Getting started with WARA ML Arena

WARA Media and Language arena use Ericsson's Xerces Cloud provided by WARA Common for the research work.

Introduction to Ericsson's Xerces Cloud

The aim of the manual is to help you familiarize yourself with the cloud computing infrastructure used in Xerces, guide you to set up a network, SSH keys and Virtual Machines (VM), and finally, give you some experience working with the Linux command prompt on the VM.

Ericsson's Xerces cloud provides Infrastructure as a Service (IaaS) and it is built using OpenStack cloud software. For more information regarding the OpenStack cloud platform, check the following link: https://docs.openstack.org/train/user/

Getting access to the Project

In order to access the arena, you need to create an account to log in to the cloud. Fill the following google form to create an account: https://forms.gle/53PE2PuUUTV9yeDm7. You will receive the credentials in a couple of days via email (userID) and SMS (Password).

Remember, the arena currently holds a single project in the cloud and you need to share the resources with other researchers. However, you can request additional resources as long as it is reasonable and possible for us to comply with the request.

Once the credentials are ready, log in to the cloud via https://xerces.ericsson.net/.

OpenStack Dashboard

Once logged in, you will enter the OpenStack dashboard (also called Horizon), which provides a web-based user interface to OpenStack services like creating VM's, Storage, Network, etc. Fig 1 shows the Openstack dashboard after the login.

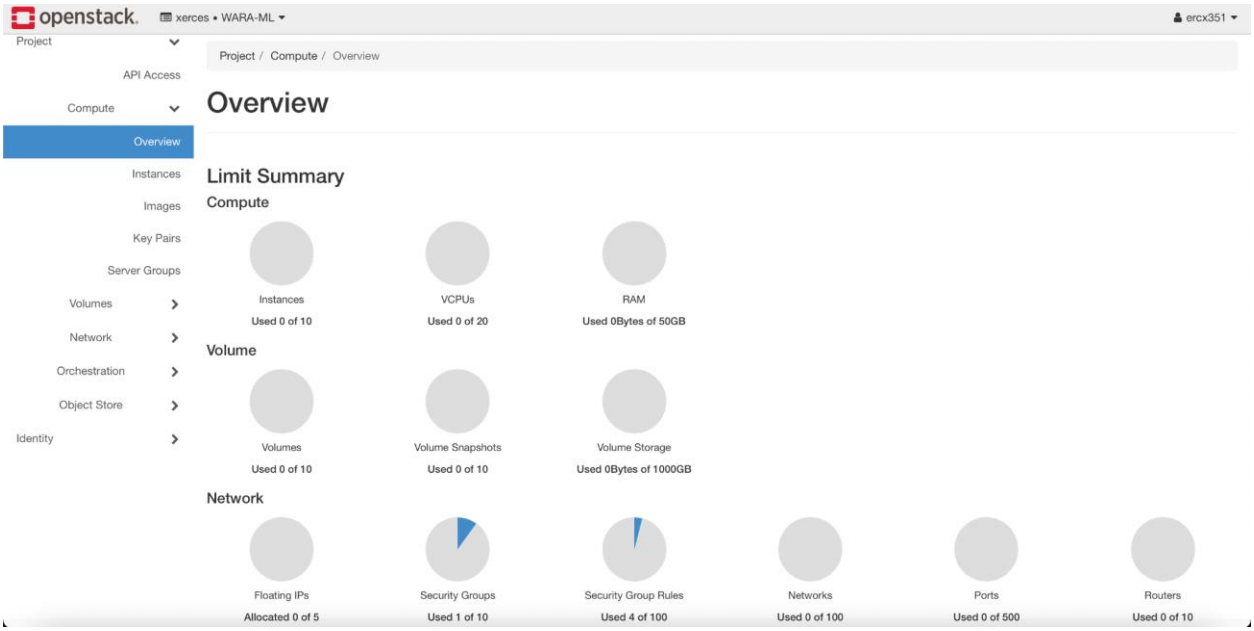

Fig 1: Project Dashboard

Setting up the Network

In this section, you will get to know about how to create a new network, set up a subnet associated with the network and create a router. If you want to use the existing network in the arena, go to step 4.

Step 1: Creating a new network

Follow the instructions below to create a network.

1. Go to the **Network** tab in the dashboard.
2. In the **Network** tab, press **Create Network** on the top right side. Provide the required information (in fig 2) and press Next.

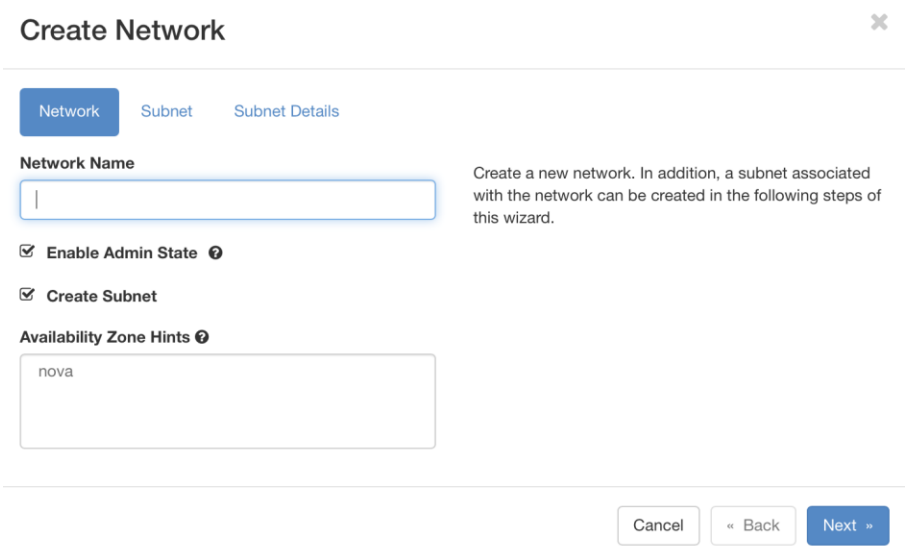

Fig 2: Creating a network.

Step 2: Setting up a subnet associated

The UI shown in fig 3 will be displayed after pressing Next in fig 2.

1. Enter the required information in the **Subnet Name** field.
2. For the **Network Address** field, it is advisable to use private IP address ranges either from class A, B or C. 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.

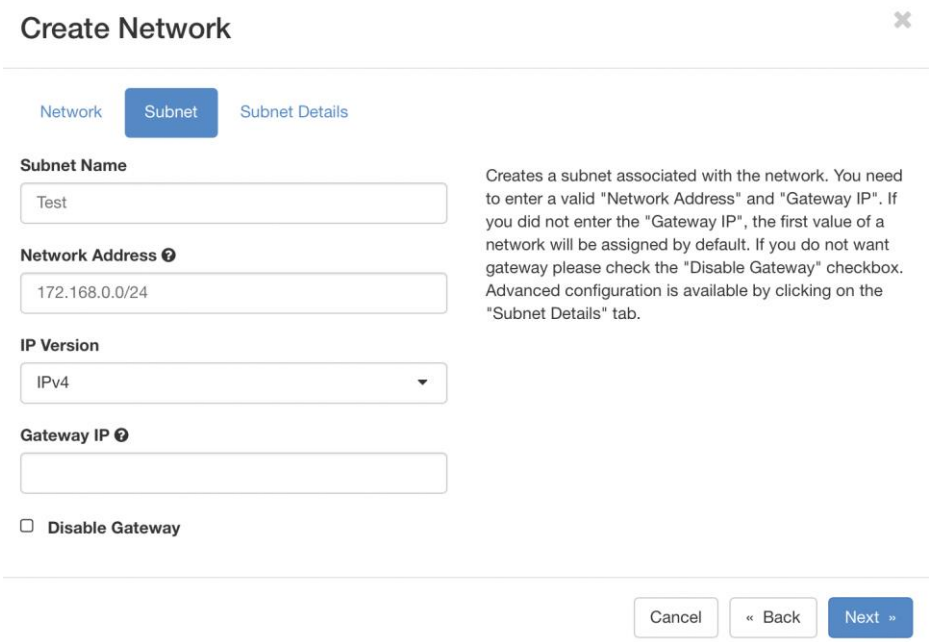

Fig 3: Creating a subnet

3. Press Next in fig 3 and press Create in fig 4. You have now created a network with a subnet.

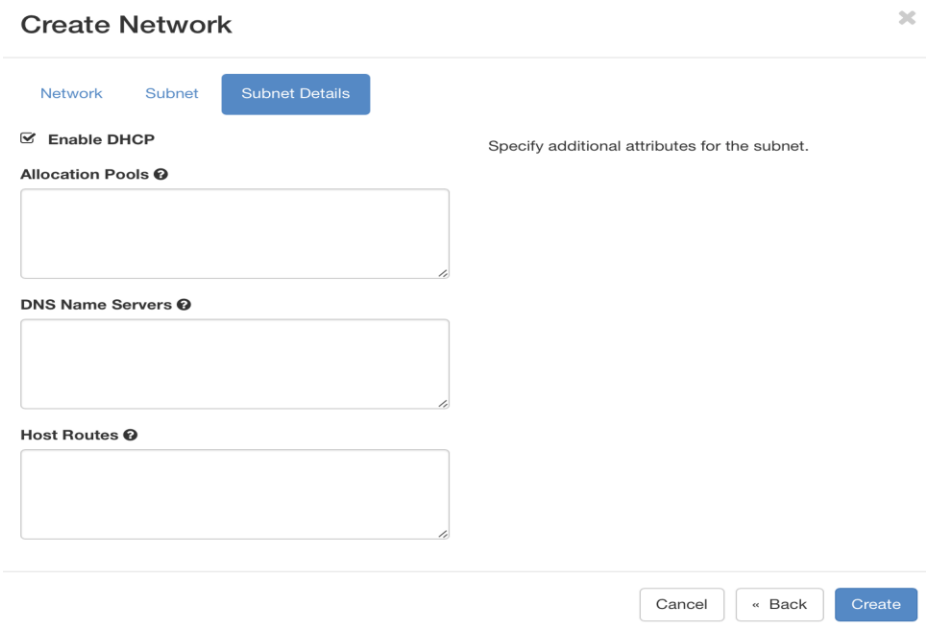

Fig 4: Creating subnet final step.

Step 3: Creating Router

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

1. On the Dashboard, open the **Network** tab.
2. Click the **Routers** tab and press **Create Router** on the top right side.
3. Specify a name for the router and choose the **External Network**.
4. Click **Create Router** (see fig 5).

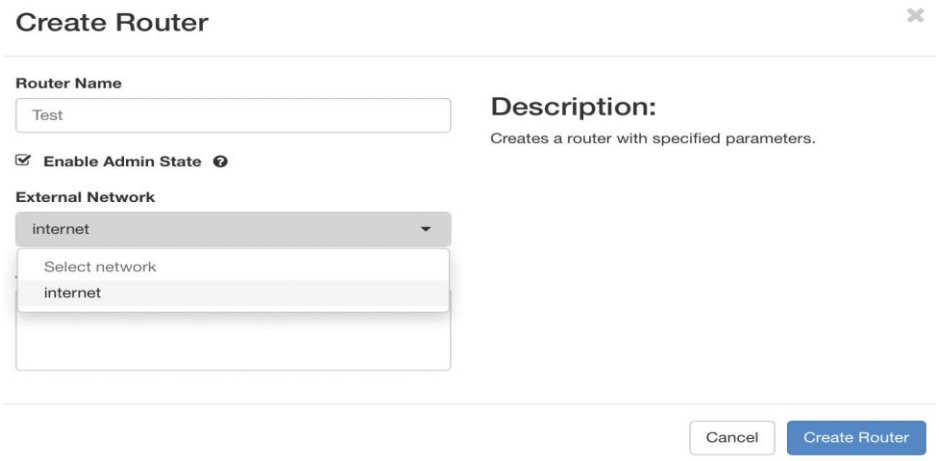

Fig 5: Creating a router

Step 4: Connect private network with a router

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

1. Go to the **Network** > **Routers** tab, click the name of the router you have created.
2. On the Router Details page, click the Interface tab and then click **Add Interface**.
3. In the **Add Interface** dialog box, select the **Subnet** you created and click **Submit** (See fig 6).

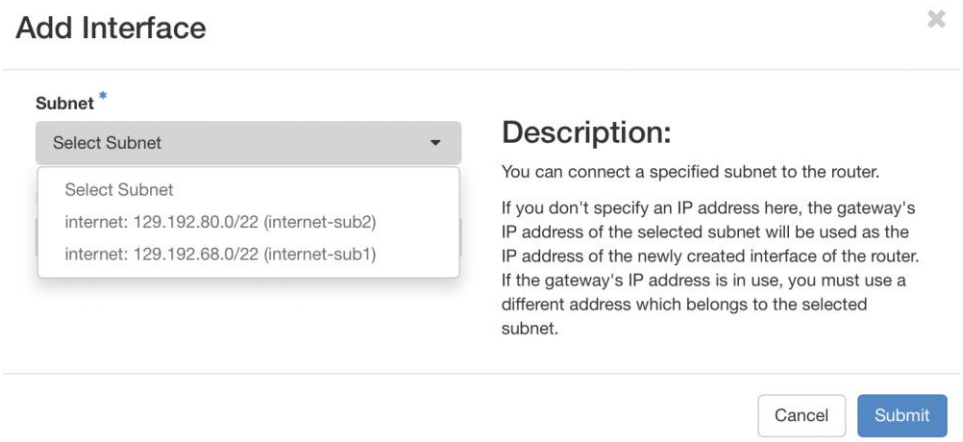

Fig 6: Connecting private network with the router

Setting up SSH keys

To access the cloud instances, the only method allowed is via ssh-keypairs. The Username/Password is disabled by default on all the cloud instances and should never be enabled for security reasons. To get familiarized with the use of ssh-keys, check out the link below:
http://blakesmith.me/2010/02/08/understanding-public-key-private-key-concepts.html

The OpenStack software helps you to create or import keys and will make sure that your public keys are injected in the instances you create. The private key should be private and is for you to safe keep on your clients.

In order to be able to SSH your instances, you need to follow the two steps:

Step 1: Enabling SSH

1. On the **Project** tab, go to the **Network** > **Security Groups** tab. You will get a window similar to fig 7 under the **Security Groups** tab.
2. Click on the **Manage Rules** button on the right side (see fig 7).

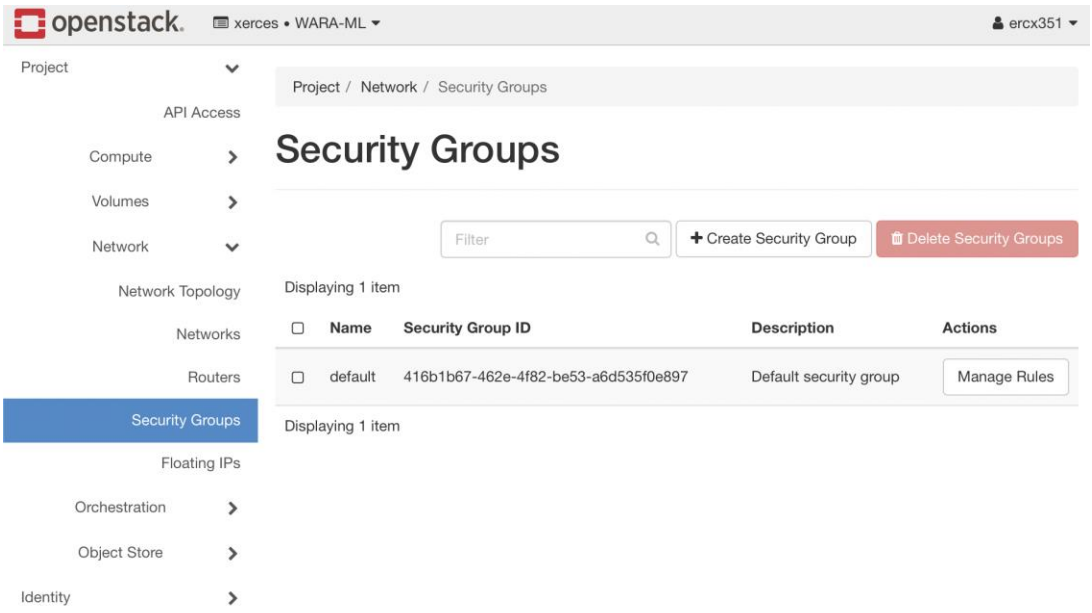

Fig 7: Security Groups

3. Check if ssh service is enabled in the list of rules (see fig 8).
4. If ssh is not in the list, select **Add Rule** which will bring up a window.
5. For the Rule option, select SSH from the drop-down menu and press **add** (see fig 9).

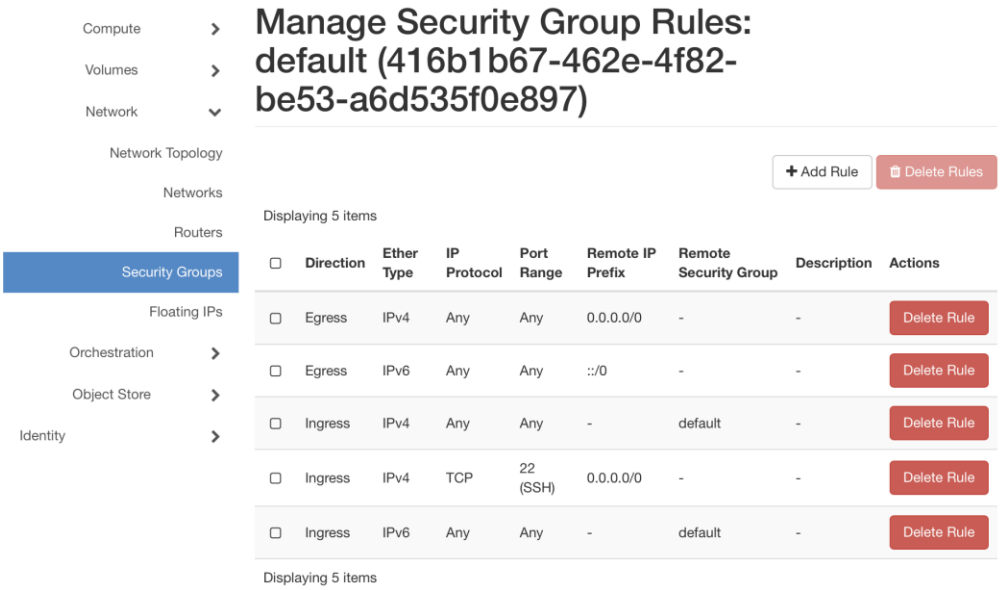

Fig 8: List of enabled services. You can see SSH enabled in the 4th rule.

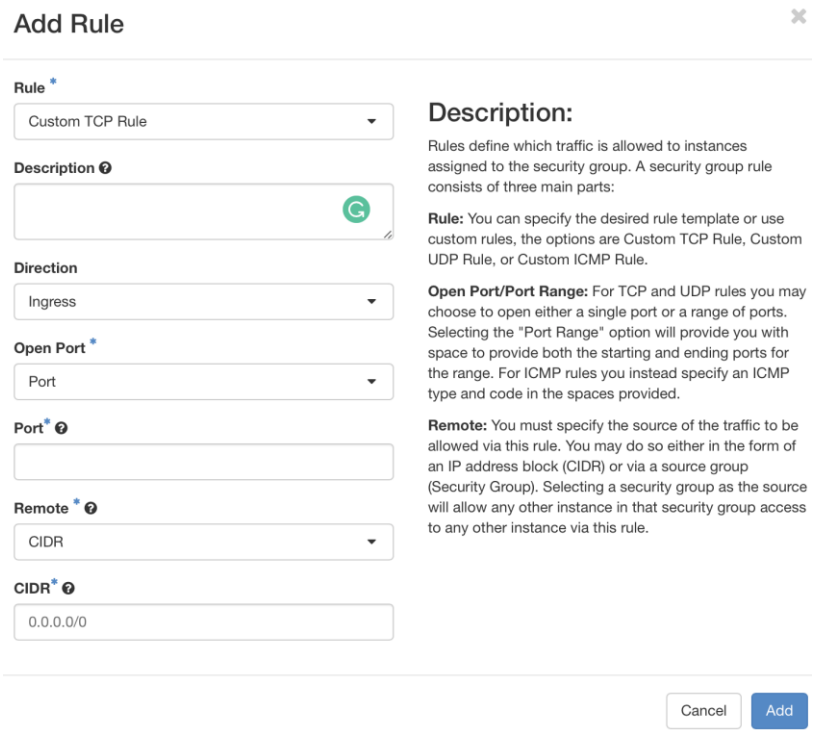

Fig 9: Adding SSH rule.

Note that you can follow the same step to enable any service, for example, HTTP. After enabling the SSH service, now you can create/ import keypairs which will be discussed in the next step.

Step 2: Creating key pair

1. On the **Project** tab, go to the **Compute** > **Key Pairs** tab. You will get an interface similar to fig 10 under the **Key Pairs** tab.
2. 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.
3. Save the download .pem file in a secure location on your computer.
4. Now you can use the SSH command to make a secure connection to your instance.
5. For Windows users, please check: https://creodias.eu/faq-openstack/-/asset_publisher/TpmSvagp3CVd/content/how-to-access-vm-from-windows-putty-?inheritRedirect=true

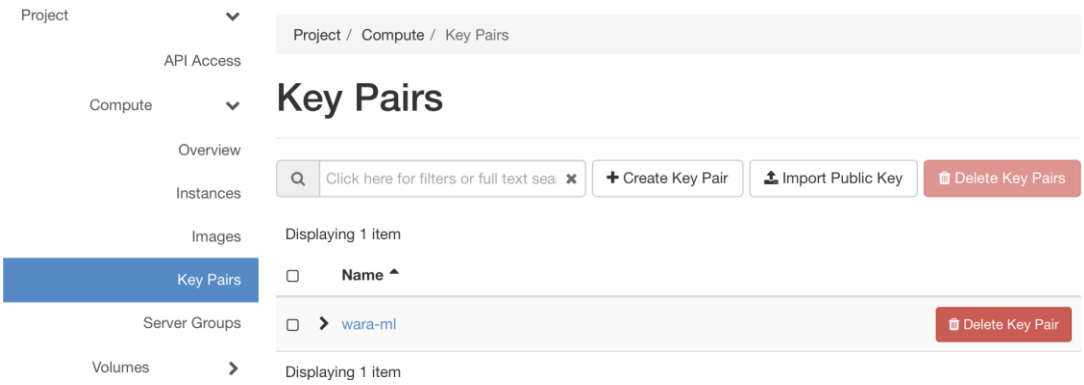

Fig 10: Creating Key Pair

Setting up a VM

In this section, you will learn how to launch an instance of a VM by booting from an existing image (which has an installed operating system).

Step 1: Creating VM

1. On the **Project** tab, go to the **Compute** > **Instances** category. The dashboard shows the instances with their name, their private and floating IP addresses, size, status, task, power state and so on.
2. Click **Launch Instance** in the top right corner and provide the instance name on the Details (see fig 11).

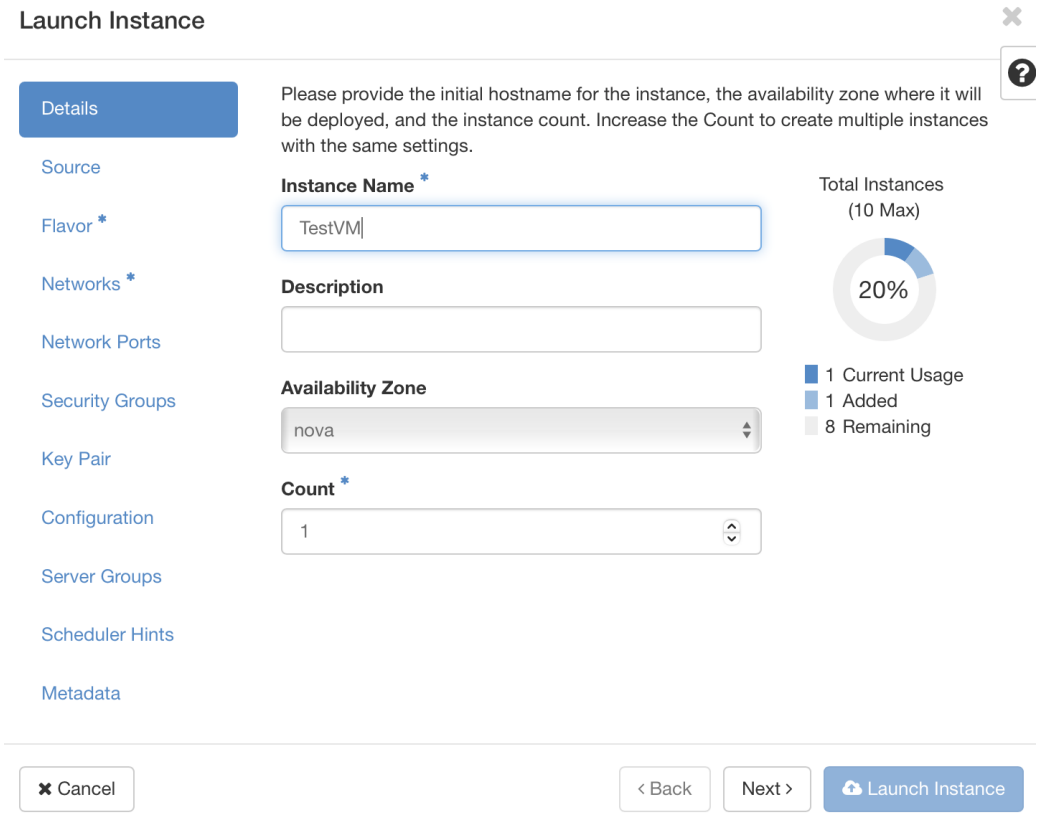

Fig 11: Creating VM

3. Select the preferred OS image as the boot source (see fig 12).
4. Select the flavor (size) depending on the needs (see fig 13).

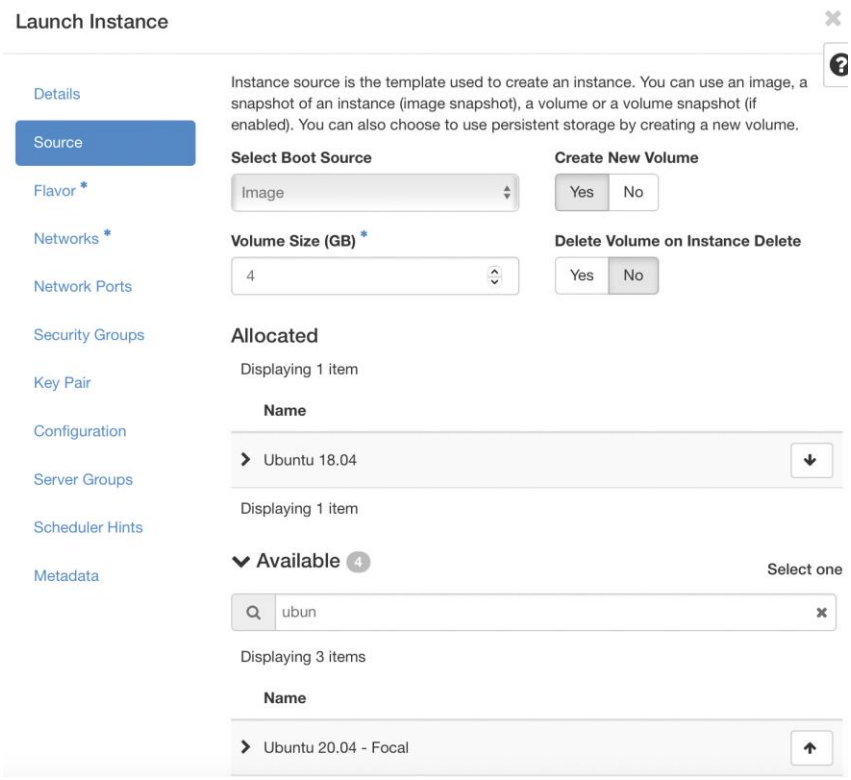

Fig 12: Selecting Source

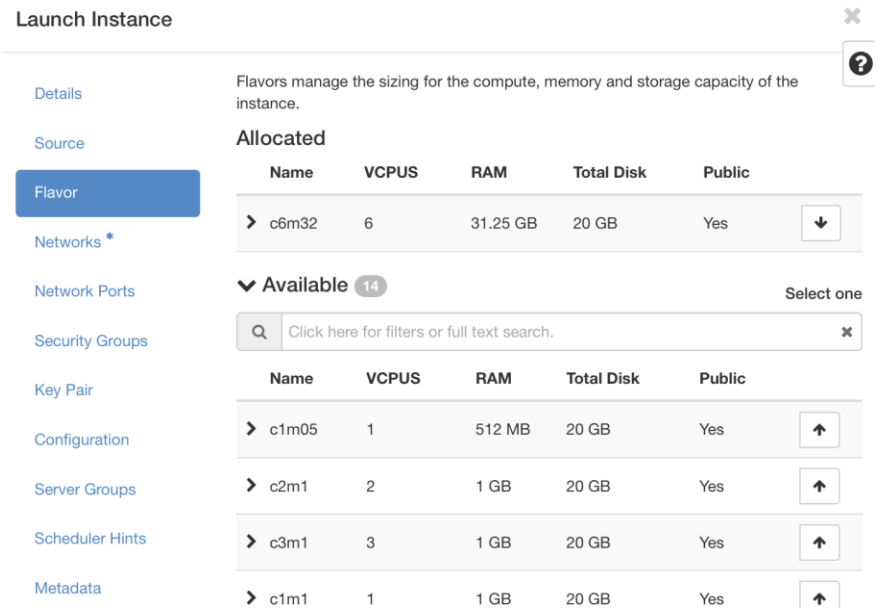

Fig 13: Selecting Flavor

5. Select the network that you have created before (see fig 14).
6. Select the keypair you have created in the previous section (see fig 15).
7. Check for other information and click **Launch Instance** at the end. Your instance will be ready soon.

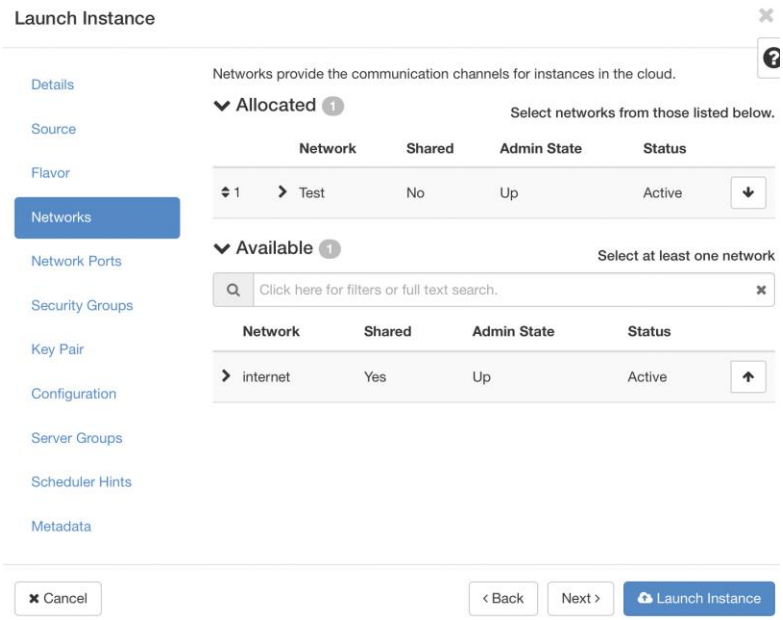

Fig 14: Network setup (choose the network you created above)

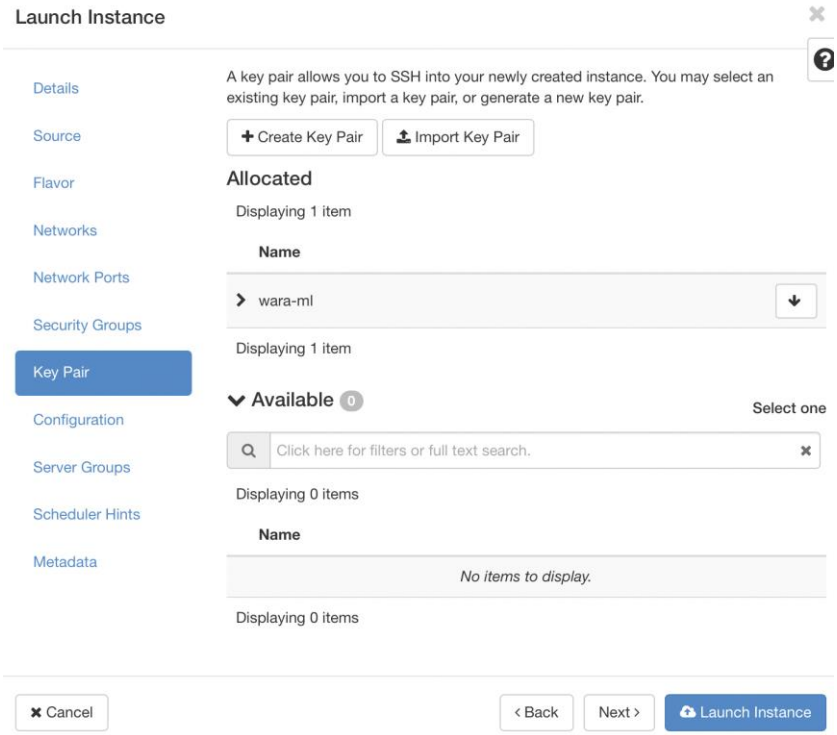

Fig 15: Select the key pair if you have more than one (Choose the key pair you created above)

Step 2: Associating floating IP to a VM

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

1. Go to **Project** > **Compute** > **Instances**.
2. On the far right side parallel to the instance click the drop-down menu and select **Associate Floating IP** (see fig 16).

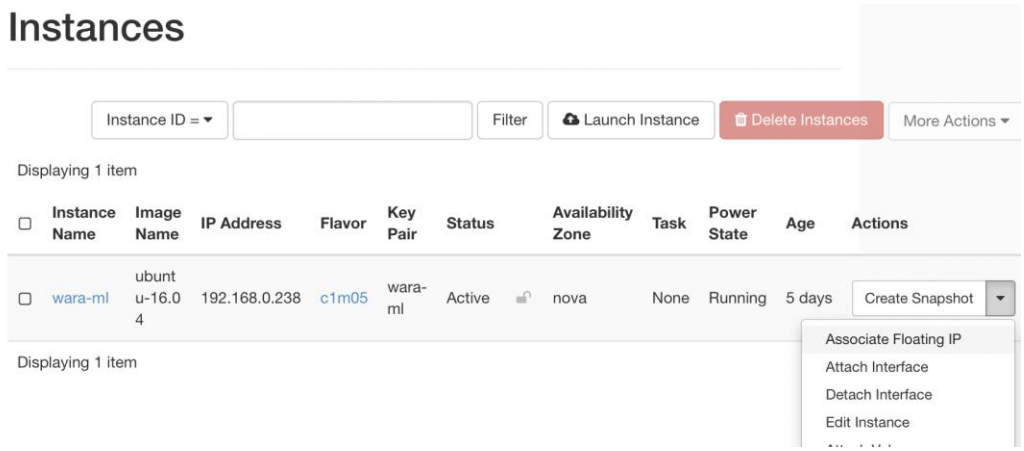

Fig 16: Select Associate Floating IP from the drop-down menu.

3. Choose from the list and click Associate (see fig 17).
4. If you don't find any Floating IPs in the list, click the "+" button to the right of the floating IP.
5. Your VM is now accessible from anywhere and remember the IP, as you will need it to log in using SSH.

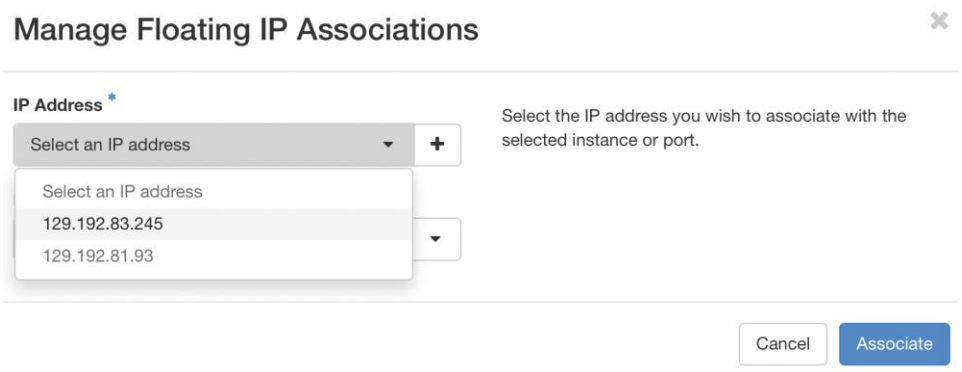

Fig 17: Associating Floating IP

Now, open the terminal on your PC. With the help of the key pair file that you downloaded before, you can log in to your VM using the following command:

ssh -i MyKey.pem ubuntu@floating_ip