

- Octavia LB** 3
- Octavia** 3
 - 3
 - 4
 - 11
 - 12
 - 12
- LoadBalancer** 14
 - network_tasks.AllocateVIP 21
 - network_tasks.PlugVIP 22
- Amphora** 24
 - Amphora Agent 25
 - AmphoraAPIClient 26
 - AmphoraePostVIPPlug 26
 - Keepalived 28
 - 31
- haproxy** 32
 - 34
 - 37
 - CalculateDelta 38
 - HandleNetworkDeltas 39
 - AmphoraePostNetworkPlug 39
 - ListenersUpdate 41
- L7policy, L7rule Health Monitor** 42
 - 1. 44
 - 2. 45
- Amphora** 47
 - CA SSL 47
 - Amphora Agent 49
 - AmphoraAPIClient가 51
- Amphora** () 52
 - Health Manager 52
 - Amphora 52
 - failover() 57
 - 61
- Neutron-lbaas vs. LBaaS v2 API vs. Octavia vs. Octavia v2 API** 64
 - 64

Octavia LB

Octavia

Octavia OpenStack

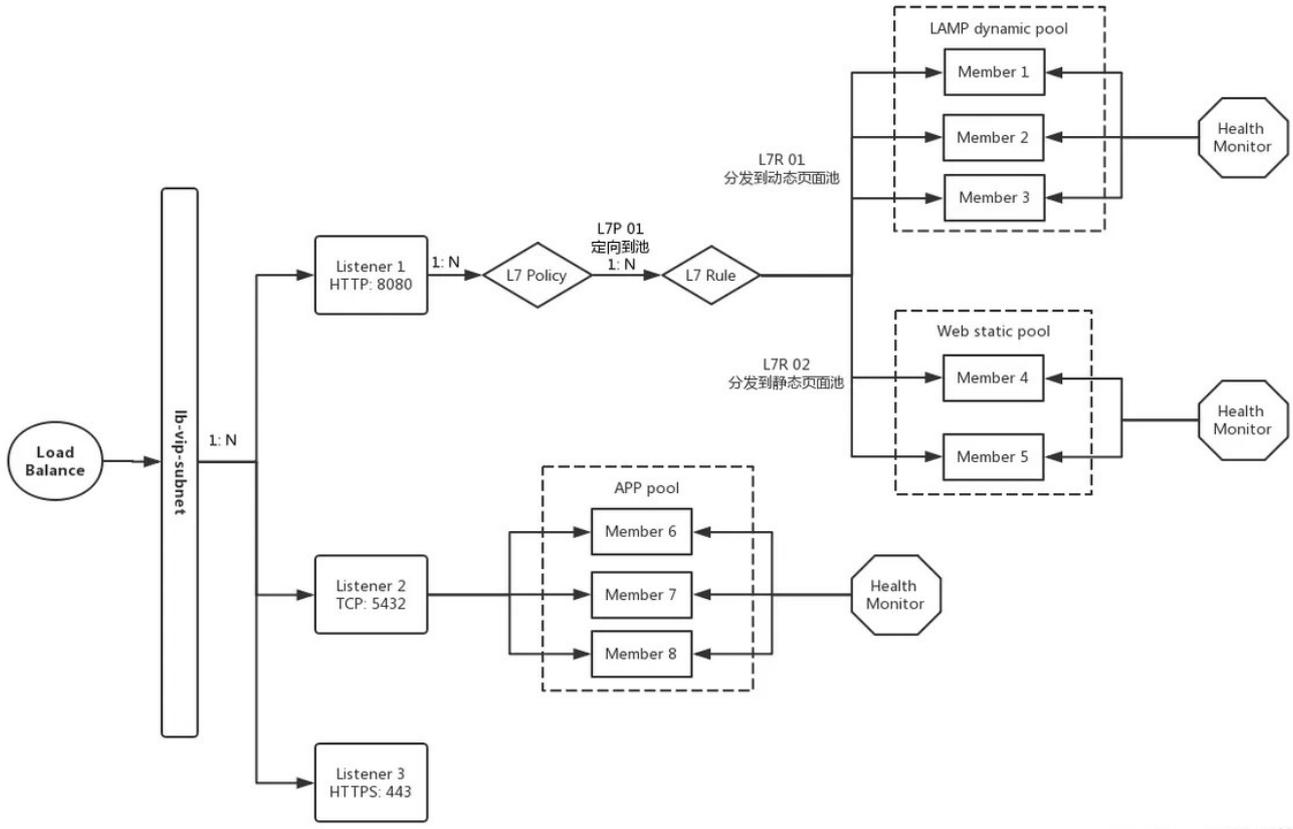
Pike OpenStack Extension Octavia Queens neutron-lbaas neutron-lbaas

. Neutron-lbaas is now deprecated.

Octavia neutron-lbaas가 API(Neutron/LBaaS/Deprecation) 가 가 neutron-lbaas가 Neutron LBaaS가

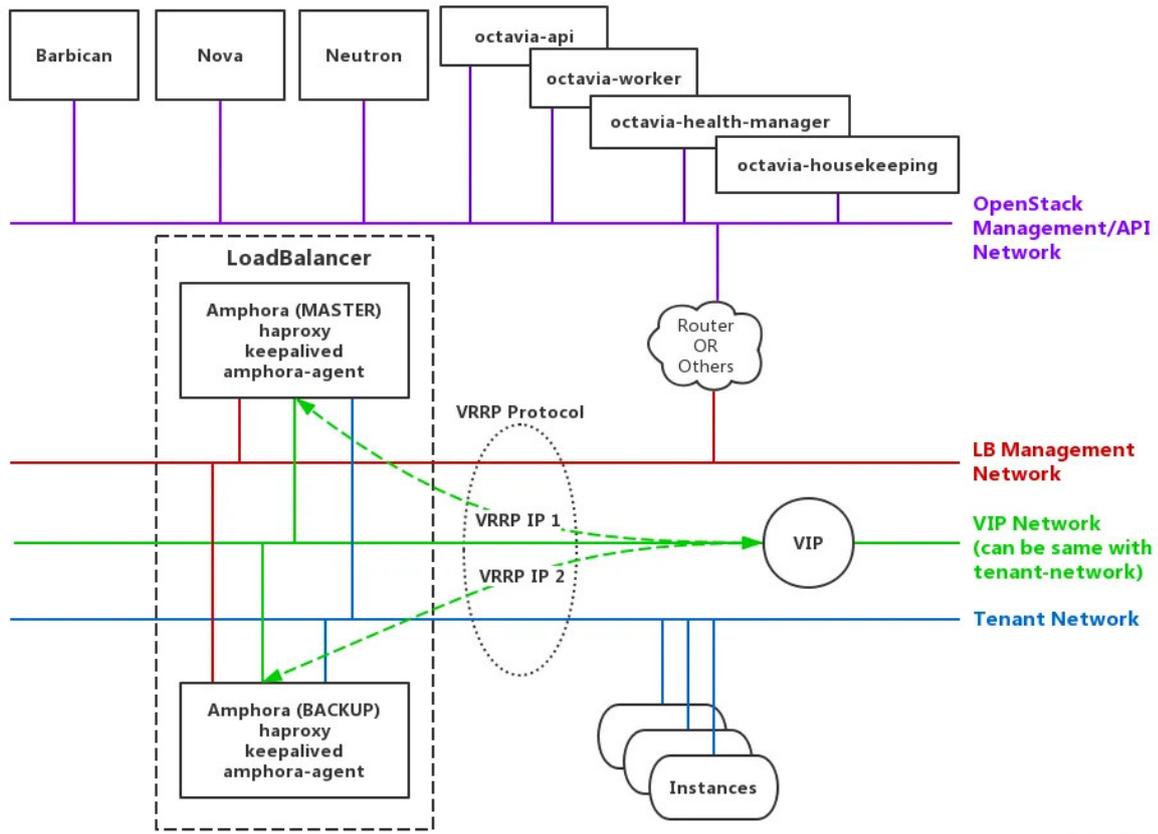
Rocky OpenStack LBaaS Octavia Octavia

- LBaaS : OpenStack LB()가 가
- loadbalancer :
- VIP : IP VIP가 , VIP (: ,)
- Pool :
- (Member) : Pool Real Server
- Health Monitor : Pool Pool Health Check
- L7 : 7 (: , URL ,)
- L7 : L7 7 (:)



<https://blog.@51CTO博客>

가?

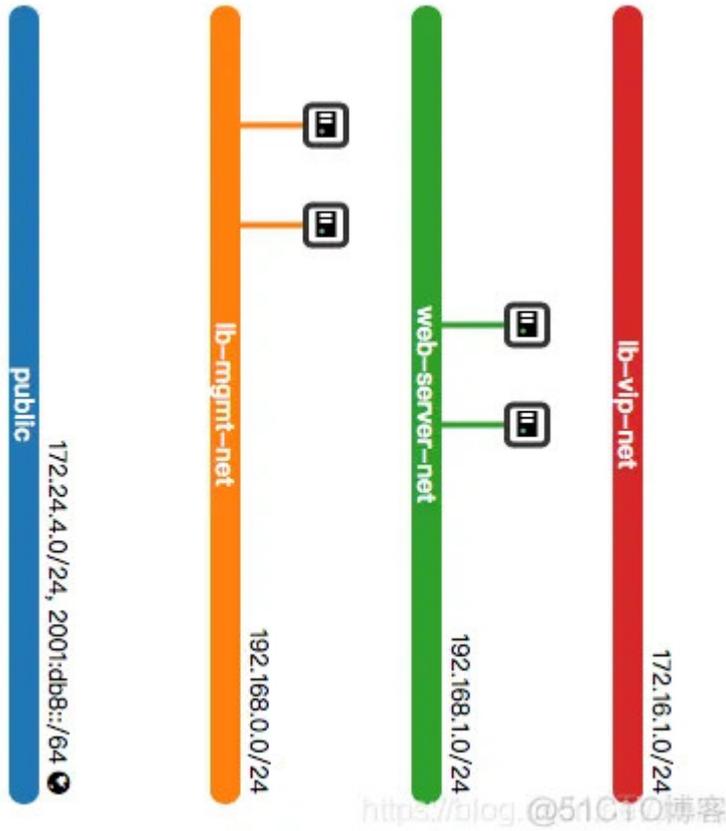


<https://blog.@51CTO博客>

Octavia가

Octavia

- Amphora(e) : Octavia
- lb-mgmt-net : OpenStack Management/API Network
Amphora Octavia
- tenant-net : 가
- vip-net : VIP
- : vip-net -



1 . VIP . VIP . DHCP .

Create Load Balancer

Provide the details for the load balancer.

Load Balancer Details

- Listener Details *
- Pool Details *
- Pool Members
- Monitor Details *

名称 (Name): Load Balancer 1

描述 (Description):

IP address

Subnet *: lb-vip-subnet

Buttons: [取消] [返回] [下一步] [Create Load Balancer]

2 . 가
http://<VIP>:8080/.

Create Load Balancer



Load Balancer Details

Provide the details for the listener.



Listener Details

名称

描述

Listener-1

Pool Details *

协议 *

Port *

Pool Members

HTTP

8080

Monitor Details *

取消

< 返回

下一步 >

Create Load Balancer

3

RR

Create Load Balancer



Load Balancer Details

Provide the details for the pool.



Listener Details

名称

描述

Pool-1

Pool Details

Method *

Pool Members

ROUND_ROBIN

Monitor Details *

取消

< 返回

下一步 >

Create Load Balancer

4

가

Create Load Balancer



Load Balancer Details

Listener Details

Pool Details

Pool Members

Monitor Details *

Add members to the load balancer pool.



Allocated Members 2

IP Address *	Subnet *	Port *	Weight	
192.168.1.14	web-server-subnet	80	1	移除
192.168.1.6	web-server-subnet	80	1	移除

Add external member

Available Instances

Q 筛选

名称	IP Address	
amphora-caa6ba0f-1a68-4f22-9be9-8521695ac4f4	192.168.0.13	添加
amphora-bcff6f9e-4114-4d43-a403-573f1d97d27e	192.168.0.11	添加
server-1	192.168.1.14	添加
server-2	192.168.1.6	添加

取消

< 返回

下一步 >

Create Load Balancer

5 . 가

PING

Create Load Balancer



Load Balancer Details

Listener Details

Pool Details

Pool Members

Monitor Details

Provide the details for the health monitor.



Monitor type *

PING

Interval (sec) *

5

Retries *

3

Timeout (sec) *

5

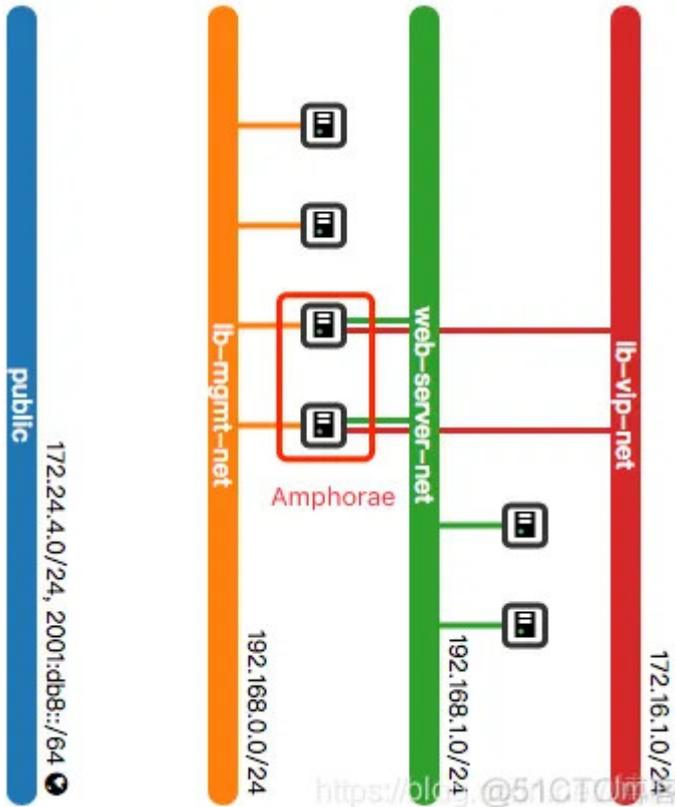
取消

< 返回

下一步 >

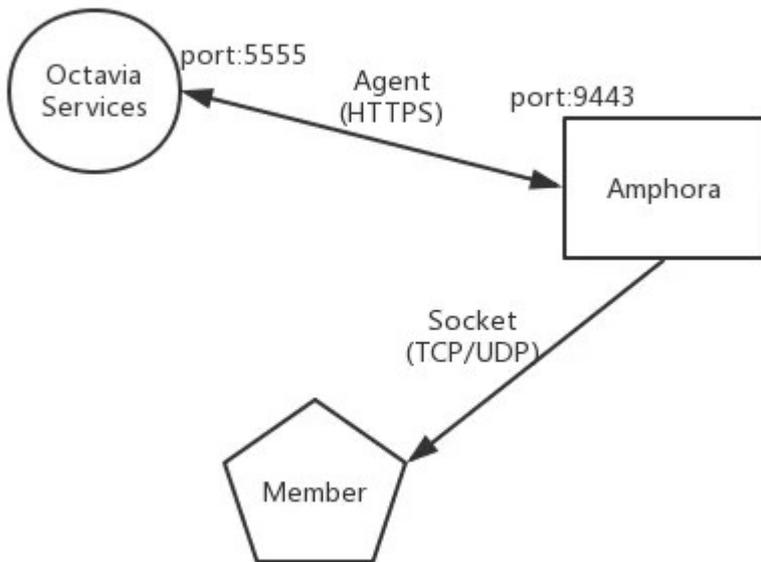
Create Load Balancer

VIP, Member Octava . Amphorae가 가



Octavia Amphora Provider

- Amphora (HAProxy) 가 (Keepalived) Octavia VIP
- Octavia haproxy keepalived
- Member가 Subnet Amphora Amphora Member Socket(IP, Port)



Octavia 가 . Amphora

. Octavia centos ubuntu

, Amphora 가 ingress (UDP/5555 egress:TCP/9443)

:

1 .

```
$ /opt/rocky/octavia/diskimage-create/diskimage-create.sh -i ubuntu  
  
$ openstack image create amphora-x64-haproxy \  
  --public \  
  --container-format=bare \  
  --disk-format qcow2 \  
  --file /opt/rocky/octavia/diskimage-create/amphora-x64-haproxy.qcow2 \  
  --tag amphora
```

2 .

[controller_worker] amp_image_owner_id, amp_image_tag

```
[controller_worker]  
amp_image_owner_id = 9e4fe13a6d7645269dc69579c027fde4  
amp_image_tag = amphora  
...
```

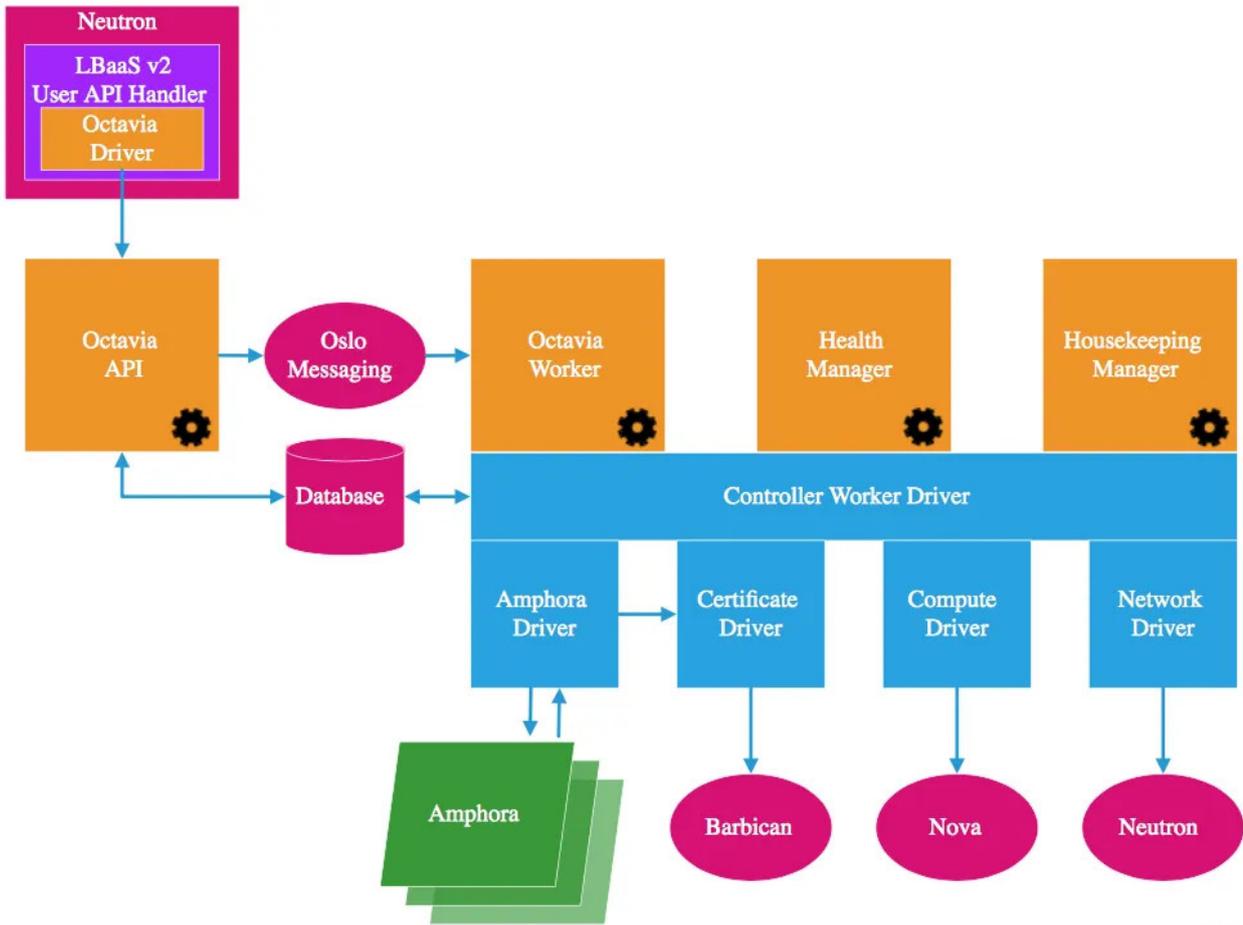
amphora :

1 . amphora

```
$ openstack security group create amphora-sec-grp --project <admin project id>  
$ openstack security group rule create --remote-ip "0.0.0.0/0" --dst-port 9443 --protocol tcp --ingress --ethertype IPv4 --project <admin project id>  
amphora-sec-grp  
$ openstack security group rule create --remote-ip "0.0.0.0/0" --dst-port 5555 --protocol udp --egress --ethertype IPv4 --project <admin project id>  
amphora-sec-grp
```

2 . amphora

```
[controller_worker]  
amp_secgroup_list = <amphora-sec-grp id>  
...
```



<https://blog.@51CTO博客>

(: Octavia 가 .)

Octavia “ - ” . API

MessageQueens .

- Octavia API : RESTful API, Octavia v2 API() LBaaS v2 API OS Neutron Octavia

Driver .

- Octavia Controller Worker : Octavia Driver & Plugin OS 가 .

- Octavia Worker : API Octavia .
- Health Manager : 가 .
- Housekeeping Manager : . SpaceAmphora, DatabaseCleanup CertRotation .

: LB Amphora Octavia
 lbaas LB (: F5) Octavia openstack/neutron-

- Octavia API
- Octavia Worker
- Octavia Health Manager
- Octavia Housekeeping

```
[root@control01 octavia]# tree -L 1 -C
├── amphorae
├── api
├── certificates
├── cmd
├── common
├── compute
├── controller
├── db
├── distributor
├── hacking
├── i18n.py
├── __init__.py
├── network
├── opts.py
├── policies
├── tests
└── version.py
```

@51CTO博客

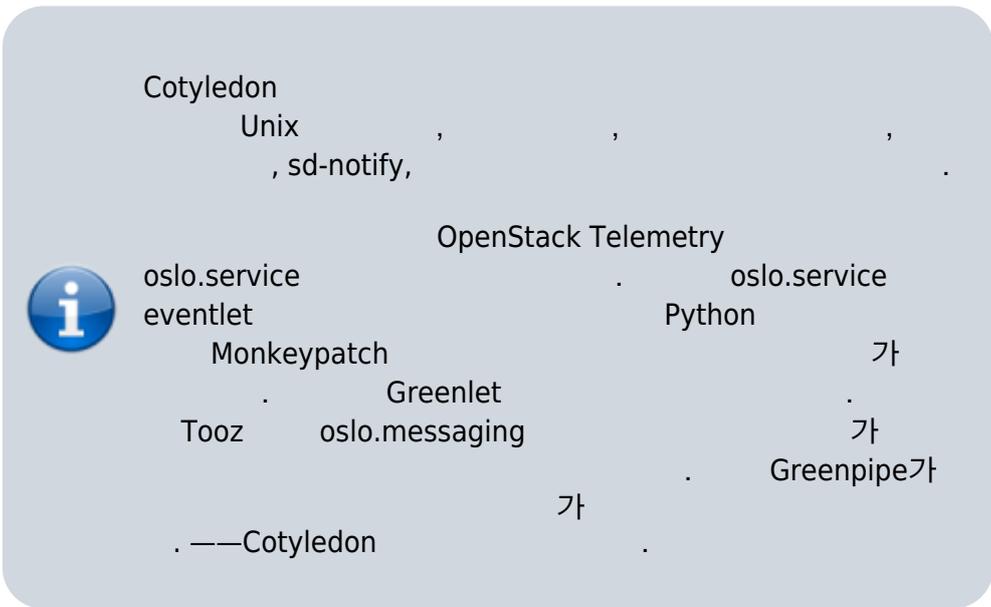
가

- amphora : amphora Rest API amphora-agent
- api : Octavia API
- certificates : CA , amphora Octavia Worker HTTPS TLS
- Compute : Compute Driver novaclient
- network : neutronclient
- db : ORM
- policies : API

```
[root@control01 octavia]# tree controller/ -L 2 -C
controller/
├── healthmanager
│   ├── health_drivers
│   ├── health_manager.py
│   ├── __init__.py
│   └── update_serializer.py
├── housekeeping
│   ├── house_keeping.py
│   ├── __init__.py
│   └── __init__.py
├── queue
│   ├── consumer.py
│   ├── endpoint.py
│   ├── event_queue.py
│   └── __init__.py
└── worker
    ├── amphora_rate_limit.py
    ├── controller_worker.py
    ├── flows
    ├── __init__.py
    ├── tasks
    └── task_utils.py
```

<https://blog.csdn.net/Jmilk>

- healthmanager : Health Manager
 - housekeeping : HouseKeeping
 - queue : cotyledon oslo_messaging RPC
 - producer:api/handlers/queue/producer.py
 - consumer:controller/queue/consumer.py
 - worker : Octavia Worker
 - flow :
 - task :
- : cotyledon oslo.service



OpenStack Octavia
 Driver LB Provider, Certificates Driver, Compute Driver Network
 Driver Vendor
 Octavia OpenStack
 가

?

LoadBalancer

가 Octavia

UML

Octavia

CLI:

```
$ openstack loadbalancer create --vip-subnet-id lb-vip-subnet --name lb1
```

API:

```
POST /v2.0/lbaas/loadbalancers
```

:

```
{
  "loadbalancer": {
    "vip_subnet_id": "c55e7725-894c-400e-bd00-57a04ae1e676",
    "name": "lb1",
    "admin_state_up": true
  }
}
```

:

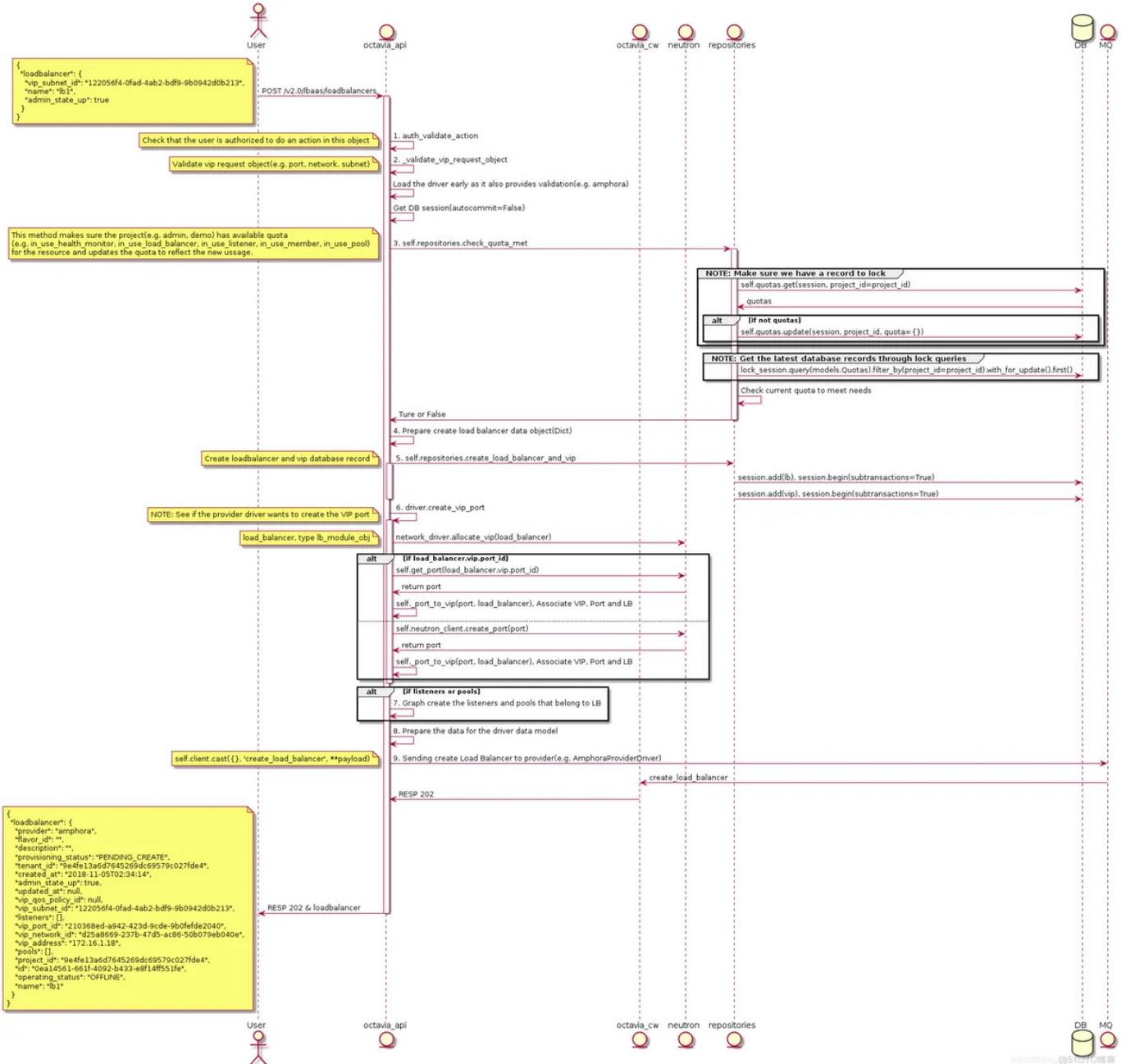
```
{
  "loadbalancer": {
    "provider": "octavia",
    "flavor_id": "",
    "description": "",
    "provisioning_status": "PENDING_CREATE",
    "created_at": "2018-10-22T02:52:04",
    "admin_state_up": true,
    "updated_at": null,
    "vip_subnet_id": "c55e7725-894c-400e-bd00-57a04ae1e676",
    "listeners": [],
    "vip_port_id": "6629fef4-fe14-4b41-9b73-8230105b2e36",
    "vip_network_id": "1078e169-61cb-49bc-a513-915305995be1",
    "vip_address": "10.0.1.7",
    "pools": [],
    "project_id": "2e560efadb704e639ee4bb3953d94afa",
  }
}
```

```

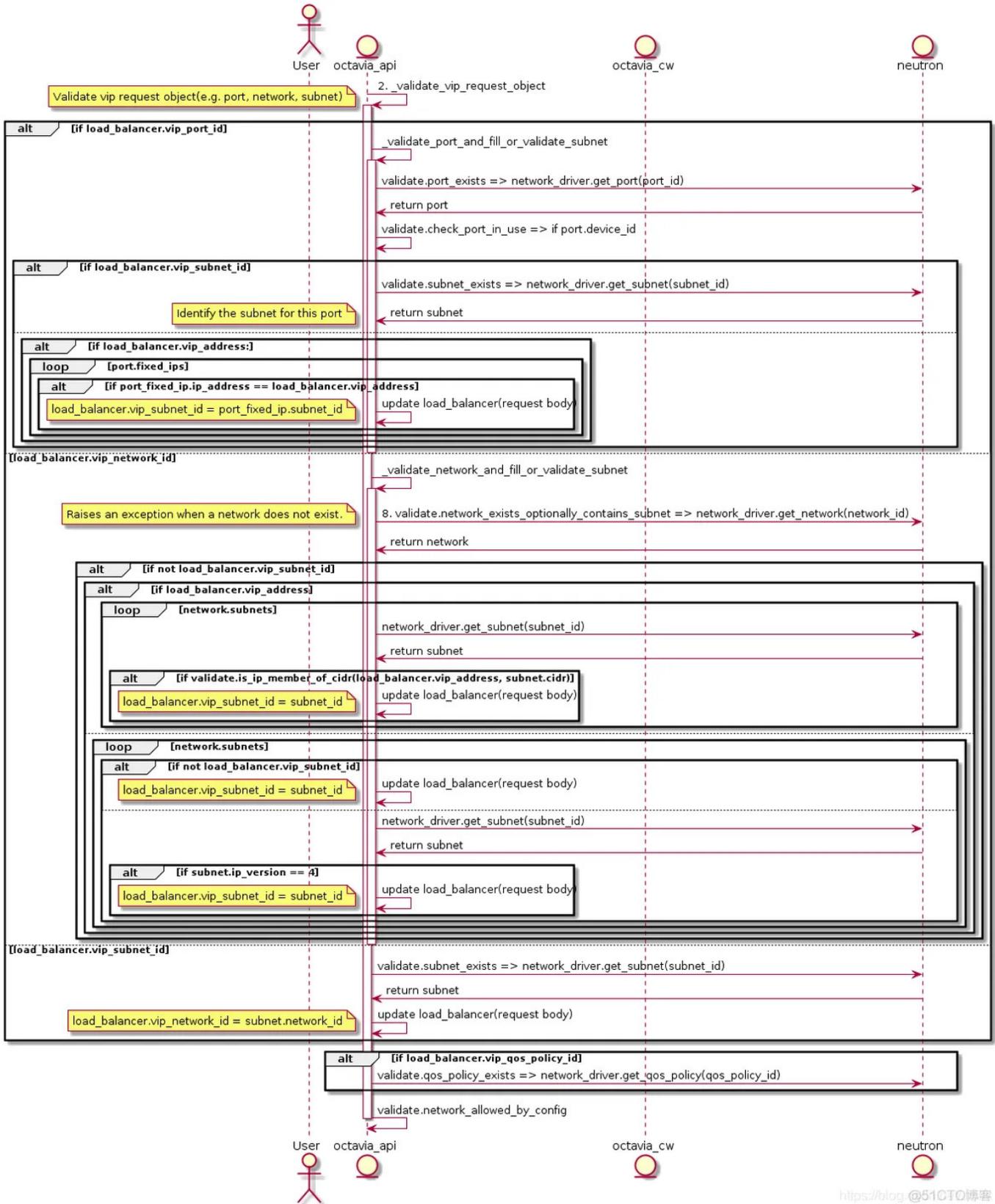
    "id": "5bcf8e3d-9e58-4545-bf80-4c0b905a49ad",
    "operating_status": "OFFLINE",
    "name": "lb1"
  }
}

```

Create LB Octavia API UML



2. _validate_vip_request_object UML



https://blog @51CTO博客

POST /v2.0/lbaas/loadbalancers octavia-api 가 :

1. .
2. VIP (: , ,) . VIP config scition [networking]
3. LB . config section [quotas]


```

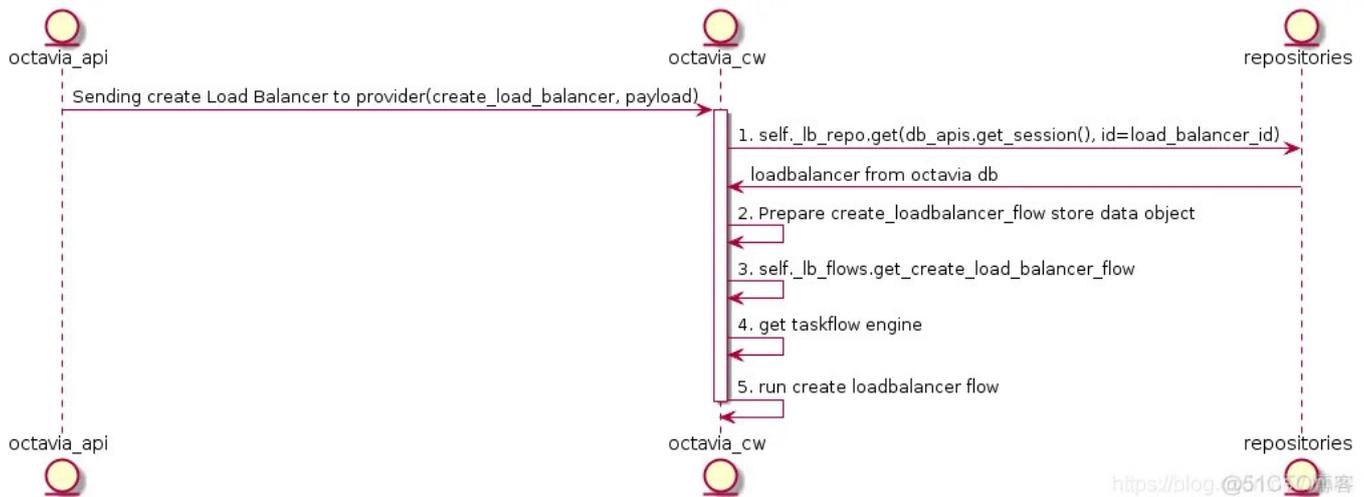
    ( : Project1
    3
    ).
4. load_balancer vip
5. Amphora ( lb ) VIP Port, VIP
    LB
6.
7. create_loadbalancer_flow
8. octavia-worker create_loadbalancer_flow

```

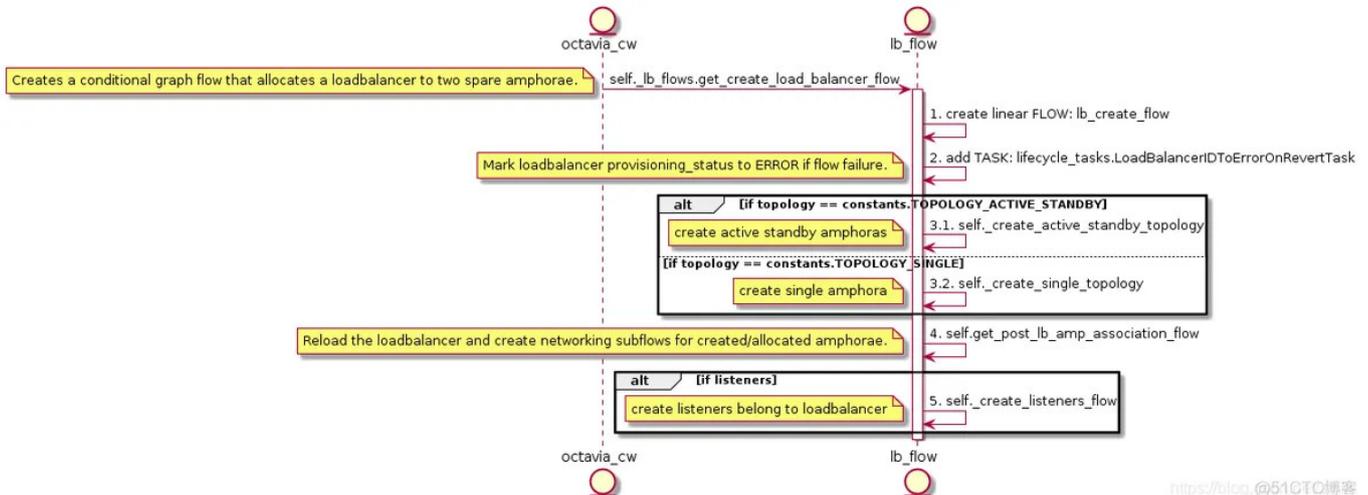
가

- openstack quota set.
- openstack loadbalancer create --listeners --pools , POST /v2.0/lbaas/loadbalancers UI/UX
- VIP 가 octavia-api neutronclient loadbalancer-<load_balancer_id> vip-net
- VIP , , VIP QoS

Create LB Octavia Controller Worker UML



3. get_create_load_balancer_flow UML



가

- amphora(e)

amphorae 가
SINGLE ACTIVE_STANDBY 가
SINGLE 가
ACTIVE_STANDBY Keepalived / -
SINGLE
Amphora UML :



<https://blog.@51CTO博客>

가

- Nova 가 ACTIVE_STANDBY [nova] enable_anti_affinity = True 가
- amphora pool . amphora for lb flow space space
 amphora pool 가 ,
 . space amphora pool Housekeeping Manager
 space amphora pool Housekeeping Manager
 [house_keeping] spare_amphora_pool_size=2 pool size
- amphora for lb flow ,
 (amp_for_lb_flow.link)

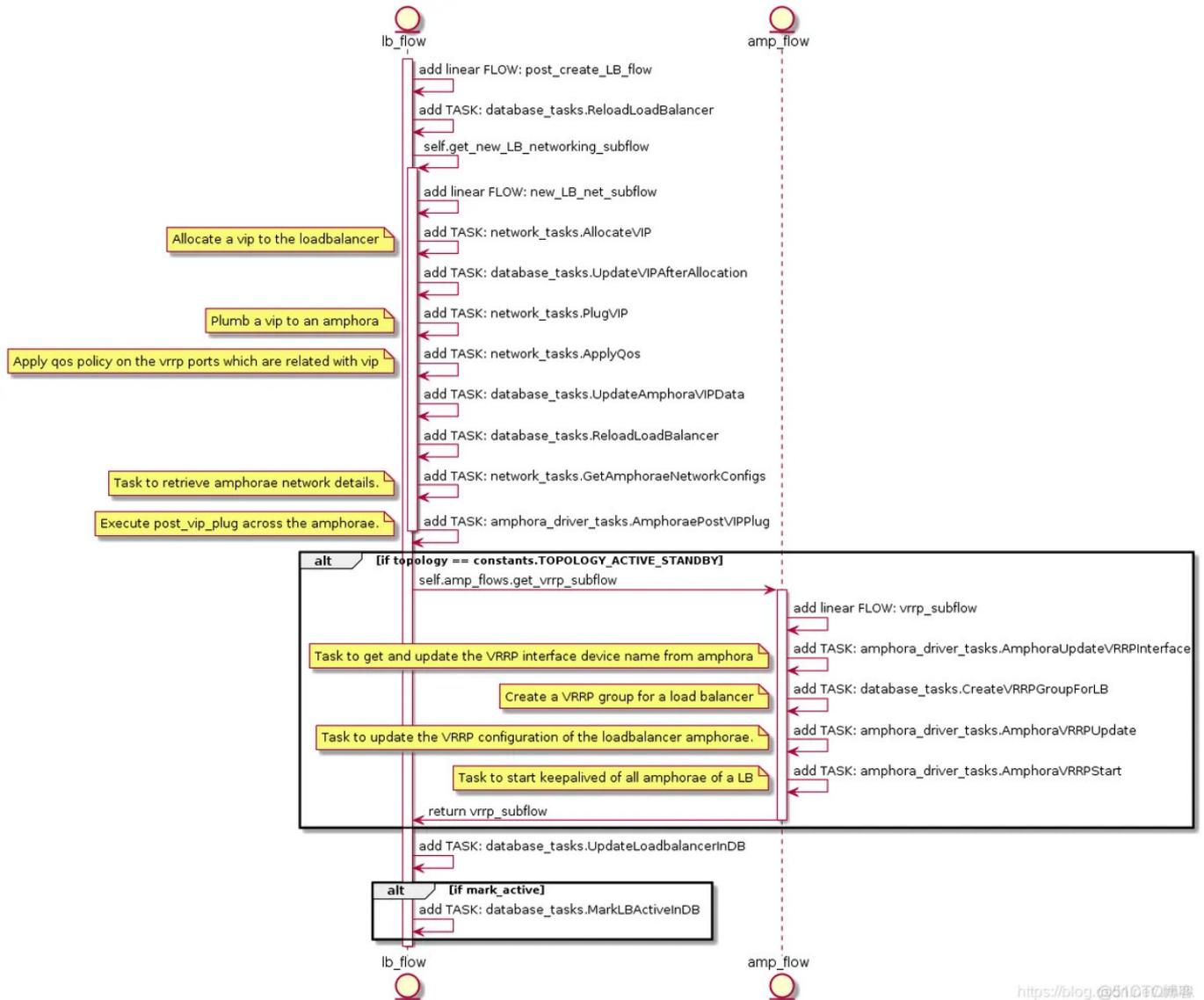
```

if loadbalancer mapping Amphora instance SUCCESS:
    Upload database associations for loadbalancer and amphora
else:
    Create amphora first
    Upload database associations for loadbalancer and amphora
  
```

```

    amphora가 lb-mgmt-net 가 loadbalancer
vip-net amphora octavia-api vip-net
    port:loadbalancer-<load_balancer_id>가
ACTIVE_STANDBY Keepalived VIP
    vip-net VRRP_port (octavia-lb-vrrp-<amphora_id>)가

amphora(e) UML :
  
```



<https://blog.@51070의博客>

Amphora

가

- network_tasks.AllocateVIP
- network_tasks.PlugVIP
- amphora_driver_tasks.AmphoraePostVIPPlug
- amphora_driver_tasks.AmphoraVRRPUpdate
- amphora_driver_tasks.AmphoraVRRPStart

Octavia Networking

network_tasks.AllocateVIP

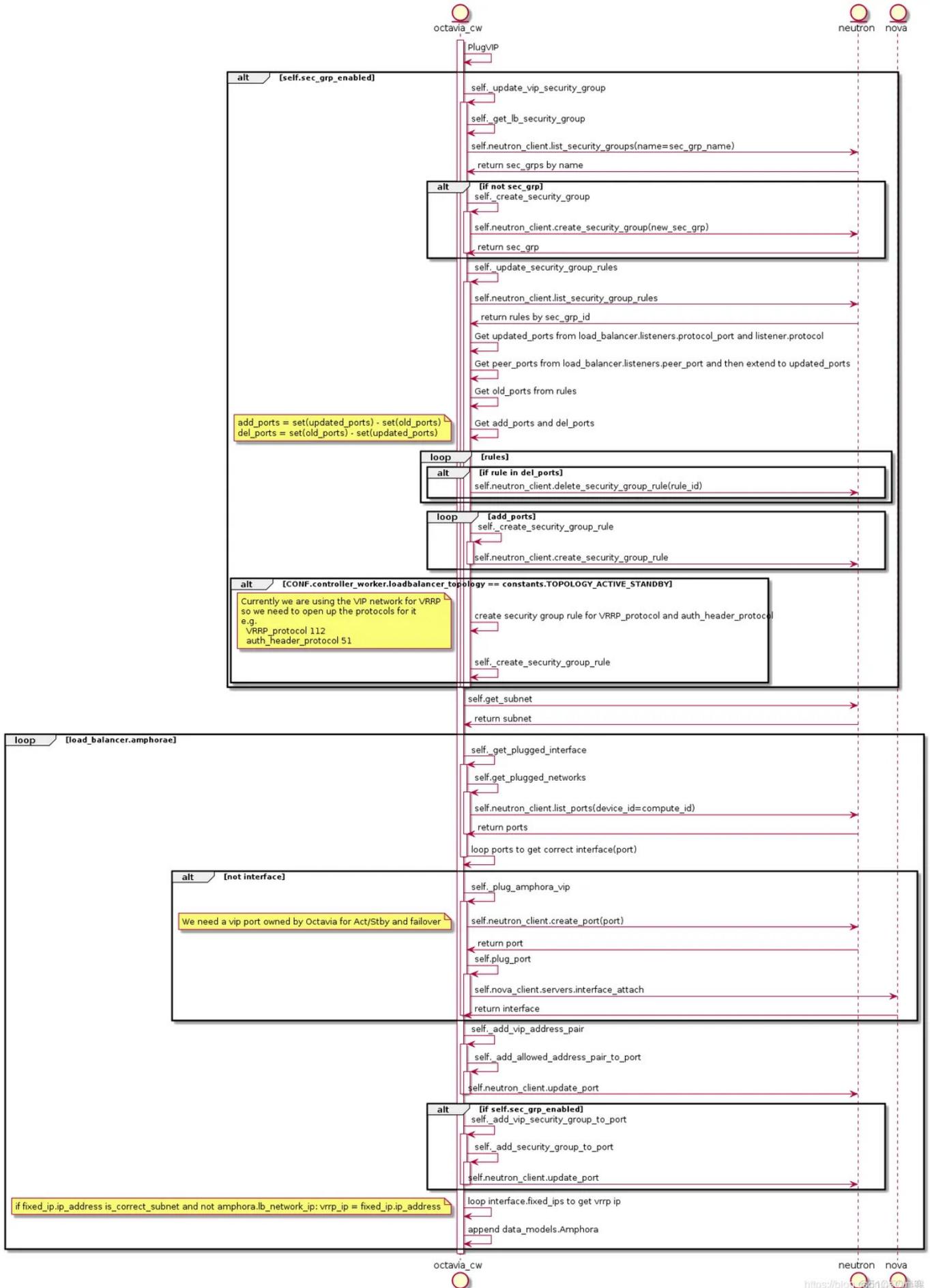
AllocateVIP VIP 가 Port, VIP LB
 data_models.Vip Neutron
 AllowedAddressPairsDriver.allocate_vip octavia-worker octavia-api
 , data_models.Vip Task:UpdateAmphoraVIPData

network_tasks.PlugVIP

AllocateVIP Neutron VIP PlugVIP Amphora VIP

.

PlugVIP UML



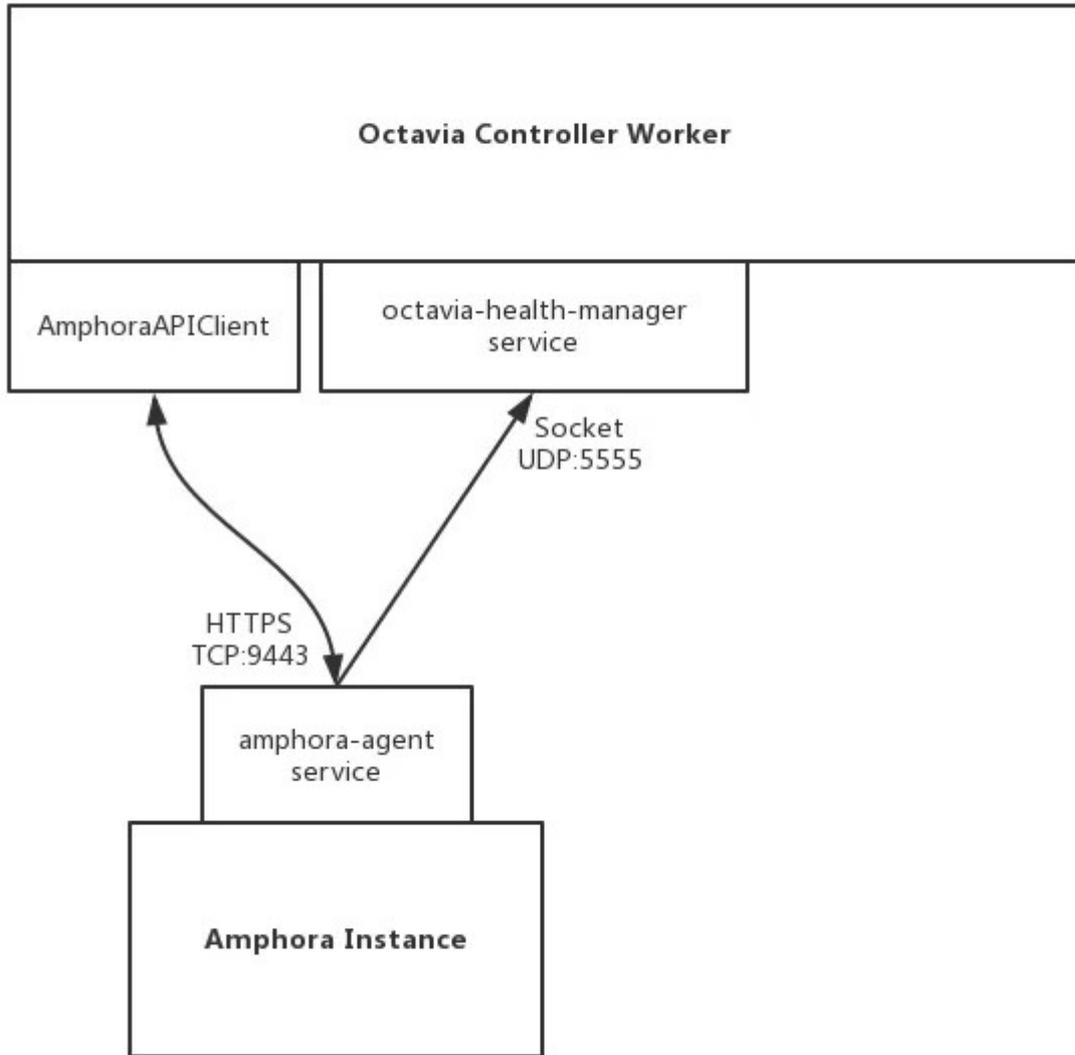
PlugVIP

가

1. VIP security_group_rules . VIP
VIP HTTP:8080 가 VIP HTTP:8080 .
2. , ,
Neutron API Nova API
TASK:AllocateVIP TASK:PlugVIP create lb flow Amphora
Amphora , Amphora 가
Amphora Controller Worker Amphora Agent
AmphoraAPIClient 가 가 가

Amphora

Amphora HAProxy Keepalived , 가
Amphora 가'
가 ? 가 ? 가 가 !
amphora-agent Octavia Controller Worker



<https://blog.@51CTO博客>

, amphora-agent가 AmphoraAPIClient

Amphora Agent

```

amphora-agent Launch Amphora
WSGI HTTP Flask & gunicorn
from octavia.cmd.agent import main 가

```

```

# file:
/opt/rocky/octavia/octavia/amphorae/backends/agent/api_server/server.py

class Server(object):
    def __init__(self):
        self.app = flask.Flask(__name__)
        ...
        self.app.add_url_rule(rule=PATH_PREFIX +
'/listeners/<amphora_id>/<listener_id>/haproxy',

```

```
view_func=self.upload_haproxy_config,
methods=['PUT'])
...

```

Flask, gunicorn, Octavia HAProxy Amphora API, amphora-agent API, route_url

AmphoraAPIClient

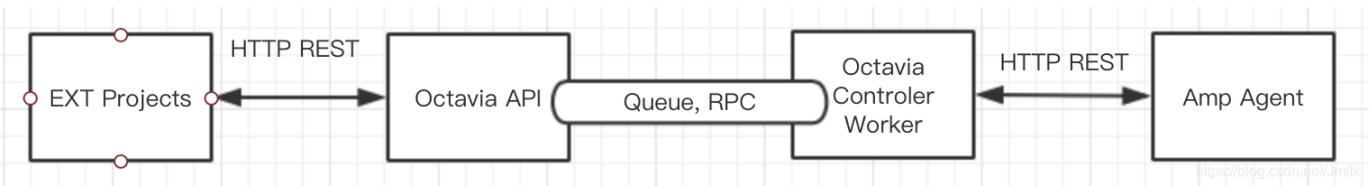
AmphoraAPIClient amphora-agent REST API, Octavia HAProxy Amphora API, URL

```
# file:
/opt/rocky/octavia/octavia/amphorae/drivers/haproxy/rest_api_driver.py

class AmphoraAPIClient(object):
    def __init__(self):
        super(AmphoraAPIClient, self).__init__()
        self.secure = False
        ...

```

- Octavia API: REST API
- Queue: RPC
- Amphora agent: Amphora Octavia Controller Worker REST API



AmphoraePostVIPPlug

TASK: AmphoraePostVIPPlug 가 , AmphoraePostVIPPlug Amphorae
 AmphoraAPIClient PUT plug/vip/{vip} amphora-agent
 VM NIC 가
 Amphora AmphoraePostVIPPlug Amphora
 lb-mgmt-net NIC가
 AmphoraePostVIPPlug VIP NIC vip-net
 Plug:plug_vip ,

Amphora lb-mgmt-net 가

```

root@amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77:~# ifconfig
ens3      Link encap:Ethernet  HWaddr fa:16:3e:b6:8f:a5
          inet addr:192.168.0.9  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:feb6:8fa5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:19462 errors:14099 dropped:0 overruns:0 frame:14099
          TX packets:70317 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1350041 (1.3 MB)  TX bytes:15533572 (15.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

Amphora가 vrrp_port 가 가 .vrrp_port
Keepalived 가 NIC (eth1)

```

root@amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77:~# ip netns exec amphora-
haproxy ifconfig
eth1      Link encap:Ethernet  HWaddr fa:16:3e:f4:69:4b
          inet addr:172.16.1.3  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fef4:694b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:12705 errors:0 dropped:0 overruns:0 frame:0
          TX packets:613211 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:762300 (762.3 KB)  TX bytes:36792968 (36.7 MB)

eth1:0    Link encap:Ethernet  HWaddr fa:16:3e:f4:69:4b
          inet addr:172.16.1.10  Bcast:172.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1

```

VRRP IP: 172.16.1.3 VIP: 172.16.1.10 lb-vip-network DHCP
octavia-lb-vrrp-<amphora_uuid> octavia-lb-<loadbalancer_uuid>
eth1

```

root@amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77:~# ip netns exec amphora-
haproxy cat /etc/network/interfaces.d/eth1

```

```
auto eth1
iface eth1 inet dhcp
root@amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77:~# ip netns exec amphora-
haproxy cat /etc/network/interfaces.d/eth1.cfg

# Generated by Octavia agent
auto eth1 eth1:0
iface eth1 inet static
address 172.16.1.3
broadcast 172.16.1.255
netmask 255.255.255.0
gateway 172.16.1.1
mtu 1450

iface eth1:0 inet static
address 172.16.1.10
broadcast 172.16.1.255
netmask 255.255.255.0
# Add a source routing table to allow members to access the VIP
post-up /sbin/ip route add 172.16.1.0/24 dev eth1 src 172.16.1.10 scope link
table 1
post-up /sbin/ip route add default via 172.16.1.1 dev eth1 onlink table 1
post-down /sbin/ip route del default via 172.16.1.1 dev eth1 onlink table 1
post-down /sbin/ip route del 172.16.1.0/24 dev eth1 src 172.16.1.10 scope
link table 1
post-up /sbin/ip rule add from 172.16.1.10/32 table 1 priority 100
post-down /sbin/ip rule del from 172.16.1.10/32 table 1 priority 100
post-up /sbin/iptables -t nat -A POSTROUTING -p udp -o eth1 -j MASQUERADE
post-down /sbin/iptables -t nat -D POSTROUTING -p udp -o eth1 -j MASQUERADE
```

Keepalived

```
가 loadbalancer_topology = ACTIVE_STANDBY Keepalived
가 , TASK:AmphoraVRRPUpdate TASK:AmphoraVRRPStart
Keepalived Keepalived .

TASK:AmphoraVRRPUpdate , amphora topology VIP port,
VRRP_ports keepalived.conf Jinja ,
AmphoraAPIClient amphora-agent PUT vrrp/upload Keepalived

TASK:AmphoraVRRPStart AmphoraAPIClient PUT vrrp/start
amphora-agent view_func:manage_service_vrrp(action=start) .
```

```
# file:
/opt/rocky/octavia/octavia/amphorae/backends/agent/api_server/keepalived.py

def manager_keepalived_service(self, action):
```

```

    action = action.lower()
    if action not in [consts.AMP_ACTION_START,
                     consts.AMP_ACTION_STOP,
                     consts.AMP_ACTION_RELOAD]:
        return webob.Response(json=dict(
            message='Invalid Request',
            details="Unknown action: {0}".format(action)), status=400)

    if action == consts.AMP_ACTION_START:
        keepalived_pid_path = util.keepalived_pid_path()
        try:
            # Is there a pid file for keepalived?
            with open(keepalived_pid_path, 'r') as pid_file:
                pid = int(pid_file.readline())
                os.kill(pid, 0)

            # If we got here, it means the keepalived process is
running.

            # We should reload it instead of trying to start it again.
            action = consts.AMP_ACTION_RELOAD
        except (IOError, OSError):
            pass

    cmd = ("/usr/sbin/service octavia-keepalived {action}".format(
        action=action))

    try:
        subprocess.check_output(cmd.split(), stderr=subprocess.STDOUT)
    except subprocess.CalledProcessError as e:
        LOG.debug('Failed to %s octavia-keepalived service: %s %s',
                 action, e, e.output)
        return webob.Response(json=dict(
            message="Failed to {0} octavia-keepalived service".format(
                action), details=e.output), status=500)

    return webob.Response(
        json=dict(message='OK',
                 details='keepalived
{action}ed'.format(action=action)),
        status=202)

```

```

    amphora-agent /usr/sbin/service octavia-keepalived start
keepalived . octavia-keepalived.service :

```

```
# file: /usr/lib/systemd/system/octavia-keepalived.service
```

```

[Unit]
Description=Keepalive Daemon (LVS and VRRP)
After=network-online.target .service

```

```
Wants=network-online.target
Requires=.service

[Service]
# Force context as we start keepalived under "ip netns exec"
SELinuxContext=system_u:system_r:keepalived_t:s0
Type=forking
KillMode=process

ExecStart=/sbin/ip netns exec amphora-haproxy /usr/sbin/keepalived -D -d -f
/var/lib/octavia/vrrp/octavia-keepalived.conf -p
/var/lib/octavia/vrrp/octavia-keepalived.pid

ExecReload=/bin/kill -HUP $MAINPID
PIDFile=/var/lib/octavia/vrrp/octavia-keepalived.pid

[Install]
WantedBy=multi-user.target
```

- keepalived namespace amphora-haproxy .
- keepalived /var/lib/octavia/vrrp/octavia-keepalived.conf .

view_func:manage_service_vrrp ,
keepalived view_func:upload_keepalived_config

keepalived 가 .

```
# file: /var/lib/octavia/vrrp/octavia-keepalived.conf

vrrp_script check_script {
    script /var/lib/octavia/vrrp/check_script.sh # VRRP check
    interval 5
    fall 2
    rise 2
}

vrrp_instance 01197be798d5440da846cd70f52dc503 { # VRRP instance name is
loadbalancer UUID
    state MASTER # Master router
    interface eth1 # VRRP IP device
    virtual_router_id 1 # VRID
    priority 100
    nopreempt
    garp_master_refresh 5
    garp_master_refresh_repeat 2
    advert_int 1
```

```

authentication {
  auth_type PASS
  auth_pass b76d77e
}

unicast_src_ip 172.16.1.3          # VRRP IP
unicast_peer {
  172.16.1.7                       # Backup router VRRP IP
}

virtual_ipaddress {
  172.16.1.10                      # VIP address
}
track_script {
  check_script
}
}

```

```

, keepalived eth1 VRRP IP VIP , eth1
TASK:AmphoraePostVIPPlug namespace amphora .
check_script.sh VIP Amphorae HAProxy

```

```

root@amphora-caa6ba0f-1a68-4f22-9be9-8521695ac4f4:~# cat
/var/lib/octavia/vrrp/check_scripts/haproxy_check_script.sh
haproxy-vrrp-check /var/lib/octavia/d367b5ec-24dd-44b3-b947-
e0ff72c75e66.sock; exit $?

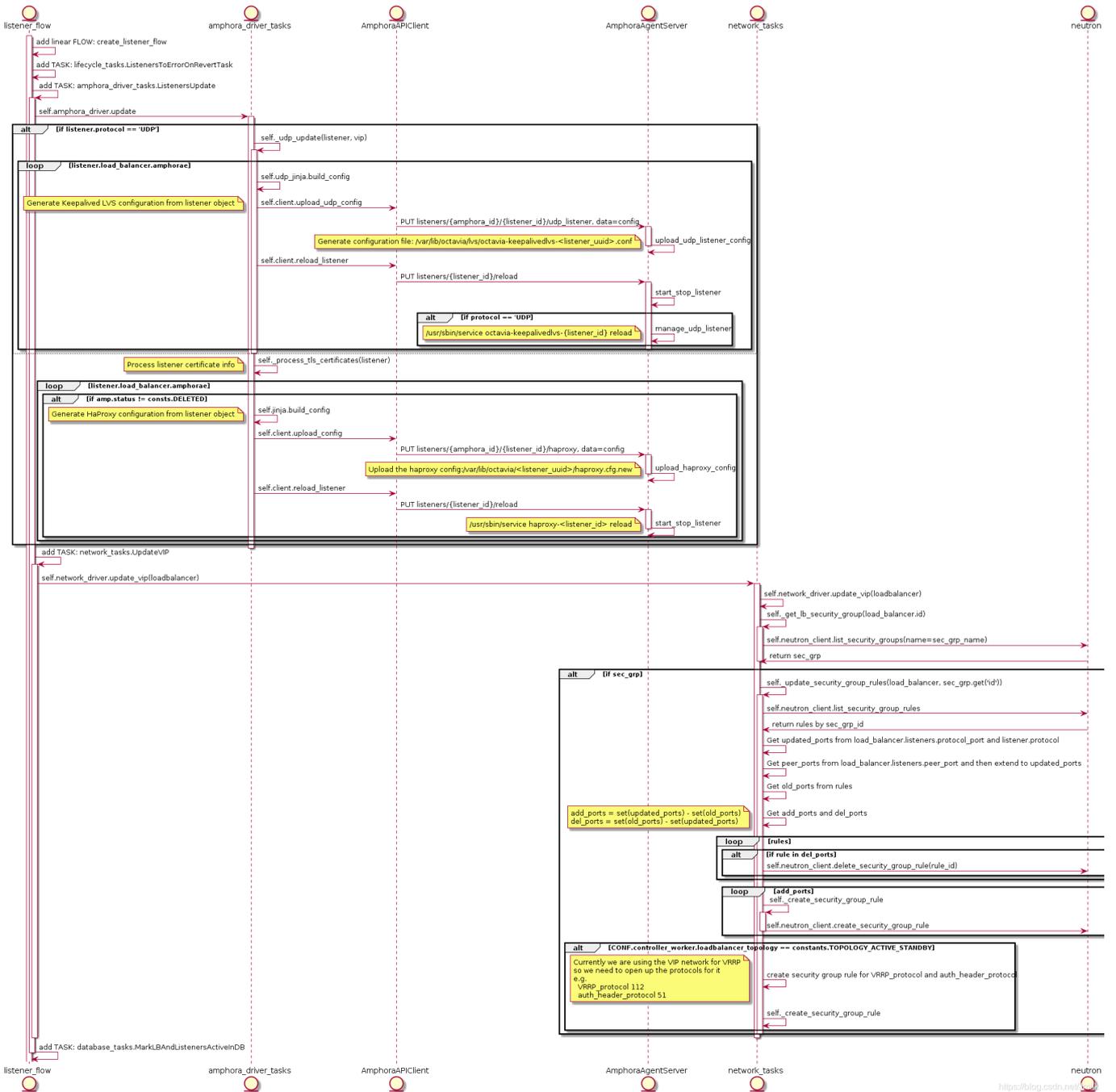
```

```

Amphora Instance amphora-agent keepalived 가
haproxy . haproxy 가
, amphorae amphorae vip-net

```

UML



, openstack loadbalancer listener create --protocol HTTP
 --protocol-port 8080 lb-1 Task:ListenersUpdate
 , AmphoraAPIClient가 :

- PUT listeners/{amphora_id}/{listener_id}/haproxy: haproxy
- PUT listeners/{listener_id}/reload: haproxy

가 haproxy 가 Listener VIP Task:UpdateVIP .

haproxy

amphora haproxy .


```
# file: /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557/haproxy.cfg,
Listener UUID: 1385d3c4-615e-4a92-aea1-c4fa51a75557

# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aea1c4fa51a75557_peers
    peer l_Ustq0qE-h-_Q1dLXLXBaiWR8U 172.16.1.7:1025
    peer 008zAgUhIv9TEXhyYZf2iHdx0kA 172.16.1.3:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
    maxconn 1000000
    bind 172.16.1.10:8080
    mode http
    timeout client 50000
```

```
HTTP 8080 fronted section bind
172.16.1.10:8080 mode http .
```

```
Amphora haproxy haproxy-1385d3c4-615e-4a92-aea1-
c4fa51a75557.service(ListenerUUID:1385d3c4-615e-4a92-aea1-c4fa51a75557)
service
```

```
# file: /usr/lib/systemd/system/haproxy-1385d3c4-615e-4a92-aea1-
c4fa51a75557.service
```

```
[Unit]
```

```
Description=HAProxy Load Balancer
```

```
After=network.target syslog.service amphora-netns.service
```

```
Before=octavia-keepalived.service
```

```
Wants=syslog.service
```

```
Requires=amphora-netns.service
```

```
[Service]
```

```
# Force context as we start haproxy under "ip netns exec"
```

```
SELinuxContext=system_u:system_r:haproxy_t:s0

Environment="CONFIG=/var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557/haproxy.cfg" "USERCONFIG=/var/lib/octavia/haproxy-default-user-group.conf" "PIDFILE=/var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557/1385d3c4-615e-4a92-aea1-c4fa51a75557.pid"

ExecStartPre=/usr/sbin/haproxy -f $CONFIG -f $USERCONFIG -c -q -L 008zAgUhIv9TEXhyYZf2iHdx0kA

ExecReload=/usr/sbin/haproxy -c -f $CONFIG -f $USERCONFIG -L 008zAgUhIv9TEXhyYZf2iHdx0kA
ExecReload=/bin/kill -USR2 $MAINPID

ExecStart=/sbin/ip netns exec amphora-haproxy /usr/sbin/haproxy-systemd-wrapper -f $CONFIG -f $USERCONFIG -p $PIDFILE -L 008zAgUhIv9TEXhyYZf2iHdx0kA

KillMode=mixed
Restart=always
LimitNOFILE=2097152

[Install]
WantedBy=multi-user.target
```

가 /usr/sbin/haproxy-systemd-wrapper , namespace amphora-haproxy , /usr/sbin/haproxy

```
Nov 15 10:12:01 amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77 ip[13206]: haproxy-systemd-wrapper: executing /usr/sbin/haproxy -f /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557/haproxy.cfg -f /var/lib/octavia/haproxy-default-user-group.conf -p /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557/1385d3c4-615e-4a92-aea1-c4fa51a75557.pid -L 008zAgUhIv9TEXhyYZf2iHdx0kA -Ds
```

, ,L7 ,L7 , Health Monitor haproxy



UML

<https://blog.csdn.net/Jrmilk>

create pool flow 가 haproxy
 Task:ListenersUpdate .

```

openstack loadbalancer pool create --protocol HTTP --lb-algorithm ROUND_ROBIN
--listener 1385d3c4-615e-4a92-aea1-c4fa51a75557
    default pool haproxy.cfg backend section 가
    backend mode http balance roundrobin .
  
```

```

# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aea1c4fa51a75557_peers
    peer l_Ustq0qE-h-_Q1dLXLXBAiWR8U 172.16.1.7:1025
    peer 008zAgUhIv9TEXhyYZf2iHdx0kA 172.16.1.3:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
  
```

```

maxconn 1000000
bind 172.16.1.10:8080
mode http
default_backend 8196f752-a367-4fb4-9194-37c7eab95714 # UUID of
pool
  timeout client 50000

backend 8196f752-a367-4fb4-9194-37c7eab95714
  mode http
  balance roundrobin
  fullconn 1000000
  option allbackups
  timeout connect 5000
  timeout server 50000

```

```

      listener uuid      loadbalancer uuid
      default pool      default pool 가      default
pool      가      loadbalancer uuid      shared pool
      가      . shared pool      가
      l7policy      .      l7policy
“      ”      ,
      .

```

```

$ openstack loadbalancer pool create --protocol HTTP --lb-algorithm
ROUND_ROBIN --loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503

```

Field	Value
admin_state_up	True
created_at	2018-11-20T03:35:08
description	
healthmonitor_id	
id	822f78c3-ea2c-4770-bef0-e97f1ac2eba8
lb_algorithm	ROUND_ROBIN
listeners	
loadbalancers	01197be7-98d5-440d-a846-cd70f52dc503
members	
name	
operating_status	OFFLINE
project_id	9e4fe13a6d7645269dc69579c027fde4
protocol	HTTP
provisioning_status	PENDING_CREATE
session_persistence	None
updated_at	None

haproxy.cfg

가

CalculateDelta

TASK:CalculateDelta Amphora Amphora NIC
NIC “ ” Task:CalculateAmphoraDelta .“

```
# file: /opt/rocky/octavia/octavia/controller/worker/tasks/network_tasks.py

class CalculateAmphoraDelta(BaseNetworkTask):

    default_provides = constants.DELTA

    def execute(self, loadbalancer, amphora):
        LOG.debug("Calculating network delta for amphora id: %s",
        amphora.id)

        # Figure out what networks we want
        # seed with lb network(s)
        vrrp_port = self.network_driver.get_port(amphora.vrrp_port_id)
        desired_network_ids = {vrrp_port.network_id}.union(
            CONF.controller_worker.amp_boot_network_list)

        for pool in loadbalancer.pools:
            member_networks = [
                self.network_driver.get_subnet(member.subnet_id).network_id
                for member in pool.members
                if member.subnet_id
            ]
            desired_network_ids.update(member_networks)

        nics = self.network_driver.get_plugged_networks(amphora.compute_id)
        # assume we don't have two nics in the same network
        actual_network_nics = dict((nic.network_id, nic) for nic in nics)

        del_ids = set(actual_network_nics) - desired_network_ids
        delete_nics = list(
            actual_network_nics[net_id] for net_id in del_ids)

        add_ids = desired_network_ids - set(actual_network_nics)
        add_nics = list(n_data_models.Interface(
            network_id=net_id) for net_id in add_ids)
        delta = n_data_models.Delta(
            amphora_id=amphora.id, compute_id=amphora.compute_id,
            add_nics=add_nics, delete_nics=delete_nics)
        return delta
```

, desired_network_ids

```

actual_network_nics 가 , delete_nics 가 add_nics
Delta data models Task:HandleNetworkDeltas Amphora NIC
가 .

```

HandleNetworkDeltas

Task:HandleNetworkDelta Amphora Delta

```
# file: /opt/rocky/octavia/octavia/controller/worker/tasks/network_tasks.py
```

```

class HandleNetworkDelta(BaseNetworkTask):
    """Task to plug and unplug networks

    Plug or unplug networks based on delta
    """

    def execute(self, amphora, delta):
        """Handle network plugging based off deltas."""
        added_ports = {}
        added_ports[amphora.id] = []
        for nic in delta.add_nics:
            interface = self.network_driver.plug_network(delta.compute_id,
                                                         nic.network_id)

            port = self.network_driver.get_port(interface.port_id)
            port.network = self.network_driver.get_network(port.network_id)
            for fixed_ip in port.fixed_ips:
                fixed_ip.subnet = self.network_driver.get_subnet(
                    fixed_ip.subnet_id)
            added_ports[amphora.id].append(port)
        for nic in delta.delete_nics:
            try:
                self.network_driver.unplug_network(delta.compute_id,
                                                    nic.network_id)

            except base.NetworkNotFound:
                LOG.debug("Network %d not found ", nic.network_id)
            except Exception:
                LOG.exception("Unable to unplug network")
        return added_ports

```

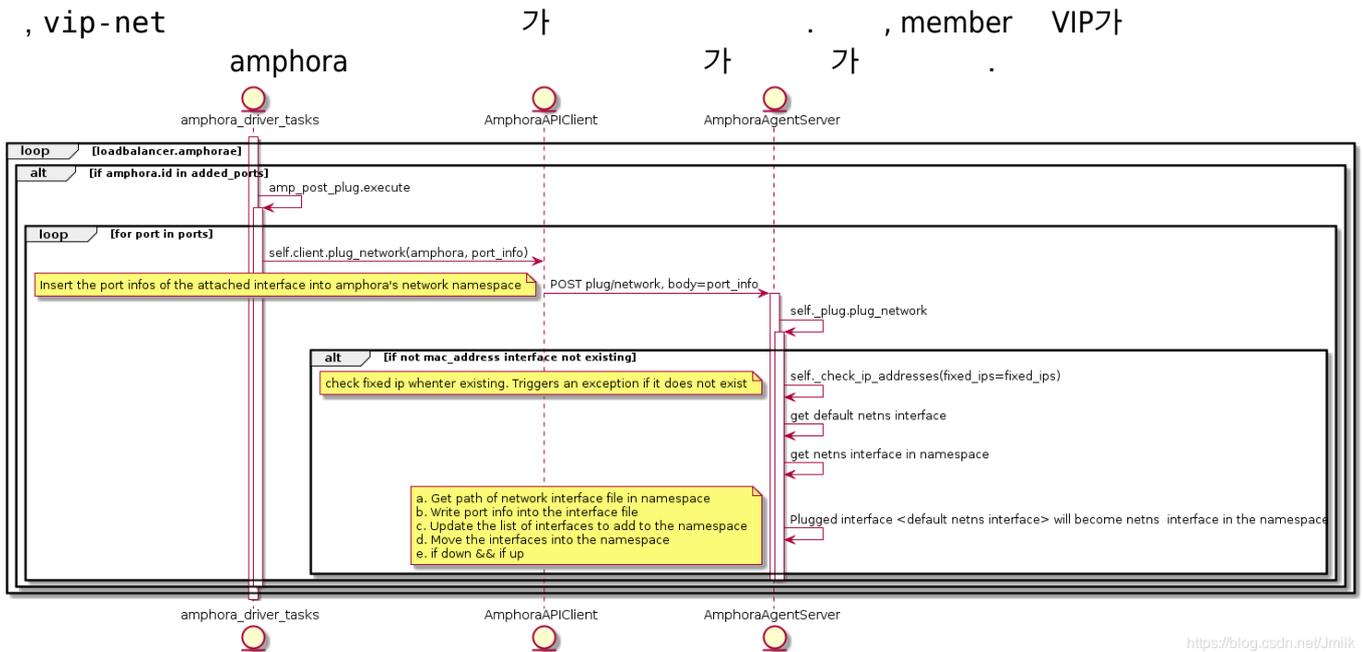
```
, added_port return TASK:AmphoraePostNetworkPlug
```

AmphoraePostNetworkPlug

```

Task: AmphoraePostNetworkPlug member가 port network
namespace . AmphoraePostNetworkPlug
AmphoraePostVIPPlug . create member flow , member
tenant-net 가 . create lb flow

```



Member 가 Amphora

```

root@amphora-cd444019-ce8f-4f89-be6b-0edf76f41b77:~# ip netns exec amphora-haproxy ifconfig
eth1      Link encap:Ethernet  HWaddr fa:16:3e:f4:69:4b
          inet addr:172.16.1.3  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fef4:694b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:12705 errors:0 dropped:0 overruns:0 frame:0
          TX packets:613211 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:762300 (762.3 KB)  TX bytes:36792968 (36.7 MB)

eth1:0    Link encap:Ethernet  HWaddr fa:16:3e:f4:69:4b
          inet addr:172.16.1.10 Bcast:172.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1

eth2      Link encap:Ethernet  HWaddr fa:16:3e:18:23:7a
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe18:237a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:8 errors:2 dropped:0 overruns:0 frame:2
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2156 (2.1 KB)  TX bytes:808 (808.0 B)
    
```

```

# Generated by Octavia agent
auto eth2
    
```



```
iface eth2 inet static
address 192.168.1.3
broadcast 192.168.1.255
netmask 255.255.255.0
mtu 1450
post-up /sbin/iptables -t nat -A POSTROUTING -p udp -o eth2 -j MASQUERADE
post-down /sbin/iptables -t nat -D POSTROUTING -p udp -o eth2 -j MASQUERADE
```

ListenersUpdate

haproxy

Task:ListenersUpdate

```
# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aealc4fa51a75557_peers
    peer l_Ustq0qE-h-_Q1dlXLXBAiWR8U 172.16.1.7:1025
    peer 008zAgUhIv9TEXhyYZf2iHdx0kA 172.16.1.3:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
    maxconn 1000000
    bind 172.16.1.10:8080
    mode http
    default_backend 8196f752-a367-4fb4-9194-37c7eab95714
    timeout client 50000

backend 8196f752-a367-4fb4-9194-37c7eab95714
    mode http
    balance roundrobin
    fullconn 1000000
    option allbackups
    timeout connect 5000
    timeout server 50000
    server b6e464fd-dd1e-4775-90f2-4231444a0bbe 192.168.1.14:80 weight 1
```

, member 가 backend(default pool) server <member_id> 가
192.168.1.14:80 weight 1 가 , 가

L7policy, L7rule Health Monitor

L7policy (: pool , URL)
, L7rule Listener .



L7Rule ,

L7policy

1.

```
$ openstack loadbalancer healthmonitor create --name healthmonitor1 --type PING --delay 5 --timeout 10 --max-retries 3 8196f752-a367-4fb4-9194-37c7eab95714
```

```
$ openstack loadbalancer l7policy create --name l7p1 --action REDIRECT_TO_POOL --redirect-pool 8196f752-a367-4fb4-9194-37c7eab95714 1385d3c4-615e-4a92-aea1-c4fa51a75557
```

```
$ openstack loadbalancer l7rule create --type HOST_NAME --compare-type STARTS_WITH --value "server" 87593985-e02f-4880-b80f-22a4095c05a7
```

haproxy.cfg

```
# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000
    external-check

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aea1c4fa51a75557_peers
    peer l_Ustq0qE-h-_Q1dLXLXBAiWR8U 172.16.1.7:1025
    peer 008zAgUhIv9TEXhyZf2iHdx0kA 172.16.1.3:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
    maxconn 1000000
    # frontend http://172.16.1.10:8080
    bind 172.16.1.10:8080
    mode http
    # ACL
    acl 8d9b8b1e-83d7-44ca-a5b4-0103d5f90cb9 req.hdr(host) -i -m beg

server
    # if ACL 8d9b8b1e-83d7-44ca-a5b4-0103d5f90cb9 backend 8196f752-a367-4fb4-9194-37c7eab95714
    use_backend 8196f752-a367-4fb4-9194-37c7eab95714 if 8d9b8b1e-83d7-44ca-
```

```

a5b4-0103d5f90cb9
#      ACL                      backend 8196f752-a367-4fb4-9194-37c7eab95714

default_backend 8196f752-a367-4fb4-9194-37c7eab95714
timeout client 50000

backend 8196f752-a367-4fb4-9194-37c7eab95714
# http
mode http
# RR
balance roundrobin
timeout check 10s
option external-check
# ping-wrapper.sh                server
external-check command /var/lib/octavia/ping-wrapper.sh
fullconn 1000000
option allbackups
timeout connect 5000
timeout server 50000
#      (real server)[]          80 ,          inter 5s fall 3 rise
3
server b6e464fd-dd1e-4775-90f2-4231444a0bbe 192.168.1.14:80 weight 1
check inter 5s fall 3 rise 3

```

Health Check Script (ping-wrapper.sh) 가 PING .

```

#!/bin/bash
if [[ $HAPROXY_SERVER_ADDR =~ ":" ]]; then
    /bin/ping6 -q -n -w 1 -c 1 $HAPROXY_SERVER_ADDR > /dev/null 2>&1
else
    /bin/ping -q -n -w 1 -c 1 $HAPROXY_SERVER_ADDR > /dev/null 2>&1
fi

```

2.

```

$ openstack loadbalancer healthmonitor create --name healthmonitor1 --type
PING --delay 5 --timeout 10 --max-retries 3 822f78c3-ea2c-4770-bef0-
e97f1ac2eba8

```

```

$ openstack loadbalancer l7policy create --name l7p1 --action
REDIRECT_TO_POOL --redirect-pool 822f78c3-ea2c-4770-bef0-e97f1ac2eba8
1385d3c4-615e-4a92-aea1-c4fa51a75557

```

```

$ openstack loadbalancer l7rule create --type HOST_NAME --compare-type
STARTS_WITH --value "server" fb90a3b5-c97c-4d99-973e-118840a7a236

```

haproxy.cfg

```
# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000
    external-check

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aea1c4fa51a75557_peers
    peer l_Ustq0qE-h-_Q1dLXLXBAiWR8U 172.16.1.7:1025
    peer 008zAgUhIv9TEXhyYZf2iHdx0kA 172.16.1.3:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
    maxconn 1000000
    bind 172.16.1.10:8080
    mode http
        acl 8d9b8b1e-83d7-44ca-a5b4-0103d5f90cb9 req.hdr(host) -i -m beg
server
    use_backend 8196f752-a367-4fb4-9194-37c7eab95714 if 8d9b8b1e-83d7-44ca-a5b4-0103d5f90cb9
        acl c76f36bc-92c0-4f48-8d57-a13e3b1f09e1 req.hdr(host) -i -m beg
server
    use_backend 822f78c3-ea2c-4770-bef0-e97f1ac2eba8 if
c76f36bc-92c0-4f48-8d57-a13e3b1f09e1
    default_backend 8196f752-a367-4fb4-9194-37c7eab95714
    timeout client 50000

backend 8196f752-a367-4fb4-9194-37c7eab95714
    mode http
    balance roundrobin
    timeout check 10s
    option external-check
    external-check command /var/lib/octavia/ping-wrapper.sh
    fullconn 1000000
    option allbackups
    timeout connect 5000
    timeout server 50000
    server b6e464fd-dd1e-4775-90f2-4231444a0bbe 192.168.1.14:80 weight 1
check inter 5s fall 3 rise 3
```

```
backend 822f78c3-ea2c-4770-bef0-e97f1ac2eba8
  mode http
  balance roundrobin
  timeout check 10s
  option external-check
  external-check command /var/lib/octavia/ping-wrapper.sh
  fullconn 1000000
  option allbackups
  timeout connect 5000
  timeout server 50000
  server 7da6f176-36c6-479a-9d86-c892ecca6ae5 192.168.1.6:80 weight 1
  check inter 5s fall 3 rise 3
```

, listener 가 shared pool 822f78c3-ea2c-4770-bef0-e97f1ac2eba8 backend section 가 .

Amphora

CA

SSL

amphora-agent Octavia Controller Worker가
 . Octavia가 CA .

 : CA가 . amphora CA .

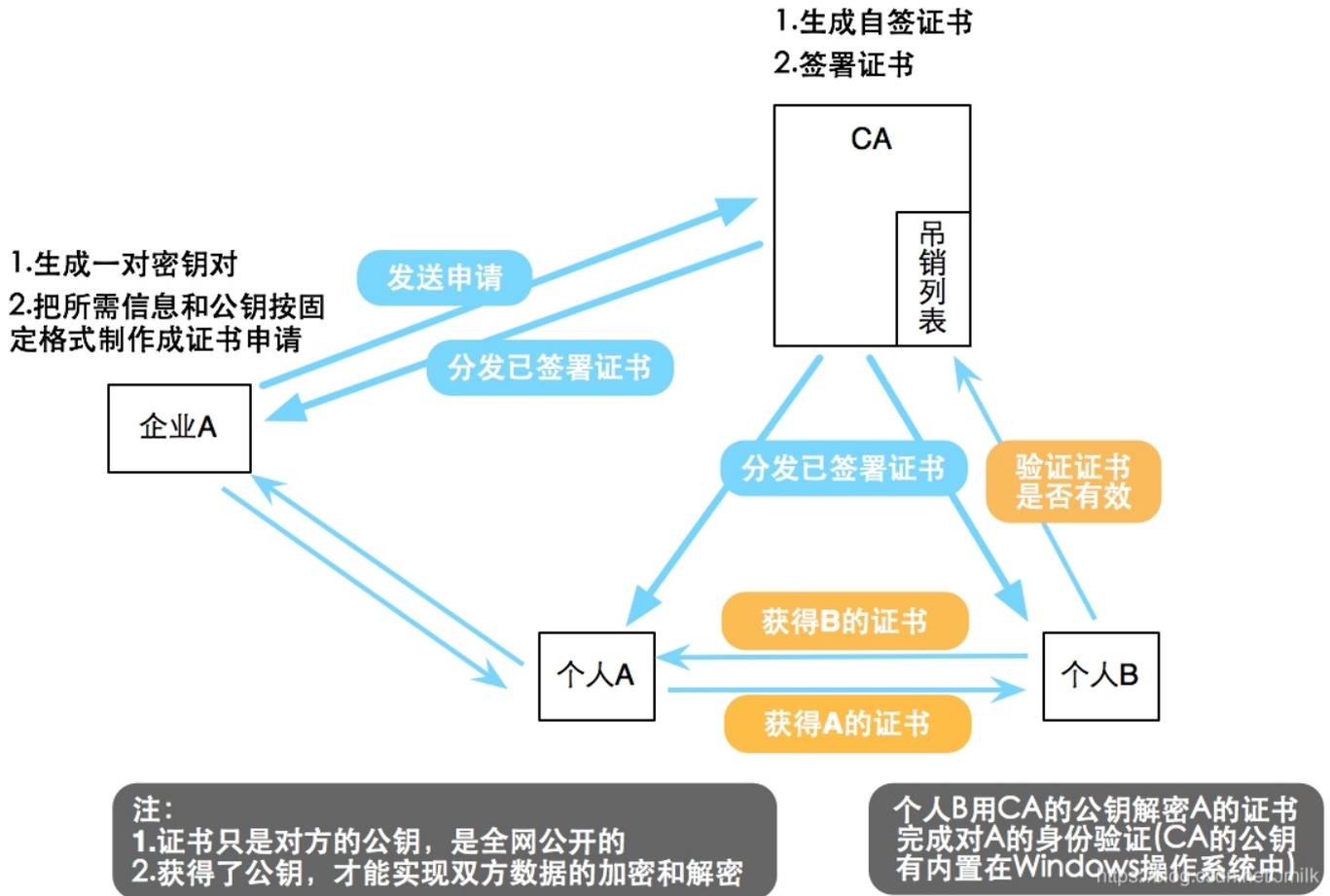
Octavia Dashboard가 OpenStack /API 가 가
 . , Octavia가 CA 가 가
 :

1. amphora-agent , API 가
2. 가 amphora ' ' OpenStack

Octavia OpenSSL CA .

```
$ source /opt/rocky/octavia/bin/create_certificates.sh /etc/octavia/certs/
/opt/rocky/octavia/etc/certificates/openssl.cnf
```

CA , CA 3 ,
 . "OpenSSL
 CA " ,



Octavia가

CA

```
$ ll /etc/octavia/certs/
total 44
-rw-r--r-- 1 stack stack 1294 Oct 26 12:51 ca_01.pem
-rw-r--r-- 1 stack stack 989 Oct 26 12:51 client.csr
-rw-r--r-- 1 stack stack 1708 Oct 26 12:51 client.key
-rw-r--r-- 1 stack stack 4405 Oct 26 12:51 client-.pem
-rw-r--r-- 1 stack stack 6113 Oct 26 12:51 client.pem
-rw-r--r-- 1 stack stack 71 Oct 26 12:51 index.txt
-rw-r--r-- 1 stack stack 21 Oct 26 12:51 index.txt.attr
-rw-r--r-- 1 stack stack 0 Oct 26 12:51 index.txt.old
drwxr-xr-x 2 stack stack 20 Oct 26 12:51 newcerts
drwx----- 2 stack stack 23 Oct 26 12:51 private
-rw-r--r-- 1 stack stack 3 Oct 26 12:51 serial
-rw-r--r-- 1 stack stack 3 Oct 26 12:51 serial.old
```

- newcerts dir: CA ()
- private dir: CA
- serial file: (e.g. 01), 가
- index.txt file:
- ca_01.pem PEM file: CA
- client.csr file: CSR()

- client.key file:
- client-.pem: PEM
- client.pem: client-.pem client.key

CA

```
# create new amphora flow          **TASK:GenerateServerPEMTask**  amphora

[certificates]
ca_private_key_passphrase = foobar
ca_private_key = /etc/octavia/certs/private/cakey.pem
ca_certificate = /etc/octavia/certs/ca_01.pem

# AmphoraAPIClient          , client.pem(          )  CA
(          )  amphora-agent  SSL
[haproxy_amphora]
server_ca = /etc/octavia/certs/ca_01.pem
client_cert = /etc/octavia/certs/client.pem

# Task:CertComputeCreate          , CA
[controller_worker]
client_ca = /etc/octavia/certs/ca_01.pem
```

SSL

1. Amphora CA , amphora-agent
가 Flask HTTPS
2. AmphoraAPIClient가 amphora-agent , CA
SSL

Amphora Agent

amphora

```
# file: /opt/rocky/octavia/octavia/controller/worker/tasks/cert_task.py

class GenerateServerPEMTask(BaseCertTask):
    """Create the server certs for the agent comm

    Use the amphora_id for the CN
    """

    def execute(self, amphora_id):
        cert = self.cert_generator.generate_cert_key_pair(
            cn=amphora_id,
            validity=CERT_VALIDITY)

        return cert.certificate + cert.private_key
```

Octavia Certificates local_cert_generator() anchor_cert_generator 가 , [certificates] cert_generator .

```
# file: /opt/rocky/octavia/octavia/certificates/generator/local.py

@classmethod
def generate_cert_key_pair(cls, cn, validity, bit_length=2048,
                           passphrase=None, **kwargs):
    pk = cls._generate_private_key(bit_length, passphrase)
    csr = cls._generate_csr(cn, pk, passphrase)
    cert = cls.sign_cert(csr, validity, **kwargs)
    cert_object = local_common.LocalCert(
        certificate=cert,
        private_key=pk,
        private_key_passphrase=passphrase
    )
    return cert_object
```

LocalCertGenerator.generate_cert_key_pair

1. Amphora
2. Amphora (CSR)
3. CA Amphora

, create_certificates.sh

Octavia

Certificates가 cryptography

TASK:GenerateServerPEMTask

Amphora

TASK:CertComputeCreate가

Nova userdata Nova Store metadata on a configuration drive

Amphora

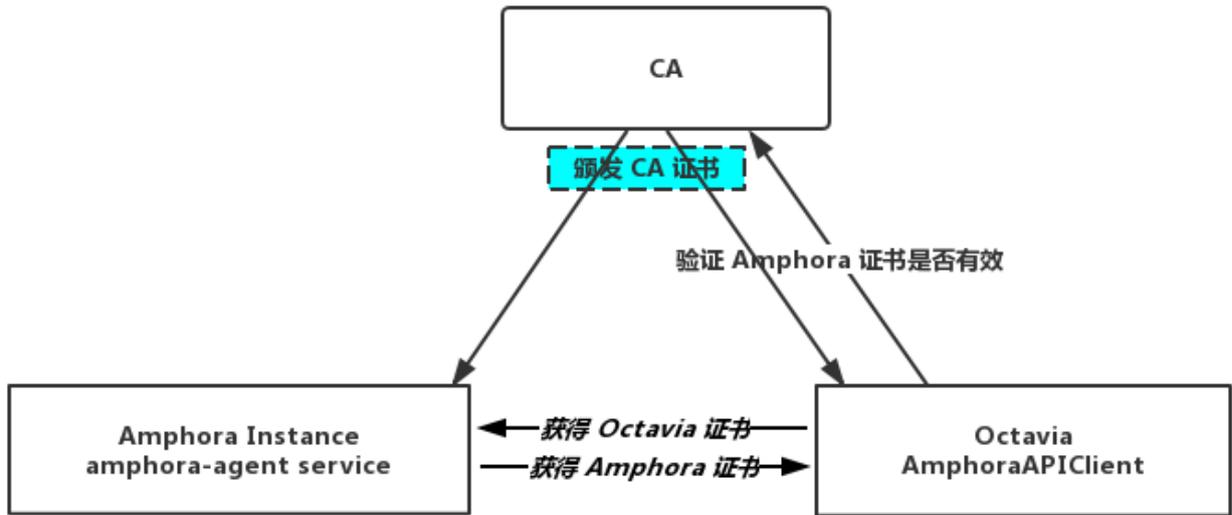
. Amphora

```
# file: /etc/octavia/amphora-agent.conf

[amphora_agent]
agent_server_ca = /etc/octavia/certs/client_ca.pem
agent_server_cert = /etc/octavia/certs/server.pem
```

Gunicorn HTTP 가

```
options = {
    'bind': bind_ip_port,
    'workers': 1,
    'timeout': CONF.amphora_agent.agent_request_read_timeout,
    'certfile': CONF.amphora_agent.agent_server_cert,
    'ca_certs': CONF.amphora_agent.agent_server_ca,
```

<https://blog.csdn.net/Jmilk>

Amphora ()

Health Manager



Health Manager - amphora
amphora가

()

, Health Manager amphora , amphora 가
() 가 .
Health Manager Service 가 amphora

Amphora

manager (octavia/cmd/health_manager.py) , octavia-health-
가 . UDPStatusGetter.check() HealthManager.health_check() 가
가 . UDPStatusGetter.check() .

```

# file: /opt/rocky/octavia/octavia/amphorae/drivers/health/heartbeat_udp.py

class UDPStatusGetter(object):
    """This class defines methods that will gather heatbeats

    The heartbeats are transmitted via UDP and this class will bind to a
    port
    and absorb them
  
```

```

"""
def __init__(self):
    self.key = cfg.CONF.health_manager.heartbeat_key
    self.ip = cfg.CONF.health_manager.bind_ip
    self.port = cfg.CONF.health_manager.bind_port
    self.sockaddr = None
    LOG.info('attempting to listen on %(ip)s port %(port)s',
             {'ip': self.ip, 'port': self.port})
    self.sock = None
    self.update(self.key, self.ip, self.port)

    self.executor = futures.ProcessPoolExecutor(
        max_workers=cfg.CONF.health_manager.status_update_threads)
    self.repo = repositories.Repositories().amphorahealth

def update(self, key, ip, port):
    """Update the running config for the udp socket server

    :param key: The hmac key used to verify the UDP packets. String
    :param ip: The ip address the UDP server will read from
    :param port: The port the UDP server will read from
    :return: None
    """
    self.key = key
    for addrinfo in socket.getaddrinfo(ip, port, 0, socket.SOCK_DGRAM):
        ai_family = addrinfo[0]
        self.sockaddr = addrinfo[4]
        if self.sock is not None:
            self.sock.close()
        self.sock = socket.socket(ai_family, socket.SOCK_DGRAM)
        self.sock.settimeout(1)
        self.sock.bind(self.sockaddr)
        if cfg.CONF.health_manager.sock_rlimit > 0:
            rlimit = cfg.CONF.health_manager.sock_rlimit
            LOG.info("setting sock rlimit to %s", rlimit)
            self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF,
                                 rlimit)
        break # just used the first addr getaddrinfo finds
    if self.sock is None:
        raise exceptions.NetworkConfig("unable to find suitable socket")

```

```

Class:UDPStatusGetter octavia-health-manager amphora
heartbeats(
    )
    . heartbeats
    . init()
    , amphora octavia-health-manager
    UDP
    , (CONF.health_manager.bind_ip,
CONF.health_manager.bind_port)
    .
: amphora octavia-health-manager
.
• Octavia
, ext-net
octavia "lb-mgmt-net"
,

```

CONF.health_manager.bind_ip IP 가 , **amphora** **octavia-health-manager** **OpenStack Management Network** **amphora**가 **ext-net** IP ,

- Octavia , **tenant network** **lb-mgmt-net** , **CONF.health_manager.bind_ip** **lb-mgmt-net** IP , **lb-mgmt-net** **OpenStack Management Network** **devstack** **lb-mgmt-net** **ex-int** , **lb-mgmt-net** **amphora** 가 **octavia-health-manager** 가

Devstack

```
$ neutron port-create --name octavia-health-manager-standalone-listen-port \
--security-group <lb-health-mgr-sec-grp> \
--device-owner Octavia:health-mgr \
--binding:host_id=<hostname> lb-mgmt-net \
--tenant-id <octavia service>

$ ovs-vsctl --may-exist add-port br-int o-hm0 \
-- set Interface o-hm0 type=internal \
-- set Interface o-hm0 external-ids:iface-status=active \
-- set Interface o-hm0 external-ids:attached-mac=<Health Manager Listen
Port MAC> \
-- set Interface o-hm0 external-ids:iface-id=<Health Manager Listen Port
ID>
# /etc/octavia/dhcp/dhclient.conf
request subnet-mask,broadcast-address,interface-mtu;
do-forward-updates false;

$ ip link set dev o-hm0 address <Health Manager Listen Port MAC>
$ dhclient -v o-hm0 -cf /etc/octavia/dhcp/dhclient.conf

o-hm0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450
inet 192.168.0.4 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::f816:3eff:fef0:b9ee prefixlen 64 scopeid 0x20<link>
ether fa:16:3e:f0:b9:ee txqueuelen 1000 (Ethernet)
RX packets 1240893 bytes 278415460 (265.5 MiB)
RX errors 0 dropped 45 overruns 0 frame 0
TX packets 417078 bytes 75842972 (72.3 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

가 , **UDPStatusGetter.check()**

```
def check(self):
    try:
```

```

        obj, srcaddr = self.dorecv()
    except socket.timeout:
        # Pass here as this is an expected cycling of the listen socket
        pass
    except exceptions.InvalidHMACException:
        # Pass here as the packet was dropped and logged already
        pass
    except Exception as e:
        LOG.warning('Health Manager experienced an exception processing
a'
                    'heartbeat packet. Ignoring this packet. '
                    'Exception: %s', e)
    else:
        self.executor.submit(update_health, obj, srcaddr)
        self.executor.submit(update_stats, obj, srcaddr)

```

- self.dorecv()
- self.executor.submit(update_health, obj, srcaddr) health
- self.executor.submit(update_stats, obj, srcaddr) stats

amphora가 heartbeats .

```

# file: /opt/rocky/octavia/octavia/cmd/agent.py

def main():
    # comment out to improve logging
    service.prepare_service(sys.argv)

    gmr.TextGuruMeditation.setup_autorun(version)

    health_sender_proc = multiprocessing.Process(name='HM_sender',
                                                target=health_daemon.run_sender,
                                                args=(HM_SENDER_CMD_QUEUE,))

    health_sender_proc.daemon = True
    health_sender_proc.start()

    # Initiate server class
    server_instance = server.Server()

    bind_ip_port = utils.ip_port_str(CONF.haproxy_amphora.bind_host,
                                      CONF.haproxy_amphora.bind_port)

    options = {
        'bind': bind_ip_port,
        'workers': 1,
        'timeout': CONF.amphora_agent.agent_request_read_timeout,
        'certfile': CONF.amphora_agent.agent_server_cert,
        'ca_certs': CONF.amphora_agent.agent_server_ca,
        'cert_reqs': True,
    }

```

```
'preload_app': True,
'accesslog': '/var/log/amphora-agent.log',
'errorlog': '/var/log/amphora-agent.log',
'loglevel': 'debug',
}
AmphoraAgent(server_instance.app, options).run()
```

amphora-agent 가 , **health_daemon.run_sender**가 ,
amphora가 **octavia-health-manager** heartbeats .

```
# file:
/opt/rocky/octavia/octavia/amphorae/backends/health_daemon/health_daemon.py

def run_sender(cmd_queue):
    LOG.info('Health Manager Sender starting.')
    sender = health_sender.UDPStatusSender()

    keepalived_cfg_path = util.keepalived_cfg_path()
    keepalived_pid_path = util.keepalived_pid_path()

    while True:

        try:
            # If the keepalived config file is present check
            # that it is running, otherwise don't send the health
            # heartbeat
            if os.path.isfile(keepalived_cfg_path):
                # Is there a pid file for keepalived?
                with open(keepalived_pid_path, 'r') as pid_file:
                    pid = int(pid_file.readline())
                    os.kill(pid, 0)

                message = build_stats_message()
                sender.dosend(message)

        except IOError as e:
            # Missing PID file, skip health heartbeat
            if e.errno == errno.ENOENT:
                LOG.error('Missing keepalived PID file %s, skipping health '
                    'heartbeat.', keepalived_pid_path)
            else:
                LOG.error('Failed to check keepalived and haproxy status due
                    'to exception %s, skipping health heartbeat.', e)

        except OSError as e:
            # Keepalived is not running, skip health heartbeat
            if e.errno == errno.ESRCH:
                LOG.error('Keepalived is configured but not running, '
```



```

        'skipping health heartbeat.')
    else:
        LOG.error('Failed to check keepalived and haproxy status due
,
        'to exception %s, skipping health heartbeat.', e)
except Exception as e:
    LOG.error('Failed to check keepalived and haproxy status due to
,
        'exception %s, skipping health heartbeat.', e)

try:
    cmd = cmd_queue.get_nowait()
    if cmd == 'reload':
        LOG.info('Reloading configuration')
        CONF.reload_config_files()
    elif cmd == 'shutdown':
        LOG.info('Health Manager Sender shutting down.')
        break
except queue.Empty:
    pass
time.sleep(CONF.health_manager.heartbeat_interval)

```

```

run_sender      build_stats_message()      heartbeats
UDPStatusSender.dosend()      , keepalived
가      heartbeats      , keepalived가
amphora      가      amphora      UDP
,      URL      CONF.health_manager.controller_ip_port_list

```

```

# file: /etc/octavia/octavia.conf

[health_manager]
bind_port = 5555
bind_ip = 192.168.0.4
controller_ip_port_list = 192.168.0.4:5555

```

```

, octavia-health-manager amphora-agent      heartbeats
amphora
failover(      )
health_manager.HealthManager.health_check()
health_check      amphora_health      "stale amphora"      가
heartbeats      가      amphora

```

```

# file: /opt/rocky/octavia/octavia/db/repositories.py

```

```
def get_stale_amphora(self, session):
    """Retrieves a stale amphora from the health manager database.

    :param session: A Sql Alchemy database session.
    :returns: [octavia.common.data_model]
    """

    timeout = CONF.health_manager.heartbeat_timeout
    expired_time = datetime.datetime.utcnow() - datetime.timedelta(
        seconds=timeout)

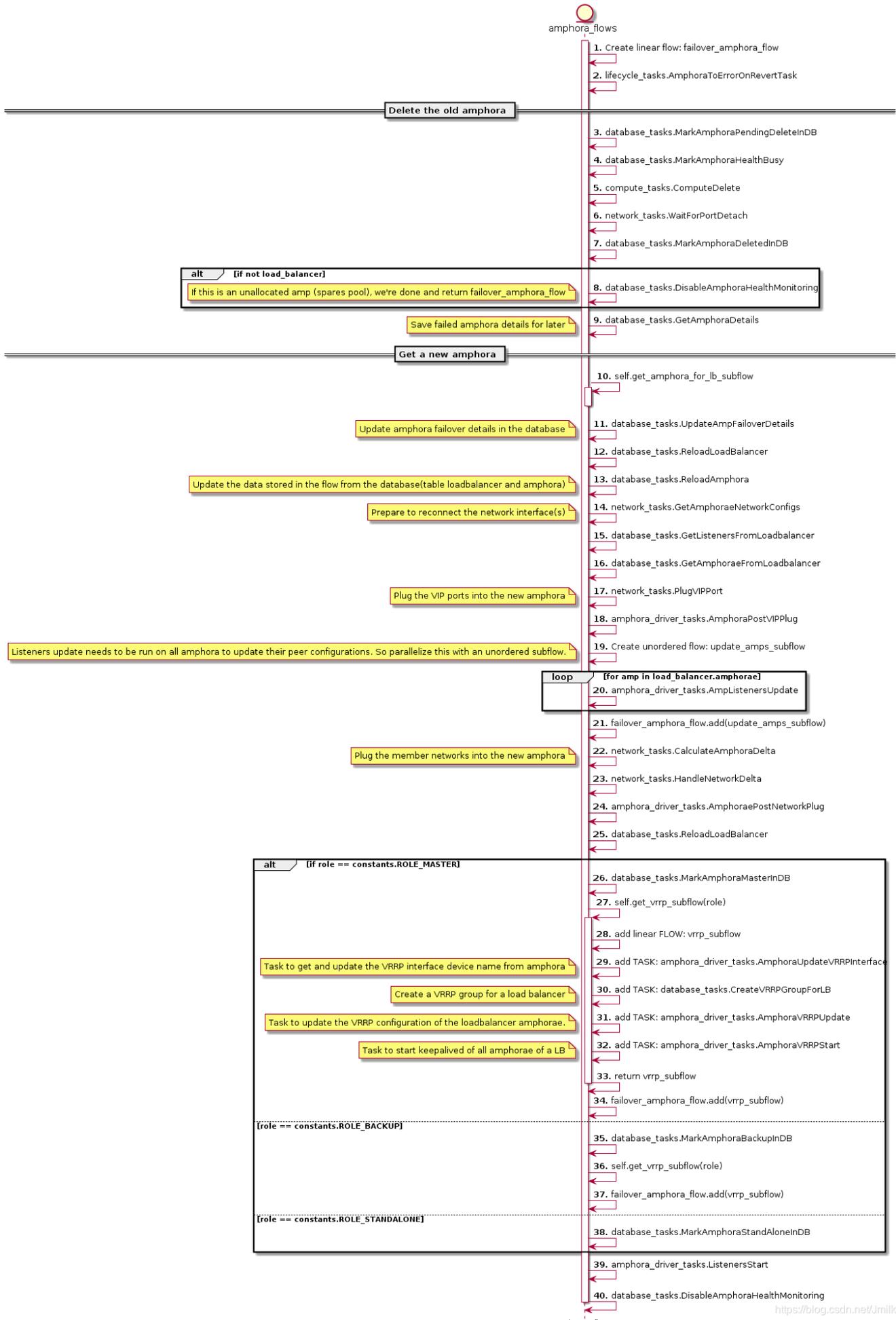
    amp = session.query(self.model_class).with_for_update().filter_by(
        busy=False).filter(
        self.model_class.last_update < expired_time).first()

    if amp is None:
        return None

    amp.busy = True

    return amp.to_data_model()
```

stale amphora가 loadbalancer 가 **PENDING_UPDATE**가 , **failover amphora** . **failover amphora** taskflow
self._amphora_flows.get_failover_flow .
failover UML



<https://blog.csdn.net/Jmilk>

, **failover_flow** “ amphora ” “ amphora ” 가 TASK , .

- delete old amphora
 - MarkAmphoraPendingDeleteInDB
 - MarkAmphoraHealthBusy
 - ComputeDelete: amphora
 - WaitForPortDetach: amphora ()
 - MarkAmphoraDeletedInDB

: 가 amphora가 **free amphora** , .

- get a new amphora
 - **get_amphora_for_lb_subflow**: 가 **free amphora**
 - **UpdateAmpFailoverDetails**: old amphora new amphora (amphora)
 - **ReloadLoadBalancer & ReloadAmphora**: loadbalancer amphora 가 flow stores
 - **GetAmphoraeNetworkConfigs & GetListenersFromLoadbalancer & GetAmphoraeFromLoadbalancer**: listener, amphora 가 flow stores , amphora
 - **PlugVIPPort**: amphora **keepalived** VIP NIC
 - **AmphoraPostVIPPlug**: amphora VIP NIC
 - **update_amps_subflow\AmpListenersUpdate**: listener amphora haproxy , flow unordered , listener가
 - **CalculateAmphoraDelta**: amphora NIC NIC
 - **HandleNetworkDelta**: NIC 가
 - **AmphoraePostNetworkPlug**: member가 가
 - **ReloadLoadBalancer**
 - **MarkAmphoraMasterInDB**
 - **AmphoraUpdateVRRPInterface**: amphora amphora VRRP (: vrrp_interface) 가
 - **CreateVRRPGroupForLB**: amphora loadbalancer / amphorae
 - **AmphoraVRRPUpdate**: amphora **keepalived** VRRP
 - **AmphoraVRRPStart**: **keepalived**
 - **ListenersStart**: haproxy
 - **DisableAmphoraHealthMonitoring**: **amphora_health**

, **amphora failover** 가 **old amphora** , 가 **new amphora** , **old amphora** (e.g.,) (e.g.,) **new amphora** .

:



<p>old amphora</p> <p>, affinity)</p>	<p>new amphora</p> <p>가 (anti- , old amphora)</p>
---	--

 **LB가 ERROR** , **amphora** API
 API 가 , **spares pool**
act/stdby

old amphora , **new amphora**
 , **old amphora** , **new amphora**
 , **loadbalancer** , **ERROR** API **new amphora**가 API
 가 , **space amphora** API API

MASTER amphora **octavia-health-manager** 가 **amphora failover**

```
Nov 22 11:22:31 control01 octavia-health-manager[29147]: INFO
octavia.controller.healthmanager.health_manager [-] Stale amphora's id is:
cd444019-ce8f-4f89-be6b-0edf76f41b77
Nov 22 11:22:31 control01 octavia-health-manager[29147]: INFO
octavia.controller.healthmanager.health_manager [-] Waiting for 1 failovers
to finish
```

old amphorae

```
2ddc4ba5-b829-4962-93d8-562de91f1dab |
amphora-4ff5d6fe-854c-4022-8194-0c6801a7478b | ACTIVE | lb-mgmt-
net=192.168.0.23 |
amphora-x64-haproxy | m1.amphora |
| b237b2b8-afe4-407b-83f2-e2e60361fa07 | amphora-bcff6f9e-4114-4d43-
a403-573f1d97d27e | ACTIVE | lb-mgmt-net=192.168.0.11
| amphora-x64-haproxy | m1.amphora |
| 46eccf47-be10-47ec-89b2-0de44ea3caec | amphora-cd444019-ce8f-4f89-
be6b-0edf76f41b77 | ACTIVE | lb-mgmt-net=192.168.0.9; web-server-
net=192.168.1.3; lb-vip-net=172.16.1.3 | amphora-x64-haproxy |
m1.amphora |
| bc043b23-d481-45c4-9410-f7b349987c98 | amphora-a1c1ba86-6f99-4f60-b469-
a4a29d7384c5 | ACTIVE | lb-mgmt-net=192.168.0.3; web-server-
net=192.168.1.12; lb-vip-net=172.16.1.7 | amphora-x64-haproxy |
m1.amphora |
```

new amphoras

```
| 712ff785-c082-4b53-994c-591d1ec0bf7b | amphora-
```

```
caa6ba0f-1a68-4f22-9be9-8521695ac4f4 | ACTIVE | lb-mgmt-net=192.168.0.13
| amphora-x64-haproxy | m1.amphora |
| 2ddc4ba5-b829-4962-93d8-562de91f1dab |
amphora-4ff5d6fe-854c-4022-8194-0c6801a7478b | ACTIVE | lb-mgmt-
net=192.168.0.23; web-server-net=192.168.1.4; lb-vip-net=172.16.1.3 |
amphora-x64-haproxy | m1.amphora |
| b237b2b8-afe4-407b-83f2-e2e60361fa07 | amphora-bcff6f9e-4114-4d43-
a403-573f1d97d27e | ACTIVE | lb-mgmt-net=192.168.0.11
| amphora-x64-haproxy | m1.amphora |
| bc043b23-d481-45c4-9410-f7b349987c98 | amphora-a1c1ba86-6f99-4f60-b469-
a4a29d7384c5 | ACTIVE | lb-mgmt-net=192.168.0.3; web-server-
net=192.168.1.12; lb-vip-net=172.16.1.7 | amphora-x64-haproxy |
m1.amphora |
```

new amphora haproxy config

```
# Configuration for loadbalancer 01197be7-98d5-440d-a846-cd70f52dc503
global
    daemon
    user nobody
    log /dev/log local0
    log /dev/log local1 notice
    stats socket /var/lib/octavia/1385d3c4-615e-4a92-aea1-c4fa51a75557.sock
mode 0666 level user
    maxconn 1000000
    external-check

defaults
    log global
    retries 3
    option redispatch

peers 1385d3c4615e4a92aea1c4fa51a75557_peers
    peer 3dVescsRZ-RdRBfYVLW6snVI9gI 172.16.1.3:1025
    peer l_Ustq0qE-h-_Q1dLXLXBAiWR8U 172.16.1.7:1025

frontend 1385d3c4-615e-4a92-aea1-c4fa51a75557
    option httplog
    maxconn 1000000
    bind 172.16.1.10:8080
    mode http
        acl 8d9b8b1e-83d7-44ca-a5b4-0103d5f90cb9 req.hdr(host) -i -m beg
server
    use_backend 8196f752-a367-4fb4-9194-37c7eab95714 if 8d9b8b1e-83d7-44ca-
a5b4-0103d5f90cb9
        acl c76f36bc-92c0-4f48-8d57-a13e3b1f09e1 req.hdr(host) -i -m beg
server
```

```
    use_backend 822f78c3-ea2c-4770-bef0-e97f1ac2eba8 if
c76f36bc-92c0-4f48-8d57-a13e3b1f09e1
    default_backend 8196f752-a367-4fb4-9194-37c7eab95714
    timeout client 50000

backend 8196f752-a367-4fb4-9194-37c7eab95714
    mode http
    balance roundrobin
    timeout check 10s
    option external-check
    external-check command /var/lib/octavia/ping-wrapper.sh
    fullconn 1000000
    option allbackups
    timeout connect 5000
    timeout server 50000
    server b6e464fd-dd1e-4775-90f2-4231444a0bbe 192.168.1.14:80 weight 1
check inter 5s fall 3 rise 3

backend 822f78c3-ea2c-4770-bef0-e97f1ac2eba8
    mode http
    balance roundrobin
    timeout check 10s
    option external-check
    external-check command /var/lib/octavia/ping-wrapper.sh
    fullconn 1000000
    option allbackups
    timeout connect 5000
    timeout server 50000
    server 7da6f176-36c6-479a-9d86-c892ecca6ae5 192.168.1.6:80 weight 1
check inter 5s fall 3 rise 3
```

new amphora keepalived config

```
vrrp_script check_script {
    script /var/lib/octavia/vrrp/check_script.sh
    interval 5
    fall 2
    rise 2
}

vrrp_instance 01197be798d5440da846cd70f52dc503 {
    state MASTER
    interface eth1
    virtual_router_id 1
    priority 100
    nopreempt
    garp_master_refresh 5
    garp_master_refresh_repeat 2
    advert_int 1
```

```
authentication {
  auth_type PASS
  auth_pass b76d77e
}

unicast_src_ip 172.16.1.3
unicast_peer {
  172.16.1.7
}

virtual_ipaddress {
  172.16.1.10
}

track_script {
  check_script
}
}
```

new amphora

old amphora

Neutron-lbaas vs. LBaaS v2 API vs. Octavia vs. Octavia v2 API

가 LBaaS v2 API Octavia v2 API 1

- Neutron-lbaas: Neutron , LBaaS
 - LBaaS v2 API: LBaaS API v2 Neutron-lbaas , ,
 - Octavia: OpenStack , LBaaS
 - Octavia v2 API: Octavia API v2 LBaaS v2 API , Neutron-lbaas
- octavia driver 가 .

- <https://www.cnblogs.com/jmilkfan-fanguiju/p/10589749.html>
- <https://blog.csdn.net/jmilk>
- https://blog.51cto.com/u_15301988/3126511

From: <https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link: https://atl.kr/dokuwiki/doku.php/octavia_lb_%EA%B5%AC%ED%98%84_%EB%B0%8F_%EB%B6%84%EC%84%9D

Last update: 2024/10/10 06:29

