

服务网格 操作指南 产品文档





【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有,未经腾讯云事先书面许可,任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况,部分产品、服务的内容可能有所调整。您 所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则, 腾讯云对本文档内容不做任何明示或默示的承诺或保证。



文档目录

操作指南 网格实例管理 概述 创建网格 升级网格 更新网格配置 Sidecar 注入与配置 删除网格 服务发现管理 概述 自动服务发现 手动服务注册 边缘代理网关 边缘代理网关管理 Gateway 配置 流量管理 概述 VirtualService 配置路由规则 DestinationRule 配置服务版本和流量策略 可观测性 概述 监控指标 Metric 调用追踪 Trace 访问日志 Access Log 安全 Authentication 认证策略配置 Authorization 授权策略配置 访问管理 概述 CAM 服务角色授权 CAM 预设策略授权 CAM 自定义策略授权 功能扩展 使用 wasm filter 扩展数据面



操作指南 网格实例管理 概述

最近更新时间:2023-12-26 11:39:13

网格(mesh)实例,是管理服务的逻辑隔离空间,同一网格内的服务之间可通信。



以下是网格生命周期状态说明:

状态	说明
创建中	网格创建过程中,网格详情无法查看。
运行中	网格正常运行状态。
升级中	网格正在进行版本升级,部分功能无法使用。
闲置	当网格管理的服务发现集群被删除或解关联后,将进入闲置状态(idle),闲置状态的网格可 正常查看,但由于没有业务实体,部分功能将不可用。您可以为网格重新添加服务发现集群 使其恢复正常状态。



无效	当网格处于闲置状态超过 30 天后,或独立网格主集群被删除,网格将进入无效状态,您将不再能对网格进行删除之外的其他操作。
异常	网格中部分组件异常导致功能网格功能已经受到影响。

网格创建过程中需要进行以下几部分配置:

添加服务发现集群

通过添加服务发现的 Kubernetes 集群自动发现集群内的服务实现,也可通过手动注册服务实现。网格中已发现的服务将出现在**服务网格控制台 > 网格详情 > 服务**列表页中。服务被发现后,可以被网格中其他服务访问。详细指引请 查看 服务发现管理。

创建边缘代理网关

边缘代理网关分为 Ingress 和 Egress 两种类型,是网格流量的出入口,其中 Ingress 类型的边缘代理网关必须创建,流量才能进入网格, Egress 类型边缘代理网关可选创建,详细指引请查看 边缘代理网关管理。

为服务注入 Sidecar

Sidecar 容器承担数据面流量管理、规则生效、监控上报等网格治理工作,是网格流量治理和观测的基础,因此对于 需要进行流量管理和观测的服务,需要为其注入 Sidecar,详细指引请查看 网格配置。

可观测性后端服务配置

可观测性分为监控指标查看、链路追踪、日志管理三部分。服务网格支持与腾讯云 Prometheus 监控 TMP、应用性 能观测 APM、日志服务 CLS 集成,提供更丰富、一体化的可观测能力,同时服务网格也支持对接第三方 Prometheus、Jaeger/Zpkin 服务,为用户提供更大的组件扩展性。详细指引请查看 可观测性。 网格创建后,对网格进行流量规则调度,可以通过控制台或提交 YAML 文件配置的形式为网格创建流量治理规则,

当前 TCM 完全兼容 lstio 原生语法,详细指引请查看 流量管理。



创建网格

最近更新时间:2023-12-26 11:43:01

操作场景

使用服务网格,首先需要创建一个服务网格实例。网格实例有地域属性,但可以管理多个地域的服务。 说明:

每个账号默认允许创建 20 个网格,如果需要创建更多个数,可以通过提交工单申请。

操作步骤

以下是新建服务网格实例的控制台操作流程:

1. 登录 服务网格控制台。

2. 选择地域,单击页面左上角的新建。

3. 在创建服务网格页面,按需填写网格创建相关配置,配置项说明请参见创建网格配置项说明,完成后单击**下一步:信息核对**。如下图所示:



		Confirm Infor	mation							
Basic Configurations										
Mesh name *	sample-mesh									
Region	Guangzhou	Shanghai	Hong Kor	ng, China	Beijing	Singapore	Shenzhen F	Finance Sil	con Valley	Cher
	Chongqing	Virginia	Moscow	Tokyo	Nanjing	Tianjin	Shenzhen	Beijing financ	e	
Mesh Mode	Managed Mesi Control plane and a Register Only Allows access to a	h Stand-a related support o Allow Any iny un-registered	alone Mesh components an d address and a	e managed a uddress withc	und maintained	by Tencent Clou overed	d			
Service discovery(i)	Cluster Add C	Cluster								
SideCar auto-injection	- A SideCar will be ir	njected automat	ically to newly o	created Pods	in the selected	l namespace. Fo	or existing Pods,	you need to resta	rt them to inj	ect SideCa
Tencent Cloud tags	+ Add									

4. 在信息核对页面确认创建配置无误后单击**提交**,即可开始网格创建流程。



Mesh Configura	Mesh Configurations > 2 Confirm information								
Basic Configuration	ons								
Mesh name	sample-mesh								
Region	Singapore								
Operation mode	Managed								
Mesh Component Vers	sion Istio 1.12.5								
Service discovery	Cluster perfey-de	mo-勿删(<mark>cls-hz8r3jks</mark>) VPC	: Default-VPC(vpc-c5ynz7i5	2)					
Egress Traffic Mode	ALLOW_ANY								
SideCar auto-injection	base(cls-hz8r3jks) default(cls-hz8r3jks) prom-	-nk02ro8s(cls-hz8r3jks)						
Tencent Cloud tags	-								
Advanced settings									
Edge Gateway									
Ingress Gateway	Name:istio- ingressgateway	namespace:istio- system	Access type:Public network	Load balancer: Automatic creation	Billing mode:Bill-by- traffic	Bandwidth Cap:10Mbps	Pre IP:/		
Egress Gateway	Disabled								
Component deployme	nt mode								
Manageme Service fee	nt fees: The unit price of s: Free for the first 100 Si	a cluster is decars. For the exceeding \$	0.2 Sidecars, the unit price is 0.0	474 CNY/hour 008 CNY/hour					

5. 网格创建流程完成后,即可在服务网格列表页查看服务网格实例。

esh (mesh-0tn3lcxt) sample-mesh				
Task		Status	Start time	En
Check environment resource		⊘ Completed	2022-08-08 20:24:37	20;
Create and initialize the control plane		C Running	2022-08-08 20:24:38	-
Configure service discovery cluster	Connecting cluster cls-hz8r3jks	Waiting	-	-
Create and initialize edge gateway	istio-ingressgateway	Waiting	-	-



创建网格配置项说明

配置项	描述	是否必填
网格名称	创建的服务网格的名称。	是
地域	服务网格控制面运行的地域,控制面运行地域可与业务负载(例如集群)地 域不同,建议就近选择业务负载(集群)所在地域。	是
网格组件版本	选择控制面和数据面的版本,服务网格 TCM 提供支持兼容 lstio 社区的最新两个大版本。	是
网格模式	选择网格控制面相关组件的部署模式,托管网格控制面相关组件由腾讯云管 理和维护,独立网格控制面相关组件会部署在您指定的集群内,您需要管理 和维护集群内的控制面组件,默认可选托管网格,独立网格需要开白名单使 用,您可以通过提交工单申请使用。	是
Egress 流量模式	配置网格内服务对外访问的放通策略,可选择 Registry Only(仅支持访问网格自动发现的服务与手动注册的服务)或 Allow Any(可访问任何地址)。	是
服务发现	指定服务网格自动服务发现的集群,集群需满足版本、权限、网段冲突等约 束条件。	否
Sidecar 自动注 入	配置自动注入 Sidecar 的 Namespace, 开启后将对所选命名空间下的所有 服务负载自动注入 Sidecar, 自动注入仅对新创建的服务负载生效, 存量服 务负载需要重启后才会注入 Sidecar。如果需要进一步自定义 Sidecar 注入 的例外情况,详情见 自定义 Sidecar 注入。	否
外部请求绕过 Sidecar	对应 excludelPRanges,默认情况下,Sidecar 会接管当前 Pod 内所有的流量,如果想让针对特定的 IP 的访问不经过 Sidecar 代理,则可以配置此选项,配置后对该 IP 范围的请求流量将无法使用 Istio 流量管理、可观测性等特性。配置变更后仅对新增 Pod 生效,存量 Pod 需重启后生效。	否
Sidecar 就绪保 障	使用 HoldApplicationUntilProxyStarts 功能,配置业务容器等待 Sidecar 完成启动后再启动,确保业务容器运行依赖 Sidecar 的 Pod 正常运行。	否
Sidecar 停止保 障	开启后 Sidecar 停止需要等待业务容器中进程完全终止,将一定程度增加 Pod 停止时长,建议对业务进程无法随时关闭的服务开启。对于 1.12 之前 的 Istio 版本,TCM 使用预置的容器 prestop 脚本检查不再有业务进程后才 允许业务容器退出,这也意味着用户如果配置其他的 prestop 脚本将对此功 能产生干扰。对于 1.12 之后的版本,使用新特性 EXIT_ON_ZERO_ACTIVE_CONNECTIONS 来实现。	否
自定义 Sidecar 资源	默认情况下,TCM为 Sidecar container 配置最高 2 核 1G 的资源限制,这 在大部分情况下是足够的。当您的网格规模扩大或 Sidecar 中逻辑增多时, 默认资源限制可能不足,您可以自行根据业务需要修改资源限制。	否



Ingress Gateway	为网格创建 Ingress Gateway,如果是 TKE/TKE Serverless集群,则默认创 建 CLB 类型的 Ingress Gateway,您需要配置 CLB 创建相关选项。如果是 注册集群,由于不确定集群是否能使用腾讯云 CLB,因此仅创建 LoadBalancer 类型的 Gateway Service。	否
Egress Gateway	如果您需要对网格的出流量进行集中管理,如,统一出口,统一鉴权,规则 配置等,则需要创建 Egress Gateway。后台将为您创建一个 ClusterIP 类 型的 Egress Gateway service。	否
网关部署模式	可以选择普通部署或专有部署模式,详情见边缘代理网关部署模式。	否
网关伸缩策略	配置部署在指定集群内的边缘代理网关的 HPA 策略。	否
网关资源定义	自定义 Ingress/Egress Gateway 的 pod 资源限制。	否
监控指标消费端	配置网格的监控指标后端服务,目前支持对接 Prometheus 监控 TMP,配置 后监控指标将上报到 TMP,TCM 控制台指标基于 TMP 数据源展示,用户也 可以在 TMP 控制台独立使用。不配置监控指标消费端,网格将无法使用监 控指标展示、拓扑图等相关监控功能。	否
调用追踪消费端	配置网格的调用追踪后端服务,目前支持对接应用性能观测 APM,配置后 Tracing 数据将从 Sidecar 将上报到 APM,TCM 控制台指标基于 APM 数据 源展示,用户也可以在 APM 控制台独立使用。不配置调用追踪消费端,网 格将无法使用调用链查看相关功能。	否
调用追踪采样率	网格采集并持久化调用 Tracing 的采样比例。Sidecar 采集与上报数据消耗资源与带宽和数据量成正相关,请按需配置,开发测试环境建议 100%,生产环境建议 1%。	否
访问日志开启范 围	为避免不必要的开销,TCM 支持针对特定网关或者特定 Namespace 开启 Sidecar 日志。	否
访问日志日志格 式	TCM 支持 Json 或 Txt 两种格式的日志格式化。	否
访问日志输出模 板	Sidecar 日志字段设置,有默认、增强两种预定义模板可选,增强在默认格 式基础上增加了 TracelD 字段输出,如果用户需要进一步更改字段设置,可 按照 Envoy 标准规范 自定义日志字段。	否
访问日志消费端	配置 Sidecar 日志的后端服务,目前支持对接日志服务 CLS,开启后会在集群节点上部署日志采集组件以确保功能正常使用。	否



升级网格

最近更新时间:2023-12-26 11:43:35

TCM 提供网格升级服务, 用户可将低版本的网格升级到高版本。升级过程遵循灰度升级原则, 分为以下几步:

- 1. TCM 部署新版本控制面,完成控制面升级。
- 2. 数据面灰度升级,用户重启业务,使存量业务 Pod 完成 Sidecar 更新。
- 3. 升级验证,验证升级后业务是否正常。
- 4. 老版本控制面下线,升级完成。
- 在老控制面下线前,可以回滚到升级前的状态。升级流程如下图所示:



操作步骤

- 1. 登录 服务网格控制台。
- 2. 当网格版本可升级时, 网格界面将会提示有可升级的新版本, 如下图所示:

ID/Name	Monitor	Status	New component version is available.	Vlesh Mode	Number of ser	Cluster
met and a second se	di	Running	Istio 1.10.3()	Managed Mesh	7	1
Total items: 1						20 💌 / page

3. 选择更多 > 升级后,根据引导进行升级。

升级将按照**控制面升级 > 数据面升级 > 旧控制面下线**三个阶段进行,在旧控制面下线前,可以回滚到升级前的状态。

控制面升级

数据面升级



升级验证

控制面升级, TCM 将部署新版本控制面组件:

Mesh upgrade		
1	2	3
Control plane upgrade	Data plane upgrade	Done
Are you sure you want to upgra	ade the component version of the service mes	h 1?
The current control plane version is Ist plane.	tio 1.10.3. In this step, the canary version Istio 1.12.5 is cre	eated. The next step is to ι
Upgrade Process		
control plane V1	control plane V1 control plane V2	control
000		
I have read the notice above and o	confirmed to upgrade	
	Confirm Cancel	

数据面升级,分为业务数据面升级和 Gateway 升级。

其中业务数据面升级是用户将指定 Namespace 的 Sidecar 自动注入改为使用新版本,勾选后,该 Namespace 下**新** 创建的业务 Pod 将注入新版本的 Sidecar,存量的业务 Pod 重建后才会更新为新版本,由于重启涉及到业务可用性 影响,平台不会自动重建业务 Pod,需要用户手动重建。

说明:

您可以通过流水线重新发布业务或直接使用 kubectl patch、kubectl rollout restart 等命令行工具手工重建工作负载。 部分场景下 Sidecar 将被卸载而不是升级,例如: Namespace 曾开启过 Sidecar 注入,并且部分业务 Pod 已成功注 入 Sidecar,之后又关闭了 Namepsace 级 Sidecar 注入,则重启业务 Pod 后,除非为 Pod 单独设置了 Sidecar 注入 标记,否则 Sidecar 将被卸载。



1		2
Control plane upgrade		Data plane upgrade
 After the switch, the neenable "Sidecar auto-ir The switch does not aff To ensure stability, you 	ew version of control pl njection". fect the existing applic should restart the Poo	ane immediately takes effect to the new Sidecars of the namesp ations on the Pods. You need to restart the Pods to update Sidec Is as soon as possible and verify the health status of the applicat
When it's enabled, the new version data, you need to manually restart	of control plane is use the Pods for Sidecar u Gateway upgra	d automatically for namespaces with "Sidecar auto-injection" en pgrade. de
Pod data plane upgrade		
Pod data plane upgrade	0 1.10.3	0 1.12.5
Pod data plane upgrade Select all default	1.10.31.10.3	1.12.51.12.5

Gateway 升级,将所有需要升级的 Gateway 组件勾选到新版本并单击右侧的升级:



1	2	
Control plane upgrade	Data plane upgrade	
 After all Gateways are the old version, until y After all Gateways are 	upgraded to the new version, you can continue the next step to discontinue the o ou complete the upgrade. rolled back to the old version, you can go back to the previous step to continue r eway, and click to upgrade or roll back:	ontrol p
Pod data plane upgrade	Gateway upgrade	

所有数据面升级完成后,单击**升级**进入下一步。

由于版本升级涉及到特性变更,可能存在兼容性问题,业务 Pod 重建后,您需要检查业务是否正常。如果发现升级导致业务异常,您可以在升级界面中选择回滚,将数据面 Sidecar 退回到原版本。

4. 选择完成或取消升级。在"数据面升级"中,单击升级或回滚将检查当前存量 Pod 是否满足进入下一步的条件。当所有 Namespace 均切换到新版本控制面,并且所有存量业务 Pod 中 Sidecar 均已更新到新版本后,可以选择升级,进入下一步下线旧版本控制面完成升级。或当所有的 Namespace 均切换回旧版本控制面,并且所有存量业务 Pod 中 Sidecar 均使用旧版本控制面时,可以单击回滚进入下一步将新版本控制面下线,取消升级。



Progress of upgrading service mes	sh	
1	2	3
Control plane upgrade	Data plane upgrade	Done
Mesh mesi Initial Istio 1.10.3 is omine	Upgrade completed	
	ОК	



更新网格配置

最近更新时间:2023-12-26 11:44:29

您可以更新运行中服务网格的配置,本文将介绍如何更新网格配置。

修改 Egress 流量模式

Egress 流量模式是配置网格内服务对外访问的放通策略,可选择 Registry Only (仅支持访问网格自动发现的服务与 手动注册的服务)或 Allow Any (可访问任何地址)。

以下是配置网格 Egress 流量模式的步骤:

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

Service me	sh Region Singapore 🔻					
	(i) Tencent Cloud plans to start information, see Billing Overv	charging on Tencent Clou <u>riew</u> IZ .	d Mesh from April 20,	2022 10:00 (UTC +8). Ple	ase delete unused meshes to avoid se	rvice exceptions caused by
	Create					
	ID/Name	Monitor	Status	Version	Mesh Mode	Number of ser
	mesh-0tn3lcxt sample-mesh 🎤	.lı	Running	Istio 1.12.5	Managed Mesh	7
	Total items: 1					

2. 在网格基本信息页面单击 Egress 流量模式栏的编辑按钮,进入调整 Egress 流量模式弹窗。

← Service mesh / me	esh	h)	
Basic information		Basic information	
Mesh topology			
Comies		Name	
Service		Mesh ID	me in
Virtual Service			
		Region	Singapore
Gateway		Mark Oren er et Marian	
Security	-	Mesh Component version	1500 1.12.5
		Mesh Mode	Managed Mesh
Add-On management			
Observability	-	Egress Mode	Allow Any
Observability		Tencent Cloud tags	- /*
		Time created	2022-08-08 20:24:37

3. 按照需要选择 Allow Any 或 Registry Only, 单击保存更新 Egress 流量模式。

Modify Egress mode		×
Egress mode	 Allow Any Registry Only Allows access to any unregistered and service-discovered addresses 	
	Confirm	

启用 HTTP1.0 支持

Istio 默认不支持 HTTP 1.0,如有需要,可以在网格基本信息中开启:



Basic information	The current mesh version is too old and will be out of the maintena <u>Cloud Mesh Version Maintenance</u> 2.
Mesh topology	
Service	
Virtual Service	Basic information
Gateway	Name
Security 💌	Mesh ID
Add-On management	Region Guangzhou
Observability 💌	Mesh Component Version Istio 1.10.3 Canary version: Istio 1.12.5 ()
Resource management	Mesh Mode Managed Mesh
	Egress Traffic Mode 🛈 🛛 Allow Any 🎤
	Tencent Cloud tags - 🖍
	Time created 2023-01-11 16:02:29
	 Advanced settings
	Sidecar configurations
	External request bypasses Sidecar 🛈 🛛 - 🧨
	Sidecar readiness guarantee (i)
	Sidecar Stop Protection 🕄
	Custom Sidecar resources CPU: 0.1 - 2 core; Memory: 128 -
	Features Support HTTP1.0 ()

关闭 HTTP 自动重试

对于失败的 HTTP 请求, Istio 默认会重试2次, 某些情况下这不符合您的预期, 可以在网格基本信息页里关闭自动重试:



Mesh topology Service Virtual Service Gateway Mesh U Security Add-On management Observability Resource management Resource management Egress Traffic Mode() Allow Any * Tencent Cloud tags • Advanced settings Sidecar configurations External request bypasses Sidecar() • Custom Sidecar resource Observation	Basic information		 New version is availa 	ible. Please	upgrade to the new version. Fo
Service Basic information Virtual Service Name Gateway Mesh ID Security * Add-On management Region Observability * Resource management Mesh Mode Gateway Allow Any * Tencent Cloud tags * Time created 2023-01-11 16:02:29 * Advanced settings Sidecar configurations Sidecar readiness guarantee () Sidecar stop Protection () Custom Sidecar resource	Mesh topology				
Virtual Service Name Gateway Mesh ID Security Region Add-On management Mesh Component Version Observability Mesh Mode Resource management Mesh Mode Resource management Egress Traffic Mode① Allow Any * Tencent Cloud tags • Advanced settings Sidecar configurations External request bypasses Sidecar ① Sidecar stop Protection ① Custom Sidecar resources	Service		Basic information		
Gateway Mesh ID Security ■ Add-On management Region Guangzhou Observability ■ Resource management Mesh Mode Managed Mesh Egress Traffic Mode① Allow Any ♪ Tencent Cloud tags - ♪ Time created 2023-01-11 16:02:29 Advanced settings Sidecar configurations External request bypasses Sidecar① - ♪ Sidecar readiness guarantee① ① Sidecar Stop Protection① ①	Virtual Service		Name		
Security Region Guangzhou Add-On management Mesh Component Version Istio 1.12.5 Observability Mesh Mode Managed Mesh Resource management Egress Traffic Mode Allow Any * Tencent Cloud tags -* Time created 2023-01-11 16:02:29 Advanced settings Sidecar configurations External request bypasses Sidecar -* Sidecar Stop Protection O Sidecar Stop Protection O Custom Sidecar resources CPU: 01 - 2 core: Memory:	Gateway		Mesh ID		
Add-On management Observability Resource management Mesh Mode Mesh Mode Egress Traffic Mode① Allow Any * Tencent Cloud tags -* Time created 2023-01-11 16:02:29 * Advanced settings Sidecar configurations External request bypasses Sidecar① Sidecar readiness guarantee① Sidecar Stop Protection① Custom Sidecar resources	Security	•	Region	Guangzł	hou
Observability Resource management Egress Traffic Mode() Allow Any * Tencent Cloud tags -* Time created 2023-01-11 16:02:29 • Advanced settings Sidecar configurations External request bypasses Sidecar() Sidecar readiness guarantee() Sidecar Stop Protection() Custom Sidecar resources CPU: 0.1 - 2 core: Memory:	Add-On management		Mesh Component Version	Istio 1.12	2.5()
Resource management Egress Traffic Mode() Allow Any * Tencent Cloud tags Time created 2023-01-11 16:02:29 Advanced settings Sidecar configurations External request bypasses Sidecar () Sidecar readiness guarantee () Sidecar Stop Protection () Custom Sidecar resources CPU: 0.1 - 2 core: Memory:	Observability	•	Mesh Mode	Manage	d Mesh
Tencent Cloud tags - * Time created 2023-01-11 16:02:29 • Advanced settings Sidecar configurations External request bypasses Sidecar () Sidecar readiness guarantee () Sidecar Stop Protection () Custom Sidecar resources	Resource management		Egress Traffic Mode	Allow Ar	ny 🎤
Time created 2023-01-11 16:02:29 • Advanced settings Sidecar configurations External request bypasses Sidecar () Sidecar readiness guarantee () Sidecar Stop Protection () Custom Sidecar resources			Tencent Cloud tags	- p	
▼ Advanced settings Sidecar configurations External request bypasses Sidecar ① Sidecar readiness guarantee ① Sidecar Stop Protection ① Custom Sidecar resources			Time created	2023-01	-11 16:02:29
Sidecar configurations External request bypasses Sidecar () Sidecar readiness guarantee () Sidecar Stop Protection () Custom Sidecar resources CPU: 0.1 - 2 core: Memory:			 Advanced settings 		
External request bypasses Sidecar () Sidecar readiness guarantee () Sidecar Stop Protection () Custom Sidecar resources CPU: 0.1 - 2 core: Memory:			Sidecar configurations		
Sidecar readiness guarantee Image: Construction Sidecar Stop Protection Image: CPU: 0.1 - 2 core: Memory:			External request bypasses S	Sidecar 🛈	- 17
Sidecar Stop Protection () Custom Sidecar resources CPU: 0.1 - 2 core: Memory:			Sidecar readiness guarante	ei	
Custom Sidecar resources CPU: 0.1 - 2 core: Memory:			Sidecar Stop Protection 🚯		
			Custom Sidecar resources		CPU: 0.1 - 2 core; Memory: 1
Features			Features		
Support HTTP1.0 ()			Support HTTP1.0		
Disable auto-retry of HTTP requests 🛈 💽			Disable auto-retry of HTTP	requests 🕃	

关闭对全网格生效,但您仍然可以对特定的 Virtual Service 显式设定重试策略。

启用 DNS 代理



istio 的 Sidecar 支持 DNS 代理, 启用后, DNS 的流量也会被拦截, 由 Sidecar 直接响应 DNS 请求, 一方面 sidecar 会有 DNS 缓存,可以加速 DNS 响应,另一方面,多集群网格的场景,跨集群访问 service 时,不需要在 client 所在 集群创建同名 service 也能正常解析 service。您可以参考以下步骤开启 DNS 转发。

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

2. 在基本信息页面,单击 DNS Proxying > DNS 转发右侧的





Sasic information	New version is available. Please upgrade to the new version. For deta
Aesh topology	
ervice	Basic information
tual Service	Name
teway	Mesh ID
urity 🔻	Decise Construction
On management	Region Guangznou
vability 💌	Mesh Component Version Istio 1.12.5()
rce management	Mesh Mode Managed Mesh
	Egress Traffic Mode 🛈 🛛 Allow Any 🎤
	Tencent Cloud tags - 🎤
	Time created 2023-01-11 16:02:29
	 Advanced settings
	Sidecar configurations
	External request bypasses Sidecar 🚯 🛛 - 🎤
	Sidecar readiness guarantee 🛈 🛛 💽
	Sidecar Stop Protection 🕢
	Custom Sidecar resources CPU: 0.1 - 2 core; Memory: 128 - 1
	Features
	Support HTTP1.0()
	Disable auto-retry of HTTP requests 🕄 💦
	DNS Proxying DNS forwarding ()
	Auto-allocate IPs 🛈 🦲



如果您需要为没有定义 addresses 的 ServiceEntry 自动分配 IP,可以开启自动 IP 分配。更多内容可参考 Address auto allocation。



Sidecar 注入与配置

最近更新时间:2023-12-26 11:45:21

配置 Sidecar 自动注入

TCM 当前支持在控制台为指定的 Namespace 开启 Sidecar 自动注入,开启后该 Namespace 下新创建的 workload 将会自动安装网格 Sidecar,由于注入是在 workload 创建过程中完成的,因此开启注入无法为已存在的 workload 自动安装 Sidecar,您可以通过重建 workload 完成 Sidecar 自动注入。

以下是配置 Namespaces 级 Sidecar 自动注入的步骤:

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

Service mes	sh Region 🔇 Singapore 🔻					
	Create					
	ID/Name	Monitor	Status	Version	Mesh Mode	Number of ser
	mesi	.lı	Running	Istio 1.12.5	Managed Mesh	7
	Total items: 1					

2. 在服务列表页点击单击Sidecar 自动注入,进入 Sidecar 自动注入配置弹窗。



Service mesh / mesł		
Basic information	Create Monitor	SideCar auto-injection
Mesh topology		
Service	Service name	Туре 🗡
Virtual Service	stock	K8S Service
Gateway	cart	K8S Service
Security	order	K8S Service
Add-On management	frontend	K8S Service
Observability •	product	K8S Service
	user	K8S Service
	kubernetes	K8S Service

3. 按需勾选需要 Sidecar 自动注入的 namespace, 单击确定完成 Sidecar 自动注入配置。



Inject SideCar	for all service LBs in the selected namespace	
Namespace	Select all	
	✓ base	
	✓ default	
Epobling or di	prom-mozroos	~
Enabling or dis Pods. You nee service details	abling "SideCar auto-injection" for the namespace will not affect the existin d to restart them to inject SideCar. The injection results can be checked on page.	g the
Enabling or dia Pods. You nee service details	abling "SideCar auto-injection" for the namespace will not affect the existin d to restart them to inject SideCar. The injection results can be checked on page.	g the
Enabling or dia Pods. You nee service details Service fees: f	abling "SideCar auto-injection" for the namespace will not affect the existin d to restart them to inject SideCar. The injection results can be checked on r page.	g the
Enabling or dia Pods. You nee service details Service fees: f isC	abling "SideCar auto-injection" for the namespace will not affect the existin d to restart them to inject SideCar. The injection results can be checked on r page.	g the

自定义 Sidecar 注入

TCM 也支持您通过编辑 yaml 为特定的工作负载开启 Sidecar 自动注入,如有需要,您可以在 Pod 上添加 label: istio.io/rev:{istio 版本号} (注意 Sidecar 注入相关的标签设置, TCM 与 istio 默认语法略有区 别),示例如下:



apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
labels:
app: nginx
istio.io/rev: 1-14-5
spec:
containers:
<pre>- name: nginx</pre>
<pre>image: nginx:latest</pre>

如果您需要为已经开启了自动注入的 Namespace 下的特定 Pod 添加特例,使其不自动注入 Sidecar,可在 Pod label 中添加: sidecar.istio.io/inject="false"。Pod 级别的注入开关优先级高于 Namespace 级别,关于 Sidecar 自动注入的更多细节,请参考 Istio 文档 安装 sidecar。

不拦截指定网段的流量

如果您不希望拦截某些流量,例如上传文件到 COS 对象存储存储的流量(169.254 开头的内网目标 IP),如果被拦截,且下载的文件较大,则可能导致 Sidecar 内存资源占用很高,因为 Sidecar 会缓存请求内容,出错自动重试时会 复用请求内容。这时,您可以在网格基本信息页中的高级设置中配置"外部请求绕过Sidecar",添加不希望拦截的网段。操作步骤如下:

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

2. 在基本信息页面,单击**外部请求绕过Sidecar**右侧的

。如下图所示:



Basic information Mesh topology	 New version is available 	ble. Please upgrade to the new version. For c
Service	Basic information	
Virtual Service	Name	XXX 🧨
Gateway	Mesh ID	mesh-im743a2z 🗖
Security 💌	Region	Guangzhou
Add-On management	Mesh Component Version	Istio 1.12.5()
Deservability	Mesh Mode	Managed Mesh
Resource management	Egress Traffic Mode	Allow Any 🎤
	Tencent Cloud tags	- /*
	Time created	2023-01-11 16:02:29
	 Advanced settings 	
	External request bypasses S	idecar (j) 🛛 - 🎤

3. 在"编辑外部请求绕过Sidecar"弹窗中,添加您不希望拦截的网段。如下图所示:

External request bypasses Sic	lecar	×
External request bypasses Sidecar	172 . 16 . 0 . 0 / 16	
	Add IP Range	
	Save Cancel	

4. 单击**保存**。



以上是全局配置,如果希望只针对某些工作负载,可以给 Pod 添加注解

traffic.sidecar.istio.io/excludeOutboundIPRanges 。详情可参考注解列表。

apiVersion: apps/v1 kind: Deployment
metadata:
name: nginx
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
'traffic.sidecar.istio.io/excludeOutboundIPRanges': '169.254.0.0/16'
labels:
app: nginx

控制 sidecar 启动顺序

Kubernetes 拉起 Pod 的时候是各个容器同时拉起,使用 istio 的场景,因为流量会被 Sidecar 拦截,当遇到 Sidecar 比业务容器启动更慢的时候,业务容器刚启动时的网络请求将会失败,因为流量被 Sidecar 拦截但 Sidecar 还没启动 就绪。比较常见的场景是:集群规模较大,Sidecar 启动时拉取 xds 较慢导致 Sidecar 启动较慢,而业务进程启动时 要从注册中心拉取配置,由于流量被 Sidecar 拦截而 Sidecar 此时还无法处理流量导致配置拉取失败,然后业务进程 报错退出,导致容器退出。

主要有两种解决方案,第一种是让业务代码更健壮,启动时请求失败要不断重试,直至成功;第二种是让 Sidecar 先 启动,等 Sidecar 就绪后再拉起业务容器。您可以参考以下步骤开启 Sidecar 就绪保障。

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

2. 在基本信息页面,单击 Sidecar 就绪保障右侧的







以上是全局配置,如果只想针对某些工作负载,可以为 Pod 添加如下注解:

proxy.istio.io/config: '{ "holdApplicationUntilProxyStarts" : true }'



apiVersion: apps/v1 kind: Deployment
metadata:
name: nginx
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
<pre>proxy.istio.io/config: '{ "holdApplicationUntilProxyStarts" : true }'</pre>
labels:
app: nginx

Sidecar 优雅终止

业务发版时,工作负载滚动更新,Pod 在终止过程中,Sidecar 默认只等待几秒然后就会强制停止,如果业务请求本 身耗时较长,或者使用长连接,可能会导致部分正常业务请求和连接中断,我们希望 Sidecar 会等待存量的业务请求 和连接结束再退出以实现优雅终止。自 istio 1.12 开始,Sidecar 支持了

EXIT_ON_ZERO_ACTIVE_CONNECTIONS 这个环境变量,作用就是等待 Sidecar "排水" 完成,在响应时,也通知 客户端去关闭长连接(对于 HTTP1 响应 "Connection: close" 这个头,对于 HTTP2 响应 GOAWAY 这个帧)。您可 以参考以下步骤开启 Sidecar 停止保障。

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

2. 在基本信息页面,单击 Sidecar 停止保障 右侧的





Basic information	 New version is available. Please upgrade to the new version. For
Mesh topology	
Service	Basic information
Virtual Service	Name
Gateway	Mesh ID
Security 💌	Region Guangzhou
Add-On management	Mesh Component Version Istio 1.12.5()
Observability 🔻	Mesh Mode Managed Mesh
Resource management	Egress Traffic Mode 🕥 🛛 Allow Any 🎤
	Tencent Cloud tags 🛛 - 🥕
	Time created 2023-01-11 16:02:29
	▼ Advanced settings
	Sidecar configurations
	External request bypasses Sidecar(i) - 🧨
	Sidecar readiness guarantee ()
	Sidecar Stop Protection 🛈

以上是全局启用的方法,通常也建议全局启用,如果只想针对某些工作负载开启,也可以为 Pod 添加如下注解:

proxy.istio.io/config: '{ "proxyMetadata": { "EXIT_ON_ZERO_ACTIVE_CONNECTIONS": "tr



apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
proxy.istio.io/config: '{ "proxyMetadata": { "EXIT_ON_ZERO_ACTIVE_CONNECTION
lapels:
app: nginx

自定义 sidecar 资源

Sidecar 作为 Pod 中的一个 container,也需要设置 request 和 limit,如有需要,也可以自定义,您可以在网格基本信息页的高级设置中可以进行全局自定义。操作步骤如下:

1. 登录 服务网格控制台,单击需要变更配置的网格 ID,进入网格的管理页面。

2. 在基本信息页面,单击自定义 sidecar 资源右侧的

。如下图所示:



Basic information	 New version is available 	ble. Please upgrade to the new version. For
Mesh topology		
Service	Basic information	
Virtual Service	Name	
Gateway	Mesh ID	
Security 🔻	Pasien	Guangzhou
Add-On management	Mark Construction	
Observability 🔻	Mesh Component Version	Istio 1.12.5
Resource management	Mesh Mode	Managed Mesh
	Egress Traffic Mode 🛈	Allow Any 🧨
	Tencent Cloud tags	- /*
	Time created	2023-01-11 16:02:29
	▼ Advanced settings	
	External request bypasses S	idecar (i) 🛛 - 🎤
	Sidecar readiness guarante	e(i)
	Sidecar Stop Protection ()	
	Custom Sidecar resources	CPU: 0.1 - 2 core; Memory: 128

3. 在"自定义 sidecar 资源"弹窗中,编辑自定义资源。如下图所示:



Edit custom Sidecar resources							
Custom Sidecar resources	CPU	request	0.1]-	limit	2	-core
	MEM	request	128	-	limit	1024	MiB
			Save		Ca	incel	

4. 单击**保存**。

如果希望对不同工作负载进行不同的自定义,也可以为 Pod 添加类似如下的注解:

```
sidecar.istio.io/proxyCPU: "0.5"
sidecar.istio.io/proxyCPULimit: '2'
sidecar.istio.io/proxyMemory: "256Mi"
sidecar.istio.io/proxyMemoryLimit: '2Gi'
```

apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
<pre>sidecar.istio.io/proxyCPU: "0.5"</pre>
<pre>sidecar.istio.io/proxyCPULimit: '2'</pre>
sidecar.istio.io/proxyMemory: "256Mi"
<pre>sidecar.istio.io/proxyMemoryLimit: '2Gi'</pre>
labels:
app: nginx

如果使用了 TKE Serverless,不希望因 Sidecar 的 request 和 limit 导致 Pod 规格变高很多,可以用注解只设置 request 不设置 limit, request 根据实际需要填写,填 "0" 就表示 Pod 规格完全不会因 Sidecar 而升配:

```
sidecar.istio.io/proxyCPU: "0"
sidecar.istio.io/proxyMemory: "0"
```





删除网格

最近更新时间:2023-12-26 11:45:42

操作场景

本文档介绍服务网格实例删除的操作方法。

操作步骤

1. 登录 服务网格控制台,进入网格列表页。

2. 页面顶部选择服务网格所属地域。

3. 网格列表中选择待删除的 mesh,单击右侧操作列删除,并确认操作。

Create						Separate keywords with	Q Ø <u>+</u>	
ID/Name	Monitor	Status	Version	Mesh Mode	Number of ser	Cluster	Tencent Cloud	Operation
mesh	di	Running	Istio 1.12.5	Managed Mesh	7	1	-	Delete
Total items: 1						20 💌 / page		/1 page 🕨 🗵


服务发现管理 概述

最近更新时间:2023-12-26 11:46:01

服务发现是将特定的业务服务加入到网格中,是可对服务进行治理和观测的先决条件。TCM 支持自动发现 K8S 集群中的服务,您只需要将集群添加到网格中即可,包括腾讯云提供的TKE、EKS集群,以及注册到TKE的第三方K8S集群。

对于K8S以外的其他服务如 VM、云数据库等,您可以通过配置ServiceEntry、WorkloadGroup、WorkloadEntry将其 手动注册到网格中。



关于如何将 K8S 集群、异构服务添加到 TCM 中,请参见以下文档: 自动服务发现 手动服务注册



自动服务发现

最近更新时间:2023-12-26 11:46:23

操作场景

一个 TCM 网格支持关联多个 TKE 集群,并自动发现其中的 K8S 服务,您可以在创建网格时或在网格基本信息页中 关联多个 TKE 集群,TCM 将自动将集群中的服务展示在**服务**页面中。

使用限制

集群配额:单个网格默认最多支持10个K8S集群,且网格中集群最多跨越3个地域,超过配额后,您无法将更多的集群 添加到网格。

集群版本:TCM 不强制要求集群版本完全相同,但集群版本应符合对于的 Istio 对 K8S 版本的要求,详情请参见 Istio K8S 支持情况。

集群权限:您需要对加入网格的集群有 admin 权限,详见 为集群添加网格权限。

VPC 网络:对于不在同一 VPC 的多个集群,为确保跨集群 Pod 正常访问,需使用 云联网 联通的多个集群,请将集 群添加到同一个云联网实例中。**云联网实例中各端的 VPC 中主机 CIDR 、容器 CIDR 不冲突。**

容器网络:如果 TKE 集群使用 Global Route 网络模式,需要 将容器网络注册至云联网,以便于新添加的容器 CIDR 可被访问。

操作步骤

创建网格页面

您可以在创建网格时配置添加自动服务发现的集群,在网格创建配置页面:

- 1. 登录 服务网格控制台。
- 2. 单击**新建**,创建服务网格。

3. 在基础配置 > 服务发现参数,点击添加集群。如下图所示:



服务团	网格
-----	----

Mesh Configuration	ons > 2	Confirm info	rmation						
Basic Configuration	\$								
Mesh name •	demo								
Region	Guangzhou	Shanghai	Hong Ko	ong, China	Beijing	Singapore	Shenzhen F	inance	
	Chongqing	Virginia	Moscow	Tokyo	Nanjing	Tianjin	Shenzhen	Beijing	g fin
	The region where	the mesh contro	ol plane runs in.	Please select	the region clo	se to the busines	s workload (clust	ter).	
Mesh Component Version	i) Istio 1.10.3	Istio 1.12.5	j -						
Mesh Mode	Managed Mes	h Stand	alone Mesh						
	Control plane and	related support	t components a	re managed a	nd maintained	by Tencent Cloud	d		
Egress Traffic Mode()	Register Only	Allow Any	y						
	Allows access to a	any un-registere	d address and	address witho	ut service disc	overed			
Service discovery	Cluster Add C	Cluster							
SideCar auto-injection									
	A SideCar will be i	njected automa	tically to newly	created Pods	in the selecte	d namespace. Fo	r existing Pods, y	ou need t	to re
Tencent Cloud tags	+ Add								
Advanced settings									
Edge Gateway									
Ingress Gateway 🚯	Enable								
Egress Gateway(i)	Enable								
 Advanced settings 									
Observability config	uration								
Monitoring metrics	Enable								
E	Basic Monitoring - Cloud N	Aonitor Enat	oled 🕕						
	Consumer end	г 🗸	encent Cloud T	MP					
1	A STOUTION ON U								



4. 选择需要添加的自动服务发现 Kubernetes 集群,可同时添加多个集群。提交网格创建请求后,创建的网格实例即可自动发现集群内的 K8S 服务。如下图所示:



Mesh Configurations	> 2	Confirm info	rmation						
Basic Configurations									
Mesh name 🔹	demo								
Region	Guangzhou	Shanghai	Hong Ko	ng, China	Beijing	Singapore	Shenzhen F	inance	
	Chongqing	Virginia	Moscow	Tokyo	Nanjing	Tianjin	Shenzhen	Beijing	fina
	The region where	the mesh contro	l plane runs in.	Please select	the region clo	se to the busines	s workload (clus	ster).	
Mesh Component Version ()	Istio 1.10.3	Istio 1.12.5							
Mesh Mode	Managed Mes	h Stand-	alone Mesh						
	Control plane and	related support	components a	re managed a	nd maintained	by Tencent Clou	d		
Egress Traffic Mode	Register Only	Allow Any							
	Allows access to a	any un-registered	d address and a	address witho	ut service disc	overed			
Service discovery(i)	Cluster Sin	gapore 🔻 Ge	neral cluster 🔻	Please sel	lect a cluster	¢ 0			
	Add	Cluster							
SideCar auto-injection				1					
	A SideCar will be	injected automat	tically to newly	roo-tooo-t ci					.
Tencent Cloud tags	+ Add			j		io, i vi o. Delau	t vi o(vpo coynz	2110/102	
Advanced settings									
Edge Gateway									
Ingress Gateway 🕢 📃 Er	nable								
Egress Gateway 🕙 📃 Er	able								
 Advanced settings 									
Observability configura	tion								
Monitoring metrics 🔽 En	able								
Basio	Monitoring - Cloud M	Monitor Enab	led 🕛						
			_						

网格详情页面



您可以在网格详情页面,查看当前网格实例的服务发现集群,并添加,或解关联自动服务发现集群。

添加服务发现集群

1. 进入网格详情页面,单击侧边栏**基本信息**,在**服务发现**模块,可查看当前网格服务发现的集群列表,单击**添加**进入**添加服务发现集群**弹窗。如下图所示:

- Service mesh / mesi	·o)				
Basic information	Basic information				
Mesh topology	Name	S #*			
Virtual Service	Mesh ID	mest)ar 🔂			
Gateway	Region	Singapore			
Security -	Mesh Component Version	Istio 1.12.5			
Add-On management	Mesh Mode	Managed Mesh			
Observability -	Tencent Cloud tags	- /			
	Advanced settings				
	Service discovery				
	Cluster (1 in tota Add	01-1-1-	Barlan	100	
	ID/Name T	Status	Region	VPC	Associated CC
	General cluster	Running	Singapore	Default-VPC	-
	Total items: 1				

2. 在**添加服务发现集群**弹窗中,选择需要添加的自动服务发现 Kubernetes 集群,可同时添加多个集群,单击**确定**。 如下图所示:



羽格

	ed 0 Cluster(s)	
Singapore *	General cluster	Please select a cluster 🔻 🗘 🙁
Add In a mesh, the Vi the clusters are r container netwo	PC subnet of a mult -c not in the same VPC, a rk to CCN, Also, checl	
container network to CCN. Also, check network in inbound/outbound rules. If cluster in the region of the mesh cortr		
Management fee	es: The unit price of a o	cluster is0.2474 CNY/hour

3. 提交添加服务发现集群后,等待集群连接完成后,即可完成服务发现集群添加。如下图所示:



- Service mesh / mesh	0)				
Basic information	Basic information				
Mesh topology	Name) <i>6</i> °			
Service Virtual Service	Mesh ID	mesh			
Gateway	Region	Singapore			
Security *	Mesh Component Version	Istio 1.12.5			
Add-On management	Mesh Mode	Managed Mesh			
Observability •	Time created	- /			
	Advanced settings				
	Service discovery Cluster (2 in total) Add ID/Name T 2(General cluster 3(c J) General cluster	Status 8) Running Connecting	Region Singapore Singapore	VPC Vpc Default-VPC Vpx i Default-VPC	Associated CC -

注意:

服务添加到网格内后,还需要为服务注入 Sidecar,才能对服务进行流量管理、可视化观测等管理操作,相关指引请参考 网格配置。

解关联服务发现集群

您需要在服务发现集群中解除关联无需参与网格管理或已删除的集群,以避免产生不必要的费用,可以按以下步骤 操作:

注意:

对于已经删除的集群,TCM不会自动帮您解关联,但不会再收取集群管理费用。

如果网格中唯一的集群为已删除状态,为确保网格体验正常,我们将会强制提示您将其解除关联。

1. 进入网格详情页面,点击侧边栏**基本信息**,在**服务发现**模块,可查看当前网格服务发现的集群列表,选择需要解 关联的集群,单击操作栏的**解关联**进入确认解关联弹窗。如下图所示:



Service mesh / m	nesh					
Basic information		Basic information				
Mesh topology		Name				
Virtual Service		Mesh ID	mesh-			
Gateway		Region	Singapore			
Security		Mesh Component Version	Istio 1.12.5			
Add-On management		Mesh Mode	Managed Mesh			
Observability	*	Tencent Cloud tags	- /			
		Time created				
		Advanced settings				
		Service discovery Cluster (2 in total) Add				
		ID/Name ▼	Status	Region	VPC	Associated CC
		O(cis-li 18 General cluster) Running	Singapore	vpc- 5 Default-V⊬C	-
		General cluster	Running	Singapore	vpc-c Default-VPC	-
		Total items: 2				

2. 在**解除集群关联**弹窗中,确认需要解关联的服务发现集群信息,单击**确定**提交解关联集群请求。注意解除集群关 联后,网格不再感知该集群下服务实例变更,相关服务请求可能出现异常。如下图所示:

Disassociation	×
Confirm to disconnect with the cluster (du ()?	
After disconnecting, the mesh can not sense the service instance change of the cluster, a service request may fail	
Confirm Cancel	

^{3.} 等待解关联操作完成即可。如下图所示:



Service mesh / mes					
Basic information	Basic information				
Mesh topology Service	Name				
Virtual Service	Mesh ID	mes r			
Gateway	Region Mesh Component Version	Istio 1.12.5			
Add-On management	Mesh Mode	Managed Mesh			
Observability	Tencent Cloud tags Time created	- /			
	Advanced settings				
	Service discovery Cluster (2 in total) Add ID/Name T	Status	Region	VPC	Associated CCI
	General cluster) Running	Singapore	vpc-c5 Default-VPC	-
) General cluster	Disassociating	Singapore	vpc Default-VPC	-
	Total items: 2				



手动服务注册

最近更新时间:2023-12-26 11:46:47

简介

借助 Istio 的 ServiceEntry、WorkloadGroup、WorkloadEntry 机制,可以在 TCM 中添加非 TKE 提供的集群中的服务,例如传统的 VM 服务、数据库服务等。如果仅需要外部服务可以在网格内能被,则可以以ServiceEntry的形式将外部服务注册到网格,例如云数据库服务。但如果期望在网格内像管理其他自动发现的K8S服务一样治理和观测外部服务,如部署在VM中的应用,则还需要通过WorkloadGroup、WorkloadEntry机制为该应用安装Sidecar。当前TCM 暂不支持此步骤的自动化安装,你需要手工安装,详细指引可参考Virtual Machine Installation。

说明:

概念介绍

ServiceEntry 类似于 K8S 中的 Service 概念,某个服务通过 Service Entry 加入网格后,将可被网格内其他自动发现的服务按照路由规则访问。

WorkloadGroup 类似于于 K8S 中的 Deployment 概念,用于管理 ServiceEntry 的部署。

WorkloadEntry 类似于 K8S 中的 Pod 概念,用于映射一个具体的实体程序。

字段名称	字段 格式	字段说明
spec.hosts	string	服务的 URL 中的 hostname,可以有多个。
spec.ports	Port[]	服务端口号,可以有多个。
spec.resolution	string	Static:使用静态的 endpoint ip 地址作为服务实例。 DNS:通过 DNS 解析服务 endpoint ip 地址,多用于外部服务;申明的 endpoint 需使用 DNS 域名,在无 endpoint 情况下将解析服务为 hosts 域名。 NONE:当服务无需 IP 解析时选择。
spec.location	string	用于标记此服务是否在网格内,部分 lstio 能力特性不能在网格外服务使用,例如 网格外的服务不支持mTLS。MESH_EXTERNAL 代表网格外的服务, MESH_INTERNAL 代表网格内服务。
spec.endpoints	String	服务的接入点,可填写多个,但最终只会同时使用一个。

ServiceEntry 重要字段说明

WorkloadGroup 重要字段说明

字段名称	字段格式	字段说明



spec.metadata.label	string	用户关联 WorkloadEntry 的标签
spec.template	string	用户生成 WorkloadEntry 的基本信息
sepc.probe	string	WorkloadEntry 健康检查相关参数设置

WorkloadEntry 重要字段说明

字段名称	字段格式	字段说明
spec.address	string	当前 endpoint 的地址,类似于 pod IP。
spec.labels	string	当前 endpoint 的标签,用于与 Service Entry 关联。
sepc.serviceAccount	string	sidecar 的权限信息,当需要为 endpoint 添加 sidecar 的时候需要指定。

关于 Service Entry、Workload Entry 的详细介绍,请参见 Service Entry 详细介绍,Workload Entry详细介绍。

手动注册服务

TCM 当前支持在控制台添加 ServiceEntry,以及通过 yaml 的形式添加 ServiceEntry,如果您需要为其他非 K8S 服务安装由于 WorkloadEntry 需要安装 Sidecar 才能正常工作。

YAML 配置示例

控制台配置示例

ServiceEntry

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
   name: external-svc-https
spec:
   hosts:
        api.dropboxapi.com
        api.facebook.com
   location: MESH_EXTERNAL
   ports:
        number: 443
        name: https
        protocol: TLS
   resolution: DNS
```



WorkloadGroup

```
apiVersion: networking.istio.io/v1alpha3
kind: WorkloadGroup
metadata:
 name: reviews
 namespace: bookinfo
spec:
 metadata:
   labels:
      app.kubernetes.io/name: reviews
      app.kubernetes.io/version: "1.3.4"
 template:
   ports:
      grpc: 3550
     http: 8080
    serviceAccount: default
 probe:
    initialDelaySeconds: 5
    timeoutSeconds: 3
    periodSeconds: 4
    successThreshold: 3
    failureThreshold: 3
   httpGet:
    path: /foo/bar
    host: 127.0.0.1
    port: 3100
     scheme: HTTPS
    httpHeaders:
     - name: Lit-Header
       value: Im-The-Best
```

WorkloadEntry

```
apiVersion: networking.istio.io/v1alpha3
kind: WorkloadEntry
metadata:
   name: details-svc
spec:
   serviceAccount: details-legacy
   address: 2.2.2.2
labels:
      app: details-legacy
      instance-id: vm1
```



1. 登录 服务网格控制台。

2. 单击 ID/名称,进入网格详情页面。

3. 单击**服务 > 新建**,填写服务基本信息,即可将非容器化的第三方服务注册到网格服务中,如下图所示:

Туре	Service Entry
Name •	Please enter the service entry nar
	The name is required. It can only contain lower-case letters, numbers, and hyphens ("-"), and must start a lower-case letter or number, and end with a lower-case letter
Namespace	defauit 👻
Hosts *	Please enter the host domain name ×
	Add Host
Service address	Enter service VIP address, separate multiple addresses with carriage returns, CIDR blocks are supported
Location of entry service	O Inside mesh Outside mesh The service is inside the mesh, you can deploy Sidecar to access heterogeneous (such as vm) deployment service and implement communication and control of heterosame as K8s services.
Register DNS	It will created selector-less TKE services with the same host. Please do not modify the corresponding server manually. When registration is enabled, the host address s
	rule.
Service port configuration *	Name Please enter the port name
	Protocol port Please select a protocol 💌 : Range: 1 - 655
	Add Port
Service discovery mode	STATIC DNS NONE Use static endpoint IP address as the service Pod
Endpoints *	Address Please enter the endpoint IP addr Tag key : value ×
	Add labels
	Add Endpoint
Save Cancel	



边缘代理网关 边缘代理网关管理

最近更新时间:2023-12-26 11:47:08

边缘代理网关是负责网格出口与入口流量负载均衡的特殊数据面,它不以 Sidecar 的形式,而是以独立 Pod 的形式 部署在您的集群内,分为 Ingress Gateway 和 Egress Gateway 两种类型。其中,一个 Ingress Gateway 实例包含数 据面的 Envoy Pod 和它关联的负载均衡 CLB 实例(公网或内网),TCM 提供托管的 边缘代理网关控制器,已经实现了 Ingress Gateway 配置与 CLB 的自动化集成,您可以通过 Istio CRD 配置 Ingress Gateway,相关设置 TCM 会自动同步至关联的 CLB 实例,同步的配置包括端口配置和增强功能的端口监听规则配置两部分。即 Envoy 容器和关联的 CLB 作为一个整体,为您提供入口边缘代理网关的能力。

如您需要网格出入口流量负载均衡的能力,您需先要创建 Ingress Gateway 或 Egress Gateway 实例,再通过 Gateway, VirtualService, DestinationRule 等 Istio CRD 配置边缘代理网关的监听规则和流量管理(路由)规则。监 听规则通过 Gateway CRD 配置,流量管理规则通过 VirtualService、Destination Rule 配置(与东西向流量管理语法 一致)。下图是边缘代理网关实例与 Istio CRD 配置的关系示意图。



创建边缘代理网关

1. 登录 服务网格控制台。



2. 您可以在网格创建页面,添加服务发现集群后,创建边缘代理网关。如下图所示:

Service Discovery(i)	Cluster	Guangzhou 🔻 General Cluster 👻 patricklai_test(cls-0877cmyk) VPC: Default-VPC(vpc-c8n8d2xz)	CCN:-▼ 🗘 😮
	,	dd Cluster	
			1. Add service c
SideCar Auto-injection	Namespace		
		Select All	
		asdf	
		default	
		tke-cluster-inspection	
	Inject SideCa	for all service LBs in the selected namespace	
Edge Gateway			
Ingress Gateway 🚯 🔽	Enable	2. Enable ingress gateway	
Ν	ame	istio-ingressgateway	
na	amespace	istio-system 🔻	
A	ccess Type	O Public Network O Private Network	
		Provides internet access capability, specifies a public CLB as the entry point for internet access.	
Lo	oad Balancer	Automatic Creation Use Existing	

或在网格详情页面边缘代理网关 Tab 单击新建创建边缘代理网关。如下图所示:

← Service Mesh /						
Basic Information	Addon Overview					Deployment Mode
Mesh Topology	Version	Istio 1.10.3				Normal Mo
Service	Mesh Mode	Stand-alone Mesh				
Virtual Service	Addon Status	Running:2				
Gateway	Occupied Resource	0.005-core 106.85MB				
Security	Creation/Last Update	2021-09-27 16:54:00 /	2021-12-09 17:08:39			
	Create Monitor	2. create ec	lge gateway			
	Control Plane	Edge Gateway	1. Chose edge g	ateway tab		
	Addon		namespace	Access Type (i)	Num	HPA Policy
	istio-ingressgatev	vay	istio-system	Public Network	1	Trigger policy: CPU Utilization (Request) 80% Number of instances: :1~5

创建边缘代理网关重要配置项说明:



配置项	描述
类型	选择创建 Ingress 网关或 Egress 网关。
接入集群	选择网关创建的 Kubernetes 集群。
namespace	选择网关创建的命名空间。
访问类型	Ingress Gateway 参数。选择 CLB 的访问类型,支持公网和内网。
负载均衡器	Ingress Gateway 参数。选择自动创建负载均衡器,或选择复用现有负载均衡器,复用已有负载均衡器更多介绍参见 Service 使用已有 CLB。
计费模式	Ingress Gateway 参数。选择 CLB 的计费模式,支持按流量计费或按带宽计费,更多关于 CLB 计费的信息,参见 CLB 计费概述。
带宽上限	Ingress Gateway 参数。选择 CLB 的带宽上限, 0 - 2048 Mbps。
CLB 直连 Pod	Ingress Gateway 参数。例如, 网关接入集群网络模式为 VPC-CNI, 即可开启 CLB 直连 Pod, 将不再进行 NodePort 转发, 具有更高性能, 支持保留客户端源 IP、Pod 层级会话保持 与健康检查, 更多详情请参见 使用 LoadBalancer 直连 Pod 模式 Service。
保留客户端源 IP	Ingress Gateway 参数。设置网关 Service 的 ExternalTrafficPolicy 为 Local,源 IP 保留,并开 启 Local 绑定与 Local 加权平衡。开启 CLB 直连 Pod 时,该参数失效。更多详情请参见 Service 后端选择。
组件配置	配置网关的 CPU 和 内存资源,以及 HPA 伸缩策略。

边缘代理部署模式

边缘代理网关有**普通模式**, **专有模式**两种部署形态,其中:

普通模式:网关服务部署在所选的业务集群中,与其他业务Pod无差别部署。

专有模式:某些场景下为了提升服务的稳定性,您需要将边缘代理网关部署在专有节点上。专有模式需要先选择将部分集群节点添加到专有资源池中,网格将会为选中的节点设置污点以保证独享。设置后所有 Ingress/Egress Gateway 都将仅部署在选中的节点中。您还可以在高级设置中为特定的 Gateway 进一步指定节点。您可以在网格创建页、或组件管理页重新调整网关部署模式:

说明:

调整部署模式会触发网关服务调度,可能会对业务流量产生影响。



Gateway Deployment Mode	Normal Mode	Exclusive Mode (Recommended)
	Number of Nodes 🕃	Available nodes in current cluster: 1
	Select Node *	► (interpreted on a formation of the second on a formation of the second on a formation of the second on the second of the second o
	The mesh component is select at least two cross	deployed to the specified exclusive cluster node, which will not be used for application services. M -AZ dedicated nodes to implement the high availability of mesh control plane. Please make sure the

更新边缘代理网关配置

边缘代理网关创建后,您可以修改边缘代理网关的关联 CLB 带宽(仅 Ingress Gateway)、实例数量、HPA 伸缩策 略及资源定义配置。

修改 CLB 带宽

您可以修改 Ingress Gateway 关联的 CLB 实例的带宽。在对应网格详情页面的**基本信息** Tab 边缘代理网关部分可以 编辑 Ingress Gateway 绑定的 CLB 配置。

Adjust the bandwidt	n	×
Billing Mode	Pay-as-you-go- traffic	
Current Bandwidth Cap	10Mbps	
New Bandwidth Cap *	III 0Mees 10 + Mbps 2048Mbps	
Fee	Configuration FeeNetwork Fee0.0286USD/hour(s)0.081USD/GB	
	Submit Close	

修改组件实例数量

在网格详情 > 组件管理可以调整组件的实例数量。



← Service Mesh / martine 17 70	- <u> </u>		
Basic Information	Addon Overview		Deployment Mode
Mesh Topology		Istio 1.10.3	
Service		Stand-alone Mesh	
Virtual Service			
Gateway		0.005-core 104.318MB	
Security 👻		2021-09	
		Modify Number of Instance	×
		Addon istio-ingressgateway	
	Control Plane	Edge G Number of Instances - 1 +	
	Addon	Save Cancel	
		way istio-system Public Network 1	Trigger policy: CPU Utilization (Request) 6 Number of instances: :1~5

修改组件 HPA 伸缩策略

在**网格详情 > 组件管理**可以编辑组件的 HPA 策略。支持按照 CPU、内存、网络、硬盘指标配置伸缩策略。如下图 所示:



← Service Mesh / ┌─			
Basic Information	Addon Overview		Deployment Mode
Mesh Topology		o 1.10.3	Consection
Service		ind-alone Mesh	
Virtual Service			
Gateway	Occupied		
Security -	Edit HPA P	olicy	×
	Name	istio-ingressgateway	
	Trigger Policy	CPU CPU Utilization (Request)	80 % ×
		Add Metric	
	Control		
	Pod Range	1 ~ 5	
		Automatically adjusted within the specified range	
	istic	Save Cancel	

修改组件资源定义

在网格详情 > 组件管理可以编辑组件的资源定义,包括 CPU、内存的 request 和 limit 值。如下图所示:

Service Mesh /							
Basic Information	Addon Overview						Deployment Mode
Mesh Topology		Istio 1.10.3					
Service		Stand-alone Mesh					
Virtual Service							
Gateway		0.005-core 105.666MB					
Security	Creation/L						
Add-On Management	Edit R	esource Definition					×
	Create		istio-ingressgatev	vay			
	Compo	nent Resource Definition	CPU request	0.1	- limit 2	-core	
	Control		MEM request	128	- limit 1024	MiB	
	Adi			Save	Cancel		
	istic	,				Number	of instances of a



删除边缘代理网关

您可以在网格详情-组件管理-边缘代理网关删除指定边缘代理网关。步骤如下:

1. 进入网格详情页,单击**组件管理**,选择**边缘代理网关**,在需要删除的网关对应行**操作**列单击**更多 > 删除**。如下图 所示:

formation Addon O	Overview				Deployment Mode
Version	Istio 1.10.3				
Mesh Mod	le Stand-alone Mesh				
Addon Stat	itus 🔾				
Creation/L:	Resource 0.005-core 105.657 ast Update 2021-09-27 16:54:0	MB 10 / 2021-12-09 17:08:39			
Management					
Create	Monitor				
Control	Plane Edge Gateway				
✓ Add	don	namespace	Access Type (j)	Num	HPA Policy
v istic	o-ingressgateway	istio-system	Public Network	1	Trigger policy: CPU Utilization (Request) 8/ Number of instances: :1~5
Control Adc istic	Plane Edge Gateway don o-ingressgateway	namespace istio-system	Access Type (i) Public Network	Num	HPA Policy Trigger policy: CPU Utilization (Req Number of instances: :1~5

2. 在删除边缘代理网关弹窗,确认删除的网关名称,单击确定完成删除。如下图所示:

	×
Delete Edge Node	
Are you sure you want to delete the edge gateway istio- ingressgateway?	
OK Cancel	



TCM 边缘代理网关控制器实现自动对接 CLB

TCM 实现了托管的边缘代理网关控制器,该控制器会实时监测下发到 Ingress Gateway 的 Gateway 配置,解析当前的端口配置,同步当前端口配置到 CLB,您无需再手动配置 CLB 端口。CLB 端口与 Ingress Gateway Service 端口、Ingress Gateway 容器端口的映射关系是一一映射,即如果在 Gateway CRD 中定义 80 端口,TCM 边缘代理 网关控制器会配置 Ingress Gateway 实例的容器端口为 80,服务端口为 80,以及同步开启关联 CLB 的 80 端口。

TCM 边缘代理网关控制器还实现了 SSL 证书解包上移至 CLB 的功能,实现在 CLB 做证书解包后, Ingress Gateway 再提供流量管理的能力。在 Gateway 中配置证书上移后,边缘代理网关控制器会解析配置上移的端口、域 名、证书,并将配置同步至 Ingress Gateway 绑定的 CLB 实例。





Gateway 配置

```
最近更新时间:2023-12-26 11:47:39
```

边缘代理网关的端口、监听规则通过 Gateway CRD 配置。以下是一个 Gateway 配置的示例,重要字段的解释通过 注释说明:

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
 name: gateway-sample
 namespace: defualt
spec:
 selector: # 根据填写的标签匹配 Gateway 配置下发的 Pod
   istio: ingressgateway
   app: istio-ingressgateway
 servers:
  - port:
     number: 80
     name: http
     protocol: HTTP
   hosts:
   - uk.bookinfo.com
   - eu.bookinfo.com
   tls:
     httpsRedirect: true # 发送 301 https 重定向
  - port:
     number: 443
     name: https-443
     protocol: HTTPS # 开启端口 HTTPS
   hosts:
    - uk.bookinfo.com
    - eu.bookinfo.com
   tls:
     mode: SIMPLE # TLS 单向认证
     serverCertificate: /etc/certs/servercert.pem # 文件挂载方式加载证书
     privateKey: /etc/certs/privatekey.pem
 - port:
     number: 9443
     name: https-9443
     protocol: HTTPS # 开启端口 HTTPS
   hosts:
   - "bookinfo-namespace/*.bookinfo.com"
   tls:
     mode: SIMPLE # TLS 单向认证
     credentialName: bookinfo-secret # 通过 SDS 方式从 Kubernetes secret 加载证书
```



```
- port:
    number: 5443
    name: https-ssl
    protocol: HTTPS # 开启端口 HTTPS
hosts:
    " *"
    tls:
      mode: SIMPLE # TLS 单向认证
      credentialName: qcloud-abcdABCD # 通过 SDS 方式从腾讯云 SSL 平台加载证书 ID 为 abc
- port:
      number: 6443
      name: clb-https-6443-ABCDabcd # 6443启用证书解包上移至 CLB, 使用 ID 为 ABCDabcd K
      protocol: HTTP
hosts:
    - "tcm.tencent.com"
```

Gateway 配置字段说明

以下是 Gateway CRD 重要字段的说明:

字段名称	字段类型	字段说明
metadata.name	string	Gateway 名称
metadata.namespace	string	Gateway 命名空间
spec.selector	<pre>map<string, string=""></string,></pre>	Gateway 使用填写的标签键值对匹配 配置下发的边缘代理网关实例
spec.servers.port.number	uint32	端口
<pre>spec.servers.port.protocol</pre>	string	通信协议,支持: HTTP, HTTPS, GRPC, HTTP2, MONGO, TCP, TLS,请注意同一网关同一端口的 协议配置需要保持一致。
spec.servers.port.name	string	端口名称,当前 TCM 实现了通过端 口名称指定 SSL 证书解包上移至 CLB 的功能,如您需要配置证书上 移,您可以按照 clb-https-{端 口号}-{ssl 平台证书 ID} 的方 式命名,证书上移功能仅在当前端口 通信协议指定为 HTTP 时生效,边缘 代理网关控制器会自动创建 CLB 7层



		监听器实现证书上移,CLB SSL 解 包完成后,CLB 实例与 Ingress Gateway Pod 采用明文通信。请注意 同一网关同一端口的证书上移配置需 要保持一致,否则会引起配置冲突。
spec.severs.hosts	string[]	域名,支持通配符 *
<pre>spec.servers.tls.httpsRedirect</pre>	bool	值为 true 时,边缘代理网关会对 所有 http 请求返回 301 重定向,要 求客户端发起 https 请求
spec.servers.tls.mode	-	配置当前端口的 TLS 安全认证模 式,如需要开启当前端口的安全认证 则需要填写。支 持: PASSTHROUGH, SIMPLE, MUTUAL, AUTO_PASSTHROUGH, ISTIO_MUTUAL
<pre>spec.servers.tls.credentialName</pre>	string	配置发现 TLS 证书密钥的 secret 的 名称,支持从 Ingress Gateway 实例 在同一 namespace 下的 Kubernetes secret 中加载证书与密钥,您需要确 保填写的 secret 中包含合适的证书与 密钥。TCM 还实现了加载腾讯云 SSL 平台证书的功能,按照 qcloud-{ssl 平台证书 ID} 格 式填写本字段,TCM 边缘代理网关 控制器即会为边缘代理网关加载 SSL 平台的证书。当前仅支持从 SSL 平 台加载单向认证 SIMPLE 模式的服务 器证书和私钥
<pre>spec.servers.tls.serverCertificate</pre>	string	设置端口的 TLS 证书密钥通过 file mount 形式(不推荐,推荐采用填写 credentialName 字段加载证书 私钥)挂载时需要填写的证书路径字 段, lstio 默认使用网关所在命名空间 下 istio-ingressgateway-certs secret 加载证书至路径 /etc/istio/ingressgateway- certs
<pre>spec.servers.tls.privateKey</pre>	string	设置端口的 TLS 证书密钥通过 file mount 形式(不推荐,推荐采用填写 credentialName 字段加载证书



		私钥) 挂载时需要填写的私钥路径字 段, lstio 默认使用网关所在命名空间 下 istio-ingressgateway-certs secret 加载私钥至路径 /etc/istio/ingressgateway- certs
<pre>spec.servers.tls.caCertificates</pre>	string	设置端口的 TLS 证书密钥通过 file mount 形式(不推荐,推荐采用填写 credentialName 字段加载证书 私钥)挂载时需要填写的跟证书路径 字段, Istio 默认使用网关所在命名空 间下 istio-ingressgateway-ca-certs 加载根证书至路径 /etc/istio/ingressgateway- ca-certs,双向认证时需要配置 根证书

示例

从 Kubernetes Secret 加载证书至边缘代理网关配置示例

YAML 配置示例

控制台配置示例

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
 name: sample-gw
  namespace: default
spec:
  servers:
    - port:
        number: 443
        name: HTTPS-443-6cph
       protocol: HTTPS
     hosts:
        _ '*'
      tls:
        mode: SIMPLE
        credentialName: {kubernetes secret 名称}
  selector:
    app: istio-ingressgateway
    istio: ingressgateway
```



控制台创建 Gateway 配置 Ingress Gateway HTTPS 协议 SSL 证书从 Kubernetes secret 加载(单向认证)的过程如下:

1. 选择协议为 HTTPS, TLS 模式为 SIMPLE。

2. 证书解包选择边缘代理网关解包。

3. 证书挂载模式选择SDS加载。

4. 证书来源选择 K8S Secret。

5. K8S Secret 选择**选择已有**,选择当前所选边缘代理网关所在 namespace 下的 Secret,请您确保所选 Secret 中包 含合适的证书/私钥/根证书。

TLS Authenticati	on SIMPLE -
Offload Mode	Recommended Terminate at Ingress gateway Terminate at CLB
	The CLB just passes through the traffic to the mesh ingress gateway. SSL/TLS offloading takes place at the ingress gateway. For the same port offload mode must be the same.
Certificate moun	t mode SDS loading File mount
	The gateway dynamically loads the private key, server certificate, and root certificate configuration required for TLS through SDS. Currently, you K8s Secret or SSL certificate Service console.
Certificate Source	e K8S Secret SSL Certificate
K8S Secret	Select Existing Create
Certificate	BEGIN CERTIFICATE MIIDkjCCAnggAwlBAgl, '-' xkKgwDOY''',AQELBQAwZz ELMAKGA1UE BhMCQ04xEJAQ
Private key	BEGIN RSA PRIVATE KEY MIIEpAIBAAKCAQE/ "GVW&~~ 'd+Xg ja2WEqC el6hqmZfHS+pJTa5NO~ JHbnyTo7Dx/JAQWE3c y36z9W+RKI fOFZF/hJkAZ2q1 .CEYx63HOMNDrgqPfA7q71Jm3UYsoD1cnRDP
Add Port	

6. 如当前 Secret 中未有合适证书,您可以选择新建K8S Secret,复制合适的证书/私钥/跟证书内容至对应输入框。



÷	Create gateway		
	Gateway Name *	test-gw	
	Namespace *	default	•
	Specify Ingress gateway •	Туре •	O ingress egress
		Access type •	O Public network
		Ingress (Egress) gateway list •	16) v
		Selector	app: istio-ingressgateway istio: ingressgateway
	Port configuration		
	Port configuration	Protocol port *	TTPS • : - 443 +
		Ple	ease ensure that the port-level configuration for the same port of the same gateway (such as SSL termination configuration) does not conflict.
		Hosts *	
		TLO Authentication	
		TES Autrientication	Recommended
		Offload Mode	Terminate at ingress gateway Terminate at CLB
		The	e CLB just passes through the traffic to the mesh ingress gateway. SSL/TLS offloading takes place at the ingress gateway. For the same port of the load mode must be the same.
		Certificate mount mode	Recommended SDS loading File mount
		Thi K8	e gateway dynamically loads the private key, server certificate, and root certificate configuration required for TLS through SDS. Currently, you can lo s Secret or SSL certificate Service console.
		Certificate Source	K8S Secret OSSL Certificate
		Credential Name	
		lf n	io suitable certificate is found, you can go to the SSL Certificate Service console 🗹 to purchase an SSL certificate.
		Add Port	
	Save Cancel		

从 SSL 平台加载证书至边缘代理网关配置示例

YAML 配置示例

控制台配置示例

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
   name: test-gw
spec:
   servers:
        - port:
        number: 443
```



```
name: HTTPS-443-9ufr
protocol: HTTPS
hosts:
    _ '*'
tls:
    mode: SIMPLE
    credentialName: qcloud-{证书ID}
selector:
    app: istio-ingressgateway
    istio: ingressgateway
```

除了通过 YAML 文件配置,您也可以在控制台上通过 UI 创建 Gateway 配置。以下是配置从 SSL 平台加载证书至边缘代理网关的配置示例,您选择证书来源为 SSL 平台证书即可选择需要加载的 SSL 平台证书。



÷	Create gateway		
	Gateway Name *	test-gw	
	Namespace *	default	▼
	Specify Ingress gateway •	Type •	O ingress ○ egress
		Access type *	Public network Private network
		Ingress (Egress) gateway li	st • [16]
		Selector	app: istio-ingressgateway istio: ingressgateway
	Port configuration	Protocol port *	HTTPS • : - 443 +
		Hosts *	
		TLS Authentication	SIMPLE
		Offload Mode	Recommended Terminate at ingress gateway Terminate at CLB
			The CLB just passes through the traffic to the mesh ingress gateway. SSL/TLS offloading takes place at the ingress gateway. For the same po offload mode must be the same.
		Certificate mount mode	Recommended SDS loading File mount
			The gateway dynamically loads the private key, server certificate, and root certificate configuration required for TLS through SDS. Currently, y K8s Secret or SSL certificate Service console.
		Certificate Source	K8S Secret OSSL Certificate
		Credential Name	uuuuuuu [™]
			If no suitable certificate is found, you can go to the SSL Certificate Service console 🗳 to purchase an SSL certificate.
		Add Port	
	Save Cancel		

SSL 证书解包上移至 CLB 配置示例

YAML 配置示例

控制台配置示例

以下是配置 443 端口证书解包上移至 CLB, 且为该端口启用 SNI, 域名 sample.hosta.org 使用证书1, 域名 sample.hostb.org 使用证书2。

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
   name: test-gw
```



```
spec:
 servers:
   - port:
       number: 443
       name: clb-https-443-{证书ID 1}
       protocol: HTTP
     hosts:
       - sample.hosta.org
    - port:
       number: 443
       name: clb-https-443-{证书ID 2}
       protocol: HTTP
     hosts:
       - sample.hostb.org
 selector:
   app: istio-ingressgateway
   istio: ingressgateway
```

在控制台 UI 创建 Gateway 配置使用证书上移功能流程如下:

1. 选择协议为 HTTPS,出现 TLS 模式选项。

2. 选择 TLS 模式为 SIMPLE。

3. 选择**证书解包**为 CLB 解包,此时端口协议将自动变化为 HTTP(选择证书上移后网关处按照明文 HTTP 处理流量)。

4. 选择合适的**服务器证书**。



Gateway Name *	test-gw	
Namespace *	default	*
Specify Ingress gateway •	Type •	o ingress egress
	Access type •	Public network O Private network
	Ingress (Egress) gateway list	Singapore istio-ingressgateway (
	Selector	app: istio-ingressgateway istio: ingressgateway
Port configuration	Protocol port • HTTP Please e	• - 443 + ensure that the port-level configuration for the same port of the same gateway (such as SSL termination configuration) does not conflict.
	Hosts • samp	le.host.com
	Offload Mode Terr	minate at ingress gateway Terminate at CLB
	SSL/TL: offloadii gateway	S offloading takes place at the CLB bound with the mesh ingress gateway. The traffic is decrypted and passed to the mesh ingress gateway. In this can ng does not occupy the CPU and memory resource of the cluster. Note that this option is only available for SIMPLE authentication mode. For the same y, the offload mode must be the same.
	Server certificate httpbi	ing y 🗘
	If there upload	is no suitable certificate, you can import a certificate from K8s Secretgo to SSL Certificate Service console or go to the SSL Certificate Service console a certificate
	Add Port	

创建成功将跳转至创建完成的 Gateway CRD 详情页面:



Basic informatio	tion					
Name	test-gw					
namespace	default					
Туре	ingress					
Selector	app: istio-ingressgateway, istio: ingressgateway					
Associate gateway	Istio-ingressgateway					
me created						
inte ereeltet						
Port configuratio	ation					
Port configuration	ation					
Port configuratio	ation figuration Protocol Hosts	Transfer security				



流量管理 概述

最近更新时间:2023-12-26 11:48:18

TCM 流量管理模型

TCM 完全兼容 Istio 管理流量的原生 CRD: Gateway, VirtualService 和 DestinationRule,并对原生流量管理语法做 了产品化呈现,下图是 TCM 的流量管理模型:



TCM 使用 Gateway, VirtualService 和 DestinationRule 管理流量。

Gateway:定义网关的端口、监听规则、证书配置, 网关与 Gateway 配置是一对多的关系, Gateway 通过 selector 字段指定配置下发的边缘代理网关。

VirtualService: 定义指定 Host 的路由规则和流量操作规则, VirtualService 通过 hosts 字段指定绑定的域名。可规定 流量来源是边缘代理网关或 mesh 内部。



DestinationRule:定义服务的版本和流量策略,包括负载均衡、健康检查、连接池等流量策略。服务与 DestinationRule 是一对一的绑定关系。

流量管理配置方式

当前 TCM 提供以下两种方式配置 Gateway, VirtualService 和 DestinationRule:

控制台 UI 配置

YAML 创建资源

您可以通过控制台 UI 创建, 删除, 更新, 查看 Gateway, VirtualService, DestinationRule。 创建 Gateway:



创建 VirtualService:





创建 DestinationRule: DestinationRule 与服务是一对一绑定关系,创建与管理在服务详情页面:

← Service mesh / mesl)				
Basic information	Create Monitor	SideCar auto-injection Namespace All	v		
Mesh topology		_			
Service	Service name	Туре т	Namespace	Source	Number of Servi
Virtual Service	stock	K8S Service	base	K8s Cluster 1	1
Gateway	cart	K8S Service	base	K8s Cluster 1	3
Security	order	K8S Service	base	K8s Cluster 1	2
Add-On management	product	K8S Service	base	K8s Cluster 1	1
Observability	user	K8S Service	base	K8s Cluster 1	1
	frontend	K8S Service	base	K8s Cluster 1	1
	kubernetes	K8S Service	default	K8s Cluster 1	0
	Total items: 7				20 🔻 / page


Service mesh / mes / Service:frontend(base)	
Basic information Associate Virtual Service Monitor Call	all trace Security
	Basic information Service name frontend Namespace base Number of Workloads 1
	Port listening protocol () http:80
	Source Workload Region Cluster Y Availability zone Running Pods/Desired Pods
	frontend Singapore General cluster Singapore Zone 3 1/1
	Service Topology Switch to the TPS By service * Last 1 min Last 5 min Last 1 hour 2022-08-08 18:56:10 to 2022-08-08 19:56:10 🛅 🗘
	out: 0.05 rps
	0.05 rps; 100.0% 0.05 rps; 100.0% 0.05 rps; 100.0% frontend fronten
	Q Q I = .
	Destination Rule Create Destination Rule

您可以通过网格管理界面右上角的YAML 创建资源创建 Istio 资源或 Kubernetes 资源。如您提交的 YAML 中含有 Kubernetes 资源,且当前网格管理了多个集群时,您需要选择 YAML 资源提交的目的集群。







VirtualService 配置路由规则

最近更新时间:2023-12-26 11:50:00

VirtualService 定义了指定 hosts 的一系列路由规则和流量操作(权重路由,故障注入等),其中每一条路由规则都 定义了指定流量协议的匹配规则,如流量匹配,则被路由至指定的服务或服务的版本。VirtualService 配置主要包含 以下部分:

hosts:定义路由规则关联的 hosts,可以是带有通配符的 DNS 名称或者 IP 地址。

gateways:定义应用路由规则的来源流量,可以是:

一个或多个网关。

网格内部的 sidecar。

路由规则:定义详细的路由规则,包括HTTP,TLS/HTTPS,TCP 三种协议类型的路由规则。

http:定义一组有序的应用于 HTTP 流量的路由规则。

tcp:定义一组有序的应用于 TCP 流量的路由规则。

tls:定义一组有序的应用于未终止的 TLS 或 HTTPS 流量的路由规则。

VirtualService 重要字段说明

以下是 VirtualService 的重要字段说明:

字段名称	字段类型	字段说明
spec.hosts	string[]	定义路由规则关联一组的 hosts,可以 符的 DNS 名称或者 IP 地址(IP 地址(来源流量为边缘代理网关)。该字段前 HTTP 和 TCP 流量。在 Kubernetes 环 使用 service 的名称作为缩写, Istio 会 VirtualService所在 namespace 补齐缩 default namespace 的 VirtualService 行 写 reviews 会被补齐为 reviews.default.svc.cluster 为避免误配置,推荐填写 host 全称
spec.gateways	string[]	定义应用路由规则的来源流量,可以5 网关,或网格内部的 sidecar,指定方 <gateway namespace="">/<gatewa name>,保留字段 mesh 表示网格 sidecar,当该参数缺省时,会默认填空 即该路由规则的来源流量为网格内部所</gatewa </gateway>
spec.http	HTTPRoute[]	定义一组有序的(优先匹配靠前的路由于 HTTP 流量的路由规则, HTTP 路由



		于网格内部的 service 端口命名为 ht http2-, grpc- 开头的流量以及 gateway 的协议为 HTTP, HTTP2, TLS-Terminated-HTTPS 的流量
spec.http.match	HTTPMatchRequest[]	定义路由的匹配规则列表,单个匹配规 条件是且关系,列表中多个匹配规则之
<pre>spec.http.route</pre>	HTTPRouteDestination[]	定义路由转发目的地列表, 一条 HTTF 是重定向或转发(默认), 转发的目的 个或多个服务(服务版本)。同时也可 重、header 操作等行为
<pre>spec.http.redirect</pre>	HTTPRedirect	定义路由重定向, 一条 HTTP 路由可 或转发(默认), 如规则中指定了 passthrough 选项, route、redi 忽略。可将 HTTP 301 重定向到另外的 Authority
<pre>spec.http.rewrite</pre>	HTTPRewrite	定义重写 HTTP URL 或 Authority head 与重定向同时配置,重写操作会在转发
<pre>spec.http.timeout</pre>	Duration	定义 HTTP 请求的超时时间
spec.http.retries	HTTPRetry	定义 HTTP 请求的重试策略
<pre>spec.http.fault</pre>	HTTPFaultInjection	定义 HTTP 流量的故障注入策略,开标 重试策略不会开启
<pre>spec.http.mirror</pre>	Destination	定义将 HTTP 流量复制到另一个指定的 被复制的流量按照"best effort"原则, s 不会等待复制流量的响应结果就会从测 响应。镜像流量的目的服务端会产生出
<pre>spec.http.mirrorPercent</pre>	uint32	定义流量镜像的复制百分比,缺省时复流量。最大值为100
<pre>spec.http.corsPolicy</pre>	CorsPolicy	定义 CORS 策略(跨域资源共享, Cri Resource Sharing, CORS),更多关 介绍请参见 CORS,关于 Istio CORS 法请参见 CorsPolicy
spec.http.headers	Headers	定义 header 操作规则,包括 request ⁵ header 的更新,增加,移除操作
spec.tcp	TCPRoute[]	定义一组有序的(优先匹配靠前的路由于 TCP 流量的路由规则,该路由规则



		何非 HTTP 和 TLS 的端口
<pre>spec.tcp.match</pre>	L4MatchAttributes[]	定义 TCP 流量路由的匹配规则列表, 则项内所有条件是且关系,列表中多个 间为或关系
spec.tcp.route	RouteDestination[]	定义 TCP 连接转发的目的端
spec.tls	TLSRoute[]	定义一组有序的(优先匹配靠前的路自 于未终止的 TLS 或 HTTPS 流量的路自 路由规则会应用于网格内部的 service https-, tls-开头的流量,来 的端口协议为 HTTPS, TLS 的未终 量, Service Entry 使用 HTTPS, TLS 口。当 https-, tls-端口未关 VirtualService 规则时将会被视为 TCP
spec.tls.match	<pre>TLSMatchAttributes[]</pre>	定义 TLS 流量路由的匹配规则列表, 则项内所有条件是且关系,列表中多个 间为或关系
spec.tls.route	RouteDestination[]	定义连接转发的目的端

配置来自 Gateway 流量(南北向)的路由规则

VirtualService 可通过控制台 UI 和 YAML 编辑两种方式配置。下面将展示将来自 Gateway 流量路由至服务 frontend 的 VirtualService 配置。相关的 Gateway 配置如下:

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: frontend-gw
  namespace: base
spec:
  servers:
    - port:
        number: 80
        name: http
        protocol: HTTP
      hosts:
        _ '*'
  selector:
    app: istio-ingressgateway
    istio: ingressgateway
```



YAML 配置示例

控制台配置示例

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
name: frontend-vs
namespace: base
spec:
hosts:
_ '*'
gateways:
_ base/frontend-gw # VS 挂载的 Gateway 填写按照 {namespace}/{Gateway 名称}
http:
_ route:
_ destination:
_ host: frontend.base.svc.cluster.local # 路由目的地填写 frontend 服务的 host
```

Name *	frontend-vs
Namespace	base v
Associate hosts •	· ©
	Please enter the host, and press Enter to complete
Mount gateway	base/frontend-gw 😒
Routing configuration	
	Condition
	un v exact v
	Add Condition
	Destination * frontend base svc.cluster.local : Please select a version * : Port Weight
	Add Destination
	Advanced settings
	Add route
Save	

配置来自 Mesh 内部流量(东西向)的路由规则



以下是设置将来自网格内部访问 product 服务 host: product.base.svc.cluster.local 的流量的路由规则: 50%的流量路由至 v1 版本, 50%的流量路由至 v2 版本的 VirtualService 配置示例(灰度发布)。其中 product 的服务版本是通过以下 DestinationRule 定义:

```
apiVersion: networking.istio.io/v1alpha3
 kind: DestinationRule
 metadata:
   name: product
   namespace: base
 spec:
   host: product
   subsets:
     - name: v1
       labels:
        version: v1
     - name: v2
       labels:
         version: v2
YAML 配置示例
控制台配置示例
```

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
 name: product-vs
 namespace: base
spec: # 缺省 gateway 参数, 表示该路由配置应用于 mesh 内部 sidecar 的流量
 hosts:
   - "product.base.svc.cluster.local" # 匹配访问该 host 的流量
 http:
   - match:
       - uri:
           exact: /product
     route:
       - destination: # 配置目的端及权重
           host: product.base.svc.cluster.local
           subset: v1
           port:
             number: 7000
         weight: 50
        - destination:
           host: product.base.svc.cluster.local
           subset: v2
           port:
             number: 7000
```



weight: 50

Name *	frontand-us
Namo	
Namespace	base •
Associate hosts *	product.base.svc.cluster.local 🔇
	Please enter the host, and press Enter to complete
Mount gateway	mesh (3)
L	Please enter the gateway, and press Enter to complete
Routing configuration	
	Condition
	uri • exact • /product
	More
	Add Condition
	Destination * product base.svc.cluster.local : v1 v1 7000 50
	product.base.svc.cluster.local : v2 v : 7000 50 3
	Add Destination
	Advanced settings
	Add route



DestinationRule 配置服务版本和流量策略

最近更新时间:2023-12-26 11:50:28

DestinationRule 定义服务的版本和路由发生后的流量策略,包括负载均衡、连接池大小、健康检查(从负载均衡后端中剔除不健康的 hosts)等流量策略。服务与 DestinationRule 是一对一的绑定关系。

DestinationRule 重要字段说明

以下是 DestinationRule 的重要字段说明:

字段名称	字段类型	字段说明
spec.host	string	关联 DestinationRule 配置的 服务名称,可以是自动发现的 服务(例如 Kubernetes service name),或通过 ServiceEntry 声明的 hosts。 如填写的服务名无法在上述源 中找到,则该 DestinationRule 中定义的规则无效
spec.subsets	Subset[]	定义服务的版本 (subsets),版本可通过标 签键值对匹配匹配服务中的 endpoints。可以在 subsets 级 覆盖流量策略配置
spec.trafficPolicy	trafficPolicy	定义流量策略,包括负载均 衡、连接池、健康检查、TLS 策略
<pre>spec.trafficPolicy.loadBalancer</pre>	-	配置负载均衡算法,可配置: 简单负载均衡算法(round robin, least conn, random),一致性哈希(会 话保持,支持按 header name, cookie, IP, query parameter 哈希),地域感知 负载均衡算法
<pre>spec.trafficPolicy.connectionPool</pre>	-	配置与上游服务的连接量,可 设置 TCP/HTTP 的连接池



<pre>spec.trafficPolicy.outlierDetection</pre>	-	配置从负载均衡池中驱逐不健 康的 hosts
<pre>spec.trafficPolicy.tls</pre>	-	连接上游服务的 client 端 TLS 相关配置,与 PeerAuthentication 策略 (server 端 TLS 模式配置) 配合使用
<pre>spec.trafficPolicy.portLevelSettings</pre>	-	配置端口级别的流量策略,端 口级别的流量策略会覆盖服务 / subsets 级别的流量策略配 置

定义服务的版本(subsets)

DestinationRule 可定义服务的版本(subsets),而 subset 则是腾讯云服务网格的最小流量管理单元。例如,您可以 配置将流量路由至某个指定服务的指定 subset。以下是 DestinationRule 定义 product 服务两个 subset 的配置示例。 YAML 配置示例

控制台配置示例

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
   name: product
   namespace: base
spec:
   host: product
   subsets:
        - name: v1
        labels:
        version: v1 # subset v1 通过标签 version:v1 来匹配该服务的endpoint
        - name: v2
        labels:
        version: v2 # subset v2 通过标签 version:v2 来匹配该服务的endpoint
```

DestinationRule 和服务是一一对应的绑定关系,配置 product 服务的 DestinationRule,需要从服务列表页进入 product 服务的详情页,在基本信息 Tab 配置。控制台配置 product 服务两个版本的步骤如下: 1. 在服务列表页面,单击进入 product 服务的详情页面。



- Service mesh / mesl)				
Basic information		Create Monitor	SideCar auto-injection Namespace All	•		
Mesh topology		Service name	Туре Т	Namespace	Source	Number of Serv
Virtual Service		stock	K8S Service	base	K8s Cluster 1	1
Gateway		cart	K8S Service	base	K8s Cluster 1	3
Security		order	K8S Service	base	K8s Cluster 1	2
Add-On management		product	K8S Service	base	K8s Cluster 1	1
Observability *		user	K8S Service	base	K8s Cluster 1	1
		frontend	K8S Service	base	K8s Cluster 1	1
		kubernetes	K8S Service	default	K8s Cluster 1	0
		Total items: 7				20 🔻 / pa
	l.					

2. 在服务详情页**基本信息**,第三个DestinationRule card 区域,单击新建 DestinationRule进入新建弹窗。



Basic Information Associate Virtual Service Monitor Call trace Security Basic Information Fasic Information Fasic Information Fasic Information Basic Information Fasic Information Service name fontend Number of Workloads 1 Number of Service Poids 1 Port listening protocol@ https80 https80	
Basic Information Service name fontend Namespace base Number of Workloads 1 Number of Service Pods 1 Port listenling protocol (2) https80	
Descention flatoring Descention flatoring Service name frontend Namespace base Number of Workloads 1 Number of Service Pods 1 Port listening protocol http:80	
Namespace base Number of Workloads 1 Number of Service Pods 1 Port listening protocol () https80	
Number of Workloads 1 Number of Service Pods 1 Port listening protocol () http:80	
Number of Service Pods 1 Port listening protocol () http://www.inter.org/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/actional/ac	
Port listening protocol () http://	
Source	
Workload Region Cluster T Availability zone Running Pods/Desired Po	ds 1
trontend Singapore General cluster (s) Singapore Zone 3 1/1	
Service Topology	
Switch to the TPS	
By service v Last 1 min Last 5 min Last 1 hour 2022-08-08 18:56:10 to 2022-08-08 19:56:10	
namespace_base	
out: 0.05 rps out: 0.05 rps out: 0.05 rps out: 0.05 rps	: 0.05 r
0.05 rps; 100.0%	
IngressGateway frontend f	ronten
Q. Q. Z	
Destination Rule	
Create Destination Rule	

3. 在弹窗页面为 product 服务添加两个版本,点击保存。



vice version					Add Versio
Service Version 1				Collapse	Delete
Name	v1				
Labels	version	: v1	\odot		
	Add label				
	Labels apply a fil	lter over the endp	oints of a service in the service registry.		
Corresponding workload	product-v	20 A Marco - A			
Service Version 2				Collapse	Delete
Service Version 2	v2			Collapse	Delete
Service Version 2 Name Labels	v2 version	: v2	8	Collapse	Delete
Service Version 2 Name Labels	v2 version Add label	: v2		Collapse	Delete
Service Version 2 Name Labels	v2 version Add label Labels apply a fil	: v2	♥ points of a service in the service registry.	Collapse	Delete
Service Version 2 Name Labels Corresponding workload	v2 version Add label Labels apply a fil	: v2	ooints of a service in the service registry.	Collapse	Delete
Service Version 2 Name Labels Corresponding workload	v2 version Add label Labels apply a fil	: v2 Iter over the endp	points of a service in the service registry.	Collapse	Delete

4. product 服务版本配置完成。



Destination Rule: product			
Service version			
Create			
Name	Тад	Co	orresponding workload
v1	version:v1	pre	oduct-v1
v2	version:v2	-	
Traffic policy Create			
Version range Load	Balancing policy Connection poo	Locality load balance	ing Health Check
		No traffic policy	

配置一致性哈希负载均衡

以下是用 DestinationRule 配置 cart 服务按照 http header name 做一致性哈希负载均衡的配置示例。

YAML 配置示例

控制台配置示例

kind: DestinationRule
metadata:
name: cart
namespace: base
spec:
host: cart
trafficPolicy:
loadBalancer:
consistentHash:
httpHeaderName: UserID # 配置访问 cart 服务的流量按照 header UserID 做一致性哈希



reate Traffic Policy		
Version range *	All versions	
Load Balancing policy	consistentHash httpHeaderName Name UerID	
Connection pool	HTTP Max http1 pending requests - 0 +	
Locality load balancing		
Health check	Advanced settings	
TLS Authentication (j)	Please select	
	Save Cancel	



可观测性 概述

最近更新时间:2023-12-26 11:51:09

腾讯云服务网格提供南北向和东西向的服务间端到端的可观测能力。

观测数据的收集,依赖于注入服务的 envoy sidecar proxy (数据面)上报来完成。您可以通过 TCM 灵活控制数据面 可观测数据的生产计算,TCM 会将观测数据集成到合适的监控周边产品,为您提供网格边缘与内部的服务间流量的 可观测能力。



TCM 提供三种类型的可观测数据:

可观测数据类型	说明
监控指标 Metric	监控指标可为您提供服务或网关的流量观测数据,适合单个服务的开发者关注。
调用追踪 Trace	调用追踪可将一次业务请求的多层调用联系成为一次调用链路,便于您观测调用结构,



	做性能分析与异常定位。
访问日志 Access Log	访问日志是 envoy proxy 生成每次请求的完整记录,包括请求层和 sidecar 代理层的信息,便于运维人员进行访问审计与错误排查。

三种类型的可观测数据信息如下表所示:

观测数据	记录信息	适用场景或角色
Metric	单个服务或网关的流量观测数据,包括但不限于延时、请求数、请求大小等指标,未采样。更多指标信息请参见 lstio Standard Metrics。	单个服务业务开发监测服务运行 状况。
Trace	记录服务间的调用依赖,相比 metric 增加 url 纬度的信息, 但记录的数据一般经过了采样	业务整体开发者,做所有服务的 调用依赖与性能分析。
Access Log	完整记录每次请求的信息,包括 sidecar 代理层丰富的信息 输出,更多信息请参见 Envoy Access Logging。	服务网格运维人员做访问审计与 错误排查。



监控指标 Metric

最近更新时间:2023-12-26 11:51:25

当前腾讯云服务网格可选择使用 Prometheus 监控 TMP 为您提供服务流量 metric 数据的收集、存储与展示。 **说明:**

服务网格将于近期支持使用第三方 Prometheus 服务作为监控后端服务。

服务网格控制台的监控图表将基于存储在 TMP 中的监控指标来展示,如果您有自定义监控的诉求,可以通过 TMP 中的 Grafana 面板设置自定义的监控面板。

操作步骤

服务网格控制台基于 Sidecar 上报到 TMP 的 Metric 数据,提供网格拓扑、服务拓扑、服务监控(请求数、请求状态 码分布、请求耗时、请求大小)图表的展示分析。

开启 TMP 监控

在 创建网格 或网格基本信息页中的可观测性配置 > 监控指标中,勾选**腾讯云 Prometheus 监控 TMP**,按需选择自 动创建或者关联已有 TMP 实例即可。开启后,Sidecar 将会将 metric 数据上报到对应的实例,您也可以在 TMP 控 制台查看该实例。

Call trace	Enable	
	Sampling rate(j)	- 1 + %
	Consumer end	Cloud Monitor ()
		✓ Application Performance Management (APM)
		Instance type Automatic creation Associate Existing
		Instance region Singapore
		External Jaeger/Zipkin service

查看监控相关图表

网格拓扑

记录服务网格所有服务的调用结构,查看网络拓扑须确保相关服务已注入 sidecar,且存在请求流量。查看指定网格的网络拓扑流程如下:

1. 登录 服务网格控制台, 在列表页面单击指定网格 ID, 进入网格详情页面。

2. 单击左侧网格拓扑页签, 即可查看当前服务网格拓扑图。如下图所示:



- Service mesh / mesi	
Basic information Mesh topology	Switch to the TPS Namespace All By service By service Last 1 min Last 5 min Last 1 hour 2022-4
Service Virtual Service	
Gateway	
Security Add-On management	lngressGateway
Observability •	0.3 rps; 100. narrespace_base in: 0.3 rps out: 0.17 rps frontend 0.03 rps; 20.0%
	in: 0.03 rps out: 0.03 rps in: 0.07 rps out: 0.07 rps in: 0.03 r cart order produc 0.03 rps; 100.0% 0.07 rps; 100.0%
	Q Q Image: State of the state of t

3. 单击节点可展示该节点相关的监控详情。如下图所示:



Service mesh / mesh		service Monitor Namespace: base Name: frontend		
Basic information	Switch to the TPS	Real-time	2022-08-07 14:23:22 ~ 2022-08-08 14:23:2	
Mesh topology	Namespace All The By Service The Last 1 min Last 5 min Last 1	Request rate	0.2 rps -	
Service			0.1 rps -	
Virtual Service			Orps	
Gateway			200 206 304 400 40	
Security •				
Add-On management		Request duration	10 ms -	
Observability •			5 ms -	
			0s -	
			avg 200 avg 206 avg 40	
	9:03 rps; 10.7%	Request size	3 KiB -	
	in: 0.03 rps out: 0.03 rps out: 0.02 rp:		2 KiB -	
	cart order		1 KiB -	
	0.03 rps; 100.0% 0.4		0B -	
	in: 0.06 rps		📥 avg 200 💼 avg 206 💼 avg 40	
	stock	Received request size	4 KiB -	
			2 KIB -	
			0B -	
			avg 200 avg 206 avg 40	

4. 界面上方可以选择数据过滤条件,包括 namespace 与时间跨度;支持切换节点的粒度,当前支持 service 粒度和 workload 粒度。如下图所示:



Service mesh / mes)
Basic information	Switch to the TPS
Mesh topology	Namespace All 🔻 Br service 🔻 Last 1 min Last 5 min Last 1 hour 2022-08-08 14:52:51 to 2022-08-08 14:57:51 📩 🕻
Service	
Virtual Service	
Gateway	\sim
Security	out: 0.44 rps
Add-On management	Ingress6ateway 0.44 rps; 100.0%
Observability *	Image control Image control <td< td=""></td<>

服务拓扑

记录某个服务的前后调用依赖关系,查看指定服务的服务拓扑流程如下:

1. 在指定网格的详情页面,单击左侧**服务**进入服务列表页。

2. 单击想要查看的服务,进入服务详情页。



← Service mesh / mesl o)										
Basic information	Create Monitor	SideCar auto-injection Namespace All	Ŧ							
Mesh topology	Service name	Туре Т	Namespace	Source						
Virtual Service	order	K8S Service	base	K8s Cluster 1						
Gateway	product	K8S Service	base	K8s Cluster 1						
Security	frontend	K8S Service	base	K8s Cluster 1						
Add-On management	user	K8S Service	base	K8s Cluster 1						
Observability	cart	K8S Service	base	K8s Cluster 1						
	stock	K8S Service	base	K8s Cluster 1						
	kubernetes	K8S Service	default	K8s Cluster 1						
	Total items: 7									

3. 在服务详情页面的"基本信息"中,即可查看该服务的服务拓扑。如下图所示:



← Service mesh / mesi) / Service:cart(base)						
Basic information Associate Virtual Service Monitor Ca	Il trace Security					
	Basic information					
	Service name	cart				
	Namespace	base				
	Number of Workloads	1				
	Number of Service Pods	3				
	Port listening protocol) http:7000				
	Source					
	Workload	Region	Cluster T		Availability zone	Runnin
	cart	Singapore	perfey General cluster	ks)	Singapore Zone 3	3/3
	Service Topology					
	Switch to the TPS	S				
	By service *	Last 1 min La	ist 5 min Last 1 hour	2022-08-08 18:0	5:10 to 2022-08-08 19:05:10	¢
		out: 0.07 rp	s		→ in: 0.07 rps → out: 0.07 rps	
			0.07 rps;	100.0%		0.07 rps; 1(
		frontend			cart	
	Q. Q. 🗔 🖷	윰				
		-				

服务监控

您可以在服务列表页面对比多个服务的监控数据(请求数、请求耗时、请求大小等),或在服务详情页面查看指定 服务的监控详情。

在服务列表页查看多个服务的监控数据:

1.1 登录 服务网格控制台, 在列表页面单击指定网格 ID, 进入网格详情页面。

1.2 选择**服务 > 监控**,单击需查看监控数据的服务并在右侧查看服务监控数据。如下图所示:



← Service mesh /			3)						Service mo	nitoring				
Basic information		Cr	reate	Monitor	SideCar auto	-injection	Namespace All	Ŧ		Real-time	2022	-08-07 19:46:	:58 ~ 2022-08-08 19:46:5		
Mesh topology										🗹 All(6	n total)		Request rate		
Service			Servio	ce name		Туре т			Names	-			0.4 rps -		
Virtual Service	ual Service					K8S Ser	vice		base	stock base			0.2 rps -		
Gateway			cart			K8S Ser	vice		base	✓ cart base			Orps -		
Security	•		 order 			K8S Ser	vice		base	✓ order			user base		
Add-On management	I-On management		🗸 produ	ct		K8S Ser	vice		base		ct		Average response tir		
Observability	•		✓ user			K8S Ser	vice		base	base		base		40 ms -	
			✓ fronter	nd		K8S Ser	vice		base	✓ user base			20 ms -		
			kuberi	netes		K8S Ser	vice		default	fronte base	nd		0s – user basefr product base		
		1	Total items	: 7									P50 response time		

在服务详情页面查看指定服务的监控数据详情图表:

1.1 在指定网格的详情页面,单击左侧**服务**进入服务列表页。

1.2 单击想要查看的服务,进入服务详情页。

1.3 在服务详情页面的"监控"中即可查看。



rmation Ass	ociate Virtual Ser	rvice	Monitor	Call trace	Security					
Last 1 hour	Last 3 hours	Last 6	hours	2022-08-08 18:47:48	3 to 2022-08-08 19	:47:48 🖬 🤇	Switch	n to the TPS		
Request ra	te (rps)									
0.2										
0.15						٨				
0.1										
0.05				\frown						
18:47	18:51	18:55	18:59	19:03	19:07	19:11	19:15	19:19	19:23	19:
Request du	iration (ms)						— 200			
Request du	iration (ms)						200			
Request du	iration (ms)						200			
Request du 30 20	iration (ms)						- 200			
Request du 30 20 10	iration (ms)						- 200			
Request do	iration (ms)						- 200	3		
Request da 30 20 10 19:03	iration (ms)							3		
Request de 30 20 10 19:03	iration (ms)						— 200 19:13 — avg (200	3		
Request du 30 20 10 19:03	iration (ms)						— 200 19:13 — avg (200	3		

关闭监控

您可以在**网格基本信息页**选择编辑可观测性配置,取消勾选**腾讯云 Prometheus 监控 TMP**,取消勾选后,服务网格 侧并不会删除 TMP 实例,如有需要,请在 TMP 控制台 进一步删除该 TMP 实例。



服务网格



调用追踪 Trace

最近更新时间:2023-12-26 14:16:50

服务网格默认集成应用性能观测 APM 作为调用追踪消费端,开启后网格将会为您创建一个 APM 实例并将 tracing 数据上报到对应的 APM 实例,您可以在服务网格控制台查看到请求在网格内的完整调用瀑布图和每层调用的 trace 日志信息,帮助您理解服务的调用依赖,以及做网格内的延迟分析。您也可以直接在 APM 控制台查看调用相关数据。除了 APM 之外,网格支持将调用数据上报到第三方 Jaeger/Zpkin 服务,如果开启使用第三方 tracing 服务,则服务 网格控制台将无法展示调用追踪相关信息,需要在第三方服务中查看。

调用追踪数据通过 sidecar 收集上报, sidecar 会自动生成 trace span。如果您需要查看完整调用链信息, 需要在业务代码中做少量改造, 传递请求上下文, 以便 TCM 可以正确关联入站和出站的 span, 形成完整调用链路。需要业务传递的 headers 包括:

x-request-id

x-b3-traceid

x-b3-spanid

x-b3-parentspanid

x-b3-sampled

x-b3-flags

x-ot-span-context

更多关于 Envoy-based tracing 的信息,请参见 Istio Distributed Tracing FAQ。

查看调用追踪

查看调用追踪的流程如下:

1. 需要关注网格某个服务的调用情况,在网格的服务列表页面单击该服务,进入服务详情页面。



- Service mesh / mest	(c					Create via YA	AML
Basic information	Create Monitor	SideCar auto-injection Names;	ace All 🔻			Search by name Q	φ±
Mesh topology	Service name	Туре Т	Namespace	Source	Number of Se	Operation	
Virtual Service	order	K8S Service	base	K8s 1 Cluster	2		
Gateway	stock	K8S Service	base	K8s 1 Cluster	1	-	
Add-On management	product	K8S Service	base	K8s 1	1		
Observability •	fronteord	K8S Service	hase	K8s 1			
			Dase	Cluster			
	user	K8S Service	base	Cluster	1	-	
	cart	K8S Service	base	K8s 1 Cluster	3		
	kubernetes	K8S Service	default	K8s 1 Cluster	0		
	Total items: 7				20 🔻 / page 🛛 H	4 1 /1 page → →	r

2. 在服务详情页面单击**调用追踪**,您可以查看到该服务作为被调方,被调用的记录列表,以及这些调用记录的耗时 分布统计直方图。



- Service mesh / mesh / Service:stock	(base)				
Basic information Associate Virtual Service Mo	nitor Call trace Security	1			
ew the legacy page					
		Our las			
Last 1 min Last 5 min Last 1 hour 2022-08	-04 14:17:23 to 2022-08-04 15:17:23	Service stock *			
Time consumption					
4					
3					
2					
1					
<1ms 5ms 10ms 50ms	200ms 400ms	600m	s 800ms	1s 1.5s	2s 3s >3s
5 Compare with yesterday ↑ 100% Compare with last week	† 100%	0 Compa	are with yesterday 0% Compare wi	th last week 0%	
Call history					
View Server Client Duration (ms) 0	30000 Return code	IP	Protoco	I	
TraceID	,				
TraceID/SpanID	Service	Return code	Collection time \$	Call	Response time 🕈
990f6d173748fee33d36ea7b1bf386ef 0deec0dcf3dbe044	stock	200	2022-08-04 15:15:57	/stock	
					1ms
7a7f8fedf2e9fc2eb5173fff16be2761	stock	200	2022-08-04 15:15:55	/stock	1ms
7a7f8fedf2e9fc2eb5173fff16be2761 99c9003fc6fe6951	stock	200	2022-08-04 15:15:55	/stock	1ms 1ms
7a7f8fedf2e9fc2ab5173fff16be2761 99c9003fc6fe6951 99b485cfd188f3d56ee7996fc2cacb12 3db0f951ba13362f	stock	200 200	2022-08-04 15:15:55 2022-08-04 15:15:05	/stock /stock	1ms 1ms 1ms

3. 被调记录列表第一列记录了调用的 URL,单击即可查看该调用相关的完整调用链路瀑布图,上方瀑布图概览可以 拖拽实现缩放。





i - informations	Associate Matural Consist	Manitan	0-11 4	Comunity.				
sic information	Associate virtual Service	Monitor	Call trace	Security				
the legacy page								
frontend.ba	se.svc.cluster.local:80/*							
Duration:17 ms De	epth: 6 Number of Spans: 6							
0 ms	2 ma	4 ms	6 ms	8 ma	10 ms	12 ms	14 ms	16 ms
Call								
Call								
Call sitio-ingressga	leway: /order							
Call istio-ingressga frontend: //	teway: /order							
Call Call fistio-ingressga frontend: //	teway: /order arder d: /order							
Call Istlo-ingressga frontend: //	teway: /order arder d: /order er /order							
Call tistlo-ingressga frontend: // frontend: // frontend: // frontend: //	teway: /order arder d: /order er: /order							
Call istlo-ingressga istlo-ingressga if frontend: /c if frontend if ord	teway: /order arder d: /order er: /order order: /stock							
Call istlo-ingressga frontend: // frontend: // ord	teway: /order order d: /order er: /order order: /stock stock: /stock							

4. 单击想要查看详情的调用,可以查看对应调用环节的详细 trace 日志。



- Service mesh / mesi	10) / Service	e:stock(base)						
Basic information	Associate Virtual Service	Monitor	Call trace	Security				
liew the legacy page								
← frontend.base	e.svc.cluster.local:80/*							
Duration:17 ms Dep	th: 6 Number of Spans: 6							
=								
0 ms	2 ms	4 ms	6 ms	8 ms	10 ms	12 ms	14 ms	16 ms
Call								
CI	ICK to show details			Details				Ø
frontend: /or	der			Operation	frontend.base.svc.cluster.local:8	0/*		
frontend:	/order			Reporter Service	istio-ingressgateway			
▼ order	: /order			Destination Service	frontend.base			
• 01	stock: /stock			Cluster ID	cls-lumxset8			
				Process IP	172.23.0.67			

5. 单击关闭按钮可关闭 span 详情,以及返回服务被调记录列表页。

information	Associate Virtual Service	Monitor	Call trace	Security				
s logacy page								
frontend.ba	se.svc.cluster.local:80/*							
uration:17 ms De	epth: 6 Number of Spans: 6							
					Ī			
0 ms	2 ms	4 ms	6 ms	8 ms	10 ms	12 ms	14 ms	16 ms
0 ms	2 ms	4 ms	6 ms	8 ms	10 ms	12 ms	14 ms	16 ms
0 ms	2 ms	4 ms	6 ms	i 8 ms	10 ms	12 ms	14 ms	16 ms
0 ms	י 2 ms	4 ms	6 ms	8 ms	10 ms	12 ms	14 ms	16 ms
0 ms ;all r istio-ingressga	2 ms	4 ms	6 ms	8 ms	10 ms	12 ms	14 ms	16 mə
0 ms	2 ms	4 ms	6 ms	8 ms Details Operation	10 ms	12 ms	14 ma	16 ms
o ms all sall formation of the second	2 ms 2 ms teway: /order order d: /order	4 ms	ê ma	8 ms Details Operation Reporter Service	10 ms	12 ms	14 ma	16 ms
0 ms	2 ms 2 ms tteway: /order dt /order dt /order ter: /order	4 ms	6 ms	8 ms Details Operation Reporter Service	10 ms	12 ms	14 ma	16 ms
o ms iall istio-ingressga frontend: // frontend: // ord	2 ms 2 ms teway: /order order d: /order er: /order order.	4 ms	6 ms	8 ms	10 ms frontend.base.svc.cluster.local:80/ istio-ingressgateway frontend.base	12 ms	14 ma	16 ms
0 ms all fistio-ingressga frontend: // fronten ord ord v	2 ms 2 ms teway: /order order d: /order er: /order order: /stock stock: /stock	4 ms	6 ms	8 ms	10 ms 10 ms frontend.base.svc.cluster.local:80/ istio-ingressgateway frontend.base cls-lumxset8	12 ms	14 ma	16 ms

6. 查询服务被调记录 Tips:您可以按照耗时、时间跨度、源端IP、Trace ID、返回码过滤被调记录,过滤完成后您可以按照**延时**和**开始时间**对调用记录排序,方便您选择查看需要关注的调用。



Last 1 min Last 5 min Last 1 hour 2	022-08-04 14:44:11 to 2022-08-0	04 15:44:11 Service stock 🔻			
Time consumption					
4 3 2 1					
<1ms 5ms 10ms 50ms	200ms	400ms 600ms	800ms	1s 1.5s	2s 3s >3s
Compare with yesterday 🕇 100% Compare with last Call history View Server Client Duration (ms) TraceID	week † 100%	Compare le IP	with yesterday 0% Compare with yesterday 0% Protoc	ol	
TraceID/SpanID	Service	Return code	Collection time	Call	Response time 🗘
990f6d173748fee33d36ea7b1bf386ef 0deec0dcf3dbe044	stock	200	2022-08-04 15:15:57	/stock	1ms
7a7f8fedf2e9fc2eb5173fff16be2761 99c9003fc6fe6951	stock	200	2022-08-04 15:15:55	/stock	1ms
99b485cfd189f3d56ee7996fc2cacb12 3db0f951ba13362f	stock	200	2022-08-04 15:15:05	/stock	1ms
1152b3372153799b3b16904e7cb2907c 87367d572f149561	stock	200	2022-08-04 15:15:03	/stock	1ms
67e5178040fabad3d3df957b3832f69f ae34cdac08102b45	stock	200	2022-08-04 15:14:57	/stock	11ms
Total items: 5				20 💌 / page	I /1 page ▶

配置调用追踪采样率

调用追踪采样率是 trace 数据的采样比例, sidecar 采集与上报数据消耗资源与带宽和采样率成正相关。通常生产环境下,不需要为所有调用都生成 trace 数据并采集上报,避免过多消耗计算资源与带宽资源,只需要配置一定比例即可。建议开发测试环境可以配置100%采样率,生产环境配置1%采样率。 您可以在网格创建时配置采样率,如下图所示:



Observability co	nfiguration							
Monitoring metrics	Cinable							
	Basic Monitoring - Cloud Monitor	Enabled ()						
	Consumer end	Cent Cloud TMP						
		Monitoring data is stored in TMP. You can check and query them in the Tencent Cloud Mesh console. Preset Grafana can be used to configure custom monitoring metrics.						
		TMP instance Automatic creation Associate Existing						
		Automatically create a TMP instance in the region where the mesh is located. The original username for Grafana is admin, and the original password is meshadmin.						
		Instance network Singapore Vo data yet Vo data yet						
Call trace	Enable							
	Sampling rate() - 1	+ %						
	Consumer end Cloud M	Nonitor 🕕						
	Applicat	tion Performance Management (APM)						
	External	I Jaeger/Zipkin service						
Access logging	Enable							
	It is recommended to use the contain	ner standard log output path and Tencent Cloud Mesh output template for features like access logging. You can also customize the configuration file after disabiling the						

也可以在网格创建后,在网格基本信息页面修改采样率配置。如下图所示:

Service mesh / mesh	.10)			Create via YAML						
Basic Information	Singapore istio-ing	essgateway 6)	•							
Mesh topology	Egress Gateway (0 in t	ess Gateway (0 in total) Create Now								
Service										
Virtual Service	Monitoring metric	3								
Gateway	Consumer end	Basic Monitoring - Cloud Monitor En	nabled 🕕							
Security *		TMP En	nabled 🧨							
Add-On management			Associated instance mest 190034 (p 2)							
Observability *		· · · · · · · · · · · · · · · · · · ·	Grafana Access Address Public network:https://cloud-grafana-intl.woa.com/grafarelity/ [2]; Private network:http://172.22.0.33 [2]							
	Call trace									
	Sampling rate	100%								
	Consumer end	Cloud Monitor	Enabled 🕕							
		Application Performance Management	(APM) apm-l .x ≠*							
		External Jaeger/Zipkin service	Disabled 💉							



访问日志 Access Log

最近更新时间:2023-12-26 14:17:26

您可以配置服务网格数据面访问日志 Access Log 输出(容器标准输出)的开启范围,输出格式,以及开启 Access Log 日志自动采集对接到云日志服务产品 CLS 的日志集-日志主题。您可以在创建网格时配置访问日志,网格创建后您也可以在基本信息页面修改访问日志配置。

访问日志配置

当前支持的访问日志配置如下表:

配置项	描述
开启范 围	配置开启访问日志输出的数据面(边缘代理网关和 istio-proxy sidecar),可以开启指定边缘代理网关、指定 namespace 下所有数据面、或网格所有数据面的访问日志到容器标准输出。
输出格 式	配置访问日志输出的字段和格式模板,默认格式输出的字段为 lstio 默认输出的字段,增强格式在默认格式基础上增加了 Trace ID 输出。
消费端	配置将数据面容器标准输出的访问日志采集到日志服务 CLS。需要选择存储访问日志的 CLS 日志 集与日志主题,可以选择自动创建日志集/主题,或关联已有的日志集/主题。自动创建的日志集命 名规则为 {mesh ID},自动创建的日志主题带有 TCM 标识,命名规则为 {mesh ID}- accesslog。开启访问日志采集到 CLS 提交后,会开启网格管理集群的日志采集功能,在网格 管理的集群中部署日志采集组件 tke-log-agent (DaemonSet),并配置 TCM 访问日志的采集规则与 索引。该功能是基于容器服务的 日志采集功能,请确保已开通 日志服务 CLS,并且容器服务的服 务角色 TKE_QCSRole 已关联日志服务运维管理的预设策 略。 QcloudAccessForTKERoleInOpsManagement ,更多说明请参见 容器服务角色权限说 明。

创建网格时配置 Access Log:



Access logging	Enable						
	It is recommended to u access logging (only fo	ise the container standard r stand-alone mesh).	l log output path and Tencer	nt Cloud Mesh output tem	plate for features like access logg	ing. You ca	
	Range	All Select Range					
	Log format	Json Text					
	Output template	O Istio Format	Trace Format Oustom				
		The output fields are the	e default output fields of Isti	io. View Sample			
	Consumer end	Tencent Cloud CLS					
		To deploy the log collecting add-on "tke-log-agent (DaemonSet)" in a mesh-managed cluster, please reserve at least 0.1 c add-on "cls-provisioner(Deployment)" will be deployed in the EKS cluster "kube-sysytem(namespace)". The Pod specifica collection-related permissions to the EKS cluster. The logs will be collected and reported to Tencent Cloud CLS, and you). The logs can only be uploaded to the log topics in the same region. If the mesh includes cross-region clusters, only the control plane will be collected to CLS.					
		Logset	Automatic creation	Associate Existing			
		Log topic	Automatic creation	Associate Existing			

网格创建完成后配置 Access Log:



- Service mesh / me	st	0)						
Basic information		Singapore istio-ing	gressgateway ô)			Ŧ		
Mesh topology		Egress Gateway (0 in	total) Create Now					
Service								
Virtual Service		Monitoring metric	CS					
Gateway		Consumer end	Basic Monitoring - Cloud Monitor	Enabled (D			
Security	•		TMP	Enabled	r			
Add-On management				Associat	ted instance	mesh-kle5d0	ar-	4 (prom 2
Observability	•			Grafana	Access Address	Public netwo network:http	rk:https://cloud-gi ://172.22.0.33 🔼	rafana-intl.woa.com
		Call trace						
		Sampling rate	100% 🖋					
		Consumer end	Cloud Monitor		Enabled 🕛			
			Application Performance Managem	ent (APM)	apm	d'		
			External Jaeger/Zipkin service		Disabled 🧨			
		Access logging						
		Range	All					
		Log format	Json					
		Output template(i)	Istio Format					
		Consumer end	Logset mesh	Ľ				
			Log topic mesi) 🖬			

访问日志查看

通过容器标准输出查看

TCM 的数据面访问日志 Access Log 是输出到容器标准输出,您可以通过您的 Kubernetes 集群 API Server 查看 istio-proxy 容器标准输出的访问日志:

kubectl -n {命名空间} logs {Pod 名称} -c istio-proxy --tail 5



通过日志服务 CLS 日志检索查看

如您开启了访问日志的消费端配置,将 TCM 数据面访问日志 Access Log 采集到了日志服务 CLS,则您可以在 CLS 控制台检索分析处选择对应日志主题查看 TCM 数据面访问日志。CLS 日志检索语法,请参见 CLS 日志检索语法与规则。

Search and Analysis Singapo	re(11) 11 log topics 🔻 Logset met	▼ Log Topic mes − Ccesslog ▼ I Il Monitoring Statistics
Index Configuration Preferences Share Create Data Processing Task		
1 response_code:200		
+ Add Filter Condition		
Possible syntax errors auto-corrected. You can click to disable auto correction.		
Raw Data Chart Analysis	_	
Search Q	Log Count 91	
Showed Field	40	
Raw logs	20	
Hidden Field	17:22:30 17:24:00	17:25:30 17:27:00 17:28:30 17:30:00 17:31:30 1
	Lin Log Time ↓	Raw logs
t _HOSTNAME_	▶ 1 08-08 17:37:18.011 -0	response code: 200 method: GET route name: default downstream remote address:
#PKG_LOGID		eived: 0 upstream_service_time: 1 bytes_sent: 939 istio_policy_status: null x_1
t _CONTENT_		 3 downstream_local_address: 172.16.0.132:80 upstream_local_address: 127.0.0.6:35
t downstream_remote		t_failure_reason: null request_id: dc6af392-83e2-917f-9f7a-d27c4306a6dc user_age 103.0) Gecko/20100101 Firefox/103.0
t path	▶ 2 08-08 17:37:18.011 EQ	response_code: 200 method: GET route_name: default downstream_remote_address: 4
t upstream_local_add		eived: 0 upstream_service_time: 1 bytes_sent: 939 istio_policy_status: null x_1 gs: - path: /product start_time: 2022-08-08T09:37:16.405Z protocol: HTTP/1.1 ut
t request id		er.local authority: product.base.svc.cluster.local:7000 downstream_local_address
	l	USAL 40034 Upstream_nost. 1/2.10.0.5:/000 Upstream_transport_Tallure_reason: NU


安全

Authentication 认证策略配置

最近更新时间:2023-12-26 14:18:16

认证策略包含 PeerAuthentication 和 RequestAuthentication。其中, PeerAuthentication 策略用于配置服务通信的 mTLS 模式, RequestAuthentication 策略用于配置服务的请求身份验证方法。

PeerAuthentication 配置字段说明

以下是 PeerAuthentication 重要字段说明:

字段名称	字段类型	字段说明
metadata.name	string	PeerAuthentication 名称
metadata.namespace	string	PeerAuthentication 命名空间
spec.selector	<pre>map<string, string></string, </pre>	PeerAuthentication 使用填写的标签键值对,配合填写的 namespace,匹配配置下发的 Workload 范围: namespace 填写 istio-system,且 selector 字段不填写时, 该策略生效范围为整个网格。 namespace 填写非 istio-system 的 namespace,且 selector 字段不填写时,策略生效范围为填写的 namespace。 namespace。 namespace 填写非 istio-system 的 namespace,且 selector 字段填写了有效键值对时,策略的生效范围为在所 填 namespace 下根据 selector 匹配到的 Workload。
spec.mtls.mode	-	配置 mTLS 的模式,支持: UNSET
<pre>spec.portLevelMtls</pre>	<pre>map<uint32, mode="" mtls=""></uint32,></pre>	设置端口级别的 mTLS 模式

说明:

mTLS 模式配置,不同选择范围的生效效力为:端口 > 服务/Workload > namespace > 网格。

使用 PeerAuthentication 配置网格内服务通信 mTLS 模式



服务网格默认网格内 mTLS 模式为 PERMISSIVE,即服务间的通信既可以使用 mTLS 加密,也可以使用 plaintext 明 文连接。

为测试 mTLS 模式配置的效果,您可以首先对您网格内的服务发起明文请求,测试明文请求的连通性。以下是登录 网格内 istio-proxy 容器对另外的服务发起明文请求的示例:

1. 在网格管理的 TKE 集群控制台,登录 istio-proxy 容器。

is pod has 2 containers			
Container Name	Status	Operation	
cart	Running	Log In	
istio-proxy	Running	Log In	

2. 输入命令 curl http://product.base.svc.cluster.local:7000/product 明文访问命名空间 base 下的 product 服务。

3. 查看明文访问结果,正确返回了 Product 信息,明文访问成功。



下面我们将会配置 base namespace 的 mTLS 模式为 STRICT,并验证配置是否生效。

YAML 配置示例

控制台配置示例



```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
   name: base-strict
   namespace: base
spec:
   mtls:
    mode: STRICT
```

Policy Name •	Please enter the policy name.
Policy Type •	PeerAuthentication RequestAuthentication Configure the mTLS mode of service communication
Namespace *	base •
Specify Service/Gateway Method	Select Service By labels
Service/Gateway	all v all v
selector	N/A
Policy Content	Mode DISABLE PERMISSIVE OSTRICT UNSET Connection is encrypted with mTLS (TLS with client certificate is required)

配置完成后,重新以明文的方式访问 base 命名空间下的 product 服务,提示访问失败,mTLS STRICT 模式配置生效。



RequestAuthentication 配置字段说明

以下是 RequestAuthentication 重要配置字段说明:



字段名称	字段类型	字段说明
metadata.name	string	RequestAuthentication 名称
metadata.namespace	string	RequestAuthentication 命名空
spec.selector	<pre>map<string, string></string, </pre>	RequestAuthentication 使用填 填写的namespace,匹配配置 围,namespace填写istio-sys 不填写时,该策略生效范围为 填写非istio-system的namesp 不填写时,策略生效范围为填 namespace填写非istio-system selector字段填写了有效键值为 为在所填namespace下根据 & Workload
spec.jwtRules.issuer	string	配置 JWT token 的 issuer,详
<pre>spec.jwtRules.audiences</pre>	string[]	配置允许访问的 JWT audienc 列表为空时,使用 service nar
spec.jwtRules.jwksUri	string	配置验证 JWT 签名的公钥 UF Discovery。同时配置 jwksUri jwksUri 将被忽略
spec.jwtRules.jwks	string	验证 JWT 签名的 JSON Web 置 jwksUri 和 jwks 字段时,jw
<pre>spec.jwtRules.fromHeaders</pre>	<pre>map<string,string> []</string,string></pre>	配置 JWT 从 header 中的提取
<pre>spec.jwtRules.fromParams</pre>	string[]	配置 JWT 从 header 中提取的 parameter mytoken(/path 取
spec.jwtRules.outputPayloadToHeader	string	配置成功验证的 JWT payload 转发的数据为 base64_encoded(jwt_pa 未填写时默认不会输出 JWT p
<pre>spec.jwtRules.forwardOriginalToken</pre>	bool	配置是否将原始 JWT 转发至 u false

使用 RequestAuthentication 配置请求 JWT 认证



```
为验证请求 JWT 认证配置的效果,您首先需要部署一个测试程序 httpbin.foo ,并配置通过 lngress Gateway
暴露此服务到公网:
创建 foo namespace, 开启 sidecar 自动注入, 部署 httpbin 服务到 foo namespace:
 apiVersion: v1
 kind: Namespace
 metadata:
   name: foo
   labels:
     istio.io/rev: 1-6-9 # 开启 namespace 的 sidecar 自动注入 (istio 版本 1.6.9)
 spec:
   finalizers:
     - kubernetes
 apiVersion: v1
 kind: ServiceAccount
 metadata:
   name: httpbin
   namespace: foo
 apiVersion: v1
 kind: Service
 metadata:
   name: httpbin
   namespace: foo
   labels:
     app: httpbin
     service: httpbin
 spec:
   ports:
   - name: http
     port: 8000
     targetPort: 80
   selector:
     app: httpbin
 apiVersion: apps/v1
 kind: Deployment
 metadata:
   name: httpbin
   namespace: foo
 spec:
   replicas: 1
   selector:
     matchLabels:
       app: httpbin
```



```
version: v1
template:
metadata:
labels:
    app: httpbin
    version: v1
spec:
    serviceAccountName: httpbin
    containers:
    - image: docker.io/kennethreitz/httpbin
    imagePullPolicy: IfNotPresent
    name: httpbin
    ports:
    - containerPort: 80
```

配置通过 Ingress Gateway 暴露 httpbin 服务至公网访问:

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
 name: httpbin-gateway
  namespace: foo
spec:
  selector:
    app: istio-ingressgateway
   istio: ingressgateway
  servers:
  - port:
     number: 80
     name: http
     protocol: HTTP
    hosts:
    _ "*"
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
 name: httpbin
  namespace: foo
spec:
 hosts:
  _ "*"
  gateways:
  - httpbin-gateway
  http:
```



```
- route:
    - destination:
        port:
            number: 8000
            host: httpbin.foo.svc.cluster.local
```

通过 curl 语句 curl "\$INGRESS_IP:80/headers" -s -o /dev/null -w "\${http_code}\\n" 测试服 务的连通性,注意您需要将代码中的 \$INGRESS_IP 替换为您的边缘代理网关 IP 地址,正常情况下会返回 200 返回码。

下面将会为边缘代理网关配置 JWT 认证规则, 放通带有符合条件的 JWT 令牌的请求。

YAML 配置示例

控制台配置示例

```
apiVersion: "security.istio.io/v1beta1"
kind: "RequestAuthentication"
metadata:
    name: "jwt-example"
    namespace: istio-system
spec:
    selector:
    matchLabels:
        istio: ingressgateway
        app: istio-ingressgateway
    jwtRules:
        - issuer: "testing@secure.istio.io"
        jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.9/security/to
```



Create Authentication	
Policy Name *	Please enter the policy name.
Policy Type *	PeerAuthentication O RequestAuthentication Configure the request authentication method of the service
Namespace •	istio-system v
Specify Service/Gateway Meth	nod Select Service By labels
Service/Gateway	istio-ingressgateway *
selector	app: istio-ingressgateway,istio: ingressgateway
JWT Rule	Rule 1 Delete
	issuer * testing@secure.istio.io
	jwksUri https://raw.githubusercontent.cor
	More -
	Add Rule
Save Cancel	

配置完成后,我们来验证配置的JWT 验证规则是否生效。

通过以下代码,携带一个非法的JWT 令牌发起访问,注意您需要将代码中的 \$INGRESS_IP 替换为您的边缘代理 网关 IP 地址。边缘代理网关不会放通携带非法 JWT 令牌的请求,因此会返回 401 返回码。

curl --header "Authorization: Bearer deadbeef" "\$INGRESS_IP:80/headers" -s -o /dev/

通过以下代码,携带一个合法的JWT 令牌发起访问,注意您需要将代码中的 \$INGRESS_IP 替换为您的边缘代理 网关 IP 地址。边缘代理网关会放通携带合法 JWT 令牌的请求,因此会返回 200 返回码。

```
TOKEN=$(curl https://raw.githubusercontent.com/istio/istio/release-1.9/security/too curl --header "Authorization: Bearer $TOKEN" "$INGRESS_IP:80/headers" -s -o /dev/nu
```

通过验证,您可以发现您为边缘代理网关配置的请求 JWT 认证规则已经生效。但此时仅仅配置了 JWT 认证规则, Ingress Gateway 仍会放通未携带 JWT 令牌的请求。限制未携带 JWT 令牌的请求需要配置 AuthorizationPolicy。应 用以下 YAML 文件至服务网格即可限制 Ingress Gateway 拒绝未携带 JWT 令牌的请求:

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
   name: frontend-ingress
   namespace: istio-system
spec:
   selector:
   matchLabels:
        app: istio-ingressgateway
        istio: ingressgateway
```



再次用未携带 JWT 令牌的方式发起访问 curl "\$INGRESS_IP:80/headers" -s -o /dev/null -w "% {http_code}\\n",发现访问失败,返回 403 返回码,AuthorizationPolicy 策略生效。



Authorization 授权策略配置

最近更新时间:2023-12-26 14:19:55

授权策略用于配置网格、namespace、服务/Workload 范围的访问管理规则。您可以通过 AuthorizationPolicy CRD 配置授权规则。AuthorizationPolicy 主要包含以下部分: selector:指定策略的生效范围。 action:指定该策略是 ALLOW 策略还是 DENY 策略。 rules:授权规则主体,由from,to,where 3 部分构成。 from:指定请求的来源特征。 to:指定请求的操作特征。 when:指定授权规则的生效条件。 当有 AuthorizationPolicy 的 ALLOW 和 DENY 策略应用于同一范围时, DNEY 策略的优先级高于 ALLOW 策 略,生效的规则如下: 1.如请求匹配任何一条 DENY 策略,则拒绝该请求的访问。 2.如该范围没有任何 ALLOW 策略,则允许该请求的访问。 3.如当前该范围存在 ALLOW 策略,且请求匹配到了任何一条 ALLOW 策略,则允许该请求的访问。 4.拒绝该请求的访问。





以下是两种特殊 AuthorizationPolicy 示例:

default namespace 的服务允许所有请求访问:

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
   name: allow-all
   namespace: default
spec:
   action: ALLOW
   rules:
   - {}
   # 规则可以匹配任何请求
```



default namespace 的服务拒绝所有请求访问:

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
    name: deny-all
    namespace: default
spec:
    {}
    # 当 action 字段没有填写时, 默认是 ALLOW, 此时请求无法匹配任何规则
```

AuthorizationPolicy 重要字段说明

以下是 AuthorizationPolicy 重要字段说明:

字段名称	字段类型	字段说明
metadata.name	string	AuthorizationPolicy 名称
metadata.namespace	string	AuthorizationPolicy 命名空间
spec.selector	<pre>map<string, string></string, </pre>	AuthorizationPolicy 使用填写的 标签键值对,配合填写的 namespace,匹配配置下发的 Workload 范围: namespace 填写 istio-system, 且 selector 字段不填写时,该策 略生效范围为整个网格 namespace 填写非 istio-system 的 namespace,且 selector 字手 不填写时,策略生效范围为填写的 namespace namespace 填写非 istio-system 的 namespace,且 selector 字手 填写了有效键值对时,策略的生 效范围为在所填 namespace 下 根据 selector 匹配到的 Workloa
spec.action	-	指定该策略是 ALLOW 策略还 是 DENY 策略
<pre>spec.rules.from.source.principals</pre>	string[]	源对等身份列表(即 service account), 匹配 source.principal 字段,



		要求启用 mTLS,未填写时则允 许任何 principal
<pre>spec.rules.from.source.requestPrincipals</pre>	string[]	请求身份列表(即iss/sub claim),匹配 request.auth.principal 字段,未填写时则允许任何 requestPrincipals
<pre>spec.rules.from.source.namespaces</pre>	string[]	请求源的 namespace 列表,匹配 source.namespace 字 段,要求启用 mTLS,未填写际 允许来自任何 namespace 的请 求
<pre>spec.rules.from.source.ipBlocks</pre>	string[]	IP block 列表, 匹配 source.ip 字段, 支持单 II 写法(如 1.2.3.4)或 CID 写法(如 1.2.3.4/24), 填写时允许任何源 IP 的访问
<pre>spec.rules.to.operation.hosts</pre>	string[]	请求的域名列表,匹配 request.host 字段,未填 写时允许任何域名,仅支持在 HTTP 协议请求中使用
<pre>spec.rules.to.operation.ports</pre>	string[]	请求的端口列表,匹配 destination.port 字段, 未填写时允许任何端口
<pre>spec.rules.to.operation.methods</pre>	string[]	请求的方法列表,匹配 request.method 字段, 使 用 gRPC 协议时该值始终应为 POST 。未填写时允许任何方 法,仅支持在 HTTP 协议请求中 使用
<pre>spec.rules.to.operation.paths</pre>	string[]	请求的路径,匹配 request.url_path 字段, 未填写时允许任何路径,仅支持 在 HTTP 协议请求中使用
<pre>spec.rules.when.condition.key</pre>	string	Istio 支持的条件字段名称,详贝 Authorization Policy Conditions
<pre>spec.rules.when.condition.values</pre>	string[]	填写对应条件的值列表



使用 AuthorizationPolicy 配置 namespace 的访问权限

为查看配置的 AuthorizationPolicy 策略效果,我们首先部署一套测试程序到网格管理的集群,部署完成后位于 test namespace 的 client 服务会自动发起对 base namespace user 服务的访问:

```
apiVersion: v1
kind: Namespace
metadata:
  name: test
  labels:
    istio.io/rev: 1-6-9 # sidecar 自动注入 (istio 1.6.9)
spec:
  finalizers:
    - kubernetes
apiVersion: apps/v1
kind: Deployment
metadata:
  name: client
  namespace: test
  labels:
    app: client
spec:
  replicas: 10
  selector:
   matchLabels:
      app: client
  template:
    metadata:
      labels:
        app: client
    spec:
      containers:
        - name: client
          image: ccr.ccs.tencentyun.com/zhulei/testclient:v1
          imagePullPolicy: Always
          env:
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: REGION
              value: "guangzhou-zoneA"
          ports:
            - containerPort: 7000
              protocol: TCP
```



```
apiVersion: v1
kind: Service
metadata:
 name: client
 namespace: test
  labels:
   app: client
spec:
  ports:
   - name: http
     port: 7000
     protocol: TCP
  selector:
   app: client
  type: ClusterIP
apiVersion: v1
kind: Namespace
metadata:
 name: base
 labels:
   istio.io/rev: 1-6-9
spec:
 finalizers:
    - kubernetes
____
apiVersion: apps/v1
kind: Deployment
metadata:
 name: user
  namespace: base
 labels:
   app: user
spec:
  replicas: 1
  selector:
   matchLabels:
     app: user
  template:
    metadata:
      labels:
       app: user
    spec:
     containers:
       - name: user
```



```
image: ccr.ccs.tencentyun.com/zhulei/testuser:v1
          imagePullPolicy: Always
          env:
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: REGION
              value: "guangzhou-zoneB"
          ports:
            - containerPort: 7000
___
apiVersion: v1
kind: Service
metadata:
 name: user
 namespace: base
  labels:
   app: user
spec:
 ports:
    - port: 7000
     name: http
  selector:
    app: user
```

查看 client 容器的日志,会发现访问成功,正确返回了 user 信息:



Clust	er (Singap	ore) /			/ [eploymer	nt:clie	nt(test)							
od mana	gement	Up	date hi	istory	Even	t Lo	g	Details	YAML						
	-						-								
Conditio	nal filterir	ıg													
		-													
Pod optior	is(i)	client	t-5f968b	69b4-29	448	- client			-						
Other opti	ons	100 c	data entr	ries		•									
	llcorlu		VIDI	truo	Namo	Kov(1p									
1 2	USEIID	· ⊥, · 1	Vip:	true,	Name	Kevin									
2	llcorTD	• ±, • 1	Vip.	true,	Name	Kevin									
	llcorTD	• 1, • 1	Vip.	true,	Name	Kevin									
5	llserTD	· -,	Vin:	true.	Name:	Kevin									
6	UserTD	· -, : 1.	Vin:	true.	Name:	Kevin									
7	UserTD	· -, : 1.	Vin:	true.	Name:	Kevin									
8	UserID	: 1.	Vip:	true.	Name:	Kevin									
9	UserID	: 1,	Vip:	true,	Name:	Kevin									
10	UserID	: 1,	Vip:	true,	Name:	Kevin									
11	UserID	: 1,	Vip:	true,	Name:	Kevin									
12	UserID	: 1,	Vip:	true,	Name:	Kevin									
13	UserID	: 1,	Vip:	true,	Name:	Kevin									
14	UserID	: 1,	Vip:	true,	Name:	Kevin									
15	UserID	: 1,	Vip:	true,	Name:	Kevin									
16	UserID	: 1,	Vip:	true,	Name:	Kevin									
17	UserID	: 1,	Vip:	true,	Name:	Kevin									
18	UserID	: 1,	Vip:	true,	Name:	Kevin									
19	UserID	: 1,	Vip:	true,	Name:	Kevin									
20	UserID	: 1,	Vip:	true,	Name:	Kevin									
21	UserID	: 1,	Vip:	true,	Name:	Kevin									
22	UserID	: 1,	Vip:	true,	Name:	Kevin									
23	UserID	: 1,	Vip:	true,	Name:	Kevin									

接下来将配置 Authorization 策略,不允许 base namespace 的服务被 test namespace 的服务访问(需要开启 mTLS)。

YAML 配置示例

控制台配置示例

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
   name: base-authz
   namespace: base
spec:
   action: DENY
   rules:
        - from:
        - source:
            namespaces:
            - test
```



Del	in the second	h				
Pol	icy Name *	base-authz				
Nar	mespace *	base	▼			
Spe	ecify Service	Select Serv	ice By labels			
Ser	vice/Gateway	all 🔻 all	~			
sele	ector	N/A				
Pol	icy	ALLOW	O DENY			
Ma	tching Rule	Rule 1			Delete	
		Source				
			namespace: •	· test	×	
			Add Source			
		Operation	Add Operation			
		Condition	Add Condition			
		Add Rule				

配置完成后再次查看 client 的容器日志,发现所有访问均失败,没有返回 user 信息,AuthorizationPolicy 生效。



I management Update history Event Log Details YAML anditional filtering d options io0 data entries No logs yet	Cluster (Singa	oore) /	/ Deploymen	it:client(test)			
anditional filtering d optional ()	d management	Update history	Event Lo	g Details	YAML		
anditional filtering d options 100 data entries 1 No logs yet							
d options	Conditional filteri	ng					
her options	od options(j)	client-5f968b69b4-29448	▼ client		•		
1 No logs yet	Other options	100 data entries	~				
1 No logs yet							
1 No logs yet							
	1 No log	js yet					

使用 AuthorizationPolicy 配置 Ingress Gateway 的 IP 黑白名单

您可以使用 AuthorizationPolicy 为边缘代理网关 Ingress Gateway 配置 IP 黑/白名单。 为验证黑白名单的配置效果,您首先需要部署一个测试程序 httpbin.foo,并配置通过 Ingress Gateway 暴露此服务到公网:

创建 foo namespace,开启 sidecar 自动注入,部署 httpbin 服务到 foo namespace:

```
apiVersion: v1
kind: Namespace
metadata:
    name: foo
    labels:
        istio.io/rev: 1-6-9 # 开启 namespace 的 sidecar 自动注入 (istio 版本 1.6.9)
spec:
    finalizers:
        - kubernetes
```



```
apiVersion: v1
kind: ServiceAccount
metadata:
 name: httpbin
  namespace: foo
____
apiVersion: v1
kind: Service
metadata:
  name: httpbin
  namespace: foo
  labels:
    app: httpbin
   service: httpbin
spec:
  ports:
  - name: http
   port: 8000
   targetPort: 80
  selector:
    app: httpbin
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpbin
  namespace: foo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: httpbin
      version: v1
  template:
    metadata:
      labels:
        app: httpbin
        version: v1
    spec:
      serviceAccountName: httpbin
      containers:
      - image: docker.io/kennethreitz/httpbin
        imagePullPolicy: IfNotPresent
        name: httpbin
        ports:
        - containerPort: 80
```



配置通过 Ingress Gateway 暴露 httpbin 服务至公网访问:

```
apiVersion: networking.istio.io/v1alpha3
 kind: Gateway
 metadata:
   name: httpbin-gateway
   namespace: foo
 spec:
   selector:
     app: istio-ingressgateway
     istio: ingressgateway
   servers:
   - port:
       number: 80
       name: http
       protocol: HTTP
     hosts:
      _ "*"
 apiVersion: networking.istio.io/v1alpha3
 kind: VirtualService
 metadata:
   name: httpbin
   namespace: foo
 spec:
   hosts:
   _ "*"
   gateways:
   - httpbin-gateway
   http:
   - route:
      - destination:
          port:
           number: 8000
         host: httpbin.foo.svc.cluster.local
通过 curl 语句 curl "$INGRESS_IP:80/headers" -s -o /dev/null -w "%{http_code}\\n" 测试服
```

务的连通性,注意您需要将代码中的 \$INGRESS_IP 替换为您的边缘代理网关 IP 地址,正常情况下会返回 200 返回码。

为使 Ingress Gateway 能正确获取真实客户端源 IP,我们需要修改 Ingress Gateway Service 的 ExternalTrafficPolicy 为 Local,保证流量仅在本节点转发不做 SNAT。



← Update access method			
	Basic Information		
	Region		
	Cluster ID		
	Gluster ID		
	Namespace		
	Resource Name		
	If you change the which may affect	service access method, the original public/private CLB created while usin your running business.	ng the Internet Access or Private Network Access w
	Service Access	ClusterIP NodePort O LoadBalancer (public network)	C LoadBalancer (private network)How to select 🗹
		After the architecture upgrade at 00:00:00 on November 2, 20 second, and 5,000 queries per second (QPS). The price now instances according to your actual needs. <u>View announceme</u>	021 (UTC +8), all CLB instances are guaranteed to s for private/public CLB instances ranges from 0.686 nt 12
		A public CLB is automatically created for Internet access (0.003 USD/h If you need to forward via internet using HTTP/HTTPS protocols or by t	our). It supports TCP/UDP protocol and is applicab URL, you can go to Ingress page to configure Ingres
	IP Version	IPV4 The IP version cannot be changed.	
	Port Mapping	Protocol (j) Target Port (j)	Port(j)
		TCP = 80	80
		TCP • 15021	15021
		15442	15449
		10440	10440
		Add Port Mapping	
	ExternalTrafficPolicv		
		Preserve the client IP, and ensure that traffic is only forwarded within th check for nodes without pods may fail, raising the risk of unbalanced tr	e node if the access mode is public network, VPC p raffic forwarding.
		- A structure	
	Local Binding	Activate When it's enabled, the load balancer will only be bound with nodes with	h pods.
	Local Weighted Balancing	Activate	
		According to the number of pods on the backend node, automatically	configure the load balancing weight forwarded to thi
	Session Affinity	ClientIP ONO	- gg.
	Update a	Cancel	

下面将会使用 AuthorizationPolicy 把本机的 IP 地址列入 Ingress Gateway 的黑名单,并验证黑名单是否生效。 YAML 配置示例

I AIVIL ELE/IVI/I

控制台配置示例

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
   name: black-list
   namespace: istio-system
spec:
   selector:
   matchLabels:
        app: istio-ingressgateway
```



Policy Name *	black-list
Namespace *	istio-system 👻
Specify Service	Select Service By labels
Service/Gateway	istio-ingressgateway 💌
selector	app: istio-ingressgateway,istio: ingressgateway
Policy	ALLOW O DENY
Matching Rule	Rule 1
	Source ipBlocks • :
	Add Source your local IP
	Operation Add Operation
	Condition Add Condition
	Add Rule

配置完成后再次通过 curl 语句 curl "\$INGRESS_IP:80/headers" -s -o /dev/null -w "\$
{http_code}\\n" 测试服务的连通性,注意您需要将代码中的 \$INGRESS_IP 替换为您的边缘代理网关 IP 地
址,此时访问失败,返回 403 返回码,黑名单策略生效。



访问管理 概述

最近更新时间:2023-12-26 14:20:21

服务网格的权限管理包含2个部分:访问管理(CAM) 权限与 容器服务(TKE) RBAC 权限。

默认情况下,子账号不具备 CAM 的权限,非集群创建者的子账号不具备相关集群 RBAC 的权限。您需要创建关联 CAM 策略和 TKE RBAC 授权策略来允许子账号访问或正常使用他们所需要的服务网格资源。

CAM 权限策略的编辑和授予是由 CAM 管理员(通常是主账号或拥有 CAM 权限的子账号)完成,更多关于 CAM 策略的基本信息,请参见 CAM 策略。TKE 集群的 RBAC 权限策略的编辑和授予通常是由相应集群管理员(通常是主账号或集群创建账号)完成,授权方式参见 TKE RBAC 授权。

说明:

若您不需要对子账户进行 TCM 相关资源的访问管理,您可以跳过此章节,跳过这些部分不会影响您对文档中其余部分的理解和使用。

TCM 基于 CAM 的权限控制

当前 TCM 支持基于 CAM 的资源级的权限控制,即能够允许指定**子账号**对指定**资源**的指定**操作**。子账号默认没有 TCM 相关 CAM 权限,您需要将策略关联至子账号完成授权。

当前 TCM 基于 CAM 的资源级权限控制颗粒度可达到网格实例级别,即您可以控制指定子账号对指定网格能够执行 指定的操作。

TCM 相关产品 TKE 的 RBAC 权限管理

使用 TCM 过程中,会涉及到对 TCM 管理的 TKE 集群内 Kubernetes 资源的读写操作,这些操作需要有足够的 TKE RBAC 权限。默认非集群创建者的子账号没有该集群的 RBAC 权限,需要集群管理员授予该子账号对应集群的 RBAC 权限,子账号才能正常使用 TCM。

在所选集群创建/删除/更新服务网格、添加/解关联服务发现集群、在所选集群创建/删除 Ingress Gateway 网关均需 要相应集群的管理员(tke:admin)权限。对网格内 Istio 资源(Gateway, VirtualService, DestinationRule, ServiceEntry 等)的操作不需要集群的 RBAC 权限。

更多 TKE Kubernetes 对象级权限控制信息,请参见 TKE Kubernetes 对象级权限控制。TKE RBAC 授权方式,请参见 授权模式对比。



CAM 服务角色授权

最近更新时间:2024-12-17 15:12:29

在使用腾讯云服务网格(Tencent Cloud Mesh, TCM)的过程中,涉及到服务网格相关云资源的使用,为了您能正常使用 TCM 的功能,您需要对 TCM 的服务角色 TCM_QCSRole 进行授权,授权后,TCM 服务才能使用相关云资源。

需要服务授权的场景主要包含 首次登录 服务网格控制台 以及 首次使用 TCM 一键体验功能 两个场景,分别对应 QcloudAccessForTCMRole 和 QcloudAccessForTCMRoleInSampleDeployment 两个预设策略。

首次登录服务网格控制台

授权场景

当您已注册并登录腾讯云账号后,首次登录 服务网格控制台时,需前往**访问管理**页面对当前账号授予腾讯云服务网 格操作容器服务(TKE)、SSL证书(SSL)、日志服务(CLS)等云资源的权限。该权限授予通过关联预设策略 QcloudAccessForTCMRole 至 TCM 服务角色 TCM_QCSRole 完成。如您之前未创建过 TCM 服务角色,该 授权流程还会涉及 TCM 服务角色的创建。

授权步骤

1. 首次登录服务网格控制台, 自动弹出服务授权窗口。



- 2. 单击前往访问管理,进入 CAM 控制台服务授权页面。
- 3. 单击同意授权, 完成身份验证后即可成功授权。



Service Authoriza	ation
After you agree to gra	ant permissions to Tencent Cloud Mesh, a preset role will be created and relevant permissions will be granted to Ter
Role Name	TCM_QCSRole
Role Type	Service Role
Description	Current role is a Tencent Cloud Mesh service role, which will access your other cloud service resources within the
Authorized Policies	Preset policy QcloudAccessForTCMRole

权限内容

容器服务(TKE)相关

权限	描述	资源
DescribeClusterSecurity	查询集群密钥	所有资源 *

SSL 证书(SSL)相关

权限	描述	资源
DescribeCertificateDetail	获取证书详情	所有资源 *

日志服务(CLS)相关

权限	描述	资源
getLogset	获取日志集详情	所有资源 *
getTopic	获取日志主题详情	所有资源 *
createLogset	创建日志集	所有资源 *
createTopic	创建日志主题	所有资源 *
modifyIndex	修改索引	所有资源 *

listLogset	获取日志集列表	所有资源	*
listTopic	获取日志主题列表	所有资源	*



CAM 预设策略授权

最近更新时间:2023-12-26 14:20:53

您可以把 CAM 中的 TCM 相关预设策略关联至子账号,快速完成 TCM 的 CAM 授权。

TCM 相关预设策略

您可以使用以下预设策略为您的子账号授予相关权限:

策略	描述
QcloudTCMFullAccess	服务网格(TCM)全读写访问权限(创建、删除等全部操作)
QcloudTCMReadOnlyAccess	服务网格(TCM)只读访问权限(可以查看 TCM 下所有资源,但无法创建、更新或删除它们)

服务网格全读写预设策略

策略名:QcloudTCMFullAccess,策略内容:

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
               "tcm:*"
        ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

服务网格只读预设策略

策略名:QcloudTCMReadOnlyAccess,策略内容:

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
            "tcm:List*",
            "t
```



```
"tcm:Describe*",
    "tcm:ForwardRequestRead"
],
    "resource": "*",
    "effect": "allow"
    }
]
}
```

TCM 相关产品的 CAM 权限

使用 TCM 还涉及到关联的 VPC、CCN、CLB、TKE等产品的 CAM 权限,您可以参考相应产品的 CAM 授权文档,为子账号授予合适的权限:

TCM 相关产品	授权指南文档
私有网络(VPC)	VPC 访问管理概述
负载均衡(CLB)	CLB 访问管理概述
容器服务(TKE)	TKE 权限管理概述

子账号关联预设策略

您可在创建子账号的"设置用户权限"步骤中,通过直接关联或随组关联方式,为该子账户关联预设策略。

直接关联

您可以直接为子账号关联策略以获取策略包含的权限。

- 1. 登录访问管理控制台,选择左侧导航栏中的用户 > 用户列表。
- 2. 在用户列表页面,选择需要设置权限的子账号所在行右侧的授权。
- 3. 在弹出的"关联策略"窗口中, 勾选需授权的策略。

4. 单击**确定**即可。

随组关联

您可以将子账号添加至用户组,该子账号将自动获取该用户组所关联策略的权限。如需解除随组关联策略,仅需将 子账号移出相应用户组即可。

- 1. 登录访问管理控制台,选择左侧导航栏中的用户 > 用户列表。
- 2. 在用户列表管理页面,选择需要设置权限的子账号所在行右侧的更多操作 > 添加到组。
- 3. 在弹出的"添加到组"窗口中,勾选需加入的用户组。



4. 单击确定即可。

登录子账号验证

登录 腾讯云服务网格控制台,验证可使用所授权策略对应功能,则表示子账号授权成功。



CAM 自定义策略授权

最近更新时间:2023-12-26 14:21:03

如您有自定义的权限管理诉求,您可以通过创建 CAM 自定义策略,并关联至子账号实现自定义授权。您可参考文本 并根据实际业务诉求进行配置。

CAM 元素参考

CAM 自定义策略核心元素包括:操作(action)、资源(resource)、生效条件(condition)以及效力(effect)。

1. 操作(action)

描述允许或拒绝的操作。操作可以是 API(以 name 前缀描述)或者功能集(一组特定的 API,以 actionName 前缀描述)。该元素是必填项。您可以查看 TCM 接入 CAM 的 API。

2. 资源(resource)

描述授权的具体数据。资源是用六段式描述。您可以查看 TCM 资源描述。

3. 生效条件(condition)

描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。

4. 效力(effect)

描述声明产生的结果是"允许"还是"显式拒绝"。包括 allow(允许)和 deny (显式拒绝)两种情况。该元素是必填 项。

5. 自定义策略样例

该策略为:允许对广州的两个mesh实例:mesh-abcd1234 和 mesh-1234abcd 做获取详情操作。



}] }

更多关于 CAM 自定义策略语法逻辑,请参见 CAM 语法逻辑。

CAM 中可授权的 TCM 资源

资源	授权策略中的资源描述方法
服务网格	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>

其中:

\$region : 描述地域信息, 应为某个 region 的 ID, 例如 gz 为广州。

\$account : 描述资源拥有者的主账号信息,表示为 uin/\${uin},例如 uin/12345678,若值为空则表示创建策略的 CAM 用户所属的主账号。

\$meshid : 描述mesh实例信息, 应为某个网格的 ID, 或者 *。 关于授权策略中的资源描述方式, 请参见 资源描述方式。

CAM 中可对 TCM 进行授权的接口

在 CAM 中, 可以对 TCM mesh 资源进行以下操作(action)的授权。

Mesh 实例相关

API 操作	API 描述	资源	
CreateMesh	创建服务网格	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/*</pre>
DeleteMesh	删除服务网格	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>
DescribeMesh	获取指定服务网格	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>
ListMeshes	获取服务网格列表	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>
ModifyMesh	修改服务网格配置	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>
UpgradeMesh	升级服务网格	mesh 资源	<pre>qcs::tcm:\$region:\$account:mesh/\$meshid</pre>

Istio 资源相关



API 操作	API 描述	资源
ForwardRequestRead	读 Istio 的 CRD 资 源	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid
ForwardRequestWrite	写 Istio 的 CRD 资 源	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid

服务发现相关

API 操作	API 描述	资源
LinkClusterList	关联集群到服务网格实 例	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid
UnlinkCluster	解除关联集群	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid

边缘代理网关相关

API 操作	API 描述	资源
CreateIngressGateway	创建 IngressGateway	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid
DeleteGatewayInstance	删除 IngressGateway	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid
DescribeIngressGatewayList	查询 IngressGateway 列 表	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid
ModifyIngressGateway	修改 IngressGateway	mesh 资源 qcs::tcm:\$region:\$account:mesh/\$meshid

体验环境相关

API 操作	API 描述	资源
CreateTrial	创建服务网格一键体验环境	只对接口进行鉴权 *
DeleteTrial	删除服务网格一键体验环境	只对接口进行鉴权 *



RetryTrialTask

只对接口进行鉴权 *



功能扩展 使用 wasm filter 扩展数据面

最近更新时间:2023-12-26 14:21:25

Wasm 是 WebAssembly 的缩写,可以编写二进制形式的指令加载到 envoy filter chain 中,实现网格数据面能力扩展。这种形式使得 envoy 和扩展组件的解耦,用户不再需要通过修改 envoy 代码、编译特殊的 envoy 版本来实现能力扩展,并且还具备动态加载和安全隔离等优势。

从 Istio 1.6 版本开始, Proxy-Wasm 沙盒 API 取代了 Mixer 作为 Istio 主要的扩展实现方案,用于实现 envoy 和 wasm 虚拟机之间的交互,因此通过 wasm filter 来扩展 envoy 需要使用 Proxy-WASM SDK。 通常编写 wasm 文件扩展网格数据面能力主要分为以下几步:

1. 编写 wasm filter,可请参见示例。

2. 将 wasm filter 注入到 configmap 中,通过 configmap 将 wasm filter 挂载到任意工作负载,避免将 wasm filter 拷贝 到多个 node 上。

kubectl create cm -n foo example-filter --from-file=example-filter.wasm

3. 将 wasm filter 挂载到业务工作负载,可利用 lstio 的 annotation 机制,在创建工作负载的时候自动挂载相应的文件:

```
sidecar.istio.io/userVolume: '[{"name":"wasmfilters-dir","configMap": {"name": "exa
sidecar.istio.io/userVolumeMount: '[{"mountPath":"/var/local/lib/wasm-filters","nam
```

将 annotation 应用到对应的工作负载之上:

kubectl patch deployment -n foo frontpage-v1 -p '{"spec":{"template":{"metadata":{"

4. 创建 envoyfilter,将 wasm filter 添加到对应工作负载的 envoy filter chain 中,使其生效。

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
   name: frontpage-v1-examplefilter
   namespace: foo
spec:
   configPatches:
        applyTo: HTTP_FILTER
        match:
        listener:
        filterChain:
        filter:
        name: envoy.http_connection_manager
```



```
subFilter:
            name: envoy.router
 patch:
    operation: INSERT_BEFORE
    value:
      name: envoy.filters.http.wasm
      typed_config:
        '@type': type.googleapis.com/envoy.extensions.filters.http.wasm.v3.Wasm
        config:
          name: example-filter
          root_id: my_root_id
          vm_config:
            code:
              local:
                filename: /var/local/lib/wasm-filters/example-filter.wasm
            runtime: envoy.wasm.runtime.v8
            vm_id: example-filter
            allow_precompiled: true
workloadSelector:
  labels:
    app: frontpage
    version: v1
```

至此, wasm filter 部署完成, 另一种 wasm filter 的使用形式是镜像, 请参见 制作 wasm filter 镜像, 利用 WASME 工 具部署, 请参见 使用 wasme 部署 wasm filter。

可以看出,wasm filter的部署较为繁琐,尤其是需要大规模部署的时候,没有工具将难以进行批量部署和管理,TCM 提供了便捷的部署工具,可以利用工具将二进制或者镜像形式的 wasm filter 批量部署到业务中,详情见使用TCM 工具批量部署 wasm filter。