

Octavia LB 3

Octavia 3

 3

 4

 27

haproxy 29

 29

Octavia LB

Octavia

Octavia OpenStack

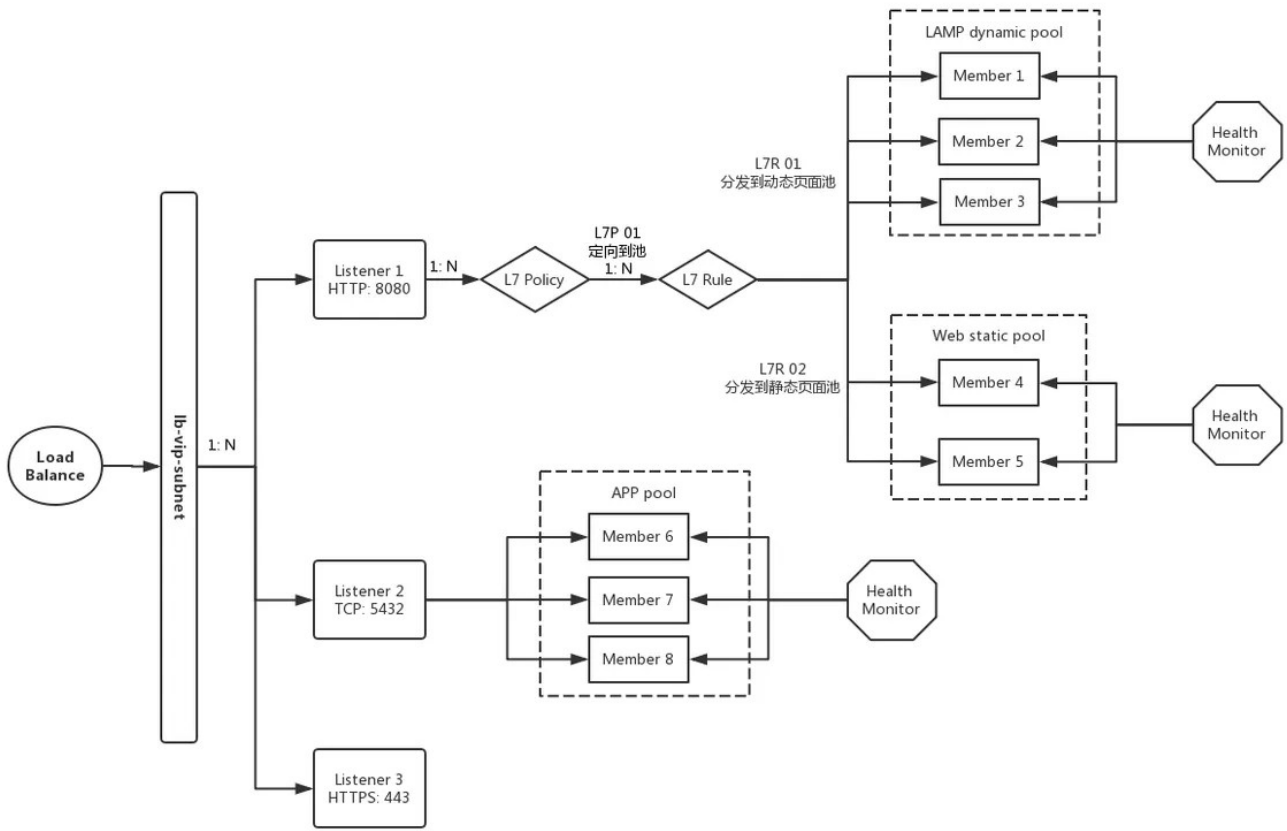
Pike OpenStack Extension Octavia Queens neutron-lbaas
neutron-lbaas

. Neutron-lbaas is now deprecated.

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

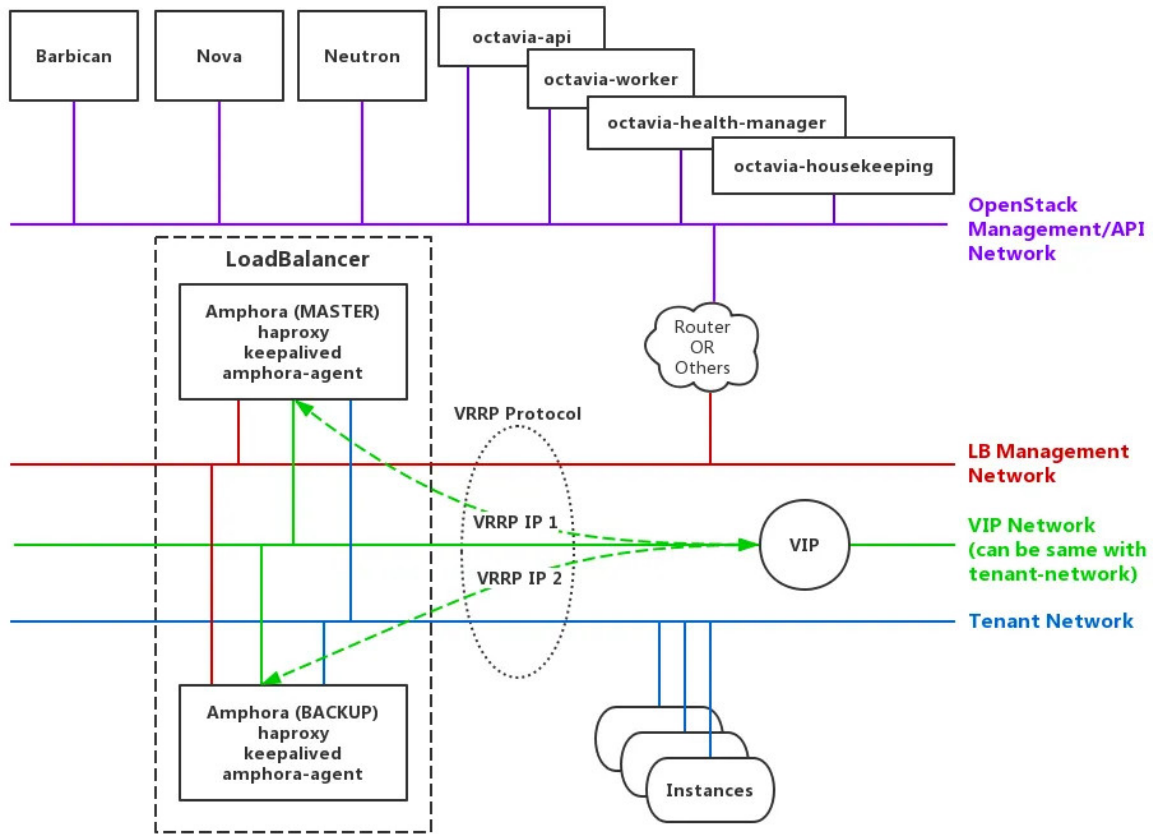
Rocky OpenStack LBaaS Octavia
Octavia

- LBaaS : OpenStack LB()가
- 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?

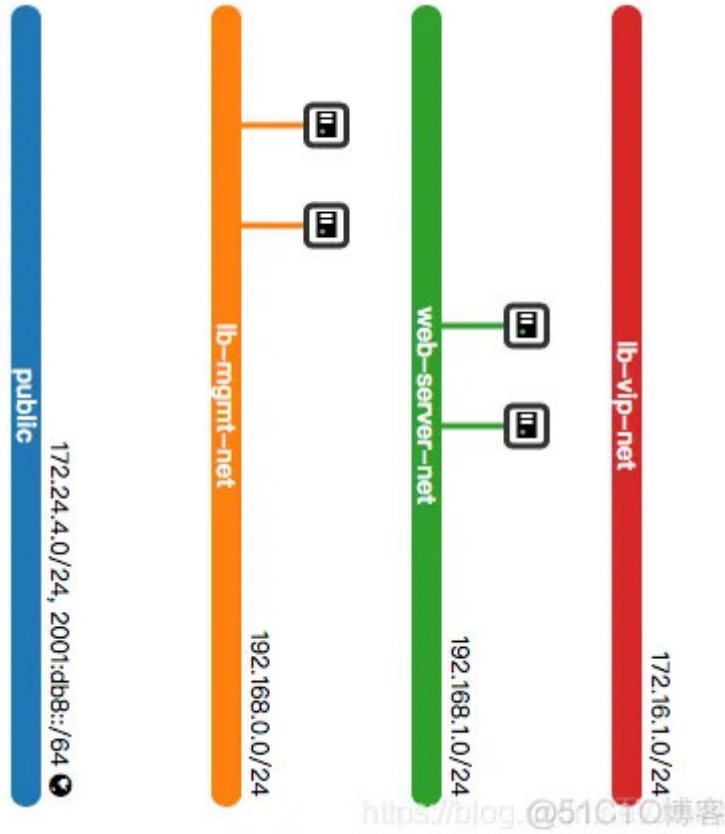


<https://blog.@51CTO博客>

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

Load Balancer Details

Provide the details for the load balancer.

名称	<input type="text" value="Load Balancer 1"/>	描述	<input type="text"/>
IP address	<input type="text"/>	Subnet *	<input type="text" value="lb-vip-subnet"/>

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

Pool Details *

协议 *

Port *

Pool Members

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 Details

Method *

Pool Members

Monitor Details *

取消

返回

下一步

Create Load Balancer

4

가

Create Load Balancer



Load Balancer Details

Add members to the load balancer pool.



Listener Details

Allocated Members 2

Pool Details

Pool Members

Monitor Details *

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

Provide the details for the health monitor.



Listener Details

Monitor type *

PING

Pool Details

Interval (sec) *

5

Retries *

3

Timeout (sec) *

5

Pool Members

Monitor Details

取消

< 返回

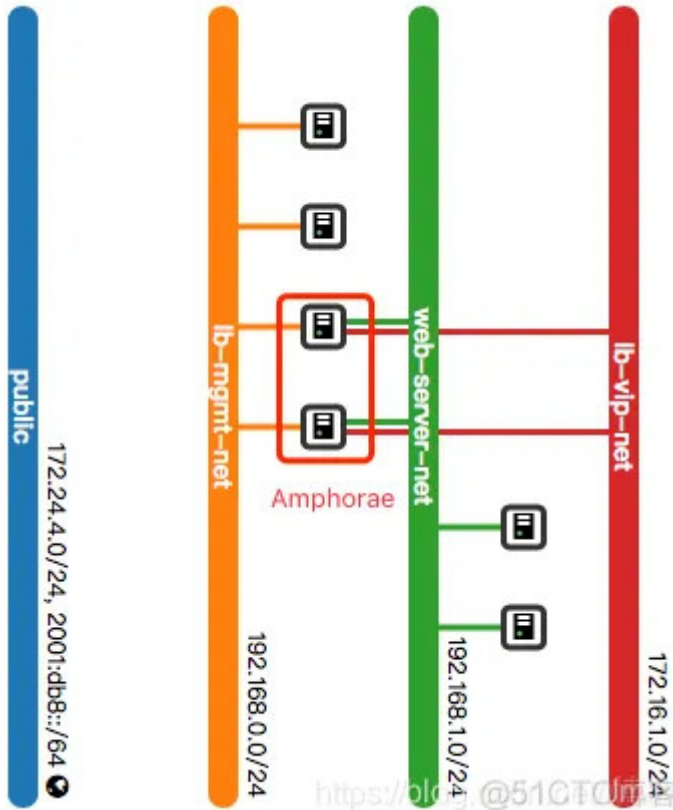
下一步 >

Create Load Balancer

VIP, Member Octava

. Amphorae가

가

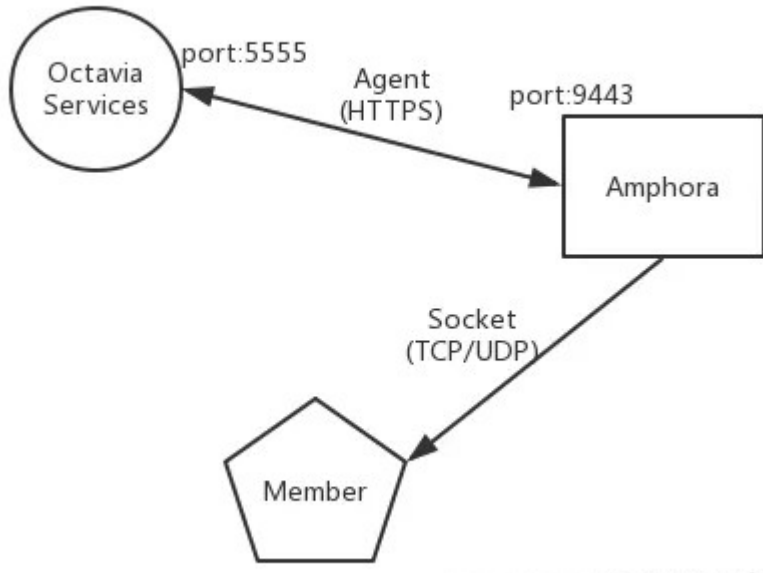


<https://blog.@51CTO博客>

Octavia Amphora Provider (HAProxy) 가 Octavia

(Keepalived)
* Octavia haproxy keepalived
Amphora Member Socket(IP, Port)

VIP
* Member가 Subnet Amphora



<https://blog.@51CTO博客>

Octavia

가 . Amphora
Octavia centos ubuntu
가 ingress (UDP/5555 egress:TCP/9443)
: 1

, Amphora

```
$ /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
```

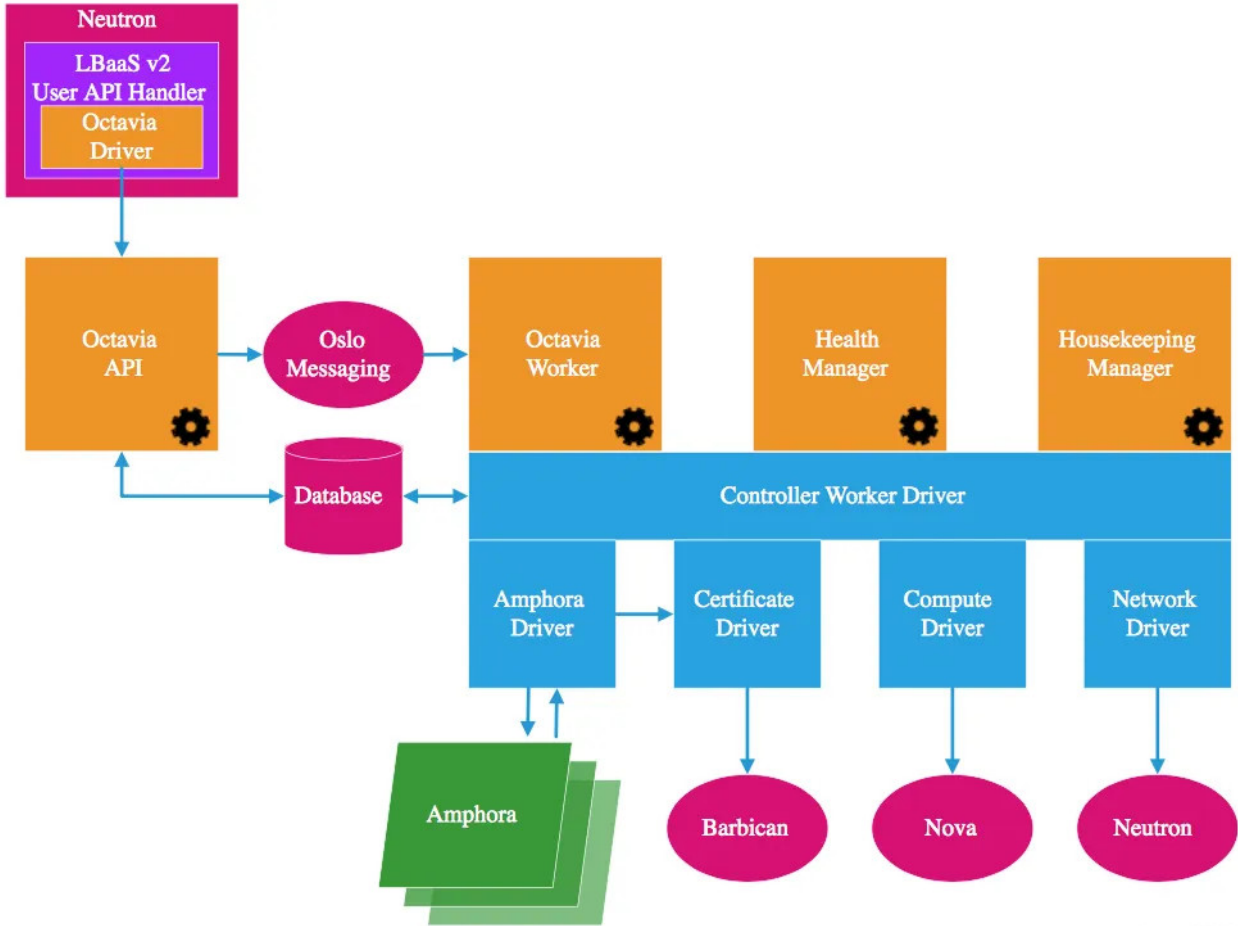
```
$ openstack security group create amphora-sec-grp --project <admin project id>  
$ openstack security group rule create --remote-ip "0.0.0.0/0" --dst-port 9443 --protocol tcp --ingress --ethertype IPv4 --project <admin project id>  
amphora-sec-grp  
$ openstack security group rule create --remote-ip "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 LB (:
 F5) . openstack/neutron-lbaas
 Octavia . =====
 ===== . * 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
```

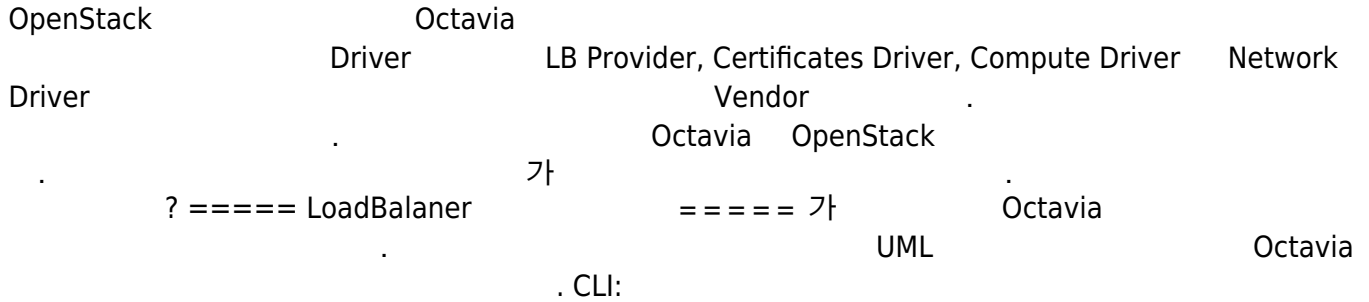
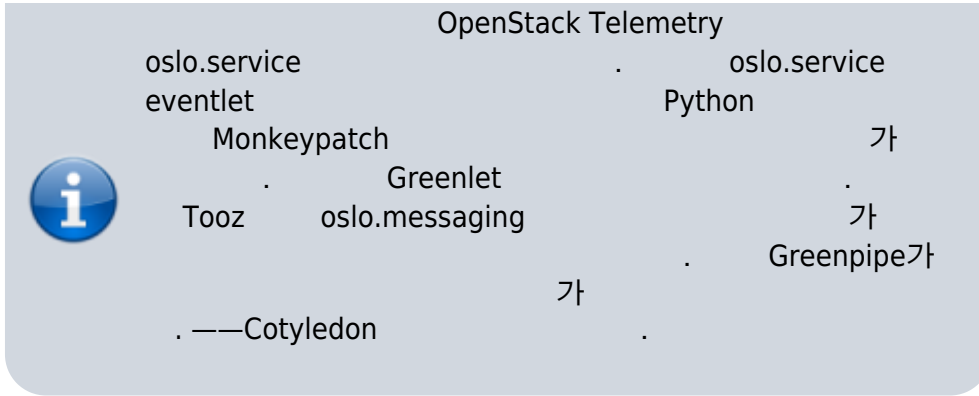
amphora Rest API amphora-agent * api : Octavia API * certificates : CA
 amphora Octavia Worker HTTPS TLS * Compute : Compute Driver
 novaclient . * network : neutronclient
 . * db : ORM * policies : API

```
[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
```

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



Cotyledon
 Unix , , ,
 , sd-notify, .



```
$ 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
  }
}
```

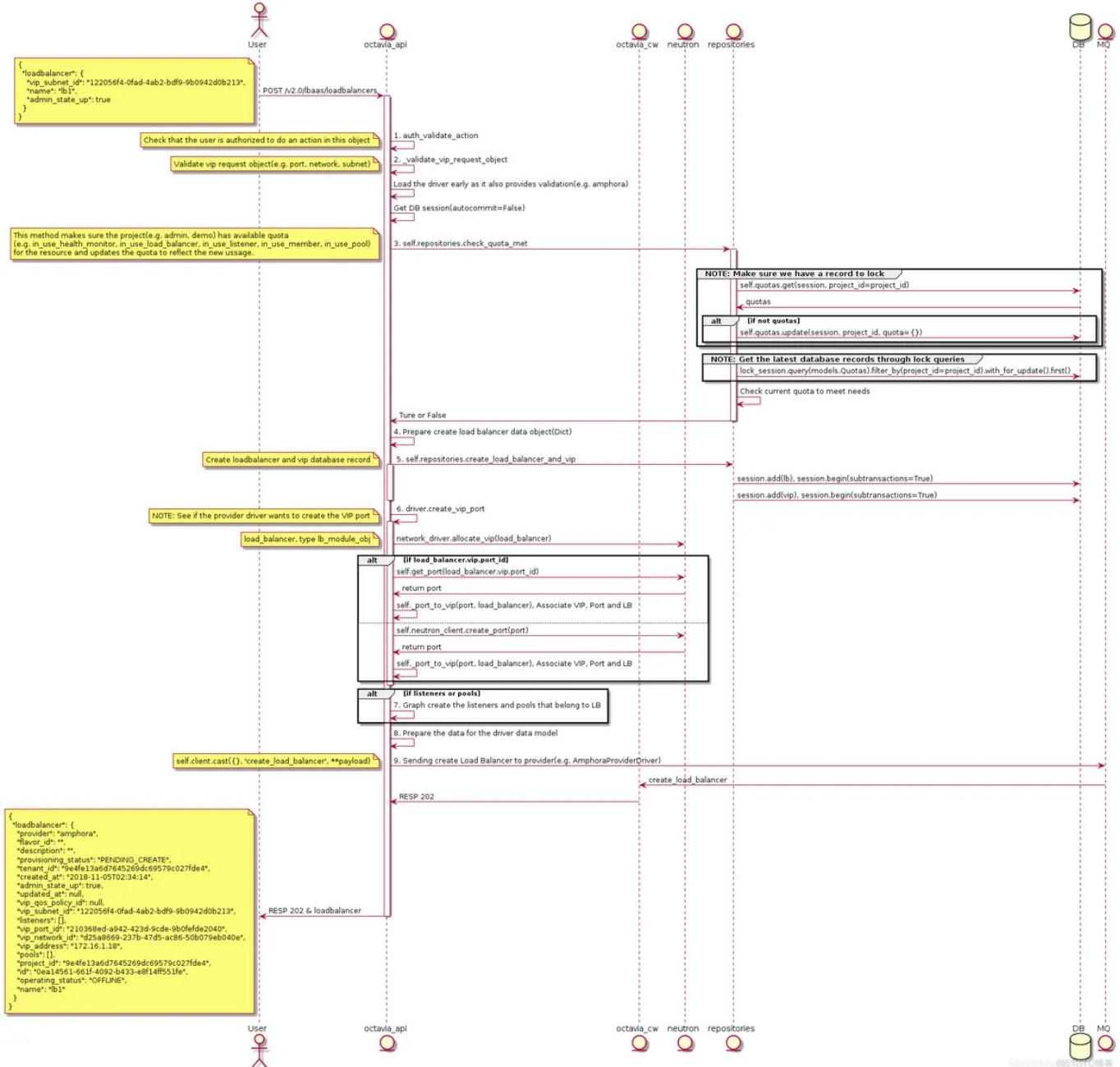
```
{
  "loadbalancer": {
    "provider": "octavia",
    "flavor_id": "",
    "description": "",
    "provisioning_status": "PENDING_CREATE",
    "created_at": "2018-10-22T02:52:04",
    "admin_state_up": true,
  }
}
```

```

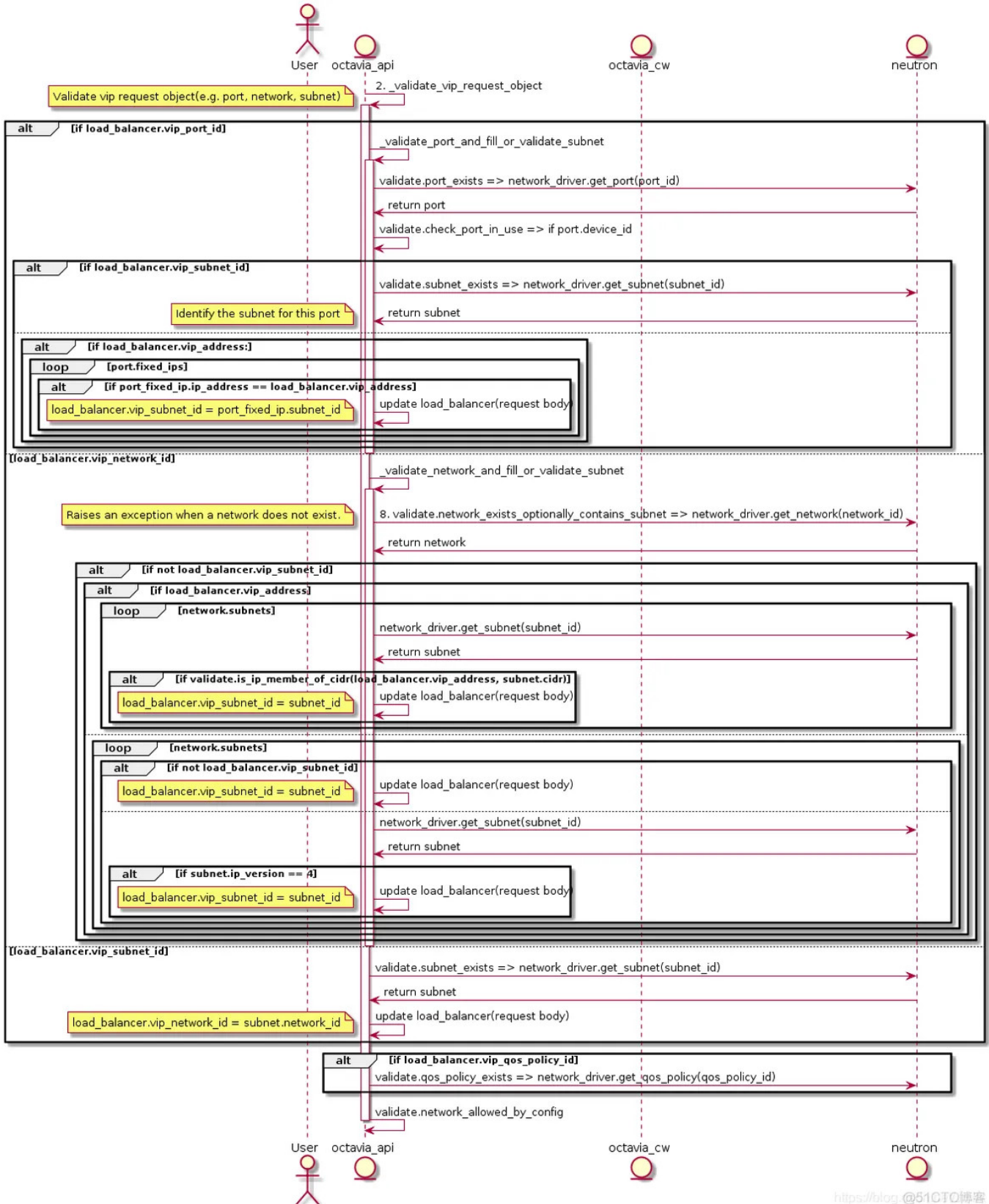
"updated_at": null,
"vip_subnet_id": "c55e7725-894c-400e-bd00-57a04ae1e676",
"listeners": [],
"vip_port_id": "6629fef4-fe14-4b41-9b73-8230105b2e36",
"vip_network_id": "1078e169-61cb-49bc-a513-915305995be1",
"vip_address": "10.0.1.7",
"pools": [],
"project_id": "2e560efadb704e639ee4bb3953d94afa",
"id": "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>

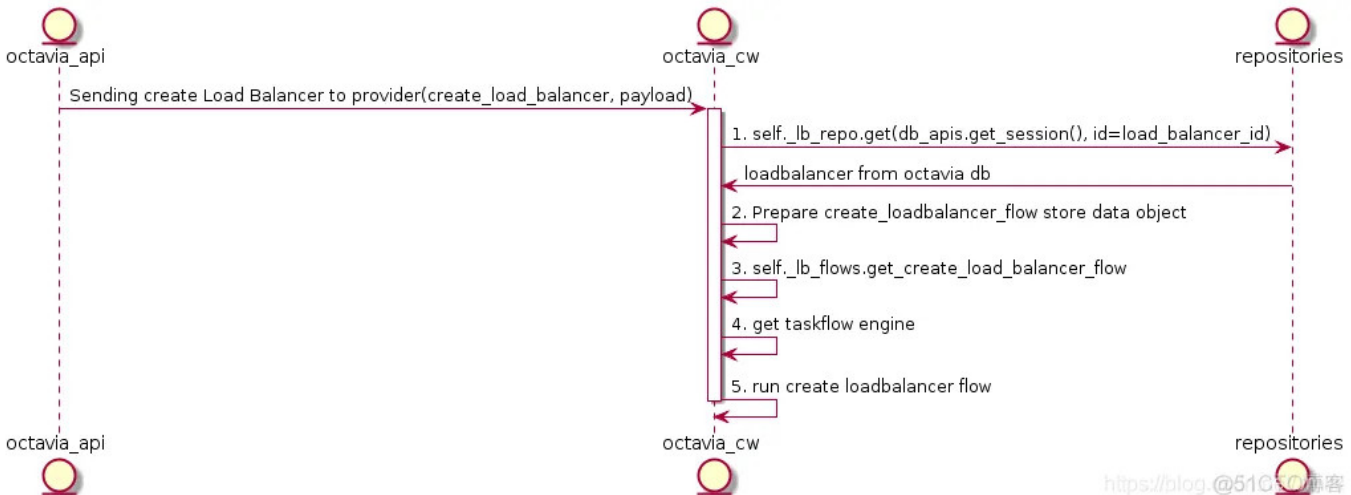
```

POST /v2.0/lbaas/loadbalancers octavia-api 가 :-
. - VIP
( : , , ) . VIP
/ config section [networking] . -
LB . config section [quotas]
( : Project1 3 ). -
load_balancer vip . - Amphora ( lb
) VIP Port, VIP LB
  
```

```

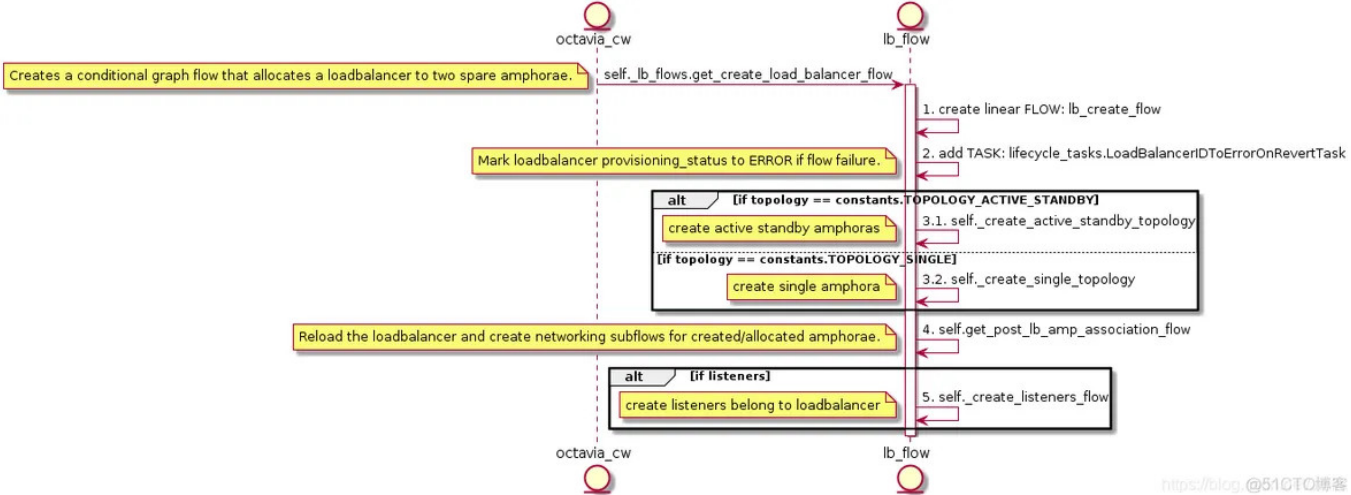
create_loadbalancer_flow
    create_loadbalancer_flow
*
loadbalancer create
    --pools
    , POST /v2.0/lbaas/loadbalancers
    UI/UX
    *
    VIP
    octavia-api
    neutronclient
    loadbalancer-<load_balancer_id>
    vip-net
    *
    VIP QoS
    . Create LB
    Octavia Controller Worker UML
    :

```



<https://blog.@51CTO博客>

3. get_create_load_balancer_flow UML

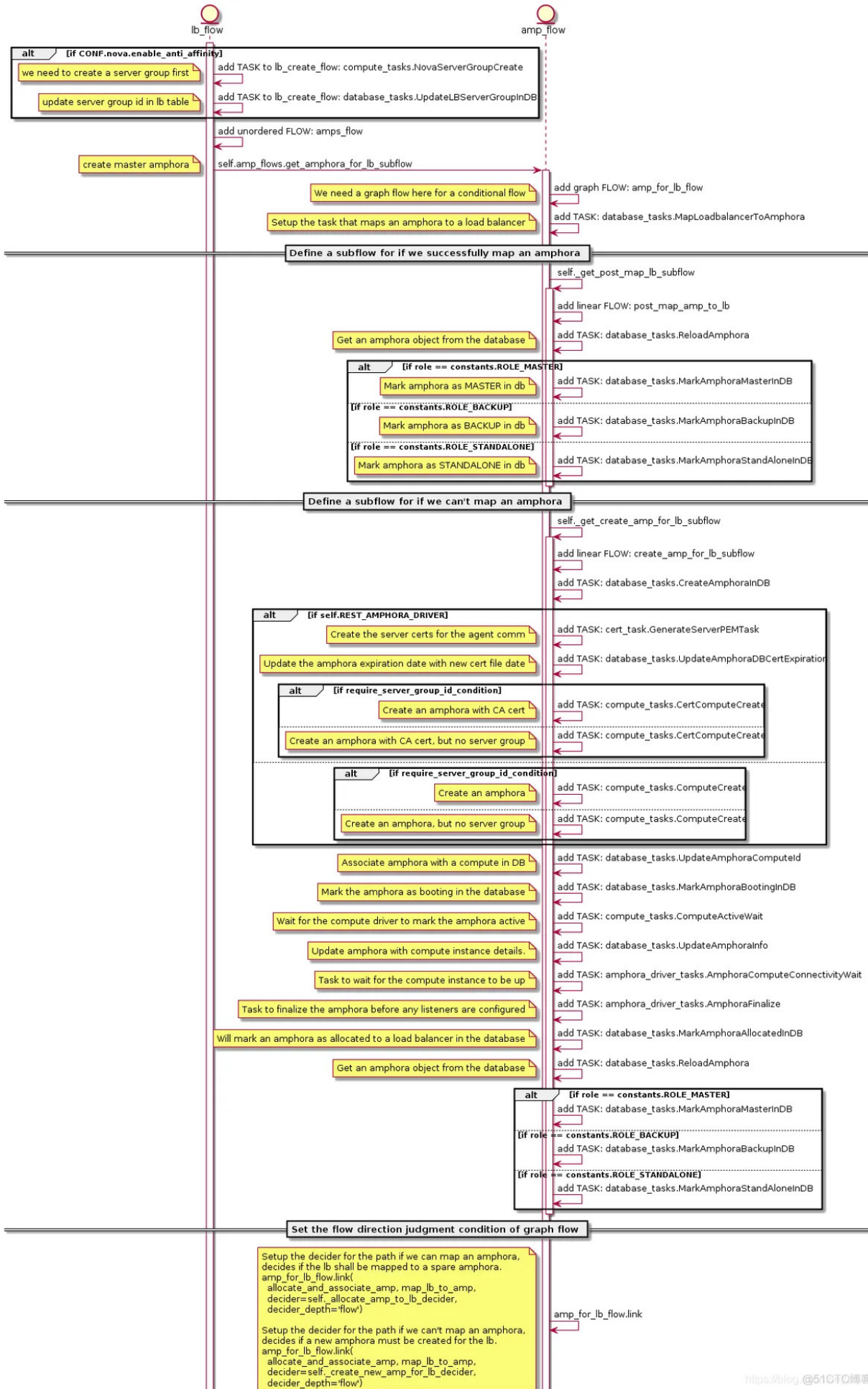


<https://blog.@51CTO博客>

```

amphora(e)
    amphorae
    SINGLE ACTIVE_STANDBY
    SINGLE ACTIVE_STANDBY Keepalived / SINGLE
    . Amphora
    UML
    :

```

https://blog @51CTO博客

가

. *

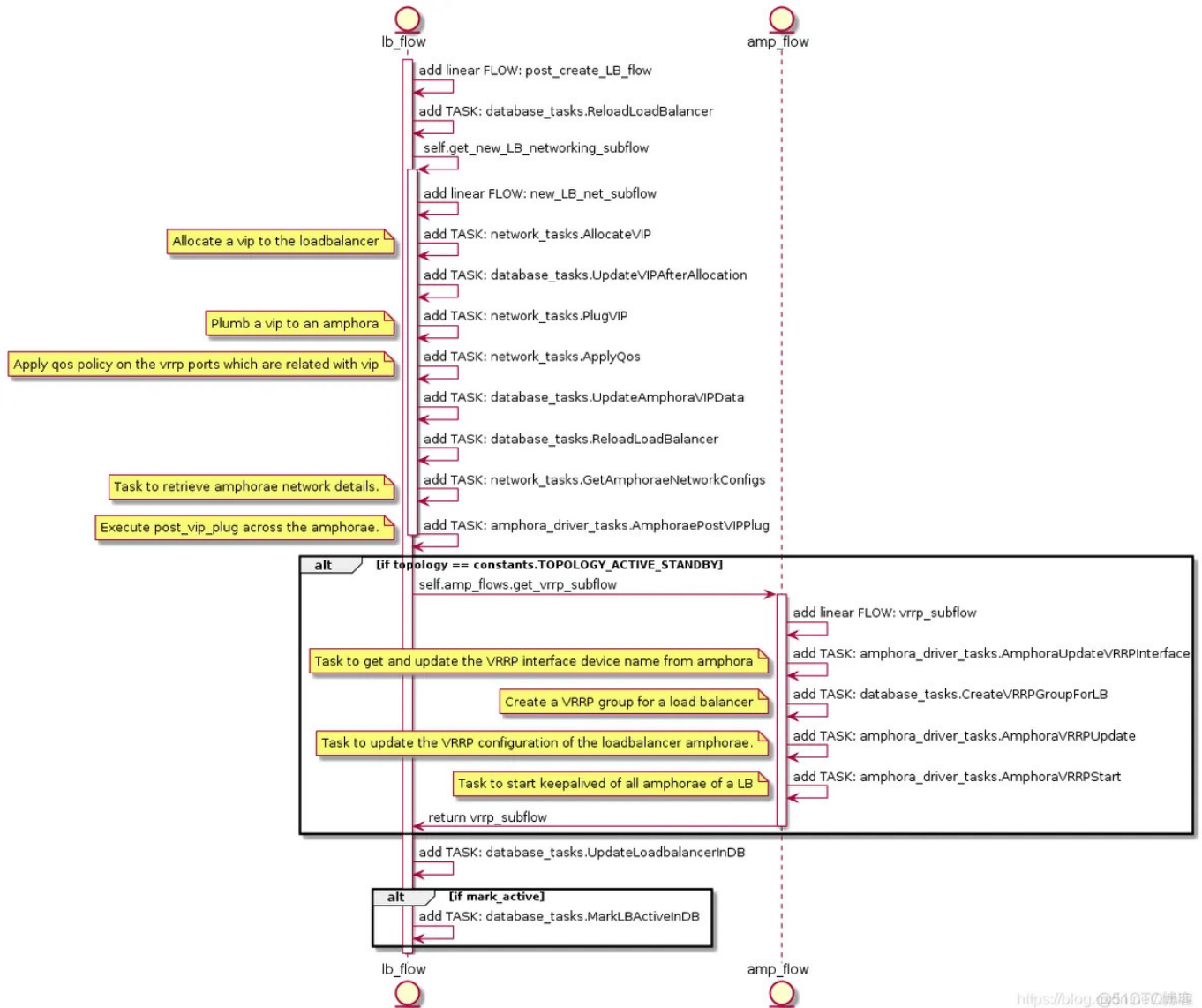
가 ACTIVE_STANDBY

[nova]

```
enable_anti_affinity = True          Nova          가
                                     *
                                     space amphora pool . amphora for lb flow
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
```

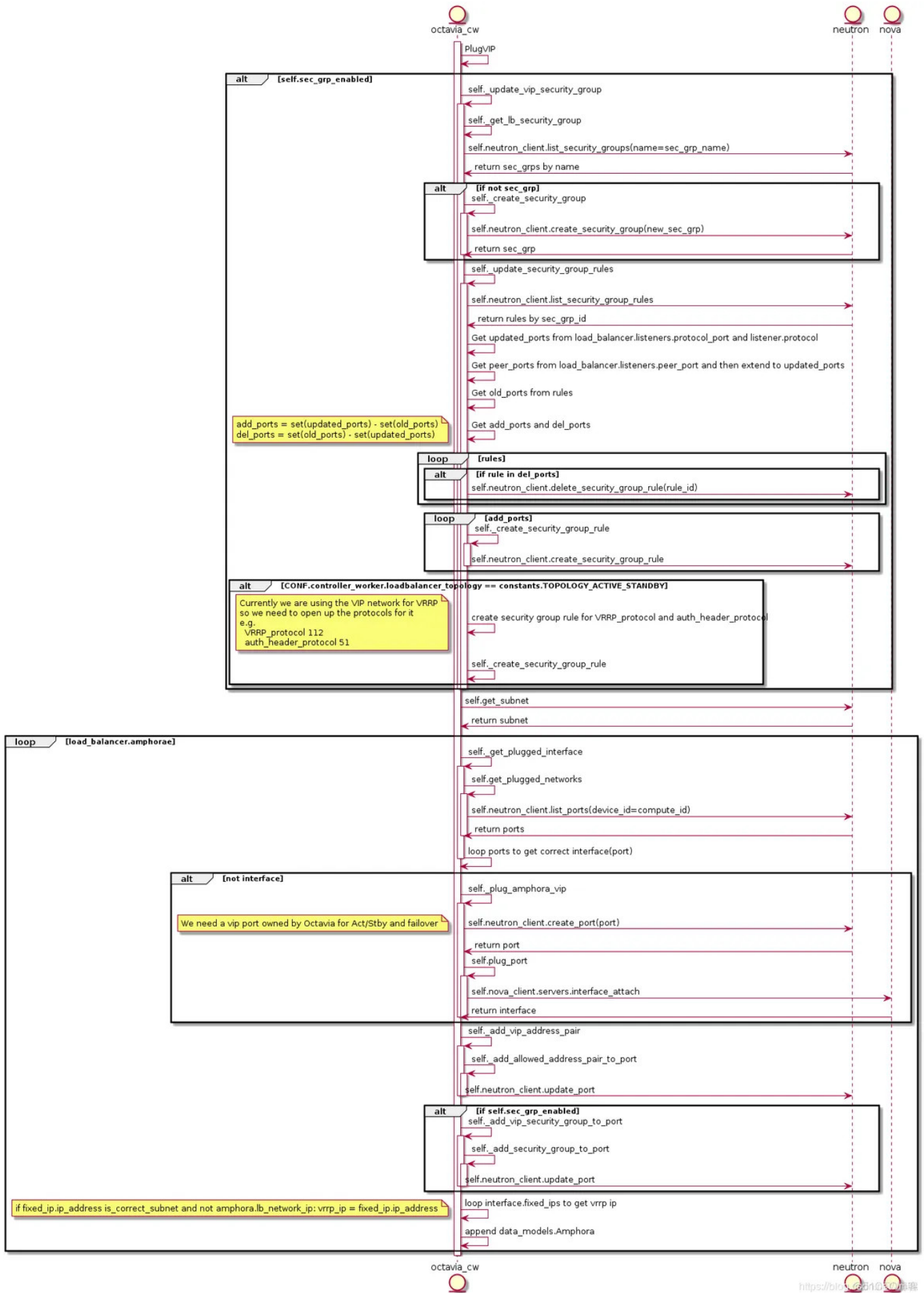
```
amphora가 lb-mgmt-net 가 loadbalancer
vip-net amphora octavia-api vip-net
port:loadbalancer-<load_balancer_id>가
ACTIVE_STANDBY Keepalived VIP
amphora(e) vip-net VRRP_port (octavia-lb-vrrp-<amphora_id>)가
UML :
```



<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

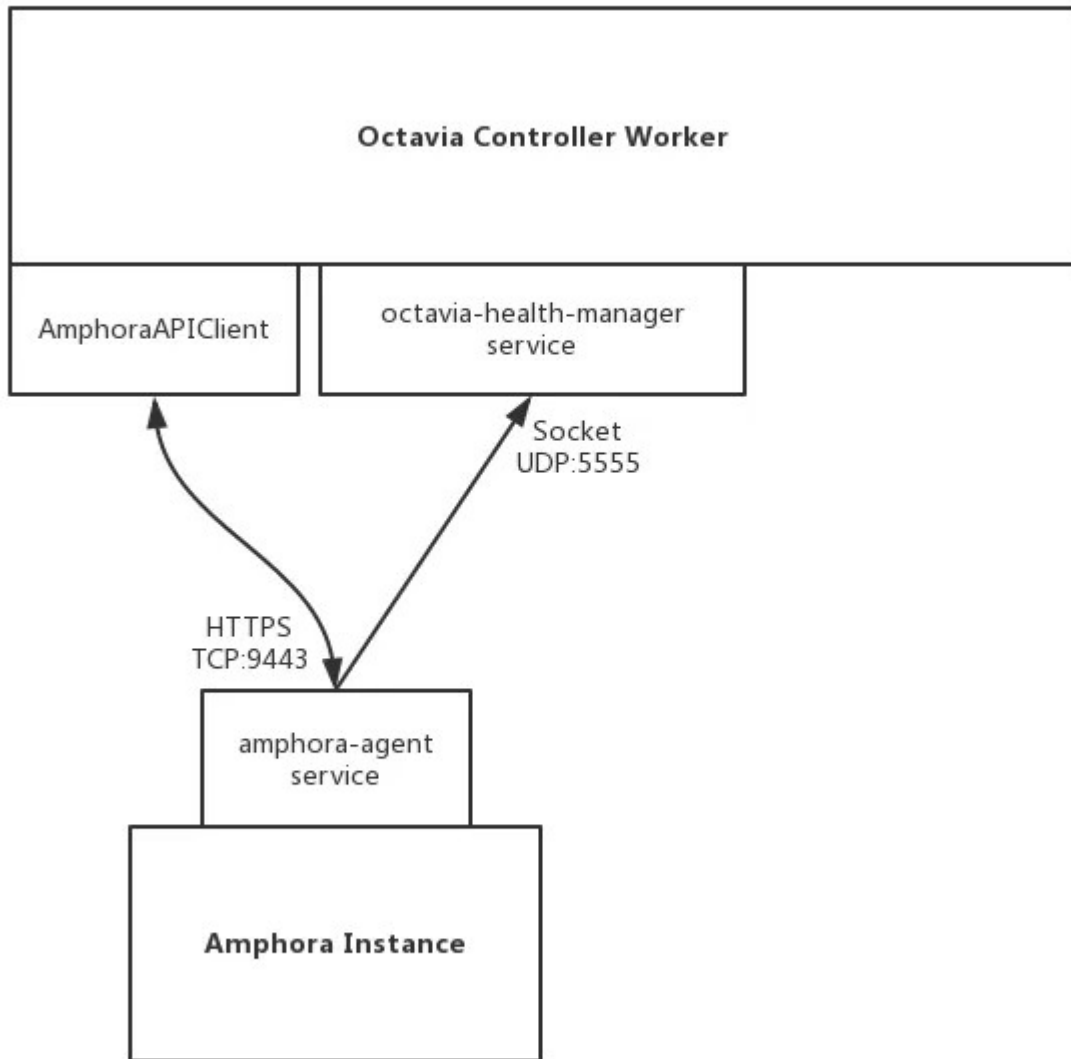
====
 network_tasks.AllocateVIP ==== AllocateVIP VIP 가 Port, VIP
 LB data_models.Vip Neutron
 AllowedAddressPairsDriver.allocate_vip . octavia-api
 octavia-worker VIP , data_models.Vip
 Task:UpdateAmphoraVIPData . ==== network_tasks.PlugVIP
 ==== AllocateVIP Neutron VIP PlugVIP Amphora VIP
 . PlugVIP UML



VIP
 VIP HTTP:8080 가
 API flow Amphora
 Worker Amphora 가
 Amphora Amphora
 Amphora Amphora
 가 ?
 Controller Worker

VIP
 HTTP:8080 가
 Neutron API
 TASK:AllocateVIP TASK:PlugVIP
 Amphora Amphora
 Amphora Amphora
 Amphora Agent AmphoraAPIClient
 가 가
 HAProxy Keepalived
 가 ?
 가 가 ! amphora-agent Octavia

Nova create lb
 Amphora
 Octavia Controller
 가
 가
 가 ?



<https://blog.@51CTO博客>

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

. ==== Amphora

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

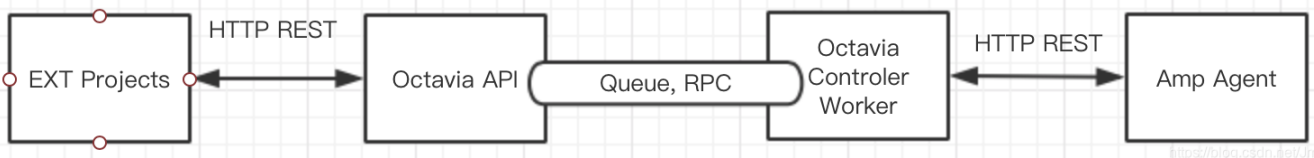
class Server(object):
    def __init__(self):
        self.app = flask.Flask(__name__)
        ...
        self.app.add_url_rule(rule=PATH_PREFIX +
            '/listeners/<amphora_id>/<listener_id>/haproxy',
                view_func=self.upload_haproxy_config,
                methods=['PUT'])
    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

```
Flask
gunicorn
Octavia HAProxy Amphora API
==== 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
REST API
* Amphora agent: Amphora Octavia Controller Worker
```



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

```

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
가 view_func:upload_keepalived_config . keepalived
```

```
# 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 $?

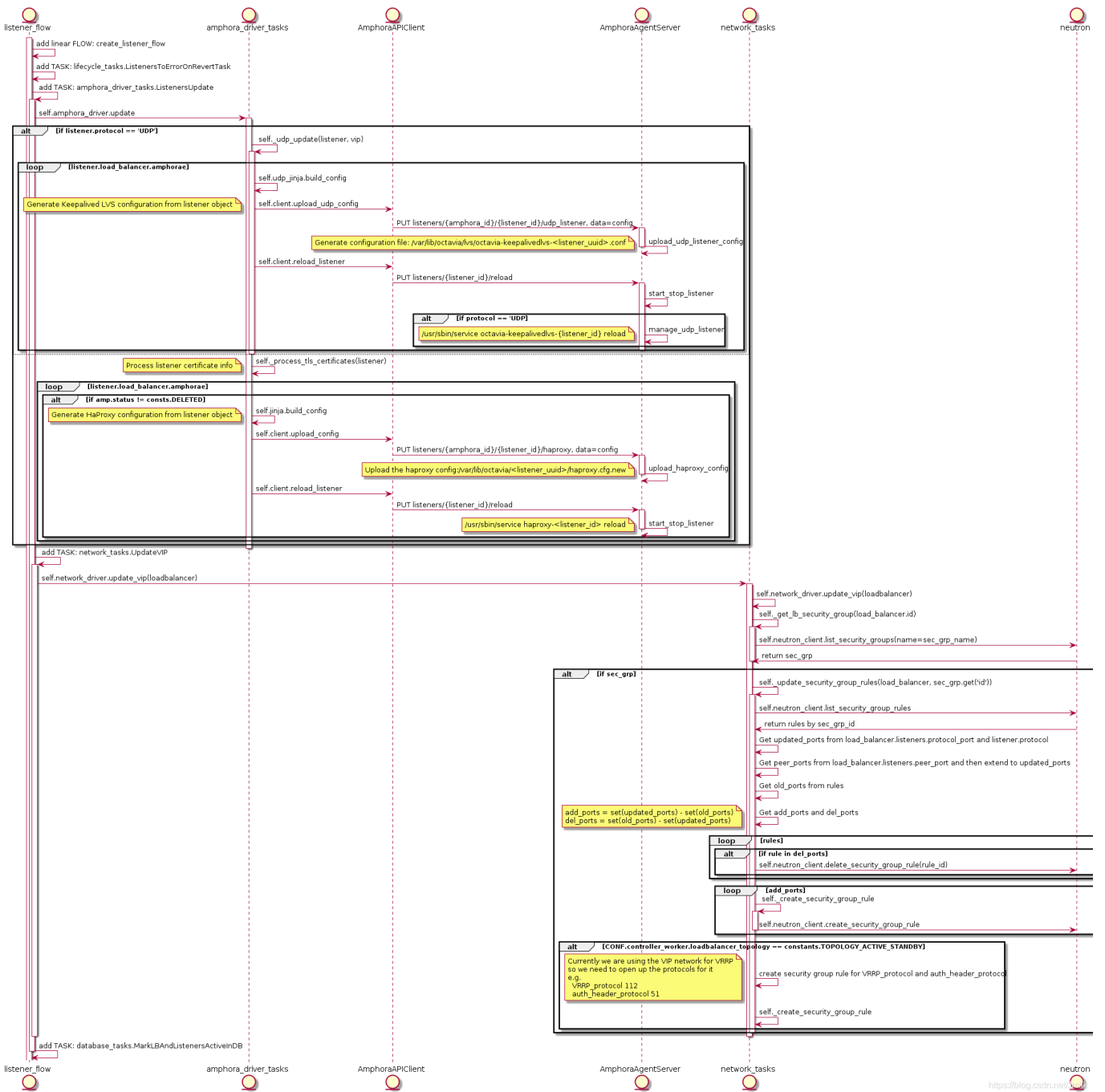
```

```

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
```

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

From:
<https://at1.kr/dokuwiki/> - AllThatLinux!

Permanent link:
https://at1.kr/dokuwiki/doku.php/octavia_lb_%EA%B5%AC%ED%98%84_%EB%B0%8F_%EB%B6%84%EC%84%9D?rev=1728534217

Last update: 2024/10/10 04:23

