**Objective 1:**

1. **To create new network of 64 IP’s and enabling DHCP for half, we give subnet mask of /64. Under Project 🡪 Networks 🡪 Create Network:**

A screenshot of a computer

Description automatically generated

Figure 1: Creating a New Network

A screenshot of a computer

Description automatically generated

Figure 2: Modifying Subnet for New Network

A screenshot of a computer

Description automatically generated

Figure 3: Allocating 32 IP Addresses in DHCP Pool

A screenshot of a computer

Description automatically generated

Figure 4: Allocating 32 IP Addresses in DHCP Pool

1. For creating a router that connects this new network with the existing “public network” using either the GUI, we go to Project 🡪 Network 🡪 Routers -🡪 Create New Router

A screenshot of a computer

Description automatically generated

Figure 5: New Router Created

A screenshot of a computer

Description automatically generated

Figure 6: New Network attached to Router

1. For starting two instances with the Cirros image present that connects to the new network of 64 IPs using either the GUI A screenshot of a computer

   Description automatically generated

Figure 7: Creating New Instance

A screenshot of a computer

Description automatically generated

Figure 8: New Instance Source Information

A screenshot of a computer

Description automatically generated

Figure 9: New Network attached to New Instance

A screenshot of a computer

Description automatically generated

Figure 10: New Test Instance of Cirros Created

A screenshot of a computer

Description automatically generated

Figure 11: Cirros Test Image showing uptime and IP via DHCP

**Associate Floating IP:**

1. Select Associate Floating IP

**A screenshot of a computer

Description automatically generated**

1. **Allocate new Floating IP: *Click on Allocate IP*  
   A screenshot of a computer

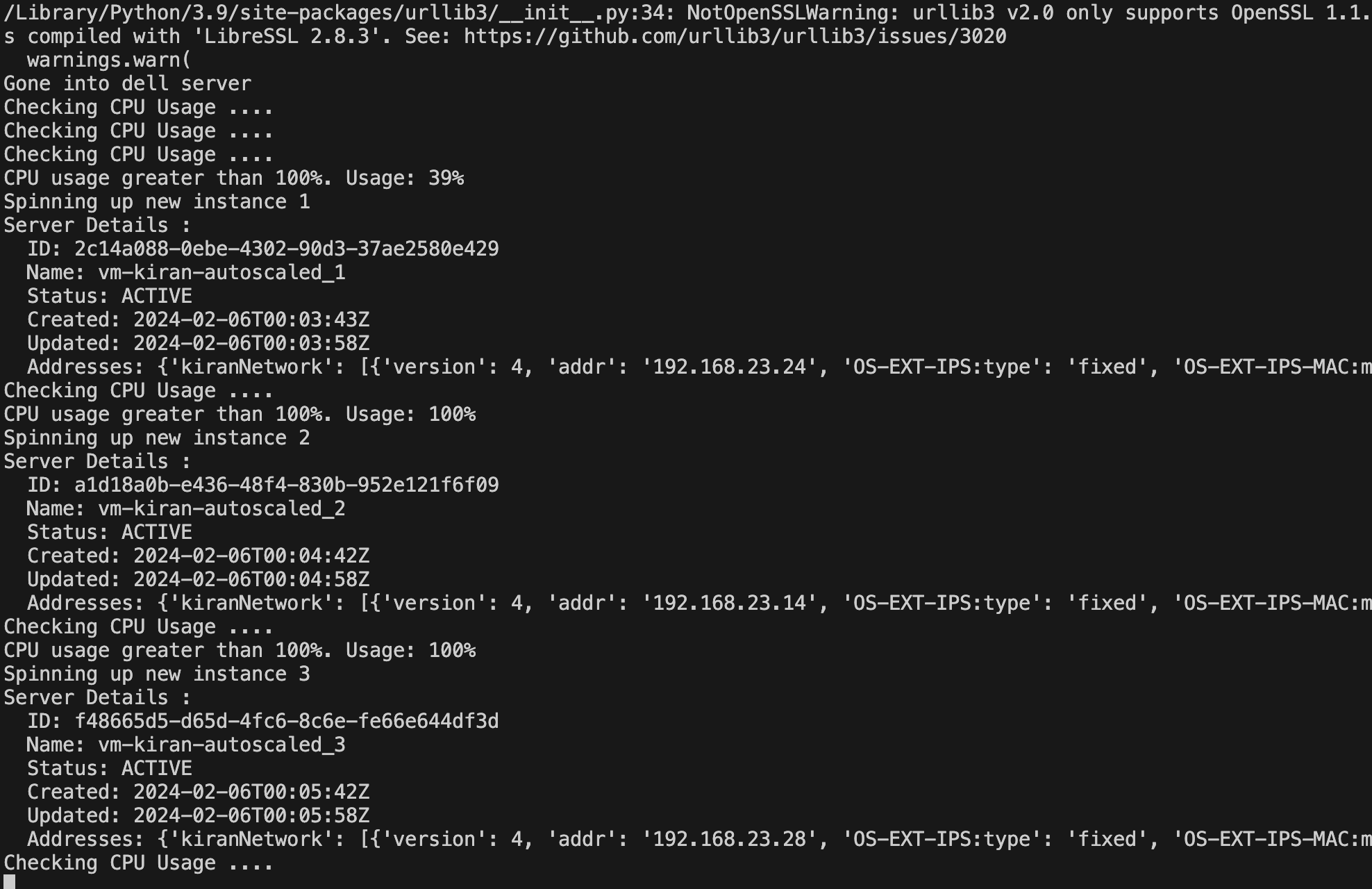
   Description automatically generated**
2. Click on Associate  **A screenshot of a computer

   Description automatically generated**

**Objective 2:**

**Algorithm -**

1. **Associate Floating IP to the VM to be monitored.**
2. **Go to the jump host via netmiko and then ssh further into Virtual Machine.**
3. **Run top command and extract the CPU idle percentage.**
4. **Extract the details like flavor, image and network details from the VM to be monitored.**
5. **Use novaClient Python API to spin new instances with similar configurations.**



**Objective 3:**

Section 1: Creating project, user, flavor and image

* Login as admin
* Under Identity tab click project and create new project

A screenshot of a computer

Description automatically generated

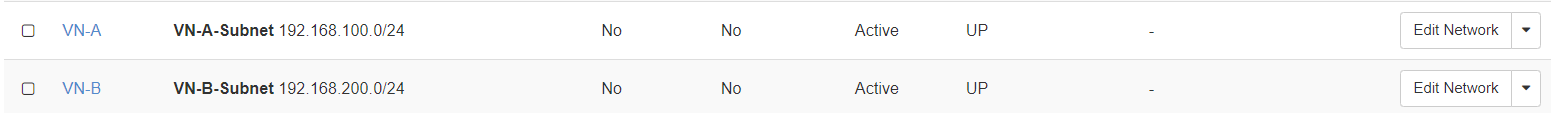
* Go to Users tab under Identity. Create the new user and assign that user to the project created above.

A screenshot of a computer

Description automatically generated

* Go to admin tab under that 🡪compute 🡪 Flavors. Create the flavor as per the requirements.
* Under the Flavor tab we can find the image tab. Create the image by uploading the file downloaded from the Internet. Almost we can use the default values. A white background with blue and red text

  Description automatically generated
* Under the Project main tab select the “Network tab” and configure the two Networks as below. We can configure DHCP by allocating the pool of IP. Don’t use the first IP as it will be allocated to the Gateway.



* Go to Compute 🡪 Instance 🡪 Launch Instance. It’s mandatory to fill the details, Source, Flavor, Networks. We can see the appropriate options under each section. Select as per requirement.
* Once the VMs are created VM1 and VM2 of Vnet A can ping each other. For Inter Vnet communication create a Router and attach one interface to “public” and other interfaces to Vnet-A and Vnet-B. Also create the floating Ips which helps the VMs to communicate over the Internet.
* We need to first create the Floating IP and then associate to the different VMs separately. For Floating IP to create it’s mandatory to create the Router beforehand.
* Ping Test

Ping from VN-A\_VM1 to VN-A\_VM2

A screenshot of a computer program

Description automatically generated

Ping from VN-A\_VM1 to VN-B\_VM1

A screenshot of a computer program

Description automatically generated

Ping from VN-A\_VM1 to VN-B\_VM1

A screenshot of a computer program

Description automatically generated

1. Assign floating IP’s to the VM’s both in VN-A and VN-B, and test connectivity to the Internet.

VN-A\_VM1

A screenshot of a computer

Description automatically generated

VN-A\_VM2

A screenshot of a computer

Description automatically generated

VN-B\_VM1

A screenshot of a computer

Description automatically generated

**Objective 4: Security Policy:**

* Create the below three security groups first

**A screenshot of a computer

Description automatically generated**

* **Inside each add the rules as below**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**🡪Associate the above security group to each of the respective VMs**