

# Octaviaの概要

Openstack Neutron ロードバランサーサービス

2017年5月

SCSK株式会社

R&Dセンター 技術戦略部

※本文中の会社名、商品名は、各社の商標及び登録商標です。

# ロードバランサーサービスとは

- Webシステムにかかる負荷を複数のサーバに分散させ、処理のバランスを調整
- パフォーマンスと可用性を向上
- 一般的には、ハードウェアで実現、負荷分散装置と呼ばれている
- ソフトウェアでも実現可能
  - HA Proxyなど

# Octaviaとは

- OpenstackのNeutron LBaaSプロジェクトから派生したクラウド向けに最適化されたプロジェクト
- 複数の仮想マシンを管理して柔軟な水平スケール可能なロードバランサーサービスを提供
- Neutron LBaaS v2 APIやCLIを使って管理
- Neutron LBaaSの置き換えではなく、Neutron LBaaS v2のプラグインである
- OctaviaのAPIもある

# Octaviaの構成要素

- HA Proxy

- フリーのソフトウェアロードバランサ
- C言語で書かれていて、高速かつ信頼性が高い
- TCP、HTTPベースアプリの要求を代理
- Linux Kernel 2.4以上で利用可能

- アンフォラ:

- ロードバランサーサービスを提供する仮想マシン
- HA Proxyが中で動作
- OSはUbuntu
- バックエンドのネットワークとLBネットワーク上に配置
- フロントエンドからのリクエストをバックエンドのサーバに分配
- コントローラとLBネットワーク上で通信

# Octaviaの構成要素

- LB(ロードバランサー) ネットワーク
  - コントローラとアンフォラが通信するネットワーク
- コントローラ
  - LBネットワークとOpenstackのコンポーネントにアクセスできるデーモン
  - Octaviaの全体を管理する4つのプロセスで構成されている
- Octaviaドライバー
  - Neutron Lbaas APIをOctavia apiプロセスに渡すコンポーネント
- VIP(Virtual IP Address):
  - ロードバランサーに紐付けている単体のIPアドレス
  - フロントエンドからこのIPアドレスにアクセス
  - HA構成では、VRRPを使って2台のアンフォラに紐付け

# Octaviaのプロセス

- APIコントローラ:

- APIリクエストを処理するプロセス
- Osloメッセージを使ってコントローラワーカーに実行を依頼する

- コントローラワーカー:

- APIコントローラからの依頼を処理するプロセス
- LBネットワーク経由でアンフォラを操作する

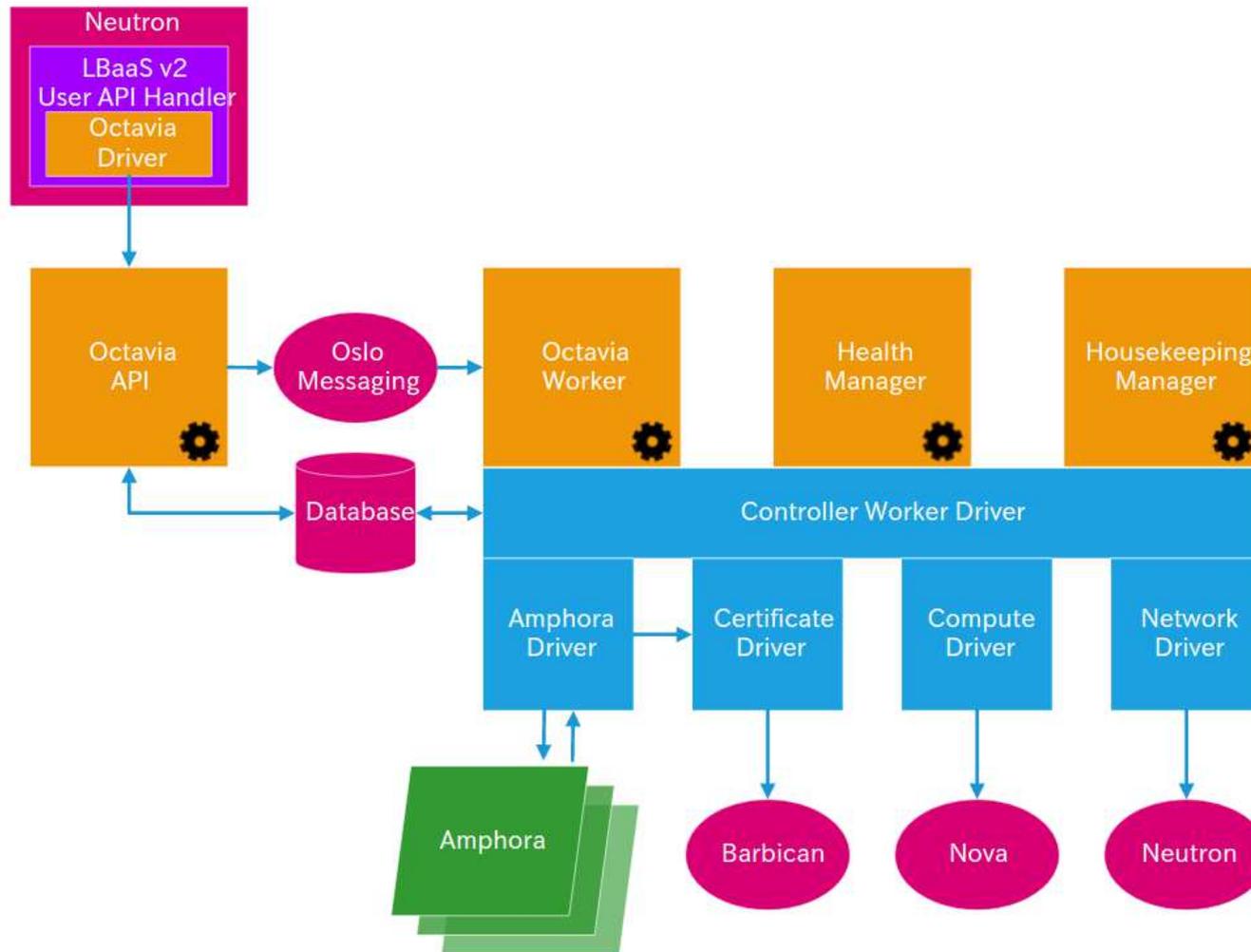
- ヘルスマネジャー:

- アンフォラを監視するプロセス
- アンフォラに異常がある場合、フェイルオーバー処理する

- ハウスキーピングマネジャー:

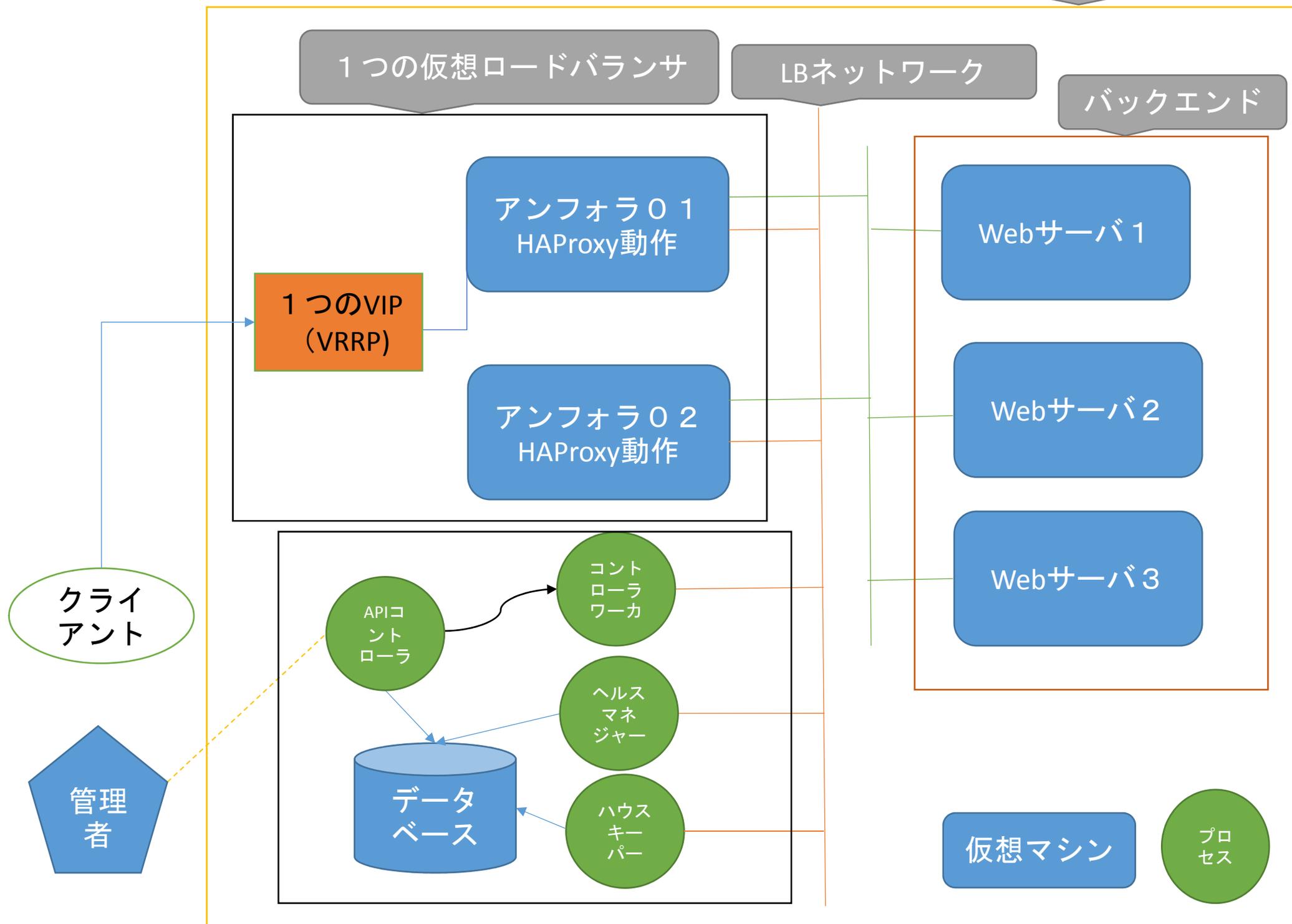
- 古いDBレコードを削除するプロセス
- スペアのプールを管理

# プロセス構成図



# Octaviaのロードバランサ構成図

Openstackの中



# OctaviaのDB構成の概念

DBのオブジェクト

プール：ロードバランサー経由のトラフィックを処理メンバーのグループ

メンバー：バックエンドサーバである。  
1つのメンバーは1つのプールにのみ紐付けされる

Webサーバ1

Webサーバ2

Webサーバ3

Webサーバ4

Webサーバ5

リスナー：あるロードバランサーでトラフィックをリッスンするポート番号を定義したもの

リスナー80

リスナー8080

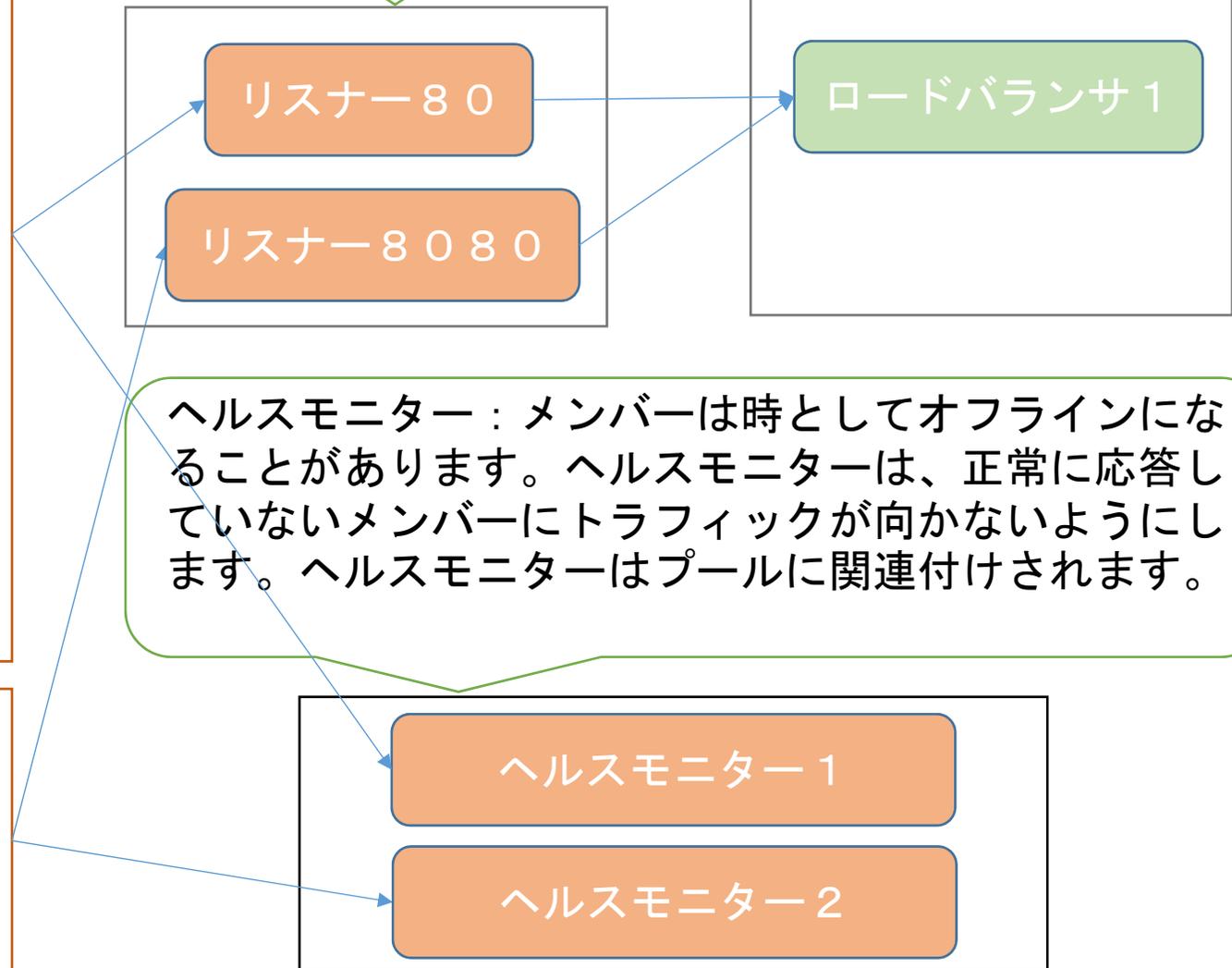
ロードバランサ：同じロードバランサーに複数のリスナーを関連付けできます

ロードバランサ1

ヘルスマニター：メンバーは時としてオフラインになることがあります。ヘルスマニターは、正常に応答していないメンバーにトラフィックが向かないようにします。ヘルスマニターはプールに関連付けされます。

ヘルスマニター1

ヘルスマニター2



# HA Proxyができること

- 負荷分散機能を提供
- TCP接続のプロキシ、アクセス経路を設定
- HTTPリバースプロキシ機能、リクエストを別のサーバに渡す
- URLやヘッダの書き換え
- SSLやHTTP圧縮を肩代わり
- ロギング機能、ネットワーク上の観測点
- トラフィックのコントロール

# HA Proxyができないこと

- キャッシュもしない
- bodyの書き換えもしない
- Webサーバではない
- IPやUDPといったパケットレベルのレイヤは見ない

# Octaviaインストール

- devstackでオールインワン環境で構築
- OS:Ubuntu16.04
- local.confに以下を追加し、stack.shを実行する
  - enable\_service q-lbaasv2
  - enable\_service octavia
  - enable\_service o-cw
  - enable\_service o-hk
  - enable\_service o-hm
  - enable\_service o-api
  - enable\_plugin neutron-lbaas <https://github.com/openstack/neutron-lbaas.git>  
stable/ocata
  - enable\_plugin octavia <https://github.com/openstack/octavia.git>  
stable/ocata

# ロードバランサーの作成

## シナリオ

- パブリックネットワーク、ルータを作成
- プライベートネットワーク上にバックエンドのサーバ2つ作成 (HTTPサービスを提供)
- Neutron LBaaS V2のCLIを使用してOctaviaのロードバランシングサービスを設定
- プライベートネットワーク上にロードバランサーを作成し、floating IPを使用してロードバランサーのVIPと紐付け、パブリック側からバックエンドのHTTPサービスにアクセスできるようにする

## Network Topology

Networks

Routers

Security Groups

Floating IPs



Launch Instance

+ Create Network

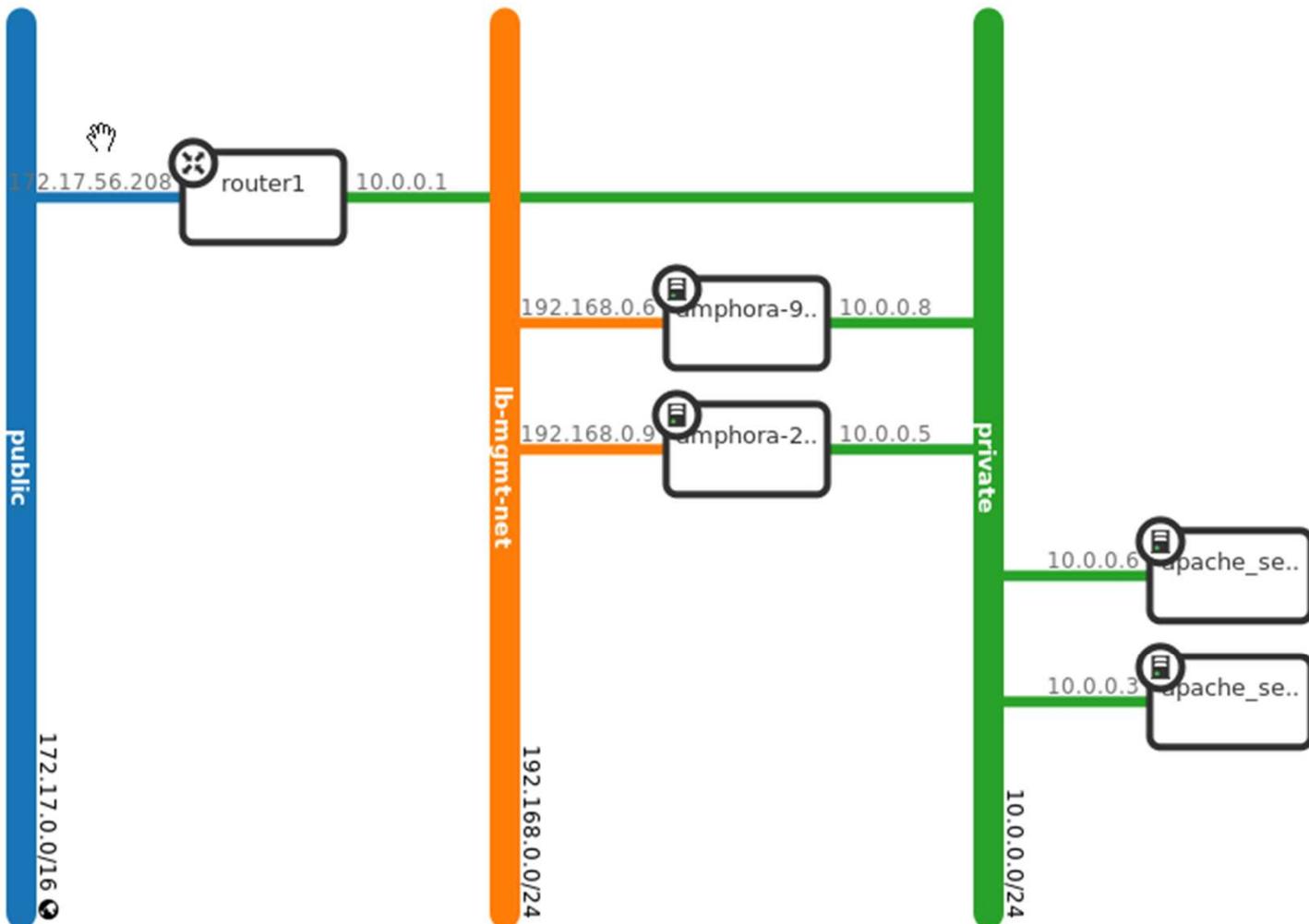
+ Create Router

Topology

Graph

Small

Normal



# ネットワークとルータを作成

- ルータを作成

- neutron router-create router1
- neutron router-gateway-set router1 public
- neutron router-interface-add router1  
subnet=private-subnet

# ネットワークとルータを作成

- パブリックネットワークを作成

- neutron net-create public --router:external -- provider:physical\_network public -- provider:network\_type flat

- neutron subnet-create --name public-subnet -- gateway 172.17.0.254 --allocation-pool start=172.17.56.200,end=172.17.56.254 --dns-nameserver 172.18.1.10 --disable-dhcp public 172.17.0.0/16

# ネットワークとルータを作成

- プライベートネットワークを作成

- neutron net-create private --  
provider:network\_type vxlan

- neutron subnet-create --name private-subnet --  
gateway 10.0.0.1 --allocation-pool  
start=10.0.0.2,end=10.0.0.254 --dns-nameserver  
172.18.1.10 private 10.0.0.0/24

# ネットワークとルータを作成

- ルータを作成

- neutron router-create router1
- neutron router-gateway-set router1 public
- neutron router-interface-add router1  
subnet=private-subnet

# ロードバランサーサービス作成順番

- ロードバランサーを作成
- リスナーを作成し、ロードバランサーに紐付け
- プールを作成(このときリスナーを参照)
- メンバーを作成して、プールに追加
- ヘルスマニターを作成
- Floating IPを作成し、VIPのポートに紐付け

# ロードバランサーを作成

- neutron lbaas-loadbalancer-create --name lb1 private-subnet

```
(neutron) lbaas-loadbalancer-show lb1
```

Field	Value
admin_state_up	True
description	
id	6585bc8b-a4ff-4768-865d-dde21b5cf2fd
listeners	{"id": "56b1f780-5521-43ab-a6e3-16db7c356772"}
name	lb1
operating_status	ONLINE
pools	{"id": "cacda9a9-7d90-40d2-bd0a-4aed51bbbd6"}
provider	octavia
provisioning_status	ACTIVE
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14
vip_address	10.0.0.9
vip_port_id	6fd12458-1c0c-4d82-9023-6e65e9864ec7
vip_subnet_id	8d0b5388-82a2-4a53-88cb-305c4fbf1617

# リスナーを作成

- neutron lbaas-listener-create --name listener1 --loadbalancer lb1 --protocol HTTP --protocol-port 80

```
(neutron) lbaas-listener-show listener1
```

Field	Value
admin_state_up	True
connection_limit	-1
default_pool_id	cacda9a9-7d90-40d2-bd0a-4aed51bbdbc6
default_tls_container_ref	
description	
id	56b1f780-5521-43ab-a6e3-16db7c356772
loadbalancers	{"id": "6585bc8b-a4ff-4768-865d-dde21b5cf2fd"}
name	listener1
protocol	HTTP
protocol_port	80
sni_container_refs	
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14

# プールを作成

- neutron lbaas-pool-create --name pool1 --lb-algorithm ROUND\_ROBIN --listener listener1 --protocol HTTP

```
(neutron) lbaas-pool-show pool1
```

Field	Value
admin_state_up	True
description	
healthmonitor_id	
id	cacda9a9-7d90-40d2-bd0a-4aed51bbdbc6
lb_algorithm	ROUND_ROBIN
listeners	{"id": "56b1f780-5521-43ab-a6e3-16db7c356772"}
loadbalancers	{"id": "6585bc8b-a4ff-4768-865d-dde21b5cf2fd"}
members	af440000-d68b-4b26-9a11-2a374000e5da 406f1b10-1c72-4182-a520-99c91e1ed377
name	pool1
protocol	HTTP
session_persistence	
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14

# メンバーを作成

- neutron lbaas-member-create --subnet private-subnet --address 10.0.0.6 --protocol-port 80 pool1

Field	Value
address	10.0.0.6
admin_state_up	True
id	406f1b10-1c72-4182-a520-99c91e1ed377
name	
protocol_port	80
subnet_id	8d0b5388-82a2-4a53-88cb-305c4fbf1617
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14
weight	1

# メンバーを作成

```
- neutron lbaas-member-create --subnet private-subnet --address 10.0.0.3 --protocol-port 80 pool1
```

Field	Value
address	10.0.0.3
admin_state_up	True
id	af440000-d68b-4b26-9a11-2a374000e5da
name	
protocol_port	80
subnet_id	8d0b5388-82a2-4a53-88cb-305c4fbf1617
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14
weight	1

# ヘルスマニターを作成

- lbaas-healthmonitor-create --delay 5 --max-retries 4 --timeout 10 --type HTTP --url\_path / --pool pool1

Field	Value
admin_state_up	True
delay	5
expected_codes	200
http_method	GET
id	f701e043-0192-402a-9116-ce6aeea83214
max_retries	4
max_retries_down	3
name	
pools	{"id": "cacda9a9-7d90-40d2-bd0a-4aed51bbbd6"}
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14
timeout	10
type	HTTP
url_path	/

# Floating IPを作成

- neutron floatingip-create public

Field	Value
created_at	2017-05-09T07:10:18Z
description	
fixed_ip_address	10.0.0.9
floating_ip_address	172.17.56.206
floating_network_id	60b6dfb0-eced-458b-baa8-47e6db00d8e6
id	3dc47a3b-becb-4a93-9e18-1b3c9852186a
port_id	6fd12458-1c0c-4d82-9023-6e65e9864ec7
project_id	6b2d4eb385a04fb7b7648b72de7a6f14
revision_number	4
router_id	cf3cc95a-e022-48c4-b54c-04e55158b76f
status	ACTIVE
tenant_id	6b2d4eb385a04fb7b7648b72de7a6f14
updated_at	2017-05-09T07:55:23Z

# Floating IPをVIPに紐付け

- neutron floatingip-associate
- 3dc47a3b-becb-4a93-9e18-1b3c9852186a
- 6fd12458-1c0c-4d82-9023-6e65e9864ec7
- **書式は** floatingip-associate <floating\_ip\_id>  
<load\_balancer\_vip\_port\_id>
- **これで外から172.17.56.206でロードバランサーに  
HTTPアクセスできる。**

Apache2 Ubuntu Default Page 2

172.17.56.206

アプリ va linux kvm openstack tech python network packages octavia DDPK devstack docker



# ubuntu

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/loaded with `/etc/init.d/apache2` or `apache2ctl`. Calling `sudo /bin/apache2`

Apache2 Ubuntu Default Page

172.17.56.206

アプリ va linux kvm openstack tech python network packages octavia DPDK devstack docker



# Apache2 Ubuntu Default Page1

## ubuntu

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server. The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `sudo /bin/apache2`

# ●HA構成(ACTIVE-STANDBY)

## ●シナリオ

- ロードバランサーサービスをHA構成で構築
- 自動でマスターとスレーブのアンフォラを作成するように、octaviaのロードバランサーポートロジックをactive-standbyに変更
- (Octaviaのデフォルトの設定はstandalone)

# HA構成

- /etc/octavia/octavia.confを以下のように編集
- loadbalancer\_topology = ACTIVE\_STANDBY
- コントロールワーカープロセスを再起動
- 同様にロードバランサーを作成

# HA構成

- アンフォラが2つ作成されていることが確認できる

ID	Name	Status	Task State	Power State	Networks
c6a50c34-8161-4e4c-98c1-1689e7fa519b	amphora-24733ab1-0ea3-42d5-be47-632b908bacb2	ACTIVE	-	Running	lb-mgmt-net=192.168.0.9; private=10.0.0.5
4ebf9da3-255e-4d99-8779-bdc7a07931a6	amphora-9e72ad76-9820-4c0b-afbb-43a9a824aaca	ACTIVE	-	Running	lb-mgmt-net=192.168.0.6; private=10.0.0.8
d3e591dd-86a7-4b92-b5b8-469cc656524d	apache_server-1	ACTIVE	-	Running	private=10.0.0.3
5f47911d-7d25-458b-9704-8bf36a773fdf	apache_server-2	ACTIVE	-	Running	private=10.0.0.6

- HA構成ロードバランサーは1つ

id	name	tenant_id	vip_address	provisioning_status	provider
1fd4be4c-3735-4aec-8d70-8014c2c46e98	lb	6b2d4eb385a04fb7b7648b72de7a6f14	10.0.0.13	ACTIVE	octavia

# HA構成

DBを覗くと  
どのアンフォラがマスターかスレーブを  
確認できる

```
mysql> select load_balancer_id, compute_id, role, vrrp_ip, lb_network_ip, ha_ip from amphora;
+-----+-----+-----+-----+-----+-----+
| load_balancer_id | compute_id | role | vrrp_ip | lb_network_ip | ha_ip |
+-----+-----+-----+-----+-----+-----+
| 1fd4be4c-3735-4aec-8d70-8014c2c46e98 | c6a50c34-8161-4e4c-98c1-1689e7fa519b | MASTER | 10.0.0.5 | 192.168.0.9 | 10.0.0.13 |
| 1fd4be4c-3735-4aec-8d70-8014c2c46e98 | 4ebf9da3-255e-4d99-8779-bdc7a07931a6 | BACKUP | 10.0.0.8 | 192.168.0.6 | 10.0.0.13 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

# HA構成 フェイルオーバー

- マスターアンフォラが落ちた場合、スレーブアンフォラがマスターアンフォラになる
- 落ちたマスターアンフォラのインスタンスが削除され、新しいアンフォラが自動的に作成され、その役割はスレーブである

ただし、このときDB上のroleカラムが変更されることはない！

つまり、フェイルオーバーが起きた時に、DBのroleと実際のアンフォラの役割が逆となる！

# どうすれば、本当の役割確認できるか

- アンフォラにログインして確認するしかない
- 以下の方法でアンフォラにsshできる
  - \$ ssh -i /etc/octavia/.ssh/octavia\_ssh\_key
  - ubuntu@192.168.0.9
  - ログインした後、`sudo ip netns exec amphora-haproxy ip addr`を実行すると、

```
ubuntu@amphora-24733ab1-0ea3-42d5-be47-632b908bacb2:~$ sudo ip netns exec amphora-haproxy ip addr
sudo: unable to resolve host amphora-24733ab1-0ea3-42d5-be47-632b908bacb2
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth1: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1450 qdisc pfifo_fast state UP group default qlen 1000
   link/ether fa:16:3e:05:9b:21 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.5/24 brd 10.0.0.255 scope global eth1
       valid_lft forever preferred_lft forever
   inet 10.0.0.13/32 scope global eth1
       valid_lft forever preferred_lft forever
   inet 10.0.0.13/24 brd 10.0.0.255 scope global secondary eth1:0
       valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fe05:9b21/64 scope link
       valid_lft forever preferred_lft forever
```

# どうすれば、本当の役割確認できるか

- \$ ssh -i /etc/octavia/.ssh/octavia\_ssh\_key
- ubuntu@192.168.0.6
- ログインした後、sudo ip netns exec amphora-haproxy ip addrを実行すると、
- Inet 10.0.0.13/32 scope global eth1がない！
- =>これはスレーブ

```
ubuntu@amphora-9e72ad76-9820-4c0b-afbb-43a9a824aaca:~$ sudo ip netns exec amphora-haproxy ip addr
sudo: unable to resolve host amphora-9e72ad76-9820-4c0b-afbb-43a9a824aaca
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth1: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1450 qdisc pfifo_fast state UP group default qlen 1000
   link/ether fa:16:3e:1e:12:a1 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.8/24 brd 10.0.0.255 scope global eth1
       valid_lft forever preferred_lft forever
   inet 10.0.0.13/24 brd 10.0.0.255 scope global secondary eth1:0
       valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fe1e:12a1/64 scope link
       valid_lft forever preferred_lft forever
```