



# **Openstack Juno Kilo and Liberty LBaaS Configuration Guide**

## Table of Contents

Table of Contents .....	I
Introduction .....	1
APV LBaaS v1.0 Configuration Guide.....	3
Deploy the Driver.....	3
Configure the LBaaS Service.....	6
vAPV LBaaS v1.0 Configuration Guide.....	9
Deploy the Driver.....	9
Configure the LBaaS Service.....	11
vAPV LBaaS v2.0 Configuration Guide.....	14
Deploy the Driver.....	14
Configure the LBaaS Service.....	16

# Introduction

Openstack is an open-source cloud operating platform that controls large pools of compute, storage and networking resources throughout a data center.

In Openstack architecture, Networking provides a pluggable, scalable and API-driven system for network and IP address management. In addition, it also has an extension framework allowing Load Balancing as a Service (LBaaS) provided by different vendors. Using the APV/vAPV LBaaS driver, you can insert APV/vAPV into the Openstack platform as one option of load balancing service providers.

For the Juno and Kilo platforms, Array Networks provides the APV/vAPV v1.0 driver solutions. For the Kilo devstack and Liberty platforms, which supports API v2.0, Array Networks provides the vAPV v2.0 driver solutions.

All of these driver solutions support the following load balancing functionalities:

- Load balancing method
  - Round robin
  - Least connection
  - Source IP
- Session persistence method
  - Source IP
  - HTTP cookie (cookie inserted by APV/vAPV)
  - APP cookie (cookie returned from real servers)

The mappings of load balancing algorithms between Openstack and APV/vAPV are shown in the following table:

**Table 1–1 Mappings of load balancing algorithms between Openstack and APV/vAPV**

OpenStack		APV	
Load Balancing Method	Session Persistence	Group Method	First Choice Method
ROUND_ROBIN	NA	rr	NA
LEAST_CONNECTION	NA	lc	NA
SOURCE_IP	NA, SOURCE_IP	hi	NA
ROUND_ROBIN	SOURCE_IP	pi	rr
LEAST_CONNECTION	SOURCE_IP	pi	lc
ROUND_ROBIN	HTTP_COOKIE	ic	rr
LEAST_CONNECTION	HTTP_COOKIE	ic	lc
ROUND_ROBIN	APP_COOKIE	hc	rr
LEAST_CONNECTION	APP_COOKIE	hc	lc

- VIP protocol
  - HTTP
  - HTTPS (SSL pass-through)
  - TCP
- Health monitor
  - Ping
  - HTTP
  - HTTPS
  - TCP

Besides these common load balancing functionalities, there are some differences among the solutions, such as the installation environment required and other features supported. The following table shows a comparison.

**Table 1–2 Array Solutions and Supported Features**

Feature	APV LBaaS v1.0 driver	vAPV LBaaS v1.0 driver	vAPV LBaaS v2.0 driver
Environment	Juno and Kilo	Juno and Kilo	Kilo devstack and Liberty
User Interface	Dashboard and CLI	Dashboard and CLI	CLI
Cluster	Support	Not support	Not support
Multi-device	Support	Support	Support
Multi-tenant	Not support	Support	Support

You can select the desired solution based on your Openstack environment, or set up a new one according to application needs. For information about how to set up the Openstack environment, you may refer to <http://docs.openstack.org/>.

# APV LBaaS v1.0 Configuration Guide

The APV LBaaS v1.0 driver solution applies to the Openstack Juno and Kilo environments.

When the environment is ready, follow these guidelines to deploy the APV LBaaS v1.0 driver and configure the LBaaS service.

## Deploy the Driver

**Step1.** Connect the APV appliance to the Openstack environment.

1. Connect the management port of the APV appliance to the OpenStack management network, and use the “**ip address**” command to assign an IP address to the management port.
2. Connect the traffic port of the APV appliance to the OpenStack environment, and use the “**ip address**” command to assign an IP address to the traffic port.
3. Run the “**user**” command to configure the Simple Object Access Protocol Application Programming Interface (SOAP API) user/password.
4. Run the “**soapapi on**” command to enable the SOAP service on the APV appliance.

**Step2.** Enable Openstack LBaaS v1.0.

1. Navigate to “/etc/neutron/neutron.conf”, and set the “service\_plugins” section as follows:

```
service_plugins = router,lbaas
```

2. Navigate to “/etc/openstack-dashboard/local\_settings.py”, and set the “OPENSTACK\_NEUTRON\_NETWORK” section as follows:

```
OPENSTACK_NEUTRON_NETWORK = {
    'enable_lb': True,
    ...
}
```

3. Run the “**service**” command to restart the neutron server.

```
service neutron-server restart
```

**Step3.** Install the APV LBaaS v1.0 driver.

1. Copy the APV LBaaS v1.0 installation package to the Openstack controller node, and then run the “**tar**” command to decompress the package.



Openstack controller node runs management services needed for OpenStack to function.

```
tar xf lbaas.tgz
```

2. Navigate to the root directory of the decompressed package, and then run the “**python setup.py install**” command to install the driver.

**Step4.** Configure the APV LBaaS v1.0 driver.

1. Navigate to “/etc/neutron/neutron.conf” on the Openstack controller node, and set the “service\_provider” section to enable the load balancing function:

```
service_provider=LOADBALANCER:Arraynetworks APV: arraynetworks_neutron.service
s.loadbalancer.drivers.arraynetworks.plugin_driver.ApvPluginDriver
```

2. (Optional) Edit the “arraynetworks” section, or you can keep it to the default, as follows:

```
[arraynetworks]
config_dir = /etc/arraynetworks
scheduler = pool
```

3. Navigate to “/etc/arraynetworks” (default) on the controller node, create a “config.py” file, and add the following contents to it.

```
root@controller:~# cat /etc/arraynetworks/config.py
devices = {
    "APV-1": {
        "status": True,                #optional
        "protocol": "http",           #optional
        "host": "10.10.45.51",        #required
        "port": 9998,                 #optional
        "user": "apv1",               #required
        "passwd": "apv1-soap"         #required
    },

    "APV-2": {
        "host": "10.10.45.62",
        "user": "apv2",
        "passwd": "apv2-soap"
    },
}

Policies = [{
    "type": "pool",
    "enable": True,
    "policy": {
        "192.168.11.0/24": {
            "type": "device",          # device/cluster
            "id": "APV-1"
        },
        "192.168.12.0/24": "APV-2",   #if string, means
```

```

#device
"192.168.14.0/255.255.255.0": "APV-2",
}
},
{
"type": "pool",
"enable": False,
"policy": {
}
}
{
"type": "uuid",
"enable": True,
  "policy": {
    "937a02a61a3a4e1bb91a30420584e9ba": "APV-1",
    "1c55e74e4afc420a9a4f40c34b6cd053": "APV-2",
  },
}
}]

```

4. Run the “**service**” command to restart the neutron-server service.

```
service neutron-server restart
```

**Step5.** Configure the APV LBaaS cluster function.

The APV LBaaS v1.0 driver provides the inbuilt cluster function. You can configure this function in the file “/etc/arraynetworks/config.py”. The following is an example:

```

.....
clusters = {
  "cluster-1": {
    "enable": True, #optional
    "devices": [{
      "id": "APV-1", #required
      "priority": 100, #required
      "ifname": "port2"
    },
    {
      "id": "APV-2",
      "priority": 102,
      "ifname": "port2"
    }
  ], #required
  "vcid": 100, #required
}
}

```



**Note:** If any modifications are made to the cluster configurations, you must restart the neutron-server service. Otherwise, it will not take effect.

## Configure the LBaaS Service

There are two ways of provisioning LBaaS: Openstack Dashboard and CLI.

### ➤ Via the Dashboard

**Step1.** Access the Openstack Dashboard.

For example, enter “http://10.10.57.42/horizon” in a Web browser.

**Step2.** Navigate to the load balancer module.

Select **Project > Network > Load Balancers** from the main menu.

**Step3.** Create a pool.

In the **Pools** tab, click **Add Pool**, set the pool parameters, and then click **Add**. The following table lists the meanings of the pool parameters.

**Table 1–3 Pool Parameters**

Parameter	Meaning
Name	Specifies the pool name.
Description	Optional. Specifies descriptions of the configured pool.
Provider	Specifies the LBaaS service provider.
Subnet	Specifies the IP subnet used by the pool.
Protocol	Specifies the protocol type of the pool. Valid values are “http”, “https”, and “tcp”.
Load Balancing Method	Specifies the load balancing method. Valid values are “ROUND_ROBIN”, “LEAST_CONNECTIONS”, and “SOURCE_IP”.
Admin State	Specifies whether to enable the configured pool.

**Step4.** Add pool members.

In the **Members** tab, click **Add Member**, select the desired pool, set the member parameters, and click **Add**. The following table lists the meanings of the parameters for adding a pool member.

**Table 1–4 Member Parameters**

Parameter	Meaning
Pool	Specifies the pool to which members will be added.
Member Source	Specifies how the pool members are defined. Available options: <ul style="list-style-type: none"> <li>Select from active instances</li> <li>Specify the member IP address</li> </ul>

Parameter	Meaning
Member(s)	Specifies the pool member name. Available when <b>Member Source</b> is set to “Select from active instances”. One or more members can be selected by using the <b>Control</b> key.
Member address	Specifies the IP address of the member to be added. Available when <b>Member Source</b> is set to “Specify member IP address”.
Weight	Specifies the weight of the member. Valid values are 1 to 256. If multiple members are selected, the same weight will be applied, but it can be modified later.
Protocol Port	Specifies the protocol port for the pool member. Valid values are 1 to 65535. If multiple members are selected, the same port number will be applied, but it can be modified later.
Admin State	Specifies whether to enable the added member.

**Step5.** Add a VIP.

In the **Pools** tab, select the desired pool, and then select **Add VIP** from the **Edit Pool** dropdown list in the **Actions** column.

In the **Specify VIP** tab, set the VIP parameters. For the meanings of these parameters, see the following table.

**Table 1–5 VIP Parameters**

Parameter	Meaning
Name	Specifies the VIP name.
Description	Optional. Specifies descriptions of the VIP.
VIP Subnet	Specifies the IP subnet for the VIP.
Protocol Port	Specifies the protocol port for the VIP. Valid values are 1 to 65535.
Protocol	Specifies the protocol for the VIP. Valid values are “http”, “https”, and “tcp”.
Session Persistence	Specifies the session persistence type. Valid values are “SOURCE_IP”, “HTTP_COOKIE”, and “APP_COOKIE”.
Connection Limit	Specifies the maximum number of concurrent connections on a VIP. If connection limit is not required, set it to -1.
Admin State	Specifies whether to enable the VIP.

**Step6.** (Optional) Configure the health monitor for a pool.

In the **Monitors** tab, click **Add Monitor** and set the health monitor parameters. For the meanings of these parameters, see the following table.

**Table 1–6 Health Monitor Parameters**

Parameter	Meaning
Type	Specifies the monitor type. Valid values are “PING”, “TCP”, “HTTP”, and “HTTPS”.
Delay	Specifies the minimum interval between two health checks in

Parameter	Meaning
	seconds.
Timeout	Specifies the maximum wait time for a reply to health check request in seconds. <b>Note:</b> The timeout value cannot be larger than the interval value.
Max Retries	Specifies the number of consecutive health check failures for setting a pool member as Down. Valid values are 1 to 10.
HTTP Method	Optional. Specifies the HTTP method used by the health monitor. Available only when the monitor type is HTTP or HTTPS. Currently, only GET is valid.
URL	Optional. Specifies the URL address of the health check request. Available only when the monitor type is HTTP or HTTPS. It must consist of 1 to 128 characters beginning with “/”.
Expected HTTP Status Codes	Optional. Specifies the status code of the expected response to the health check request. Available only when the monitor type is HTTP or HTTPS. It supports a single value (e.g. 200), a list of values (e.g. 200, 202, 204), and a value range (e.g. 200-204).
Admin State	Specifies whether to enable the health monitor.

➤ **Via CLI**

1. Run the “**neutron lb-pool-create**” command to create the pool. Set the protocol type, subnet, provider and load balancing method. For example:

```
neutron lb-pool-create --name "pool-1" --protocol HTTP --subnet-id <your subnet UUID>
--provider "arraynetworks vapv" --lb-method ROUND_ROBIN
```

2. Run the “**neutron lb-member-create**” command to add members to the pool. Set the IP address and port number of each member. For example:

```
neutron lb-member-create --address 192.168.11.14 --protocol-port 80 pool-1
neutron lb-member-create --address 192.168.11.15 --protocol-port 80 pool-1
```

3. Run the “**neutron lb-vip-create**” command to add a VIP. Set the name, IP address, port number, protocol and subnet ID of the VIP, and then select a session persistence type. For example:

```
neutron lb-vip-create --name vip-1 --address 10.10.45.81 --protocol-port 8888 --protocol
HTTP --subnet-id <your VIP subnet UUID> pool-1 --session-persistence type=dict
type="APP_COOKIE",cookie_name="test"
```

# vAPV LBaaS v1.0 Configuration Guide

The vAPV LBaaS v1.0 driver also applies to the Openstack Juno and Kilo environments.

When the environment is ready, follow these guidelines to deploy the vAPV LBaaS v1.0 driver and configure the LBaaS service.

## Deploy the Driver

**Step1.** Enable Openstack LBaaS v1.0.

1. Locate the file “/etc/neutron/neutron.conf”, and set the “service\_plugins” section as follows:

```
service_plugins = router,lbaas
```

2. Navigate to “/etc/openstack-dashboard/local\_settings.py”, and set the “OPENSTACK\_NEUTRON\_NETWORK” section as follows:

```
OPENSTACK_NEUTRON_NETWORK = {
    'enable_lb': True,
    ...
}
```

3. Run the “**service**” command to restart the neutron server.

```
service neutron-server restart
```

**Step2.** Install the vAPV LBaaS v1.0 driver.

1. Copy the vAPV LBaaS v1.0 installation package to the Openstack controller node and each compute node, and then run the “**tar**” command to decompress the package.



The controller node runs management services needed for OpenStack to function.

The compute node receives requests from the controller node and hosts virtual machine instances. You can configure multiple compute nodes for service scalability.

```
tar xf lbaas.tgz
```

2. Navigate to the root directory of the decompressed package on the controller node, and run the “**python setup.py install --driver**” command to install the driver.
3. Navigate to the root directory of the decompressed package on each compute node, and run the “**python setup.py install --agent**” command to install the driver.
4. Run the “**service arraynetworks-lbaas-agent status**” command to verify that the service is running successfully.

```
* arraynetworks_lbaas_agent is running
```

**Step3.** Configure the vAPV LBaaS v1.0 driver.

1. Navigate to “/etc/neutron/neutron.conf” on the Openstack controller node, and set the “service\_provider” section to enable the load balancing function:

```
service_provider=LOADBALANCER:Arraynetworks vAPV:
arraynetworks_neutron.services.loadbalancer.drivers.arraynetworks.plugin_driver.ApvPlugin
Driver
```

2. Edit the “arraynetworks” section.

Set “vapv\_image” to the name of the vAPV image to be created, for example, vAPV-qga-VL.

Set “vapv\_license” to the license key obtained from Array Networks.

```
[arraynetworks]
vapv_image = vAPV-qga-VL
vapv_flavor = m1.medium
vapv_license = xxxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-
xxxxxxx-xxxxxxx-xxxxxxx
```

**Step4.** Load the vAPV image.

1. Access the Openstack dashboard. For example, enter “http://10.10.57.42/horizon” in a Web browser.

Choose **Admin > System > Images** from the main menu. Click **Create Image** and then set the image parameters, as instructed in the following table.

**Table 1–7 vAPV Image Parameters**

Parameter	Meaning
Name	Specifies the vAPV image name. It must be the same as the value of “vapv_image” pre-configured in the “arraynetworks” section of the file “/etc/neutron/neutron.conf” (Step3).
Description	Optional. You can add descriptions about the created vAPV image.
Image Source	Specifies where to load the vAPV image. You can load it from an HTTP URL address or a local image file.
Format	Specifies the format of the vAPV image.
Architecture	Optional. Specifies the CPU architecture that must be supported by the hypervisor. You can run the “ <b>uname -m</b> ” command to display the CPU architecture of a machine.
Minimum Disk (GB)	Optional. Specifies the minimum disk size required to boot the vAPV image. The default value is 0, indicating no limitation.
Minimum RAM (MB)	Optional. Specifies the minimum memory size required to boot the vAPV image. The default value is 0, indicating no limitation.
Public/Protected	Specifies whether to share the vAPV image with other tenants. Select the <b>Public</b> checkbox so that other tenants can also use the vAPV image. Otherwise, select the <b>Protected</b> checkbox.

- Run the “**glance image-update**” command to set the property of the vAPV image. Note that “vAPV-qga-VL” is the name of the vAPV image.

```
glance image-update --property hw_qemu_guest_agent=yes <vAPV-qga-VL>
```

- Run the “**service**” command to restart the neutron-server service.

```
service neutron-server restart
```

## Configure the LBaaS Service

There are two ways of provisioning LBaaS: Openstack dashboard and CLI.

### ➤ Via the Dashboard

**Step1.** Create a pool.

In the **Pools** tab, click **Add Pool**, set the pool parameters, and then click **Add**. The following table lists the meanings of the pool parameters.

**Table 1-8 Pool Parameters**

Parameter	Meaning
Name	Specifies the pool name.
Description	Optional. Specifies descriptions of the configured pool.
Provider	Specifies the LBaaS service provider.
Subnet	Specifies the IP subnet used by the pool.
Protocol	Specifies the protocol type of the pool. Valid values are “http”, “https”, and “tcp”.
Load Balancing Method	Specifies the load balancing method. Valid values are “ROUND_ROBIN”, “LEAST_CONNECTIONS”, and “SOURCE_IP”.
Admin State	Specifies whether to enable the configured pool.

**Step2.** Add pool members.

In the **Members** tab, click **Add Member**, select the desired pool, set the member parameters, and click **Add**. The following table lists the meanings of the parameters for adding a pool member.

**Table 1-9 Member Parameters**

Parameter	Meaning
Pool	Specifies the pool to which members will be added.
Member Source	Specifies how the pool members are defined. Available options: <ul style="list-style-type: none"> <li>Select from active instances</li> <li>Specify the member IP address</li> </ul>
Member(s)	Specifies the pool member name. Available when <b>Member Source</b> is set to “Select from active instances”. One or more members can be selected by using the <b>Control</b> key.

Parameter	Meaning
Member address	Specifies the IP address of the member to be added. Available when <b>Member Source</b> is set to “Specify member IP address”.
Weight	Specifies the weight of the pool member. Valid values are 1 to 256. If multiple members are selected, the same weight will be applied, but it can be modified later.
Protocol Port	Specifies the protocol port for the pool member. Valid values are 1 to 65535. If multiple members are selected, the same port number will be applied, but it can be modified later.
Admin State	Specifies whether to enable the added pool member.

**Step3.** Add a VIP.

In the **Pools** tab, select the desired pool, and then select **Add VIP** from the **Edit Pool** dropdown list in the **Actions** column.

In the **Specify VIP** tab, set the VIP parameters. For the meanings of these parameters, see the following table.

**Table 1–10 VIP Parameters**

Parameter	Meaning
Name	Specifies the VIP name.
Description	Optional. Specifies descriptions of the VIP.
VIP Subnet	Specifies the IP subnet for the VIP.
Protocol Port	Specifies the protocol port for the VIP. Valid values are 1 to 65535.
Protocol	Specifies the protocol for the VIP. Valid values are “http”, “https”, and “tcp”.
Session Persistence	Specifies the session persistence type. Valid values are “SOURCE_IP”, “HTTP_COOKIE”, and “APP_COOKIE”.
Connection Limit	Specifies the maximum number of concurrent connections on a VIP. If connection limit is not required, set it to -1.
Admin State	Specifies whether to enable the VIP.

**Step4.** (Optional) Configure the health monitor for a pool.

In the **Monitors** tab, click **Add Monitor** and set the health monitor parameters. For the meanings of these parameters, see the following table.

**Table 1–11 Health Monitor Parameters**

Parameter	Meaning
Type	Specifies the monitor type. Valid values are “PING”, “TCP”, “HTTP”, and “HTTPS”.
Delay	Specifies the minimum interval between two health checks in seconds.
Timeout	Specifies the maximum wait time for a reply to health check

Parameter	Meaning
	request in seconds. <b>Note:</b> The timeout value cannot be larger than the interval value.
Max Retries	Specifies the number of consecutive health check failures for setting a pool member as Down. Valid values are 1 to 10.
HTTP Method	Optional. Specifies the HTTP method used by the health monitor. Available only when the monitor type is HTTP or HTTPS. Currently, only GET is valid.
URL	Optional. Specifies the URL address of the health check request. Available only when the monitor type is HTTP or HTTPS. It must consist of 1 to 128 characters beginning with “/”.
Expected HTTP Status Codes	Optional. Specifies the status code of the expected response to the health check request. Available only when the monitor type is HTTP or HTTPS. It supports a single value (e.g. 200), a list of values (e.g. 200, 202, 204), and a value range (e.g. 200-204).
Admin State	Specifies whether to enable the health monitor.

➤ **Via CLI**

1. Run the “**neutron lb-pool-create**” command to create the pool. Set the protocol type, subnet, provider and load balancing method.

```
neutron lb-pool-create --name "pool-1" --protocol HTTP --subnet-id <your subnet UUID>
--provider "arraynetworks vapv" --lb-method ROUND_ROBIN
```

2. Run the “**neutron lb-member-create**” command to add members to the pool. Set the IP address and port number of each member.

```
neutron lb-member-create --address 192.168.11.14 --protocol-port 80 pool-1
neutron lb-member-create --address 192.168.11.15 --protocol-port 80 pool-1
```

3. Run the “**neutron lb-vip-create**” command to add a VIP. Set the name, IP address, port number, protocol and subnet ID of the VIP, and then select a session persistence type.

```
neutron lb-vip-create --name vip-1 --address 10.10.45.81 --protocol-port 8888 --protocol
HTTP --subnet-id <your VIP subnet UUID> pool-1 --session-persistence type=dict
type="APP_COOKIE",cookie_name="test"
```

# vAPV LBaaS v2.0 Configuration Guide

The vAPV LBaaS v2.0 driver applies to the Openstack Kilo devstack and Liberty environments.

When the environment is ready, follow these guidelines to deploy the vAPV LBaaS v2.0 driver and configure the LBaaS service.

## Deploy the Driver

### Step1. Enable Openstack LBaaS v2.0.

1. Navigate to “/etc/neutron/neutron.conf”, and set the “service\_plugins” section as follows:

```
service_plugins = neutron.services.l3_router.l3_router_plugin.L3RouterPlugin,neutron_lbaas.services.loadbalancer.plugin.LoadBalancerPluginv2
```

2. Run the “service” command to restart the neutron server.

```
service neutron-server restart
```

### Step2. Install the vAPV LBaaS v2.0 driver.

1. Copy the vAPV LBaaS v2.0 installation package to the Openstack controller node, and each compute node, and then run the “tar” command to decompress the package.



The controller node runs management services needed for OpenStack to function.

The compute node receives requests from the controller node and hosts virtual machine instances. You can configure multiple compute nodes for service scalability.

```
tar xf lbaas_v2.tgz
```

2. Navigate to the root directory of the decompressed package on the controller node, and run the “python setup.py install --driver” command to install the driver.
3. Navigate to the root directory of the decompressed package on each compute node, and run the “python setup.py install --agent” command to install the driver.
4. Run the “service arraynetworks-lbaas-agent status” command to verify that the service is running successfully.

```
* arraynetworks_lbaas_agent is running
```

### Step3. Configure the vAPV LBaaS v2.0 driver.

1. Navigate to “/etc/neutron/neutron\_lbaas.conf” on the Openstack controller node, and set the “service\_provider” section to enable the load balancing function:

```
service_provider=LOADBALANCER:Arraynetworks vAPV: arraynetworks_neutron.services.loadbalancer.drivers.arraynetworks.plugin_driver.ApvPluginDriver
```

2. Edit the “arraynetworks” section.

Set “vapv\_image” to the name of the vAPV image to be created, for example, vAPV-qga-VL.

Set “vapv\_license” to the license key obtained from Array Networks.

```
[arraynetworks]
vapv_image = vAPV-qga-VL
vapv_flavor = m1.medium
vapv_license = xxxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-
xxxxxxx-xxxxxxx-xxxxxxx
```

**Step4.** Load the vAPV image.

1. Access the Openstack dashboard via a Web browser. For example, enter “http://10.10.57.42/horizon” in the address bar.

Choose **Admin > System > Images** from the main menu. Click **Create Image** and then set the image parameters. Table 1-12 shows the meanings of the parameters.

**Table 1–12 vAPV Image Parameters**

Parameter	Meaning
Name	Specifies the vAPV image name. It must be the same as the value of “vapv_image” pre-configured in the “arraynetworks” section of the file “/etc/neutron/neutron.conf” (Step3).
Description	Optional. Specifies vAPV image descriptions.
Image Source	Specifies where to load the vAPV image. You can load it from an HTTP URL address or a local image file.
Format	Specifies the format of the vAPV image.
Architecture	Optional. Specifies the CPU architecture that must be supported by the hypervisor. You can run the “uname -m” command to display the CPU architecture of a machine.
Minimum Disk (GB)	Optional. Specifies the minimum disk size required to boot the vAPV image. The default value is 0, indicating no limitation.
Minimum RAM (MB)	Optional. Specifies the minimum memory size required to boot the vAPV image. The default value is 0, indicating no limitation.
Public/Protected	Specifies whether to share the vAPV image with other tenants. Select the <b>Public</b> checkbox so that other tenants can also use the vAPV image. Otherwise, select the <b>Protected</b> checkbox.

2. Run the “**glance image-update**” command to set the property of the vAPV image. Note that “vAPV-qga-VL” is the name of the vAPV image.

```
glance image-update --property hw_qemu_guest_agent=yes <vAPV-qga-VL>
```

3. Run the “**service**” command to restart the neutron-server service.

```
service neutron-server restart
```

## Configure the LBaaS Service

Openstack LBaaS v2.0 supports only the CLI way to configure the LBaaS service.

1. Run the “**neutron lbaas-loadbalancer-create**” command to set the LBaaS service provider.

```
neutron lbaas-loadbalancer-create --name lb2 --provider "ArrayNetworks vAPV" ext-subnet
```

2. Run the “**neutron lbaas-listener-create**” command to create the VIP.

```
neutron lbaas-listener-create --loadbalancer lb2 --protocol HTTP --protocol-port 80 --name listener2
```

3. Run the “**neutron lbaas-pool-create**” command to create the pool.

```
neutron lbaas-pool-create --lb-algorithm ROUND_ROBIN --listener listener2 --protocol HTTP --name pool2
```

4. Run the “**neutron lbaas-member-create**” command to add members to the pool.

```
neutron lbaas-member-create --subnet provider_net --address 192.168.11.123 --protocol-port 80 pool2
neutron lbaas-member-create --subnet provider_net --address 192.168.11.133 --protocol-port 80 pool2
```

5. Run the “**neutron lbaas-healthmonitor-create**” command to configure a health monitor for the pool.

```
neutron lbaas-healthmonitor-create --expected-codes '200' --http-method get --url-path '/hc' --max-retries 10 --timeout 3 --type HTTP --delay 10 --pool pool2
```