

1. Istio
2. 文件
3. 参考
4. 组态
5. 交通管理
6. 虚拟服务

# 虚拟服务

🕒 22分钟阅读

影响流量路由的配置。以下是一些在流量路由环境中定义的有用术语。

**Service** 绑定到服务注册表中唯一名称的应用程序行为单位。服务由在 pod，容器，VM 等上运行的工作负载实例实现的多个网络端点组成。

**Service versions (a.k.a. subsets)** – 在持续部署方案中，对于给定服务，可以存在运行应用程序二进制文件的不同变体的不同实例子集。这些变体不一定是不同的 API 版本。它们可以是对同一服务的迭代更改，部署在不同的环境中（prod，staging，dev 等）。发生这种情况的常见情况包括 A / B 测试，金丝雀推出等。可以基于各种标准（标题，URL 等）和/或分配给每个版本的权重来决定特定版本的选择。每个服务都有一个默认版本，包含其所有实例。

**Source** – 呼叫服务的下游客户端。

**Host** – 客户端在尝试连接服务时使用的地址。

**Access model** – 应用程序仅在不知道各个服务版本（子集）的情况下仅对目标服务（主机）进行寻址。版本的实际选择由代理/边车确定，使应用程序代码能够脱离依赖服务的演变。

A **VirtualService** 定义了一组要在主机被寻址时应用的流量路由规则。每个路由规则定义特定协议的流量的匹配标准。如果流量匹配，则将其发送到注册表中定义的命名目标服务（或其子集/版本）。

流量来源也可以在路由规则中匹配。这允许为特定客户端上下文定制路由。

以下 Kubernetes 示例，默认情况下将所有 HTTP 流量路由到评论服务的 pod，标签为“version: v1”。此外，带有以 / wpcatalog / 或 / consumercatalog / 开头的路径的 HTTP 请求将被重写为 / newcatalog 并发送到标签为“version: v2”的 pod。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews.prod.svc.cluster.local
  http:
  - match:
    - uri:
        prefix: "/wpcatalog"
    - uri:
        prefix: "/consumercatalog"
    rewrite:
      uri: "/newcatalog"
    route:
    - destination:
        host: reviews.prod.svc.cluster.local
        subset: v2
    - route:
        destination:
          host: reviews.prod.svc.cluster.local
          subset: v1
```

路由目的地的子集/版本通过对必须在相应中声明的命名服务子集的引用来标识 **DestinationRule**。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews-destination
spec:
  host: reviews.prod.svc.cluster.local
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

---

## CorsPolicy

描述给定服务的跨源资源共享（CORS）策略。有关跨源资源共享的更多详细信息，请参阅[CORS1](#)。例如，以下规则使用HTTP POST / GET将跨源请求限制为源自example.com域的请求，并将**Access-Control-Allow-Credentials**标头设置为false。此外，它只暴露**X-Foo-bar**标题并设置1天的到期时间。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - route:
    - destination:
        host: ratings.prod.svc.cluster.local
        subset: v1
  corsPolicy:
    allowOrigin:
    - example.com
    allowMethods:
    - POST
    - GET
    allowCredentials: false
    allowHeaders:
    - X-Foo-Bar
    maxAge: "24h"
```

领域	类型	描述
allowOrigin	string[]	允许执行CORS请求的源的列表。内容将序列化为Access-Control-Allow-Origin标头。通配符*将允许所有来源。
allowMethods	string[]	允许访问资源的HTTP方法列表。内容将序列化为Access-Control-Allow-Methods标头。
allowHeaders	string[]	请求资源时可以使用的HTTP标头列表。序列化为Access-Control-Allow-Headers标头。
exposeHeaders	string[]	允许浏览器访问的HTTP标头的白名单。序列化为Access-Control-Expose-Headers标头。
maxAge	google.protobuf.Duration2	指定可以缓存预检请求结果的时间。转换为Access-Control-Max-Age标题。
allowCredentials	google.protobuf.BoolValue3	指示是否允许调用者使用凭据发送实际请求（而不是预检）。转换为Access-Control-Allow-Credentials标题。

# 目的地

Destination表示处理路由规则后请求/连接将发送到的网络可寻址服务。destination.host应明确引用服务注册表中的服务。Istio的服务注册表由平台服务注册表中的所有服务组成（例如，Kubernetes服务，Consul服务），以及通过[ServiceEntry](#)资源声明的 [服务](#)。

*Kubernetes用户注意事项：* 当使用短名称（例如“review”而不是“reviews.default.svc.cluster.local”）时，Istio将根据规则的命名空间而不是服务来解释短名称。包含主机“评论”的“默认”命名空间中的规则将被解释为“reviews.default.svc.cluster.local”，而不管与评论服务相关联的实际命名空间。为避免潜在的错误配置，建议始终在短名称上使用完全限定的域名。

以下Kubernetes示例在Kubernetes环境中默认将所有流量路由到评论服务的pod，其标签为“version: v1”（即子集v1），一些为子集v2。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
  namespace: foo
spec:
  hosts:
  - reviews # interpreted as reviews.foo.svc.cluster.local
  http:
  - match:
    - uri:
        prefix: "/wpcatalog"
    - uri:
        prefix: "/consumercatalog"
    rewrite:
      uri: "/newcatalog"
    route:
    - destination:
        host: reviews # interpreted as reviews.foo.svc.cluster.local
        subset: v2
  - route:
    - destination:
        host: reviews # interpreted as reviews.foo.svc.cluster.local
        subset: v1
```

以及相关的DestinationRule

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews-destination
  namespace: foo
spec:
  host: reviews # interpreted as reviews.foo.svc.cluster.local
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

对于Kubernetes中对productpage.prod.svc.cluster.local服务的所有调用，以下VirtualService设置超时5秒。请注意，此规则中没有定义子集。Istio将从服务注册表获取productpage.prod.svc.cluster.local服务的所有实例，并填充sidecar的负载平衡池。另请注意，此规

则是在istio-system命名空间中设置的，但使用产品页面服务的完全限定域名productpage.prod.svc.cluster.local。因此，规则的命名空间对解析productpage服务的名称没有影响。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: my-productpage-rule
  namespace: istio-system
spec:
  hosts:
  - productpage.prod.svc.cluster.local # ignores rule namespace
  http:
  - timeout: 5s
    route:
    - destination:
        host: productpage.prod.svc.cluster.local
```

要控制绑定到网格外部服务的流量的路由，必须首先使用ServiceEntry资源将外部服务添加到Istio的内部服务注册表。然后可以定义VirtualServices以控制绑定到这些外部服务的流量。例如，以下规则为wikipedia.org定义服务，并为http请求设置5秒的超时。

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: external-svc-wikipedia
spec:
  hosts:
  - wikipedia.org
  location: MESH_EXTERNAL
  ports:
  - number: 80
    name: example-http
    protocol: HTTP
  resolution: DNS

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: my-wiki-rule
spec:
  hosts:
  - wikipedia.org
  http:
  - timeout: 5s
    route:
    - destination:
        host: wikipedia.org
```

领域	类型	描述
----	----	----

领域	类型	描述
host	string	需要。服务注册表中的服务名称。从平台的服务注册表（例如，Kubernetes服务，Consul服务等）以及ServiceEntry声明的主机中查找服务名称。转发到两者中任何一个都找不到的目的地的流量将被删除。  <i>Kubernetes用户注意事项：</i> 当使用短名称（例如“review”而不是“reviews.default.svc.cluster.local”）时，Istio将根据规则的命名空间而不是服务来解释短名称。包含主机“评论”的“默认”命名空间中的规则将被解释为“reviews.default.svc.cluster.local”，而不管与评论服务相关联的实际命名空间。为避免潜在的错误配置，建议始终在短名称上使用完全限定的域名。
subset	string	服务中子集的名称。仅适用于网格内的服务。必须在相应的DestinationRule中定义子集。
port	PortSelector	指定要寻址的主机上的端口。如果服务仅公开单个端口，则不需要显式选择端口。

## HTTPFaultInjection

HTTPFaultInjection可用于指定在将http请求转发到路由中指定的目标时要注入的一个或多个故障。故障规范是VirtualService规则的一部分。错误包括从下游服务中止Http请求，和/或延迟请求的代理。故障规则必须延迟或中止或两者兼而有之。

注意：即使两者同时指定，延迟和中止故障也是相互独立的。

领域	类型	描述
delay	HTTPFaultInjection.Delay	转发前延迟请求，模拟各种故障，如网络问题，上游服务过载等。
abort	HTTPFaultInjection.Abort	中止Http请求尝试并将错误代码返回到下游服务，给人的印象是上游服务有故障。

## HTTPFaultInjection.Abort

中止规范用于使用预先指定的错误代码过早中止请求。以下示例将针对“评级”服务“v1”的每1000个请求中的1个返回HTTP 400错误代码。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - route:
    - destination:
        host: ratings.prod.svc.cluster.local
        subset: v1
    fault:
      abort:
        percentage:
          value: 0.1
        httpStatus: 400
```

该则`httpStatus`字段被用来指示HTTP状态代码返回给调用者。可选的`百分比`字段可用于仅中止一定百分比的请求。如果未指定，则中止所有请求。

领域	类型	描述
percent	int32	使用提供的错误代码中止请求的百分比（0–100）。 <code>percent</code> 不推荐使用整数值。请改用双 <code>percentage</code> 字段。
httpStatus	int32 (oneof)	需要。用于中止Http请求的HTTP状态代码。
percentage	Percent	使用提供的错误代码中止请求的百分比。

## HTTPFaultInjection.Delay

延迟规范用于将延迟注入请求转发路径。以下示例将在所有带有标签`env`的pod的“v1”版本的“v1”版本中，每1000个请求中有1个引入5秒延迟：`prod`

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews.prod.svc.cluster.local
  http:
  - match:
    - sourceLabels:
      env: prod
    route:
    - destination:
      host: reviews.prod.svc.cluster.local
      subset: v1
    fault:
      delay:
        percentage:
          value: 0.1
        fixedDelay: 5s
```

所述*FIXEDDELAY*字段用于指示秒的延迟量。可选的*百分比*字段可用于仅延迟一定百分比的请求。如果未指定，则所有请求都将延迟。

领域	类型	描述
percent	int32	注入延迟的请求百分比（0–100）。 <i>percent</i> 不推荐使用整数值。请改用双 <i>percentage</i> 字段。
fixedDelay	google.protobuf.Duration (oneof)2	需要。在转发请求之前添加固定延迟。格式：1h / 1m / 1s / 1ms。必须> = 1ms。
percentage	Percent	注入延迟的请求百分比。

## HTTPMatchRequest

HttpMatchRequest指定要满足的一组条件，以便将规则应用于HTTP请求。例如，以下限制规则仅匹配URL路径以/ ratings / v2 /开头的请求，并且请求包含*end-user*带值的自定义标头*jason*。



```

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - match:
    - headers:
      end-user:
        exact: jason
      uri:
        prefix: "/ratings/v2/"
    route:
    - destination:
        host: ratings.prod.svc.cluster.local

```

HTTPMatchRequest不能为空。

领域	类型	描述
uri	StringMatch	匹配值的URI区分大小写，格式如下： <ul style="list-style-type: none"> <li>• <b>exact:</b> "value" 用于精确的字符串匹配</li> <li>• <b>prefix:</b> "value" 用于基于前缀的匹配</li> <li>• <b>regex:</b> "value" 用于ECMAScript样式的基于正则表达式的匹配</li> </ul>
scheme	StringMatch	URI Scheme值区分大小写，格式如下： <ul style="list-style-type: none"> <li>• <b>exact:</b> "value" 用于精确的字符串匹配</li> <li>• <b>prefix:</b> "value" 用于基于前缀的匹配</li> <li>• <b>regex:</b> "value" 用于ECMAScript样式的基于正则表达式的匹配</li> </ul>
method	StringMatch	HTTP方法值区分大小写，格式如下： <ul style="list-style-type: none"> <li>• <b>exact:</b> "value" 用于精确的字符串匹配</li> <li>• <b>prefix:</b> "value" 用于基于前缀的匹配</li> <li>• <b>regex:</b> "value" 用于ECMAScript样式的基于正则表达式的匹配</li> </ul>
authority	StringMatch	HTTP Authority值区分大小写，格式如下： <ul style="list-style-type: none"> <li>• <b>exact:</b> "value" 用于精确的字符串匹配</li> <li>• <b>prefix:</b> "value" 用于基于前缀的匹配</li> <li>• <b>regex:</b> "value" 用于ECMAScript样式的基于正则表达式的匹配</li> </ul>

领域	类型	描述
headers	map<string, StringMatch>	<p>标题键必须为小写，并使用连字符作为分隔符，例如 <code>x-request-id</code>。</p> <p>标头值区分大小写，格式如下：</p> <ul style="list-style-type: none"> <li><b>exact:</b> "value" 用于精确的字符串匹配</li> <li><b>prefix:</b> "value" 用于基于前缀的匹配</li> <li><b>regex:</b> "value" 用于ECMAScript样式的基于正则表达式的匹配</li> </ul> <p>注：该键 <code>uri</code>，<code>scheme</code>，<code>method</code>，和 <code>authority</code> 将被忽略。</p>
port	uint32	指定要寻址的主机上的端口。许多服务仅使用它们支持的协议公开单个端口或标签端口，在这些情况下，不需要显式选择端口。
sourceLabels	map<string, string>	一个或多个标签，用于将规则的适用性限制为具有给定标签的工作负载。如果VirtualService具有在顶部指定的网关列表，则它应包括保留的网关 <code>mesh</code> ，以使该字段适用。
gateways	string[]	应该应用规则的网关的名称。重写VirtualService顶部的网关名称（如果有）。网关匹配独立于sourceLabels。
queryParams	map<string, StringMatch>	<p>查询匹配参数。</p> <p>例如： – 对于像“? key = true”这样的查询参数，映射键为“key”，字符串匹配可以定义为<b>exact:</b> <code>"true"</code>。 – 对于像“? key”这样的查询参数，映射键可以是“key”，字符串匹配可以定义为<b>exact:</b> <code>""</code>。 – 对于像“? key = 123”这样的查询参数，映射键为“key”，字符串匹配可以定义为<b>regex:</b> <code>"\d+\$"</code>。请注意，此配置仅匹配“123”之类的值，但不匹配“a123”或“123a”。</p> <p>注意： <b>prefix</b> 目前不支持匹配。</p>

## HTTPRedirect

HTTPRedirect可用于向调用者发送301重定向响应，其中可以使用指定值交换响应中的Authority / Host和URI。例如，以下规则将评级服务上的 / v1 / getProductRatings API请求重定向到bookratings服务提供的 / v1 / bookRatings。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - match:
    - uri:
        exact: /v1/getProductRatings
    redirect:
      uri: /v1/bookRatings
      authority: newratings.default.svc.cluster.local
  ...
```

领域	类型	描述
uri	string	在重定向上，使用此值覆盖URL的Path部分。请注意，无论请求URI是否与精确路径或前缀匹配，都将替换整个路径。
authority	string	在重定向上，使用此值覆盖URL的Authority / Host部分。

## HTTPRetry

描述HTTP请求失败时要使用的重试策略。例如，以下规则将呼叫评级时的最大重试次数设置为3：v1 service，每次重试尝试超时2次。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - route:
    - destination:
        host: ratings.prod.svc.cluster.local
        subset: v1
    retries:
      attempts: 3
      perTryTimeout: 2s
      retryOn: gateway-error,connect-failure,refused-stream
```

领域	类型	描述
----	----	----

领域	类型	描述
<code>attempts</code>	<code>int32</code>	需要。给定请求的重试次数。重试之间的间隔将自动确定（25ms +）。实际重试次数取决于 <code>httpReqTimeout</code> 。
<code>perTryTimeout</code>	<code>google.protobuf.Duration2</code>	针对给定请求的每次重试尝试超时。格式：1h / 1m / 1s / 1ms。必须 $\geq 1\text{ms}$ 。
<code>retryOn</code>	<code>string</code>	指定重试发生的条件。可以使用“，”分隔列表指定一个或多个策略。见 <a href="#">支持政策4</a> 和 <a href="#">这里5</a> 了解更多详情。

## HTTPRewrite

HTTPRewrite可用于在将请求转发到目标之前重写HTTP请求的特定部分。重写原语只能用于 `HTTPRouteDestination`。以下示例演示了如何在进行实际API调用之前将api调用（`/ ratings`）的URL前缀重写为评级服务。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings-route
spec:
  hosts:
  - ratings.prod.svc.cluster.local
  http:
  - match:
    - uri:
        prefix: /ratings
    rewrite:
      uri: /v1/bookRatings
    route:
    - destination:
        host: ratings.prod.svc.cluster.local
        subset: v1
```

领域	类型	描述
<code>uri</code>	<code>string</code>	使用此值重写URI的路径（或前缀）部分。如果原始URI基于前缀匹配，则此字段中提供的值将替换相应的匹配前缀。
<code>authority</code>	<code>string</code>	使用此值重写Authority / Host标头。

# HTTPRoute

描述用于路由HTTP / 1.1, HTTP2和gRPC流量的匹配条件和操作。有关用法示例, 请参阅VirtualService。

领域	类型	描述
<code>match</code>	<code>HTTPMatchRequest[]</code>	匹配要激活的规则要满足的条件。单个匹配块内的所有条件都具有AND语义, 而匹配块列表具有OR语义。如果任何一个匹配块成功, 则匹配规则。
<code>route</code>	<code>HTTPRouteDestination[]</code>	http规则可以重定向或转发(默认)流量。转发目标可以是服务的多个版本之一(请参阅文档开头的术语表)。与服务版本关联的权重确定其接收的流量比例。
<code>redirect</code>	<code>HTTPRedirect</code>	http规则可以重定向或转发(默认)流量。如果在规则中指定了traffic passthrough选项, 则将忽略route / redirect。重定向原语可用于将HTTP 301重定向发送到不同的URI或授权。
<code>rewrite</code>	<code>HTTPRewrite</code>	重写HTTP URI和Authority标头。重写不能与Redirect原语一起使用。转发前将执行重写。
<code>timeout</code>	<code>google.protobuf.Duration</code>	HTTP请求超时。
<code>retries</code>	<code>HTTPRetry</code>	重试HTTP请求的策略。
<code>fault</code>	<code>HTTPFaultInjection</code>	故障注入策略应用于客户端的HTTP流量。请注意, 在客户端启用故障时, 将不会启用超时或重试。
<code>mirror</code>	<code>Destination</code>	除了将请求转发到预期目标之外, 还将HTTP流量镜像到另一个目标。镜像流量是尽力而为的, 其中边车/网关在从原始目的地返回响应之前不会等待镜像群集响应。将为镜像目标生成统计信息。
<code>corsPolicy</code>	<code>CorsPolicy</code>	跨源资源共享策略(CORS)。有关跨源资源共享的更多详细信息, 请参阅 <a href="#">CORS6</a> 。

领域	类型	描述
<code>appendHeaders</code>	<code>map&lt;string, string&gt;</code>	使用的 <code>append_headers</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>removeResponseHeaders</code>	<code>string[]</code>	使用的 <code>remove_response_header</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>appendResponseHeaders</code>	<code>map&lt;string, string&gt;</code>	使用的 <code>append_response_headers</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>removeRequestHeaders</code>	<code>string[]</code>	使用的 <code>remove_request_headers</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>appendRequestHeaders</code>	<code>map&lt;string, string&gt;</code>	使用的 <code>append_request_headers</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>headers</code>	<code>Headers</code>	标题操作规则

## HTTPRouteDestination

每个路由规则都与一个或多个服务版本相关联（请参阅文档开头的术语表）。与版本关联的权重确定其接收的流量比例。例如，以下规则将“评论”服务的25%流量路由到具有“v2”标记的实例，将剩余流量（即75%）路由到“v1”。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews.prod.svc.cluster.local
  http:
  - route:
    - destination:
        host: reviews.prod.svc.cluster.local
        subset: v2
        weight: 25
    - destination:
        host: reviews.prod.svc.cluster.local
        subset: v1
        weight: 75
```

以及相关的DestinationRule

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews-destination
spec:
  host: reviews.prod.svc.cluster.local
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

流量也可以分为两个完全不同的服务，而无需定义新的子集。例如，以下规则将25%的流量转发至 reviews.com 至 dev.reviews.com

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route-two-domains
spec:
  hosts:
  - reviews.com
  http:
  - route:
    - destination:
        host: dev.reviews.com
        weight: 25
    - destination:
        host: reviews.com
        weight: 75
```

领域	类型	描述
<code>destination</code>	<code>Destination</code>	需要。目标唯一地标识应将请求/连接转发到的服务的实例。
<code>weight</code>	<code>int32</code>	需要。要转发到服务版本的流量比例。（0–100）。跨目的地的权重总和应该= 100.如果规则中只有一个目的地，则权重值假定为 100。
<code>removeResponseHeaders</code>	<code>string[]</code>	使用的 <code>remove_response_header</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>appendResponseHeaders</code>	<code>map&lt;string, string&gt;</code>	使用的 <code>append_response_headers</code> 是过时了。请改用该 <code>headers</code> 字段。
<code>removeRequestHeaders</code>	<code>string[]</code>	使用的 <code>remove_request_headers</code> 是过时了。请改用该 <code>headers</code> 字段。

领域	类型	描述
appendRequestHeaders	map<string, string>	使用的append_request_headers是过时了。请改用该headers 字段。
headers	Headers	标题操作规则

# 头

## 标题操作规则

领域	类型	描述
request	Headers.HeaderOperations	在将请求转发到目标服务之前应用的标头操作规则
response	Headers.HeaderOperations	在将响应返回给调用者之前应用的标头操作规则

# Headers.HeaderOperations

## HeaderOperations描述要应用的标头操作

领域	类型	描述
set	map<string, string>	使用给定值覆盖key指定的标头
add	map<string, string>	将给定值附加到keys指定的标头（将创建以逗号分隔的值列表）
remove	string[]	删除指定的标头

# L4MatchAttributes

L4连接匹配属性。请注意，L4连接匹配支持不完整。



领域	类型	描述
destinationSubnets	string[]	具有可选子网的目标的IPv4或IPv6 IP地址。例如，abcd / xx表格或只是abcd
port	uint32	指定要寻址的主机上的端口。许多服务仅使用它们支持的协议公开单个端口或标签端口，在这些情况下，不需要显式选择端口。
sourceLabels	map<string, string>	一个或多个标签，用于将规则的适用性限制为具有给定标签的工作负载。如果VirtualService具有在顶部指定的网关列表，则它应包括保留的网关mesh，以使该字段适用。
gateways	string[]	应该应用规则的网关的名称。重写VirtualService顶部的网关名称（如果有）。网关匹配独立于sourceLabels。

## 百分

百分比指定[0.0,100.0]范围内的百分比。

领域	类型	描述
value	double	

## PortSelector

PortSelector指定用于匹配或选择最终路由的端口号。

领域	类型	描述
number	uint32 (oneof)	有效的端口号

# RouteDestination

L4路由规则加权目的地。

领域	类型	描述
destination	Destination	需要。目标唯一地标识应将请求/连接转发到的服务的实例。
weight	int32	需要。要转发到服务版本的流量比例。如果规则中只有一个目的地，则无论权重如何，所有流量都将路由到该目的地。

# StringMatch

描述如何匹配HTTP标头中的给定字符串。匹配区分大小写。

领域	类型	描述
exact	string (oneof)	确切的字符串匹配
prefix	string (oneof)	基于前缀的匹配
regex	string (oneof)	ECMAScript样式基于正则表达式的匹配

# TCPRoute

描述用于路由TCP流量的匹配条件和操作。以下路由规则将到达端口27017的流量mongo.prod.svc.cluster.local转发到端口5555上的另一个Mongo服务器。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo-Mongo
spec:
  hosts:
  - mongo.prod.svc.cluster.local
  tcp:
  - match:
    - port: 27017
    route:
    - destination:
      host: mongo.backup.svc.cluster.local
      port:
        number: 5555
```

领域	类型	描述
match	L4MatchAttributes[]	匹配要激活的规则要满足的条件。单个匹配块内的所有条件都具有AND语义，而匹配块列表具有OR语义。如果任何一个匹配块成功，则匹配规则。
route	RouteDestination[]	应将连接转发到的目标。

## TLSMatchAttributes

TLS连接匹配属性。

领域	类型	描述
sniHosts	string[]	需要。要匹配的SNI（服务器名称指示符）。通配符前缀可用于SNI值，例如*.com将匹配foo.example.com以及example.com。SNI值必须是相应虚拟服务器主机的子集（即，落在域内）。
destinationSubnets	string[]	具有可选子网的目标的IPv4或IPv6 IP地址。例如，abcd / xx表格或只是abcd
port	uint32	指定要寻址的主机上的端口。许多服务仅使用它们支持的协议公开单个端口或标签端口，在这些情况下，不需要显式选择端口。

领域	类型	描述
sourceLabels	map<string, string>	一个或多个标签，用于将规则的适用性限制为具有给定标签的工作负载。如果VirtualService具有在顶部指定的网关列表，则它应包括保留的网关mesh，以使该字段适用。
gateways	string[]	应该应用规则的网关的名称。重写VirtualService顶部的网关名称（如果有）。网关匹配独立于sourceLabels。

## TLSRoute

描述用于路由未终止的TLS流量的匹配条件和操作（TLS / HTTPS）以下路由规则基于SNI值将到达网关端口443的未终止的TLS流量（称为“mygateway”）转发到网格中的内部服务。

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo-sni
spec:
  hosts:
  - "*.bookinfo.com"
  gateways:
  - mygateway
  tls:
  - match:
    - port: 443
      sniHosts:
      - login.bookinfo.com
    route:
    - destination:
        host: login.prod.svc.cluster.local
  - match:
    - port: 443
      sniHosts:
      - reviews.bookinfo.com
    route:
    - destination:
        host: reviews.prod.svc.cluster.local
```

领域	类型	描述
match	TLSMatchAttributes[]	需要。匹配要激活的规则要满足的条件。单个匹配块内的所有条件都具有AND语义，而匹配块列表具有OR语义。如果任何一个匹配块成功，则匹配规则。
route	RouteDestination[]	应将连接转发到的目标。

# VirtualService

领域	类型	描述
<code>hosts</code>	<code>string[]</code>	<p>需要。正在向其发送流量的目标主机。可以是带有通配符前缀或IP地址的DNS名称。根据平台，也可以使用短名称而不是FQDN（即名称中没有点）。在这种情况下，将基于底层平台导出主机的FQDN。</p> <p>单个VirtualService可用于描述相应主机的所有流量属性，包括多个HTTP和TCP端口的流量属性。或者，可以使用多个VirtualService定义主机的流量属性，但有一些警告。有关详细信息，请参阅 <a href="#">操作指南</a>。</p> <p><i>Kubernetes用户注意事项：</i> 当使用短名称（例如“review”而不是“reviews.default.svc.cluster.local”）时，Istio将根据规则的命名空间而不是服务来解释短名称。包含主机“评论”的“默认”命名空间中的规则将被解释为“reviews.default.svc.cluster.local”，而不管与评论服务相关联的实际命名空间。为避免潜在的错误配置，建议始终在短名称上使用完全限定的域名。</p> <p>hosts字段适用于HTTP和TCP服务。网格内的服务，即服务注册表中的服务，必须始终使用其字母数字名称来引用。仅允许通过网关定义的服务使用IP地址。</p>
<code>gateways</code>	<code>string[]</code>	<p>应该应用这些路线的网关和边车的名称。单个VirtualService用于网格内的边车以及一个或多个网关。可以使用协议特定路由的匹配条件中的源字段覆盖此字段强加的选择条件。保留字<code>mesh</code>用于暗示网格中的所有边车。省略此字段时，将使用默认网关（<code>mesh</code>），这会将规则应用于网格中的所有边线。如果提供了网关名称列表，则规则将仅适用于网关。要将规则应用于网关和边车，请指定<code>mesh</code>其中一个网关名称。</p>
<code>http</code>	<code>HTTPRoute[]</code>	<p>HTTP流量的路由规则的有序列表。HTTP路由将应用于名为'http-'/http2- '/grpc- *'的平台服务端口，具有协议HTTP / HTTP2 / GRPC / TLS-terminated-HTTPS的网关端口和使用HTTP / HTTP2 / 的服务入口端口GRPC协议。使用匹配传入请求的第一个规则。</p>
<code>tls</code>	<code>TLSRoute[]</code>	<p>非终止TLS和HTTPS流量的路由规则的有序列表。通常使用ClientHello消息提供的SNI值来执行路由。TLS路由将应用于名为“https-”，“tls-”的平台服务端口，使用HTTPS / TLS协议的未终止网关端口（即具有“直通”TLS模式）和使用HTTPS / TLS协议的服务入口端口。使用匹配传入请求的第一个规则。注意：没有关联虚拟服务的流量“https-”或“tls-”端口将被视为不透明的TCP流量。</p>

领域	类型	描述
<code>tcp</code>	<code>TCPRoute[]</code>	不透明TCP流量的有序路由规则列表。TCP路由将应用于任何非HTTP或TLS端口的端口。使用匹配传入请求的第一个规则。
<code>exportTo</code>	<code>string[]</code>	<p>导出此虚拟服务的命名空间列表。导出虚拟服务允许由其他名称空间中定义的sidecars和网关使用。此功能为服务所有者和网格管理员提供了一种机制，用于控制跨命名空间边界的虚拟服务的可见性。</p> <p>如果未指定名称空间，则默认情况下将虚拟服务导出到所有名称空间。</p> <p>值“.”是保留的，并定义导出到声明虚拟服务的同一名称空间。类似地，值“*”是保留的，并定义导出到所有名称空间。</p> <p>注意：在当前版本中，该<code>exportTo</code>值仅限于“.”或“*”（即当前命名空间或所有命名空间）。</p>

---

## Links

1. [https://developer.mozilla.org/en-US/docs/Web/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS)
2. <https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#duration>
3. <https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#boolvalue>
4. [https://www.envoyproxy.io/docs/envoy/latest/configuration/http\\_filters/router\\_filter#x-envoy-retry-on](https://www.envoyproxy.io/docs/envoy/latest/configuration/http_filters/router_filter#x-envoy-retry-on)
5. [https://www.envoyproxy.io/docs/envoy/latest/configuration/http\\_filters/router\\_filter#x-envoy-retry-grpc-on](https://www.envoyproxy.io/docs/envoy/latest/configuration/http_filters/router_filter#x-envoy-retry-grpc-on)
6. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>