1. Istio
2. 文件
3. 参考
4. 组态
5. 交通管理
6. 目的地规则

目的地规则

13分钟阅读

`DestinationRule`定义在路由发生后应用于服务的流量的策略。这些规则指定负载平衡的配置，来自边车的连接池大小以及异常检测设置，以从负载平衡池中检测和驱逐不健康的主机。例如，评级服务的简单负载平衡策略如下所示：

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
```

可以通过定义命名`subset`并覆盖服务级别指定的设置来指定特定于版本的策略 。以下规则对进入名为testversion的子集的所有流量使用循环负载平衡策略，该子集由具有标签的端点（例如，pod）组成（版本：v3）。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
  subsets:
  - name: testversion
    labels:
      version: v3
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
```

注意：在路由规则明确向此子集发送流量之前，为子集指定的策略不会生效。

流量策略也可以定制到特定端口。以下规则对端口80的所有流量使用最小连接负载平衡策略，而对端口9080的流量使用循环负载平衡设置。

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings-port
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy: # Apply to all ports
    portLevelSettings:
    - port:
        number: 80
      loadBalancer:
        simple: LEAST_CONN
    - port:
        number: 9080
      loadBalancer:
        simple: ROUND_ROBIN
```

ConnectionPoolSettings

上游主机的连接池设置。这些设置适用于上游服务中的每个主机。有关 详细信息，请参阅Envoy的断路器1。连接池设置可以在TCP级别以及HTTP级别应用。

例如，以下规则设置了与redis服务的100个连接的限制，称为myredissrv，连接超时为30ms

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-redis
spec:
  host: myredissrv.prod.svc.cluster.local
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
        connectTimeout: 30ms
        tcpKeepalive:
          time: 7200s
          interval: 75s
```

领域	类型	描述
tcp	ConnectionPoolSettings.TCPSettings	HTTP和TCP上游连接共有的设置。
http	ConnectionPoolSettings.HTTPSettings	HTTP连接池设置。

ConnectionPoolSettings.HTTPSettings

适用于HTTP1.1 / HTTP2 / GRPC连接的设置。

领域	类型	描述
http1MaxPendingRequests	int32	到目标的最大待处理HTTP请求数。默认1024。
http2MaxRequests	int32	对后端的最大请求数。默认1024。

领域	类型	描述
maxRequestsPerConnection	int32	每个连接到后端的最大请求数。将此参数设置为1将禁用保持活动状态。
maxRetries	int32	在给定时间内群集中所有主机可以执行的最大重试次数。默认为1024。
idleTimeout	google.protobuf.Duration2	上游连接池连接的空闲超时。空闲超时定义为没有活动请求的时间段。如果未设置，则没有空闲超时。达到空闲超时后，将关闭连接。请注意，基于请求的超时意味着HTTP / 2 PING不会保持连接活动。适用于HTTP1.1和HTTP2连接。

ConnectionPoolSettings.TCPSettings

HTTP和TCP上游连接共有的设置。

领域	类型	描述
maxConnections	int32	到目标主机的最大HTTP1 / TCP连接数。默认1024。
connectTimeout	google.protobuf.Duration2	TCP连接超时。
tcpKeepalive	ConnectionPoolSettings.TCPSettings.TcpKeepalive	如果设置，则在套接字上设置SO_KEEPALIVE以启用TCP Keepalive。

ConnectionPoolSettings.TCPSettings.TcpKeepalive

TCP keepalive。

领域	类型	描述
probes	uint32	在决定连接失效之前，无响应发送的最大keepalive探测数。默认是使用操作系统级别配置（除非被覆盖，Linux默认为9.）
time	google.protobuf.Duration2	在保持活动探测开始发送之前连接需要空闲的持续时间。默认是使用操作系统级别配置（除非被覆盖，Linux默认为7200s（即2小时）。
interval	google.protobuf.Duration2	保持活动探针之间的持续时间。默认是使用操作系统级别配置（除非被覆盖，Linux默认为75秒。）

DestinationRule

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ServiceEntries声明的主机中查找服务名称。为服务注册表中不存在的服务定义的规则将被忽略。 *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。* 请注意，host字段适用于HTTP和TCP服务。
`trafficPolicy`	TrafficPolicy	要应用的流量策略（负载均衡策略，连接池大小，异常检测）。
`subsets`	Subset[]	一个或多个命名集，表示服务的各个版本。可以在子集级别覆盖流量策略。
`exportTo`	`string[]`	导出此目标规则的命名空间列表。要应用于服务的目标规则的解析发生在命名空间层次结构的上下文中。导出目标规则允许将其包含在其他命名空间中的服务的解析层次结构中。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目标规则的可见性。 如果未指定名称空间，则默认情况下将目标规则导出到所有名称空间。 值“。”是保留的，并定义导出到声明目标规则的同一名称空间。类似地，值“*”是保留的，并定义导出到所有名称空间。 注意：在当前版本中，该`exportTo`值仅限于“。”或“*”（即当前命名空间或所有命名空间）。

LoadBalancerSettings

负载平衡策略以应用于特定目标。有关 详细信息，请参阅Envoy的负载平衡 文档3。

例如，以下规则对进入评级服务的所有流量使用循环负载平衡策略。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: ROUND_ROBIN
```

以下示例使用用户cookie作为哈希键，为相同评级服务的基于评级服务哈希的负载均衡器设置粘性会话。

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      consistentHash:
        httpCookie:
          name: user
          ttl: 0s
```

领域	类型	描述
simple	LoadBalancerSettings.SimpleLB (oneof)	
consistentHash	LoadBalancerSettings.ConsistentHashLB (oneof)	

LoadBalancerSettings.ConsistentHashLB

一致的基于散列的负载平衡可用于基于HTTP头，cookie或其他属性提供软会话亲和性。此负载平衡策略仅适用于HTTP连接。在目标服务中添加/删除一个或多个主机时，对特定目标主机的亲缘关系将丢失。

领域	类型	描述
httpHeaderName	string (oneof)	基于特定HTTP标头的哈希。
httpCookie	LoadBalancerSettings.ConsistentHashLB.HTTPCookie (oneof)	哈希基于HTTP cookie。
useSourceIp	bool (oneof)	散列基于源IP地址。
minimumRingSize	uint64	用于哈希环的最小虚拟节点数。默认为1024.较大的环大小会导致更精细的负载分布。如果负载平衡池中的主机数大于环大小，则将为每个主机分配一个虚拟节点。

LoadBalancerSettings.ConsistentHashLB.HTTPCookie

描述将用作Consistent Hash负载均衡器的哈希键的HTTP cookie。如果cookie不存在，将生成它。

领域	类型	描述
name	string	需要。Cookie的名称。
path	string	为cookie设置的路径。
ttl	google.protobuf.Duration2	需要。cookie的生命周期。

LoadBalancerSettings.SimpleLB

标准负载平衡算法，无需调整。

名称	描述
ROUND_ROBIN	循环政策。默认
LEAST_CONN	最少请求负载均衡器使用O（1）算法，该算法选择两个随机健康主机并选择具有较少活动请求的主机。
RANDOM	随机负载均衡器选择随机健康主机。如果未配置健康检查策略，则随机负载平衡器通常比循环执行得更好。
PASSTHROUGH	此选项将连接转发到调用方请求的原始IP地址，而不进行任何形式的负载平衡。必须小心使用此选项。它适用于高级用例。有关详细信息，请参阅Envoy中的原始目标负载均衡器。

OutlierDetection

断路器实现，用于跟踪上游服务中每个主机的状态。适用于HTTP和TCP服务。对于HTTP服务，持续返回API调用的5xx错误的主机将在池中弹出预定义的时间段。对于TCP服务，在测量连续错误度量时，给定主机的连接超时或连接失败将计为错误。有关 更多详细信息，请参阅Envoy的异常值检测4。

以下规则设置连接池大小为100个连接和1000个并发HTTP2请求，与“评论”服务的请求/连接不超过10个。此外，它配置每5分钟扫描一次的上游主机，这样任何连续7次出现5XX错误代码的主机都将被弹出15分钟。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews-cb-policy
spec:
  host: reviews.prod.svc.cluster.local
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
      http:
        http2MaxRequests: 1000
        maxRequestsPerConnection: 10
    outlierDetection:
      consecutiveErrors: 7
      interval: 5m
      baseEjectionTime: 15m
```

领域	类型	描述
consecutiveErrors	int32	从连接池中弹出主机之前的错误数。默认为5.当通过HTTP访问上游主机时，502,503或504返回代码有资格作为错误。通过不透明的TCP连接访问上游主机时，连接超时和连接错误/失败事件有资格作为错误。

领域	类型	描述
interval	google.protobuf.Duration2	喷射扫描分析之间的时间间隔。格式：1h / 1m / 1s / 1ms。必须> = 1ms。默认值是10秒。
baseEjectionTime	google.protobuf.Duration2	最短射血持续时间 主机将保持弹出的时间等于最小弹射持续时间和主机弹出次数的乘积。该技术允许系统自动增加不健康上游服务器的弹射周期。格式：1h / 1m / 1s / 1ms。必须> = 1ms。默认值为30秒。
maxEjectionPercent	int32	负载平衡池中可以弹出的上游服务的最大主机百分比。默认为10%。
minHealthPercent	int32	只要关联的负载平衡池在健康模式下具有至少最小*健康*百分比主机，就会启用异常值检测。当负载平衡池中健康主机的百分比降至此阈值以下时，将禁用异常值检测，并且代理将在池中的所有主机（健康和不健康）之间进行负载平衡。默认值为50%。

子集

服务端点的子集。子集可用于A / B测试等场景，或路由到特定版本的服务。有关在这些方案中使用子集的示例，请参阅VirtualService文档。此外，可以在子集级别覆盖在服务级别定义的流量策略。以下规则对进入名为testversion的子集的所有流量使用循环负载平衡策略，该子集由具有标签的端点（例如，pod）组成（版本：v3）。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
  subsets:
  - name: testversion
    labels:
      version: v3
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
```

注意：在路由规则明确向此子集发送流量之前，为子集指定的策略不会生效。

通常需要一个或多个标签来标识子集目的地，然而，当对应的DestinationRule表示支持多个SNI主机（例如，出口网关）的主机时，没有标签的子集可能是有意义的。在这种情况下，具有TLSSettings的流量策略 可用于识别与指定子集相对应的特定SNI主机。

领域	类型	描述
name	string	需要。子集的名称。服务名称和子集名称可用于路由规则中的流量分割。
labels	map<string, string>	标签在服务注册表中的服务端点上应用筛选器。有关用法示例，请参阅路由规则。

领域	类型	描述
trafficPolicy	TrafficPolicy	适用于此子集的流量策略。子集继承DestinationRule级别指定的流量策略。在子集级别指定的设置将覆盖在DestinationRule级别指定的相应设置。

TLSSettings

上游连接的SSL / TLS相关设置。有关 详细信息，请参阅Envoy的TLS上下文5。这些设置对HTTP和TCP上游都是通用的。

例如，以下规则将客户端配置为使用相互TLS连接到上游数据库集群。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: db-mtls
spec:
  host: mydbserver.prod.svc.cluster.local
  trafficPolicy:
    tls:
      mode: MUTUAL
      clientCertificate: /etc/certs/myclientcert.pem
      privateKey: /etc/certs/client_private_key.pem
      caCertificates: /etc/certs/rootcacerts.pem
```

以下规则将客户端配置为在与域名匹配* .foo.com的外部服务进行通信时使用TLS。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: tls-foo
spec:
  host: "*.foo.com"
  trafficPolicy:
    tls:
      mode: SIMPLE
```

以下规则将客户端配置为在与评级服务交谈时使用Istio相互TLS。

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: ratings-istio-mtls
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

领域	类型	描述
mode	TLSSettings.TLSmode	REQUIRED：指示是否应使用TLS保护与此端口的连接。此字段的值确定如何强制执行TLS。
clientCertificate	string	如果模式是必需的MUTUAL。保存客户端TLS证书的文件的路径。如果是模式，则应为空ISTIO_MUTUAL。

领域	类型	描述
privateKey	string	如果模式是必需的MUTUAL。保存客户端私钥的文件的路径。如果是模式，则应为空ISTIO_MUTUAL。
caCertificates	string	可选：包含证书颁发机构证书的文件的路径，用于验证显示的服务器证书。如果省略，代理将不验证服务器的证书。如果是模式，则应为空ISTIO_MUTUAL。
subjectAltNames	string[]	用于验证证书中主题标识的备用名称列表。如果指定，代理将验证服务器证书的主题alt名称是否与指定值之一匹配。如果指定，则此列表将覆盖ServiceEntry 中主题*alt*名称的值。
sni	string	在TLS握手期间向服务器提供的SNI字符串。

TLSSettings.TLSmode

TLS连接模式

名称	描述
DISABLE	不要设置到上游端点的TLS连接。
SIMPLE	发起与上游端点的TLS连接。
MUTUAL	通过提供客户端证书进行身份验证，使用相互TLS保护与上游的连接。
ISTIO_MUTUAL	通过提供客户端证书进行身份验证，使用相互TLS保护与上游的连接。与Mutual模式相比，此模式使用Istio自动生成的证书进行mTLS身份验证。使用此模式时，所有其他字段TLSSettings应为空。

TrafficPolicy

跨所有目标端口申请特定目的地的流量策略。有关示例，请参阅DestinationRule。

领域	类型	描述
loadBalancer	LoadBalancerSettings	控制负载均衡器算法的设置。
connectionPool	ConnectionPoolSettings	控制与上游服务的连接量的设置
outlierDetection	OutlierDetection	控制从负载平衡池中驱逐不健康主机的设置
tls	TLSSettings	与上游服务连接的TLS相关设置。

领域	类型	描述
portLevelSettings	TrafficPolicy.PortTrafficPolicy[]	针对各个端口的流量策略。请注意，端口级别设置将覆盖目标级别设置。当被端口级设置覆盖时，将不会继承目标级别指定的流量设置，即默认值将应用于端口级流量策略中省略的字段。

TrafficPolicy.PortTrafficPolicy

适用于服务的特定端口的流量策略

领域	类型	描述
port	PortSelector6	指定要应用此策略的目标服务上的端口名称或端口号。 名称必须符合DNS标签语法（rfc1035），因此不能与数字冲突。如果具有相同协议的服务上有多个端口，则名称应为<protocol-name> - <DNS label>形式。
loadBalancer	LoadBalancerSettings	控制负载均衡器算法的设置。
connectionPool	ConnectionPoolSettings	控制与上游服务的连接量的设置
outlierDetection	OutlierDetection	控制从负载平衡池中驱逐不健康主机的设置
tls	TLSSettings	与上游服务连接的TLS相关设置。

Links

1. https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/circuit_breaking
2. https://developers.google.com/protocol-buffers/docs/reference/google.protobuf#duration
3. https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/load_balancing
4. https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/outlier
5. https://www.envoyproxy.io/docs/envoy/latest/api-v2/api/v2/auth/cert.proto.html
6. https://istio.io/docs/reference/config/networking/v1alpha3/virtual-service.html#PortSelector