

# Linux网桥：自助服务网络

« (deploy-lb-provider.html) » (deploy-lb-ha-vrrp.html) ✖ (https://bugs.launchpad.net/neutron/+filebug?field.title=Linux%20bridge%3A%20Self-service%20networks%20in%20Neutron&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: \_\_\_\_%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A-----%0ARelease:%2012.0.1.dev11%20on%202018-03-07%2021:05%0ASHA:%2043df2709acbdce86686a40b75fd34e96880427d0%0ASource:%20https://git.openstack.org/cgiit/openstack/neutron/tree/doc/source/admin/deploy-lb-selfservice.rst%0AURL: https://docs.openstack.org/neutron/queens/admin/deploy-lb-selfservice.html&field.tags=doc)

更新日期：2018-03-07 21:05

这个架构示例扩展了Linux网桥：提供商网络 (deploy-lb-provider.html#deploy-lb-provider)以支持几乎无限量的完全虚拟网络。虽然网络服务支持VLAN自助服务网络，但本示例重点介绍VXLAN自助服务网络。有关自助服务网络的更多信息，请参阅自助服务网络 (intro-os-networking.html#intro-os-networking-selfservice)。

✔ 注意

Linux网桥代理不支持其他覆盖协议，如GRE和Geneve。

## 先决条件

使用以下组件添加一个网络节点：

- 三个网络接口：管理，提供商和覆盖。
- OpenStack网络Linux桥接器第2层代理，第3层代理和任何依赖。

使用以下组件修改计算节点：

- 添加一个网络接口：叠加。

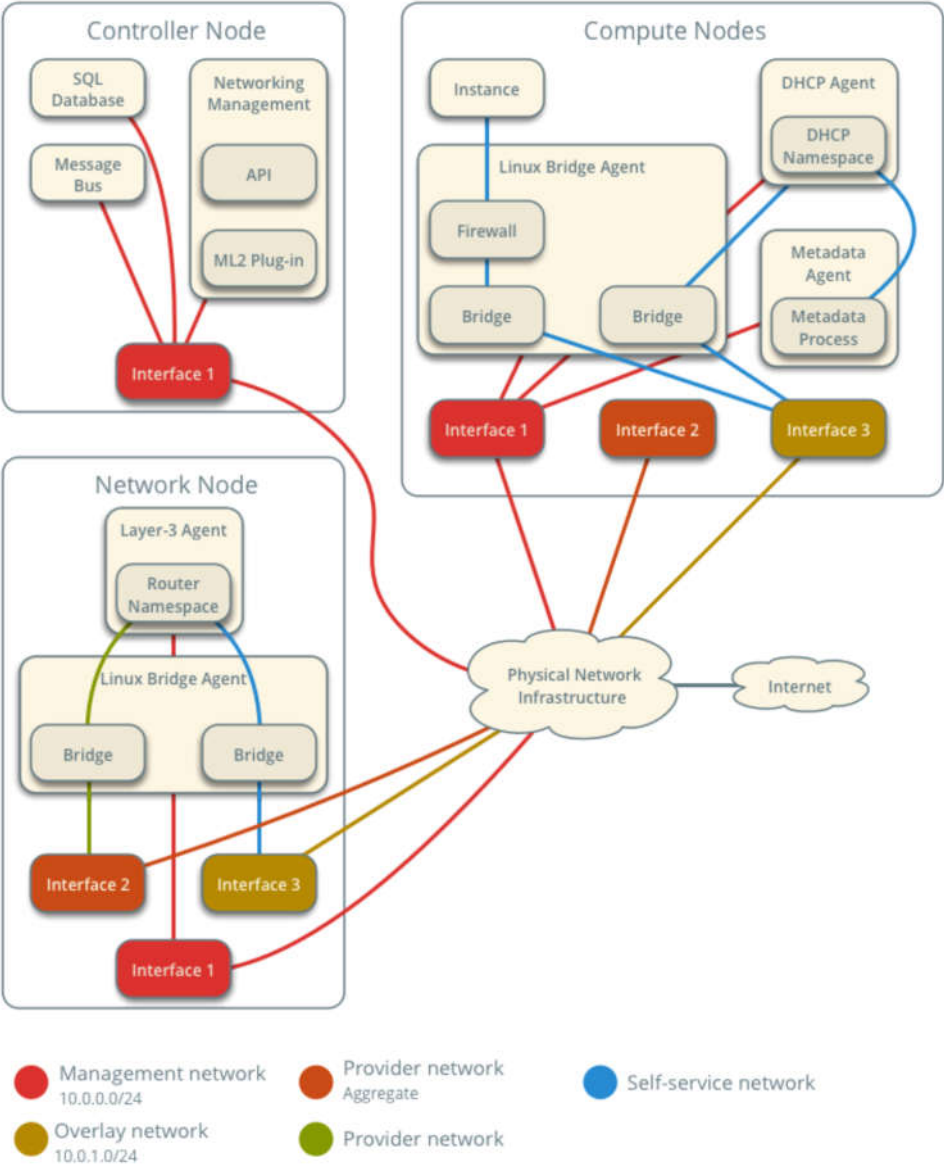
✔ 注意

您可以将DHCP和元数据代理保留在每个计算节点上或将它们移动到网络节点。

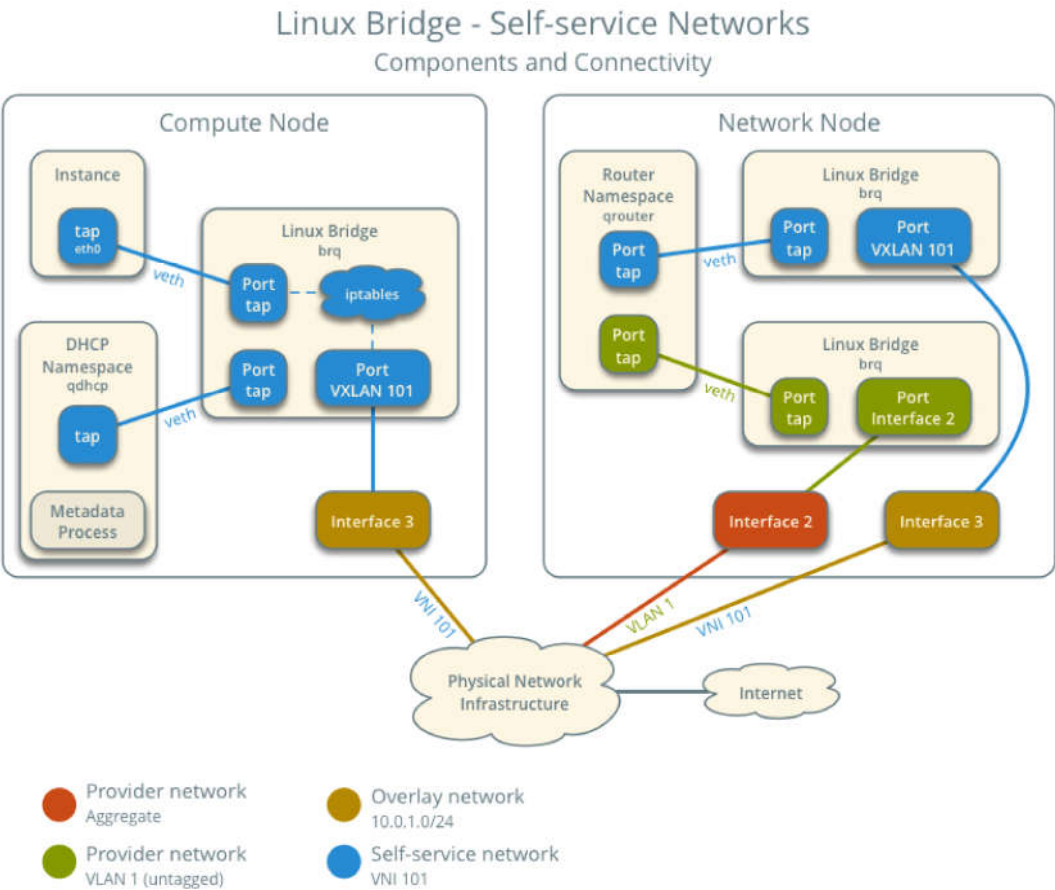
## 建筑

# Linux Bridge - Self-service Networks

## Overview



下图显示了一个自助服务网络和一个未标记（平面）提供商网络的组件和连接。在这种情况下，实例驻留在与网络的DHCP代理程序相同的计算节点上。如果DHCP代理驻留在另一个计算节点上，则后者只包含一个DHCP名称空间和一个在覆盖物理网络接口上具有端口的Linux桥。



## 配置示例¶

使用以下示例配置作为模板，将对自助服务网络的支持添加到支持提供商网络的现有操作环境中。

### 控制器节点¶

- 在 `neutron.conf` 文件中：
  - 启用路由并允许重叠的IP地址范围。

```
[DEFAULT]
service_plugins = router
allow_overlapping_ips = True
```

- 在 `m12_conf.ini` 文件中：
  - 添加 `vxlan` 键入驱动程序和项目网络类型。

```
[m12]
type_drivers = flat, vlan, vxlan
tenant_network_types = vxlan
```

- 启用第2层人口机制驱动程序。

```
[m12]
mechanism_drivers = linuxbridge, l2population
```

- 配置VXLAN网络号 ( VNI ) 范围。

```
[m12_type_vxlan]
vni_ranges = VNI_START: VNI_END
```

更换 `VNI_START` 和 `VNI_END` 适当的数值。

- 重新启动以下服务：
  - 服务器

### 网络节点¶

1. 安装网络服务第3层代理。
2. 在该`neutron.conf`文件中，配置常用选项：

```
[DEFAULT]
core_plugin = ml2
auth_strategy = keystone

[数据库]
#...

[keystone_authtoken]
#...

[新星]
#...

[代理商]
#...
```

查看安装教程和指南 (<https://docs.openstack.org>)，并 配置参考 (<https://docs.openstack.org>)您的OpenStack版本，以获得适当的额外的配置为 `[DEFAULT]`，`[database]`，`[keystone_authtoken]`，`[nova]`，和 `[agent]`部分。

3. 在该`linuxbridge_agent.ini`文件中，配置第2层代理。

```
[linux_bridge]
physical_interface_mappings = 提供者: PROVIDER_INTERFACE

[vxlan]
enable_vxlan = True
l2_population = True
local_ip = OVERLAY_INTERFACE_IP_ADDRESS

[securitygroup]
firewall_driver = iptables
```

替换`PROVIDER_INTERFACE`为处理提供商网络的底层接口的名称。例如，`eth1`。

替换`OVERLAY_INTERFACE_IP_ADDRESS`为处理自助服务网络的VXLAN叠加层的接口的IP地址。

4. 在该`l3_agent.ini`文件中，配置三层代理。

```
[DEFAULT]
interface_driver = linuxbridge
external_network_bridge =
```

#### ● 注意

该`external_network_bridge`选项有意不包含任何值。

5. 开始以下服务：
  - Linux桥代理
  - 第3层代理

## 计算节点¶

1. 在该`linuxbridge_agent.ini`文件中，启用VXLAN支持，包括第2层人口。

```
[vxlan]
enable_vxlan = True
l2_population = True
local_ip = OVERLAY_INTERFACE_IP_ADDRESS
```

替换`OVERLAY_INTERFACE_IP_ADDRESS`为处理自助服务网络的VXLAN叠加层的接口的IP地址。

2. 重新启动以下服务：
  - Linux桥代理

## 验证服务操作¶

1. 获取管理项目凭证。
2. 验证代理的存在和操作。

```
$ openstack网络代理列表
```

ID	代理类型	主机	可用区域	活着	状态	二进制
09de6af6-c5f1-4548-8b09-18801f068c57	Linux桥代理	compute2		True	UP	neutron-linuxbridge-agent
188945d1-9e70-4803-a276-df924e0788a4	Linux桥代理	compute1		True	UP	neutron-linuxbridge-agent
e76c440d-d5f6-4316-a674-d689630b629e	DHCP代理	compute1	新星	True	UP	neutron-dhcp-agent
e67367de-6657-11e6-86a4-931cd04404bb	DHCP代理	compute2	新星	True	UP	neutron-dhcp-agent
e8174cae-6657-11e6-89f0-534ac6d0cb5c	元数据代理	compute1		True	UP	neutron-metadata-agent
ece49ec6-6657-11e6-bafb-c7560f19197d	元数据代理	compute2		True	UP	neutron-metadata-agent
598f6357-4331-4da5-a420-0f5be000bec9	L3代理	network1	新星	True	UP	中子-13-代理
f4734e0f-bcd5-4922-a19d-e31d56b0a7ae	Linux桥代理	network1		True	UP	neutron-linuxbridge-agent

## 创建初始网络<sup>1</sup>

该配置支持多个VXLAN自助服务网络。为简单起见，以下过程在平面提供商网络上创建一个自助服务网络和一个带网关的路由器。路由器对IPv4网络流量使用NAT，并直接路由IPv6网络流量。

### 注意

与自助服务网络的IPv6连接通常需要向节点和物理网络基础设施添加静态路由。

1. 获取管理项目凭证。
2. 更新提供商网络以支持自助服务网络的外部连接。

```
$ openstack network set --external provider1
```

### 注意

该命令不提供输出。

3. 定期（非行政）项目凭证。
4. 创建一个自助服务网络。

```
$ openstack network create selfservice1
+-----+-----+
| 字段 | 值 |
+-----+-----+
| admin_state_up | UP |
| mtu | 1450 |
| 名称 | selfservice1 |
| port_security_enabled | True |
| 路由器: 外部 | 内部 |
| 共享 | False |
| 状态 | ACTIVE |
+-----+-----+
```

5. 在自助服务网络上创建一个IPv4子网。

```
$ openstack subnet create --subnet-range 192.0.2.0 / 24 \
--network selfservice1 --dns-nameserver 8.8.4.4 selfservice1-v4
+-----+-----+
| 字段 | 值 |
+-----+-----+
| allocation_pools | 192.0.2.2-192.0.2.254 |
| cidr | 192.0.2.0/24 |
| dns_nameservers | 8.8.4.4 |
| enable_dhcp | True |
| gateway_ip | 192.0.2.1 |
| ip_version | 4 |
| 名称 | selfservice1-v4 |
+-----+-----+
```

6. 在自助服务网络上创建一个IPv6子网。

```
$ openstack subnet create --subnet-range fd00: 192: 0: 2 :: / 64
--ip- version 6 \ --ipv6-ra-mode slaac --ipv6-address-mode slaac --network selfservice1 \
- dns-nameserver 2001: 4860: 4860 :: 8844 selfservice1-v6
+ ----- + ----- +
| 字段 | 值 |
+ ----- + ----- +
| allocation_pools | fd00: 192: 0: 2 :: 2-fd00: 192: 0: 2: ffff: ffff: ffff: ffff |
| cidr | fd00: 192: 0: 2 :: / 64 |
| dns_nameservers | 2001: 4860: 4860 :: 8844 |
| enable_dhcp | True |
| gateway_ip | fd00: 192: 0: 2 :: 1 |
| ip_version | 6 |
| ipv6_address_mode | slaac |
| ipv6_ra_mode | slaac |
| 名称 | selfservice1-v6 |
+ ----- + ----- +
```

#### 7. 创建一个路由器。

```
$ openstack router create router1
+ ----- + ----- +
| 字段 | 值 |
+ ----- + ----- +
| admin_state_up | UP |
| 名称 | router1 |
| 状态 | ACTIVE |
+ ----- + ----- +
```

#### 8. 将IPv4和IPv6子网添加为路由器上的接口。

```
$ openstack 路由器添加子网router1 selfservice1-v4
$ openstack 路由器添加子网router1 selfservice1-v6
```

#### 注意

这些命令不提供输出。

#### 9. 将提供商网络添加为路由器上的网关。

```
$ neutron router-gateway-set router1 provider1
设置路由器router1的网关
```

## 验证网络操作1

#### 1. 在每个计算节点上，验证是否创建第二个qdhcp名称空间。

```
# IP netns
qdhcp-8b868082-e312-4110-8627-298109d4401c
qdhcp-8fbc13ca-cfe0-4b8a-993B-e33f37ba66d1
```

#### 2. 在网络节点上，验证qrouter命名空间的创建。

```
# IP netns
qrouter-17db2a15-e024-46d0-9250-4cd4d336a2cc
```

#### 3. 定期（非行政）项目凭证。

#### 4. 创建适当的安全组规则以允许ping和使用网络SSH访问实例。

```
$ openstack安全组规则create --proto icmp default
+ ----- + ----- +
| 字段| 值|
+ ----- + ----- +
| 方向| 入口|
| ethertype | IPv4 |
| 协议| icmp |
| remote_ip_prefix | 0.0.0.0/0 |
+ ----- + ----- +

$ openstack安全组规则create --ethertype IPv6 --proto ipv6-icmp default
+ ----- + ----- +
| 字段| 值|
+ ----- + ----- +
| 方向| 入口|
| ethertype | IPv6 |
| 协议| ipv6-icmp |
+ ----- + ----- +

$ openstack安全组规则create --proto tcp --dst-port 22默认值
+ ----- + ----- +
| 字段| 值|
+ ----- + ----- +
| 方向| 入口|
| ethertype | IPv4 |
| port_range_max | 22 |
| port_range_min | 22 |
| 协议| tcp |
| remote_ip_prefix | 0.0.0.0/0 |
+ ----- + ----- +

$ openstack安全组规则create --ethertype IPv6 --proto tcp --dst-port 22默认值
+ ----- + ----- +
| 字段| 值|
+ ----- + ----- +
| 方向| 入口|
| ethertype | IPv6 |
| port_range_max | 22 |
| port_range_min | 22 |
| 协议| tcp |
+ ----- + ----- +
```

5. 使用自助服务网络上的接口启动实例。例如，使用风味ID 1的Cirros图像。

```
$ openstack server create --flavor 1 --image cirros --nic net-id = NETWORK_ID selfservice-instance1
```

替换**NETWORK\_ID**为自助服务网络的ID。

6. 确定实例的IPv4和IPv6地址。

```
$ openstack服务器列表
+ ----- + ----- + ----- + ----- +
| ID | 名称 | 状态 | 网络 |
+ ----- + ----- + ----- + ----- +
| c055cdb0-ebb4-4d65-957c-35cbdbd59306 | selfservice-instance1 | ACTIVE | selfservice1 = 192.0.2.4, fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0 |
+ ----- + ----- + ----- + ----- +
```

### 警告

IPv4地址位于私有IP地址范围（RFC1918）。因此，网络服务执行源网络地址转换（SNAT）以便实例访问诸如因特网的外部网络。从外部网络（如Internet）访问实例需要浮动IPv4地址。网络服务执行从自助服务网络上的浮动IPv4地址到实例IPv4地址的目标网络地址转换（DNAT）。另一方面，由于地址空间和NAT的复杂性大得多，用于IPv6的网络服务体系结构不支持NAT。因此，IPv6的浮动IP地址不存在，并且网络服务仅在自助服务网络上执行IPv6子网的路由。换一种说法，

7. 在控制器节点或任何可以访问提供者网络的主机 **ping**上，实例的IPv6地址。

```
$ ping6 -c 4 fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0
PING fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0 (fd00: 192: 0: 2: f816: 3eff: fe30 : 9cb0) 56个数据字节
fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0: icmp_seq = 1 ttl = 63时间= 2.08 ms
64字节fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0: icmp_seq = 2 ttl = 63时间= 1.88 ms
来自fd00的192字节: 192: 0: 2: f816: 3eff: fe30: 9cb0: icmp_seq = 3 ttl = 63时间= 1.55 ms
来自fd00的64字节: 192: 0: 2: f816: 3eff: fe30: 9cb0: icmp_seq = 4 ttl = 63时间= 1.62 ms

--- fd00: 192: 0: 2: f816: 3eff: fe30: 9cb0 ping statistics ---
发送4个数据包, 收到4个数据包, 丢包0%, 时间3004ms
rtt min / avg / max / mdev = 1.557 / 1.788 / 2.085 / 0.217毫秒
```

8. （可选）启用来自外部网络（如Internet）的IPv4访问实例。

a. 在提供商网络上创建浮动IPv4地址。

```
$ openstack floating ip create provider1
+ ----- + ----- +
|  字段 |  值 |
+ ----- + ----- +
| fixed_ip | 无 |
| id | 22a1b088-5c9b-43b4-97f3-970ce5df77f2 |
| instance_id | 无 |
| ip | 203.0.113.16 |
| 游泳池 | provider1 |
+ ----- + ----- +
```

b. 将浮动IPv4地址与实例相关联。

```
$ openstack server add floating ip selfservice-instance1 203 .0.113.16
```

注意

该命令不提供输出。

c. 在控制器节点或任何可以访问提供商网络的主机 ping上，实例的浮动IPv4地址。

```
$ ping -c 4 203 .0.113.16
PING 203.0.113.16 (203.0.113.16) 56 (84) 字节的数据:
来自203.0.113.16的64字节: icmp_seq = 1 ttl = 63时间=
3.41ms来自203.0.113.16的64字节: icmp_seq = 2ttl = 63时间=
1.67ms来自203.0.113.16的64字节: icmp_seq = 3ttl = 63时间= 1.47
来自203.0.113.16的ms 64个字节: icmp_seq = 4 ttl = 63时间= 1.59ms

--- 203.0.113.16 ping statistics ---
发送4个数据包，收到4个数据包，丢包0%，时间3005ms
rtt min / avg / max / mdev = 1.473 / 2.040 / 3.414 / 0.798 ms
```

9. 获取对实例的访问权限。

10. 测试互联网或其他外部网络的IPv4和IPv6连接。

# 网络流量

以下各节将介绍几种常见情况下的网络流量。*南北*网络流量在实例和Internet等外部网络之间传输。*东西向*网络流量在相同或不同网络上的实例之间传播。在所有情况下，物理网络基础架构都处理提供商网络和外部网络（如Internet）之间的交换和路由。每个案例引用一个或多个以下组件：

- 提供商网络（VLAN）
  - VLAN ID 101（标记）
- 自助服务网络1（VXLAN）
  - VXLAN ID（VNI）101
- 自助服务网络2（VXLAN）
  - VXLAN ID（VNI）102
- 自助服务路由器
  - 提供商网络上的网关
  - 自助服务网络接口1
  - 自助服务网络接口2
- 实例1
- 实例2

## 南北方案1：具有固定IP地址的实例

对于具有固定IPv4地址的实例，网络节点对从自助服务传递到外部网络（如Internet）的南北传输执行SNAT。对于具有固定IPv6地址的实例，网络节点在自助服务网络和外部网络之间执行传统的流量路由。

- 该实例驻留在计算节点1上并使用自助服务网络1。
- 该实例将数据包发送到Internet上的主机。

以下步骤涉及计算节点1：

1. 实例接口（1）通过veth配对将数据包转发到自助服务网桥实例端口（2）。
2. 自助服务网桥上的安全组规则（3）处理数据包的防火墙和连接跟踪。
3. 自助服务网桥将数据包转发到VXLAN接口（4），该接口使用VNI 101封装数据包。
4. VXLAN接口的底层物理接口（5）通过覆盖网络（6）将数据包转发给网络节点。

以下步骤涉及网络节点：

1. VXLAN接口的底层物理接口（7）将数据包转发到解包数据包的VXLAN接口（8）。
2. 自助网桥路由器端口（9）将数据包转发到路由命名空间中的自助服务网络接口（10）。
  - 对于IPv4，路由器对将源IP地址更改为提供商网络上的路由器IP地址的数据包执行SNAT，并通过提供商网络（11）上的网关接口将其发送到提供商网络上的网关IP地址。
  - 对于IPv6，路由器通过提供商网关接口（11）将数据包发送到下一跳IP地址，通常是提供商网络上的网关IP地址。



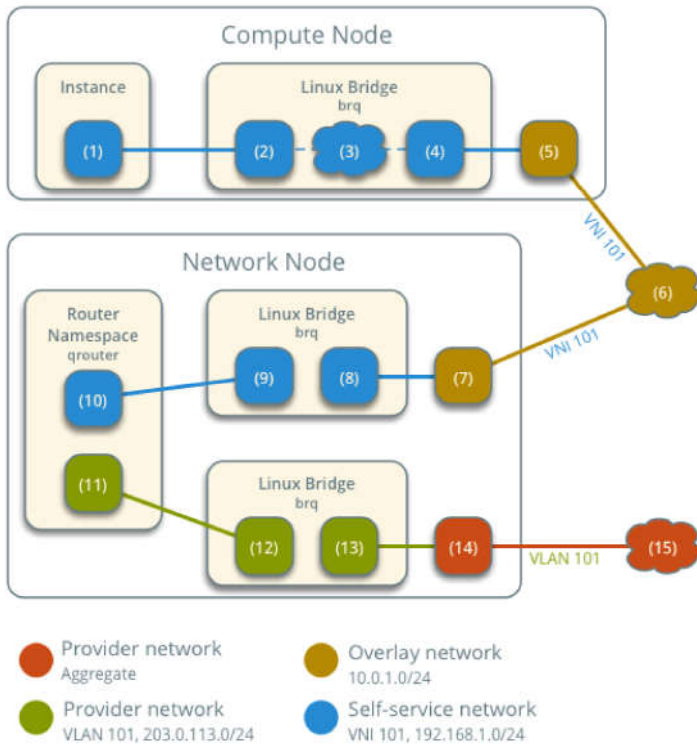
3. 路由器将数据包转发给提供商网桥路由端口（12）。
4. 提供商网桥上的VLAN子接口端口（13）将数据包转发给提供商物理网络接口（14）。
5. 提供商物理网络接口（14）将VLAN标签101添加到分组并将其经由物理网络基础设施（15）转发到因特网。

#### 注意

返回流量反过来也采用类似的步骤。但是，如果没有浮动IPv4地址，则提供商或外部网络上的主机无法发起与自助服务网络上实例的连接。

## Linux Bridge - Self-service Networks

### Network Traffic Flow - North/South Scenario 1



## 南北方案2：带有浮动IPv4地址的实例

对于具有浮动IPv4地址的实例，网络节点在从实例传递到外部网络（例如因特网和DNAT）的南北流量上执行SNAT，从而将南北流量从外部网络传送到实例。浮动IP地址和NAT不适用于IPv6。因此，网络节点在这种情况下路由IPv6流量。

- 该实例驻留在计算节点1上使用自助服务网络1。
- Internet上的主机将数据包发送到实例。

以下步骤涉及网络节点：

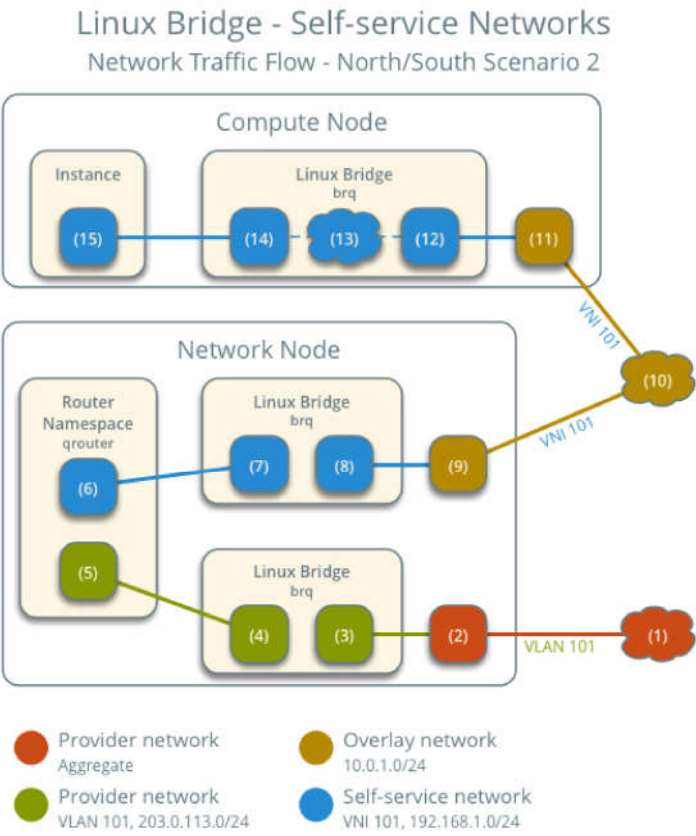
1. 物理网络基础设施（1）将数据包转发给提供商物理网络接口（2）。
2. 提供商物理网络接口删除VLAN标签101，并将数据包转发到提供商网桥上的VLAN子接口。
3. 提供商网桥将数据包转发给提供商网络（5）上的自助服务路由器网关端口。
  - 对于IPv4，路由器对自助服务网络上将目标IP地址更改为实例IP地址的数据包执行DNAT，并通过自助服务接口将其发送到自助服务网络上的网关IP地址（6）。
  - 对于IPv6，路由器通过自助服务接口（6）将数据包发送到下一跳IP地址，通常是自助服务网络上的网关IP地址。
4. 路由器将数据包转发到自助桥路由端口（7）。
5. 自助服务网桥将数据包转发到VXLAN接口（8），该接口使用VNI 101封装数据包。
6. VXLAN接口的底层物理接口（9）通过覆盖网络（10）将数据包转发给网络节点。

以下步骤涉及计算节点：

1. VXLAN接口的底层物理接口（11）将数据包转发到解包数据包的VXLAN接口（12）。
2. 自助服务网桥上的安全组规则（13）处理数据包的防火墙和连接跟踪。
3. 自助服务网桥实例端口（14）通过veth配对将数据包转发给实例接口（15）。

#### 注意

出口实例流量类似于南北方向1，但SNAT将数据包的源IP地址更改为浮动IPv4地址，而不是提供商网络上的路由器IP地址。



东西方案例1：同一网络上的实例1

在同一网络上具有固定IPv4 / IPv6或浮动IPv4地址的实例在包含这些实例的计算节点之间直接通信。

默认情况下，VXLAN协议缺乏目标位置的知识并使用多播来发现它。发现后，它将位置存储在本地转发数据库中。在大型部署中，发现过程可以生成所有节点必须处理的大量网络。为了消除后者并且通常提高效率，网络服务包括自动填充VXLAN接口的转发数据库的第二层人口机制驱动程序。示例配置启用此驱动程序。有关更多信息，请参阅ML2插件 (config-ml2.html#config-plugin-ml2)。

- 实例1驻留在计算节点1上并使用自助服务网络1。
- 实例2驻留在计算节点2上并使用自助服务网络1。
- 实例1向实例2发送一个数据包。

以下步骤涉及计算节点1：

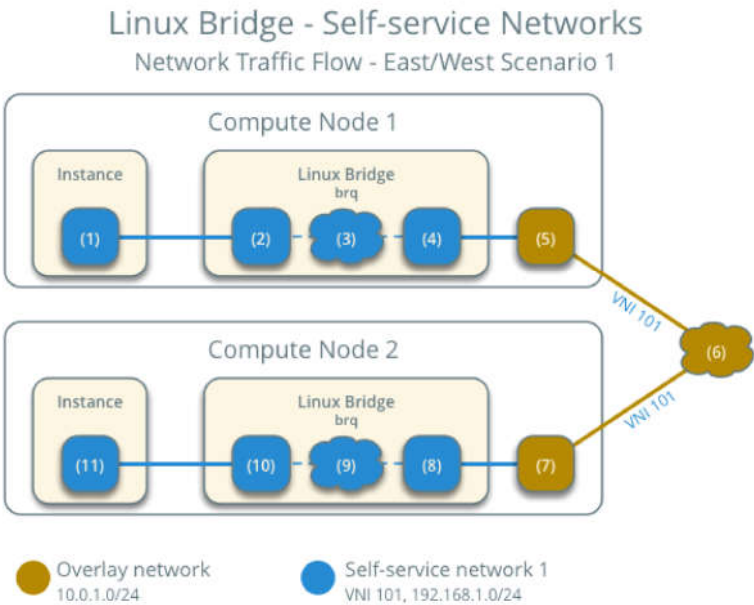
- 实例1接口（1）通过veth配对将数据包转发到自助服务网桥实例端口（2）。
- 自助服务网桥上的安全组规则（3）处理数据包的防火墙和连接跟踪。
- 自助服务网桥将数据包转发到VXLAN接口（4），该接口使用VNI 101封装数据包。
- VXLAN接口的底层物理接口（5）通过覆盖网络（6）将数据包转发给计算节点2。

以下步骤涉及计算节点2：

- VXLAN接口的底层物理接口（7）将数据包转发到解包数据包的VXLAN接口（8）。
- 自助服务网桥上的安全组规则（9）处理数据包的防火墙和连接跟踪。
- 自助网桥实例端口（10）通过veth配对将数据包转发给实例1接口（11）。

注意

返回流量反过来也采用类似的步骤。



## 东西方案2：不同网络上的实例II

使用固定IPv4 / IPv6地址或浮动IPv4地址的实例通过网络节点上的路由器进行通信。自助服务网络必须驻留在同一台路由器上。

- 实例1驻留在计算节点1上使用自助服务网络1。
- 实例2驻留在计算节点1上使用自助服务网络2。
- 实例1向实例2发送一个数据包。

✔ 注意

两个实例都位于同一个计算节点上，以说明VXLAN如何使多个叠加层使用同一个三层网络。

以下步骤涉及计算节点：

- 实例1接口（1）通过veth配对将数据包转发到自助服务网桥实例端口（2）。
- 自助服务网桥上的安全组规则（3）处理数据包的防火墙和连接跟踪。
- 自助服务网桥将数据包转发到VXLAN接口（4），该接口使用VNI 101封装数据包。
- VXLAN接口的底层物理接口（5）通过覆盖网络（6）将数据包转发给网络节点。

以下步骤涉及网络节点：

- VXLAN接口的底层物理接口（7）将数据包转发到解包数据包的VXLAN接口（8）。
- 自助网桥路由器端口（9）将分组转发到路由器命名空间中的自助服务网络1接口（10）。
- 路由器通过自助服务网络2接口（11）将分组发送到下一跳IP地址，通常是自助服务网络2上的网关IP地址。
- 路由器将数据包转发到自助服务网络2桥路由器端口（12）。
- 自助服务网络2网桥将分组转发到VXLAN接口（13），VXLAN接口使用VNI 102对分组进行封装。
- 用于VXLAN接口的物理网络接口（14）经由覆盖网络（15）将分组发送到计算节点。

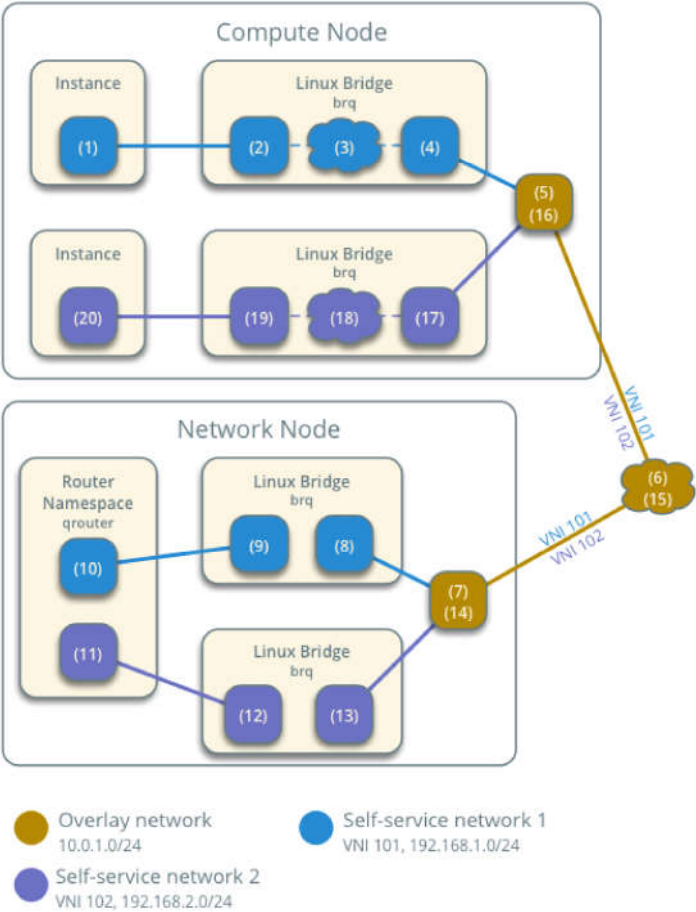
以下步骤涉及计算节点：

- VXLAN接口的底层物理接口（16）将数据包发送到展开数据包的VXLAN接口（17）。
- 自助服务网桥上的安全组规则（18）处理数据包的防火墙和连接跟踪。
- 自助网桥实例端口（19）通过veth配对将数据包转发给实例2接口（20）。

✔ 注意

返回流量反过来也采用类似的步骤。

Linux Bridge - Self-service Networks  
Network Traffic Flow - East/West Scenario 2



« (deploy-lb-provider.html) » (deploy-lb-ha-vrrp.html) ✖ (https://bugs.launchpad.net/neutron/+filebug?field.title=Linux%20bridge%3A%20Self-service%20networks%20in%20Neutron&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: \_\_\_\_%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A-----%0ARELEASE:%2012.0.1.dev11%20on%202018-03-07%2021:05%0ASHA:%2043df2709acbdce86686a40b75fd34e96880427d0%0ASOURCE:%20https://git.openstack.org/cgit/openstack/neutron/tree/doc/source/admin/deploy-lb-selfservice.rst%0AURL: https://docs.openstack.org/neutron/queens/admin/deploy-lb-selfservice.html&field.tags=doc)

更新日期：2018-03-07 21:05



(https://creativecommons.org/licenses/by/3.0/) 除另有说明外，本文档受 Creative Commons Attribution 3.0 许可的授权 (https://creativecommons.org/licenses/by/3.0/)。查看所有 OpenStack 法律文件 (http://www.openstack.org/legal)。

✖ 发现错误？报告错误 (HTTPS://BUGS.LAUNCHPAD.NET/NEUTRON/+FILEBUG?FIELD.TITLE=LINUX%20BRIDGE%3A%20SELF-SERVICE%20NETWORKS%20IN%20NEUTRON&FIELD.COMMENT=%0A%0A%0ATHIS BUG TRACKER IS FOR ERRORS WITH THE DOCUMENTATION, USE THE FOLLOWING AS A TEMPLATE AND REMOVE OR ADD FIELDS AS YOU SEE FIT. CONVERT [ ] INTO [X] TO CHECK BOXES:%0A%0A- [ ] THIS DOC IS INACCURATE IN THIS WAY: \_\_\_\_%0A- [ ] THIS IS A DOC ADDITION REQUEST.%0A- [ ] I HAVE A FIX TO THE DOCUMENT THAT I CAN PASTE BELOW INCLUDING EXAMPLE: INPUT AND OUTPUT. %0A%0AIF YOU HAVE A TROUBLESHOOTING OR SUPPORT ISSUE, USE THE FOLLOWING RESOURCES:%0A%0A - ASK OPENSTACK: HTTP://ASK.OPENSTACK.ORG%0A - THE MAILING LIST: HTTP://LISTS.OPENSTACK.ORG%0A - IRC: 'OPENSTACK' CHANNEL ON FREENODE%0A%0A-----%0ARELEASE:%2012.0.1.DEV11%20ON%202018-03-07%2021:05%0ASHA:%2043DF2709ACBDCE86686A40B75FD34E96880427D0%0ASOURCE:%20HTTPS://GIT.OPENSTACK.ORG/CGIT/OPENSTACK/NEUTRON/TREE/DOC/SOURCE/ADMIN/DEPLOY-LB-SELF-SERVICE.RST%0AURL: HTTPS://DOCS.OPENSTACK.ORG/NEUTRON/QUEENS/ADMIN/DEPLOY-LB-SELF-SERVICE.HTML&FIELD.TAGS=DOC)

❓ 问题吗？(HTTP://ASK.OPENSTACK.ORG)



OpenStack 文档 ▾

- 安装指南 (../install/index.html)
- OpenStack网络指南 (index.html)
  - 介绍 (intro.html)
  - 组态 (config.html)
  - 部署示例 (deploy.html)
  - 操作 (ops.html)
  - 移民 (migration.html)
  - 杂 (misc.html)
  - 存档的内容 (archives/index.html)
- 中子配置选项 (../configuration/index.html)
- 命令行界面参考 (../cli/index.html)
- 中子特征分类 (../feature\_classification/index.html)
- 贡献者指南 (../contributor/index.html)

页面内容

- 先决条件
- 建筑
- 示例配置
  - 控制器节点
  - 网络节点
  - 计算节点
  - 验证服务操作
  - 创建初始网络
  - 验证网络运行
- 网络流量
  - 南北方案1：具有固定IP地址的实例
  - 南北方案2：带有浮动IPv4地址的实例
  - 东西方案例1：在同一网络上的实例
  - 东西方案2：不同网络上的实例

OpenStack的

- 项目 (<http://openstack.org/projects/>)
- OpenStack安全 (<http://openstack.org/projects/openstack-security/>)
- 常见问题 (<http://openstack.org/projects/openstack-faq/>)
- 博客 (<http://openstack.org/blog/>)
- 新闻 (<http://openstack.org/news/>)

社区

- 用户组 (<http://openstack.org/community/>)
- 活动 (<http://openstack.org/community/events/>)
- 工作 (<http://openstack.org/community/jobs/>)
- 公司 (<http://openstack.org/foundation/companies/>)
- 有助于 (<http://docs.openstack.org/infra/manual/developers.html>)

文档

- OpenStack手册 (<http://docs.openstack.org>)
- 入门 (<http://openstack.org/software/start/>)
- API文档 (<http://developer.openstack.org>)
- 维基 (<https://wiki.openstack.org>)

品牌与法律

- 标志和指南 (<http://openstack.org/brand/>)
- 商标政策 (<http://openstack.org/brand/openstack-trademark-policy/>)
- 隐私政策 (<http://openstack.org/privacy/>)
- OpenStack CLA ([https://wiki.openstack.org/wiki/How\\_To\\_Contribute#Contributor\\_License\\_Agreement](https://wiki.openstack.org/wiki/How_To_Contribute#Contributor_License_Agreement))

保持联系

(<https://twitter.com/QueensCloud>) (<https://www.youtube.com/user/OpenStackFoundation>)

OpenStack项目是在Apache 2.0许可 (<http://www.apache.org/licenses/LICENSE-2.0>)下提供的。Openstack.org由 [Rackspace云计算提供支持 \(http://rackspace.com\)](http://rackspace.com)。