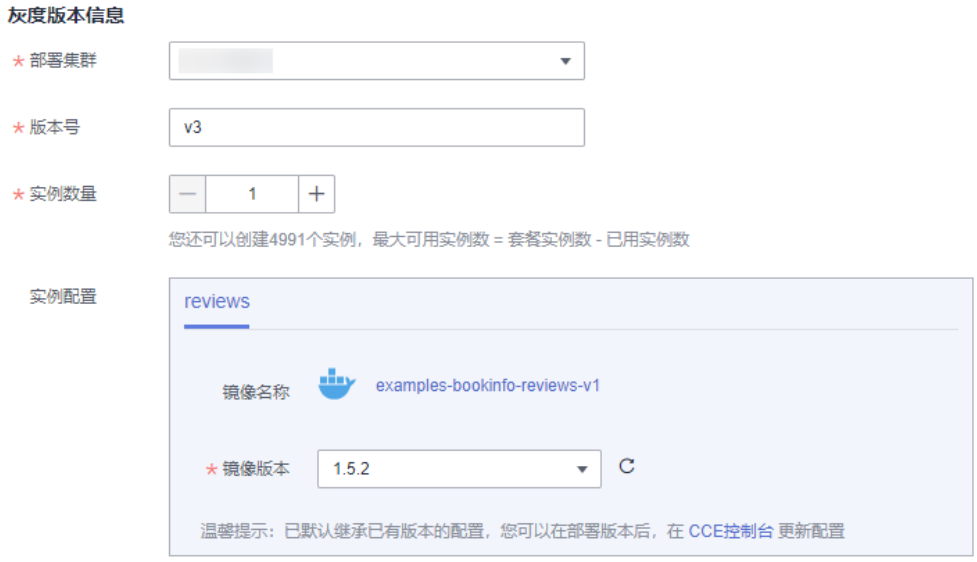
- **镜像名称**

  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。

图 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。

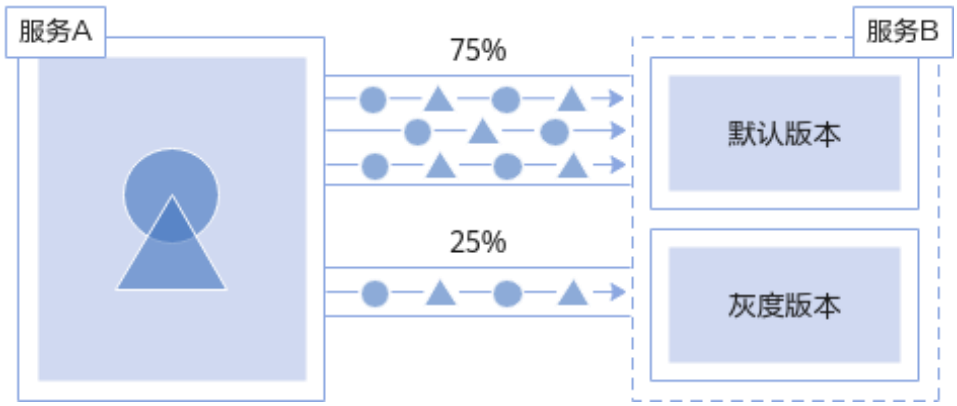
步骤 5 （仅金丝雀发布涉及）单击“配置灰度策略”，进行灰度策略配置。

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 基于流量比例

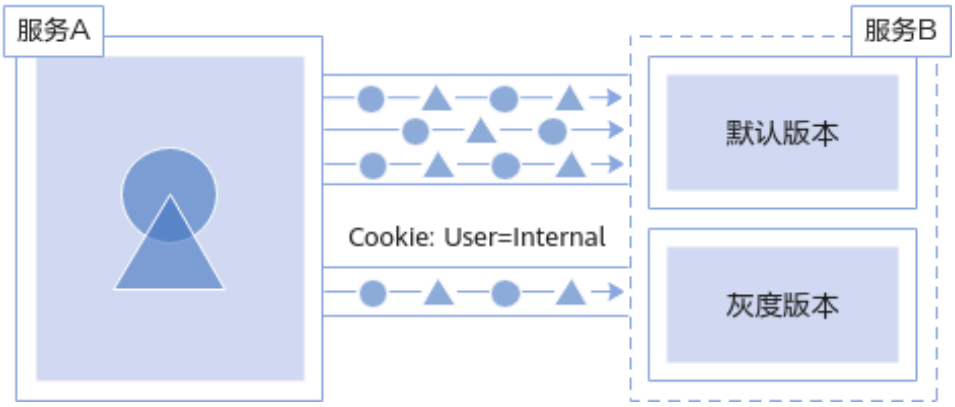


流量配比：可以为默认版本与灰度版本设置流量配比，系统将根据输入的流量配比来确定流量在两个版本间分发的比重。

- **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 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（详情可参考如何使用 Istio 调用链埋点），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

7.3 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，原本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 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

8 网格配置

8.1 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、可观测性以及创建时间。
- “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。

8.2 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

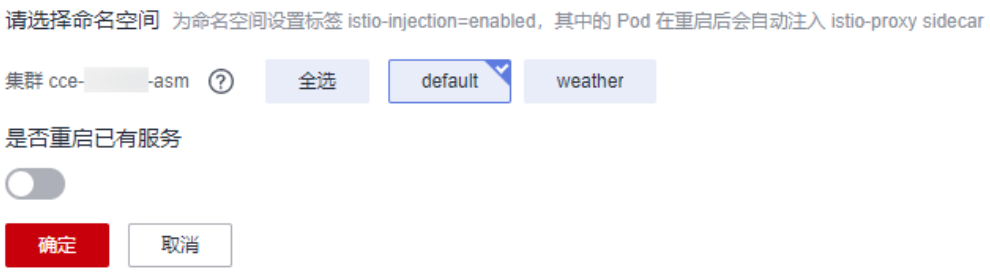
可展示当前已注入 sidecar 的命名空间及所属集群。如果还未做过注入操作，或者需要为更多命名空间注入 sidecar，请参考以下操作：

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 注入”，选择命名空间，判断是否重启已有服务，单击“确定”。

图 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。当然了，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。

----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 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

8.3 istio 资源管理

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。

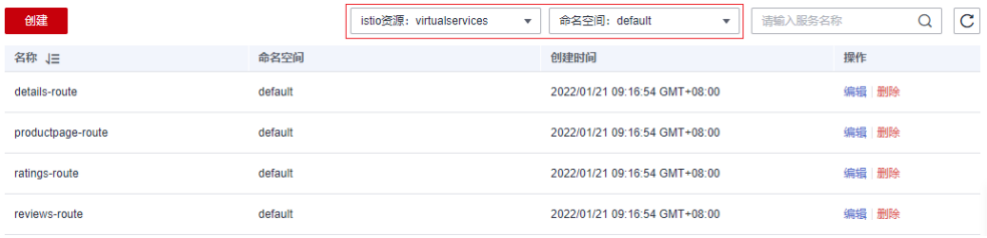
编辑已有 istio 资源

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 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后单击“确定”保存。

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 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 确认信息无误后，单击“确定”。

----结束

8.4 升级

8.4.1 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。

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	1.8.4-r4	补丁更新
	1.8.4-r2	1.8.4-r4	补丁更新
	1.8.4-r3	1.8.4-r4	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。

- **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”。

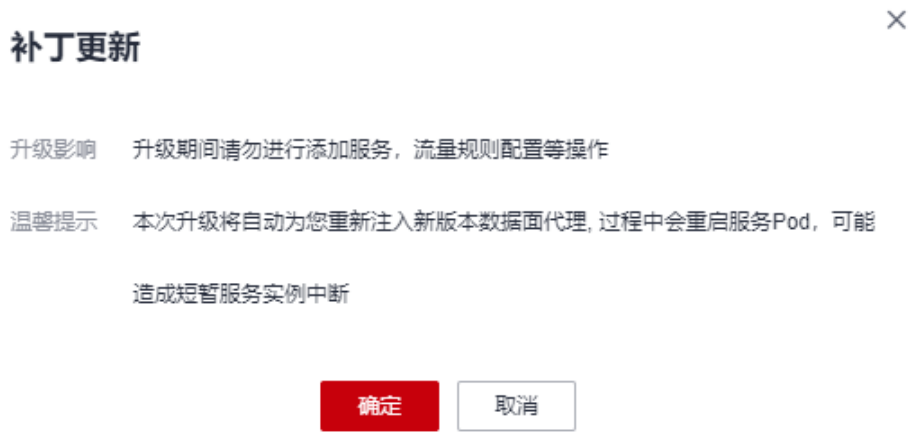
  说明

  1.3.0 版本升级至 1.8.4 时，VirtualService 格式检查项的结果为“警告”，原因是 1.8 及以上版本网格界面仅支持显示 delegate 格式的 VirtualService，需要根据8.4.5 1.3 升级 1.8 VirtualService 支持 Delegate 切换进行修改，否则升级后将看不到创建的网关路由。不过，该检查项并不会导致升级失败。

- **补丁更新**

  单击“补丁更新”，在弹出的提示框中单击“确定”。

  图 补丁更新



----结束

8.4.2 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
- 私有版本 Virtual Service Delegation 上线，提前为大客户提供解耦 Gateway 流量配置

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

8.4.3 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（API 和 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
- 支持控制面托管和非托管形态
- 支持多端口服务基于端口粒度的服务治理

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

8.4.4 1.8 版本特性

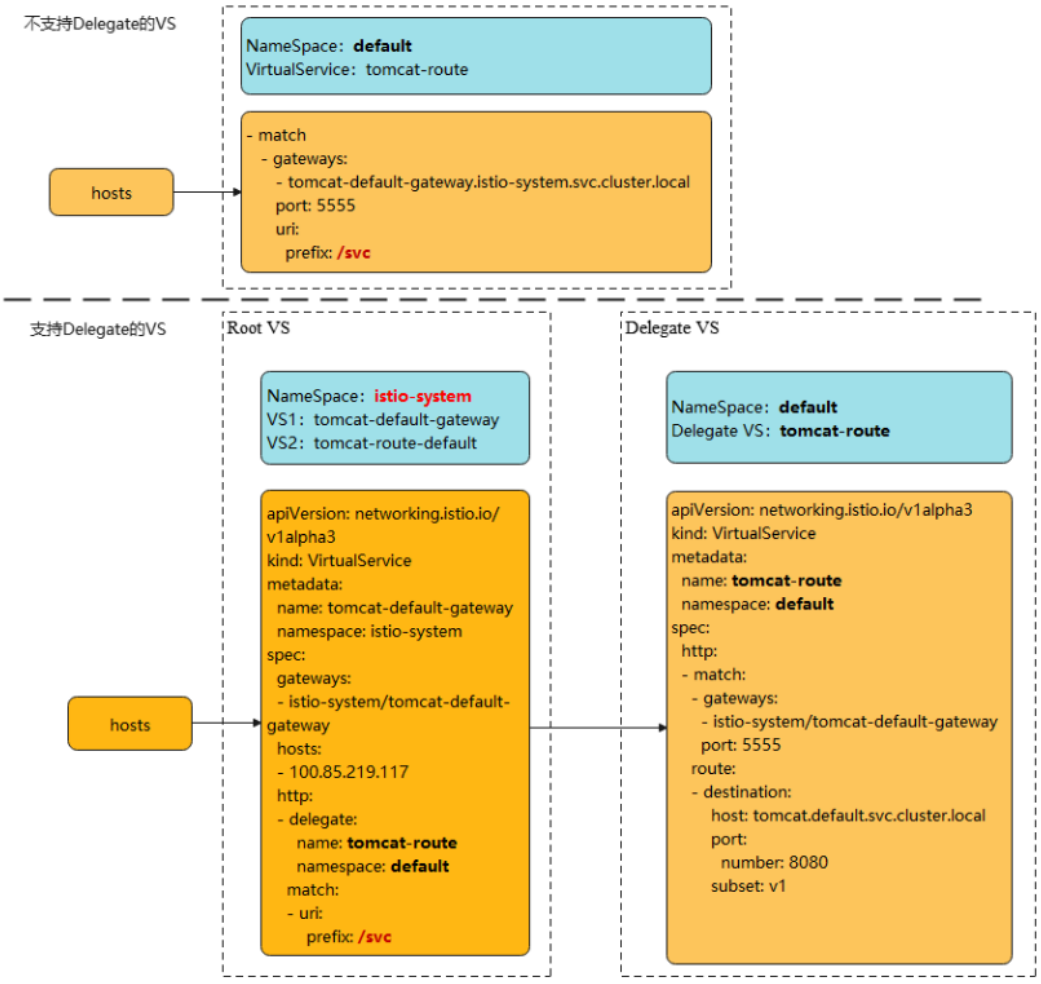
- 托管版本支持扁平网络多集群
- 托管版本支持非扁平网络多集群
- Mixer 组件正式下线，访问日志、调用链和监控均基于数据面采集
- EnvoyFilter 增强，支持更多灵活的 Insert 操作
- 支持 VM 类型服务治理
- 启用基于 AuthorizationPolicy 的新的授权策略

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

8.4.5 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，则升级后进行如下修改，若升级前进行修改则 apiVersion 不变：

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

二、若升级前服务添加了网关，则升级后进行如下修改，若升级前进行修改则 apiVersion 不变：

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

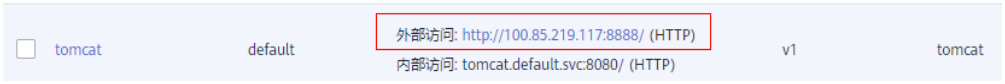
   b. tomcat-default-gateway.istio-system.svc.cluster.local 替换成 istio-system/tomcat-default-gateway

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

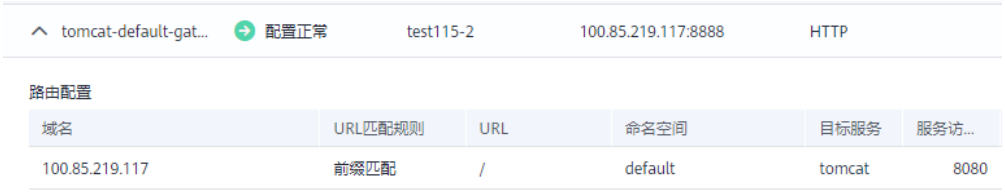
   d. destination.host:格式为{服务名}.{namespace}.svc.cluster.local

```
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~~
  http:
  - match:
    - gateways:
      - istio-system/tomcat-default-gateway
      port: 5555
~~    uri:~~
~~      prefix: /test~~
    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。



9 常见问题

9.1 网格集群

9.1.1 启用服务网格后，状态一直为安装中

问题描述

为 CCE 集群启用服务网格（即创建网格）后，网格状态一直显示为“安装中”，鼠标放上去提示“正在启用 istio 服务网格：开通用户安全组规则成功”。

问题定位

登录 CCE 控制台，在“资源管理 > 命名空间”中查看对应集群的 istio-system 命名空间是否存在。

原因分析

存在 istio-system 命名空间残留。

解决方法

删除已有的 istio-system 命名空间后即可继续安装。

9.1.2 卸载服务网格后，状态一直为未就绪

问题描述

在 ASM 控制台卸载服务网格后，网格状态一直显示为“未就绪”。

问题定位

步骤 1 登录 CCE 控制台，在左侧导航栏选择“模板市场 > 示例模板”。

步骤 2 单击“模板实例”页签，在搜索框中选择对应集群，查看模板实例和卸载失败最新事件。

可以看到 istio-master 模板实例的执行状态为“卸载失败”，并且最新事件提示如下信息：

```
deletion failed with 1 error(s): clusterroles:rbac.authorization.k8s.io "istio-cleanup-secrets-istio-system" already exists
```

----结束

原因分析

helm 对中断状态支持不好，客户异常操作会导致 istio 的 helm 模板卡在中间状态，使卸载过程中留下残留资源，从而导致卸载失败。

解决方法

步骤 1 通过 kubectl 连接到 CCE 集群。

步骤 2 执行以下命令，清理 istio 相关资源。

```
kubectl delete ServiceAccount -n istio-system `kubectl get ServiceAccount -n istio-system | grep istio | awk '{print $1}'`
kubectl delete ClusterRole -n istio-system `kubectl get ClusterRole -n istio-system | grep istio | awk '{print $1}'`
kubectl delete ClusterRoleBinding -n istio-system `kubectl get ClusterRoleBinding -n istio-system | grep istio | awk '{print $1}'`
kubectl delete job -n istio-system `kubectl get job -n istio-system | grep istio | awk '{print $1}'`
kubectl delete crd -n istio-system `kubectl get crd -n istio-system | grep istio | awk '{print $1}'`
kubectl delete mutatingwebhookconfigurations -n istio-system `kubectl get mutatingwebhookconfigurations -n istio-system | grep istio | awk '{print $1}'`
```

步骤 3 登录 ASM 控制台，重新执行卸载操作。

----结束

9.2 网格管理

9.2.1 为什么我的集群不能启用网格？

问题描述

集群不能启用网格。

原因分析

暂不支持 v1.15 以下版本集群启用网格。

解决方法

步骤 1 检查您的集群版本，目前仅对 v1.15、v1.17 和 v1.19 版本的集群生效。

步骤 2 检查您的浏览器，请尽量使用 Chrome 浏览器访问我们的服务，火狐等浏览器可能因为适配的问题，导致启用网格按钮灰化。

----结束

9.2.2 Istio 卸载之后，为什么独享节点还在？

问题描述

Istio 卸载后独享节点还在。

原因分析

Istio 仅会卸载 Istio 相关控制面组件，不会主动卸载您的节点资源。

解决方法

卸载后的节点，您可以作为普通负载节点使用。如不再需要，请登录 CCE 控制台，在“资源管理 > 节点管理”中，删除该节点。

9.2.3 如何为集群开放命名空间注入？

为集群的命名空间注入 sidecar 时，若集群未开放命名空间注入，请参考如下指导修改集群配置：

步骤 1 通过 kubectl 连接集群。

步骤 2 执行 **kubectl edit cm -nistio-system istio-sidecar-injector**，修改 istio-sidecar-injector 配置项。

步骤 3 将 data.config.policy 改为 enabled。

```
data:
  config: |-
    policy: enabled
```

----结束

9.2.4 某些工作负载不注入 sidecar，该如何配置？

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。不过有些工作负载因为种种原因不能注入 sidecar，可参考如下指导进行配置：

步骤 1 登录 CCE 控制台，在工作负载所在行单击“编辑 YAML”。

步骤 2 找到 spec.template.metadata.annotations 字段，添加 sidecar.istio.io/inject: 'false'。

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

```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: reviews
      version: v1
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: reviews
        release: istio-bookinfo
        version: v1
      annotations:
        sidecar.istio.io/inject: 'false'
```

您可以单击 Automatic Sidecar Injection 了解更多 sidecar 注入的知识。

----结束

9.3 添加服务

9.3.1 添加的对外访问方式不能生效，如何排查？

出现上述问题可能是访问相关的资源配置有缺失或错误，请按照如下方法进行排查：

- 通过弹性负载均衡服务界面查看使用的 ELB 是否成功监听使用的外部端口和弹性云服务器。
- 登录集群，使用 **kubectl get gateway -n istio-system** 命令查看使用的 gateway 是否配置好使用的 IP/域名和端口。使用 **kubectl get svc -n istio-system** 命令查看使用的 ingressgateway 是否有对应的 IP 和端口，且未处于 pending 状态。
- 核实加入服务网格的内部访问协议和添加网络配置的外部访问协议一致。
- 如果通过浏览器访问出现“ERR_UNSAFE_PORT”错误，是因为该端口被浏览器识别为危险端口，此时应更换为其他外部端口。

9.3.2 一键创建体验应用为什么启动很慢？

体验应用包含 productpage、details、ratings 和 reviews 4 个服务，需要创建所有相关的工作负载和 Istio 相关的资源（DestinationRule、VirtualService、Gateway）等，因此创建时间较长。

9.3.3 一键创建体验应用部署成功以后，为何不能访问页面？

问题描述

一键创建体验应用部署成功后不能访问页面。

原因分析

弹性负载均衡 ELB 未成功监听端口。

解决方法

请在弹性负载均衡 ELB 中查看该端口监听器是否创建，后端服务器健康状态是否正常。

9.3.4 添加路由时，为什么选不到对应的服务？

添加路由时，目标服务会根据对应的网关协议进行过滤。过滤规则如下：

- HTTP 协议的网关可以选择 HTTP 协议的服务
- TCP 协议的网关可以选择 TCP 协议的服务
- GRPC 协议的网关可以选择 GRPC 协议的服务
- HTTPS 协议的网关可以选择 HTTP、GRPC 协议的服务
- TLS 协议的网关如果打开了 TLS 终止，只能选择 TCP 协议的服务；关闭了 TLS 终止，只能选择 TLS 协议的服务

9.4 灰度发布

9.4.1 灰度发布部署版本为什么不能更换镜像？

问题描述

灰度发布部署灰度版本时不能更换镜像类型。

原因分析

灰度发布针对服务的同一镜像，只允许选择不同的版本号。

解决方法

将所需镜像打包成同一镜像的不同版本并上传至镜像仓库。

9.5 流量监控

9.5.1 Pod 刚刚启动后，为什么不能立即看到流量监控数据？

流量监控对采集到的数据进行了聚合处理，需积累一分钟才能看到数据。

9.5.2 总览页面，上面的异常响应数据和时延数据，为什么不准确？

异常响应数据和时延数据，是显示您帐户下全部的集群的全部组件的 topN 数据，且是近一分钟的数据。所以请确保您的组件在近 1 分钟内，有访问流量产生。

9.5.3 流量占比与流量监控图为什么数据不一致？

流量占比的数据是 10 秒轮询，流量监控是 10 秒取值。