

# Load Balancing as a Service

Mitaka and Beyond

---

**Doug Fish**

  
irc: doug-fish

**Michael Johnson**

 **HPE Helion**  
irc: johnsom

**Stephen Balukoff**

 **bluebox**  
AN IBM COMPANY  
irc: sbalukoff

users  
**openstack**  
summit  
devs

# Agenda

- Introduction
  - Who's involved?
  - OpenStack User Survey
- LBaaS v2.0 in Mitaka
  - Horizon Dashboard (doug-fish)
  - L7, Pool Sharing (sbalukoff)
- Octavia (johnsom)
  - Overview
  - What's New
  - Roadmap
- Related Sessions and Design Summit

# Who's Involved?



# April 2016 OpenStack User Survey

Which Neutron features are actively used, interested in, or planned for use?

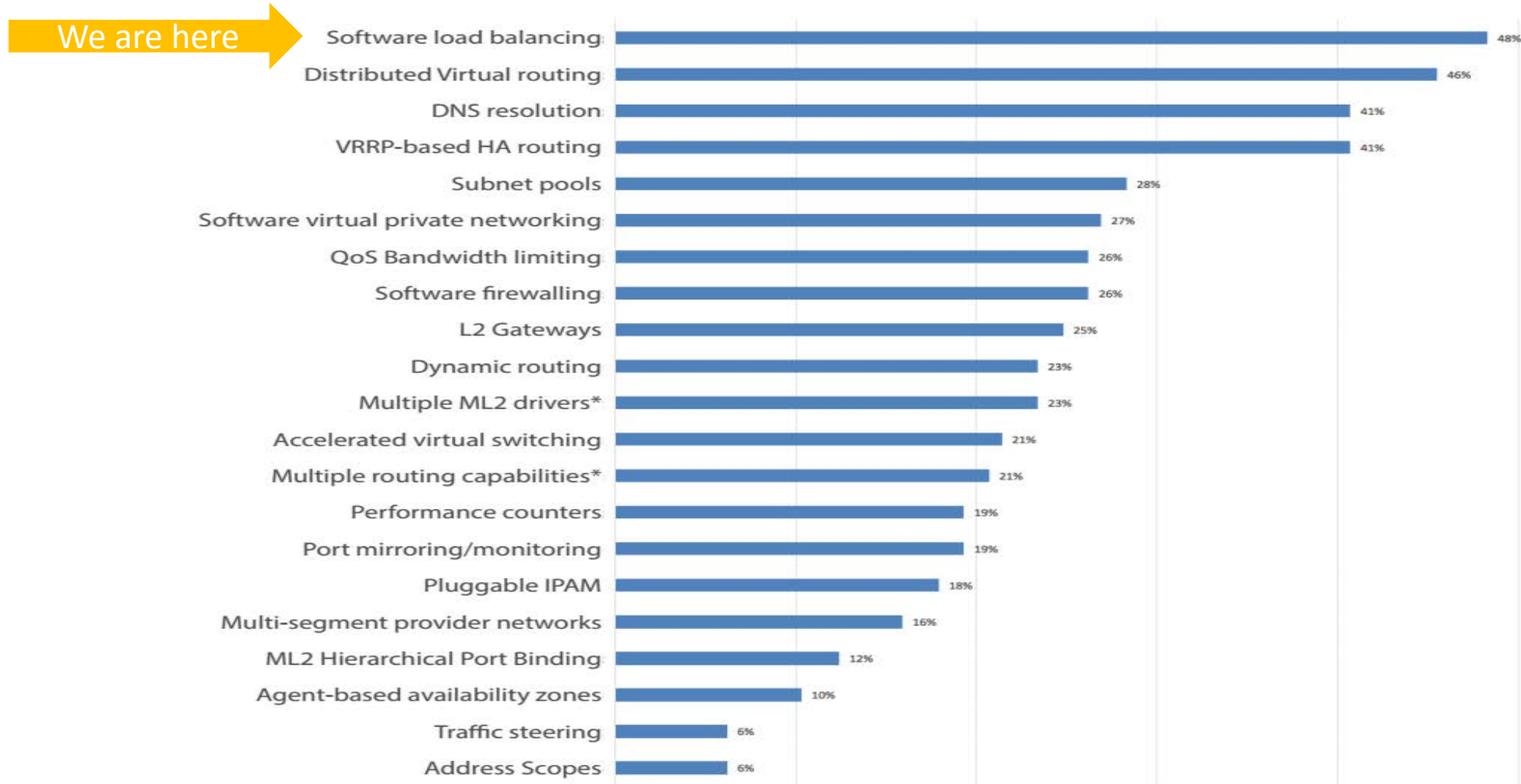


Figure 7.2 n=145

\*in the same deployment

Percentages are rounded to the nearest whole number; bar length shows fractions.

# LBaaS v2.0 in Mitaka



# Horizon Dashboard

openstack demo demo

## Load Balancers

Filter + Create Load Balancer Delete Load Balancers

<input type="checkbox"/>	Name ^	Description	Operating Status	Provisioning Status	IP Address	Listeners	Actions
<input type="checkbox"/>	> Load Balancer 1	-	Online	Active	10.0.0.10	1	Edit ▾

Displaying 1 item

Project ^

Compute ▾

Network ^

- Network Topology
- Networks
- Routers

Load Balancers

Orchestration ▾

Identity ▾

Developer ▾

# What can you do with the dashboard?

- Focus of our efforts was on creating a new load balancer, including
  - the associated listener
  - default pool
  - health monitor
  - populating the pool with members
- You can also
  - Add and remove listeners
  - Add and remove members from the default pool
  - Remove and re-create the default pool
  - Update the health monitor
- What can't you do?
  - You can't do L7 load balancing
  - You can't work with pools outside of the default pool on a listener
  - Monitoring



## Log in

User Name

Password

Connect



# How to try out the dashboard

In your localrc add:

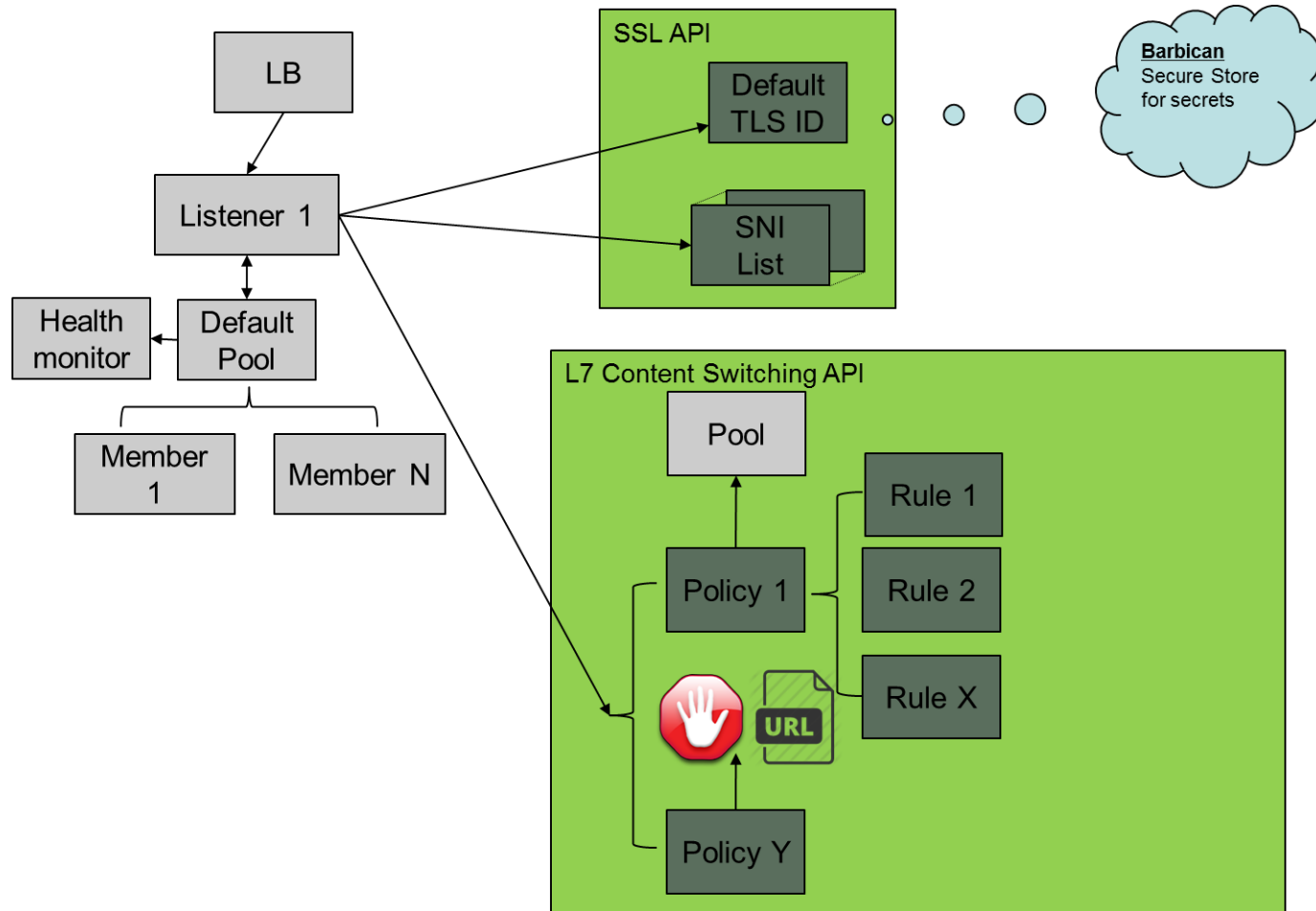
```
enable_plugin neutron-lbaas-dashboard https://github.com/openstack/neutron-lbaas-dashboard
```

or

```
enable_plugin neutron-lbaas-dashboard https://github.com/openstack/neutron-lbaas-dashboard stable/mitaka
```

Note: Your version of neutron-lbaas-dashboard needs to match the level of Horizon. Although the master version of neutron-lbaas-dashboard is compatible with stable/mitaka Horizon the day I made this slide, I don't expect that to last long!

# L7 Content Switching



# Why L7?

- By default all requests get routed to the listener's default\_pool
- As the client application grows, this behavior is sometimes not desirable for all requests. (ex. “application server” versus “static content” pools.)
- L7 functionality allows request routing decisions to be made based on information embedded within the request, so all back-ends appear to come from the same front-end (from the client’s perspective).
- Works for HTTP / Terminated HTTP protocols only (right now)
- Horizon UI for L7 should land in Newton

# L7 Rules

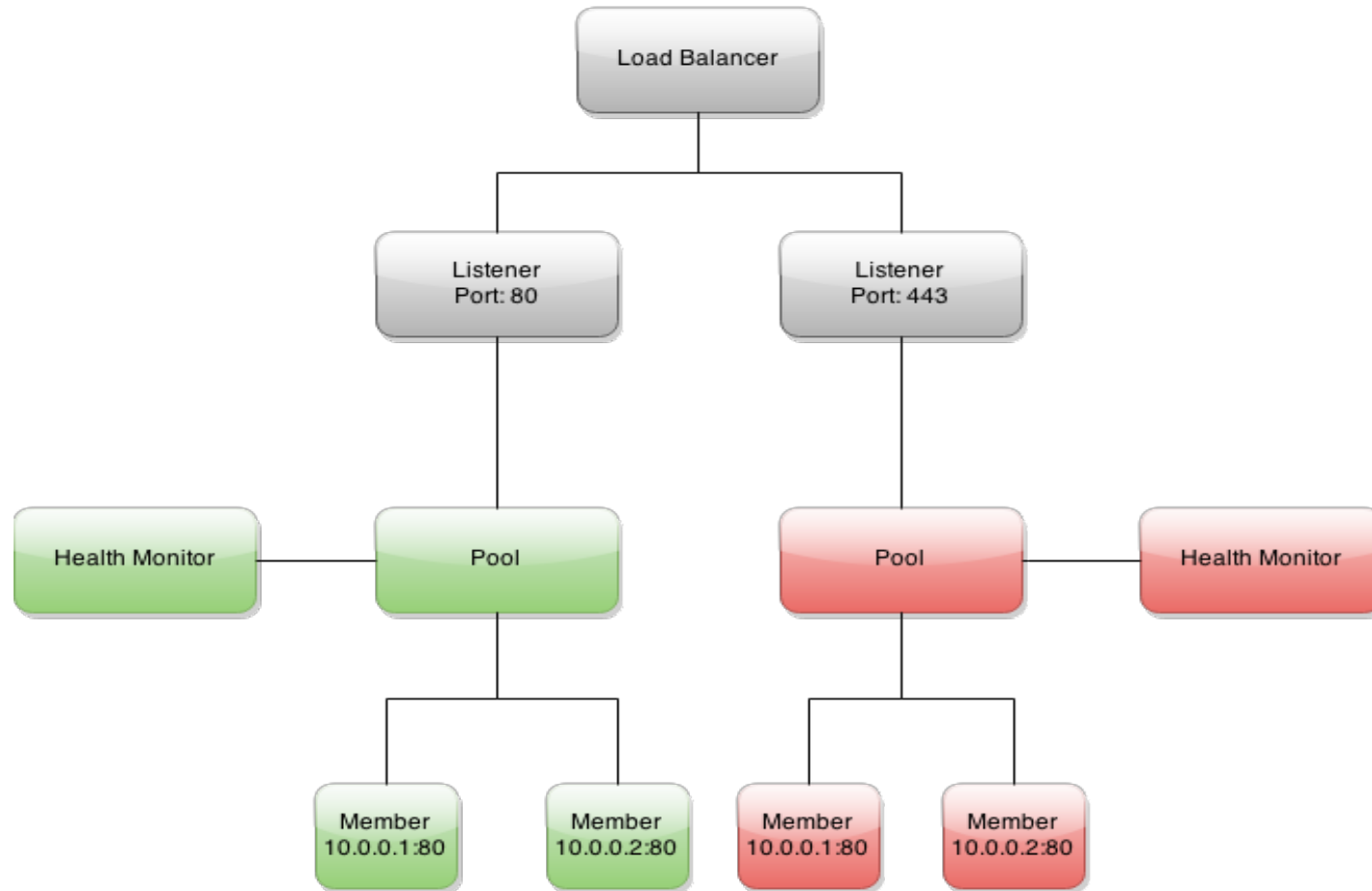
- An L7 Rule is a single statement of logic that will be matched against client requests.
- L7 Rules evaluate to True or False
- Examples:
  - Request URL starts with `"/api"`
  - Request cookie `"client-group"` is equal to the string `"group1"`
  - Request header `"X-My-header"` matches the regular expression `".*somestring.*"`
- See documentation for complete listing of rule and comparison types.

# L7 Policies

- L7 Policies are a collection of L7 Rules
- L7 Policies are assigned to a listener.
- All L7 Rules on a given policy are logically ANDed together
- If a logical OR is needed, create multiple policies with the same action (or use regular expressions, if possible).
- L7 Policies define an action that will be taken if all the policy's rules match (will usually be REDIRECT\_TO\_POOL)

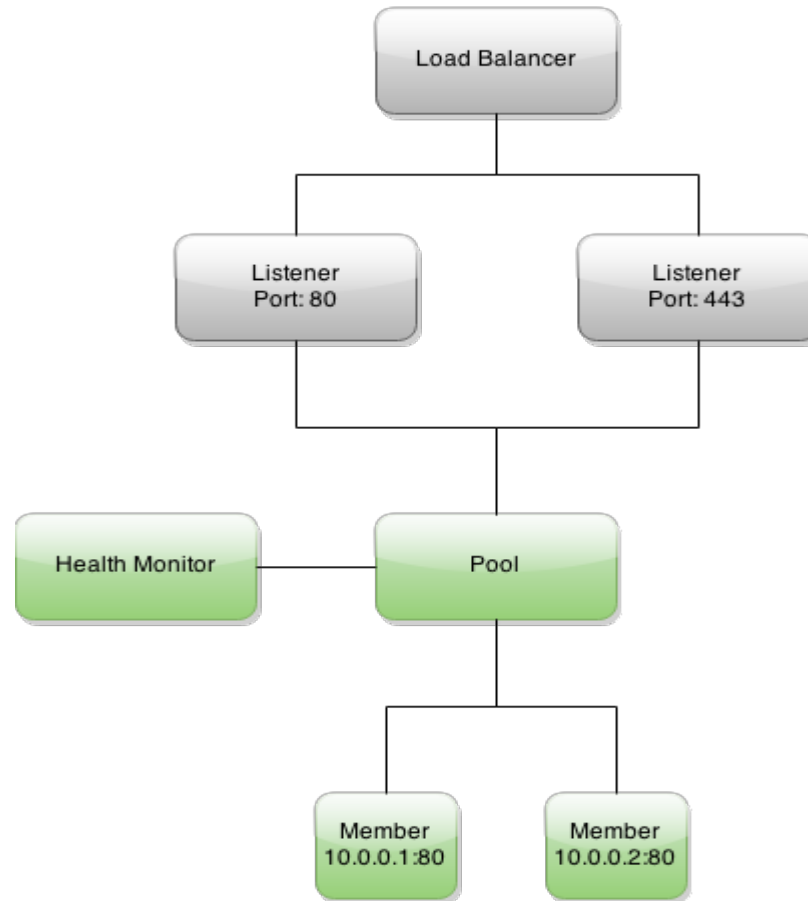
# Pool Sharing

Without Pool Sharing



# Pool Sharing

With Pool Sharing



# Layer 7 Rules / Shared Pools Demo

- The goal:
  - Create an HTTP listener1 with a default pool (pool1 that contains server1)
  - Add an L7 Policy and L7 rule which sends all requests which start with "/api" to pool2 (which contains server2)
  - Create a listener2 which uses pool2 as its default pool
- The Setup:
  - Mitaka devstack using neutron-lbaas with the Octavia driver
  - Before these slides, I launched two application servers on the private subnet with simple web servers.
  - Each server responds with a line identifying the server
  - Security groups are set up permissively (for this demo)



# Demo setup

```
stack@empty: ~  
stack@empty:~$ curl http://10.0.0.5:8000/  
You are on server1  
stack@empty:~$ curl http://10.0.0.6:8000/  
You are on server2  
stack@empty:~$ █
```

# Create load balancer "lb1"

```
stack@empty: ~  
stack@empty:~$ neutron lbaas-loadbalancer-create --name lb1 private-subnet  
Created a new loadbalancer:  
+-----+-----+  
| Field                | Value                                     |  
+-----+-----+  
| admin_state_up       | True                                     |  
| description          |                                           |  
| id                   | 922a035e-f2c7-4241-b7ba-3c4c1a711d5d   |  
| listeners            |                                           |  
| name                 | lb1                                     |  
| operating_status     | OFFLINE                                 |  
| pools               |                                           |  
| provider             | octavia                                 |  
| provisioning_status  | PENDING_CREATE                         |  
| tenant_id           | 491762c4226e4b7ba0a3d1a76352343e     |  
| vip_address          | 10.0.0.7                               |  
| vip_port_id         | 0e816516-486b-411d-92af-82160fbdb690  |  
| vip_subnet_id       | 2ca46d87-5c43-4f06-9cb1-4cc72cac002e  |  
+-----+-----+  
stack@empty:~$
```

# Create listener "listener1"

```
stack@empty: ~
stack@empty:~$ neutron lbaas-listener-create --name listener1 --loadbalancer lb1
--protocol HTTP --protocol-port 80
Created a new listener:
+-----+-----+
| Field                | Value                                |
+-----+-----+
| admin_state_up       | True                                  |
| connection_limit     | -1                                    |
| default_pool_id      |                                        |
| default_tls_container_ref |                                        |
| description          |                                        |
| id                   | 397a2527-a352-4694-91a5-82de2db17844 |
| loadbalancers        | {"id": "922a035e-f2c7-4241-b7ba-3c4c1a711d5d"} |
| name                 | listener1                             |
| protocol             | HTTP                                   |
| protocol_port        | 80                                     |
| sni_container_refs   |                                        |
| tenant_id            | 491762c4226e4b7ba0a3d1a76352343e    |
+-----+-----+
stack@empty:~$ █
```

# Expected output for a listener with no pool

```
stack@empty: ~  
stack@empty:~$ curl http://10.0.0.7/  
<html><body><h1>503 Service Unavailable</h1>  
No server is available to handle this request.  
</body></html>  
stack@empty:~$
```

# Create pool "pool1"; Make it listener1's default pool

```
stack@empty: ~
stack@empty:~$ neutron lbaas-pool-create --name pool1 --lb-algorithm ROUND_ROBIN
--listener listener1 --protocol HTTP
Created a new pool:
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| admin_state_up | True                                     |
| description     |                                           |
| healthmonitor_id |                                           |
| id              | 87c169eb-ffc3-47df-82a5-b16c8cf77229    |
| lb_algorithm    | ROUND_ROBIN                             |
| listeners       | {"id": "397a2527-a352-4694-91a5-82de2db17844"} |
| loadbalancers   | {"id": "922a035e-f2c7-4241-b7ba-3c4c1a711d5d"} |
| members        |                                           |
| name            | pool1                                    |
| protocol        | HTTP                                     |
| session_persistence |                                           |
| tenant_id      | 491762c4226e4b7ba0a3d1a76352343e      |
+-----+-----+
stack@empty:~$ curl http://10.0.0.7/
<html><body><h1>503 Service Unavailable</h1>
No server is available to handle this request.
</body></html>
stack@empty:~$
```

# Create member "member1"

```
stack@empty: ~
stack@empty:~$ neutron lbaas-member-create --subnet private-subnet --address 10.0.0.5 --protocol-port 8000 pool1
Created a new member:
+-----+
| Field          | Value                                     |
+-----+
| address        | 10.0.0.5                                 |
| admin_state_up | True                                     |
| id             | 95a1dc26-ef95-40e3-80da-0296c8f37f2e    |
| name           |                                           |
| protocol_port  | 8000                                     |
| subnet_id      | 2ca46d87-5c43-4f06-9cb1-4cc72cac002e   |
| tenant_id      | 491762c4226e4b7ba0a3d1a76352343e      |
| weight         | 1                                        |
+-----+
stack@empty:~$ curl http://10.0.0.7/
You are on server1
stack@empty:~$ curl http://10.0.0.7/
You are on server1
stack@empty:~$ curl http://10.0.0.7/
You are on server1
stack@empty:~$ █
```

# Create "pool2" on "lb1"(not associated with any listener)

```
stack@empty: ~
stack@empty:~$ neutron lbaas-pool-create --name pool2 --lb-algorithm ROUND_ROBIN
--loadbalancer lb1 --protocol HTTP
Created a new pool:
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| admin_state_up | True                                     |
| description    |                                           |
| healthmonitor_id |                                           |
| id             | 3ffc3b20-9eac-4c5d-bc8a-85e4e58c0bab    |
| lb_algorithm   | ROUND_ROBIN                             |
| listeners      |                                           |
| loadbalancers  | {"id": "922a035e-f2c7-4241-b7ba-3c4c1a711d5d"} |
| members       |                                           |
| name           | pool2                                    |
| protocol       | HTTP                                     |
| session_persistence |                                           |
| tenant_id      | 491762c4226e4b7ba0a3d1a76352343e      |
+-----+-----+
stack@empty:~$
```

# Create “member2” on pool2

```
stack@empty: ~
stack@empty:~$ neutron lbaas-member-create --subnet private-subnet --address 10.0.0.6 --protocol-port 8000 pool2
Created a new member:
+-----+-----+
| Field          | Value                                |
+-----+-----+
| address        | 10.0.0.6                             |
| admin_state_up| True                                  |
| id             | 45aba8db-6eb1-46a7-bce7-a6d8ca8315a0 |
| name          |                                       |
| protocol_port  | 8000                                  |
| subnet_id     | 2ca46d87-5c43-4f06-9cb1-4cc72cac002e |
| tenant_id     | 491762c4226e4b7ba0a3d1a76352343e    |
| weight        | 1                                     |
+-----+-----+
stack@empty:~$ curl http://10.0.0.6:8000/
You are on server2
stack@empty:~$
```



# Create L7 Policy “policy1” on listener1

```
stack@empty: ~  
stack@empty:~$ neutron lbaas-l7policy-create --action REDIRECT_TO_POOL --redirect-pool pool2 --listener listener1 --name policy1  
Created a new l7policy:  
+-----+  
| Field          | Value                                     |  
+-----+  
| action         | REDIRECT_TO_POOL                         |  
| admin_state_up | True                                      |  
| description    |                                           |  
| id             | 6660e598-b350-4d0f-908d-38de769af452    |  
| listener_id    | 397a2527-a352-4694-91a5-82de2db17844    |  
| name           | policy1                                   |  
| position       | 1                                         |  
| redirect_pool_id | 3ffc3b20-9eac-4c5d-bc8a-85e4e58c0bab    |  
| redirect_url    |                                           |  
| rules          |                                           |  
| tenant_id      | 491762c4226e4b7ba0a3d1a76352343e       |  
+-----+  
stack@empty:~$ █
```



# Create “listener2” with “pool2” as its default pool

```
stack@empty: ~
stack@empty:~$ neutron lbaas-listener-create --name listener2 --default-pool pool2 --protocol HTTP --protocol-port 81
Created a new listener:
+-----+-----+
| Field                | Value                                     |
+-----+-----+
| admin_state_up       | True                                     |
| connection_limit     | -1                                       |
| default_pool_id      | 3ffc3b20-9eac-4c5d-bc8a-85e4e58c0bab    |
| default_tls_container_ref |                                           |
| description          |                                           |
| id                   | b70b3468-26f4-48b1-b5f2-64ac5702b07f   |
| loadbalancers        | {"id": "922a035e-f2c7-4241-b7ba-3c4c1a711d5d"} |
| name                 | listener2                               |
| protocol             | HTTP                                     |
| protocol_port        | 81                                       |
| sni_container_refs  |                                           |
| tenant_id           | 491762c4226e4b7ba0a3d1a76352343e     |
+-----+-----+
stack@empty:~$ curl http://10.0.0.7:81/
You are on server2
stack@empty:~$ █
```

# It works!

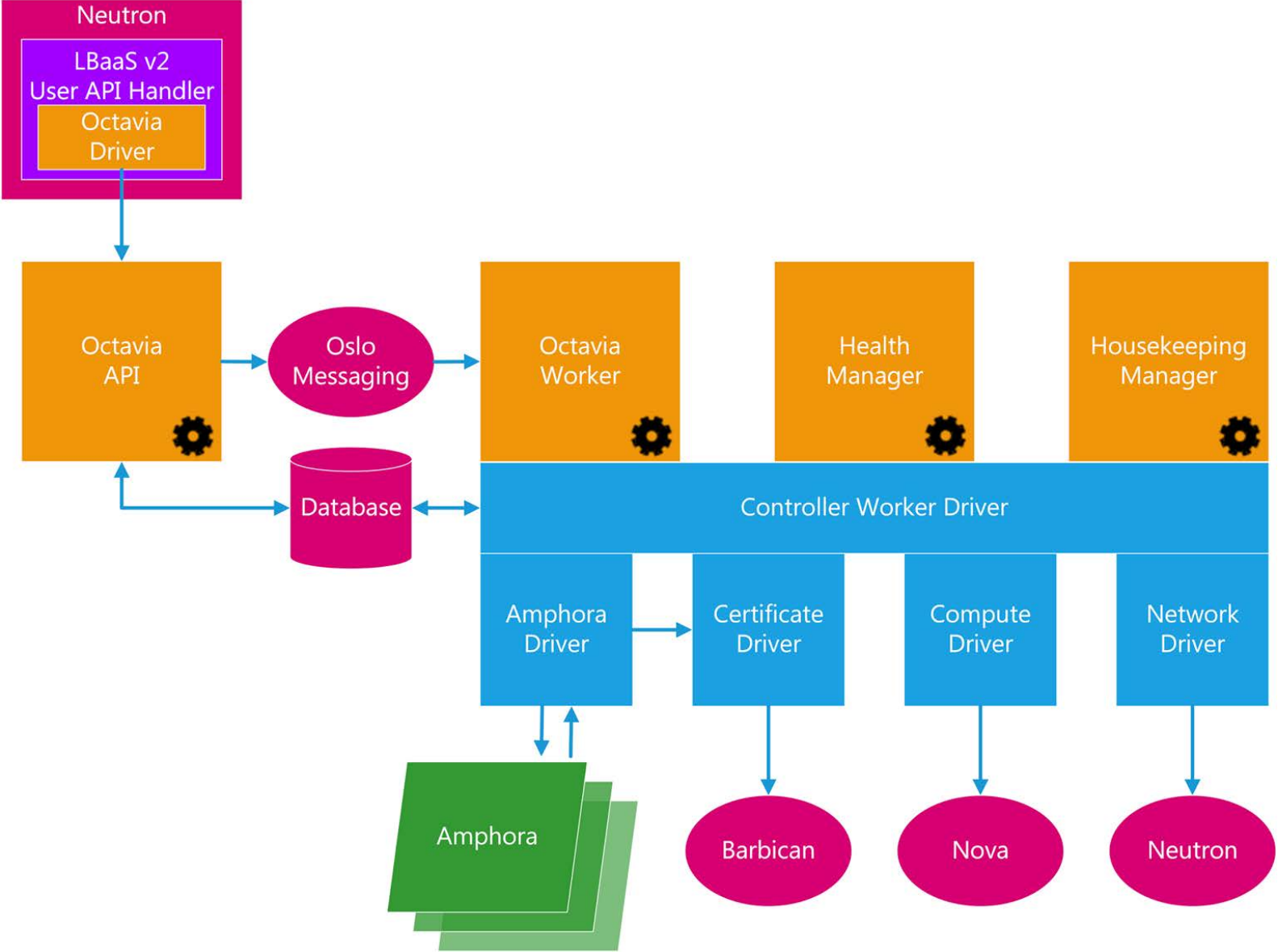
```
stack@empty: ~  
stack@empty:~$ curl http://10.0.0.7/  
You are on server1  
stack@empty:~$ curl http://10.0.0.7:81/  
You are on server2  
stack@empty:~$ curl http://10.0.0.7/api  
You are on server2  
stack@empty:~$ curl http://10.0.0.7/api/with/other/stuff/in/path  
You are on server2  
stack@empty:~$ curl http://10.0.0.7/without/other/stuff/in/path  
You are on server1  
stack@empty:~$ curl http://10.0.0.7/this/goes/to/pool1/  
You are on server1  
stack@empty:~$ curl http://10.0.0.7:81/this/goes/to/pool2/  
You are on server2  
stack@empty:~$ curl http://10.0.0.7/api/this/also/goes/to/pool2/  
You are on server2  
stack@empty:~$ █
```



# Octavia

OpenStack Load Balancing

# Octavia Component Design v0.5



# What is New in Octavia

## Active / Standby

- Transitions between active and standby Amphora in seconds (demonstrated in Tokyo)
- Optional anti-affinity for active and standby instance using nova anti-affinity filter
- Failed instance will be automatically rebuilt using the amphora failover flow
- Will not preempt standby in active state when new primary is built

## Automatic Amphora certificate rotation

- Octavia Housekeeping service rotates the TLS client certificate prior to expiration

## Layer 7 Rules

- Shared pools
- REJECT, REDIRECT\_TO\_POOL, REDIRECT\_TO\_URL policies
- HOST\_NAME, PATH, FILE\_TYPE, HEADER, COOKIE rules

## Single Call Actions – WIP

- Cascading delete
- Get-Me-A-Load-Balancer

## Glance tags for the Amphora image

- Allows you to change the Amphora boot image without restarting the Octavia worker

## OpenStack Bandit check gate

- Security scans every commit

## Amphora HAProxy running in a network namespace (0.8.1 releasing soon)

# Octavia Roadmap

Note: This roadmap WILL change based on the design sessions this week.

Octavia v0.5 Liberty



- Feature parity with existing reference driver
- Service virtual machines
- Spares pool failover

Octavia v1.0 – Mitaka?

- Amphora Active/Standby
- High Availability control plane
- Layer 7 rules
- Container support
- Flavor framework support

Octavia v2.0?

- Active/Active amphora
- Amphora horizontal scale

As Presented in Tokyo



# Octavia Roadmap

Note: This roadmap WILL change based on the design sessions this week.

## Octavia v0.5 Liberty ✓

- Feature parity with existing reference driver
- Service virtual machines
- Spares pool failover

## Octavia v0.8 – Mitaka ✓

- Amphora Active/Standby ✓
- Layer 7 rules ✓

## Octavia - Future

- Active/Active amphora
- Amphora horizontal scale
- Container support
- Flavor framework support
- High Availability control plane
- Single call actions

Austin Update

# Try Octavia yourself on DevStack

In your localrc add:

```
enable_plugin neutron-lbaas https://git.openstack.org/openstack/neutron-lbaas
```

```
enable_plugin octavia https://git.openstack.org/openstack/octavia.git
```

```
ENABLED_SERVICES+=,q-lbaasv2,octavia,o-cw,o-hk,o-hm,o-api
```

Operator API is at: <http://127.0.0.1:9876>

Operator API documentation: <http://www.octavia.io/review/master/main/octaviaapi.html>

neutron client: neutron lbaas-[loadbalancer-create]

Sample Vagrant and local.conf files are available under octavia/devstack/samples



# OpenStack Octavia

We are looking for contributors!

Freenode IRC: #openstack-lbaas

- <https://wiki.openstack.org/wiki/octavia>
- <http://www.octavia.io>
- <https://launchpad.net/octavia>
- <https://github.com/openstack/octavia>



 Photo by [Stuart Seeger](#)

# Related Sessions and Design Summit

## Sessions

Turn up the Heat with LBaaS v2

Thu 28 1:30pm-2:10pm - Austin Convention Center - Level 4 - MR 17 A/B

Deep Dive into Elastic Load Balancing Using Octavia

Thu 28 5:00pm-5:40pm - Austin Convention Center - Level 4 - Ballroom D

## Hands-on lab - RSVP required

Writing an AngularJS Plugin for Horizon

Thu 28 11:00am-12:30pm - JW Marriott Austin - Level 3 - Salon D

Install and Configure OpenStack Octavia

Thu 28 1:30pm-3:00pm - JW Marriott Austin - Level 3 - Salon E

## Design Summit

Neutron: Development track: future of \*-aaS high level services

Wed 27 1:50pm-2:30pm - Hilton Austin - Salon B

# Q & A / Panel discussion

<https://wiki.openstack.org/wiki/Neutron/LBaaS>

<https://wiki.openstack.org/wiki/Octavia>

IRC: #openstack-lbaas

We are irc: doug-fish, irc: johnsom, irc: sbalukoff