OpenStack Tutorial

I. OpenStack Dashboard

The goal of this part is to familiarize yourself with OpenStack dashboard. The focus will be on the following operations (for more information: https://docs.openstack.org/user-guide/dashboard.html):

- Setting up SSH keys
- Setting up networks
- Creating VM

In order to perform these operations you need to login Ericsson's cloud via https://129.192.68.4/. This will give you access to your project(s). Once logged in, you will get a page similar to Fig 1.



Setting up SSH keys

In order to be able to SSH your instances, you need to create and download your keypair file. For that to happen you need to do the following two steps: enable ssh service and create keypair.

1)Enabling SSH

On the Project tab, open the Network tab, then Security Groups. You will get a window similar to Fig 2 under Security Group tab. Check if ssh service is enabled by clicking Manage Rules button on far right (see Fig 3).

Project		~	Pro	pject / Network / Security Groups					
	Compute	>							
	Network	~	Se	ecurity Groups					
Network Topology		pology							
	Ne	etworks				Filter	Q	+ Create Security Group	Delete Security Groups
	1	Routers	Disp	playing 1 item					
	Security	Groups		Name	Description				Actions
	Floa	ting IPs		default	Default security group				Manage Rules
(Orchestration	>	Disp	playing 1 item					
I de active e									

Fig 2. Security Group

Project *	Μ	Manage Security Group Rules: default								
Compute ~	Security Group Rules									
Overview	(Direction	Ether Type	IP Protocol	Port Range	Remote	Actions			
Instances	(Egress	IPv4	Any	-	0.0.0.0/0 (CIDR)	Delete Rule			
Volumes	(Ingress	IPv4	Any	-	default	Delete Rule			
Images	(Egress	IPv6	Any	-	::/0 (CIDR)	Delete Rule			
Access & Security	ĺ	Ingress	IPv6	Any	-	default	Delete Rule			
Network >	(Ingress	IPv4	ТСР	22 (SSH)	0.0.0.0/0 (CIDR)	Delete Rule			
Object Store >	(Ingress	IPv4	ТСР	80 (HTTP)	0.0.0.0/0 (CIDR)	Delete Rule			
Urcnestration >	D	isplaying 6 items								

Fig 3. List of enabled services

ssh is enabled

If **ssh** is not in the list, select +Add Rule which will bring up a popup. Select **ssh** from the drop down and press Add (see fig 4).

Add Rule	×
Rule • SSH Remote • • CIDR CIDR • 0.0.0.0/0	Description: Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts: Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule. Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided. Remote: You must specify the source of the traffic to be
	allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.
	Cancel

Fig 4. Adding ssh rule.

Note that you can follow the same step to enable any service, for example http.

2)Creating key pair

On the Project tab, open the Compute tab, then Key Pairs. You will get similar to Fig 5 under Key Pairs tab. Select +Create Key Pair and provide a name to your key pair. The file will be downloaded automatically (if not please download the file manually). Now you can use the **ssh** command to make a secure connection to your instance (We will see how that later once a VM is created).

Compute	~					
	Overview	Key Pairs				
	Instances					
	Volumes			Filter Q	+ Create Key Pair	1 Import Key Pair
	Images	Key Pair Name	Fingerprint		Actions	
	Key Pairs		No items to disp	play.		
,	API Access					

Fig 5. Creating Key Pair.

<pre>\$ ssh -i MyKey.pem ubuntu@10.0.0.2 Mykey.pem is set to owner.</pre>	2 #make	sure	that th	e permission	to

For **Windows** user please check:

https://github.com/davidheijkamp/docs/wiki/Howto:-Creating-and-using-OpenStack--SSH-keypairs-on-Windows

Setting up Network

This section will show you how to create a new network, set up a subnet associated with the network and create router.

1)Creating network and its associated subnet

On the Network tab, open the Networks tab, then press +Create Network on the top right side. Provide the required information and press Next (see fig 6).

Create Network	×
Network * Subnet * Subnet Name Test Admin State * UP	Create a new network. In addition, a subnet associated with the network can be created in the next panel.
Fig 6. Creating network.	Cancel « Back Next »

The UI shown in fig 7 will be displayed after pressing Next in fig 6. Provide the necessary information and press Next. It is advisable to use private IP address ranges either from class A, B, or C in Network Address field. You do not have to specify a subnet when you create a network, but if you do not specify a subnet, the network cannot be attached to an instance.

Create Network	×
Network * Subnet * Subnet Details	
 Create Subnet Subnet Name Test 	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.
Network Address * 😧	
IP Version *	
Gateway IP @	
	Cancel « Back Next »

Fig 7: Creating subnet.

Press Next in fig 7 and press Create in Fig 8. You have now created a network with subnet!!

Create Network

Network * Subnet * Subnet Detail	
Enable DHCP	Specify additional attributes for the subnet.
Allocation Pools	
DNS Name Servers	
8.8.4.4 8.8.8.8	
Host Routes 🕢	
« Ba	Create

Fig 8: creating subnet final step.

Creating Router

For VMs to communicate with the external world you need to set up a router.

On the Project tab, open the Network tab and click Routers category and press +Create Router. Specify a name for the router and External Network, and click Create Router(see fig 9).

Create Router

Router Name *	
Test_router	Description:
Admin State	Creates a router with specified parameters.
UP 💠	
Select network	
✓ net04_ext	
	Cancel Create Router

Fig 9: Creating router.

To connect the private network created above to the newly created router, perform the following steps:

- a. On the **Routers** tab, click the name of the router you created.
- b. On the Router Details page, click the Interfaces tab, and then click Add Interface.

>

×

c. In the Add Interface dialog box, select the Subnet you created above (see Fig 10).

Add Interface	×
Select Subnet cristi: 192.168.0.0/24 (cristi-blah) CloudCourse: 10.0.0.0/24 (CloudCourse-subnet) ✓ Test: 172.16.0.0/24 (Test) IP Address (optional) Router Name * Test_router	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.
Router ID *	
8c0fac42-76ea-41e4-b1ce-b2737e66f945	
	Cancel Add interface

Fig 10: Connecting private network with router.

Setting up a VM

It is now time to create a VM and play with it!!

1) Creating VM

On the **Project** tab, open the **Compute** tab and click **Instances** category. The dashboard shows the instances with its name, its private and floating IP addresses (we will come to this later), size, status, task, power state, and so on.

Click Launch Instance in the top right corner and provide the necessary information (see figs 11-15). To see the progress on how the VM is initializing, click the name of the VM instance in the list of instances, then click 'Log'.

Details	Please provide the initial hostname for the instance, the availability zo count. Increase the Count to create multiple instances with the same	one where it will be deployed, and the instance settings.
Source *	Instance Name *	Total Instances (10 Max)
Flavor *	TestVM	30%
Notworko *	Availability Zone	3070
Networks	nova	Current Usage
Network Ports	Count *	1 Added 7 Remaining
Security Groups	1	٢
Key Pair		
Configuration		
Server Groups		
Scheduler Hints		
Metadata		
× Cancel		< Back Next >

Fig 11: Creating VM.

Launch Instance

Details	Insta (imag	nce source is the te ge snapshot), a volu	emplate used to create an ins ime or a volume snapshot (if	tance. enable	You can us d). You can	e an imag also choc	e, a snapshot o ose to use persi	of an instance istent storage by	>
Source	Sele	ct Boot Source			Create N	ew Volum	ie		
Flavor	Ima	age		\$	Yes	No			
Networks	Volu	me Size (GB) *			Delete Vo	olume on	Instance Delet	te	
Network Ports	4			٢	Yes	No			
Security Groups	Alloc	ated							
Key Pair		Name	Updated		Size	Ту	ype Visik	bility	
Configuration	>	ubuntu 16.04	5/11/17 4:31 PM		2.20 GB	ra	aw Publ	lic 🔸	
Server Groups	✓ A	vailable 2						Select one	
Scheduler Hints	Q	Click here for filte	ers.					×	
Motodoto		Name	Updated	Size		Туре	Visibilit	ty	
Metauata	>	CirrOS-raw	5/11/17 8:50 AM	39.22	MB	raw	Public	•	
	>	CentOS 7	5/11/17 2:01 PM	8.00 (ЭB	raw	Public	•	
≭ Cancel						< Back	Next >	Launch Instance	

Fig 12. Selecting Source.

Launch Instance								×
Details	Flavors manage the sizing for the compute, memory and storage capacity of the instance.							0
Source	Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
Flavor	> c2m3	2	3 GB	20 GB	20 GB	0 GB	Yes	↓
Networks	✓ Available	9					s	Select one
Network Ports	Q Click h	ere for filters						×
Security Groups	Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
Key Pair	> c2m2	2	2 GB	20 GB	20 GB	0 GB	Yes	^
Configuration	> c2m1	2	1 GB	20 GB	20 GB	0 GB	Yes	^
Server Groups	> c3m2	3	2 GB	20 GB	20 GB	0 GB	Yes	•
Scheduler Hints	▶ c3m4	3	4 GB	20 GB	20 GB	0 GB	Yes	↑
motadata	▶ c2m4	2	4 GB	20 GB	20 GB	0 GB	Yes	•
	▶ c3m1	3	1 GB	20 GB	20 GB	0 GB	Yes	•
X Cancel						< Back Next >	🗗 Launc	h Instance

Fig 13. Selecting Flavor.

Х

Launch Instance				×	
Details	A key pair allows you t pair, or generate a new	o SSH into your newly / key pair.	created instance. You may select an existing key pair, import a key	8	
Source *	+ Create Key Pair	1 Import Key Pair			
Flavor *	Allocated				
Networks	Displaying 0 items				
Networks	Name		Fingerprint		
Network Ports		Select a key	pair from the available key pairs below.		
Security Groups	Displaying 0 items				
Key Pair	✓ Available 1		Sele	ect one	
Configuration	Q Click here for fi	Iters.		×	
Server Groups	Displaying 1 item				
Scheduler Hints	Name	Fingerprint 🖨			
Metadata	> bayuh	19:fb:c8:71:6d:70:55:e6:9b:02:06:01:e3:2c:c8:cf			
	Displaying 1 item				

Fig 14: Select the key pair if you have more than one (Choose the key pair you created above).

Launch Instance								×
Details	Networks provide	e the communi	cation channels for	instances in t	he cloud.	Select networks	from those liste	ed below
Source *	Net	twork	Subnets Associa	ited	Shared	Admin State	Status	
Flavor *	♦ 1 > Tes	tNetwork	test		No	Up	Active	•
Networks	✓ Available 3					Sel	ect at least one	network
Network Ports	Q Click here	o for filters.						×
Security Groups	Network	Subnets	Associated	Shared	Adr	nin State	Status	
Key Pair	> ck-net	ck-subn	et	No	Up		Active	•
Configuration	> ekernet	ekernet-	sub	No	Up		Active	•
Server Groups	> internet	internet-	sub1	Yes	Up		Active	•
Scheduler Hints								
Metadata								
¥ Cancel					< B	ack Next >	合 Launch Ir	nstance

Fig 15: Network setup (choose the network you created above)

2) Associated floating IP to a VM

× Cancel

Associating floating IP to a VM helps to associate public IP address to your VM so that it can be accessed externally.

On the **Project** tab, open the **Compute** tab and click **Instances** category. On the far right parallel to the instance click the drop down menu and select **Associate Floating IP**. Choose from the list and click **Associate** (see fig 16). Sometimes, you may not see

any Floating IPs, in which case you need to click the "+" button to the right of the floating IP. Your VM is now accessible from anywhere!! Please remember the IP, you will need it soon!!

IP Address *		
IP Address *	Select the IP address you wish to assoc	iate with the
 Select an IP address 	 selected instance or port. 	
94.246.117.148		
94.246.117.149		
94.246.117.147	\$	
	Cancel	Associate

Fig 16: Associating floating IP.

It is now time to play with the VM. Let us login to the VM. Remember the key pair file that you downloaded sometime ago. It is time to use it now to connect to the vm.

Use the **ssh** command to make a secure connection to the instance as shown below.

\$ ssh -i MyKey.pem ubuntu@floating_ip

II. Python Script

The operations that you did using the dashboard can also be done using python. To do that we use the VM that you created above as a devVM. Please login to the selected devVM. Once you are there, you need to install the following OpenStack libraries (This tutorial uses python-novaclient 7.1.0).

\$ sudo apt-get update
\$ sudo apt install python-dev python-pip
\$ export LC_ALL=C
\$ sudo pip install python-novaclient==7.1.0
\$ sudo pip install python-swiftclient

VM-related Operations

We have provided a python script for some of the operations. Please clone <u>https://github.com/ewnetu/WASP.git</u> and check **vm-operations.py** and **vm-init.sh** scripts (the scripts are the basis for the micro services section below). The script is based on python-novaclient 7.1.0 and some of the operations may not work if you use a different version (for more: <u>https://docs.openstack.org/developer/python-novaclient/ref/v2/</u>).

\$ git clone https://github.com/ewnetu/WASP.git

In order to run the script you need to specify the right information in **config.properties** file. Below is a sample example (Note that you only need to change the values for the username, password,projectName, keyName and netId properties). **Don't forget to put your key file in the same directory as vm-operations.py** since it will be used during VM instantiation (check createVM() method)

[user] username:your username password: your pass [openstack] projectName:your project name user_domain_name:xerces project_domain_name:xerces project_domain_id: to get project id login to the dashboard select identity tab then projects authUrl:https://xerces.ericsson.net:5000/v3 keyName:your keyname netId:your network name

Add the following line in /etc/hosts file.

129.192.68.4 xerces.ericsson.net

The examples below show how to create a vm called 'WASP" as well as list all VMs for the tenant specified in the property file. Go through the source and try out the different operations implemented in the script and see their effect.

```
$ python vm-operations.py -o create -n WASP
$ python vm-operations.py -o listVM
```

Micro services

We have also added a small micro service inside WASP vm. Get the IP address of the WASP VM either from the dashboard or using python as shown below.

```
$ python vm-operations.py -o VMIP -n WASP
```

Then you can access the service as follows (Make sure to enable port 5000) (it is based on Flask rest API, for more <u>http://flask.pocoo.org/docs/0.12/quickstart/</u>):

\$ curl -i <u>http://10.0.0.37:5000/v1/hello</u>

Please note that the VM might take as much as 5 minutes to download the necessary packages. If the above command does not work the first time, retry a few minutes later or check the VM log for errors.

We will now introduce a simple master-worker service with three VMs (Frontend, RabbitMQ and Backend) each having different roles. The Frontend posts messages to RabbitMQ while the Backend pulls messages from RabbitMQ and prints them. For more information on RabbitMQ visit <u>https://www.rabbitmq.com/tutorials/tutorial-one-python.html</u>.

\$ git clone https://github.com/muyiibidun/WASP.git

In order to deploy the three VMs, please create *credentials.txt* file and put the following info (and don't forget to replace the values with your credentials and project name):

[auth]

username:your username
password: your pass
tenant_name: your project name
user_domain_name:xerces
project_domain_name:xerces
project_domain_id: to get project id login to the dashboard select identity tab then projects
auth_url: https://xerces.ericsson.net:5000/v3
net_id: net_id

pkey_id:pkey_id

Now you can deploy the three VMs as:

\$./deploy-waspmq.sh

Get the IP addresses of the Frontend, RabbitMQ, and the Backend. ssh to the Frontend and Backend VMs and perform the following operations. You must copy the SSH key (PEM file) generated at the start of the tutorial on the devVM. You need to open 3 terminals, one with the devVM, one with the frontend VM (via the devVM), and one with the backend VM (via the devVM).

a) Frontend VM Once you are inside the Frontend VM, perform the following operations.

\$ cd /var/www/WASP/waspmq

Then edit credentials.txt file and insert the IP address of rabbitmq server. Once you are done editing, run the service.

\$ python frontend.py -c credentials.txt

b) Backend VM

Once you are inside the Backend VM, perform the following operations.

\$ cd /usr/local/WASP/waspmq

Then edit credentials.txt file and insert the IP address of rabbitmq server. Once you are done editing, run the service.

\$ python backend.py -c credentials.txt

Now everything is ready, go back to your devVM and test the services.

\$ curl -i http://[frontend-ip]:5000/v1/waspmq/hej

\$ curl -i http://[frontend-ip]:5000/v1/waspmq/welcome+to+umea

Notice how the backend VM is picking up the 'work' submitted through the frontend VM. Thanks to their decoupling through a message queue (here RabbitMQ), you can scale up the application by adding several frontend VMs and backend VMs.

Exercise

How can the above manual operations (setting IP address of rabbitMQ to frontend and backend config files) be automated? (Hint: Start the rabbitMQ service first, get its IP and set the IP to credentials.txt. This should be done after the cloning but before starting the services.)