目录

负载均衡监听器概述

创建负载均衡器后，需要为负载均衡器配置监听器。监听器负责监听负载均衡实例上的请求，执行策略分发流量至后端真实服务器。同时，负载均衡监听器还可以配置相应的 **会话保持** 和 **健康检查** 策略。

负载均衡监听器可以通过监听负载均衡实例上的四层和七层请求，并将这些请求分发到后端服务器上进行处理。四层和七层负载均衡的区别，主要体现在对后台的服务器进行负载均衡时，是依据四层的信息还是七层的信息来决定如何转发流量。其中四层为传输层协议，主要通过 VIP 结合端口接受请求并分配流量到后端服务器；七层为应用层协议，则基于 URL、HTTP 头部等应用层信息进行流量分发。

支持的协议类型

典型的 Web 应用程序之间的通信需要经由网络的各个分层，每层都会提供特定的通信功能。依据开放式系统互联（Open System Interconnect，OSI）网络模型，各个分层中都有标准的通信格式。金山云负载均衡涉及网络模型中的 四层（传输层） 和 七层（应用层）。

金山云负载均衡支持以下协议的请求转发：

公网负载均衡

- HTTP：全部地域
- HTTPS：全部地域
- TCP：全部地域
- UDP：香港地区暂不支持 UDP 协议转发

私网负载均衡

- HTTP：华北1（北京）、华东1（上海）
- TCP：全部地域
- UDP：香港地区暂不支持 UDP 协议转发

四层协议

如果使用四层协议转发，负载均衡实例会直接将请求转发到后端实例，而不修改任何数据包。负载均衡收到请求之后，会尝试在监听器配置中指定的端口上打开与后端实例的 TCP 连接。

七层协议

如果前端和后端连接均使用七层协议转发，负载均衡器会解析请求中有意义的应用层内容，并根据其内容选择后端服务器。因此，七层负载均衡器需要先代理后端服务器和客户端建立连接（三次握手）后，才可能接受到客户端发送的真正应用层内容的报文，然后再根据该报文中的特定字段，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。

HTTPS 监听器的安全机制

HTTPS 是一种安全的 HTTP 连接，通过使用证书来确保服务器端和客户端的可信度。负载均衡通过证书解密来自客户端的请求，然后再将请求发送到后端实例上。有关更多信息，请参阅 证书管理 相关内容。

四层负载均衡和七层负载均衡的区别

四到七层负载均衡，就是在对后台的服务器进行负载均衡时，依据四层的信息或七层的信息来决定如何转发流量。

四层的负载均衡，就是通过三层的 IP 地址（VIP），然后加四层的端口号，来决定哪些流量需要做负载均衡，对需要处理的流量进行网络地址转换处理，转发至后台服务器。

七层的负载均衡，就是在四层的基础上，再考虑应用层的特征（如 HTTP 头部、URL 等），比如同一个 Web 服务器的负载均衡，除了根据 VIP 加 80 端口辨别是否需要处理的流量，还可根据七层的 URL 来决定是否要进行负载均衡。七层负载均衡也称为“内容交换”，也就是主要通过报文中的真正有意义的应用层内容，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。 负载均衡设备如果要根据真正的应用层内容再选择服务器，只能先代理最终的服务器和客户端建立连接（三次握手）后，才可能接受到客户端发送的真正应用层内容的报文，然后再根据该报文中的特定字段，再加上负载均衡设备设置的服务器选择方式，决定最终选择的内部服务器。负载均衡设备在这种情况下，更类似于一个代理服务器。负载均衡和前端的客户端以及后端的服务器会分别建立 TCP 连接。

协议支持及端口配置

金山云支持 四层转发 和 七层转发 两种监听器模式，针对不同的协议类型，分别作用于网络模型中的传输层和应用层。

四层转发监听器协议及端口配置

负载均衡监听器监听负载均衡实例上的四层请求，并将 TCP 请求分发至后端服务器进行处理。四层转发能力通过以下配置实现：

前后端协议	前端端口（负载均衡端口）	后端端口（服务器端口）	备注
四层协议，即 TCP/UDP	负载均衡器对外或对内提供服务时接收请求的前端端口。端口范围：1-65535 端口，其中 445 端口不可用	接受负载均衡分发流量的端口。在同一个负载均衡实例中，一个负载均衡端口可以对应多个云服务器端口	前端端口在同一个负载均衡实例内不可以重复。后端端口在同一个负载均衡实例/同一监听器内可以重复，同一个监听器内的同一个真实服务器的端口不可重复。

七层转发监听器协议及端口配置

负载均衡监听器监听负载均衡实例上的七层请求，并将 HTTP(S) 请求分发至后端服务器进行处理。七层转发能力通过以下配置实现：

前后端协议	前端端口（负载均衡端口）	后端端口（服务器端口）	备注
七层协议，即 HTTP/HTTPS	负载均衡器对外或对内提供服务时接收请求的前端端口。端口范围：1-65535 端口，其中 445 端口不可用	接受负载均衡分发流量的端口。在同一个负载均衡中，一个负载均衡端口可以对应多个云服务器端口	前端端口在同一个负载均衡实例内不可以重复。后端端口在同一个负载均衡实例/同一监听器内可以重复，同一个监听器内的同一个真实服务器的端口不可重复。

转发方式

转发方式是指负载均衡向后端服务器分配流量的算法，根据不同的转发方式及后端服务器的权重设置，可以达到不同的效果。

加权轮询算法

加权轮询算法就是以轮叫的方式依次将请求调度不同的服务器。加权轮询调度算法可以解决服务器间性能不一致的情况，它用相应的权值表示服务器的处理性能，按权值的高低和轮询方式分配请求到各服务器。权值高的服务器先收到连接，权值高的服务器比权值低的服务器处理更多的连接，相同权值的服务器处理相同数目的连接。

优势：算法简洁。无需记录当前所有连接的状态，所以它是一种无状态调度。

劣势：不适用于请求服务时间变化比较大，或者每个请求所消耗的时间不一致的情况，此时轮询调度算法容易导致服务器间的负载不平衡。

适用场景：每个请求所占用的后端服务器时间基本相同，常用于短连接服务，例如 HTTP 等服务。

加权最小连接数算法

在实际情况中，客户端的每一次请求在服务器停留的时间可能会有较大的差异，随着工作时间的延伸，如果采用简单的轮询算法，每一台服务器上的连接进程数目可能会产生极大的不同，这样实际上并没有达到真正的负载均衡。

最小连接数调度是一种动态调度算法，它通过服务器当前活跃的连接数来估计服务器的负载情况。调度器需要记录各个服务器已建立连接的数目，当一个请求被调度到某台服务器，其连接数加一；当连接中止或超时，其连接数减一。

加权最小连接数算法是在最小连接数调度算法的基础上，根据服务器的不同处理能力，给每个服务器分配不同的权值，使其能够接受相应权值数的服务请求，是在最小连接数调度算法基础上的改进。

1) 假设各台后端服务器的权值依次为 wi，当前连接数依次为 ci ，依次计算 ci/wi ，值最小的后端服务器实例作为下一个分配的实例。

2) 如果存在 ci/wi 相同的后端服务器实例，再使用加权轮询的方式调度。

优势：此种均衡算法适合处理长时的请求服务，如 FTP 等应用。

劣势：相较于加权轮询算法，加权最小连接数算法需要保存服务器现有的连接数目，它是一种有状态调度。

适用场景：每个请求所占用的后端时间相差较大的场景，常用于长连接服务。如 2ms 和 2s 这种数量级的差距时，推荐使用加权最小连接数算法实现负载均衡。

主备转发

若您对传统主备模式依赖较强，需要利用两台服务器实现主备模式，使用主备转发策略即可满足需求。

后端服务器中仅有一台主机和一台备机。当主机工作正常时，流量将转发至主机；若主机宕机，流量将切换至备机。

使用限制

1.主备服务器组仅可适用于 TCP 和 UDP 监听上。

2.主备服务器组只能添加两台后端服务器，且为一主一备。

3.主备转发规则设置成功后不可与其他转发方式互相切换。

4.主备转发方式下，会话保持配置不生效。

注意：

目前仅以下地区支持主备转发算法：

- 华东1（上海）
- 华南1（广州）
- 华北1（北京）
- 华北金融1（北京）
- 华东金融1（上海）
- 华北政务1（北京）

基于域名/URL路径转发

7层负载均衡支持配置基于域名和路径的转发策略。可以将来自不同域名或URL路径的请求转发给不同的后端服务器组，合理分配服务器资源。

URL可以配置独立于监听器的转发方式、会话保持和健康检查规则。开关控制URL是否继承监听器配置，若开关打开，继承监听器配置；若关闭，以URL下的新配置的转发方式、会话保持和健康检查规则为准。

域名和路径转发介绍

域名规则：支持精确域名匹配，支持字符集：字母（a-z A-Z）、数字（0-9）、点（.）、连字符（-），如：www.ksyun.com。可添加空域名，空域名下，URL规则不可配置“/”。

URL规则：URL 必须以/开头，支持字符集：字母（a-z A-Z ）、数字（0-9）、特殊字符（. - / % ? # &） 。URL转发会按照前缀最长匹配原则，例如有/abc和/abcd两个规则，访问/abcde，优先匹配/abcd规则。

在一个监听下添加多条转发策略，每条转发策略关联不同的后端服务器组。例如您可以将所有读请求转发到一组后端服务器上，写请求转发到另一组后端服务器上，这样可以更灵活地分配资源。

每个域名都可绑定自身的证书。若添加了空域名，自动采用监听器证书，且不可调整。监听器与其下域名使用的证书可重复。

配额（暂不支持调整）

配额项	值
域名	每监听器 4 个
URL	每域名 15 个

转发策略

配置转发策略后，7 层负载均衡匹配策略如下：

1.首先判断报文中是否包含域名，例如：www.ksyun.com；

2.若匹配不到该域名，则将流量转发到监听器后端服务器；若存在匹配该域名的转发策略，则继续匹配URL部分；

3.若URL部分也能匹配，则将请求转发到对应的后端服务器组；若URL部分未能命中该域名下的任何规则，流量转发至监听器后端服务器。

配置域名和路径转发

添加域名和 URL 路径转发策略，参考以下步骤：

1. 登录 负载均衡控制台 。
2. 进入监听器页面，选择目标七层监听器，点击**添加转发策略**。
3. 根据以下规则配置域名和 URL 路径转发策略：

   域名：输入请求域名。支持字符：字母（a-z A-Z）、数字（0-9）、点（.）、连字符（-），如：www.ksyun.com。
   URL：输入请求路径。必须以/开头，支持字符：字母（a-z A-Z ）、数字（0-9）、特殊字符（. - / % ? # &）。
   真实服务器组：选择该转发策略关联的后端服务器组。
   同步监听器配置：选择是否同步监听器的转发方式、会话保持和健康检查规则。

4. 点击**确定**添加本条转发策略。
5. 您也可以通过选中目标监听器后，在弹出的监听器详情面板中，选中转发策略添加新的域名或路径。

编辑域名和 URL 路径转发策略，参考以下步骤：

1. 登录 负载均衡控制台 。
2. 进入监听器页面，选中目标七层监听器，在弹出的监听器详情面板中，选中转发策略。
3. 在转发策略列表中，点击目标转发策略的**编辑**完成编辑。

会话保持

会话保持可使得来自同一 IP（网段）的请求被转发到同一台后端服务器上。默认情况下，负载均衡会将每个请求分别路由到不同后端服务器实例负载。但是，您可以使用会话保持功能使特定用户的请求被路由到同一台后端服务器实例上。

四层会话保持

四层转发支持简单会话保持能力，可以设置会话保持的时间，超过该时间阀值，会话中无新请求则断开连接。

七层会话保持

HTTP/HTTPS 监听可使用植入 cookie 和重写 cookie 来进行会话保持。

植入cookie

植入 cookie 是指由负载均衡服务器来给客户端设置 cookie，即 HTTP/HTTPS 响应报文中插入 SERVERID 字串和客户配置时指定的超时时间，在此时间内会将同一客户端的请求传入到同一个后端服务器，当客户端浏览器再次通过此 cookie 访问时，负载均衡不会传给后端的服务器，即插入 cookie 关键字与值对后端服务器来说是不需要知道的。

重写cookie

重写 cookie 是指负载均衡实例的拥有者可以按照自己的需要自定义在后端的服务器回复 HTTP/HTTPS 响应中插入 cookie 关键字与值，后端的服务器上同时需要维护此 cookie 的会话保持时间，在此响应报文经过负载均衡时，负载均衡会基于一定规则重写 cookie 用于会话保持，当携带 cookie 关键字与值的请求到来时会将此 cookie 关键字与值传入到初始插入 cookie 的后端服务器；但与初始相比 cookie 值的内容已经改变。

长连接和会话保持的关系

场景1：HTTP 七层业务

假设客户端端访问是 HTTP/1.1 协议，头部信息中设置 Connection:keep-alive 。通过 SLB，再访问到后端 KEC，此时不开会话保持，下一次访问，能否访问到同一台 KEC？

答：不能。

HTTP keep-alive 是由客户端跟 SLB 建立的，若此时没有开启 cookie 会话保持，则下一次访问，SLB会根据轮询策略，随机挑选后端的一台 KEC，此前的长连接等于白费了。

因此建议开启会话保持。

场景2：TCP 四层业务

假设客户端发起访问，传输层协议是 TCP，启用长连接。但没有开基于源 IP 的会话保持。下一次访问，同一个客户端，能否访问到同一个机器？

答：不一定。

首先，根据四层的实现机制，当 TCP 启用长连接时，如果该长连接一直没有断开，前后两次访问都是同一条连接，则可以访问到同一台机器。如果第二次访问时，第一条连接由于其他原因（网络重启、连接超时）被释放，这时第二次访问就有可能调度到其他后端云服务器上。且长连接默认全局的超时时间是 900s，即若没有新请求，则释放。

健康检查

金山云负载均衡实例可以定期向后端服务器发送 Ping、尝试连接或发送请求来测试后端服务器运行的状况，这些测试称为健康检查。

当后端服务器实例被判定为不健康时，负载均衡实例将不会把请求转发到该实例上。但健康检查会对所有后端服务器进行，当不健康实例恢复健康状态时，负载均衡实例将恢复把新的请求转发给它。

健康检查配置字段的含义

健康检查：选择是否开启健康检查服务。

健康检查间隔：进行健康检查的时间间隔。

健康阈值：如果连续 n 次（n 为填写的数值）收到了健康检查结果为成功状态，则识别为健康。

不健康阈值： 如果连续 n 次(n 为填写的数值)收到了健康检查结果失败状态，则识别为不健康。

HTTP请求链接：仅在 HTTP(S) 检查方式时才有，HTTP 协议的健康检查将使用 HEAD 方法请求该 URL，因为需要确保您填写的连接在相应 HEAD 方法时能正确返回。

域名：仅在 HTTP(S) 检查方式时才有，健康检查的服务域名，系统默认使用 VIP 作为 host。

四层转发健康检查配置

四层转发的健康检查机制由负载均衡器向配置中指定的服务器端口发起访问请求，如果端口访问正常则视为后端服务器运行正常，否则视为后端服务器运行异常。对于 TCP 的业务，使用 SYN 包进行探测。

- 健康检查：开启
- 检查间隔： 2-300 秒，默认为 5s
- 健康阈值：2-10 次，默认为 5 次（不健康后端服务器出现此指定次数响应超时后，视为健康）
- 不健康阈值：2-10 次，默认为 4 次 （健康后端服务器出现此指定次数响应超时后，视为不健康）

七层转发健康检查配置

七层转发的健康检查机制由负载均衡器向后端服务器发送 HTTP 请求来检测后端服务，负载均衡器会通过 HTTP 返回值是否为预设的值来判断服务是否正常。

- 健康检查：开启
- 检查间隔： 2-300 秒，默认为 5s
- 健康阈值：2-10 次，默认为 5 次（不健康后端服务器出现此指定次数响应超时后，视为健康）
- 不健康阈值：2-10 次，默认为 4 次 （健康后端服务器出现此指定次数响应超时后，视为不健康）

证书配置

证书格式要求

- 用户要申请的证书为：linux 环境下 pem 格式的证书。
- 如果是通过 root CA 机构颁发的证书， 您拿到的证书为唯一的一份，不需要额外的证书，配置的站点即可被浏览器等访问设备认为可信。
- 如果是通过中级 CA 机构颁发的证书，您拿到的证书文件包含多份证书，需要人为的将服务器证书与中间证书合并在一起上传。
- 当您的证书有证书链时，请将证书链内容，转化为 PEM 格式内容，与证书内容合并上传。
- 拼接规则为：服务器证书放第一份，中间证书放第二份，中间不要有空行。注：一般情况下，机构在颁发证书的时候会有对应说明， 请注意规则说明。

以下为证书格式和证书链格式范例，请确认格式正确后上传：

1. root CA 机构颁发的证书：证书格式为 linux 环境下 pem 格式。样例如下：

证书规则为：

- [————BEGIN CERTIFICATE————, ————END CERTIFICATE————] 开头和结尾；请将这些内容一并上传；
- 每行64字符，最后一行不超过64字符；

2. 中级机构颁发的证书链： -----BEGIN CERTIFICATE----- -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- -----END CERTIFICATE----- -----BEGIN CERTIFICATE----- -----END CERTIFICATE-----

证书链规则：

- 证书之间不能有空行；
- 每一份证书遵守第一点关于证书的格式说明；

RSA私钥格式要求

样例如下：

rsa私钥规则：

- [-----BEGIN RSA PRIVATE KEY-----, -----END RSA PRIVATE KEY-----] 开头结尾；请将这些内容一并上传；
- 每行64字符，最后一行长度可以不足64字符。

如果您不是按照上述方案生成私钥，得到[-----BEGIN PRIVATE KEY-----, -----END PRIVATE KEY-----] 这种样式的私钥，您可以按照如下方式转换：

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

然后将 new_server_key.pem 的内容与证书一起上传。

证书转换为 PEM 格式说明

目前负载均衡只支持 PEM 格式的证书，其他格式的证书需要转换成 PEM 格式后才能上传到负载均衡中，建议通过 openssl 工具进行转换。下面是几种比较流行的证书格式转换为PEM格式的方法。

DER转换为 PEM

DER格式一般出现在 java 平台中。

证书转换：`openssl x509 -inform der -in certificate.cer -out certificate.pem`

私钥转换：`openssl rsa -inform DER -outform PEM -in privatekey.der -out privatekey.pem`

P7B转换为 PEM

P7B 格式一般出现在windows server和tomcat中。

证书转换：`openssl pkcs7 -print_certs -in incertificat.p7b -out outcertificate.cer`

获取 outcertificat.cer 里面 [-----BEGIN CERTIFICATE-----， -----END CERTIFICATE-----] 的内容作为证书上传。

私钥转换：无私钥

PFX转换为 PEM

PFX 格式一般出现在 windows server 中。

证书转换：`openssl pkcs12 -in certname.pfx -nokeys -out cert.pem`

私钥转换：`openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes`