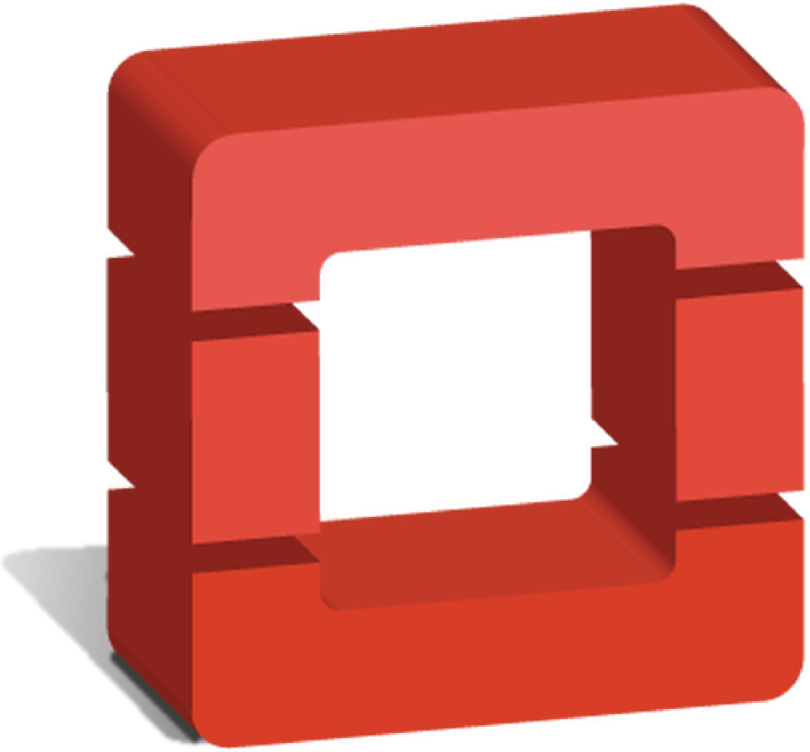
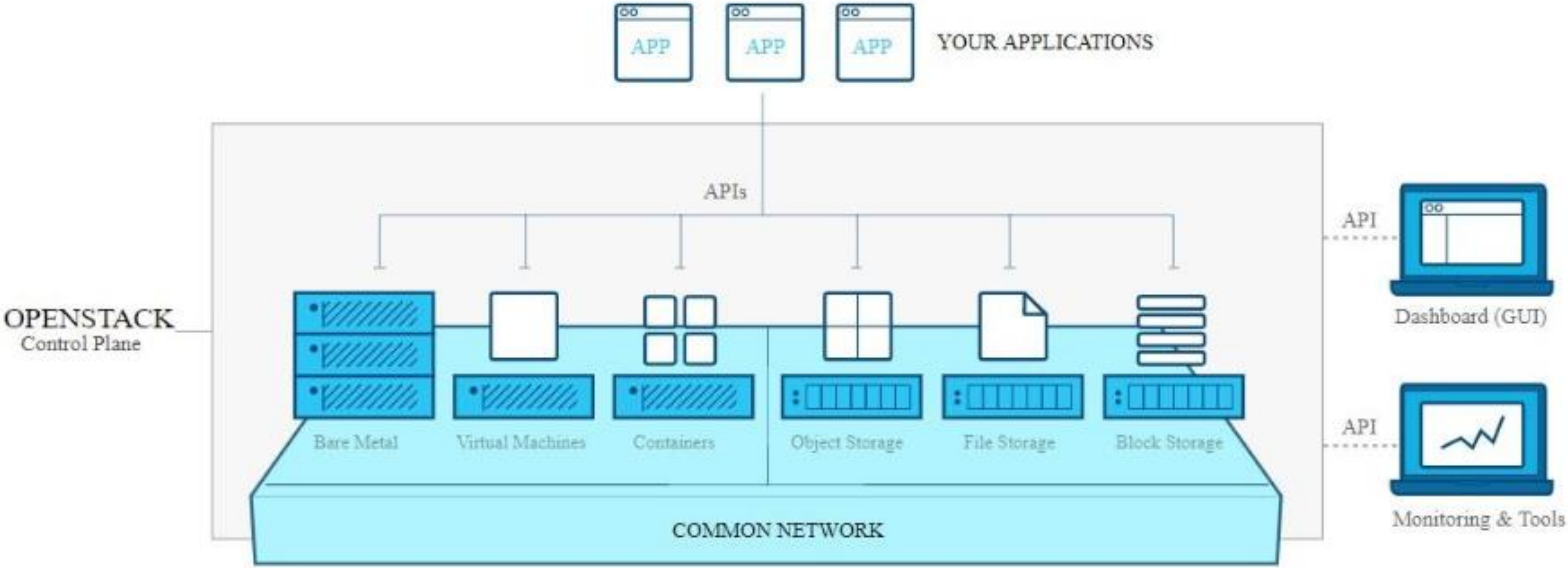


openstack™

CLOUD SOFTWARE

What is Openstack

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering the users to provision resources through a web interface.


APP
APP
APP
YOUR APPLICATIONS
APIs
API
Dashboard (GUI)
OPENSTACK
Control Plane
Bare Metal
Virtual Machines
Containers
Object Storage
File Storage
Block Storage
API
Monitoring & Tools
COMMON NETWORK

Introduction

- OpenStack began as a collaboration between NASA and Rackspace in 2010
- OpenStack is considered as Infrastructure as a Service (IaaS) making it easy for users to quickly add new instance.
- It is basically a set of tools for building and managing cloud computing platforms for public and private clouds.
- It is an open source software, which means that anyone who chooses to can access the source code, make any changes or modifications they need
- OpenStack is being run by research institutions, government agencies, financial institutions, e-commerce and media companies and biomedical companies globally in both public and private cloud environment.

Openstack Community

- The OpenStack Foundation promotes the global development, distribution and adoption of the OpenStack cloud operating system. The Foundation serves more than 60,000 Individual Members from over 180 countries around the world.

1,202

32%

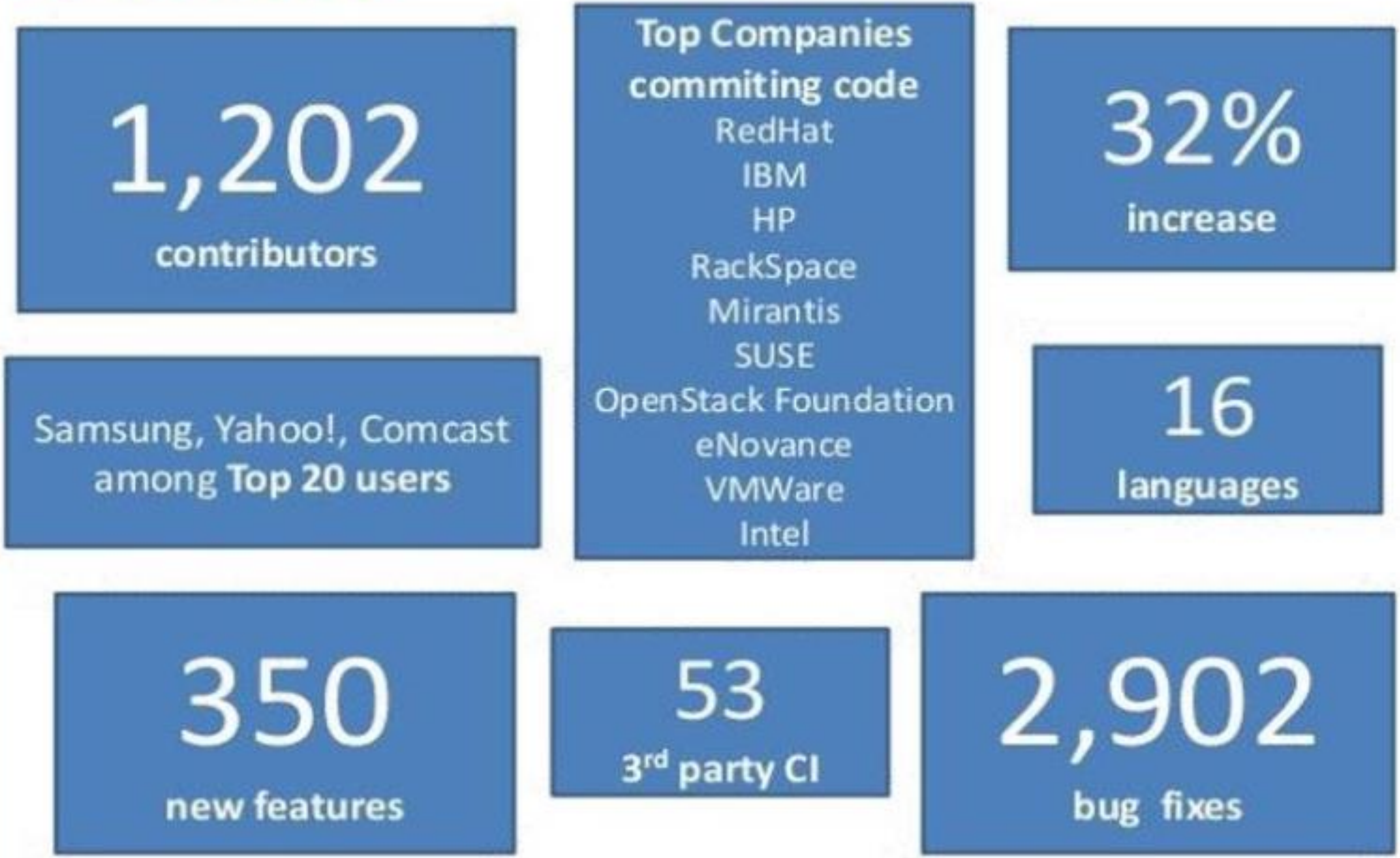
350

2,902

- Hundreds of the world's largest brands rely on OpenStack to run their businesses every day, reducing costs and helping them move faster.

- OpenStack has a strong ecosystem, and users seeking commercial support can choose from different OpenStack-powered products and services in the Marketplace.

By the numbers



1,202

contributors

Samsung, Yahoo!, Comcast among **Top 20 users**

Top Companies commiting code

RedHat
IBM
HP
RackSpace
Mirantis
SUSE
OpenStack Foundation
eNovance
VMWare
Intel

32%

increase

16

languages

350

new features

53

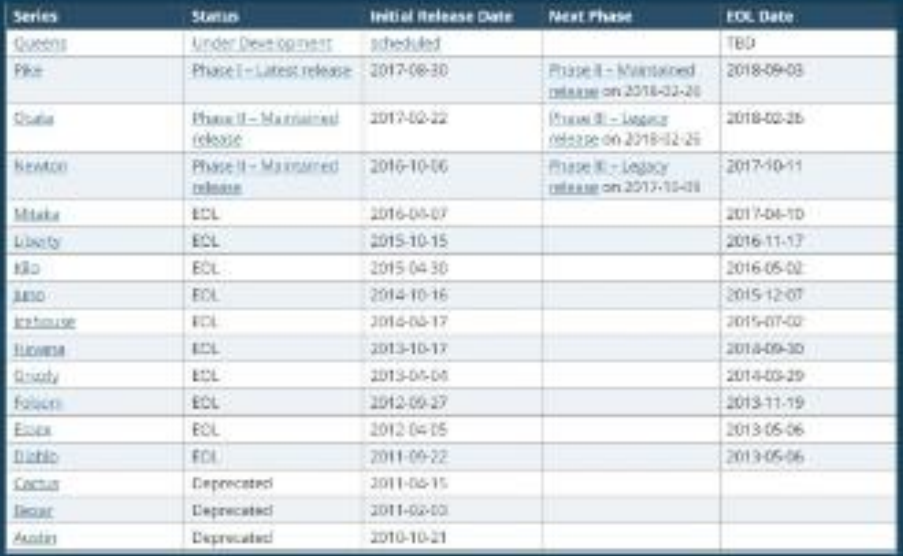
3rd party CI

2,902

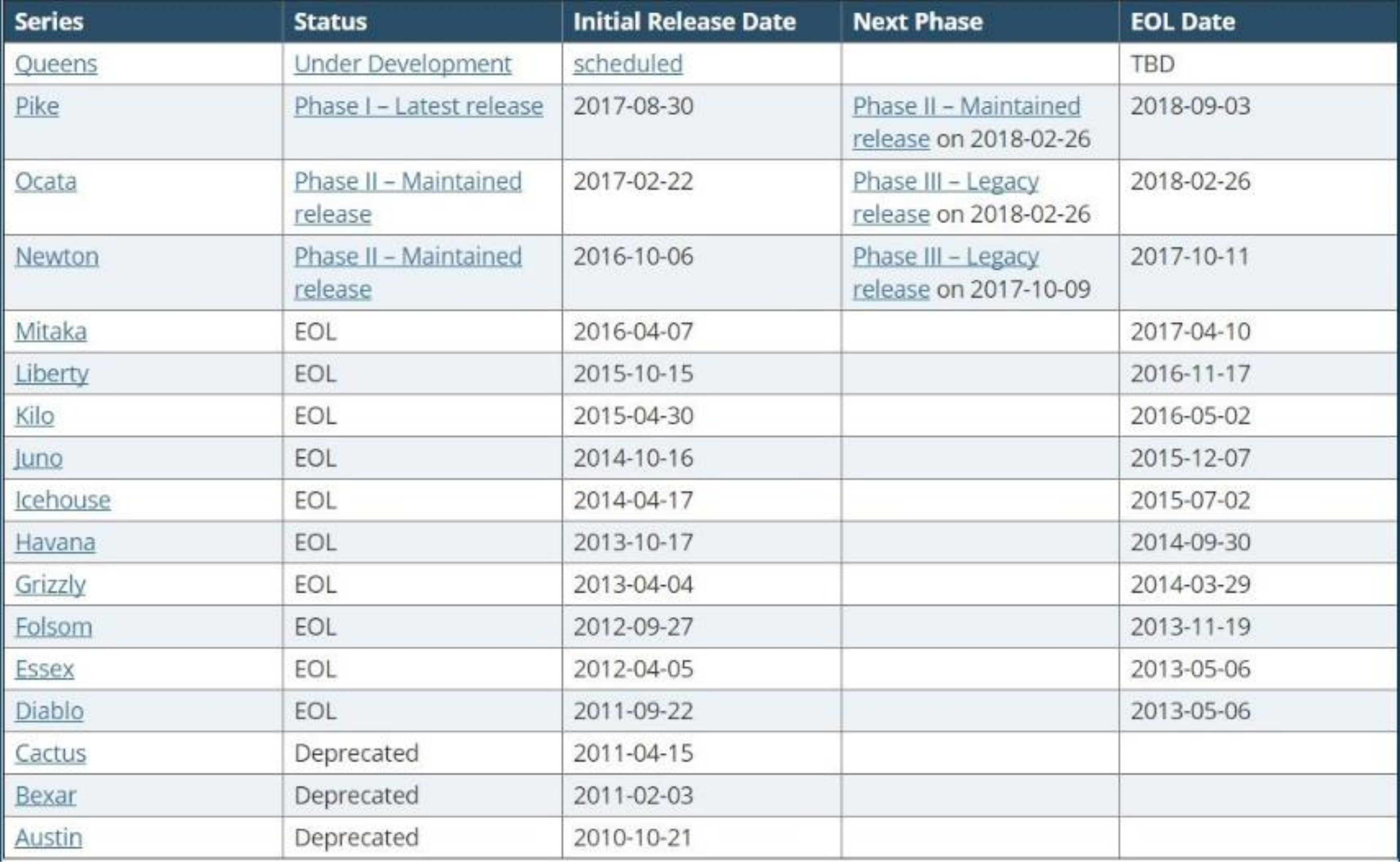
bug fixes

Openstack Versions

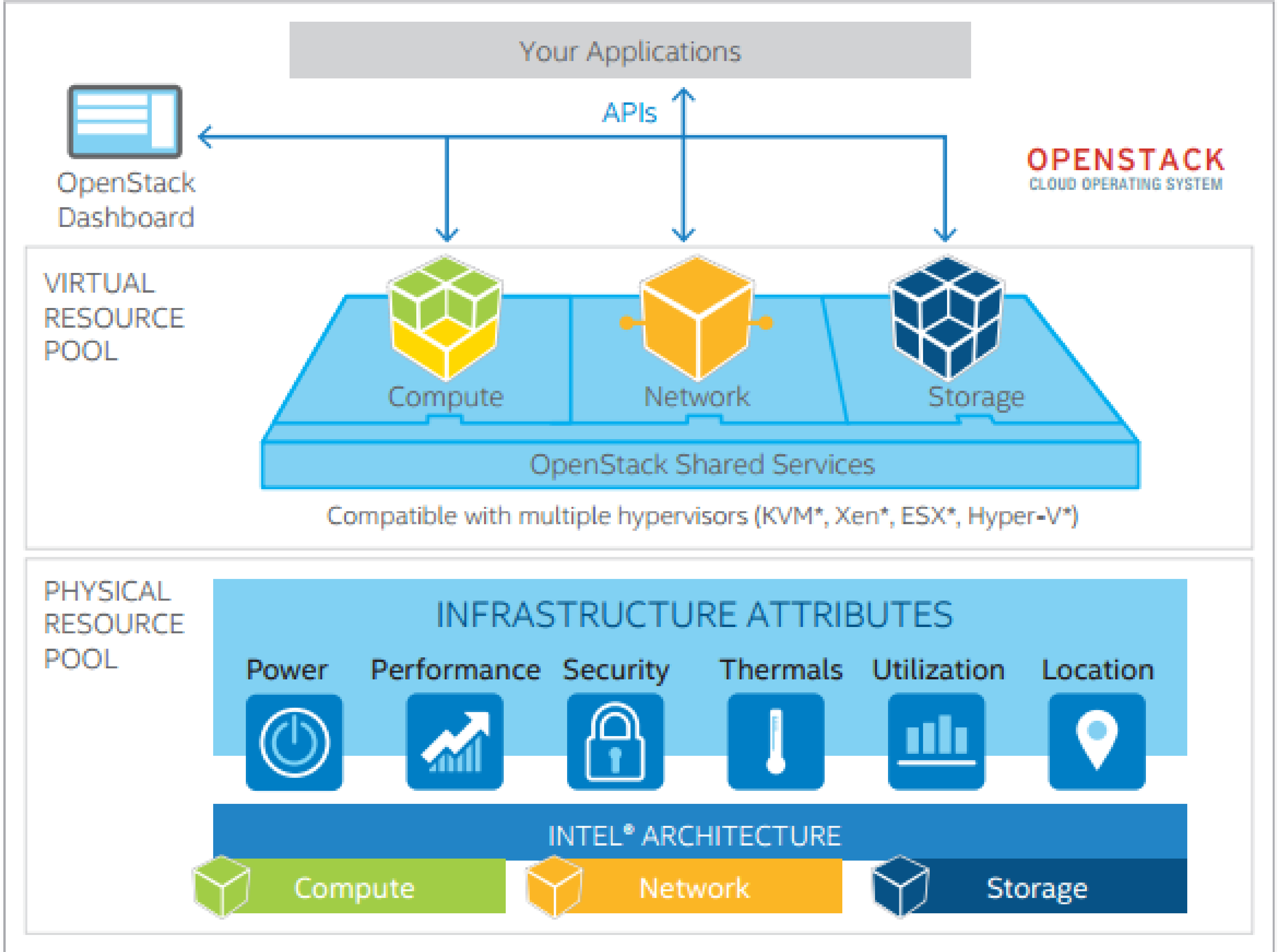
- OpenStack is developed and released around 6-month cycles. After the initial release, additional stable point releases will be released in each release series.



Series	Status	Initial Release Date	Next Phase	EOL Date
Queens	Under Development	scheduled		TBD
Pike	Phase I – Latest release	2017-08-30	Phase II – Maintained release on 2018-02-26	2018-09-03
Ocata	Phase II – Maintained release	2017-02-22	Phase III – Legacy release on 2018-02-26	2018-02-26
Newton	Phase II – Maintained release	2016-10-06	Phase III – Legacy release on 2017-10-09	2017-10-11
Mitaka	EOL	2016-04-07		2017-04-10
Liberty	EOL	2015-10-15		2016-11-17
Kilo	EOL	2015-04-30		2016-05-02
Juno	EOL	2014-10-16		2015-12-07
Icehouse	EOL	2014-04-17		2015-07-02
Havana	EOL	2013-10-17		2014-09-30
Grizzly	EOL	2013-04-04		2014-03-29
Folsom	EOL	2012-09-27		2013-11-19
Essex	EOL	2012-04-05		2013-05-06
Diablo	EOL	2011-09-22		2013-05-06
Cactus	Deprecated	2011-04-15		
Bexar	Deprecated	2011-02-03		
Austin	Deprecated	2010-10-21		



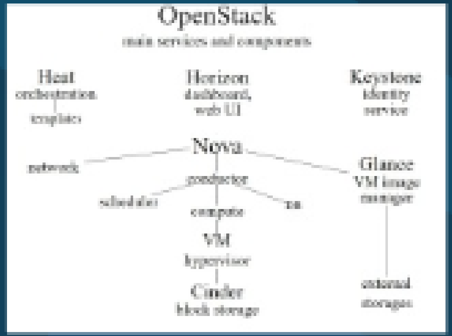
Series	Status	Initial Release Date	Next Phase	EOL Date
Queens	Under Development	scheduled		TBD
Pike	Phase I – Latest release	2017-08-30	Phase II – Maintained release on 2018-02-26	2018-09-03
Ocata	Phase II – Maintained release	2017-02-22	Phase III – Legacy release on 2018-02-26	2018-02-26
Newton	Phase II – Maintained release	2016-10-06	Phase III – Legacy release on 2017-10-09	2017-10-11
Mitaka	EOL	2016-04-07		2017-04-10
Liberty	EOL	2015-10-15		2016-11-17
Kilo	EOL	2015-04-30		2016-05-02
Juno	EOL	2014-10-16		2015-12-07
Icehouse	EOL	2014-04-17		2015-07-02
Havana	EOL	2013-10-17		2014-09-30
Grizzly	EOL	2013-04-04		2014-03-29
Folsom	EOL	2012-09-27		2013-11-19
Essex	EOL	2012-04-05		2013-05-06
Diablo	EOL	2011-09-22		2013-05-06
Cactus	Deprecated	2011-04-15		
Bexar	Deprecated	2011-02-03		
Austin	Deprecated	2010-10-21		

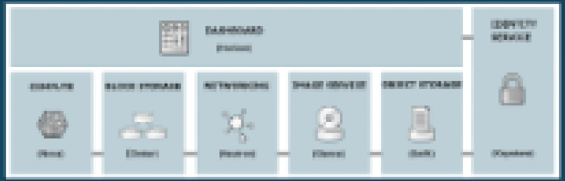

Your Applications
APIs
OpenStack Dashboard
OPENSTACK
CLOUD OPERATING SYSTEM
VIRTUAL RESOURCE POOL
Compute
Network
Storage
OpenStack Shared Services
Compatible with multiple hypervisors (KVM*, Xen*, ESX*, Hyper-V*)
PHYSICAL RESOURCE POOL
INFRASTRUCTURE ATTRIBUTES
Power
Performance
Security
Thermals
Utilization
Location
INTEL® ARCHITECTURE
Compute
Network
Storage

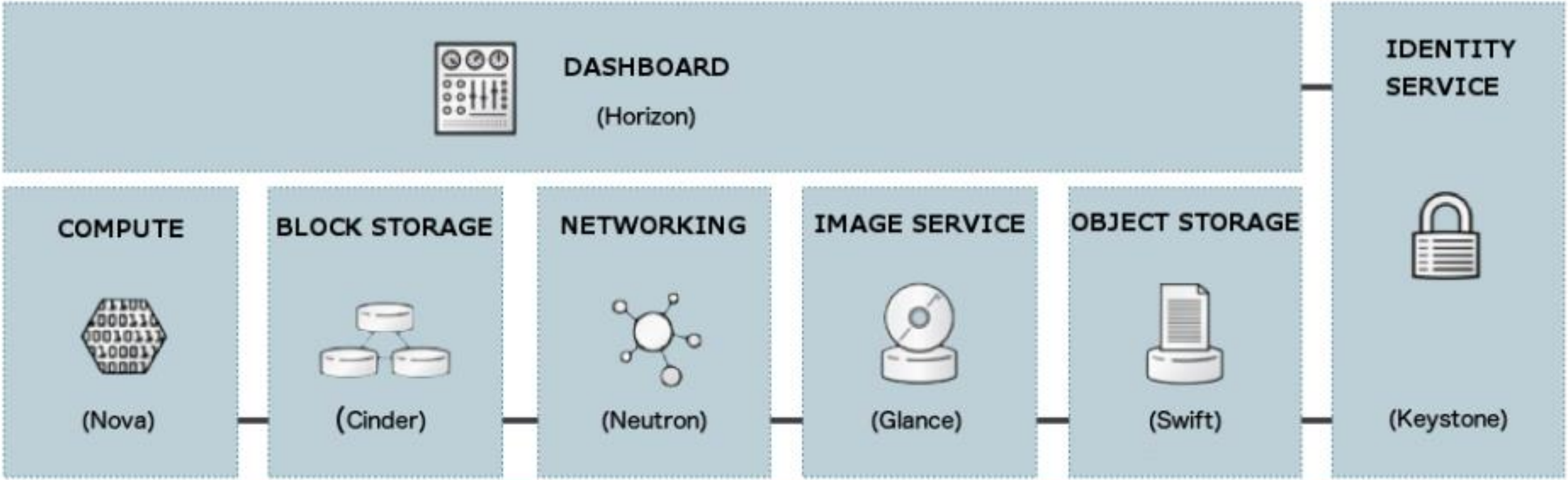
Openstack Components

- OpenStack has a modular architecture with various code names for its components

1. OpenStack Computation (Nova)
2. Openstack Dashboard (Horizion)
3. OpenStack Image Service (Glance)
4. OpenStack Storage (Swift)
5. Identity Management (Keystone)
6. Creating Network (Neutron)
7. Load Balancer (Octavia)
8. Queueing (Rabbit)
9. Database (Trove)

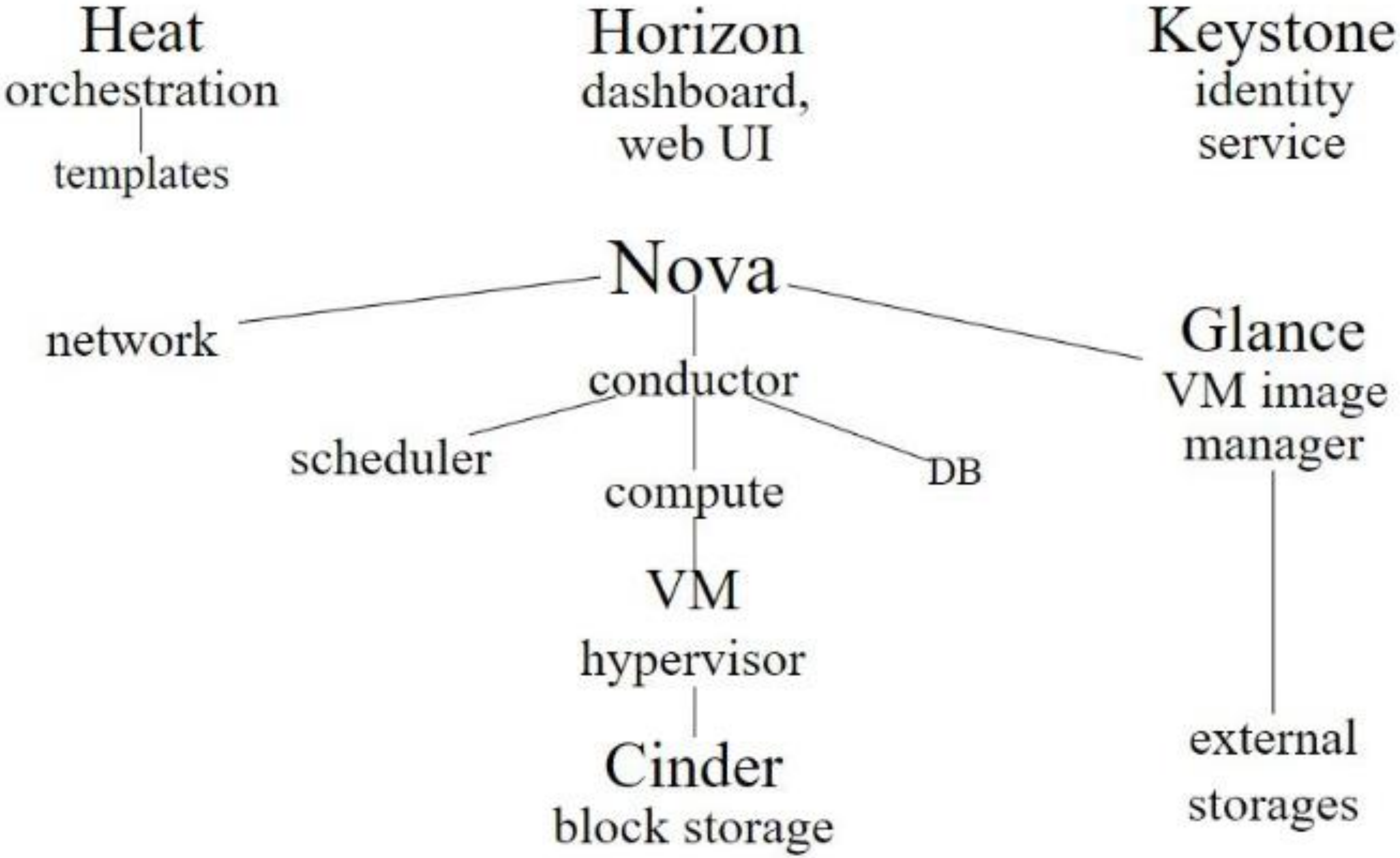





DASHBOARD
(Horizon)
IDENTITY SERVICE
(Keystone)
COMPUTE
(Nova)
BLOCK STORAGE
(Cinder)
NETWORKING
(Neutron)
IMAGE SERVICE
(Glance)
OBJECT STORAGE
(Swift)

OpenStack

main services and components



Heat
orchestration
templates

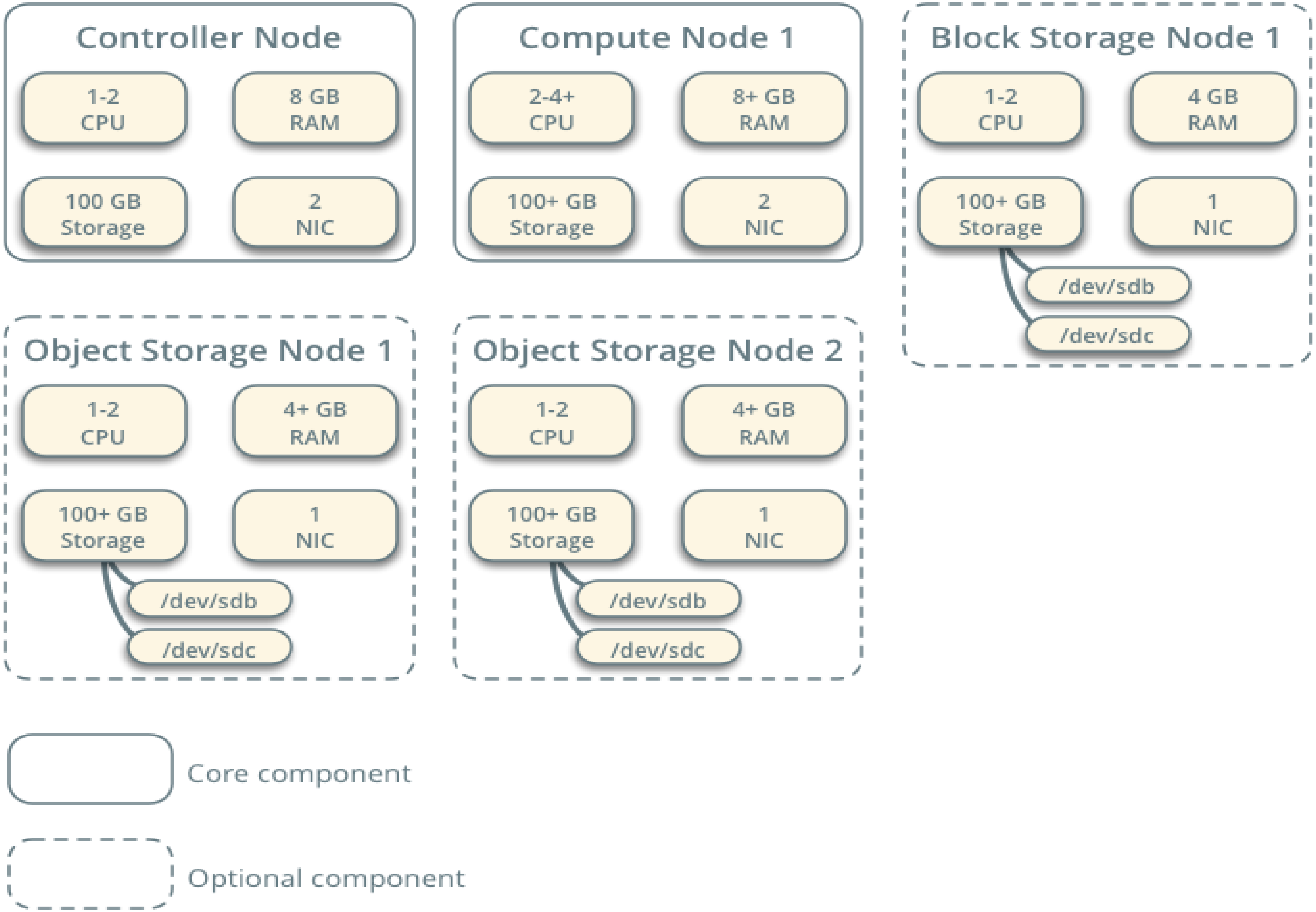
Horizon
dashboard,
web UI

Keystone
identity
service

Nova
network
conductor
scheduler
compute
DB
VM
hypervisor
Cinder
block storage

Glance
VM image
manager
external
storages

Hardware Requirements



Controller Node

- 1-2 CPU
- 8 GB RAM
- 100 GB Storage
- 2 NIC

Compute Node 1

- 2-4+ CPU
- 8+ GB RAM
- 100+ GB Storage
- 2 NIC

Block Storage Node 1

- 1-2 CPU
- 4 GB RAM
- 100+ GB Storage
  - /dev/sdb
  - /dev/sdc
- 1 NIC

Object Storage Node 1

- 1-2 CPU
- 4+ GB RAM
- 100+ GB Storage
  - /dev/sdb
  - /dev/sdc
- 1 NIC

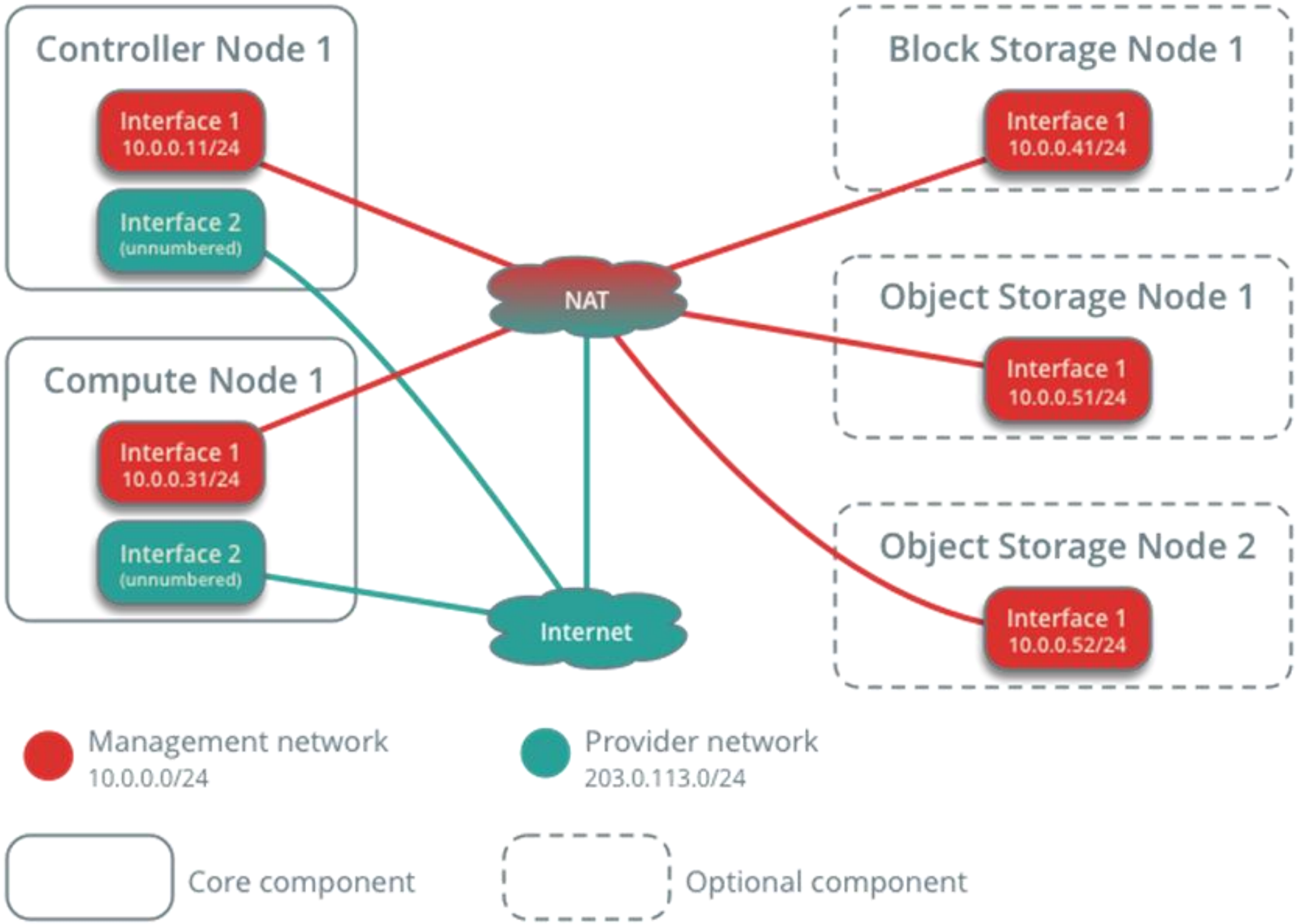
Object Storage Node 2

- 1-2 CPU
- 4+ GB RAM
- 100+ GB Storage
  - /dev/sdb
  - /dev/sdc
- 1 NIC

Core component

Optional component

Network Layout



Controller Node 1

Interface 1
10.0.0.11/24

Interface 2
(unnumbered)

Compute Node 1

Interface 1
10.0.0.31/24

Interface 2
(unnumbered)

NAT

Internet

Block Storage Node 1

Interface 1
10.0.0.41/24

Object Storage Node 1

Interface 1
10.0.0.51/24

Object Storage Node 2

Interface 1
10.0.0.52/24

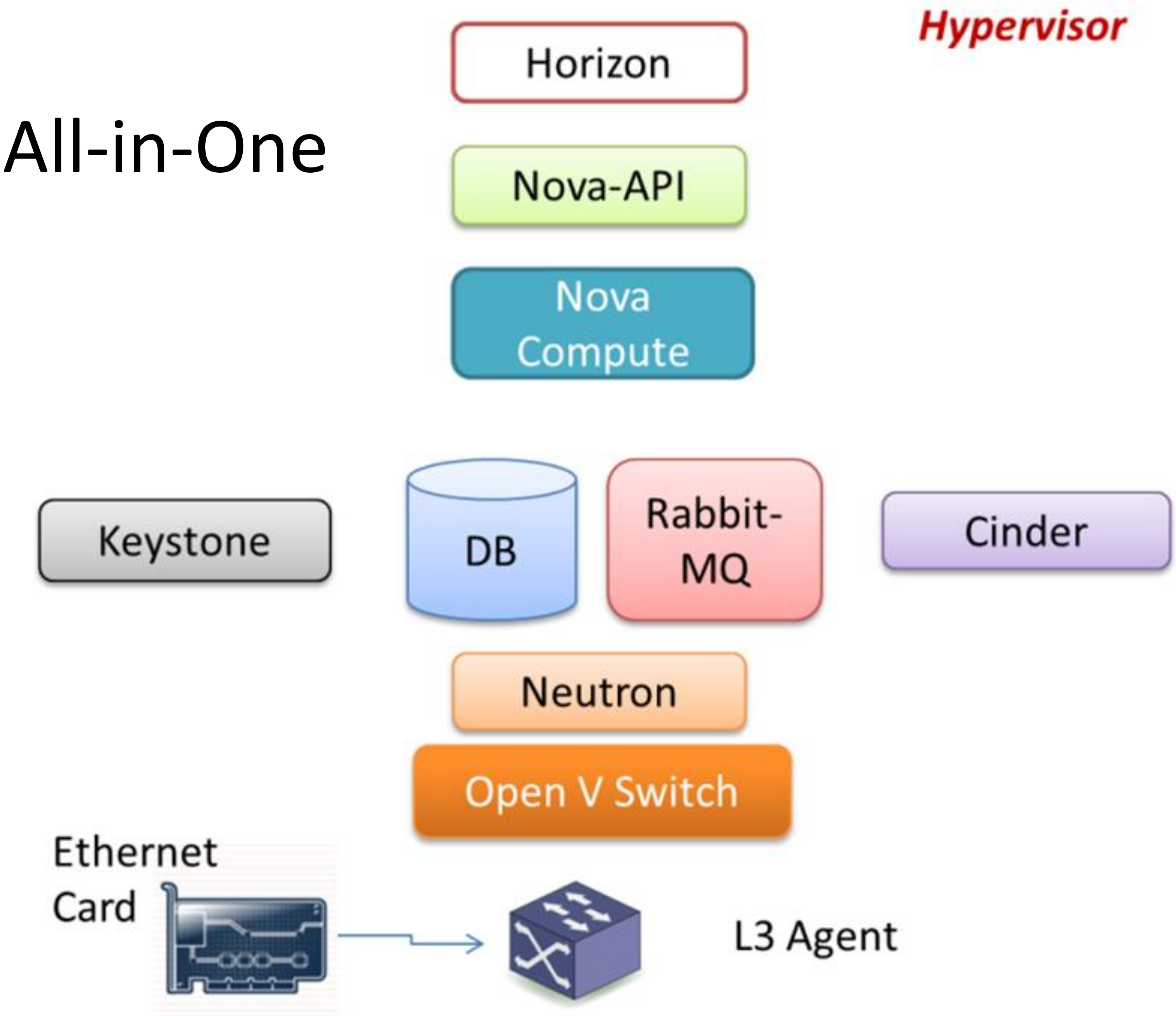
● Management network
10.0.0.0/24

● Provider network
203.0.113.0/24

□ Core component

⬚ Optional component

OpenStack All-in-One



Hypervisor

Horizon

Nova-API

Nova Compute

Keystone

DB

Rabbit-MQ

Cinder

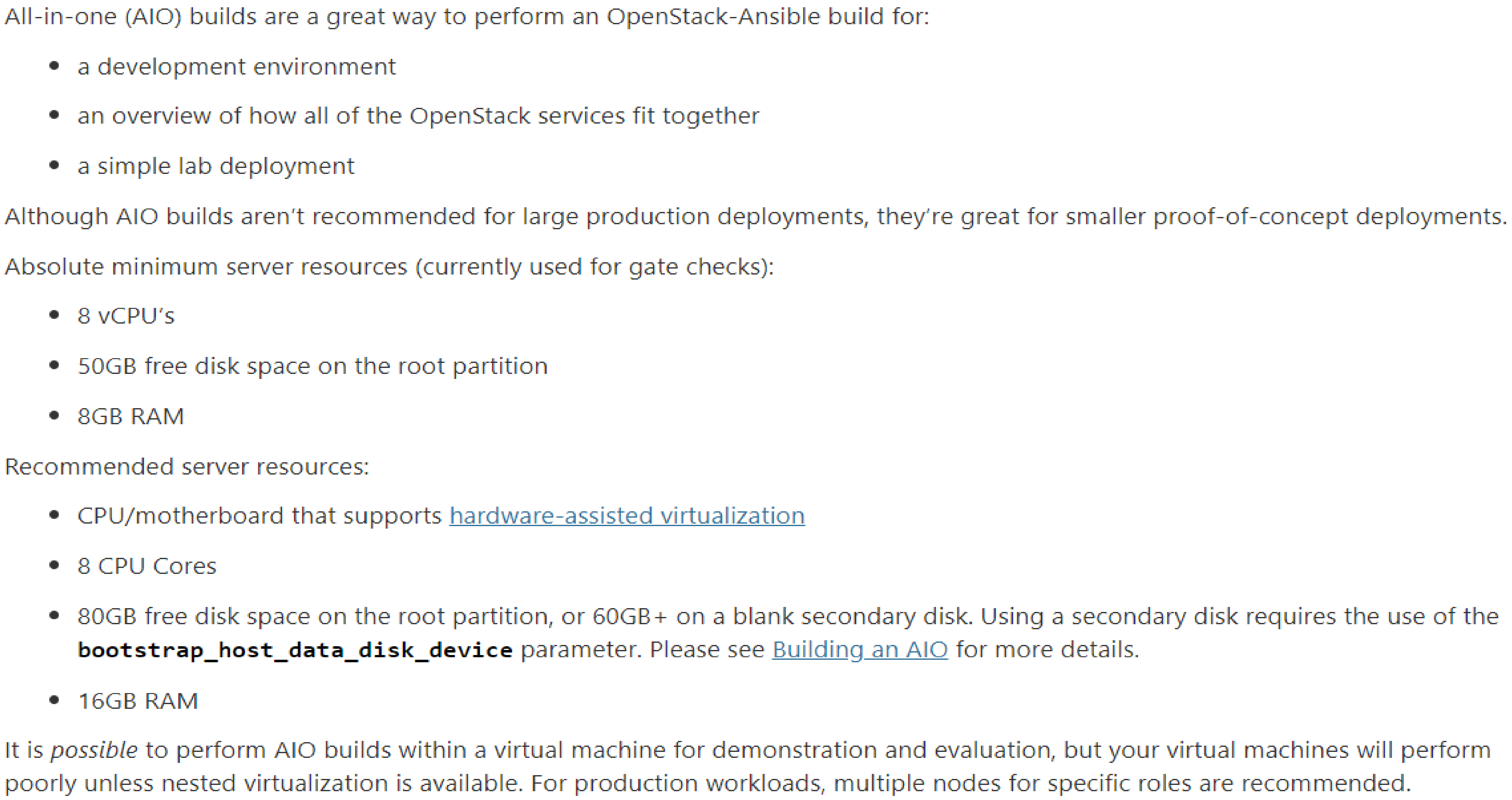
Neutron

Open V Switch

Ethernet Card

L3 Agent

Hardware requirement for All-in-One



All-in-one (AIO) builds are a great way to perform an OpenStack-Ansible build for:

- a development environment
- an overview of how all of the OpenStack services fit together
- a simple lab deployment

Although AIO builds aren't recommended for large production deployments, they're great for smaller proof-of-concept deployments.

Absolute minimum server resources (currently used for gate checks):

- 8 vCPU's
- 50GB free disk space on the root partition
- 8GB RAM

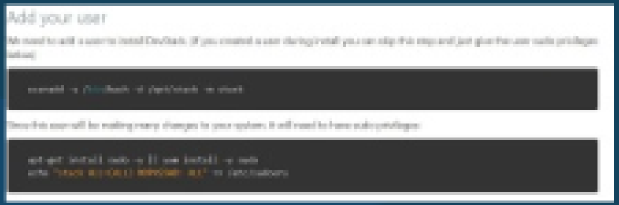
Recommended server resources:

- CPU/motherboard that supports hardware-assisted virtualization
- 8 CPU Cores
- 80GB free disk space on the root partition, or 60GB+ on a blank secondary disk. Using a secondary disk requires the use of the **bootstrap_host_data_disk_device** parameter. Please see Building an AIO for more details.
- 16GB RAM

It is *possible* to perform AIO builds within a virtual machine for demonstration and evaluation, but your virtual machines will perform poorly unless nested virtualization is available. For production workloads, multiple nodes for specific roles are recommended.

Installing Openstack

- Install Ubuntu on Virtual box and launch a Ubuntu Linux virtual box by using the latest Ubuntu operating system.
- After installing Ubuntu update the operating system with latest system updates by using commands.
  sudo apt-get upgrade
  sudo apt-get update
- Next install Git using command
  sudo apt-get install git



Add your user

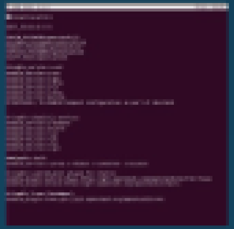
We need to add a user to install DevStack. (if you created a user during install you can skip this step and just give the user sudo privileges below)

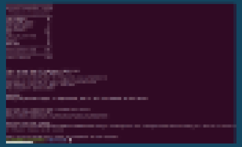
```
useradd -s /bin/bash -d /opt/stack -m stack
```

Since this user will be making many changes to your system, it will need to have sudo privileges:

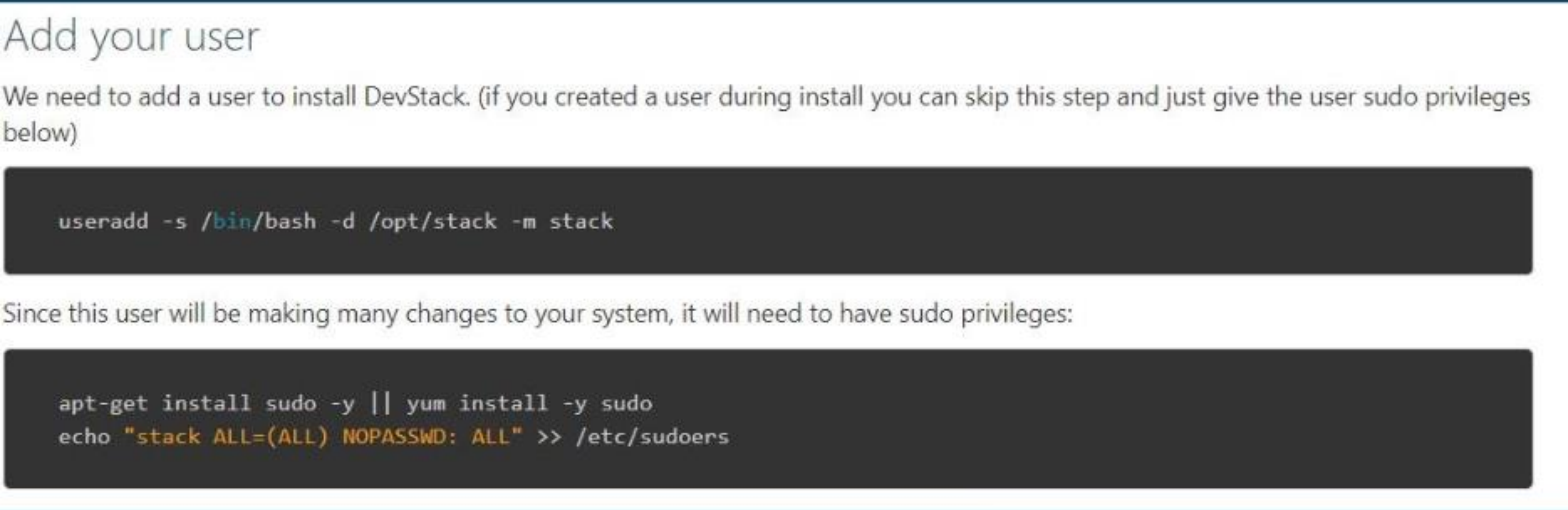
```
apt-get install sudo -y || yum install -y sudo
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

- Now login to the new user stack by
  sudo su - stack
- Now clone the OpenStack repository from git to your local device by using command
  git clone https://git.openstack.org/openstack-dev/devstack
- In the next step enter the devstack directory which you just cloned from github
  cd devstack
- In the devstack directory we need to create a configuration file local.conf before we install OpenStack.
  sudo nano local.conf





sudo apt-get install git



Add your user

We need to add a user to install DevStack. (if you created a user during install you can skip this step and just give the user sudo privileges below)

```
useradd -s /bin/bash -d /opt/stack -m stack
```

Since this user will be making many changes to your system, it will need to have sudo privileges:

```
apt-get install sudo -y || yum install -y sudo
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

- Now login to the new user stack by

  ```
  sudo su - stack
  ```

- Now clone the OpenStack repository from git to your local device by using command

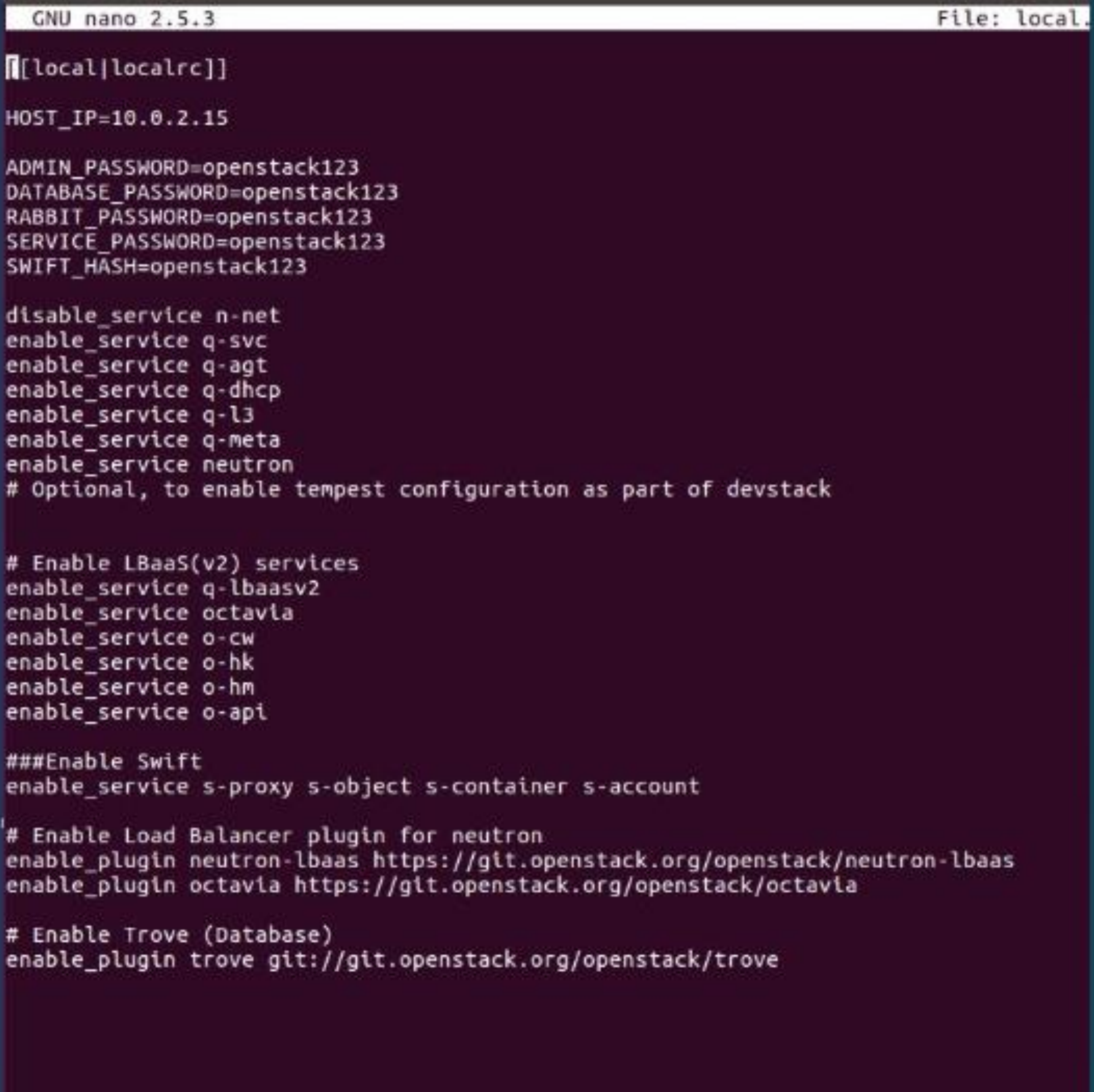
  ```
  git clone https://git.openstack.org/openstack-dev/devstack
  ```

- In the next step enter the devstack directory which you just cloned from github

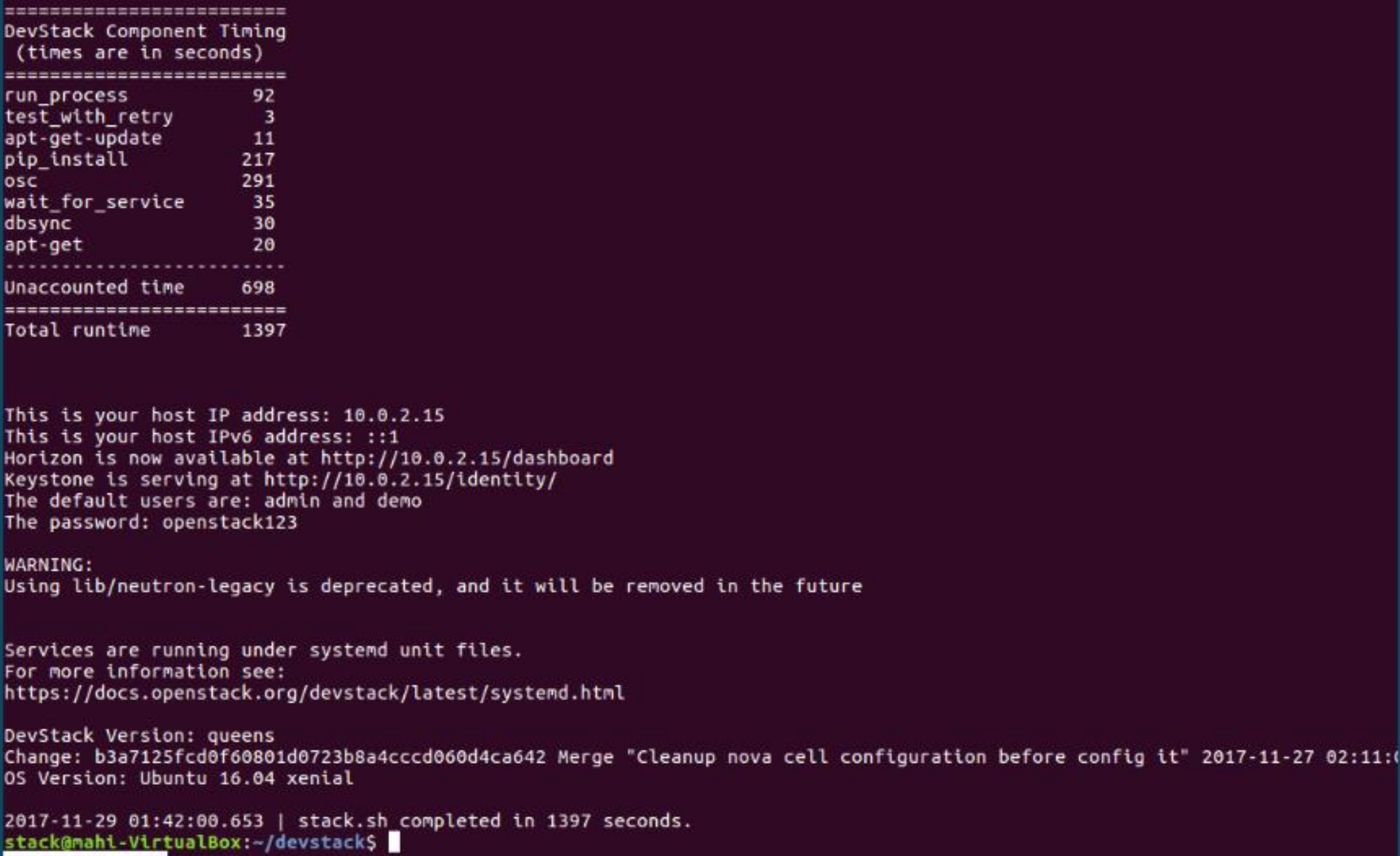
  ```
  cd devstack
  ```

- In the devstack directory we need to create a configuration file local.conf before we install OpenStack.

  ```
  sudo nano local.conf
  ```



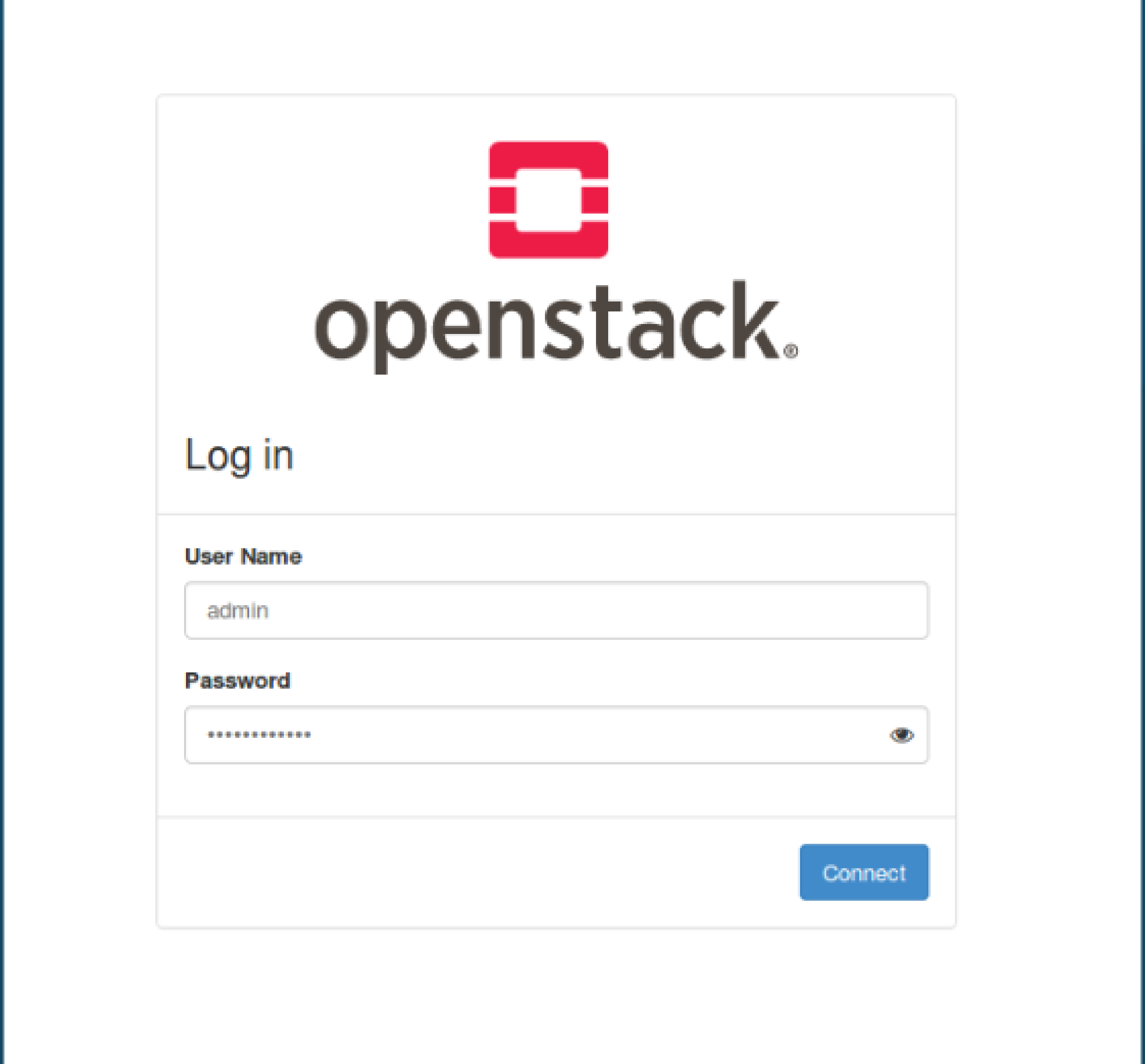
```
[[local|localrc]]

HOST_IP=10.0.2.15

ADMIN_PASSWORD=openstack123
DATABASE_PASSWORD=openstack123
RABBIT_PASSWORD=openstack123
SERVICE_PASSWORD=openstack123
SWIFT_HASH=openstack123

disable_service n-net
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service neutron
# Optional, to enable tempest configuration as part of devstack


# Enable LBaaS(v2) services
enable_service q-lbaasv2
enable_service octavia
enable_service o-cw
enable_service o-hk
enable_service o-hm
enable_service o-api

###Enable Swift
enable_service s-proxy s-object s-container s-account

# Enable Load Balancer plugin for neutron
enable_plugin neutron-lbaas https://git.openstack.org/openstack/neutron-lbaas
enable_plugin octavia https://git.openstack.org/openstack/octavia

# Enable Trove (Database)
enable_plugin trove git://git.openstack.org/openstack/trove
```



```
=========================
DevStack Component Timing
 (times are in seconds)
=========================
run_process            92
test_with_retry         3
apt-get-update         11
pip_install           217
osc                   291
wait_for_service       35
dbsync                 30
apt-get                20
-------------------------
Unaccounted time      698
=========================
Total runtime         1397



This is your host IP address: 10.0.2.15
This is your host IPv6 address: ::1
Horizon is now available at http://10.0.2.15/dashboard
Keystone is serving at http://10.0.2.15/identity/
The default users are: admin and demo
The password: openstack123

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future


Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: queens
Change: b3a7125fcd0f60801d0723b8a4cccd060d4ca642 Merge "Cleanup nova cell configuration before config it" 2017-11-27 02:11:0
OS Version: Ubuntu 16.04 xenial

2017-11-29 01:42:00.653 | stack.sh completed in 1397 seconds.
stack@mahi-VirtualBox:~/devstack$
```

Openstack Dashboard (Horizion)

- Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift, Keystone, etc.
- Now we can log in to our OpenStack console using horizon Dashboard service at http://10.0.2.15/dashboard
- And the user default users are admin and demo and use the password which you used in local.conf



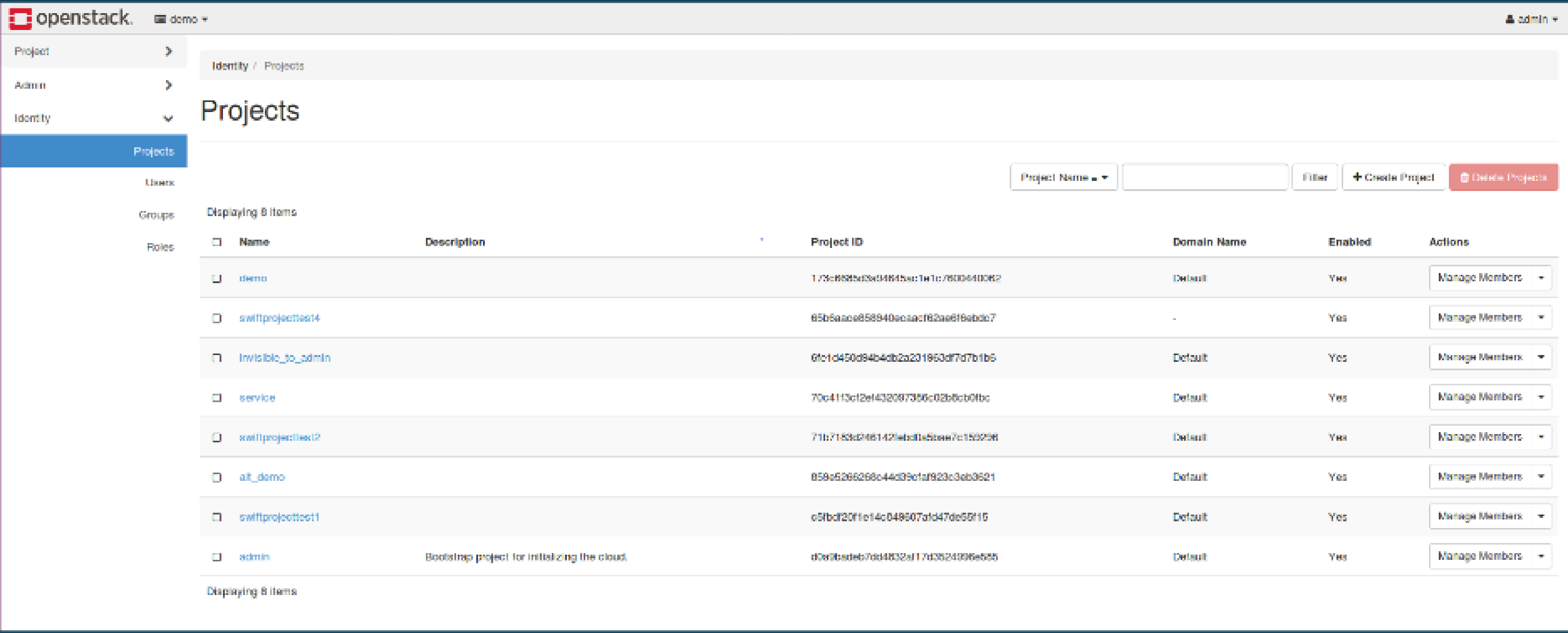
Log in

User Name

admin

Password

•••••••••••••



Identity / Projects

Projects

Displaying 8 items

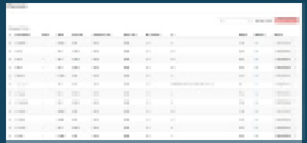
□	Name	Description	Project ID	Domain Name	Enabled	Actions
□	demo		173c6695d3a94645ac1e1c7600440062	Default	Yes	Manage Members
□	swiftprojecttest4		65b6aace858940ecaacf62ae616ebdc7	-	Yes	Manage Members
□	invisible_to_admin		6fc1d450d94b4db2a231963df7d7b1b6	Default	Yes	Manage Members
□	service		70c4113cf2ef432087356c02b8cb0fbc	Default	Yes	Manage Members
□	swiftprojecttest2		71b7183d2461421ebd1e5bee7c159296	Default	Yes	Manage Members
□	alt_demo		859e5266268c44d39cfaf923c3eb3621	Default	Yes	Manage Members
□	swiftprojecttest1		c5fbdf20f1e14c048607afd47de55f15	Default	Yes	Manage Members
□	admin	Bootstrap project for initializing the cloud.	d0e9badeb7dd4832af17d3524396e555	Default	Yes	Manage Members

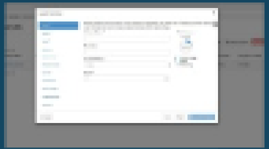
Displaying 8 items

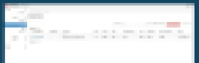
Openstack Compute (NOVA)

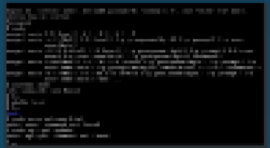
- Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers).
- Nova supports creating virtual machines, bare metal servers, and has limited support for system containers
- It requires the following additional OpenStack services for basic function:

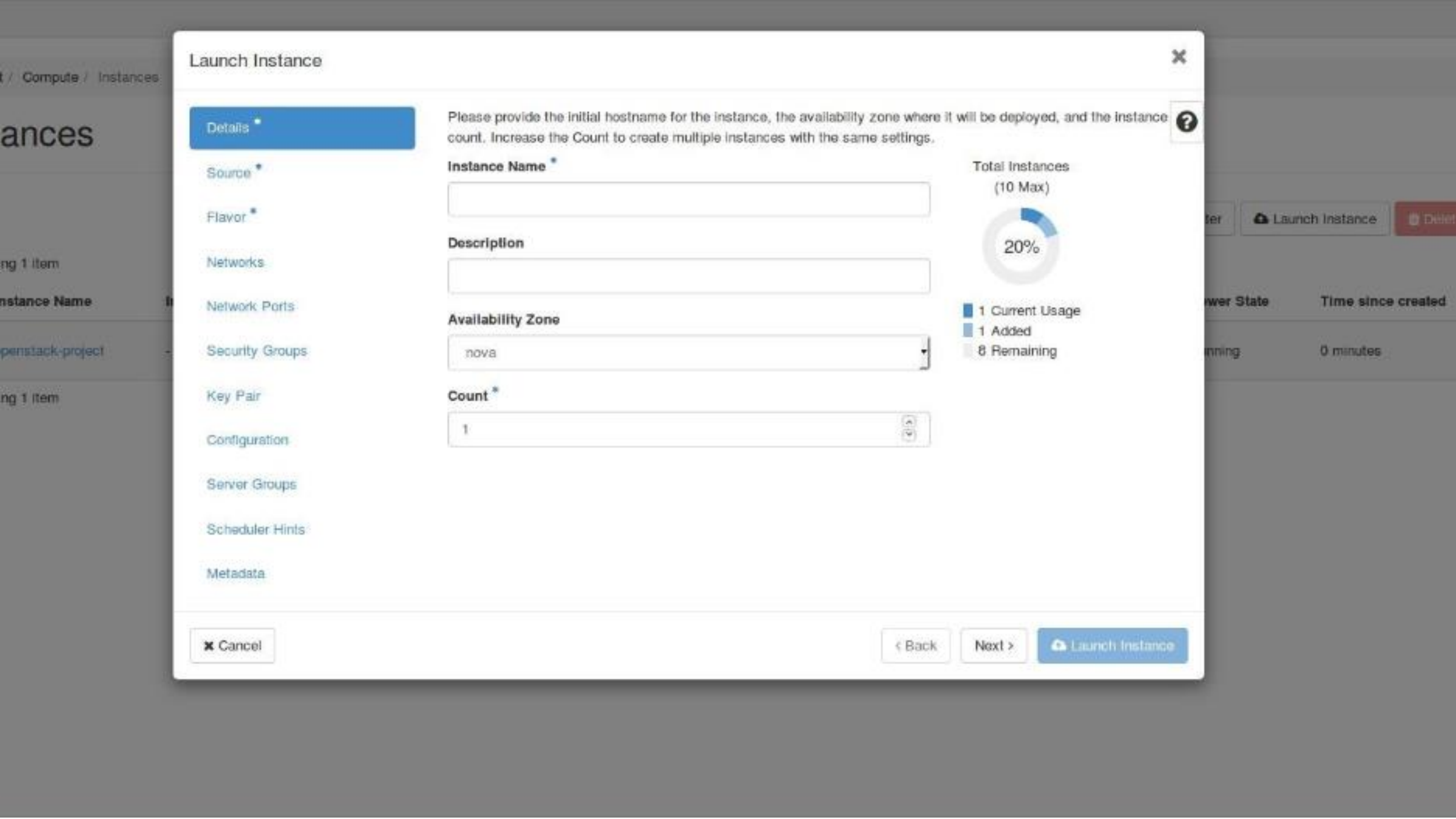
  Keystone
  Glance
  Neutron











Launch Instance

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name *

Description

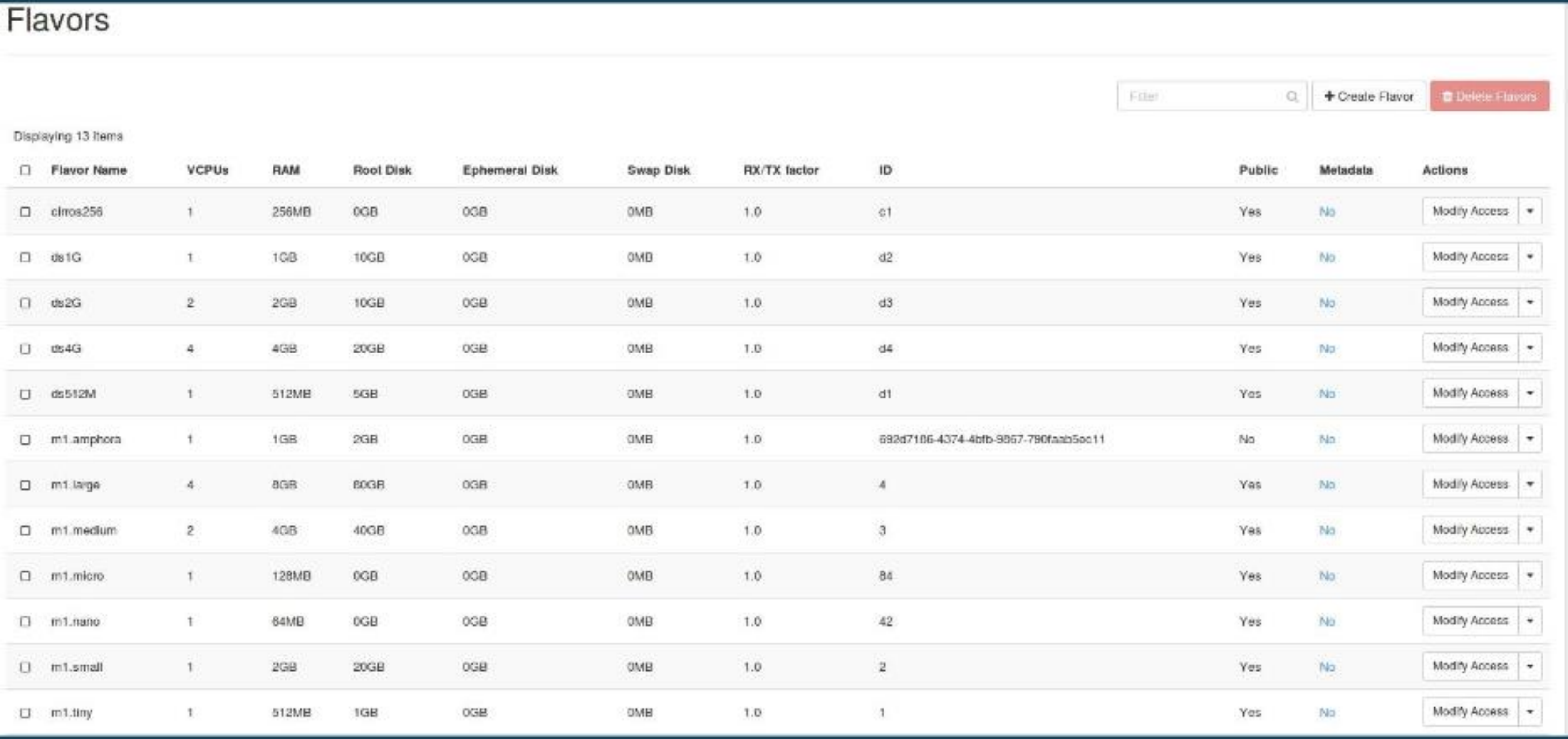
Availability Zone: nova

Count *: 1

Total Instances (10 Max)

20%

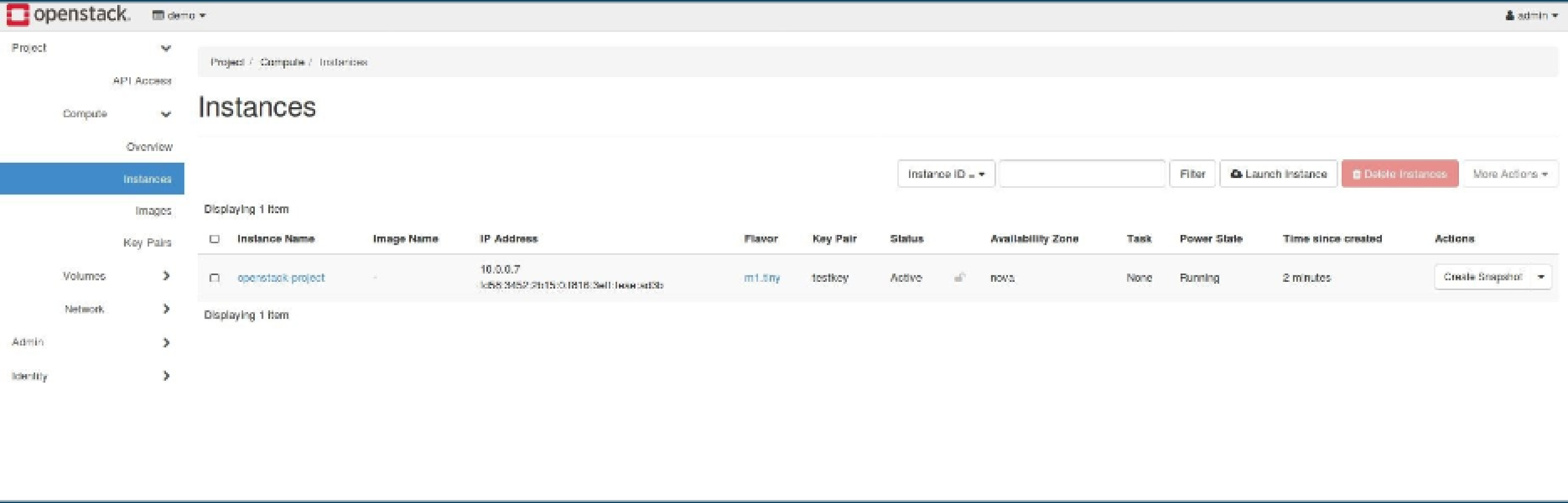
- 1 Current Usage
- 1 Added
- 8 Remaining



Flavors

Displaying 13 items

□	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
□	cirros256	1	256MB	0GB	0GB	0MB	1.0	c1	Yes	No	Modify Access ▾
□	ds1G	1	1GB	10GB	0GB	0MB	1.0	d2	Yes	No	Modify Access ▾
□	ds2G	2	2GB	10GB	0GB	0MB	1.0	d3	Yes	No	Modify Access ▾
□	ds4G	4	4GB	20GB	0GB	0MB	1.0	d4	Yes	No	Modify Access ▾
□	ds512M	1	512MB	5GB	0GB	0MB	1.0	d1	Yes	No	Modify Access ▾
□	m1.amphora	1	1GB	2GB	0GB	0MB	1.0	692d7106-4374-4bfb-9867-790faab5ec11	No	No	Modify Access ▾
□	m1.large	4	8GB	80GB	0GB	0MB	1.0	4	Yes	No	Modify Access ▾
□	m1.medium	2	4GB	40GB	0GB	0MB	1.0	3	Yes	No	Modify Access ▾
□	m1.micro	1	128MB	0GB	0GB	0MB	1.0	84	Yes	No	Modify Access ▾
□	m1.nano	1	64MB	0GB	0GB	0MB	1.0	42	Yes	No	Modify Access ▾
□	m1.small	1	2GB	20GB	0GB	0MB	1.0	2	Yes	No	Modify Access ▾
□	m1.tiny	1	512MB	1GB	0GB	0MB	1.0	1	Yes	No	Modify Access ▾



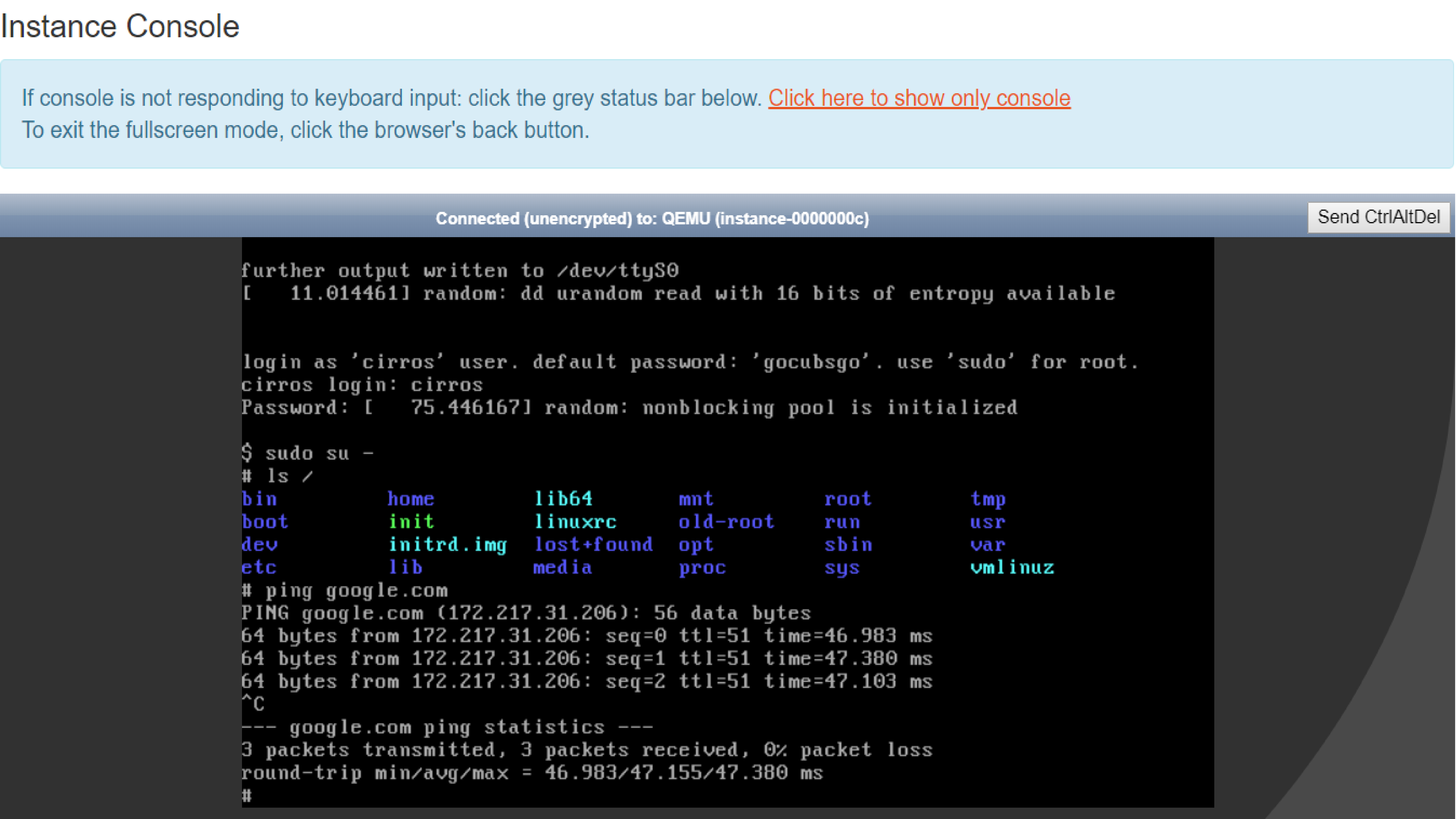
Project / Compute / Instances

Instances

Displaying 1 item

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	openstack-project	-	10.0.0.7 fd58:3452:2b15:0:f816:3eff:feee:a83b	m1.tiny	testkey	Active	nova	None	Running	2 minutes	Create Snapshot ▾

Displaying 1 item



Instance Console

If console is not responding to keyboard input: click the grey status bar below. Click here to show only console
To exit the fullscreen mode, click the browser's back button.

Connected (unencrypted) to: QEMU (instance-0000000c)

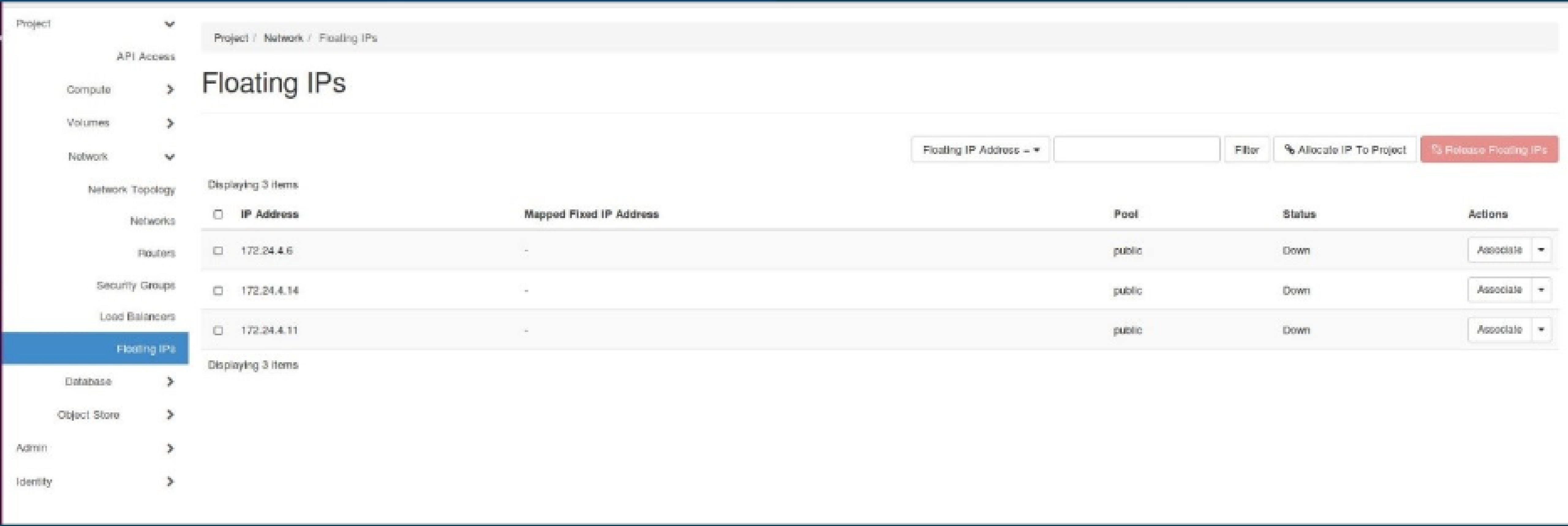
```
further output written to /dev/ttyS0
[   11.014461] random: dd urandom read with 16 bits of entropy available


login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root.
cirros login: cirros
Password: [   75.446167] random: nonblocking pool is initialized

$ sudo su -
# ls /
bin          home         lib64        mnt          root         tmp
boot         init         linuxrc      old-root     run          usr
dev          initrd.img   lost+found   opt          sbin         var
etc          lib          media        proc         sys          vmlinuz
# ping google.com
PING google.com (172.217.31.206): 56 data bytes
64 bytes from 172.217.31.206: seq=0 ttl=51 time=46.983 ms
64 bytes from 172.217.31.206: seq=1 ttl=51 time=47.380 ms
64 bytes from 172.217.31.206: seq=2 ttl=51 time=47.103 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 46.983/47.155/47.380 ms
#
```

Floating IP

- You can associate or change a floating IP address of a virtual machine any time after virtual machine creation. Although, OpenStack automatically assigns a private IP address to a VM at the moment of creation, you may want to assign a floating IP address, so that the VM can communicate with external networks

Security Groups

- Security groups are sets of IP filter rules that are applied to all project instances, which define networking access to the instance.



Project / Network / Floating IPs

Floating IPs

Displaying 3 items

□	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
□	172.24.4.6	-	public	Down	Associate
□	172.24.4.14	-	public	Down	Associate
□	172.24.4.11	-	public	Down	Associate

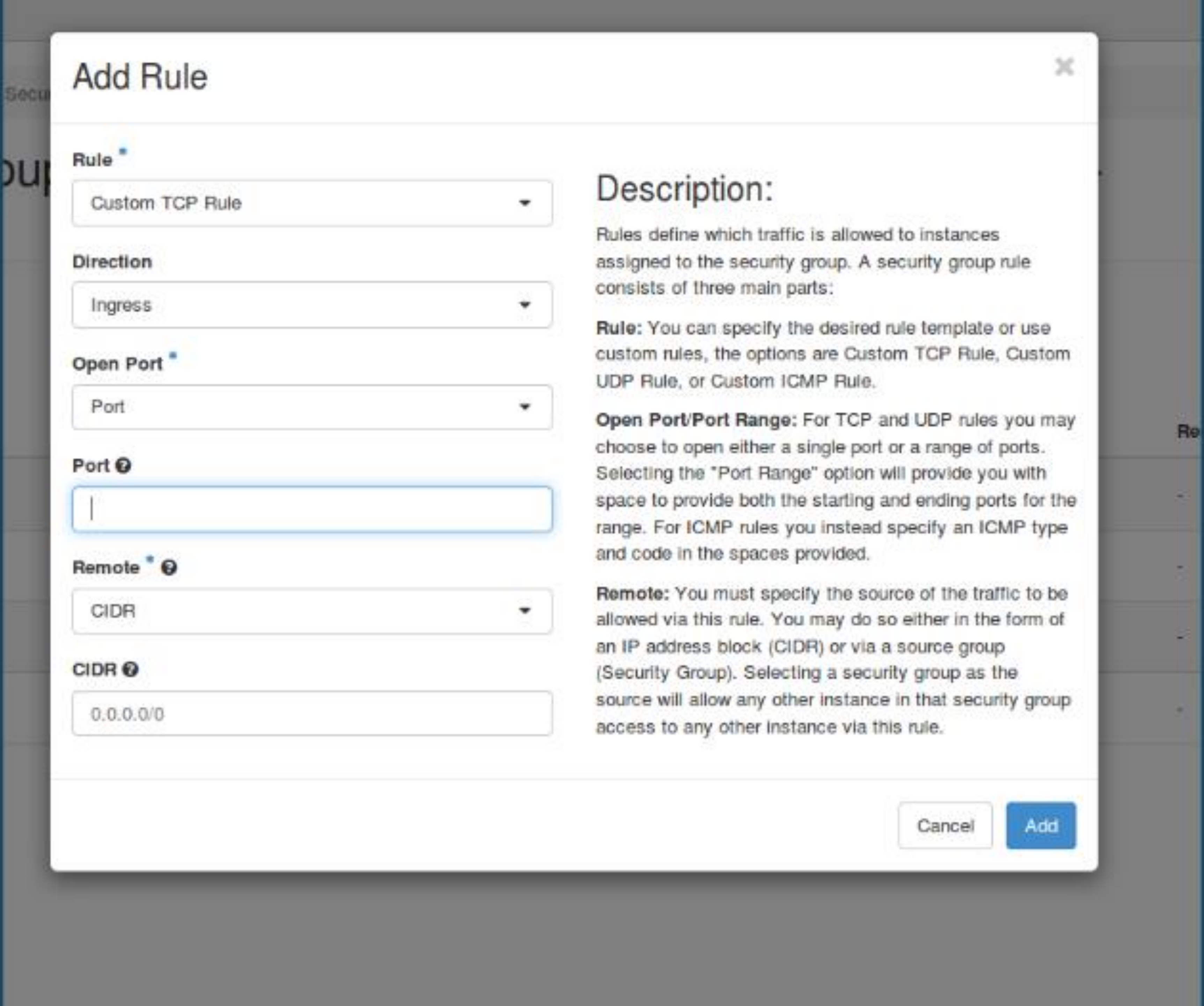
Displaying 3 items

private IP address to a VM at the moment of creation, you may want to assign a floating IP address, so that the VM can communicate with external networks

Security Groups

- Security groups are sets of IP filter rules that are applied to all project instances, which define networking access to the instance.

- All projects have a default security group which is applied to any instance that has no other defined security group. Unless you change the default, this security group denies all incoming traffic and allows only outgoing traffic to your instance.



Add Rule

Rule *

Custom TCP Rule

Direction

Ingress

Open Port *

Port

Port

Remote *

CIDR

CIDR

0.0.0.0/0

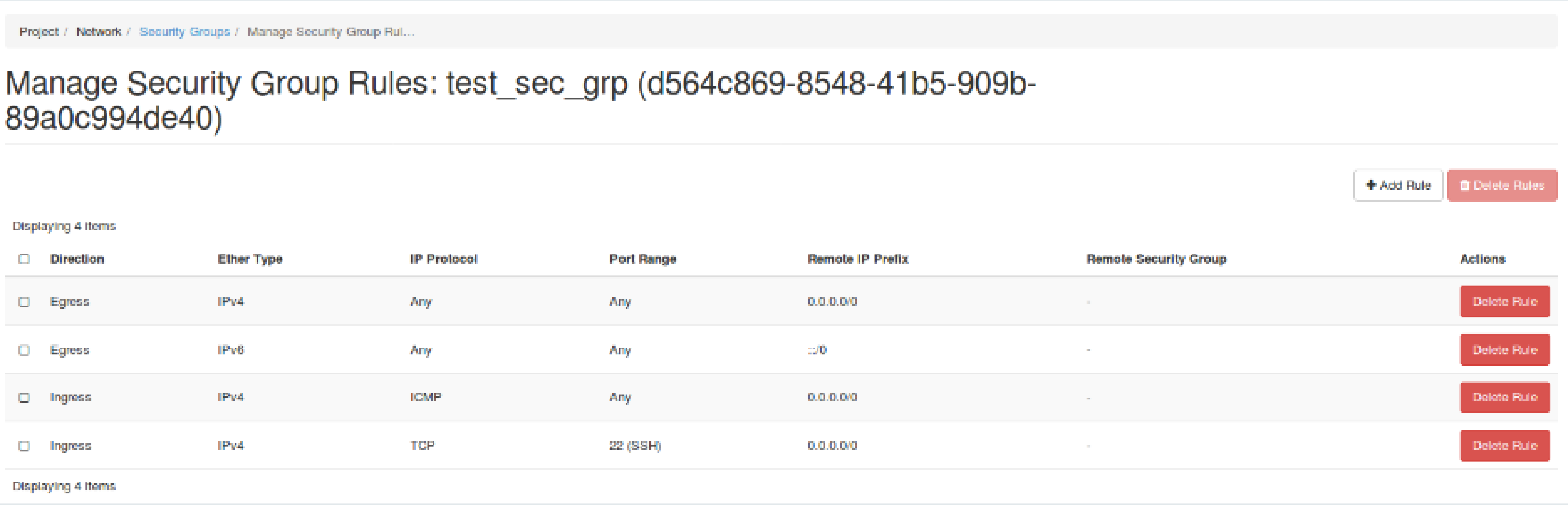
Description:

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.



Project / Network / Security Groups / Manage Security Group Rul...

Manage Security Group Rules: test_sec_grp (d564c869-8548-41b5-909b-89a0c994de40)

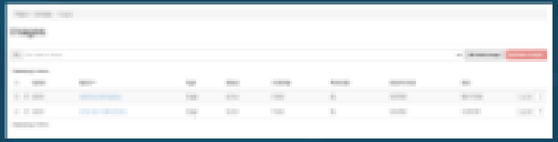
Displaying 4 items

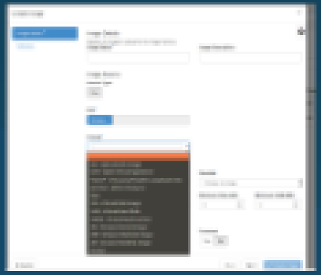
□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
□	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
□	Egress	IPv6	Any	Any	::/0	-	Delete Rule
□	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule

Displaying 4 items

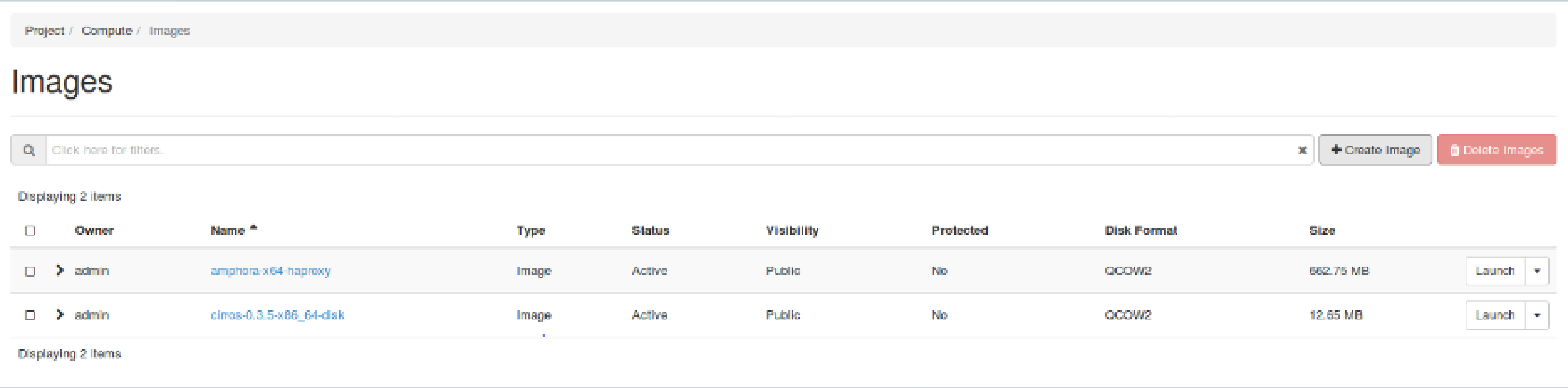
OpenStack Image Service (Glance)

- Provides image services to OpenStack. In this case, "images" refers to images (or virtual copies) of hard disks
- Glance allows these images to be used as templates when deploying new virtual machine instances
- By default we are given with two images. However, we can create more images according to our requirements. We can also import various types of image formats into OpenStack like Amazon machine Image, VDI, ISO, VHD etc.





ID etc.

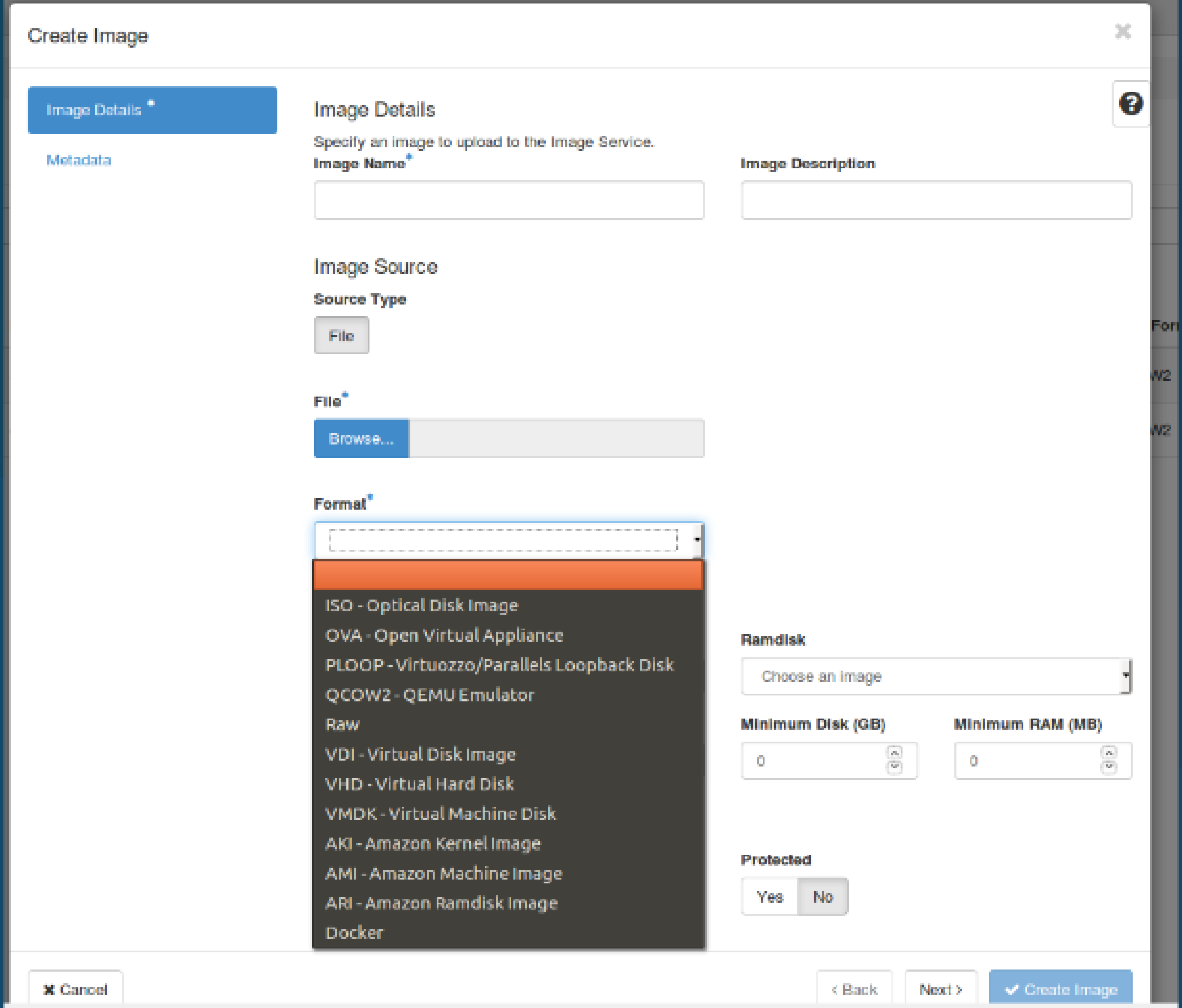


Images

Displaying 2 items

□	Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size	
□	admin	amphora-x64-haproxy	Image	Active	Public	No	QCOW2	662.75 MB	Launch
□	admin	cirros-0.3.5-x86_64-disk	Image	Active	Public	No	QCOW2	12.65 MB	Launch

Displaying 2 items



Create Image

Image Details

Specify an image to upload to the Image Service.

Image Name*

Image Description

Image Source

Source Type: File

File*

Format*

Ramdisk: Choose an image

Minimum Disk (GB): 0

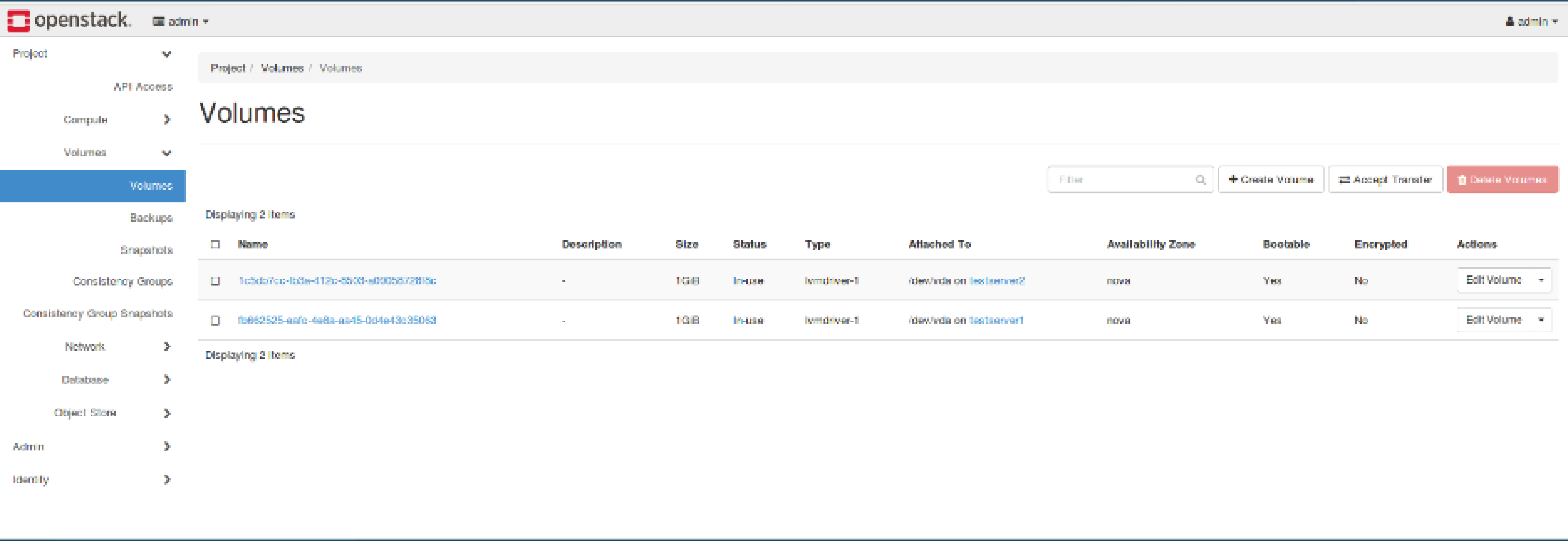
Minimum RAM (MB): 0

Protected: Yes [✓ No]

Image Details *
Metadata
Browse...
ISO - Optical Disk Image
OVA - Open Virtual Appliance
PLOOP - Virtuozzo/Parallels Loopback Disk
QCOW2 - QEMU Emulator
Raw
VDI - Virtual Disk Image
VHD - Virtual Hard Disk
VMDK - Virtual Machine Disk
AKI - Amazon Kernel Image
AMI - Amazon Machine Image
ARI - Amazon Ramdisk Image
Docker
✖ Cancel
‹ Back
Next ›
✔ Create Image

OpenStack Block Storage Service (Cinder)

- Cinder is a Block Storage service for OpenStack. It's designed to present storage resources to end users that can be consumed by the OpenStack Compute Project (Nova). This is done through the use of either a reference implementation (LVM) or plugin drivers for other storage.
- Virtualizes the management of block storage devices and provides end users with a self service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device.

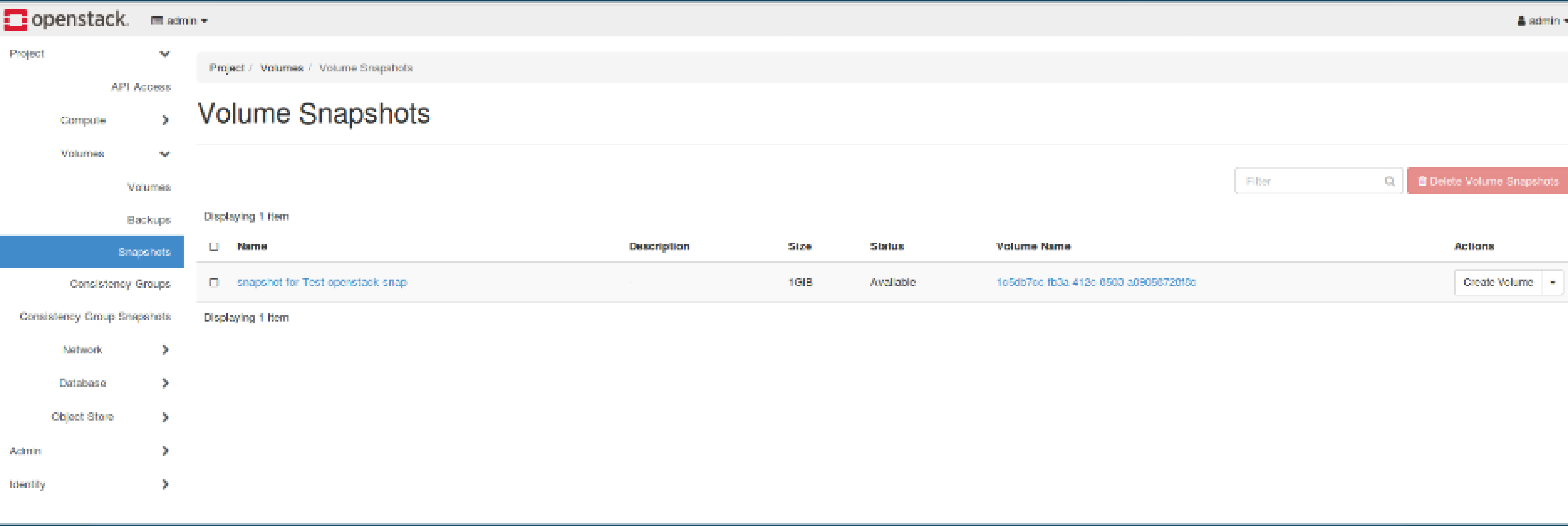


Volumes

Displaying 2 items

□	Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
□	1c5db7cc-fb0e-412c-8503-e080557285c	-	1GiB	In-use	lvmdriver-1	/dev/vda on testserver2	nova	Yes	No	Edit Volume ▾
□	fb682525-eafc-4e8a-aa45-0d4e43c35063	-	1GiB	In-use	lvmdriver-1	/dev/vda on testserver1	nova	Yes	No	Edit Volume ▾

Displaying 2 items



Volume Snapshots

Displaying 1 item

□	Name	Description	Size	Status	Volume Name	Actions
□	snapshot for Test openstack snap	-	1GiB	Available	1c5db7cc fb3a 412c 0503 a0905872018c	Create Volume

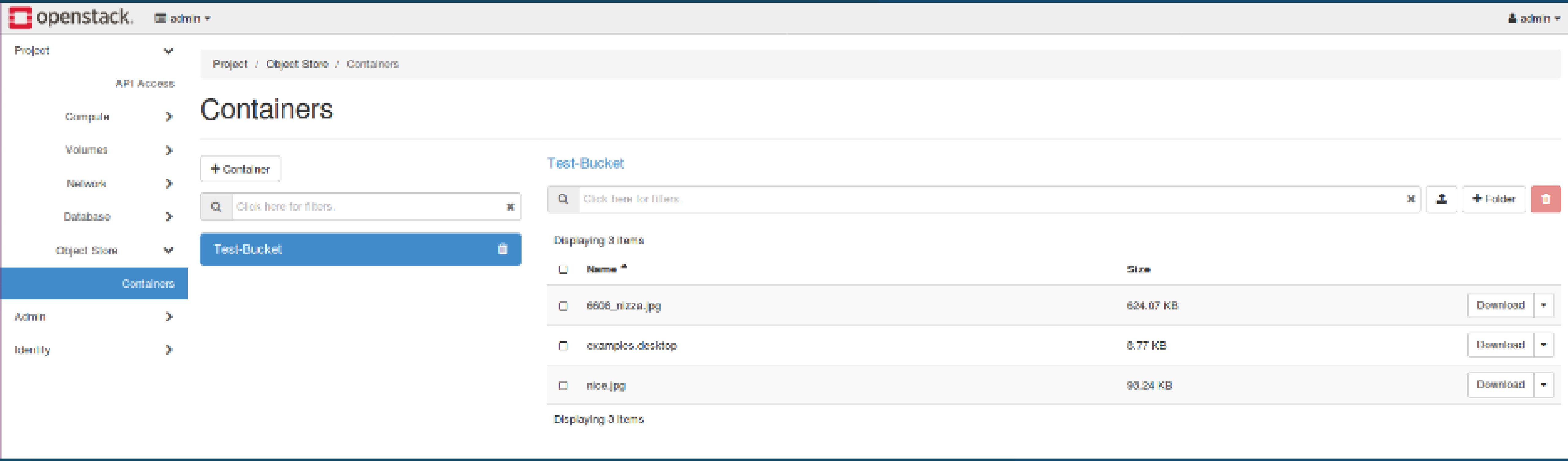
Displaying 1 item

OpenStack Object Storage Service (Swift)

- Swift, offers cloud storage software so that you can store and retrieve lots of data with a simple API. It's built for scale and optimized for durability, availability, and concurrency across the entire data set.
- Swift is ideal for storing unstructured data that can grow without bound. OpenStack Swift is best suited to backup and archive unstructured data such as documents, images, audio and video files, email and machine images.
- Containers

  A container is a storage compartment for your data and provides a way for you to organize your data.
- Objects

  Objects and files are written to multiple drives, and the Swift software ensures the data is replicated across a server cluster.



Project / Object Store / Containers

Containers

Test-Bucket

Test-Bucket

Displaying 3 items

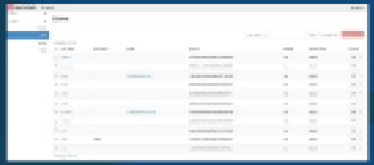
□	Name ▲	Size
□	6608_pizza.jpg	624.07 KB
□	examples.desktop	8.77 KB
□	nice.jpg	93.24 KB

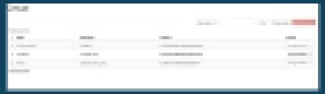
Displaying 3 items

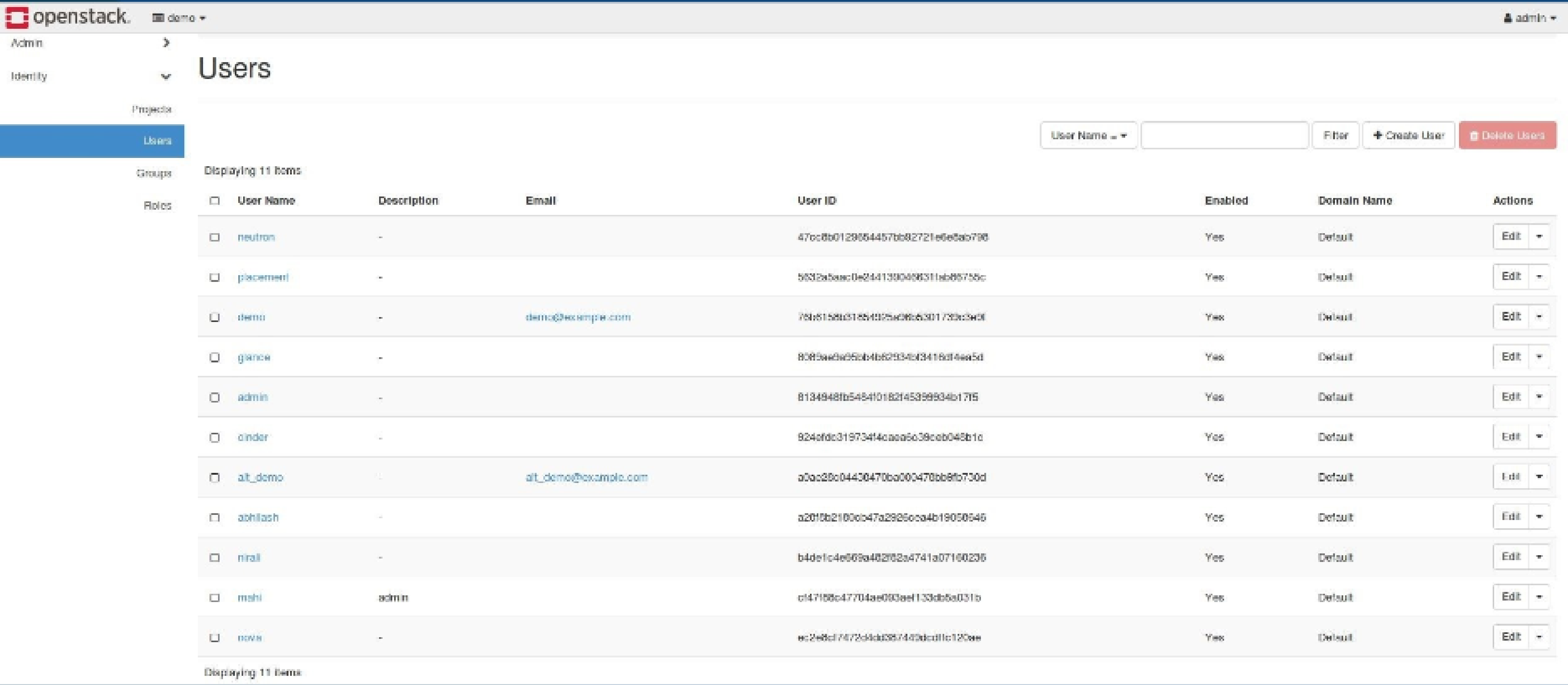
Identity Management (Keystone)

- Keystone is the identity service used by OpenStack for authentication
- It currently supports token-based and user-service authorization.

1. Authentication
2. Credentials
3. User
4. Group
5. Role





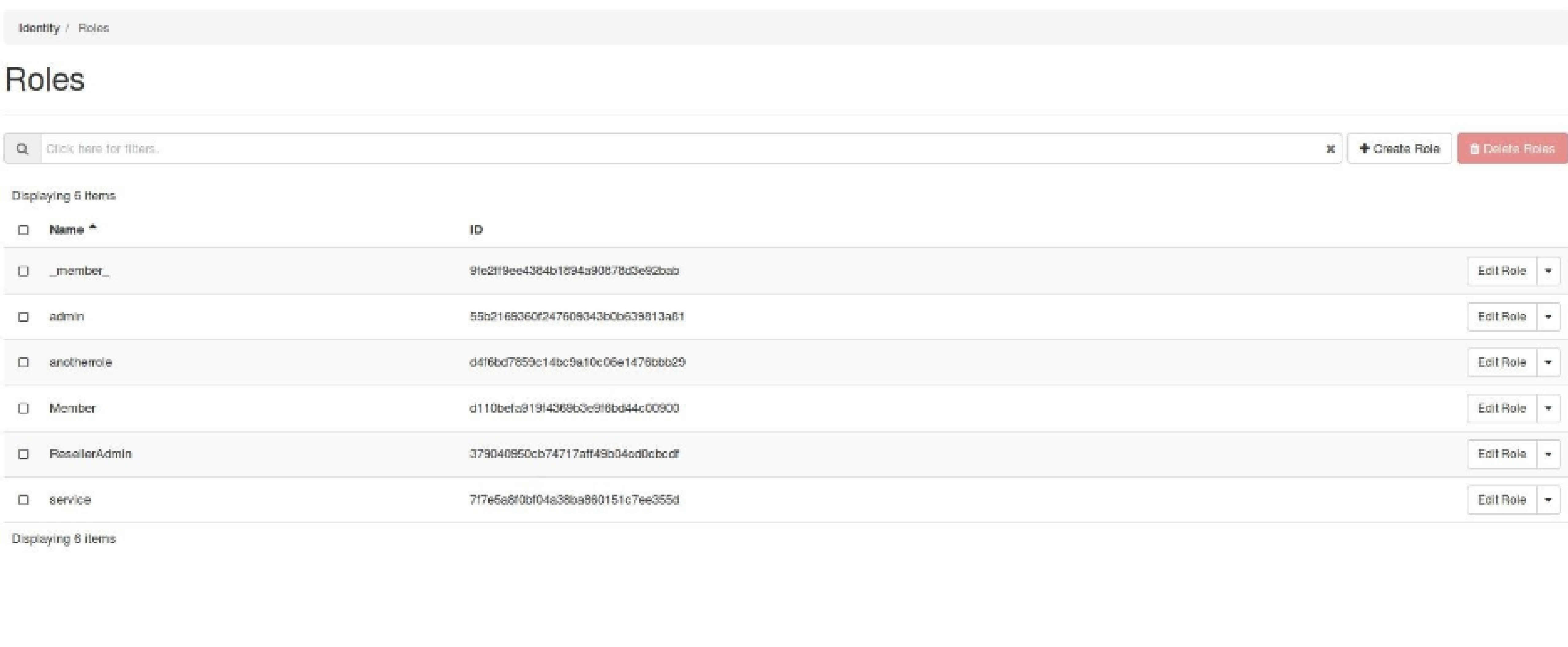


Users

Displaying 11 items

☐	User Name	Description	Email	User ID	Enabled	Domain Name	Actions
☐	neutron	-		47cc8b0129854457bb82721e6e8ab798	Yes	Default	Edit ▾
☐	placement	-		5632a5aac0e2441390498311ab86755c	Yes	Default	Edit ▾
☐	demo	-	demo@example.com	76b8158d31854925a06b6301733c3e0f	Yes	Default	Edit ▾
☐	glance	-		8089ae0e95bb4b82934bf3418df4ea5d	Yes	Default	Edit ▾
☐	admin	-		8134948fb545410182145399834b17f5	Yes	Default	Edit ▾
☐	cinder	-		924cfdc319734f4caea6c39ceb048b1c	Yes	Default	Edit ▾
☐	alt_demo	-	alt_demo@example.com	a0ac28c04430470ba000470bb8fb730d	Yes	Default	Edit ▾
☐	abhilash	-		a2015b2180cb47a2926eca4b19050646	Yes	Default	Edit ▾
☐	nirali	-		b4de1c4e669a482f82a4741a07160236	Yes	Default	Edit ▾
☐	mahi	admin		cf47f88c47704ae093ae1133db5a031b	Yes	Default	Edit ▾
☐	nova	-		ec2e8cf7472d4dd987443dcdffc120ae	Yes	Default	Edit ▾

Displaying 11 items



Roles

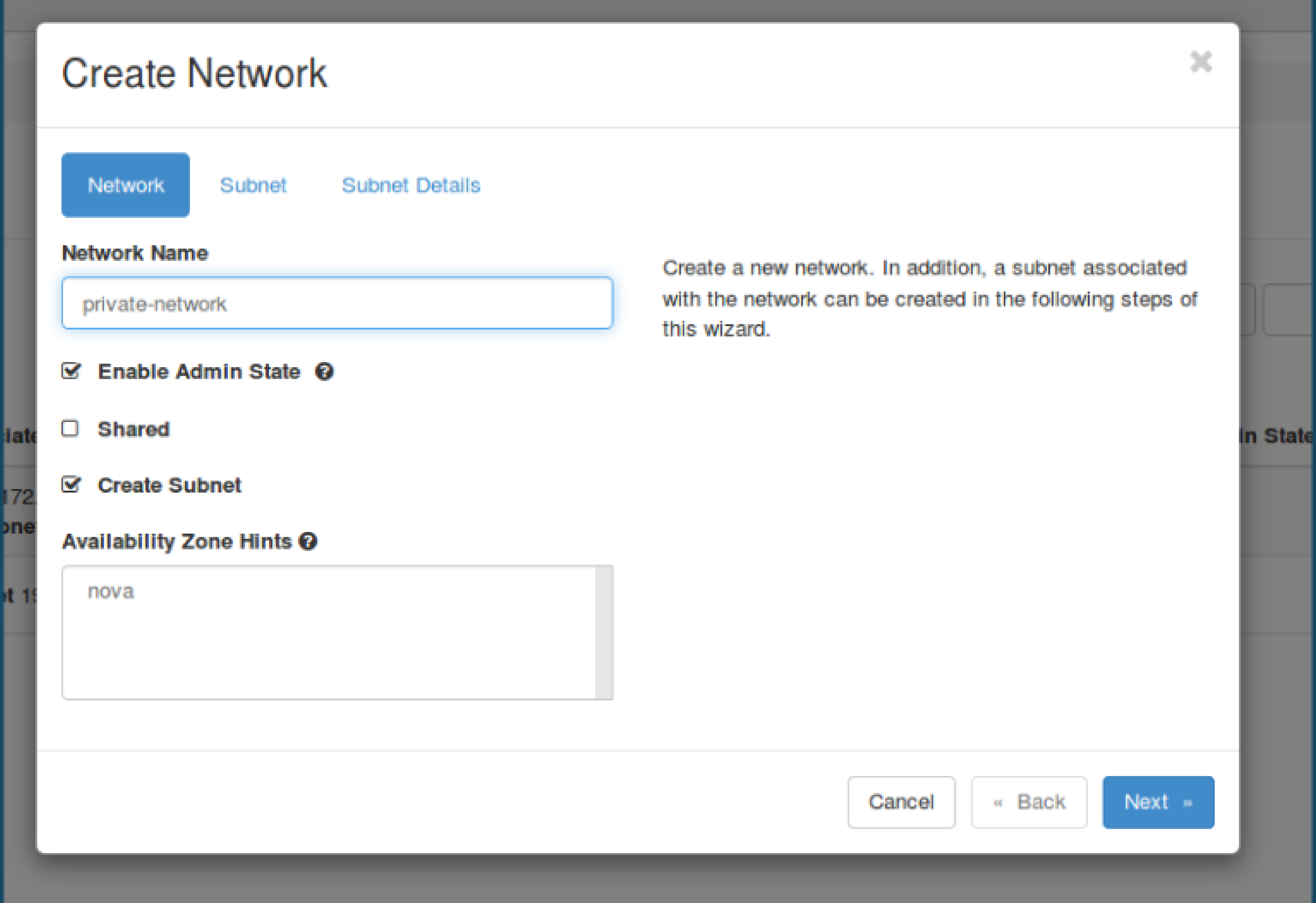
Displaying 6 items

□	Name ▲	ID	
□	_member_	9fe2ff9ee4384b1894a90878d3e92bab	Edit Role ▾
□	admin	55b2169360f247609343b0b639813a81	Edit Role ▾
□	anotherrole	d4f6bd7859c14bc9a10c06e1476bbb29	Edit Role ▾
□	Member	d110befa919f4369b3e9f6bd44c00900	Edit Role ▾
□	ResellerAdmin	379040960cb74717aff49b04ad0cbcdf	Edit Role ▾
□	service	7f7e5a8f0bf04a38ba860151c7ee355d	Edit Role ▾

Displaying 6 items

OpenStack Networking (Neutron)

- OpenStack Networking (Neutron) is a system for managing networks and IP addresses.
- OpenStack Networking ensures the network is not a bottleneck and gives users self-service ability, even over network configurations.
- Users can create their own networks, control traffic, and connect servers and devices to one or more networks.
- Administrators can use software defined networking (SDN) technologies like open flow to support high levels of multi tenancy and massive scale



Create Network

Network Name

private-network

☑ Enable Admin State

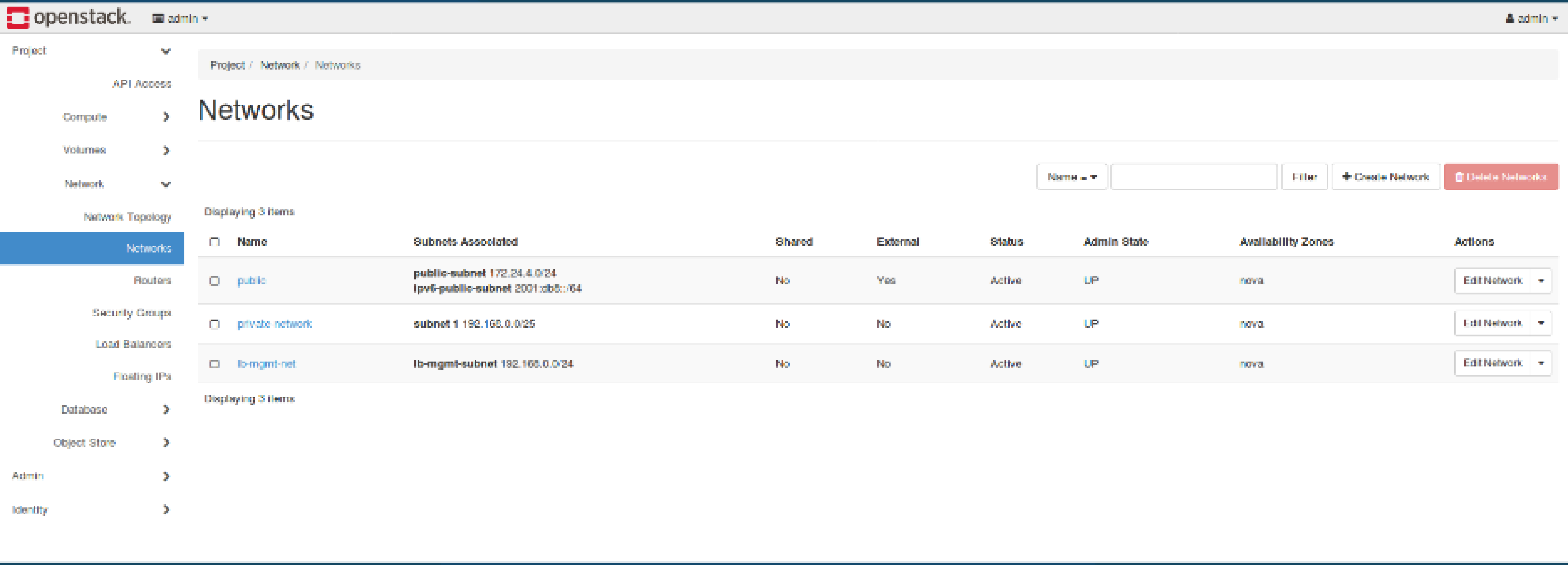
□ Shared

☑ Create Subnet

Availability Zone Hints

nova

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

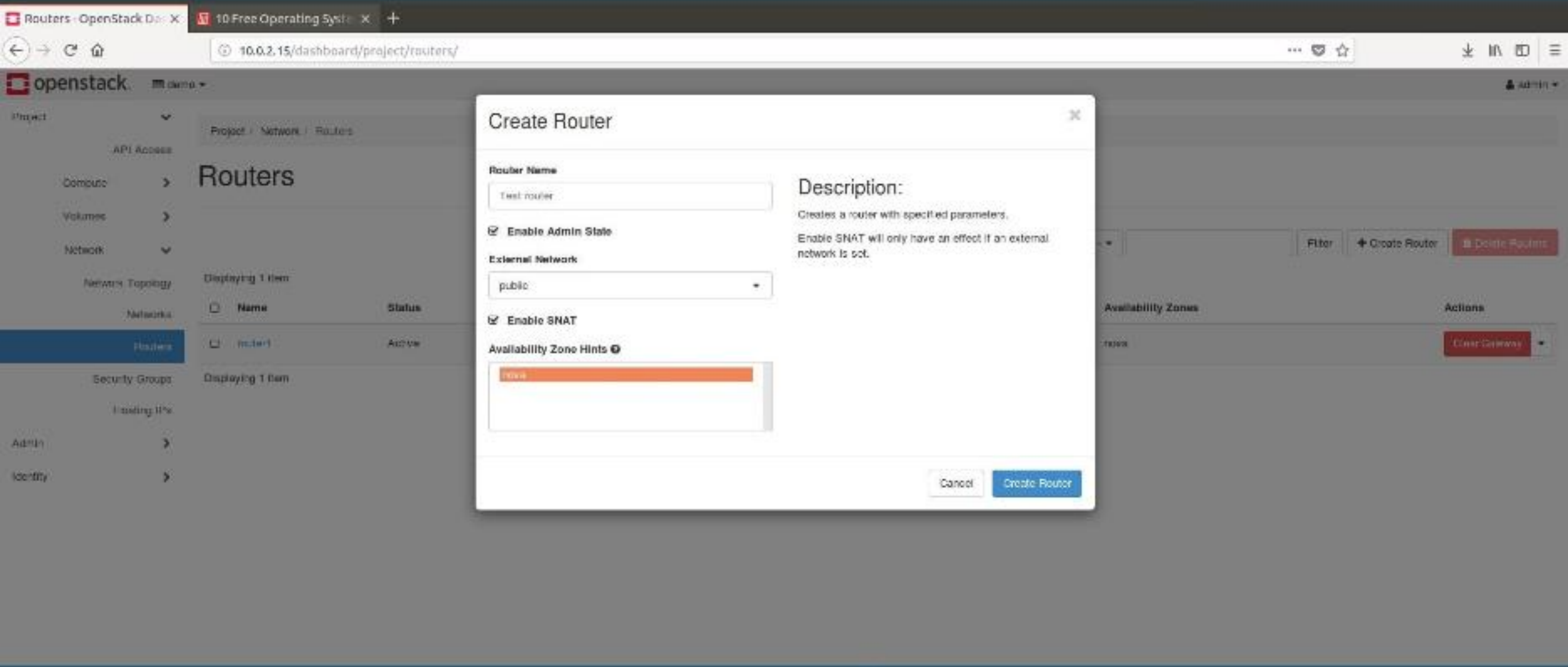


Networks

Displaying 3 items

□	Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
□	public	**public-subnet** 172.24.4.0/24 **ipv6-public-subnet** 2001:db8::/64	No	Yes	Active	UP	nova	Edit Network ▾
□	private network	**subnet 1** 192.168.0.0/25	No	No	Active	UP	nova	Edit Network ▾
□	lb-mgmt-net	**lb-mgmt-subnet** 192.168.0.0/24	No	No	Active	UP	nova	Edit Network ▾

Displaying 3 items



Routers

Displaying 1 item

□	Name	Status	Availability Zones	Actions
□	router1	Active	nova	Clear Gateway

Displaying 1 item

Create Router

Router Name

Test router

☑ **Enable Admin State**

External Network

public

☑ **Enable SNAT**

Availability Zone Hints

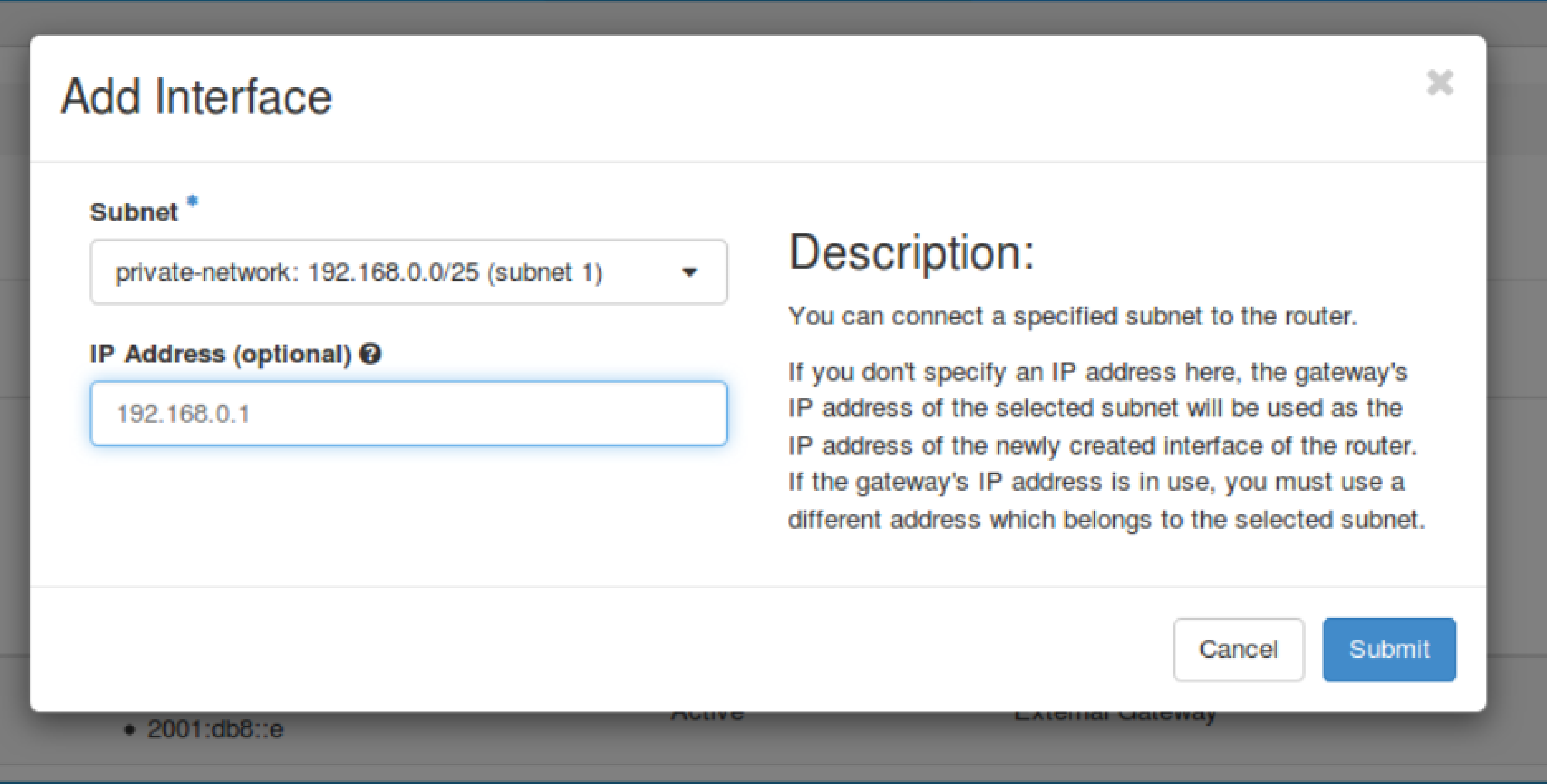
nova

Description:

Creates a router with specified parameters.

Enable SNAT will only have an effect if an external network is set.

Cancel   Create Router



Add Interface

Subnet *

private-network: 192.168.0.0/25 (subnet 1)

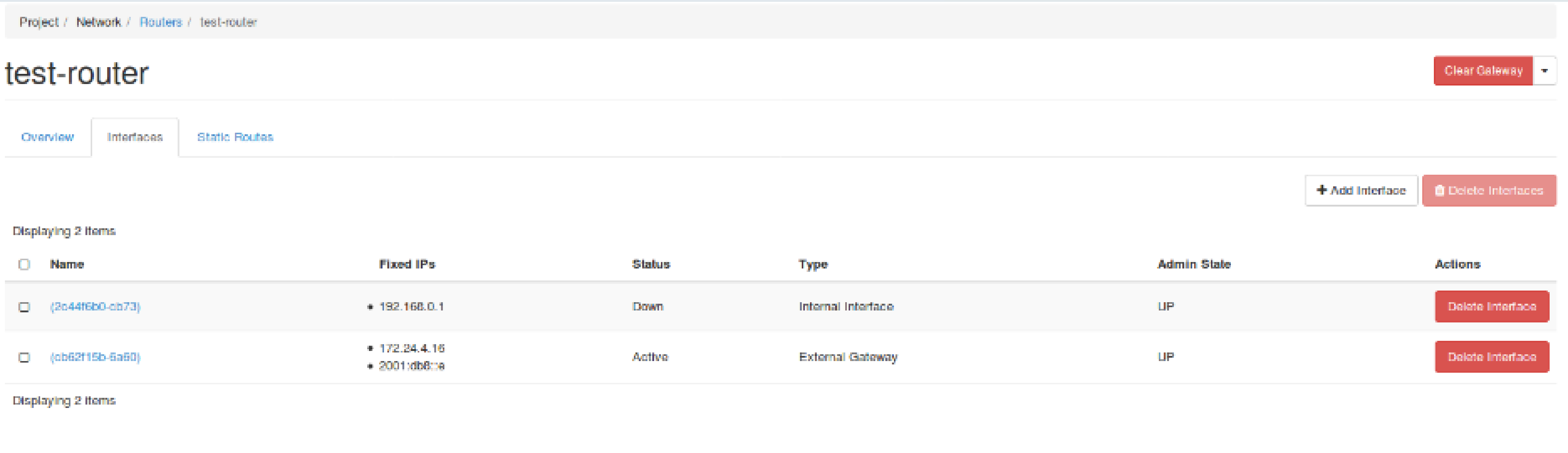
IP Address (optional)

192.168.0.1

Description:

You can connect a specified subnet to the router.

If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

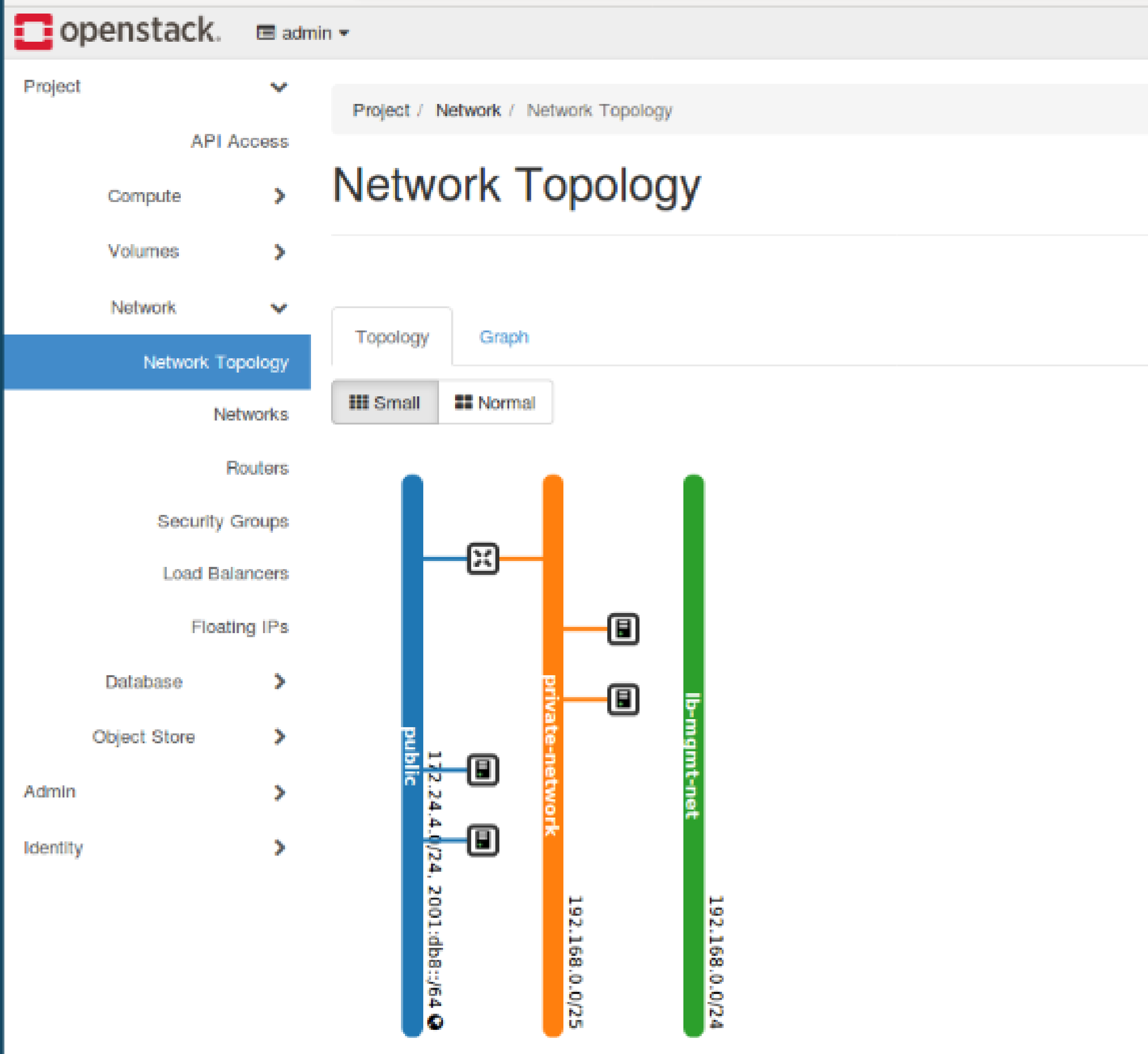


test-router

Displaying 2 items

□	Name	Fixed IPs	Status	Type	Admin State	Actions
□	(2c44f6b0-cb73)	• 192.168.0.1	Down	Internal Interface	UP	Delete Interface
□	(cb62f15b-6a60)	• 172.24.4.16 • 2001:db8::e	Active	External Gateway	UP	Delete Interface

Displaying 2 items

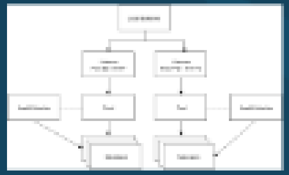


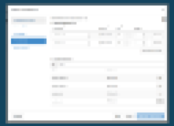
openstack.
admin
Project
API Access
Compute
Volumes
Network
Network Topology
Networks
Routers
Security Groups
Load Balancers
Floating IPs
Database
Object Store
Admin
Identity
Project / Network / Network Topology
Network Topology
Topology
Graph
Small
Normal
public
172.24.4.0/24, 2001:db8::/64
private-network
192.168.0.0/25
lb-mgmt-net
192.168.0.0/24

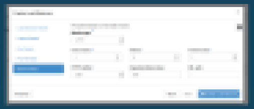
Load Balancer as a Service (Octavia)

- Octavia is an open source, operator-scale load balancing solution designed to work with OpenStack.
- Octavia accomplishes its delivery of load balancing services by managing a fleet of virtual machines known as amphorae.
- This on-demand, horizontal scaling feature differentiates Octavia from other load balancing solutions.

- Load balancer
  The load balancer occupies a neutron network port and has an IP address assigned from a subnet.
- Listener
  Load balancers can listen for requests on multiple ports. Each one of those ports is specified by a listener.
- Pool
  A pool holds a list of members that serve content through the load balancer.
- Member
  Members are servers that serve traffic behind a load balancer. Each member is specified by the IP address and port that it uses to serve traffic.
- Health monitor
  Members may go offline from time to time and health monitors divert traffic away from members that are not responding properly. Health monitors are associated with pools.







- Load balancer

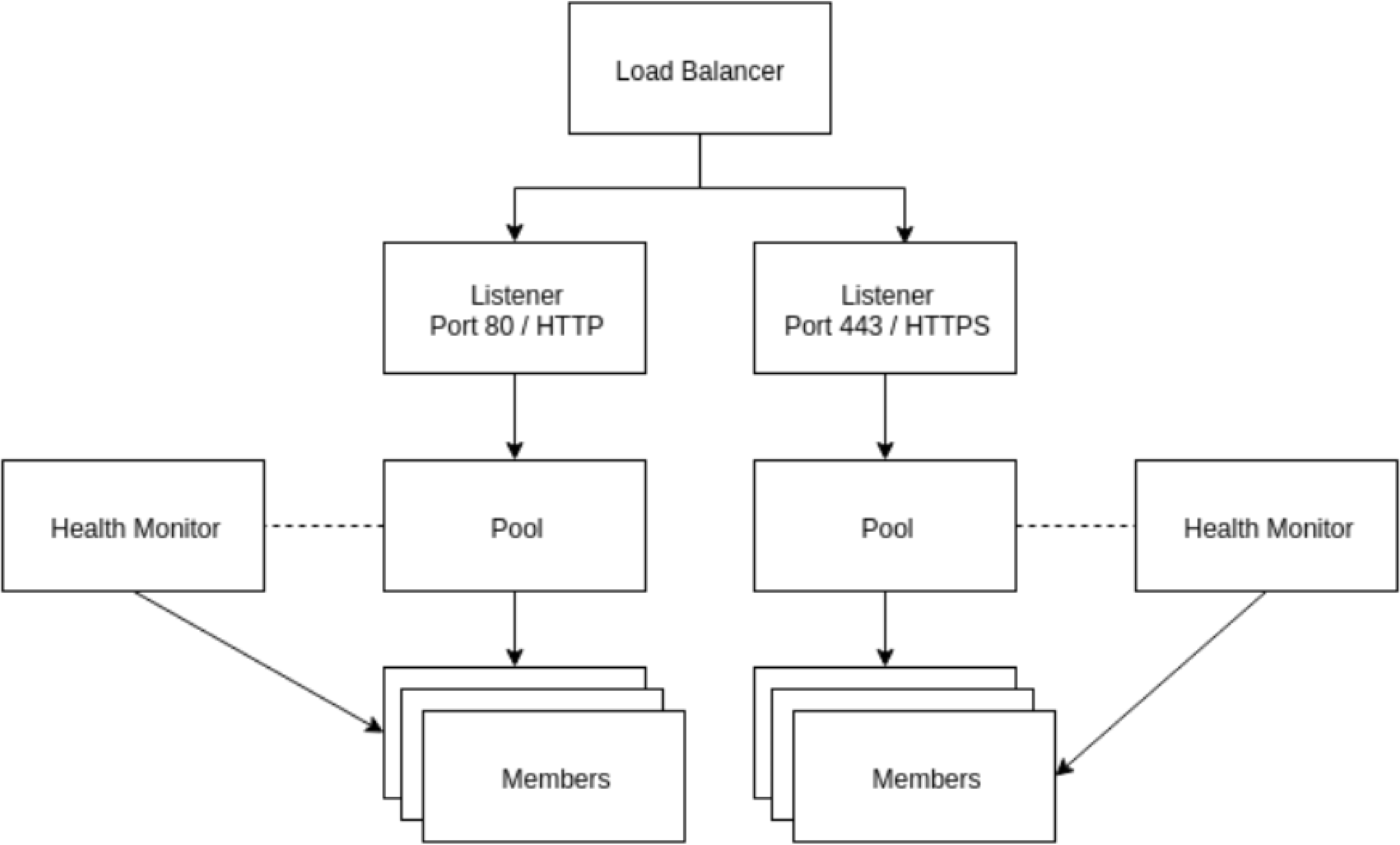
  The load balancer occupies a neutron network port and has an IP address assigned from a subnet.
- Listener

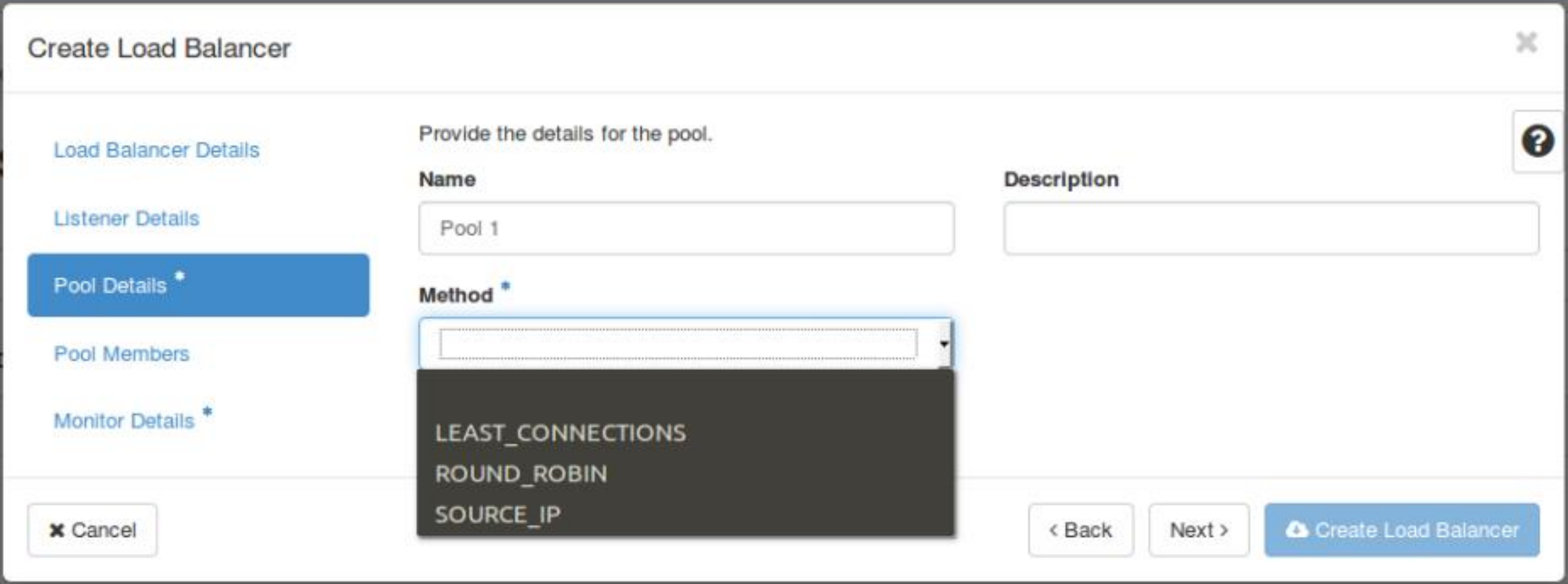
  Load balancers can listen for requests on multiple ports. Each one of those ports is specified by a listener.
- Pool

  A pool holds a list of members that serve content through the load balancer.
- Member

  Members are servers that serve traffic behind a load balancer. Each member is specified b the IP address and port that it uses to serve traffic.
- Health monitor

  Members may go offline from time to time and health monitors divert traffic away from members that are not responding properly. Health monitors are associated with pools.


Load Balancer
Listener
Port 80 / HTTP
Listener
Port 443 / HTTPS
Health Monitor
Pool
Pool
Health Monitor
Members
Members



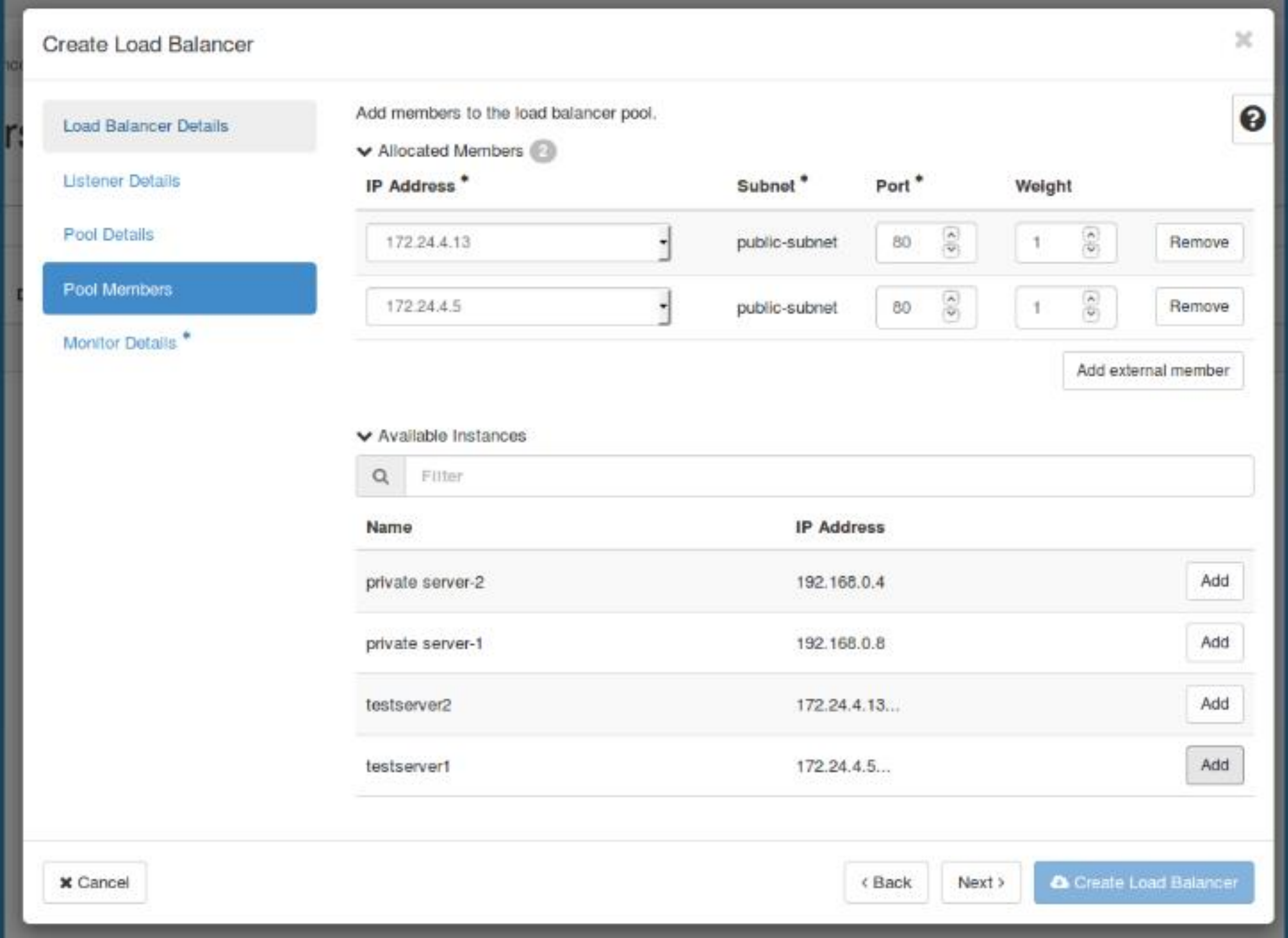
Create Load Balancer

Provide the details for the pool.

Name: Pool 1

Description:

Method *:



Create Load Balancer

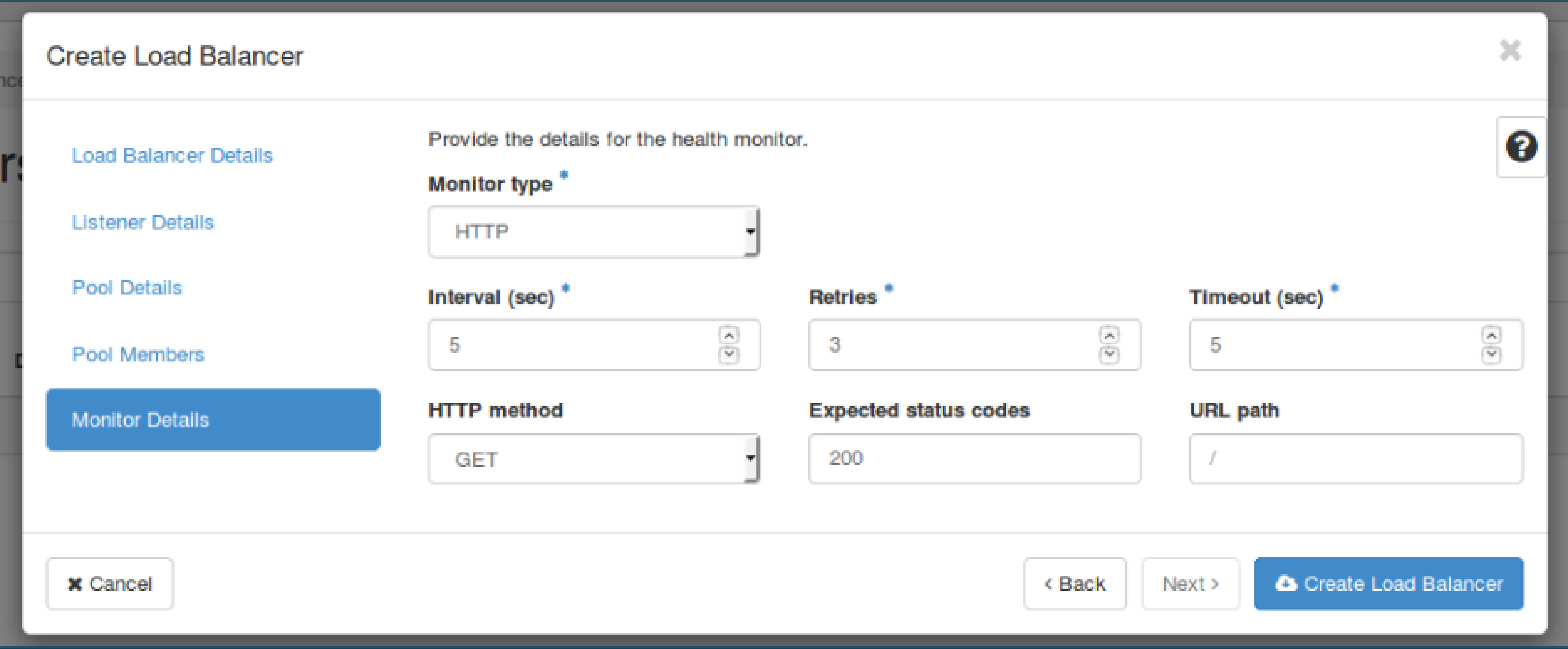
Add members to the load balancer pool.

Allocated Members 2

IP Address *	Subnet *	Port *	Weight	
172.24.4.13	public-subnet	80	1	Remove
172.24.4.5	public-subnet	80	1	Remove

Available Instances

Name	IP Address	
private server-2	192.168.0.4	Add
private server-1	192.168.0.8	Add
testserver2	172.24.4.13...	Add
testserver1	172.24.4.5...	Add



Create Load Balancer

Provide the details for the health monitor.

Monitor type *: HTTP

Interval (sec) *: 5

Retries *: 3

Timeout (sec) *: 5

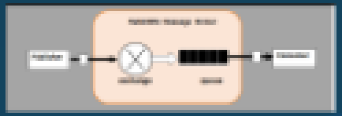
HTTP method: GET

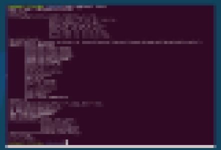
Expected status codes: 200

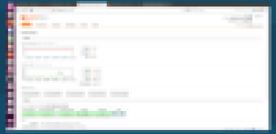
URL path: /

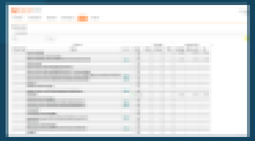
Messaging Queue (RabbitMQ)

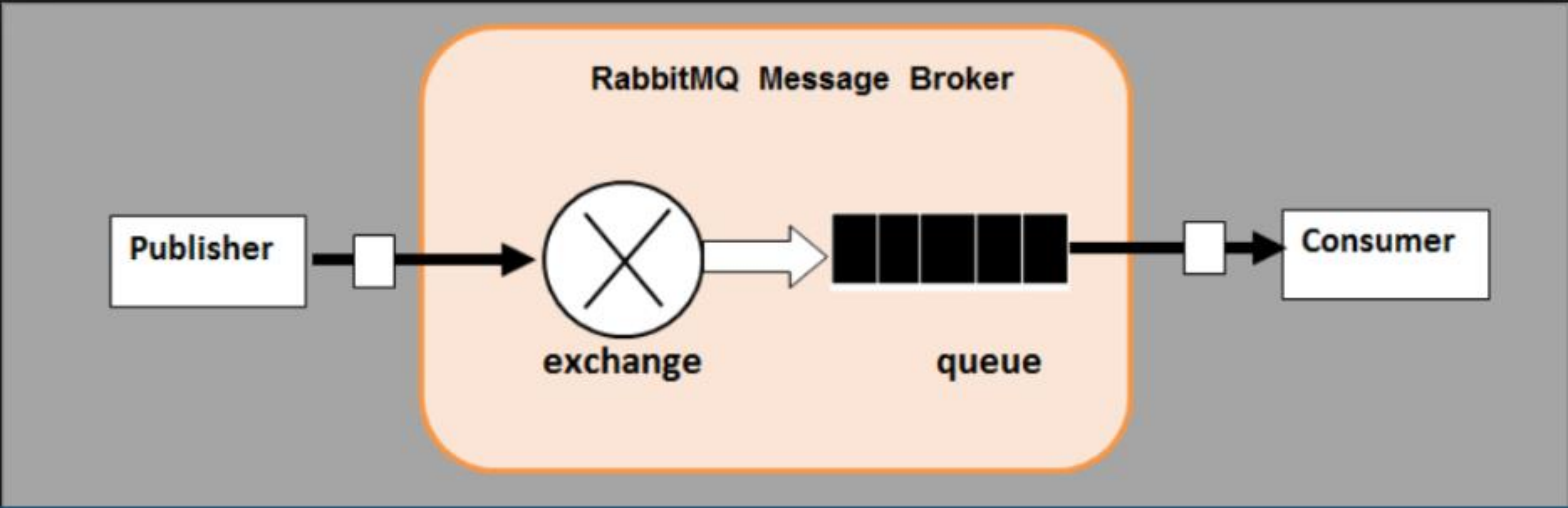
- Message queuing services facilitate inter-process communication in OpenStack.
- OpenStack supports RabbitMQ, Qpid, and ZeroHQ.
- The message queue runs on the controller node

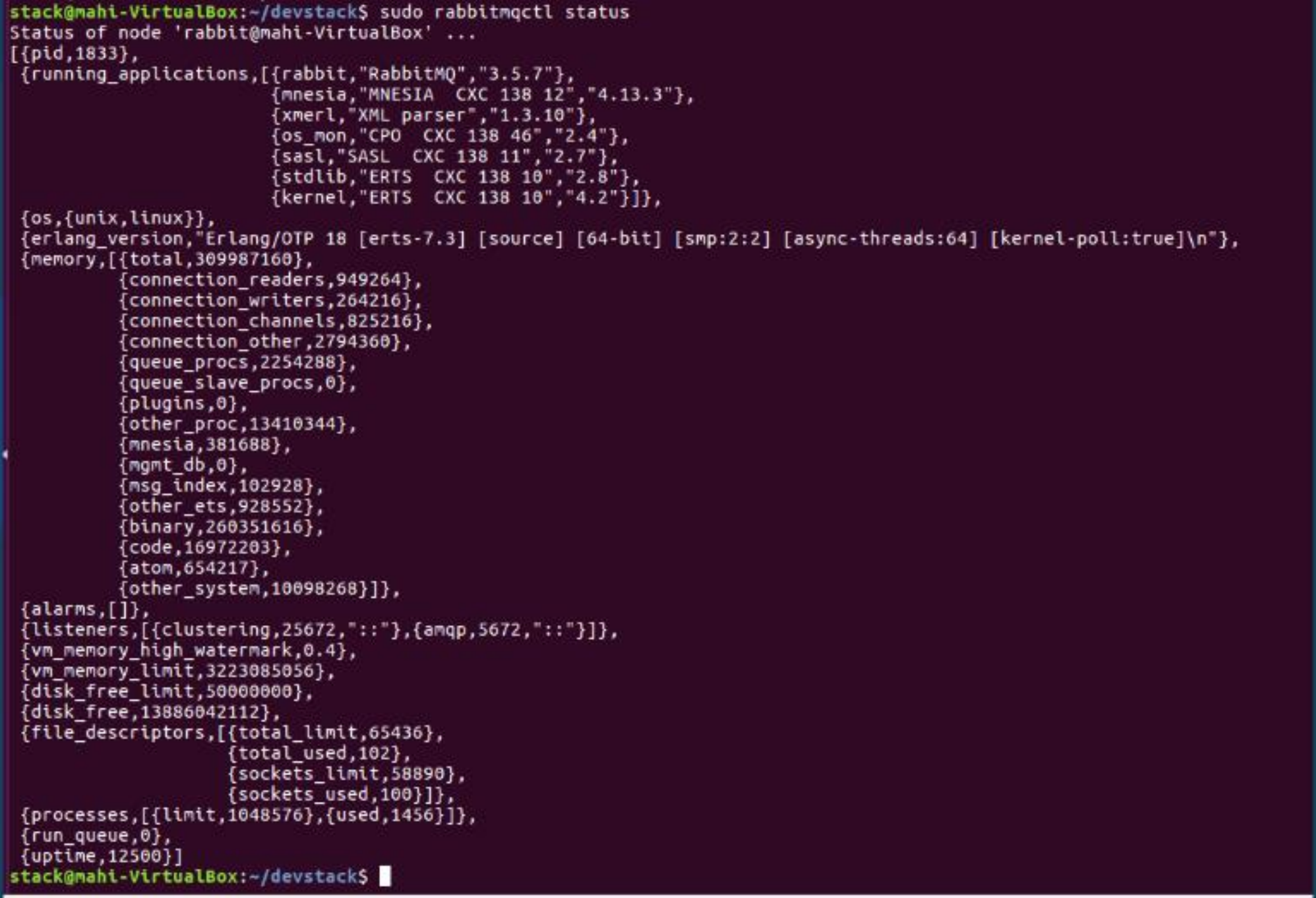




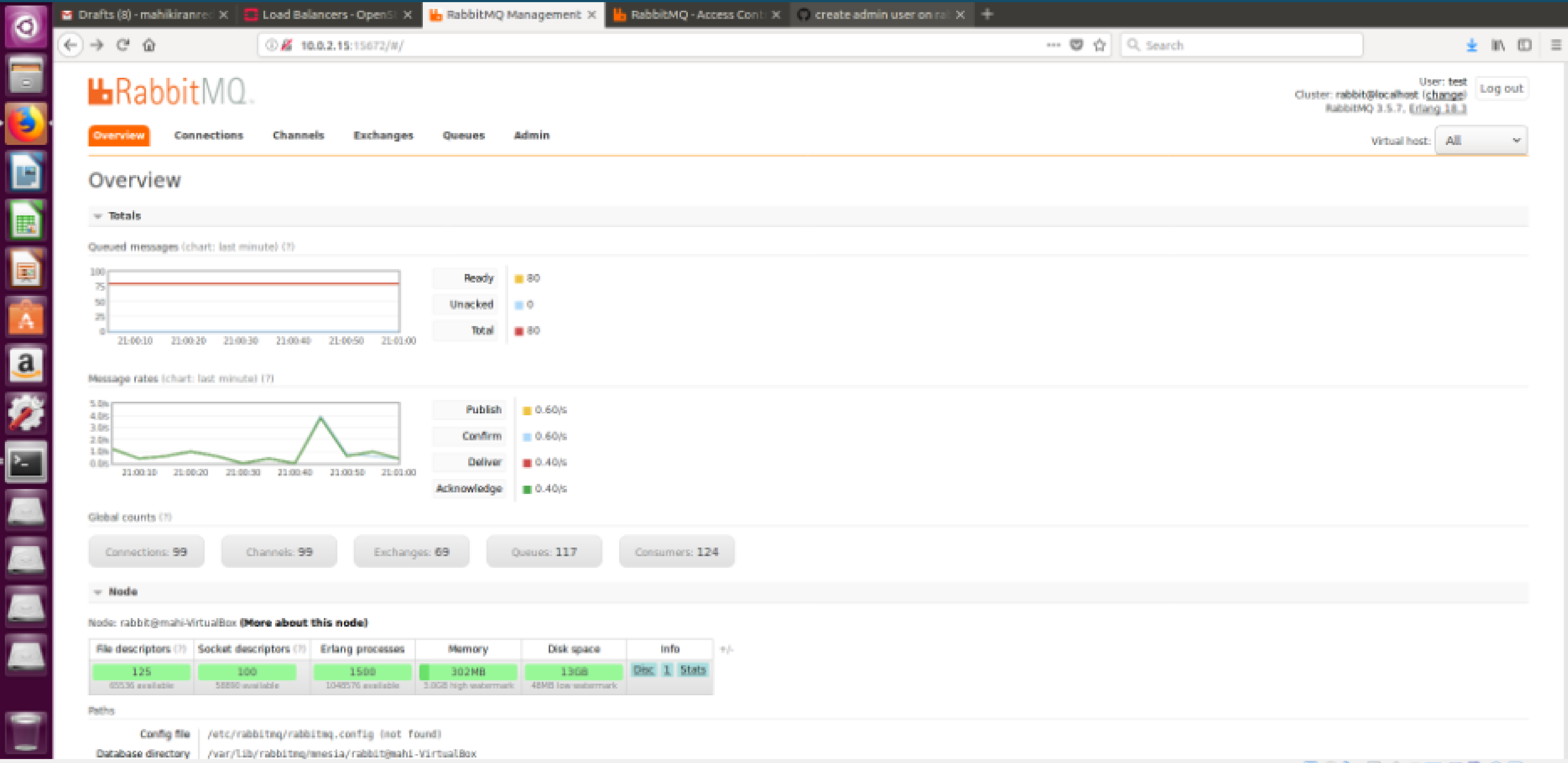





RabbitMQ Message Broker
Publisher
exchange
queue
Consumer



```
stack@mahi-VirtualBox:~/devstack$ sudo rabbitmqctl status
Status of node 'rabbit@mahi-VirtualBox' ...
[{pid,1833},
 {running_applications,[{rabbit,"RabbitMQ","3.5.7"},
                        {mnesia,"MNESIA  CXC 138 12","4.13.3"},
                        {xmerl,"XML parser","1.3.10"},
                        {os_mon,"CPO  CXC 138 46","2.4"},
                        {sasl,"SASL  CXC 138 11","2.7"},
                        {stdlib,"ERTS  CXC 138 10","2.8"},
                        {kernel,"ERTS  CXC 138 10","4.2"}]},
 {os,{unix,linux}},
 {erlang_version,"Erlang/OTP 18 [erts-7.3] [source] [64-bit] [smp:2:2] [async-threads:64] [kernel-poll:true]\n"},
 {memory,[{total,309987160},
          {connection_readers,949264},
          {connection_writers,264216},
          {connection_channels,825216},
          {connection_other,2794360},
          {queue_procs,2254288},
          {queue_slave_procs,0},
          {plugins,0},
          {other_proc,13410344},
          {mnesia,381688},
          {mgmt_db,0},
          {msg_index,102928},
          {other_ets,928552},
          {binary,260351616},
          {code,16972203},
          {atom,654217},
          {other_system,10098268}]},
 {alarms,[]},
 {listeners,[{clustering,25672,"::"},{amqp,5672,"::"}]},
 {vm_memory_high_watermark,0.4},
 {vm_memory_limit,3223085056},
 {disk_free_limit,50000000},
 {disk_free,13886042112},
 {file_descriptors,[{total_limit,65436},
                    {total_used,102},
                    {sockets_limit,58890},
                    {sockets_used,100}]},
 {processes,[{limit,1048576},{used,1456}]},
 {run_queue,0},
 {uptime,12500}]
stack@mahi-VirtualBox:~/devstack$ 
```



User: test
Cluster: rabbit@localhost (change)
RabbitMQ 3.5.7, Erlang 18.3

Virtual host: All

Overview

Totals

Queued messages (chart: last minute) (?)

Ready	80
Unacked	0
Total	80

Message rates (chart: last minute) (?)

Publish	0.60/s
Confirm	0.60/s
Deliver	0.40/s
Acknowledge	0.40/s

Global counts (?)

Connections: 99 Channels: 99 Exchanges: 69 Queues: 117 Consumers: 124

Node

Node: rabbit@mahi-VirtualBox **(More about this node)**

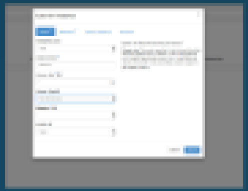
File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Info
125 65536 available	100 58890 available	1500 1048576 available	302MB 3.0GB high watermark	13GB 48MB low watermark	Disc 1 Stats

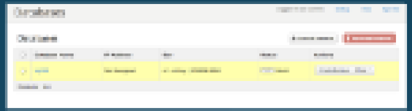
Paths

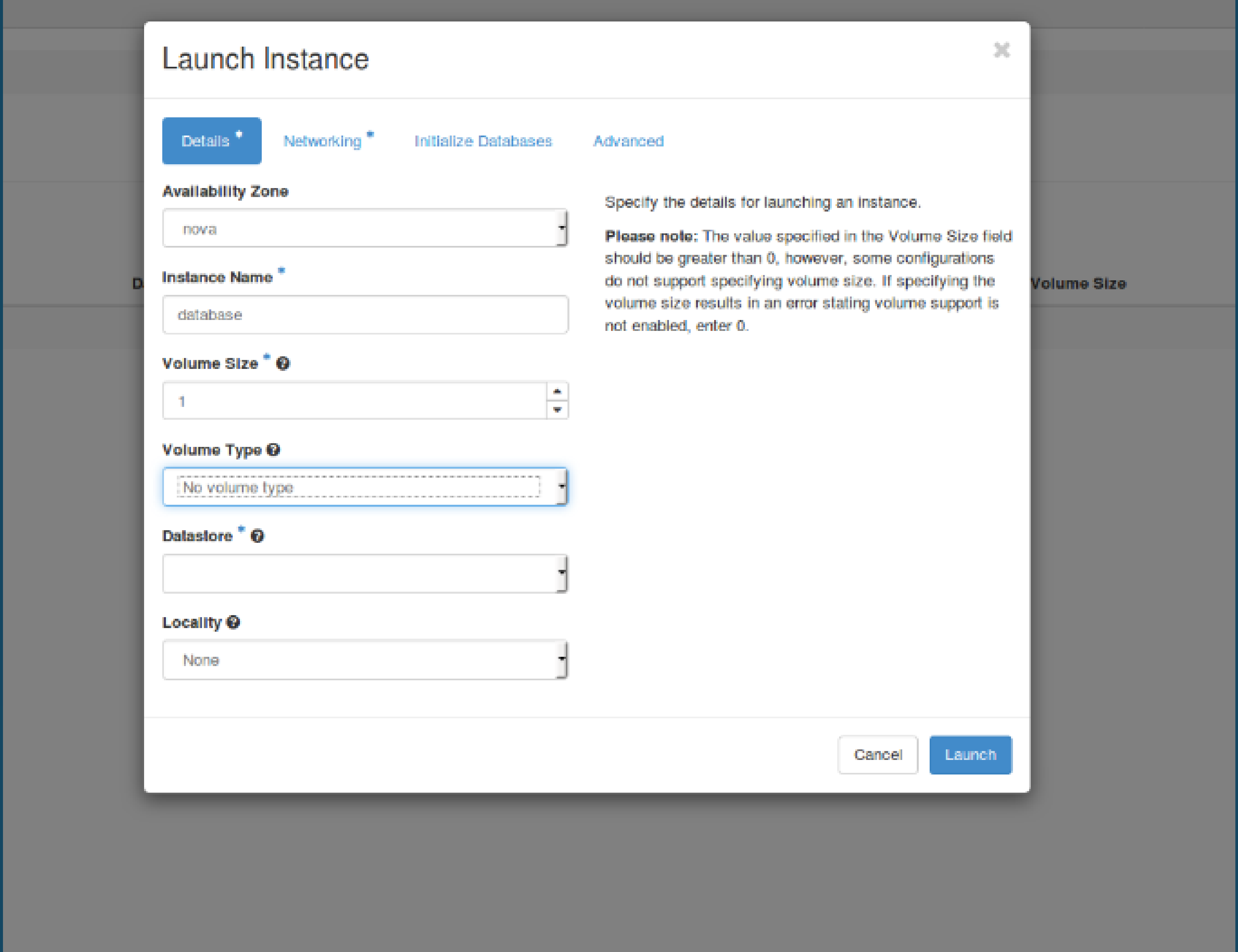
Config file	/etc/rabbitmq/rabbitmq.config (not found)
Database directory	/var/lib/rabbitmq/mnesia/rabbit@mahi-VirtualBox

Database as a Service (Trove)

- Trove is Database as a Service for OpenStack
- Designed to run entirely on OpenStack with the goal of allowing users to quickly and easily utilize the features of a relational or non-relational database without the burden of handling complex administrative tasks.
- Trove supports Cassandra, CouchBase, CouchDB, DataStax Enterprise, DB2, MariaDB, MongoDB, MySQL, Oracle, Percona Server, PostgreSQL, Redis and Vertica
- We can launch clusters of databases using trove dashboard







Launch Instance

Availability Zone: nova

Instance Name *: database

Volume Size * : 1

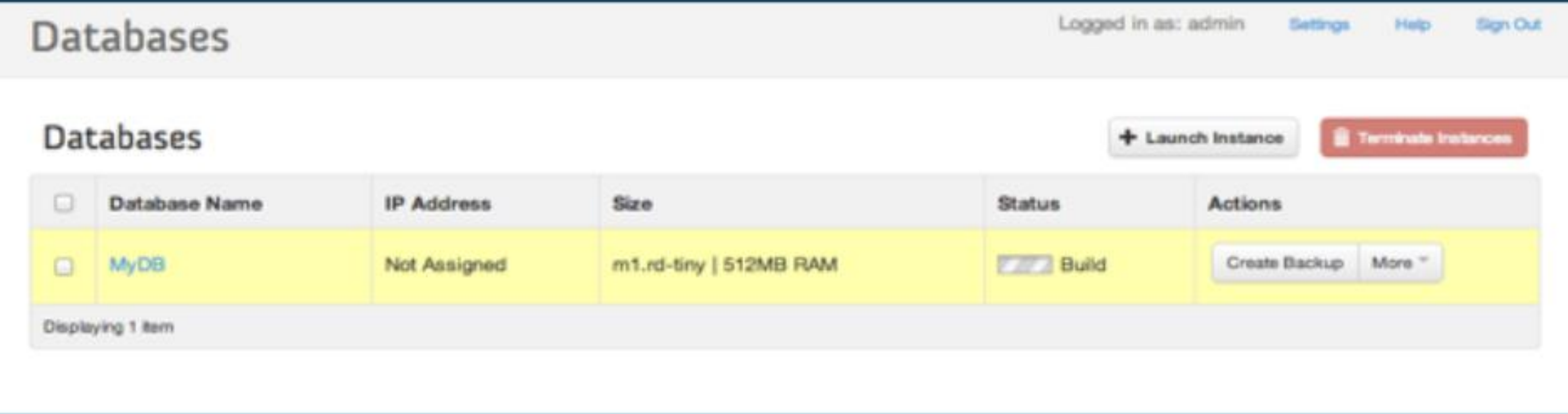
Volume Type: No volume type

Datastore *:

Locality: None

Specify the details for launching an instance.

Please note: The value specified in the Volume Size field should be greater than 0, however, some configurations do not support specifying volume size. If specifying the volume size results in an error stating volume support is not enabled, enter 0.



Databases

Databases

□	Database Name	IP Address	Size	Status	Actions
□	MyDB	Not Assigned	m1.rd-tiny | 512MB RAM	Build	Create Backup More

Displaying 1 item

Start
DEMO