CSCI 5380
Network Virtualization and Orchestration

Lab 2
OpenStack: Multi-tenants

University of Colorado Boulder
Department of Computer Science
Network Engineering

Professor Levi Perigo, Ph.D.

PART 1: OpenStack and multitenancy

Objective 1 - OpenStack: Overview

1. Explain the following components of OpenStack -
    a. Nova- The OpenStack project called Nova offers a method for allocating compute instances, sometimes known as virtual servers. With the help of ironic, Nova may be used to create virtual machines, baremetal servers, and system containers with restricted support. To deliver such service, Nova operates as a collection of daemons atop already-existing Linux servers.
    b. Swift- OpenStack's object storage component is called Swift. Swift allows businesses to store large amounts of data effectively, affordably, and safely. It is designed to be scalable and to be as durable, available, and concurrent as possible over the whole data set. Swift is the best option for storing unstructured data that has infinite growth potential.
    c. Cinder- The OpenStack Block Storage service Cinder is used to supply volumes to containers, Ironic bare metal hosts, Nova virtual machines, and more.
    Component based architecture: Quickly add additional behaviors.
    Extremely accessible: Scale to very significant workloads.
    Isolated processes prevent cascade failures, making them fault tolerant.
    Recoverable: Errors must be simple to identify, troubleshoot, and fix.
    Open Standards: Serve as a model implementation for an API driven by the community.
    d. Neutron- An OpenStack project called Neutron aims to offer "network connectivity as a service" between interface devices (like vNICs) that are under the control of different OpenStack services (like nova).
    e. Glance- Glance is the image service component of OpenStack. It manages and catalogs virtual machine images used to create instances in Nova. Glance allows users to store, browse, and retrieve virtual machine images efficiently.

f. Keystone- OpenStack's identification service component is called Keystone. For all other OpenStack services, it offers permission and authentication services. By controlling system and user credentials, Keystone guarantees safe access to all of the OpenStack environment's components.

g. Horizon- The OpenStack online dashboard is called Horizon. It gives customers access to a graphical user interface (GUI) via which they may interact and control cloud resources. By providing administrators and end users with an intuitive interface, Horizon streamlines the administration of OpenStack services.

2. What is the difference between Users and Roles?

   A role is an identity that a user takes on to carry out a certain set of tasks. A role comes with several advantages and rights. A user is a distinct customer that belongs to a domain. Projects or domains are expressly linked to a user through a role. Access to OpenStack resources is denied to a user who is not assigned any roles.

3. What is a hypervisor and which hypervisors are supported in OpenStack?

   A hypervisor is a piece of computer hardware or firmware that provides virtual machine instances for cloud computing. There are several kinds of hypervisors, which are explained below: KVM, VMware, Containers, Xen, and Hyper-V:

   QEMU-KVM: This hypervisor runs on top of the Kernel-based Virtual Machine (KVM) and serves as its virtualization layer.

   VMware: This type is classified as an enterprise-class version of the ESXi hypervisor.

   Containers: Specifically created for containerized systems, containers represent a cloud-native hypervisor.

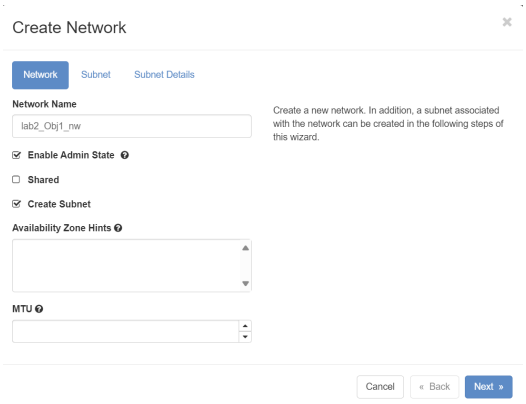
   Xen: Referred to as a microkernel-style hypervisor, Xen's purpose is to offer services that enable the simultaneous operation of several computer operating systems on the same hardware.

   Hyper-V: Also referred to as Windows Server Virtualization, this hypervisor is used to build virtual machines on the Windows OS for x86-64 platforms.

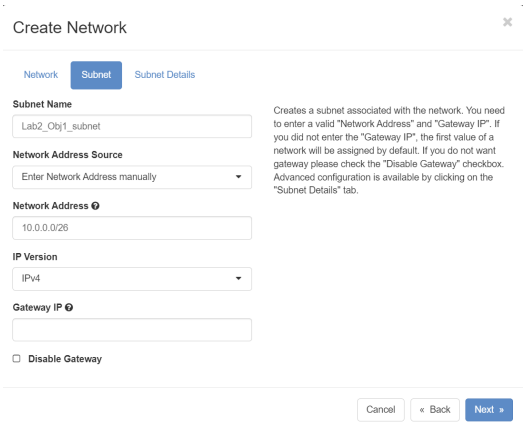
4. Explain the meaning of ‘flavor’ in OpenStack.

The computation, memory, and storage capabilities of nova computing instances in OpenStack are defined by flavors. In short, a flavor is a server's available hardware configuration. The maximum size of a virtual server that can be launched is specified. The compute host on which a flavor can be utilized to start an instance can also be determined by flavors.

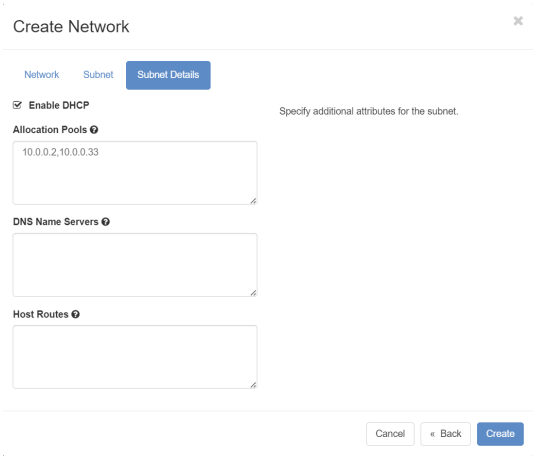
5. Create a new network of 64 IP addresses in the Network tab and enable DHCP for 32 of the IPs using either the GUI or the CLI.



In this part, we need to define the new network name and enable the admin state to bring the network up. Also, if we check the create subnet option, we need to define a subnet in the next step.



If we marked the “enable subnet” in the previous session, here we have to set the subnet name for the network as well as the address source, (we can use either the predefined address source which is 10.0.0.0/26 or we can create a new source).
Add the network address, which IP version we are going to use for this subnet (IPv6 or IPv4 and also the IP address of the default gateway for our instances.

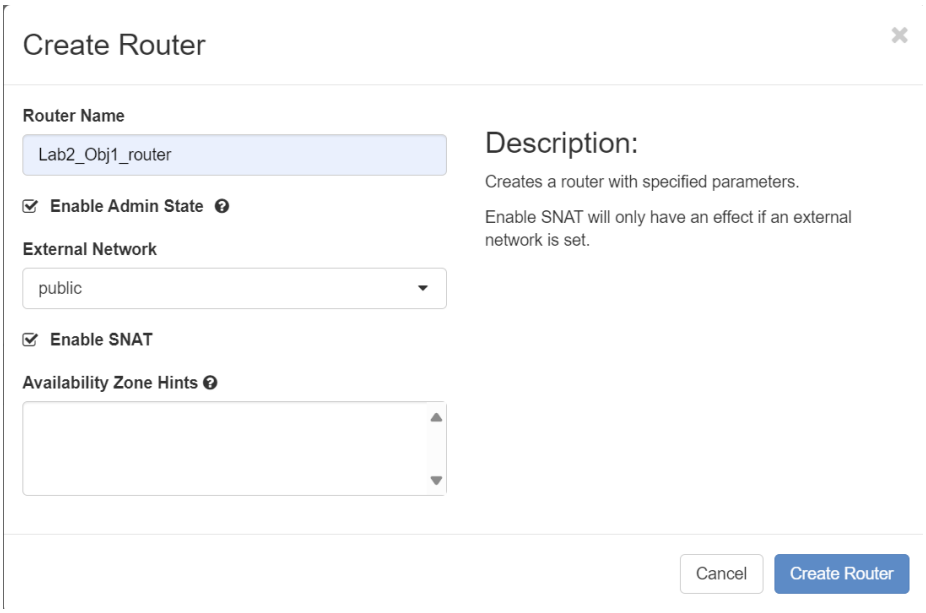


According to the requirement of this assignment we need to make the network as a kind of DHCP server to give ip addresses automatically to the instances. We need to set the first ip in the pool and the last ip in the pool separated with a comma.

The next field is the DNS servers.

In the last field we can add static routes for the hosts or instances. The format is : network/subnet, next-hop

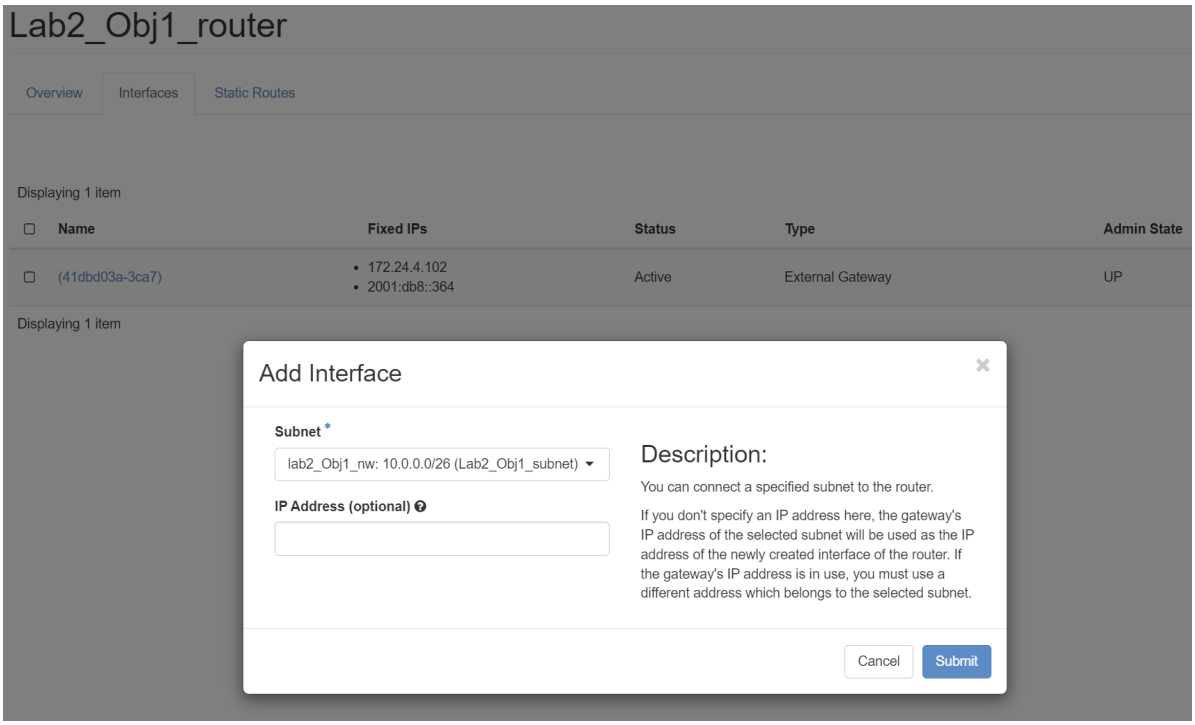
6. Create a router that connects this new network with the existing “public network” using either the GUI or the CLI.



By adding a router we can connect different networks to each other, first we need to set a name for the router and enable the admin state to bring the ports up.

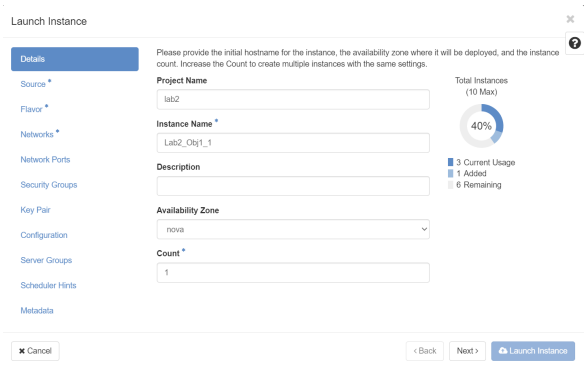
If we have multiple external networks, we can choose the network from the “External Network” dropdown list to connect to the router.

Keep the enable SNAT to translate source traffic from the host machine.

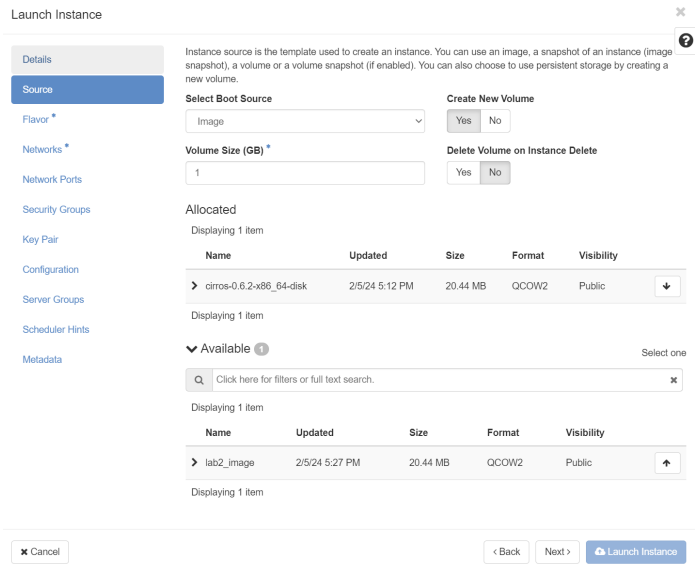


Each router must have an interface to work as the default gateway of the instances in the same network.

7. Start two instances with the Cirros image present that connects to the new network of 64 IPs using either the GUI or the CLI.



After clicking the Launch instance, in the first step of the wizard, we need to give a name for the instance, if we have multiple availability zones, we need to select that also.

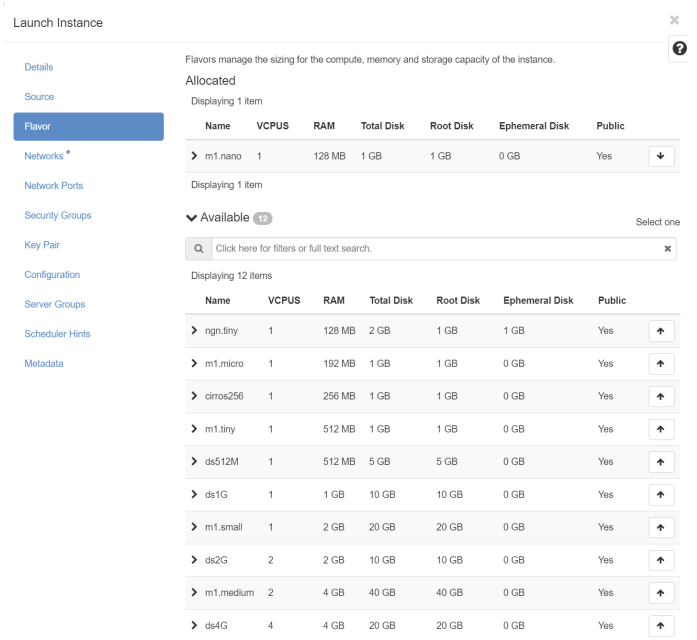


Next step is the image and flavor selection,

The boot source can be an image, (any linux image) or volume, (if we have already created a bootable volume) or we can choose a snapshot.

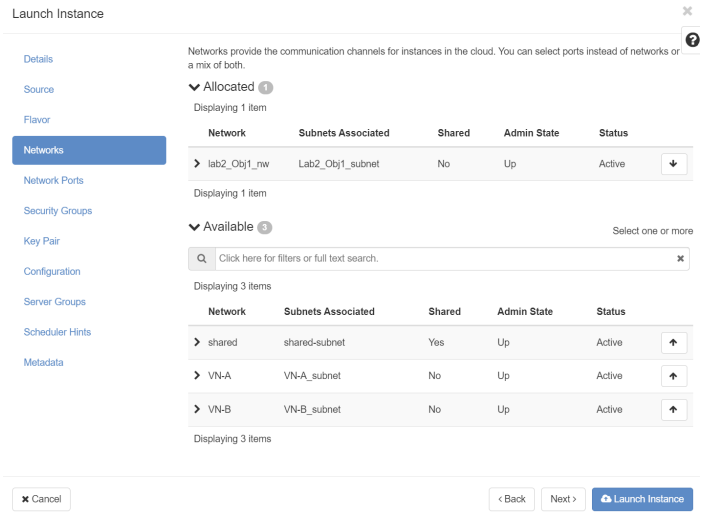
next, field select the size based on the image type and flavor we are going to use.

next, select the image file,



Next step is the flavor selection,

based on the number of the devices and instances and the amount of the hardware resources we have in our project we can choose a flavor from the list.



In the Networks step, we have to define to which network this instance should connect to and select the desired network from the list and for this part of the assignment these configurations would be enough.

Click launch instance and wait for the build process to complete and the state of the instance becomes “running”.