# **Introduction**

Sometime we need to have more than one external network for our environment and very often we realized it when our cloud already deployed. Fortunately, it is not so complicated to add a couple of new external networks.

If you search this topic in Internet you, of course, will find a several pretty good posts and articles how to do it:

* <http://www.ajo.es/post/86497974174/using-multiple-external-networks-in-openstack>
* <http://blog.oddbit.com/2014/05/28/multiple-external-networks-wit/>

But usually you will spent some time on investigation and testing it on some test environment and face a couple of tricks, since, as I mentioned before, usually cloud is already deployed and has some workload.

So, this article has 3 purposes:

1. Gather all information in one place
2. Adjust it for MOS cloud
3. Share a script which can help you to create additional external networks more easily.

# **MOS default configuration**

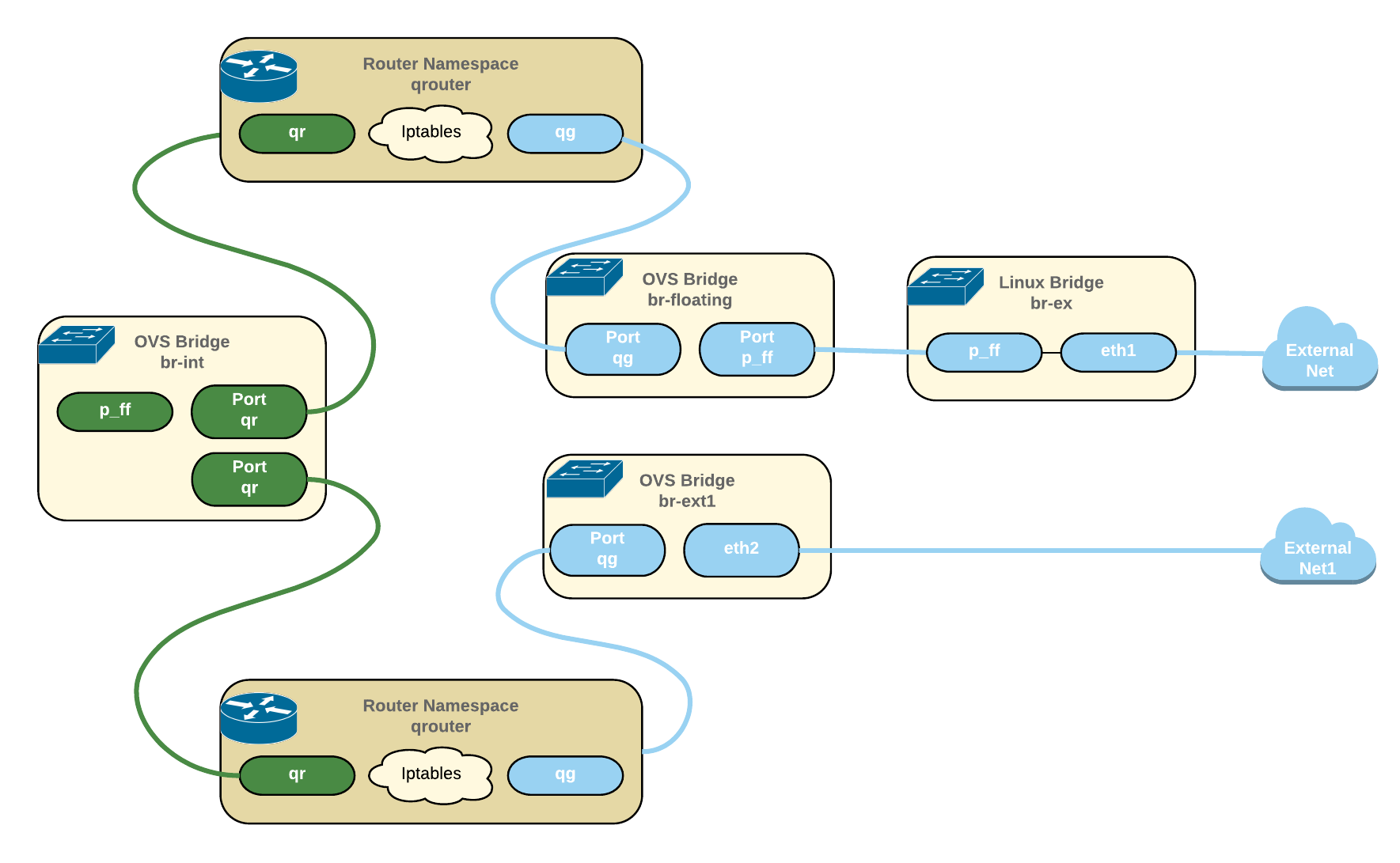
By default MOS cloud has **only** one external network which is connected to **br-floating** (in earlier versions br-ex).

# **Neutron with only external network - New Page.png**

And it is configured in /etc/neutron/l3-agent.ini file which contains the following lines:

|  |
| --- |
| # Name of bridge used for external network traffic. This should be set to  # empty value for the linux bridge. when this parameter is set, each L3 agent  # can be associated with no more than one external network.  # external\_network\_bridge = br-ex  **external\_network\_bridge = br-floating** |

The network infrastructure with 2 external networks might be looked like this:



As you can see the additional external network **ext1** is directly attached to the the dedicated OVS bridge **br-ext1**. In default FUEL architecture additional Linux bridge is used for avoiding any problem with ovs-bonding and some issues with old kernels. If you faced some problem with it you can also use the default FUEL schema and add additional Linux bridge before OVS bridge **br-ext1.**

# **Step by step manual**

## **1. Prepare cloud for multiple external networks**

These small 3 sub-steps help just reconfigured cloud to allow you add more external networks. The 2 first steps should be performed on all nodes, the 3rd step will be enough to perform just on any controller.

After performing them you can restart neutron services (**neutron-server**, **neutron-l3-agent**, **neutron-plugin-openvswitch-agent**) and you will have the same situation that was before with only different that you can add new external networks as more as you want.

### **1.1 etc/neutron/l3-agent.ini**

For this we should make a couple of changes in **/etc/neutron/l3\_agent.ini** configuration file. These 2 parameters should be configured as empty:

**gateway\_external\_network\_id =**

**external\_network\_bridge =**

|  |
| --- |
| .........  # When external\_network\_bridge is set, each L3 agent can be associated  # with no more than one external network. This value should be set to the UUID  # of that external network. To allow L3 agent support multiple external  # networks, both the external\_network\_bridge and gateway\_external\_network\_id  # must be left empty.  # gateway\_external\_network\_id =  **gateway\_external\_network\_id =**  .........  # Name of bridge used for external network traffic. This should be set to  # empty value for the linux bridge. when this parameter is set, each L3 agent  # can be associated with no more than one external network.  # external\_network\_bridge = br-ex  **external\_network\_bridge =**  ......... |

### 

### **1.2 /etc/neutron/plugin.ini**

Since we cleared **external\_network\_bgidge** in previous step we should create a new physical network for our current external network **net04\_ext** and specify it in bridge mapping.

|  |
| --- |
| [ovs]  .........  bridge\_mappings=physnet2:br-prv,**physnet-ex:br-floating** |

### **1.3 Update or recreate the current external network**

This step might be performed once on any controller.

Then we should update configuration of the current external network. This can be done by 2 ways:

1. Recreate this external network;
2. Update a couple options in DB for this network;

I guess it will be more convenient to use the 2 way, since usually we already have some workload on the cloud and network recreating looks more complicated.

So, we should update 2 options:

**provider:network\_type**

**provider:physical\_network**

|  |
| --- |
| root@node-1:~# neutron net-show **net04\_ext**  +---------------------------+--------------------------------------+  | Field | Value |  +---------------------------+--------------------------------------+  | admin\_state\_up | True |  | id | dca98cc7-d74e-4c0a-b903-db202472733e |  | mtu | 0 |  | name | net04\_ext |  | **provider:network\_type** | **local**  |  | **provider:physical\_network** | |  | provider:segmentation\_id | |  | router:external | True |  | shared | False |  | status | ACTIVE |  | subnets | 0dd9ef0e-3ddf-44fa-8e75-c71c45fed555 |  | tenant\_id | 0a28b0455dd744a5885a8f92d240528d |  +---------------------------+--------------------------------------+  root@node-1:~# mysql neutron -e "select network\_id, network\_type, physical\_network from ml2\_network\_segments;"  +--------------------------------------+--------------+------------------+  | network\_id | network\_type | physical\_network |  +--------------------------------------+--------------+------------------+  | 02439fb5-23f3-4ca1-809a-b6e03278cc7b | vlan | physnet2 |  | ac6a599a-9c61-404b-b722-ffd8043d7acb | vlan | physnet2 |  | dca98cc7-d74e-4c0a-b903-db202472733e | **local** | **NULL** |  +--------------------------------------+--------------+------------------+  root@node-1:~# mysql neutron -e "update ml2\_network\_segments set network\_type='**flat**', physical\_network='**physnet-ex**' where network\_type='**local**';"  root@node-1:~# mysql neutron -e "select network\_id, network\_type, physical\_network from ml2\_network\_segments;"  +--------------------------------------+--------------+------------------+  | network\_id | network\_type | physical\_network |  +--------------------------------------+--------------+------------------+  | 02439fb5-23f3-4ca1-809a-b6e03278cc7b | vlan | physnet2 |  | ac6a599a-9c61-404b-b722-ffd8043d7acb | vlan | physnet2 |  | dca98cc7-d74e-4c0a-b903-db202472733e | **flat** | **physnet-ex**  |  +--------------------------------------+--------------+------------------+ |

## **2. Create an infra for new external network**

Now our cloud is ready for adding a new external network.

### **2.1 Create dedicated OVS bridge**

First of all we need to create a OVS bridge

|  |
| --- |
| root@node-1:~# **ovs-vsctl add-br br-ext1** |

### **2.2 Add external interface to the dedicated OVS bridge**

|  |
| --- |
| root@node-1:~# **ovs-vsctl add-port br-ext1 eth2** |

### **2.3 Add new physnet into /etc/neutron/plugin.ini**

|  |
| --- |
| .........  # (IntOpt) Segment MTU. The maximum permissible size of an  # unfragmented packet travelling a L2 network segment. If <=0,  # the segment MTU is indeterminate and no calculation takes place.  # segment\_mtu = 0  # (ListOpt) Physical network MTUs. List of mappings of physical  # network to MTU value. The format of the mapping is  # <physnet>:<mtu val>. This mapping allows specifying a  # physical network MTU value that differs from the default  # segment\_mtu value.  # physical\_network\_mtus =  physical\_network\_mtus =physnet2:1500**,physnet-ext1:1500**  # Example: physical\_network\_mtus = physnet1:1550, physnet2:1500  .........  [ovs]  enable\_tunneling=False  integration\_bridge=br-int  bridge\_mappings=physnet2:br-prv,physnet-ex:br-floating,**physnet-ext1:br-ext1** |

If you want to add more than one external network you should repeat this step for new each external network.

## **3. Restart Neutron services**

Then we need to restart the following Neutron services:

1. Neutron-server *(on controllers)*
2. Neutron-plugin-openvswitch-agent *(on controllers and computes)*
3. neutron-l3-agent/neutron-vpn-agent *(on controllers and if you use DVR on compute nodes as well)*

Please keep in mind that services which are under pacemaker should be restarted using pacemaker CLI utility.

## **4. Create new external network and subnetwork**

Now we are ready to create our new external network:

|  |
| --- |
| root@node-1:~# **neutron net-create net-ext1 --provider:network\_type flat --provider:physical\_network physnet-ext1 --router:external** |

and then subnet:

|  |
| --- |
| root@node-1:~#  **neutron subnet-create net-ext1 ${CIDR} --name subnet-ext1 --gateway ${GATEWAY} --enable\_dhcp=False --allocation-pool start=${FIP\_START},end=${FIP\_END}** |

If you add more than one external networks and repeated the 2nd step N time you should repeat this step N times as well.

# **Testing**

Now you can test and use the all your external networks.

# **Automation script**

Also I’ve implemented a set of scripts which can help you to add external networks more easily:

<https://github.com/aepifanov/openstack-add-network>

Enjoy!

P.S.

Please, don’t hesitate to contribute to this repo and let me know if you will face any issue with this script.

[aepifanov@mirantis.com](mailto:aepifanov@mirantis.com)