HAProxy

Ryan O'Hara
Principal Software Engineer, Red Hat
September 17, 2014

HAProxy

- Overview
- Capabilities
- Configuration
- OpenStack HA
- Neutron LBaaS
- Resources
- Questions

Overview

- Load Balancer
  - Layer 4 (TCP) and Layer 7 (HTTP)
  - Reverse Proxy
- Fast, reliable
  - Easy to handle 10k connections per second
- High Availability
  - Alone HAProxy is SPOF
  - Can use with Pacemaker or Keepalived for HA
- Comprehensive statistics and monitoring

Availability

- Available for RHEL6, RHEL7 and Fedora
  - Considered Tech Preview in RHEL6.5
  - Planned, but not commited, support in RHEL6.6
  - Load Balancer AddOn in RHEL6
  - Base OS in RHEL7

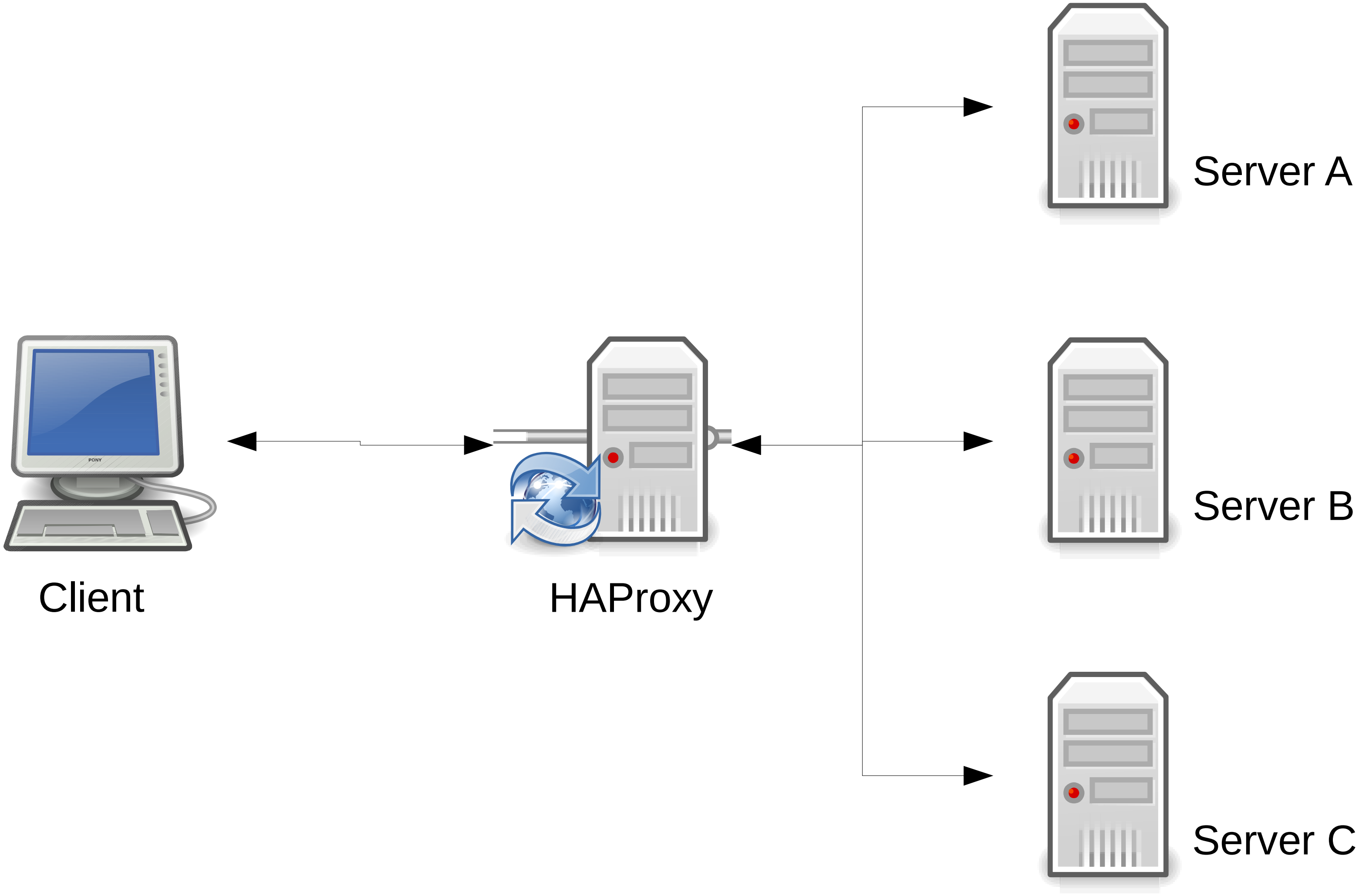
Capabilities

- ACLs
  - Extract some information, make decision
  - Block request, select backend, rewrite headers, etc.
- TCP mode (Layer 4)
  - Basic TCP services, SSL passthrough
  - Some ACLs available
- HTTP mode (Layer 7)
  - HTTP header inspection ACLs
  - Persistence with cookie insertion

Capabilities

- Stick Tables
  - Record information in table, eg. source address
  - Check table for new connections, select backend
- New features in HAProxy 1.5
  - SSL termination
  - Compression
  - Peers

HAProxy



HAProxy

- Client makes connection to HAProxy
- HAProxy makes connection to backend server
- Only first connection is considered for algorithms
- Backend server sees connection from HAProxy
  - HTTP can insert X-Forwarded-For header

Configuration

```
global
    daemon
    log /dev/log local2
    option redispatch
    retries 3
    maxconn 1000
    user haproxy
    group haproxy
```

Configuration

```
defaults
    log global
    option dontlognull
    timeout connect 500ms
    timeout client 30s
    timeout server 30s
```

Configuration – Proxy

```
listen http-proxy 192.168.1.201:80
    mode http
    option httpchk GET /test
    balance roundrobin
    timeout server 30s
    timeout client 30s
    server server-01 192.168.1.101:80 check inter 2s
    server server-02 192.168.1.102:80 check inter 2s
    server server-03 192.168.1.103:80 check inter 2s
```

Configuration – Proxy

```
frontend http-frontend
    bind 192.168.100.101:80
    bind 192.168.100.102:80
    default_backend http_backend

backend http-backend
    balance leastconn
    server server-01 192.168.1.101:81 check inter 2s
    server server-02 192.168.1.102:81 check inter 2s
    server server-03 192.168.1.103:81 check inter 2s
```

Configuration – Cookie Insertion

```
frontend horizon-proxy
    ...
    cookie SERVERID insert indirect nocache
    default_backend horizon-servers

backend horizon-servers
    ...
    server horizon-01 192.168.16.91:80 check inter 1s cookie horizon-01
    server horizon-02 192.168.16.92:80 check inter 1s cookie horizon-02
```

Configuration – Statistics

```
listen stats
    bind *:81
    mode http
    stats enable
    stats-uri /haproxy?stats
```

Configuration – SSL Termination

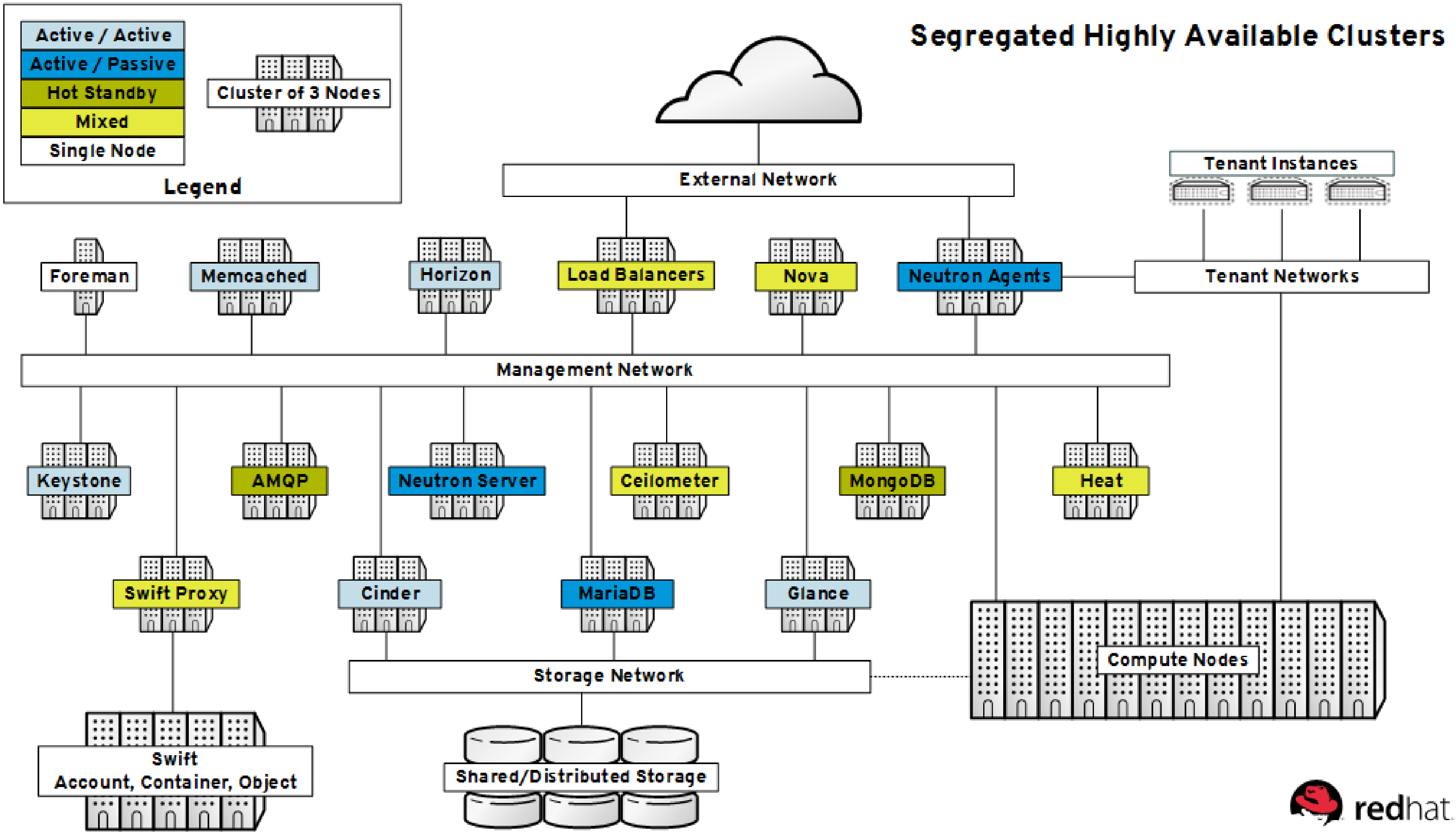
```
frontend http-proxy
    mode http
    bind 10.15.85.31:80
    redirect scheme https if !{ ssl_fc }

frontend https-proxy
    mode http
    bind 10.15.85.31:443 ssl crt /etc/pki/haproxy.pem
    default_backend http-servers
```

Configuration – Stick Tables

```
backend galera-servers
    ...
    stick-table type ip size 1
    stick on dst
    ...
    server galera-01 192.168.16.58:3306 check inter 1s port 9200
    server galera-01 192.168.16.59:3306 check inter 1s port 9200
    ...
```

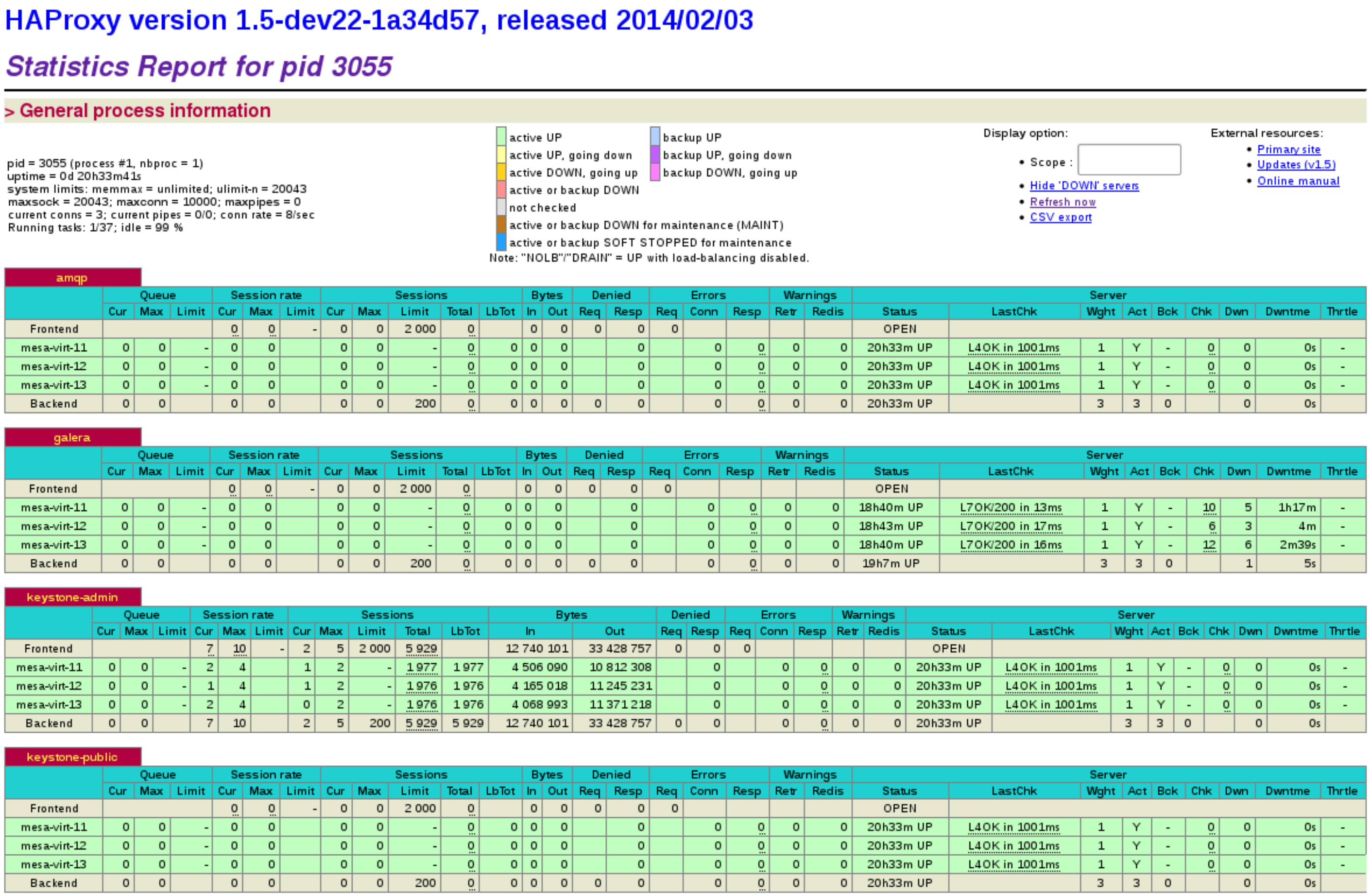
OpenStack HA



OpenStack HA

- Key component in OpenStack HA architecture
- Pacemaker provides IP failover, service monitoring
- HAProxy performs load balaning, health checks
  - All OpenStack API services have unique VIPs
  - VIPs on both public and private networks
  - MySQL (Galera), QPID, RabbitMQ

OpenStack HA



HAProxy version 1.5-dev22-1a34d57, released 2014/02/03

Statistics Report for pid 3055

> General process information

pid = 3055 (process #1, nbproc = 1)
uptime = 0d 20h33m41s
system limits: memmax = unlimited; ulimit-n = 20043
maxsock = 20043; maxconn = 10000; maxpipes = 0
current conns = 3; current pipes = 0/0; conn rate = 8/sec
Running tasks: 1/37; idle = 99 %

🟩 active UP	🟦 backup UP
🟨 active UP, going down	🟪 backup UP, going down
🟧 active DOWN, going up	🟪 backup DOWN, going up
🟥 active or backup DOWN	
⬜ not checked	
🟫 active or backup DOWN for maintenance (MAINT)	
🟦 active or backup SOFT STOPPED for maintenance	

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:

- Primary site
- Updates (v1.5)
- Online manual

amqp

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0		0	0	0	0	0					OPEN								
mesa-virt-11	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-12	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-13	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	200	0	0	0	0	0	0		0	0	0	0	20h33m UP		3	3	0		0	0s	

galera

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0		0	0	0	0	0					OPEN								
mesa-virt-11	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	18h40m UP	L7OK/200 in 13ms	1	Y	-	10	5	1h17m	-
mesa-virt-12	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	18h43m UP	L7OK/200 in 17ms	1	Y	-	6	3	4m	-
mesa-virt-13	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	18h40m UP	L7OK/200 in 16ms	1	Y	-	12	6	2m39s	-
Backend	0	0		0	0		0	0	200	0	0	0	0	0	0		0	0	0	0	19h7m UP		3	3	0		1	5s	

keystone-admin

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				7	10	-	2	5	2 000	5 929		12 740 101	33 428 757	0	0	0					OPEN								
mesa-virt-11	0	0	-	2	4		1	2	-	1 977	1 977	4 506 090	10 812 308		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-12	0	0	-	1	4		1	2	-	1 976	1 976	4 165 018	11 245 231		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-13	0	0	-	2	4		0	2	-	1 976	1 976	4 068 993	11 371 218		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
Backend	0	0		7	10		2	5	200	5 929	5 929	12 740 101	33 428 757	0	0		0	0	0	0	20h33m UP		3	3	0		0	0s	

keystone-public

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0		0	0	0	0	0					OPEN								
mesa-virt-11	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-12	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
mesa-virt-13	0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	20h33m UP	L4OK in 1001ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	200	0	0	0	0	0	0		0	0	0	0	20h33m UP		3	3	0		0	0s	

OpenStack LBaaS

- Load balance client traffic from one network to application services running on virtual machines
- TCP and HTTP
- Session persistence
- Health monitoring
- API for rapid deployment
- Drivers for various load balancers, including HAProxy

OpenStack LBaaS

- Pools
  - Logical group of members
  - Virtual machines that provide service (eg. Httpd)
  - Same subnet as members
  - Policy for health checking
  - Load-balancing algorithm
    - ROUND_ROBIN, LEAST_CONNECTIONS, SOURCE_IP
  - Protocol
    - TCP, HTTP, HTTPS

OpenStack LBaaS

- Member
  - Correspond to a service running on a virtual machine
  - Designated by IP address and port
  - Also have weight attribute
    - Default weight is 1
    - 0 : Member will not be considered for new connection, can still handle persistent connections
    - N : Member will receive traffic proportional to weight, relative to total combined weight of all members in pool

OpenStack LBaaS

- Health monitors
  - Periodic check of members
  - Associated with pool
  - All members of pool get health check
  - Multiple health monitors per pool
    - Active member must pass all health checks
  - Types (PING, TCP, HTTP, HTTPS)
  - Delay, retry, timeout

OpenStack LBaaS

- Member
  - Service running on backend virtual machine
  - IP address and port
- Health monitor
  - Check member health
  - Must be associated with a pool

OpenStack LBaaS

- VIP
  - IP address and port for incoming connections
  - Protocol (TCP, HTTP, HTTPS*)
  - Connection limit
    - Default is -1 (unlimited)
  - Session persistence
    - SOURCE_IP, HTTP_COOKIE, APP_COOKIE
  - Associate with pool
    - Starts haproxy

OpenStack LbaaS – Deployment

- Create a pool
  - # neutron lb-pool-create
- Create members, add members to pool
  - # neutron lb-member-create
- Create health monitors, associate with pool
  - # neutron lb-healthmonitor-create
  - # neutron lb-healthmonitor-associate
- Create VIP, associate with pool
  - # neutron lb-vip-create

OpenStack LBaaS

- HAProxy not started until VIP is created
- One haproxy process per VIP
- Network namespace
  - qlbaas-${POOL_ID}
- State information
  - /var/lib/neutron/lbaas/${POOL_ID}/...
    - conf
    - pid
    - sock

OpenStack LbaaS – Future

- New LbaaS v2 API
- Layer 7 rules
- SSL termination
- Octavia
  - Potential replacement to LbaaS
  - Deploy load balancer in virtual machine
  - New project, still being defined

Resources

- http://www.haproxy.org/
- http://cbonte.github.io/haproxy-dconv/configuration-1.5.html
- https://openstack.redhat.com/LBaaS
- https://wiki.openstack.org/wiki/Neutron/LBaaS

Questions?