**负载均衡即服务**

**有v1和v2两个版本，v2版为成熟插件**

OpenStack网络服务Neutron-LBaaS服务插件提供了一个名为“LBaaS v2”的负载均衡器特性。

LBaaS v2将侦听器的概念添加到LBaaS v1负载均衡器中。LBaaS v2允许您在单个负载均衡器IP地址上配置多个侦听器端口。

LBaaS v2有两个参考实现。一个是基于代理的基于HAProxy的实现。代理处理HAProxy配置并管理HAProxy守护进程。另一个LBaaS v2实现，Octavia，有一个单独的API和独立的工作进程，在虚拟机中构建负载均衡器，这些虚拟机是由计算服务管理的虚拟机。您不需要Octavia的代理。

|  |
| --- |
| **注意：**  **LBaaS v1在Newton发行版本中被移除。以下链接提供了关于LBaaS v1如何工作以及如何配置它的更多细节:**  **[Load-Balancer-as-a-Service(LBaaS)概述](http://docs.openstack.org/admin-guide/networking-introduction.html" \l "load-balancer-as-a-service-lbaas-overview)**  **[基本的Load-Balancer-as-a-Service操作](http://docs.openstack.org/admin-guide/networking-adv-features.html" \l "basic-load-balancer-as-a-service-operations)** |

|  |
| --- |
| **警告：**  **目前，v1和v2负载均衡器之间没有迁移路径。如果您选择从v1转换到v2，则必须重新创建所有负载均衡器、池和健康监视器。** |

**LBaaS v2的概念**

LBaaS v2有几个新概念需要理解:

IMG_256

图中每一块都为openstack所管理的块

**负载均衡器（Load Balancer）**

负载均衡器占据一个Neutron网络端口，并有一个从子网分配的IP地址。

**侦听器（Listener）**

负载均衡器可以侦听多个端口上的请求。每个端口都由侦听器指定。

**池（Pool）**

池中包含通过负载均衡器服务内容的成员列表。

**成员（Member）**

成员是在负载均衡器背后提供流量服务的服务器。每个成员由其用于服务流量的IP地址和端口指定。

**健康状况监视器（Health Monitor）**

成员们可能会时不时地下线，而健康监测器则会分流那些没有正确反应的会员。健康监测器与泳池有关。

LBaaS v2通过不同的服务插件有多个实现。两种最常见的实现使用代理或Octavia服务。这两种实现都使用[LBaaS v2 API](http://developer.openstack.org/api-ref/networking/v2/" \l "lbaas-2-0-stable)。

**配置**

**将LBaaS v2配置为代理**

LBaaS v2服务插件添加到service\_plugins /etc/neutron/neutron.conf配置指令。插件列表是逗号分隔的:

|  |
| --- |
| **service\_plugins = [existing service plugins],neutron\_lbaas.services.loadbalancer.plugin.LoadBalancerPluginv2** |

将LBaaS v2服务提供者添加到[service\_provider]区域内的service\_provider配置指令(service\_provider)部分，在/etc/neutron/neutron-lbaas.conf中:

|  |
| --- |
| **service\_provider = LOADBALANCERV2:Haproxy:neutron\_lbaas.drivers.haproxy.plugin\_driver.HaproxyOnHostPluginDriver:default** |

如果您为其他网络服务插件(如VPNaaS或FWaaS)提供了现有的服务提供者，那么在[service\_provider]部分中添加了service\_provider线作为单独的行。这些配置指示是可重复的，并且不是逗号分隔的。

在/etc/neutron/lbaas\_agent.ini中选择消息接口的驱动

|  |
| --- |
| **[DEFAULT]**  **interface\_driver = INTERFACE\_DRIVER** |

将INTERFACE\_DRIVER替换为在您的环境中使用的layer-2代理的接口驱动程序。例如，openvswitch或linuxbridge。

运行neutron-lbaas数据库迁移:

|  |
| --- |
| **neutron-db-manage --subproject neutron-lbaas upgrade head** |

如果您已经部署了LBaaS v1，那么现在停止LBaaS v1代理。v1和v2代理不能同时运行。

启动LBaaS v2代理:

|  |
| --- |
| **neutron-lbaasv2-agent \**  **--config-file /etc/neutron/neutron.conf \**  **--config-file /etc/neutron/lbaas\_agent.ini** |

重新启动网络服务以激活新的配置。现在可以使用LBaaS v2代理创建负载均衡器。

**将LBaaS v2配置为Octavia**

Octavia为负载均衡器提供了额外的功能，包括使用计算驱动程序来构建作为负载均衡器的实例。在东京的OpenStack峰会上，实验室安装和配置OpenStack Octavia会话提供了Octavia的概述。

DevStack文档提供了一个简单的方法来部署Octavia，并使用冗余负载均衡器实例对服务进行测试。如果您已经在您的环境中安装和配置了Octavia，您可以配置网络服务来使用Octavia:

LBaaS v2服务插件添加到service\_plugins /etc/neutron/neutron.conf配置指令。插件列表是逗号分隔的:

|  |
| --- |
| **service\_plugins = [existing service plugins],neutron\_lbaas.services.loadbalancer.plugin.LoadBalancerPluginv2** |

将Octavia服务提供者添加到service\_provider配置指令/etc/neutron/neutron.conf[service\_providers]部分中:  
确保LBaaS v1和v2服务提供者从[service\_provider]部分中删除。它们不与Octavia同时使用。验证所有LBaaS代理都停止了。

|  |
| --- |
| **service\_provider = LOADBALANCERV2:Octavia:neutron\_lbaas.drivers.octavia.driver.OctaviaDriver:default** |

重新启动网络服务以激活新的配置。现在，您可以创建和管理使用Octavia的负载均衡器。

**LBaaS v2操作**

样的neutron命令用于LBaaS v2代理或Octavia。

**构建LBaaS v2负载均衡器**

首先在网络上创建一个负载均衡器。在这个示例中，私有网络是一个带有两个web服务器实例的独立网络:

|  |
| --- |
| **$ neutron lbaas-loadbalancer-create --name test-lb private-subnet** |

您可以使用neutron lbaas-loadbalancer show命令查看负载平衡状态和IP地址:

|  |
| --- |
| **$ neutron lbaas-loadbalancer-show test-lb**  **+---------------------+------------------------------------------------+**  **| Field               | Value                                          |**  **+---------------------+------------------------------------------------+**  **| admin\_state\_up      | True                                           |**  **| description         |                                                |**  **| id                  | 7780f9dd-e5dd-43a9-af81-0d2d1bd9c386           |**  **| listeners           | {"id": "23442d6a-4d82-40ee-8d08-243750dbc191"} |**  **|                     | {"id": "7e0d084d-6d67-47e6-9f77-0115e6cf9ba8"} |**  **| name                | test-lb                                        |**  **| operating\_status    | ONLINE                                         |**  **| provider            | haproxy                                        |**  **| provisioning\_status | ACTIVE                                         |**  **| tenant\_id           | fbfce4cb346c4f9097a977c54904cafd               |**  **| vip\_address         | 192.168.1.22                                   |**  **| vip\_port\_id         | 9f8f8a75-a731-4a34-b622-864907e1d556           |**  **| vip\_subnet\_id       | f1e7827d-1bfe-40b6-b8f0-2d9fd946f59b           |**  **+---------------------+------------------------------------------------+** |

更新安全组以允许流量到达新的负载均衡器。创建一个新的安全组和ingress规则，允许流量进入新的负载均衡器。负载均衡器的neutron端口被显示为上面的vip\_port\_id。创建一个安全组和规则，允许TCP端口80、TCP端口443和所有ICMP流量:

|  |
| --- |
| **$ neutron security-group-create lbaas**  **$ neutron security-group-rule-create \**  **--direction ingress \**  **--protocol tcp \**  **--port-range-min 80 \**  **--port-range-max 80 \**  **--remote-ip-prefix 0.0.0.0/0 \**  **lbaas**  **$ neutron security-group-rule-create \**  **--direction ingress \**  **--protocol tcp \**  **--port-range-min 443 \**  **--port-range-max 443 \**  **--remote-ip-prefix 0.0.0.0/0 \**  **lbaas**  **$ neutron security-group-rule-create \**  **--direction ingress \**  **--protocol icmp \**  **lbaas** |

使用neutron lbaas-loadbalancer show命令的vip\_port\_id参数将安全组应用到负载均衡器的网络端口:  
这个负载均衡器是活动的，并且已经准备在192.168.1.22提供服务。

|  |
| --- |
| **$ neutron port-update \**  **--security-group lbaas \**  **9f8f8a75-a731-4a34-b622-864907e1d556** |

在做进一步操作之前，验证负载均衡器能对ping信号作出响应:

|  |
| --- |
| **$ ping -c 4 192.168.1.22**  **PING 192.168.1.22 (192.168.1.22) 56(84) bytes of data.**  **64 bytes from 192.168.1.22: icmp\_seq=1 ttl=62 time=0.410 ms**  **64 bytes from 192.168.1.22: icmp\_seq=2 ttl=62 time=0.407 ms**  **64 bytes from 192.168.1.22: icmp\_seq=3 ttl=62 time=0.396 ms**  **64 bytes from 192.168.1.22: icmp\_seq=4 ttl=62 time=0.397 ms**  **--- 192.168.1.22 ping statistics ---**  **4 packets transmitted, 4 received, 0% packet loss, time 2997ms**  **rtt min/avg/max/mdev = 0.396/0.402/0.410/0.020 ms** |

**添加一个HTTP监听器**

通过在线负载均衡器，您可以为80端口上的HTTP通信添加一个侦听器:

|  |
| --- |
| **$ neutron lbaas-listener-create \**  **--name test-lb-http \**  **--loadbalancer test-lb \**  **--protocol HTTP \**  **--protocol-port 80** |

**您可以开始创建一个池，并向池中添加成员，以在端口80上提供HTTP内容。对于这个示例，web服务器是192.168.1.16和192.168.1.17:**

|  |
| --- |
| **$ neutron lbaas-pool-create \**  **--name test-lb-pool-http \**  **--lb-algorithm ROUND\_ROBIN \**  **--listener test-lb-http \**  **--protocol HTTP**  **$ neutron lbaas-member-create \**  **--subnet private-subnet \**  **--address 192.168.1.16 \**  **--protocol-port 80 \**  **test-lb-pool-http**  **$ neutron lbaas-member-create \**  **--subnet private-subnet \**  **--address 192.168.1.17 \**  **--protocol-port 80 \**  **test-lb-pool-http** |

您可以使用curl来通过负载均衡器来验证连接到您的web服务器:

|  |
| --- |
| **$ curl 192.168.1.22**  **web2**  **$ curl 192.168.1.22**  **web1**  **$ curl 192.168.1.22**  **web2**  **$ curl 192.168.1.22**  **web1** |

在本例中，负载均衡器使用轮询算法，后端web服务器之间的流量交替。

您可以添加一个健康监视器，这样就可以从池中删除未响应的服务器:

|  |
| --- |
| **$ neutron lbaas-healthmonitor-create \**  **--delay 5 \**  **--max-retries 2 \**  **--timeout 10 \**  **--type HTTP \**  **--pool test-lb-pool-http** |

在这个示例中，如果在两个5秒的间隔的健康检查失败，health monitor会从池中删除服务器。当服务器恢复并开始响应健康检查时，它再次被添加到池中。

**添加一个HTTPS监听器**

您可以在端口443上为HTTPS传输添加另一个侦听器。LBaaS v2在负载均衡器上提供了SSL/TLS终止，但这个示例采用了一种更简单的方法，并允许在每个成员服务器上终止加密连接。

首先创建一个监听器，附加一个池，然后添加成员:

|  |
| --- |
| **$ neutron lbaas-listener-create \**  **--name test-lb-https \**  **--loadbalancer test-lb \**  **--protocol HTTPS \**  **--protocol-port 443**  **$ neutron lbaas-pool-create \**  **--name test-lb-pool-https \**  **--lb-algorithm LEAST\_CONNECTIONS \**  **--listener test-lb-https \**  **--protocol HTTPS**  **$ neutron lbaas-member-create \**  **--subnet private-subnet \**  **--address 192.168.1.16 \**  **--protocol-port 443 \**  **test-lb-pool-https**  **$ neutron lbaas-member-create \**  **--subnet private-subnet \**  **--address 192.168.1.17 \**  **--protocol-port 443 \**  **test-lb-pool-https** |

您还可以为HTTPS池添加一个健康监视器:

|  |
| --- |
| **$ neutron lbaas-healthmonitor-create \**  **--delay 5 \**  **--max-retries 2 \**  **--timeout 10 \**  **--type HTTPS \**  **--pool test-lb-pool-https** |

负载均衡器现在在端口80和443上处理流量。

**关联一个浮动的IP地址**

在公共或提供者网络上部署的负载均衡器不需要指定的浮动IP地址。外部客户端可以直接访问这些负载均衡器的虚拟IP地址(VIP)。

然而，部署在私有或隔离网络上的负载均衡器需要一个浮动的IP地址，如果它们必须可以访问外部客户端。要完成此步骤，您必须在私有和公共网络和可用的浮动IP地址之间有一个路由器。

您可以从本节开头使用neutronlbaas-loadbalancershow命令来定位vip\_port\_id。vip\_port\_id是分配给负载均衡器的网络端口的ID。您可以将一个自由浮动的IP地址与负载均衡器联系起来，使用neutron floatingip-associate：

|  |
| --- |
| **$ neutron floatingip-associate FLOATINGIP\_ID LOAD\_BALANCER\_PORT\_ID** |

**设置LBaaS v2的配额**

配额限制了负载均衡器和负载平衡池的数量。默认情况下，两个配额都设置为10。

你可以使用neutron quota-update命令来调整配额:

|  |
| --- |
| **$ neutron quota-update --tenant-id TENANT\_UUID --loadbalancer 25**  **$ neutron quota-update --tenant-id TENANT\_UUID --pool 50** |

设置为- 1，取消租户的配额。

**负载均衡器检索数据**

LBaaS v2代理每隔6秒为每个负载均衡器收集四类统计数据。用户可以使用neutron lbaas-loadbalancer-stats命令查询这些统计信息:

|  |
| --- |
| **$ neutron lbaas-loadbalancer-stats test-lb**  **+--------------------+----------+**  **| Field              | Value    |**  **+--------------------+----------+**  **| active\_connections | 0        |**  **| bytes\_in           | 40264557 |**  **| bytes\_out          | 71701666 |**  **| total\_connections  | 384601   |**  **+--------------------+----------+** |

active\_connections计数是在代理轮询负载均衡器时激活的连接总数。自负载均衡器上次启动以来，其他三个统计数据是累计的。例如，如果负载均衡器由于系统错误或配置更改而重新启动，这些统计数据将被重置。