

Advantech Remote Evaluation Service Portal Tutorial

Contents

1	System Requirements (only for command line)	3
1.1	On Linux	3
1.2	On Windows	4
1.2.1	Installation	4
2	Login and download OpenStack RC file	4
2.1	Login	4
2.2	Download RC file (needed for command line) and set environment variables	4
2.2.1	Set environment variables using the OpenStack RC file (Linux)	4
2.2.2	Set environment variables using the OpenStack RC file (Windows)	5
3	Add an image	6
3.1	Using GUI	6
3.2	Using cli	7
3.3	Share an image with an other project	8
4	Create a network	8
4.1	Using GUI	8
4.2	Using cli	9
4.3	External network	10
4.3.1	Creation	10
4.3.2	DNS resolution	10
4.3.3	Floating IP	11
5	Create a router	12
5.1	Using GUI	12
5.2	Using cli	12
6	Adding networks to router	13
6.1	Add a standard network	13
6.1.1	Using GUI	13
6.1.2	Using cli	13
6.2	Add an external network	13
6.2.1	Using GUI	13
6.2.2	Using cli	14
7	Launch a VM	14
7.1	Using GUI	14
7.2	Using cli	16
7.3	Cloud-init	18
7.3.1	GUI	18
7.3.2	cli	19

7.4	Specify an availability zone	19
7.4.1	GUI	19
7.4.2	Cli	20
8	Multiqueues	21
8.1	For Virtio Linux driver	21
8.1.1	Enabling multiqueue manually	21
8.1.2	Boot VM with several queues	22

The Advantech platform comprises a flexible OpenStack environment/platform for the construction of test scenarios. This file describes tools the user needs to control and create test scenarios using the OpenStack platform GUI or command line.

In addition, several images are already uploaded in the OpenStack platform for the purpose of creating virtual machines (VMs):

- Fedora 20 template (no cloud-init)
- Fedora 20 cloud image (with cloud-init)
- Cirros 0.3.3
- Ubuntu Cloud (with cloud-init)
- Ubuntu template (no cloud-init)

Note: the user for template image is root and the password is vbreizh.

On specific request to 6WIND support, the following images can be enabled:

- 6WIND Turbo IPsec
- 6WIND Turbo Router

1 System Requirements (only for command line)

In order to execute OpenStack command line, client software must be installed on your PC. The client packages are python-novaclient, python-neutronclient, python-glanceclient (to manage image).

1.1 On Linux

On Ubuntu/Debian 8 distributions

```
sudo apt-get install python-novaclient python-neutronclient python-glanceclient
```

Debian 7 doesn't have this packages on its repository, but they can be installed using pip:

```
sudo apt-get install python-pip python-dev
sudo pip install python-novaclient
sudo pip install python-neutronclient
sudo pip install python-glanceclient
```

On RedHat, CentOS, Fedora distributions

```
sudo yum install python-novaclient python-neutronclient python-glanceclient
```

1.2 On Windows

Reference:

http://docs.openstack.org/user-guide/common/cli_install_openstack_command_line_clients.html

1.2.1 Installation

Install Python 2.7 or later. Currently, the clients do not support Python 3.

Go to your Control Panel\System and Security\System. Go into Advanced System Settings. Click on 'Environment Variables' under the Advanced tab add ;C:\Python27;C:\Python27\Scripts to the end of the PATH variable in system variables.

Install setuptools: <https://pypi.python.org/pypi/setuptools> the easy way is to download ez_setup.py using your preferred web browser or other technique and "run" that file.

Then, open a console windows (cmd.exe) and do the following command:

```
easy_install pip
pip install python-novaclient
pip install python-neutronclient
pip install python-glanceclient
```

2 Login and download OpenStack RC file

2.1 Login

You are now ready to access the 6WIND Accelerated Virtual Environment. Point your browser to <http://6wind.testdrive-advantech-nfv.com> and login to your account. If you do not have an account, please register. Once approved, an email with account access information is mailed. Use the credentials in the email to login at <http://6wind.testdrive-advantech-nfv.com>. It will allow the forwarding of your IP to OpenStack services.

On successful login, you will land in the 'Overview' tab. Select the "VNF Test Drive" tab. A set of OpenStack services are shown. Access the Horizon service dashboard by selecting the "Go!" button or select the service endpoint associated with the Horizon service (at <http://6wind.testdrive-advantech-nfv.com:888/dashboard/>). A new window opens and requests login credentials. Use the same username and password.

2.2 Download RC file (needed for command line) and set environment variables

To use command line with nova, neutron or glance, you need to download the rc file. At the dashboard select Project/Compute/Access & Security/API Access. Then select "Download OpenStack RC File".

2.2.1 Set environment variables using the OpenStack RC file (Linux)

For Linux, source the file you just downloaded with the command

```
$ source "$Projectname-openrc.sh"
```

\$Projectname-openrc.sh is the name of the downloaded file, check its name. When prompted for a password, use the password used to logon to <http://6wind.testdrive-advantech-nfv.com> For Windows, you need to create environmental variables

2.2.2 Set environment variables using the OpenStack RC file (Windows)

After this installation you need to set up environment variable, go to your Control Panel\System and Security\System. Go into Advanced System Settings. Click on 'Environment Variables' under the Advanced tab. Click New in User variable to create the environmental variables.

You need to download the OpenStackrc file (explained on the next section) For each variable export on the openrc.sh file you need to create environmental variables. For example with the following rc file,

```
#!/bin/bash

# To use an Openstack cloud you need to authenticate against keystone, which
# returns a **Token** and **Service Catalog**. The catalog contains the
# endpoint for all services the user/tenant has access to - including nova,
# glance, keystone, swift.
#
# *NOTE*: Using the 2.0 *auth api* does not mean that compute api is 2.0. We
# will use the 1.1 *compute api*
export OS_AUTH_URL=http://6wind.testdrive-advantech-nfv.com:5000/v2.0

# With the addition of Keystone we have standardized on the term **tenant**
# as the entity that owns the resources.
export OS_TENANT_ID=9acf7800e85f4dea862ef2424258ad26
export OS_TENANT_NAME="demo"

# In addition to the owning entity (tenant), openstack stores the entity
# performing the action as the **user**.
export OS_USERNAME="demo"

# With Keystone you pass the keystone password.
echo "Please enter your OpenStack Password: "
read -sr OS_PASSWORD_INPUT
export OS_PASSWORD=$OS_PASSWORD_INPUT

# If your configuration has multiple regions, we set that information here.
# OS_REGION_NAME is optional and only valid in certain environments.
export OS_REGION_NAME="RegionOne"
# Don't leave a blank variable, unset it if it was empty
if [ -z "$OS_REGION_NAME" ]; then unset OS_REGION_NAME; fi
```

You must create 6 variables:

named OS_AUTH_URL value : <http://6wind.testdrive-advantech-nfv.com:5000/v2.0>

named OS_TENANT_ID value : 9acf7800e85f4dea862ef2424258ad26

named OS_TENANT_NAME value : demo

named OS_USERNAME value : demo

named OS_PASSWORD value : demo

named OS_REGION_NAME value : RegionOne

Or in cmd.exe you can do

```
set OS_AUTH_URL=http://6wind.testdrive-advantech-nfv.com:5000/v2.0
```

```
set OS_TENANT_ID=9acf7800e85f4dea862ef2424258ad26
```

```
set OS_TENANT_NAME=demo
```

```
set OS_USERNAME=demo
```

set OS_PASSWORD=demo

set OS_REGION_NAME=RegionOne

Double quotes can cause issues when there is no space in the content, you can try without it.

After this, you can use the command normally (nova list, neutron net-list, etc...)

3 Add an image

3.1 Using GUI

In the dashboard, go to Project/Compute/Images, and select "+ Create Image"

Fill the form (name, source, location and format are mandatory)

[×](#)

Create An Image

Name *

Description

Image Source

Image Location ?

Format *

Architecture

Minimum Disk (GB) ?

Minimum RAM (MB) ?

Public

Protected

Description:

Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

3.2 Using cli

```
$ glance image-create --name 'Fedora 20 cloud' --disk-format qcow2 \  
--container-format bare \  
--copy-from http://cloud.fedoraproject.org/fedora-20.x86_64.qcow2 --progress  
  
$ glance image-create --name "Cirros 0.3.3" \  
--file /tmp/images/cirros-0.3.3-x86_64-disk.img \  
--disk-format qcow2 --container-format bare --progress
```

Attention: if Public box is checked or the option `--is-public True` all other projects will see your image. You can also add `--is-protected {True,False}` which prevent image from being deleted.

Use the "glance image-list" command to verify the added images (you may see more than two images due to the 'public' option).

```
$ glance image-list  
+-----+-----+-----+-----+-----+-----+  
| ID | Name | Disk Format | Container Format | Size | Status |  
+-----+-----+-----+-----+-----+-----+  
| 6e68a12f-66c4-42a7-984c-71f7675e9c68 | Cirros 0.3.3 | qcow2 | bare | 13200896 | active |  
| da0e551d-aa61-4cc8-8399-4c0eda521fcc | Fedora 20 cloud | qcow2 | bare | 210829312 | active |  
+-----+-----+-----+-----+-----+-----+
```

Disk Format

The disk format of a virtual machine image is the format of the underlying disk image. Virtual appliance vendors have different formats for laying out the information contained in a virtual machine disk image. You can set your image's disk format to one of the following:

- raw: This is an unstructured disk image format
- vhd: This is the VHD disk format, a common disk format used by virtual machine monitors from VMware, Xen, Microsoft, VirtualBox, and others
- vmdk: Another common disk format supported by many common virtual machine monitors
- vdi: A disk format supported by VirtualBox virtual machine monitor and the QEMU emulator
- iso: An archive format for the data contents of an optical disc (e.g. CDROM).
- qcow2: A disk format supported by the QEMU emulator that can expand dynamically and supports Copy on Write
- aki: This indicates what is stored in Glance is an Amazon kernel image
- ari: This indicates what is stored in Glance is an Amazon ramdisk image
- ami: This indicates what is stored in Glance is an Amazon machine image

Container Format

The container format refers to whether the virtual machine image is in a file format that also contains metadata about the actual virtual machine. Note that the container format string is not currently used by Glance or other OpenStack components, so it is safe to simply specify bare as the container format if you are unsure. You can set your image's container format to one of the following:

- bare: This indicates there is no container or metadata envelope for the image
- ovf: This is the OVF container format
- aki: This indicates what is stored in Glance is an Amazon kernel image
- ari: This indicates what is stored in Glance is an Amazon ramdisk image
- ami: This indicates what is stored in Glance is an Amazon machine image
- ova: This indicates what is stored in Glance is an OVA tar archive file

(src: <http://docs.openstack.org/developer/glance/formats.html>)

3.3 Share an image with an other project

When we add image via *glance image-create* or Horizon, the image can either be visible by all the project, either be private and only visible by the project itself.

But we can share the image with others projects which will be able to see and use it. For this, we need to use the command *glance member-create*, we need to specify the id of the image (can be list with *glance image-list*) and the id of the other project (also called tenant, can be view on Identity tab in Horizon or on the openStack RC file on the variable OS_TENANT_ID)

```
glance member-create [--can-share] image_ID project_ID
```

The option `--can-share` allow the specified tenant to share this image.

4 Create a network

4.1 Using GUI

Go into Project/Network/Networks, select "+ Create Network". Fill the "Network Name" field then Next, fill the "Subnet Name " and "Network Address", then Next and Create

x

Create Network

Network
Subnet
Subnet Detail

Network Name

Admin State * ?

UP
▼

Create a new network. In addition a subnet associated with the network can be created in the next panel.

« Back

Next »

Create Network



Network * Subnet * Subnet Detail

Create Subnet

Subnet Name

Network Address ?

IP Version *

Gateway IP ?

Disable Gateway

Create a subnet associated with the new network, in which case "Network Address" must be specified. If you wish to create a network without a subnet, uncheck the "Create Subnet" checkbox.

« Back Next »

4.2 Using cli

neutron net-create nameNetwork neutron subnet-create nameSubnetNetwork a.b.c.d/X

```
$ neutron net-create net_test
Created a new network:
+-----+-----+
| Field          | Value                               |
+-----+-----+
| admin_state_up | True                                |
| id             | 905dd0a6-b7c8-4041-81a1-3fd0d8df842f |
| name          | net_test                            |
| router:external | False                               |
| shared        | False                               |
| status        | ACTIVE                              |
| subnets      |                                       |
| tenant_id     | 8ce7f2e52136404eb691bf01dc472268   |
+-----+-----+

$ neutron subnet-create --name subnet_test net_test 10.0.0.0/24
Created a new subnet:
+-----+-----+
| Field          | Value                               |
+-----+-----+
```

```

+-----+-----+
| allocation_pools | {"start": "10.0.0.2", "end": "10.0.0.254"} |
| cidr              | 10.0.0.0/24                               |
| dns_nameservers  |                                             |
| enable_dhcp      | True                                       |
| gateway_ip       | 10.0.0.1                                   |
| host_routes      |                                             |
| id               | 3ae3ee8d-76f0-497d-b9df-8d0ea6a21094    |
| ip_version       | 4                                          |
| ipv6_address_mode |                                             |
| ipv6_ra_mode     |                                             |
| name             | subnet_test                               |
| network_id       | 905dd0a6-b7c8-4041-81a1-3fd0d8df842f    |
| tenant_id        | 8ce7f2e52136404eb691bf01dc472268      |
+-----+-----+

```

You can list the network

```

$ neutron net-list
+-----+-----+-----+
| id | name | subnets |
+-----+-----+-----+
| 905dd0a6-b7c8-4041-81a1-3fd0d8df842f | net_test | 3ae3ee8d-76f0-497d-b9df-8d0ea6a21094 10.0.0.0/24 |
+-----+-----+-----+

$ neutron subnet-list
+-----+-----+-----+-----+
| id | name | cidr | allocation_pools |
+-----+-----+-----+-----+
| 3ae3ee8d-76f0-497d-b9df-8d0ea6a21094 | subnet_test | 10.0.0.0/24 | {"start": "10.0.0.2", "end": "10.0.0.254"} |
+-----+-----+-----+-----+

```

4.3 External network

4.3.1 Creation

To provide internet access to the VMs, an external network must be created.

It's only possible using command line.

The following network is already created in your OpenStack project. The commands are shown for reference;

DO NOT EXECUTE.

```

$ # for OVS
$ neutron net-create publicNetwork -- --router:external=True
$ # for LinuxBridge
$ neutron net-create public --provider:physical_network physpublic \
  --provider:network_type flat --router:external=True

$ neutron subnet-create --gateway 10.168.215.10 --disable-dhcp \
  --allocation-pool start=10.168.215.11,end=10.168.215.100 \
  --name publicNetwork_subnet publicNetwork 10.168.215.0/24

```

4.3.2 DNS resolution

To resolve DNS requests provide the IP address of the nameserver to the network where the VMs reside : Add the option `--dns_nameservers list=true 8.8.8.7 8.8.8.8` at the end of command line when creating the subnet

```
neutron subnet-create --name subnet_test net_test 10.0.0.0/24 \  
--dns_nameservers list=true 8.8.8.7 8.8.8.8
```

Or you can update an existing subnet:

```
neutron subnet-update subnet_test --dns_nameservers list=true 8.8.8.7 8.8.8.8
```

To clear this value,

```
neutron subnet-update subnet_test --dns_nameservers action=clear
```

If the VM is already launched you can modify the file `/etc/resolv.conf` on the VM itself, for example adding the nameserver at 8.8.8.8

```
echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

4.3.3 Floating IP

Attention, floating IPs are associated with a 'launched' VM. Please see section 7 Launching a VM for more information.

You must assign a floating IP to the VMs if you plan to access the VM using ssh/http/https. After setting the floating IP you must logout and re-login to the portal <http://6wind.testdrive-advantech-nfv.com/> to assure connection is possible.

Using GUI

On Compute / Instances, click on the arrow on the Action column, select "Associate Floating IP".

If no IPs are available on "IP Address", click on the plus (+) button, then select the public network and create the floating IP.

In "Manage Floating IP Associations", select the port and click "Associate".

×

Manage Floating IP Associations

IP Address *

IP Address *

Select an IP address ▼ +

Port to be associated *

No ports available ▼

Select the IP address you wish to associate with the selected instance.

Cancel Associate

Using cli

If no floating IPs are available, create a new one.

```
neutron floatingip-create publicNetwork
```

To list the floating IPs and the ports

```
neutron floatingip-list  
neutron port-list
```

Associate the floating IP to the port

```
neutron floatingip-associate <floatingip_id> <port_id>
```

5 Create a router

A router is not mandatory when there is only a single network

5.1 Using GUI

Go in Project / Network / Routers, click on Create Router.

✕

Create Router

Router Name *

5.2 Using cli

```
neutron router-create routerName
```

```
$ neutron router-create router_test  
Created a new router:  
+-----+-----+  
| Field           | Value                               |  
+-----+-----+  
| admin_state_up  | True                                 |  
| external_gateway_info |                                     |  
| id              | b010e860-7d3d-4824-849c-61e21087d714 |  
| name            | router_test                          |  
| routes          |                                       |  
| status          | ACTIVE                               |  
| tenant_id       | 8ce7f2e52136404eb691bf01dc472268    |  
+-----+-----+
```

6 Adding networks to router

6.1 Add a standard network

6.1.1 Using GUI

Go in Project / Network / Routers, click on the router name then "Add Interface" and select the network

✕

Add Interface

Subnet *

IP Address (optional) ?

Router Name *

Router ID *

Description:

You can connect a specified subnet to the router.

The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.

6.1.2 Using cli

```
$ neutron router-interface-add router_test subnet_test
Added interface 457c3894-73c8-4f61-aea7-65158a91dd4e to router router_test.
```

6.2 Add an external network

In order to connect through an external network, you need to connect a gateway.

6.2.1 Using GUI

Go in Project / Network / Routers, in Actions column click on "Set Gateway" and select your external network

Set Gateway x

External Network *

public

Description:

You can connect a specified external network to the router. The external network is regarded as a default route of the router and the router acts as a gateway for external connectivity.

Router Name *

router_test

Router ID *

ff835279-9fb3-4de1-a255-2198f0fe7de0

Cancel

Set Gateway

6.2.2 Using cli

```
$ neutron router-gateway-set router_test publicNetwork
Set gateway for router router_test
```

7 Launch a VM

7.1 Using GUI

In Project / Compute / Instances, click on "Launch Instance" Fill the name, choose the flavor and source. If several networks have been created, select the network(s) in the Networking tab.

Note: When spawning a VM on a node with 6WIND Virtual Accelerator installed, select a flavor configured with hugepages attribute. Flavors configured with hugepages have names ending in "_hugepages". for example, the flavors `nodisk.xlarge_hugepages`, `nodisk.xlarge_8GB_hugepages`, `nodisk.3cores_hugepages` and `nodisk.tiny_hugepages` are configured with the hugepages attribute. If a hugepage flavor is not used, the VM interfaces will be managed by linux (the physical interface is still managed by VA) and performance will be sub-optimal.

Launch Instance



Details *

Access & Security *

Networking *

Post-Creation

Advanced Options

Availability Zone

Any Availability Zone ▼

Instance Name *

Test

Flavor *

m1.tiny ▼

Instance Count *

1

Instance Boot Source *

Boot from image ▼

Image Name

Fedora 20 cloud (201.1 MB) ▼

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

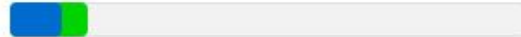
Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

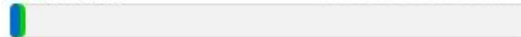
Number of Instances 2 of 10 Used



Number of VCPUs 2 of 20 Used



Total RAM 1,024 of 51,200 MB Used



Cancel

Launch

Launch Instance

The screenshot shows the 'Launch Instance' dialog box with the 'Networking' tab selected. The 'Selected networks' section contains one entry: 'NIC:1 test (f1c3d57e-1cbb-42f0e65-d74e551aed3f)'. The 'Available networks' section is empty. A text box on the right explains: 'Choose network from Available networks to Selected networks by push button or drag and drop, you may change NIC order by drag and drop as well.' At the bottom right, there are 'Cancel' and 'Launch' buttons.

For cloud image, see cloud-init section to set up a password. (the Post-creation will be used)

Launch Instance

The screenshot shows the 'Launch Instance' dialog box with the 'Post-Creation' tab selected. The 'Customization Script Source' dropdown is set to 'Direct Input'. The 'Script Data' field contains the following text:

```
#cloud-config
password: demo
chpasswd: {expire: False}
sudo: [ALL=(ALL) NOPASSWD:ALL]
groups: sudo
shell: /bin/bash
```

 A text box on the right explains: 'You can customize your instance after it has launched using the options available here. "Customization Script" is analogous to "User Data" in other systems.' At the bottom right, there are 'Cancel' and 'Launch' buttons.

Then click on "Launch"

7.2 Using cli

To launch a VM we use the command `nova boot --flavor XX --image XX --nic net-id=$NET_ID --security-group default $NAME_VM`

If you don't know the available images, you can list them:


```
$ glance image-list
```

ID	Name	Disk Format	Container Format	Size	Status
6e68a12f-66c4-42a7-984c-71f7675e9c68	Cirros 0.3.3	qcow2	bare	13200896	active
da0e551d-aa61-4cc8-8399-4c0eda521fcc	Fedora 20 cloud	qcow2	bare	210829312	active

Same approach for the flavors:

Note: When spawning a VM on a node with 6WIND Virtual Accelerator installed, select a flavor configured with hugepages attribute. Flavors configured with hugepages have names ending in "_hugepages". for example, the flavors `nodisk.xlarge_hugepages`, `nodisk.xlarge_8GB_hugepages`, `nodisk.3cores_hugepages` and `nodisk.tiny_hugepages` are configured with the hugepages attribute. If a hugepage flavor is not used, the VM interfaces will be managed by linux (the physical interface is still managed by VA) and performance will be sub-optimal.

```
# nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public
1	ml.tiny	512	1	0		1	1.0	True
labd0da8-56bf-4564-a925-362b66f47e55	nodisk.xlarge_hugepages	16384	0	0		8	1.0	True
lbdfb1a7-8a4d-421e-a6cb-e79b832d720b	nodisk.xlarge_8GB_hugepages	8192	0	0		8	1.0	True
lca3f95b-ec57-4535-9d9c-10b17d11c6d0	nodisk.3cores_hugepages	3072	0	0		3	1.0	True
2efacaf0-afa9-441c-bc42-e520b9fece07	nodisk.tiny_hugepages	512	0	0		1	1.0	True
3	ml.medium	4096	40	0		2	1.0	True
4	ml.large	8192	80	0		4	1.0	True
4786e2d4-73dc-4424-abc4-1f6c7e4b59ca	nodisk.normal	2048	0	0		2	1.0	True
4def405a-c1ba-4623-a04c-6c50b2bd262f	nodisk.large	8192	0	0		4	1.0	True
5	ml.xlarge	16384	160	0		8	1.0	True
6	nodisk.tiny	512	0	0		1	1.0	True
62c8251b-6842-4930-b63d-bbce36af5516	nodisk.xlarge	16384	0	0		8	1.0	True
6a782f0b-9236-4704-9315-4df340b5857a	nodisk.3cores	3096	0	0		3	1.0	True
704bc63f-0289-4e5f-a13b-9447061f6c6c	ml.mq	2048	0	0		4	1.0	True
86553db8-40e3-47f7-ba94-5b0097c03728	nodisk.small_hugepages	2048	0	0		1	1.0	True
a1d4b912-3f91-45f7-b61d-f07b6dc7b687	tgen.vm_mq	4096	0	0		4	1.0	True
aaae5c3-4ec8-4c96-9e5b-5f147efe698b	nodisk.5cores4queues_hugepages	2048	0	0		5	1.0	True
ada10c52-ffce-4c6d-9d72-6976133e6de6	nodisk.large_4GB	4096	0	0		4	1.0	True
bbd3f244-7038-407a-8177-f57d9b4a76da	nodisk.large_4GB_hugepages	4096	0	0		4	1.0	True
bcd9f01a-leaa-4c1c-ac25-929a66d50fd6	nodisk.large_hugepages	8192	0	0		4	1.0	True
c051b75-f95f-433a-886f-fe87db76190a	nodisk.4cores3queues_hugepages	4096	0	0		4	1.0	True
c0d16493-0ad1-4d87-bff2-48fff3dd23b5	nodisk.xlarge_8GB	8192	0	0		8	1.0	True
c2f1f808-8b85-498f-a29f-68b6a039bc48	nodisk.small	2048	0	0		1	1.0	True
c60cc599-da69-43cb-9d1e-08086852ef08	nodisk.medium	4096	0	0		2	1.0	True
d533fld3-0ffc-4ce8-822d-1401778f53df	ml.small.eph	2048	10	20		1	1.0	True
d64e544a-a169-4c17-b62d-5981e07e6ce8	nodisk.medium_hugepages	4096	0	0		2	1.0	True
dd1c646b-641c-4f15-a521-27458aa099fe	ml.small	2048	30	0		1	1.0	True
e94dea29-2052-4a08-ab5a-f1d6d83010e9	nodisk.normal_hugepages	2048	0	0		2	1.0	True

you need to specify on which network(s) it will be with `--nic net-id=$NET_ID` option

```
$ nova boot --flavor nodisk.tiny \
--image "Cirros 0.3.3" \
--nic net-id=905dd0a6-b7c8-4041-81a1-3fd0d8df842f \
--security-group default vml
```

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	nova
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	-
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	-
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
adminPass	dM6ngEe2KuJm
config_drive	
created	2015-05-11T08:53:16Z
flavor	nodisk.tiny (6)
hostId	3c000e6ed15f2f3c3f335def6fdfca9ccb9075d72d4c237cc0990249

id	3cf57f5e-3bed-4981-9c11-f5ffcac52630
image	Cirros 0.3.3 (6e68a12f-66c4-42a7-984c-71f7675e9c68)
key_name	-
metadata	{}

```

| name | vm1
| os-extended-volumes:volumes_attached | []
| progress | 0
| security_groups | default
| status | BUILD
| tenant_id | 8ce7f2e52136404eb691bf01dc472268
| updated | 2015-05-11T08:53:16Z
| user_id | 5c75d650cb244edbb8e8edf35a24ab4a
+-----+
$ nova list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+-----+-----+
| 3cf57f5e-3bed-4981-9c11-f5ffcac52630 | vm1 | ACTIVE | - | Running | net_test=10.0.0.2 |
+-----+-----+-----+-----+-----+-----+

```

For cloud image, see cloud-init section to set up a password.

7.3 Cloud-init

For cloud image, cloud-init can be used. Cloud init can set a lot things at the first boot, in our case we will use this config:

```

#cloud-config
password: demo
chpasswd: {expire: False}
sudo: ['ALL=(ALL) NOPASSWD:ALL']
groups: sudo
shell: /bin/bash

```

This configuration, sets "demo" as password and sets the user ubuntu or fedora on the sudo group with bash as default shell.

Requirement: To use cloud-init, a router is required on the network where the VMs reside (it is the gateway for the VM and allows communication with the meta-data server at 169.254.169.254).

7.3.1 GUI

Go in "Post-Creation" tab, then select "Direct Input" in Customization Script Source and in Script Data, put the configuration above.

Launch Instance x

Details *Access & Security *Networking *Post-CreationAdvanced Options

Customization Script Source

Direct Input ▼

Script Data ⓘ

```
#cloud-config
password: demo
chpasswd: {expire: False}
sudo: [ALL=(ALL) NOPASSWD:ALL]
groups: sudo
shell: /bin/bash
```

You can customize your instance after it has launched using the options available here.

"Customization Script" is analogous to "User Data" in other systems.

Cancel Launch

7.3.2 cli

In a file (for example user_data.file), write the configuration above and add the following option to the nova boot command

```
--user-data user_data.file
```

7.4 Specify an availability zone

You can specify the availability zone in which you want to span the VM.

There are four areas available:

- Accelerated compute 1 and 2 are two compute nodes with [6WIND Virtual Accelerator](#) installed
- Standard Linux 1 and 2 are two compute nodes without [6WIND Virtual Accelerator](#) installed
- Ubuntu 40G 1 and 2 are two compute nodes with [6WIND Virtual Accelerator](#) installed and Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+.

7.4.1 GUI

In the Details tab after "Launch Instance", you can choose the area in which you want to spawn the VM using the Availability Zone field.

Launch Instance

×

Details * Access & Security * Networking * Post-Creation Advanced Options

Availability Zone

Any Availability Zone ▼

Instance Name *

Test

Flavor * ?

m1.tiny ▼

Instance Count * ?

1

Instance Boot Source * ?

Boot from image ▼

Image Name

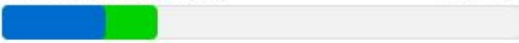
Fedora 20 cloud (201.1 MB) ▼

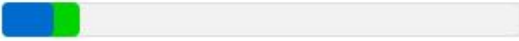
Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

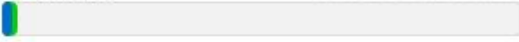
Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

Number of Instances 2 of 10 Used


Number of VCPUs 2 of 20 Used


Total RAM 1,024 of 51,200 MB Used


7.4.2 Cli

Areas can be seen with nova command:

```
$ nova availability-zone-list
+-----+-----+
| Name                | Status |
+-----+-----+
| Accelerated compute 2 | available |
| Accelerated compute 1 | available |
| Standard Linux 1     | available |
| Standard Linux 2     | available |
| Ubuntu 40G-1         | available |
| Ubuntu 40G-2         | available |
+-----+-----+
```

Now, we can specify in which Availability zone we want to spawn the new VM using the `--availability-zone` option in `nova boot`:

```
$ nova boot --flavor nodisk.tiny --image "CirrOS 0.3.3" \
--nic net-id=905dd0a6-b7c8-4041-81a1-3fd0d8df842f \
--security-group default --availability-zone "Accelerated compute 1" vm2
```

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	Accelerated compute 1
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	-
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	-
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
adminPass	bhS7TuCUpm5E
config_drive	
created	2015-05-12T09:30:23Z
flavor	nodisk.tiny (6)
hostId	eb4c271bdd30a2e3dab174255138e4056642548760c7e792e2faf144
id	6edfc537-da64-496b-a5eb-93898e50606d
image	CirrOS 0.3.3 (6e68a12f-66c4-42a7-984c-71f7675e9c68)
key_name	-
metadata	{}
name	vm2
os-extended-volumes:volumes_attached	[]
progress	0
security_groups	default
status	BUILD
tenant_id	8ce7f2e52136404eb691bf01dc472268
updated	2015-05-12T09:30:23Z
user_id	5c75d650cb244edbb8e8edf35a24ab4a

```
:width:100%
```

8 Multiqueues

8.1 For Virtio Linux driver

Any standard Linux VM supporting virtio device can benefit of the [6WINDGate fast path acceleration](#), thanks to vhost-user driver that pushes the packet through some fast IOs toward the virtual ports of the VMs.

8.1.1 Enabling multiqueue manually

A template image can be configured to enable by default multiqueue support for virtio devices.

To enable multiqueue, the following steps need to be done:

- Add the image to Glance, setting the `hw_vif_multiqueue_enabled` metadata.
- On the VM, `ethtool` needs to be installed to configure queues. Example to setup 4 queues for `eth0`:

```
# ethtool -L eth0 combined 4
```

- Configure the right number of queues

```
for DEVICE in $( ip a | grep ^[0-9]*: | awk '{print $2}' | sed -e 's://://' )
do
    [ "$DEVICE" != "lo" ] || continue
```

```

nb_queues=`ethtool -l $DEVICE | grep Combined: | awk '{print $2}' | head -n1`
ethtool -L $DEVICE combined $nb_queues

# configure tx queues
nb_processor=`cat /proc/cpuinfo | grep processor | wc -l`
nb_xps=$nb_processor
if [ "$nb_queues" -lt $nb_xps ]; then
    nb_xps=$nb_queues
fi

for i in `seq 0 $(( $nb_xps - 1 ))`;
do
    let "mask_cpus=1 << $i"
    echo $mask_cpus > /sys/class/net/$DEVICE/queues/tx-$i/xps_cpus
done
done

```

```
# glance image-create {...} --property hw_vif_multiqueue_enabled=true
```

8.1.2 Boot VM with several queues

To benefit of best networking performance, it's recommended to have a queue for each virtual CPU of the VM processing packets. This section explains how to boot a VM with several queues. Only those with administrator privilege can create or modify a flavor. The following is only for reference. All the flavors have been configured with the option `vif_multiqueue_enabled`. The flavor with "Xqueues" on the name have been configured with the property `hw:vif_number_queues=X`

- Create a new nova flavor with the multiqueue property:

```

source /root/admin-openrc.sh
nova flavor-create m1.vm_mq auto 512 3 4
nova flavor-key m1.vm_mq set hw:vif_multiqueue_enabled=true

```

- Optional: You can specify the exact number of queues with the `hw:vif_number_queues` key:

```
# nova flavor-key m1.vm_mq set hw:vif_number_queues=4
```

- Then, boot the VM with the new flavor:

```

# source keystone_admin
# nova boot --flavor m1.vm_mq --image fedora-virtio fedora20_multiqueue \
--availability-zone nova:compute1 --nic net-id=$(neutron net-list | grep private | awk '{ print $2 }') \
--user-data cloud.cfg

```