STC Cloud Services

Bluvalt Brand

Bluvalt JIDDAH VDC Administrator Manual

Version: 2.2

Date: 12/7/2017

Table of Contents

Confidentiality Agreement

The information contained in this document is the property of STC and may not be copied or communicated to a third party or used for any purpose other than that for which it is supplied, without the express written consent of STC. ***bluvalt*** *is a registered trademark of STC.*

Acronyms

LIST OF ACRONYMS USED IN THIS DOCUMENT	
VDC	Virtual Data Center
IaaS	Infrastructure as a Service
VMs	Virtual Machines
VNC	Virtual Network Computing
IPAM	IP Address Management
VIFs	Virtual Network Interfaces
RFB Protocol	Remote Framebuffer

BLUVALT VDC ADMINISTRATOR MANUAL

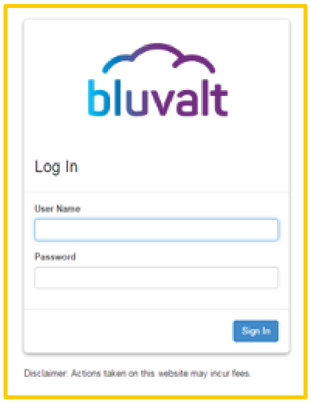
Preface

bluvalt Infrastructure as a Service (IaaS) platform is based on OpenStack, which is a free open-source software platform for cloud computing. The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout the data center. Users can manage their resources through a web-based dashboard.

This guide enables administrators to utilize ***bluvalt*** to its full potential, as per their rights & roles in managing the platform on the cloud.

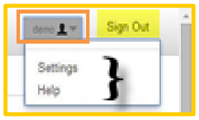
1. Dashboard Overview

1.1. Getting Started

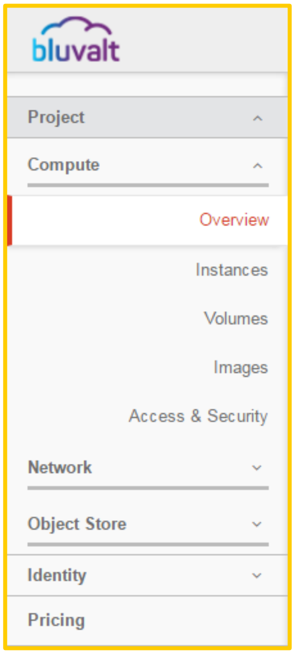


Disclaimer: Actions taken on this website may incur fees.

- Log in using your Username and Password. Then, click Sign In.
- Upon a successful login, the interface header will show your username.
- You can apply any adjustments on your account using Settings option, or check the interface Help through the provided drop-down list.
- The Sign Out option is also available on the dashboard.



1.2. Interface Components



bluvalt platform provides the following navigation pane to manage all menu items effectively, as per business needs. Users can click **Project** pane tab to manage **Compute**, **Network** & **Object Store**, along with their sub menu items. The navigation pane also provides major pane tabs for both **Identity** & **Pricing**.

1.2.1. Project - Compute Section

- **Overview:** Check the project reports.
  - **Instances:** View, launch or reboot instances, and perform various actions on instances
  - **Volumes**: View, create, edit, and delete volumes and volume snapshots.
  - **Images:** View images created by project users, along with any image that is publicly available. This menu item will enable users to create, edit, and delete images. Hence, they can launch new instances from these images.
- **Access & Security:** It lists the following sub-items:
  - **Security Groups:** View, create, edit, and delete *Security Groups* and *Security Group Rules*.
  - **Key Pairs:** View, create, edit, and import *SSH key pairs*, or delete key pairs.
  - **Floating IPs:** Allocate a floating IP address to/release it from a project.

1.2.2. Project - Network Section

- **Network Topology:** View the current network topology.
    - **Networks:** Create and manage public and private networks.
    - **Routers:** Create and manage subnets.
    - **Load Balancers:** Create and manage *Load Balancer Pools*, *Members* and *Health Monitors*.

1.2.3. Project - Object Store Section

- **Containers:** Create and manage object storage.

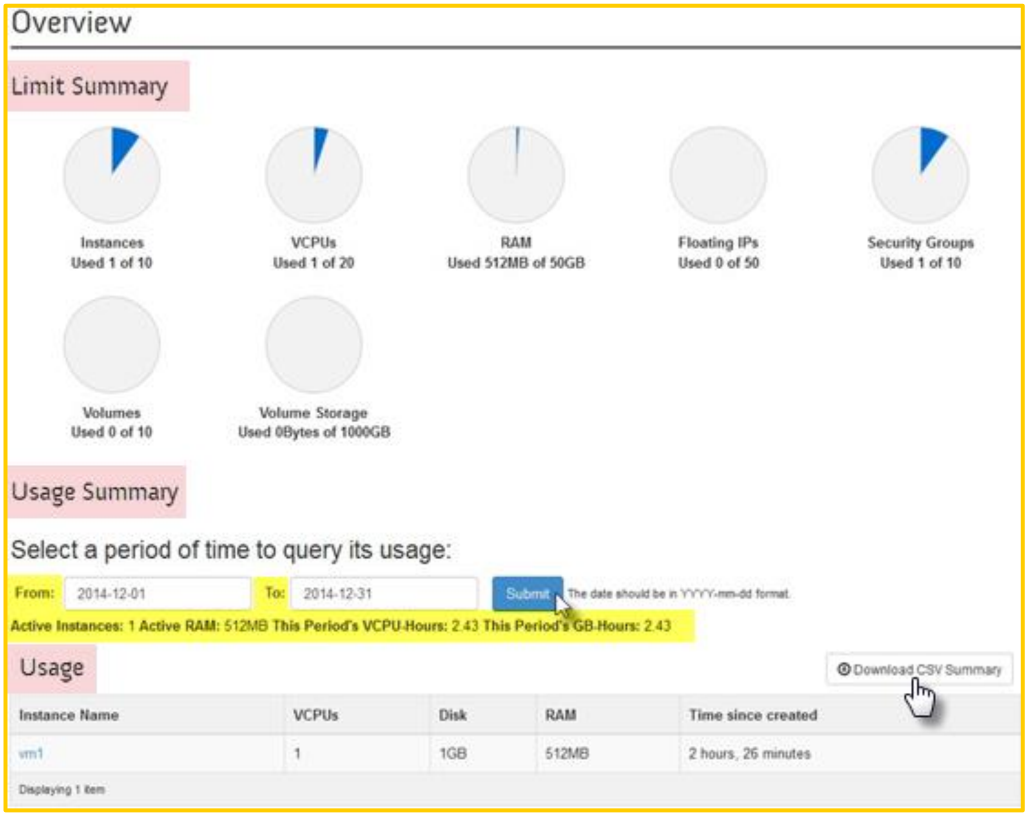
2. Compute Menu Functions

bluvalt **Compute service** provides you with the capability to manage and monitor an Infrastructure-as-a-Service (IaaS) cloud computing platform. It enables users to provision instances and networks, along with empowering users to manage cloud access through **Users** and **Projects**.

2.1. Track Instances Usage

Project → Compute → Overview

The **Overview** menu item demonstrates a simple graphical recap for the VMs' usage limit dimensions, like the number of VCPUs, disks, RAM, and uptime for all of these instances. This page also provides a summary of how the project is doing in relation to the project quotas under the **'Limit Summary'** section. On the other hand, the **'Usage Summary'** section displays the overall consumed quota of the currently running instances/active Instances - within a specified period of time – along with detailed information on usage per VM at the **'Usage'** section.



Overview

Limit Summary

Usage Summary

Select a period of time to query its usage:

From: 2014-12-01 To: 2014-12-31 Submit The date should be in YYYY-mm-dd format.

Active Instances: 1 Active RAM: 512MB This Period's VCPU-Hours: 2.43 This Period's GB-Hours: 2.43

Usage

Instance Name	VCPUs	Disk	RAM	Time since created
vm1	1	1GB	512MB	2 hours, 26 minutes

Displaying 1 item

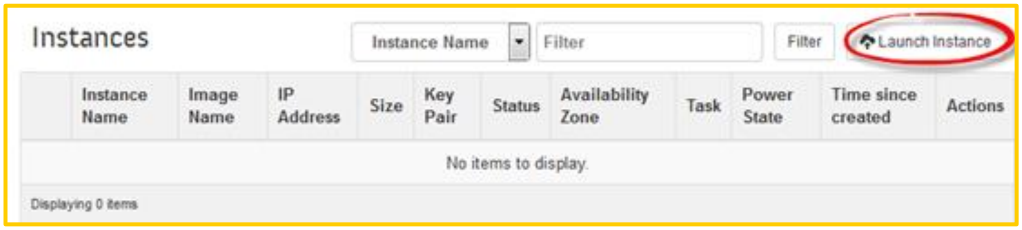
A user can simply enter the desired **From - To** dates in (*YYYY-mm-dd*) format, and click **Submit.** Then, you can preview all usage details, or export the grid content to a CSV file via **Download CSV Summary**.

2.2. Launch an Instance

Project → Compute → Instances

Instances are virtual machines (VMs) that run in the cloud. To launch a new instance, please apply the following steps:

- On the **Instances** page, click the **Launch Instance** button at the top right corner. Then, the **Launch Instance** dialog will appear:

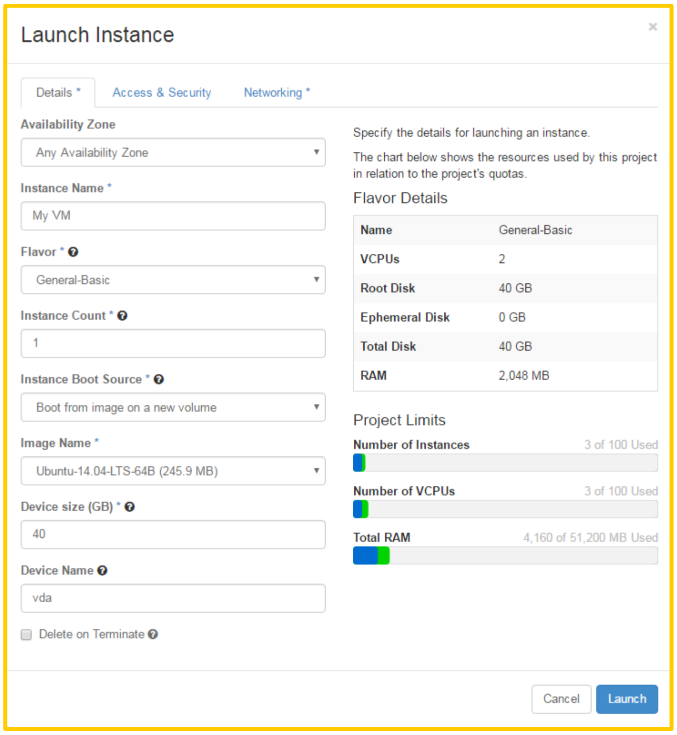


Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
					No items to display.						

Displaying 0 items

- Specify the needed values at the **Launch Instance 'Details'** tab:



Launch Instance

Availability Zone: Any Availability Zone

Instance Name *: My VM

Flavor *: General-Basic

Instance Count *: 1

Instance Boot Source *: Boot from image on a new volume

Image Name *: Ubuntu-14.04-LTS-64B (245.9 MB)

Device size (GB) *: 40

Device Name: vda

□ Delete on Terminate

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	General-Basic
VCPUs	2
Root Disk	40 GB
Ephemeral Disk	0 GB
Total Disk	40 GB
RAM	2,048 MB

Project Limits

Number of Instances: 3 of 100 Used

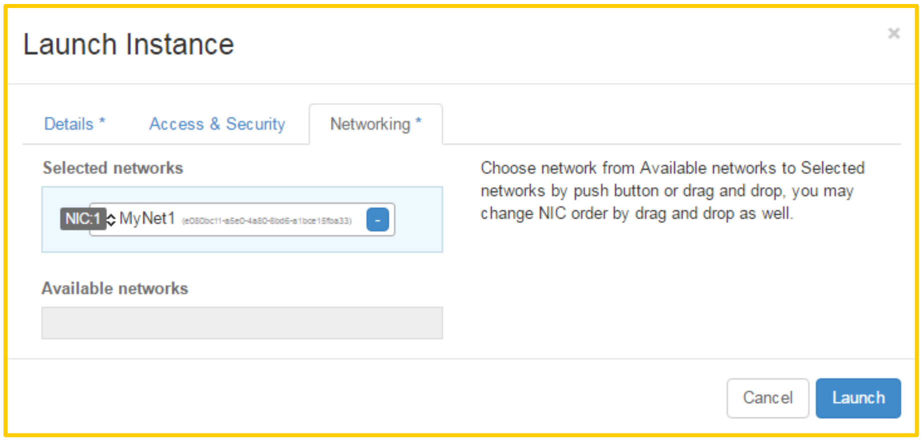
Number of VCPUs: 3 of 100 Used

Total RAM: 4,160 of 51,200 MB Used

Field	Description
Availability Zone	Availability zones provide instances a form of physical isolation and redundancy from other availability zones.
Flavor	An available hardware configuration for an instance. ***bluvalt*** comes with a list of default flavors to be used by all users. The **Launch Instance** dialog provides the ***Flavor details*** of the selected flavor and how it will count against the *Project Quotas*.
Instance Boot Source	Boot from a predefined image with an Operating system or boot from an existing volume with an operating system.
Access & Security	This tab controls connectivity to an instance via SSH key pairs, security groups, and other mechanisms.
Device Size	The size of the root volume (C: drive in Windows) of the instance.
Device Name	The mount point for the volume on a LINUX instance (this option has no effect on Windows instances).

- In the **Launch Instance 'Access & Security'** tab, select the needed security groups that you want to apply on the VM. In addition, you can select the Key pair that will be used to connect to the instance via SSH (this does not apply to Windows instances).

- In the **Launch Instance 'Networking'** tab, select the networks that will be connected to the instance. The instance will get a network interface for each selected network.



Selected networks

NIC:1 MyNet1 (e080bc11-a5e0-4a80-8bd6-a1bce15fba33)

Available networks

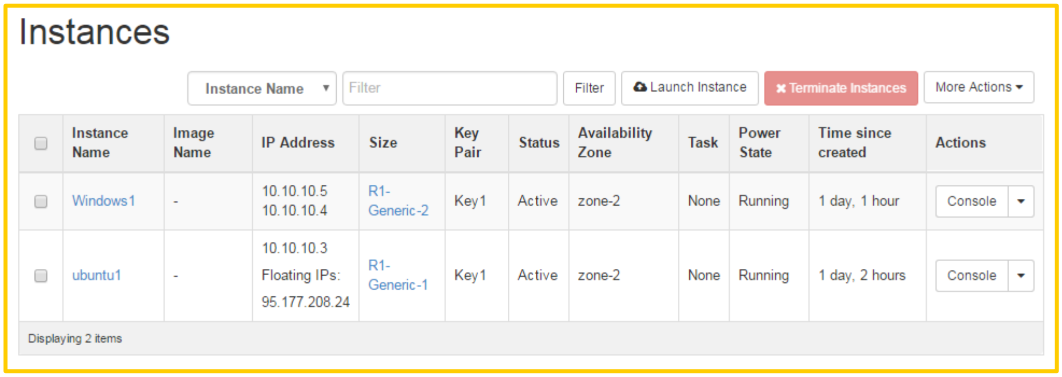
Choose network from Available networks to Selected networks by push button or drag and drop, you may change NIC order by drag and drop as well.

- Afterwards, press the **Launch** button and pay attention to how the *Status, Task* and *Power State* fields change for the new VM.
- Once the instance is in the **Active** status, then the **Instance** creation is completed and it is ready to be used.

2.3. Manage an Instance

Project → Compute → Instances

Users can perform various management tasks on the created instance, as shown below. Click the **More** ▾ button per instance record, to expand the drop down list of actions available for this instance.

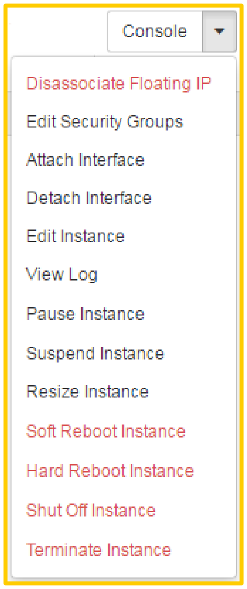


□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	Windows1	-	10.10.10.5 10.10.10.4	R1-Generic-2	Key1	Active	zone-2	None	Running	1 day, 1 hour	Console ▾
□	ubuntu1	-	10.10.10.3 Floating IPs: 95.177.208.24	R1-Generic-1	Key1	Active	zone-2	None	Running	1 day, 2 hours	Console ▾

Displaying 2 items

There are multiple actions available for an instance in **Active** status:

Action	Description
Console	Use it as an alternative way to navigate to the page with VNC console.
Associate/ Disassociate Floating IP	Use this option to associate/disassociate a public IP address for this instance.
Edit Security Groups	Select this option to apply required changes on preset Security Groups.
Attach Interface	Attach a new network interface to the instance.
Detach Interface	Removes a network interface from the instance.
Edit Instance	Click this option to apply any required changes upon the instance record details.
View Log	Click here to navigate to the **Instance Console Log** page to check the instance *dmesg logs (this*



	works only for LINUX VMs, as Windows VMs does not support log viewing).
Pause Instance	Apply this option to store the state of the VM in RAM. A paused instance continues to run in a frozen state.
Suspend Instance	Apply this option to store VM state and memory on the disk, and then stop the VM. Suspending an instance is similar to placing a computer in hibernation.
Resize Instance	Change the size of a server by changing its flavor.
Soft Reboot Instance	Use this option when you attempt to gracefully shut down and restart the instance.
Hard Reboot Instance	Power cycle the instance.
Shut Off Instance	This option is equivalent to cutting the energy source of physical hardware.
Terminate Instance	Click here to delete an instance when it is no longer needed.

2.4. Manage IP Addresses

Project → Compute → Instances

Each instance has a private fixed IP address and can also have a public *(floating)* IP address. *Private IP Addresses* are used for communications between instances that are connected to same virtual router (vRouter). *Public (floating) IP Addresses* are used for communication between instances that are connected to a different virtual router or that are located outside of ***bluvalt***.

When you launch an instance, it is assigned a private IP address on its network interfaces automatically, which stays the same until you *'detach the interface'* or '*Terminate the Instance'*. Note that, '*Rebooting an Instance'* has no effect on the private IP address.

The project quota defines the maximum number of floating IP addresses that you can allocate to the project. After allocating a floating IP address to a project, an admin user can:

- **Associate** the *floating IP address* with an *instance* of the project. Only one floating IP address can be allocated to a Virtual Network Interface instance any given time.
- **Disassociate** a *floating IP address* from an *instance* in the project.
- **Release** a *floating IP* from the project, which automatically **deletes** this *IP's associations*.

Note:

- For more information on how to assign a floating IP address, check elaborations in '5.4 External Network Connectivity to VM (Floating IP)'.

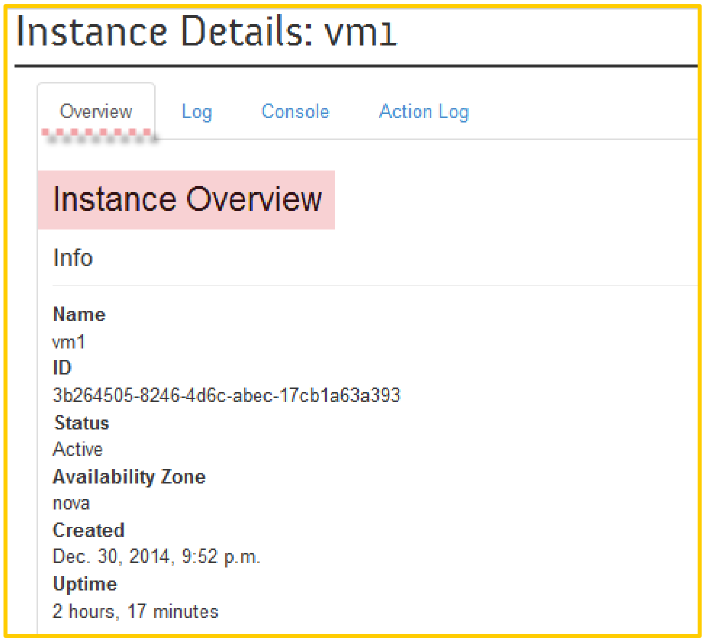
2.5. Connect to an Instance Using the Console

Project → Compute → Instances

In computing, **Virtual Network Computing** (VNC) is a graphical desktop sharing system that uses the RFB protocol to remotely control another computer. It sends keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction over a network. In an Infrastructure-as-a-Service system, *VNC console is a very convenient tool for users to connect to the instances from the user interface*.

1. Follow the above path to reach the **Instances** page, and click the *Instance Name*.

   The **Instance Details** page shows the **'Overview'** tab, associated with more detailed information, as shown below.



Instance Details: vm1

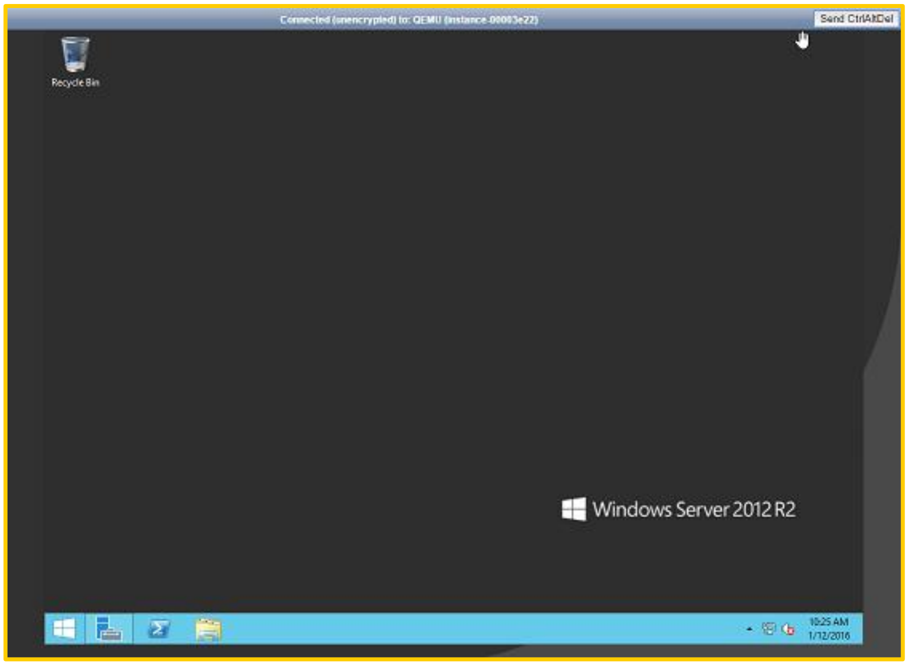
Instance Overview

Info

Name
vm1
ID
3b264505-8246-4d6c-abec-17cb1a63a393
Status
Active
Availability Zone
nova
Created
Dec. 30, 2014, 9:52 p.m.
Uptime
2 hours, 17 minutes

2. Then, click the **'Console'** tab, in order to open the VNC console window on the **Instance Console** page.

3. Log in to the VM using your *Username* and *Password*. If you cannot type at the prompt, click in the grey area first, and then type at the prompt. If you see a command prompt, then *you are successfully connected to the instance using VNC console*.



Important Note:

- LINUX VMs cannot be accessed through the *Console* by default. **You must use KeyPairs to access the LINUX instance for the first time**. If needed, you can set a password for the instance after accessing it through SSH, and then use it with the username to access the LINUX instance through the *Console* window.

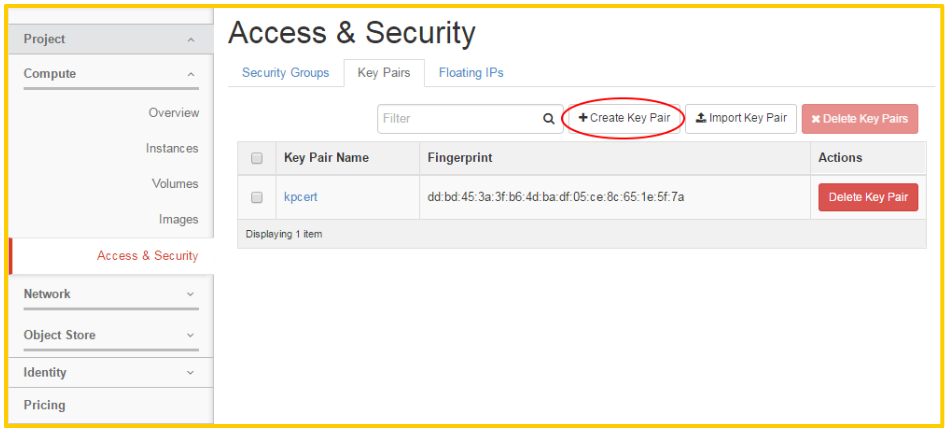
2.6. Connect to a New Linux Instance using Key Pairs

Project → Compute → Access & Security → Key Pairs.

A more secure way to connect to your LINUX VMs is by using **Key Pairs**. Each key pair has two parts, the ***public key*** and the ***private key***. The public key is what is injected into your LINUX instances under the *authorized_key file*. The private key is what you save in a *.pem file* on your local machine. You can use your private key to SSH into your LINUX instances. These keys are injected into your LINUX instances to make password-less SSH access to the instance possible.

To use a key pair, you should first create one, and then assign it to an instance during the VM creation process.

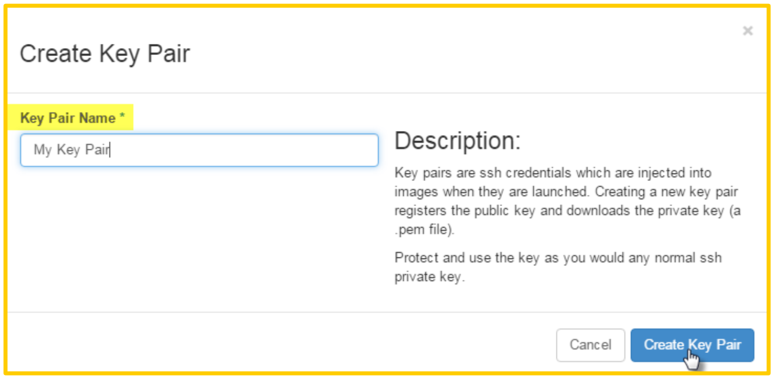
- Navigate to this path; ***Project → Compute → Access & Security → Key Pairs***. Then, click the **+ Create Key Pair** button, as shown below.



□	Key Pair Name	Fingerprint	Actions
□	kpcert	dd:bd:45:3a:3f:b6:4d:ba:df:05:ce:8c:65:1e:5f:7a	Delete Key Pair

Displaying 1 item

- Enter the **Key Pair Name** at the provided field, and then click **Create Key Pair** button.



Create Key Pair

Key Pair Name *

My Key Pair

Description:

Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).

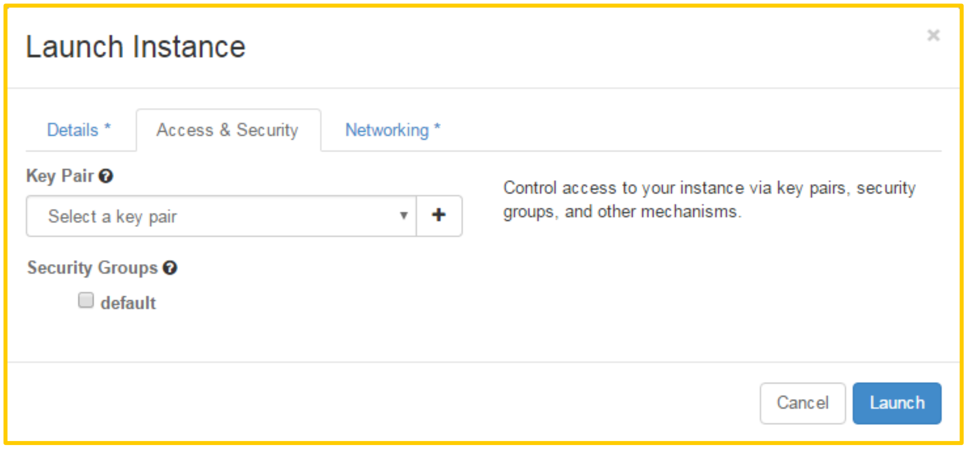
Protect and use the key as you would any normal ssh private key.

Hence, the key *.pem file* will be downloaded to your browser, and you will be able to use it to connect remotely to your server through SSH.

Important Note:

- Please **save** your key pair private key and don't lose it, for security reasons *re-downloading the key pair private key again is not possible*. In case you lost the Key pair private key, you can take a snapshot of the VM and launch a new instance from the snapshot and assign a new key pair to it.

  - Launch a new instance and assign the key pair through the **'Access and Security'** tab.



Launch Instance

Key Pair

Select a key pair

Security Groups

□ default

Control access to your instance via key pairs, security groups, and other mechanisms.

  - Assign a *floating IP* to the instance.
  - Make sure that port 22 (SSH) is allowed through the Security Group used by the instance. For more elaborations, check heading '2.9. Working with Security

Groups'.

Notes:

- Upon creating & configuring an instance, it will be accessible to users through **SSH** from their workstations.
- LINUX and UNIX based VMs default username can be found in the information of the image that is used to create the VM. For more information, please check section '4. Images'.

- With the native OpenSSH client on a *LINUX workstation*, you can use this *.pem-file* directly. If you are using a *Windows workstation*, you will need to use a third party SSH app, as Windows does not include an SSH client.
- The most popular Windows SSH Clients are “Xshell” and “Putty”. “Xshell” works by importing the *.pem-file* directly, while “PuTTY” does not work with .pem-files. Thus, you have to convert your key first, by completing the following steps:
  - a) Download “PuTTYgen” and “PuTTY” from (*www.putty.org*).
  - b) Start PuTTYgen.
  - c) Click **Load**.
  - d) Browse to the location of the private key file that you want to convert (*Note*: PuTTYgen displays only files with extension *.ppk* by default. Therefore, you will need to change that to display files of *all types*, in order to see your *.pem key file*).
  - e) Select your *.pem key* file and click **Open**.
  - f) When you click **OK**, PuTTYgen will display a dialog box with information about the loaded key, such as the **Public Key** and the **Fingerprint**.
  - g) *Optional step*: Enter and confirm a key **Passphrase**. Note that, if you use a passphrase, you will have to enter this passphrase whenever you authenticate with your key.
  - h) Click on **Generate** to generate the public/private key pair.
  - i) Click **Save Private Key** to save the key in PuTTY's format.
- Now you can use the converted private key in Putty by following these steps:
  - **Open** Putty.
  - Go in the tree on the left to ***Connection → SSH → Auth.***
  - Click on the **Browse** button under Private key file for authentication.

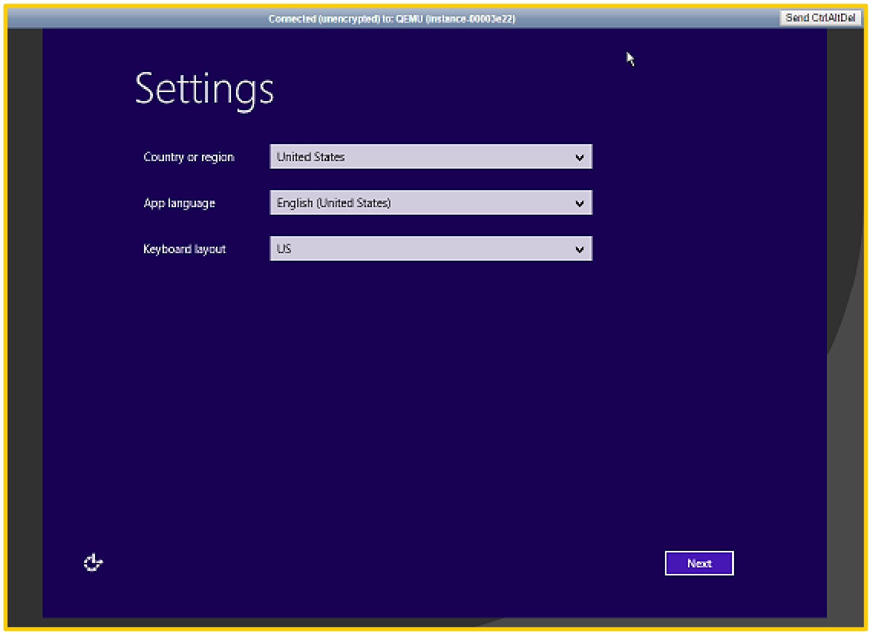
- Select the PPK-file (your private key) you just saved.
- Go in the tree on the left to **Session**.
- Enter in the **Host Name (or IP address)** field, the username and floating IP address of the instance.
- *Optional step*: Enter a name for the session in the **Saved Sessions** field and click **Save**. This saves all the settings, including the private key for this session.
- Click **Open** to connect to the instance.
- When you connect for the first time, you will be asked if you trust this computer. Normally you can click **Yes**.
- If everything works alright you are now logged in.
- If you haven't entered the username with the IP/hostname in the **Host Name** field, you will be prompted to enter it within instance login.

2.7. Connect to a New Windows Instance through RDP

Project → Compute → Instances

To access a Windows Instance through RDP, apply the following steps:

1. **Launch** a new Windows instance.
2. On the **Instance** page, click the **Instance Name**. Then, the **Instance Details** page appears with **'Overview'** tab displayed by default.
3. Click the **'Console'** tab. You should see the VNC console window on the **Instance Console** page.



Connected (unencrypted) to: QEMU (instance-00003e22)

Settings

Country or region: United States

App language: English (United States)

Keyboard layout: US

4. Complete Windows Settings.

5. *Assign a Floating IP* to the instance. For more elaborations, check '5.4 External Network Connectivity to VM (Floating IP)' below.

   - Make sure that port 3389 (RDP) is allowed through the **Security Group** used by the instance. For more elaborations, check heading '2.9: Working with Security Groups'.

   - Make Sure RDP is enabled in the Windows VM and allowed through its firewall.

6. Use your **RDP client** to connect to the Windows instance.

2.8. Configure Access and Security for Instances

A **Security Group** is a collection of network access rules that are used to limit the types of traffic that have access to instances. Hence, **Security Groups** are applied to instances ports directly.

- When you launch an instance, you can assign one or more security groups to it.

- If you launch an instance and did not assign a security group to it, it will automatically be assigned to the default security group.

- The associated rules in each security group control the traffic to instances in the group.
- You can add rules to allow inbound and outbound traffic.
- You cannot add rules that deny traffic; any inbound and outbound traffic that is not matched by a rule is denied access by default.
- You cannot modify rules for the default security group.

The '*Default' Security group* is available by default and it cannot be deleted or changed. It includes the following rules:

- Allow all outbound network traffic for VMs.
- Allow inbound network traffic *between* VMs only.

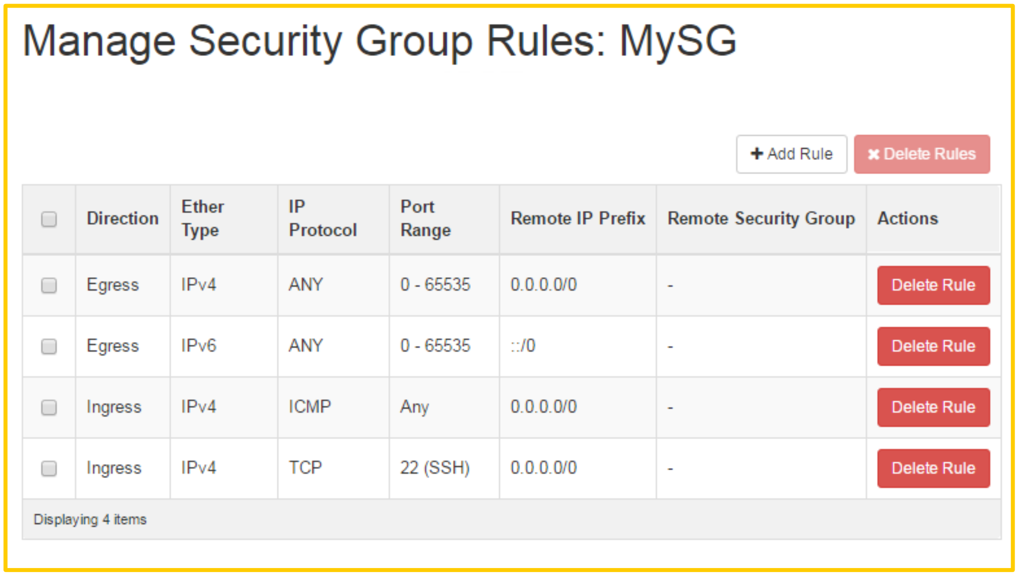
Important Notes:

- All external remote access (SSH, RDP or HTTP) attempts to any VM are not allowed by default. Thus, an appointed admin user should configure access rules first to allow it.
- When creating a new security group, it will contain a rule by default to allow all outgoing traffic. You can delete this rule, if required.

To create a New Rule that allows access through a protocol and port in a **Security Group**, apply the following steps:

1. Navigate to this path; Project → Compute → Access & Security.
2. Click on the **'Security Groups'** tab to show the security groups that are available for this project.
3. Select the required security group, and click **Manage Rules**.

4. Then, click **+ Add Rule** button.

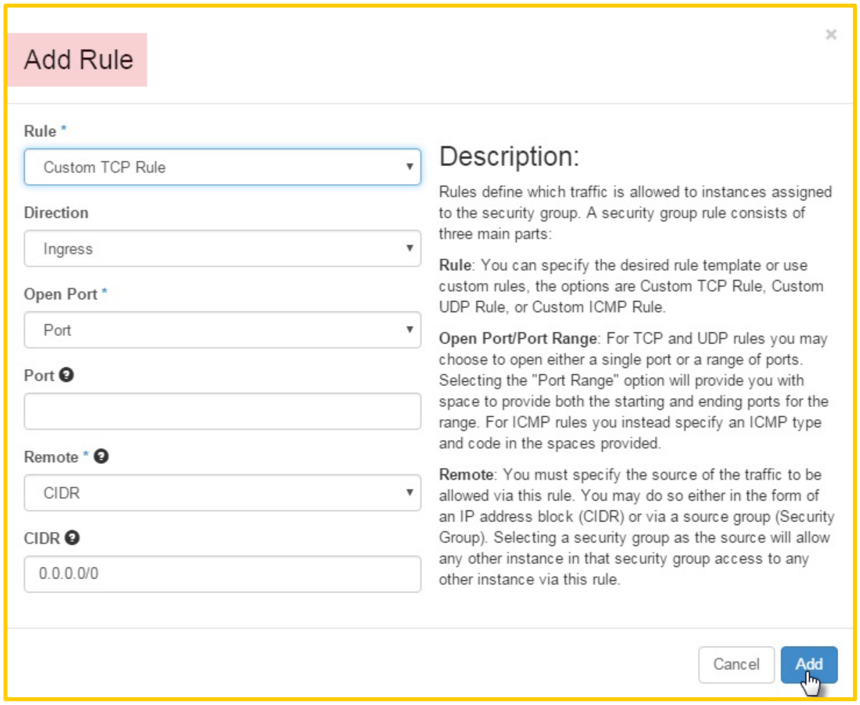


Manage Security Group Rules: MySG

□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
□	Egress	IPv4	ANY	0 - 65535	0.0.0.0/0	-	Delete Rule
□	Egress	IPv6	ANY	0 - 65535	::/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

Enter the required rule data at the **Add Rule** dialog box, as follows:



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.

Field	Description
Rule	Choose the type of rule you want to create. Note that, there are pre-defined rules, yet a user is eligible to create his own TCP/UDP/ICMP custom rule.
Direction	*Ingress* (inbound rule) or *Egress* (outbound rule).
Open Port	Choose either *Port* or *Port Range*.
Port	In case of choosing a custom rule, enter the port number.
Port Range	In case of choosing a custom rule, enter your port range.
Remote	Choose either CIDR to select an *IP range* or *Security Group* to allow a specific security group: ○ **CIDR**: (Classless Inter-Domain Routing) If you chose this option, enter an IP range. (e.g. 192.168.10.0/24). ○ **Security Group**: if you chose this option, select the group to allow relevant access.

5. Finally, click **Add** button to save the record. Once you made a change to the security group rules and saved the record, the change will be effective immediately for all instances using that security group.

2.9. Working with Security Groups

2.9.1. Allowing Inbound and Outbound Traffic between VMs

Here we will need to configure the security group to allow Ingress and Egress traffic between VMs in the security groups:

- Add a rule in the used *Security Group* for the source VM to allow outgoing traffic from the **Source** VM:
    - **Direction**: Egress.
    - **Port Range**: Choose a single port or define a range.
    - **Remote**: Enter the Security Group name**.** (you can use CIDR instead by entering the source IP address/range)
- Add a rule in the used *Security Group* for the destination VM to allow incoming traffic to the **Destination** VM:
    - **Direction**: Ingress.
    - **Port Range**: Choose a single port or define a range.
    - **Remote**: Enter the Security Group name. (you can use CIDR instead by entering the destination IP address/range)

Example

Allowing ping (ICMP) traffic (inbound and outbound) between 2 instances in the same network and associated with different Security Groups SG1 and SG2 prospectively

To accomplish this, you will need to add 2 rules in each Security Group, one to allow outbound ICMP from the VMs assigned to it and the other to allow inbound ICMP to the VMs assigned it:

SG1 Rule 1 configurations:

- **Rule**: ALL ICMP
- Direction: Egress
- **Remote**: Security Group
- Security Group: SG1

SG1 Rule 2 configurations:

- **Rule**: ALL ICMP
- **Direction**: Ingress
- **Remote**: Security Group
- Security Group: SG2

SG2 Rule 1 configurations:

- **Rule**: ALL ICMP
- Direction: Egress
- **Remote**: Security Group
- Security Group: SG2

SG2 Rule 2 configurations:

- **Rule**: ALL ICMP
- **Direction**: Ingress
- **Remote**: Security Group
- Security Group: SG1

2.9.2. Allowing Outbound Traffic from a VM to outside the Cloud

Here is how to configure **Security Groups** to control network access:

- Add a rule in the used *Security Group* for the **source** VM to allow outgoing traffic to the **Destination** IP(s) from the **Source** IP(s):
    - **Direction**: Egress.
    - **Port Range**: Choose a single port or define a range.
    - **Remote**: Enter the IP address(s) of the **Destination** IP(s).

2.9.3. Allowing Inbound Traffic to a VM from outside the Cloud

Here is how to configure **Security Groups** to control the network access:

- Add a rule in the used *Security Group* for the **Destination** VM to allow incoming traffic to the **Destination** VM from the **Source** IP(s):
    - **Direction**: Ingress.
    - **Port Range**: Choose a single port or define a range.
    - **Remote**: Enter the public IP address of the **Source** IP(s).

Example

Open incoming SSH traffic from the IP address 192.168.100.100 (outside the cloud) to instances in the *Security Group* “SG1”.

- To accomplish this, add 1 rule to the **Security Group** “SG1”:
    - **Rule**: Custom TCP Rule
    - **Direction**: Ingress
    - Port Range: 22
    - **Remote**: CIDR

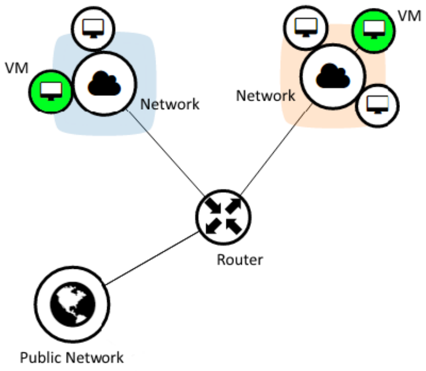
- **CIDR**: 192.168.100.100/32

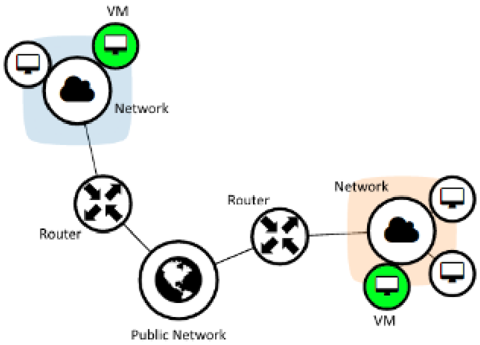
You can also use the preconfigured SSH Template:

- **Rule**: SSH
- **Remote**: CIDR
- **CIDR**: 192.168.100.100/32

2.9.4. Allowing Inbound Traffic to a VM from another VM on a different Network connected to a different router

If two networks are connected to the same router the VMs inside them can communicate using their private IP addresses, therefore we can use the VMs private IPs in the Security Groups rules. However, if the two networks are connected to different routers, the traffic will pass through the public network to reach the other router and then to the destination network. This means that the appointed user will need to allow traffic in the Security Groups using the Public (floating) IP addresses of the VMs. The difference is explained in the below figures:





Communication through Private IPs

Communication through Public (Floating) IPs

NOTE:

- For further explanation on how networking works, please check section '5. Networking'.

Here is how to configure the Security Groups to control the network access in the above scenario:

- Add a rule in the used *Security Group* for the **Source** VM, to allow outgoing traffic from the **Source** VM to the **Destination** VM:
  - **Direction**: Egress.
  - **Port Range**: Choose a single port or define a range.
  - **Remote**: Enter the **Public** (Floating) **IP** address of the **Destination** VM.
- Add a rule in the used *Security Group* for the **Destination** VM, to allow incoming traffic to the **Destination** VM from the **Source** VM:
  - **Direction**: Ingress.
  - **Port Range**: Choose a single port or define a range.
  - **Remote**: Enter the **Public** (Floating) **IP** address of the **Source** VM.

Example

Allowing inbound communications on port 3306 (MySQL) from the instances “VM1” in the network “My Net1” which is connected to the router “My Router 1”, the VM is associated with *Security Group* “SG1”, to the MySQL instance “MySQLVM” in the network “My Net2” which is connected to the router “My Router 2”, the VM is associated with the *Security Group* “SG2”.

	Source	Destination
Instance	VM1	MySQLVM
Private IP	10.10.10.13	30.30.30.11
Public IP	192.168.100.11	192.168.200.12
Security Group	SG1	SG2
Network name	My Net	My Net 2
Router	My Router 1	My Router 2

Since the source and destination are on different networks that are connected to different routers, then they will not be able to communicate through their private IP addresses. They can only communicate through their public (floating) IP addresses. Therefore, we will need to configure two rules, one to *allow outbound traffic from “SG1” to the destination public IP address*, and another *rule in “SG2” to allow inbound traffic from the source public IP address*.

“SG1” Configurations:

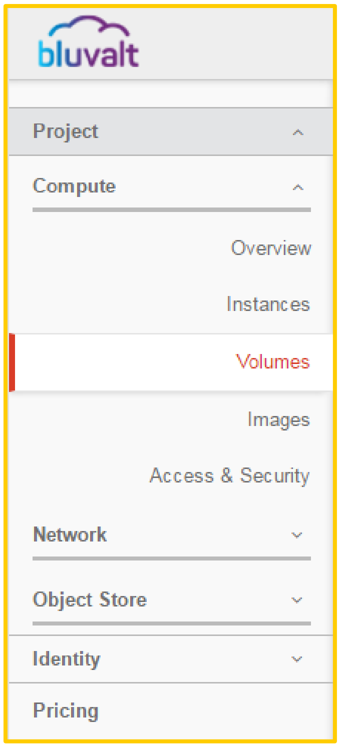
- **Rule**: Custom TCP Rule
- Direction: Egress
- Port Range: 3306

- **Remote**: CIDR
- **CIDR**: 192.168.200.12/32

“SG2” Configurations:

- **Rule**: Custom TCP Rule
- **Direction**: Ingress
- Port Range: 3306
- **Remote**: CIDR
- **CIDR**: 192.168.100.11/32

3. Volumes



bluvalt Volumes are provided as a **block storage service**. When you create a new instance in ***bluvalt***, it gets stored on a new volume. You can create a custom volume and attach it to a running instance or detach a volume and attach it to another instance at any time. You can also create a snapshot from /or delete a volume. Block volumes storage is persistent, so the data is not affected by attaching/detaching it to other instances, the data will be lost only when you delete the volume.

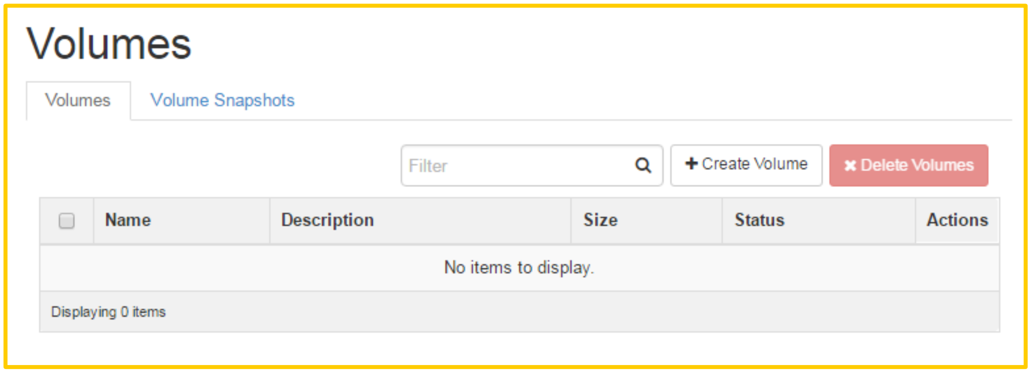
bluvalt **volumes (Block Storage service)** provide a reliable way to store vital data; such as database files and application data.

3.1. Create a Volume

Project → Compute → Volumes

To create a volume in the Block Storage service and attach it to a VM, apply the following steps:

1. Navigate to this path; ***Project → Compute → Volumes***, in order to check available volumes.
2. Then, click **+ Create Volume** button.



□	Name	Description	Size	Status	Actions
No items to display.					

Displaying 0 items

3. Enter/select the following values at the dialog box listed fields, as follows:
   - **Volume Name**: Enter a significant name for this volume.
   - **Description**: It is optional if you choose to provide a brief description for this volume.
   - **Size (GB):** Enter the size of the volume in gigabytes (GB).
   - **Availability zone:** Availability zones provide volumes a form of physical isolation and redundancy from other availability zones.
4. After entering the required data, click **Create Volume**. The new volume will be created and is ready to be assigned to an instance.

3.2. Attach a Volume to an Instance

Project → Compute → Volumes

After creating one or more volumes, a user can attach them to instances. You can attach a volume to one instance at a time, as follows:

1. Follow the above path to reach the **Volumes** page.
2. Select the required volume to add to an instance, and click **Manage Attachments**.
3. In the **Manage Volume Attachments** dialog box, select the required *Instance*.
4. It is optional to choose to enter the name of the device from which the volume is accessible by the instance *(this is valid only for LINUX VMs only)*
5. Click Attach Volume.

Notes:

- The dashboard shows the *Instance* to which the volume is now attached and the *Device Name*.
- It is worth mentioning; a user can view the status of a volume in the **Volumes** tab at the dashboard. The volume is either *Available* or *In-Use*.
- To use the attached volume, you will need to log in to the instance, mount and format it. You can also detach the volume from the VM and attach it to another VM, as required.

Important Note:

- You will need to unmount the device from your instance prior to detaching it, to avoid any data corruption in the volume.
- Detaching a volume before unmounting it from the OS may also cause the VM to hang.

3.3. Extend a Volume

Project → Compute → Volumes

A volume size can only be extended and not reduced. To extend a volume size:

- Follow the above path to reach the **Volumes** page.
- Select the volume to be extended.
- In the **Actions** column of the volume to be extended, click **Extend Volume**. (Verify that the volume is not attached to any VM, or the Extend option will not be available).
- In the **Extend Volume** dialog box, enter the new size of the volume.
- Click **Extend Volume**.

3.4. Volume Snapshot

Project → Compute → Volumes

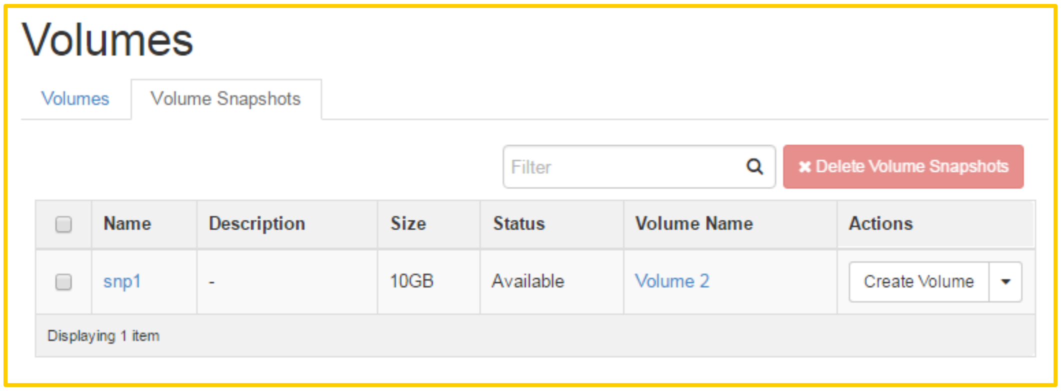
A Snapshot is a copy of a volume at a given point in time. Snapshots are used to restore a VM or attached volume to a particular point in time when a failure or system error occurs. By taking a snapshot of a volume, ***bluvalt*** creates a *replica of your volume*. Then, you can create a new volume based on the snapshot and boot a new instance from it (if the volume is a bootable OS) or attach it to an existing instance. You can also create a new volume from the volume snapshot and increase the volume size.

To create a volume snapshot:

- Follow the above path to reach the **Volumes** page.
- In the **Actions** column of the volume you want to take a snapshot of, click **Create Snapshot** (In some cases, creating a snapshot from an attached volume can result in a corrupted snapshot).
- In the **Create Volume Snapshot** dialog box, enter the snapshot Name.

- Click Create Volume Snapshot.

You will find the snapshot in the **'Volume Snapshots'** tab, as shown below.



Volumes

□	Name	Description	Size	Status	Volume Name	Actions
□	snp1	-	10GB	Available	Volume 2	Create Volume ▾

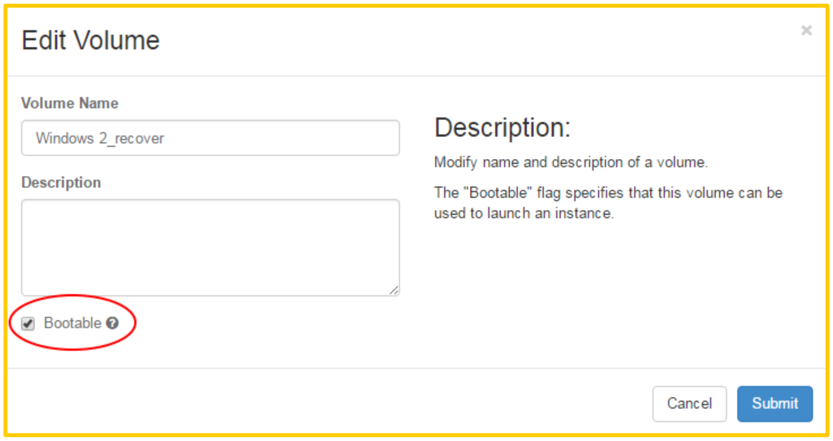
Displaying 1 item

- You can then create a new volume from the snapshot and attach it to an instance.

3.5. Launch an Instance from a Volume Snapshot

Snapshots for an instance volume are taken as a recovery point. You can use this to recover your VM in case of an OS corruption. To launch a new instance from a created instance volume snapshot, apply the following steps:

1. Navigate to the **Volume Snapshots** and locate the instance volume snapshot and create a new volume from it.
2. You will see the new volume in the **Volumes** list.
3. On the Action menu of the volume, click on **Edit Volume** and make sure that it is bootable.



Volume Name: Windows 2_recover

Description

☑ Bootable

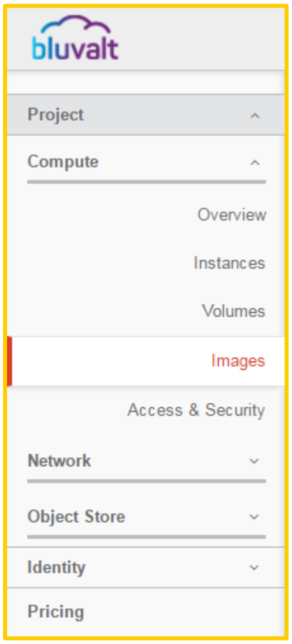
Description:

Modify name and description of a volume.

The "Bootable" flag specifies that this volume can be used to launch an instance.

4. Under *Project* → *Compute* → *Instances*, Click Launch Instance.
5. Fill in the required information.
6. Under **instance Boot Source,** select "*boot from an existing volume*".
7. Under **Volume,** you will find the volume that you created from the snapshot.
8. Complete the launch instance process and the new VM instance will be created with everything retained to the VM snapshot point.

4. Images

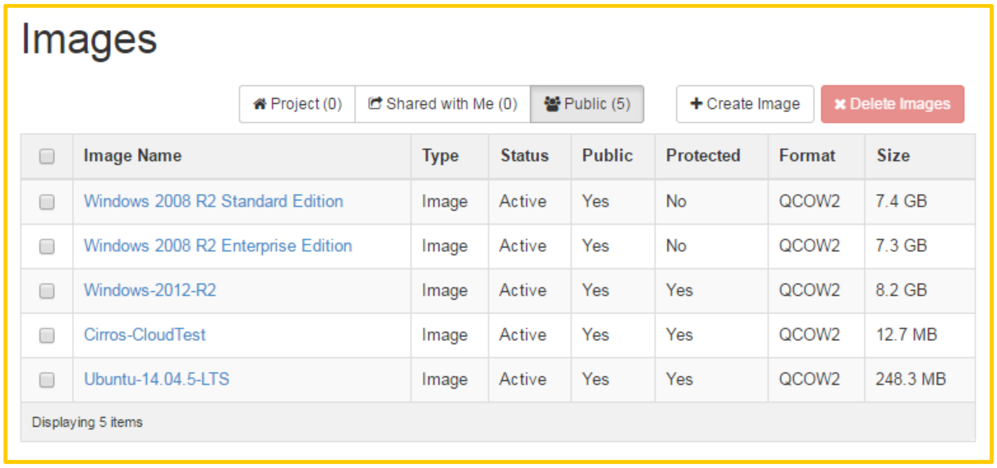


A **Virtual Machine Image**, referred to in this document simply as an image, is a single file that contains a virtual disk, which has a bootable operating system installed. Images are used to create virtual machine instances within the cloud.

By default, ***bluvalt*** includes a variety of OS images that can be used to launch new instances from. By clicking on the image name you can find information regarding each OS image, like the creation and image update date, image size and minimum required volume size.

Important Note:

- LINUX and UNIX based VMs information also include the default username that is used to access the instance through SSH.



□	Image Name	Type	Status	Public	Protected	Format	Size
□	Windows 2008 R2 Standard Edition	Image	Active	Yes	No	QCOW2	7.4 GB
□	Windows 2008 R2 Enterprise Edition	Image	Active	Yes	No	QCOW2	7.3 GB
□	Windows-2012-R2	Image	Active	Yes	Yes	QCOW2	8.2 GB
□	Cirros-CloudTest	Image	Active	Yes	Yes	QCOW2	12.7 MB
□	Ubuntu-14.04.5-LTS	Image	Active	Yes	Yes	QCOW2	248.3 MB

Displaying 5 items

4.1. Upload a Custom Image

If the list of available images does not include the required image, ***bluvalt*** enables you to upload an OS image of your choice and launch a new instance from the image you uploaded.

Bluvalt only accepts images in QCOW2 format, so in order for your image to work it must be converted to the QCOW2 format first. To do that you will need to use an image converting tool. If you are on a Linux system, you can download the **qemu-img** package and use it for the conversion. If you are on a Windows system, you can download the **QEMU disk image utility (quemu-img.exe)** and use it for the conversion

Converting a Linux based VM:

1. if the Linux VM is hosted in a hypervisor, Export the Linux VM from the hypervisor, or you can take a snapshot of the VM and locate the snapshot image location
2. Use the qemu image converter tool to convert the image to the QCOW2 format. For example:
    - In Linux: *qemu-img convert source.vhd -O qcow2 dest.qcow2*
    - In Windows*: qemu-img.exe convert source.vhd -O qcow2 dest.qcow2*

Converting a Windows based VM:

Converting a Windows VM is not a straightforward process; you will need to install the required drivers in the VM to make it compatible with ***Bluvalt*** platform before converting it to the QCOW2 format.

1. Please follow the documented guide located in our support pages on how to convert a Windows VM and install all the required drivers on the VM: https://support.bluvalt.com

2. After the VM shuts down locate its location

3. Use the qemu image converter tool to convert the image to the QCOW2 format. For example:

    - In Linux: *qemu-img convert source.vhd -O qcow2 dest.qcow2*
    - In Windows*: qemu-img.exe convert source.vhd -O qcow2 dest.qcow2*

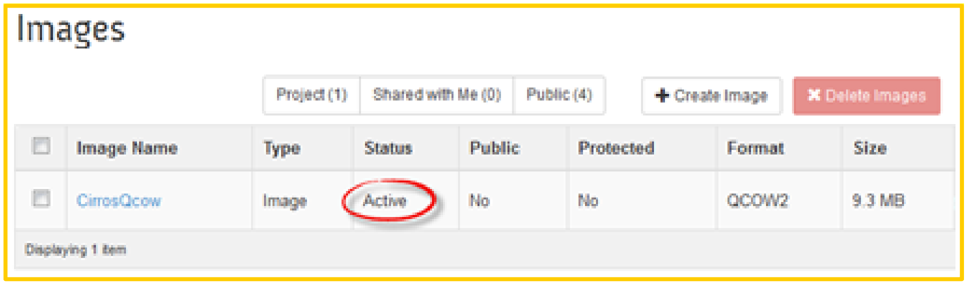
Uploading the image to *Bluvalt*:

After converting the VM image to the compatible format, you will be ready to upload the image, you will need to contact our support team; they will instruct you on how to upload your image.

Important Note:

Microsoft product use rights do not allow the use of License Mobility for Windows licenses. Therefore, to upload a Windows image you must provide us with the admin password so we can change the license key.

Once the image have been uploaded you will find it in the images list under the **Projects** tab



Images

□	Image Name	Type	Status	Public	Protected	Format	Size
□	CirrosQcow	Image	Active	No	No	QCOW2	9.3 MB

Displaying 1 item

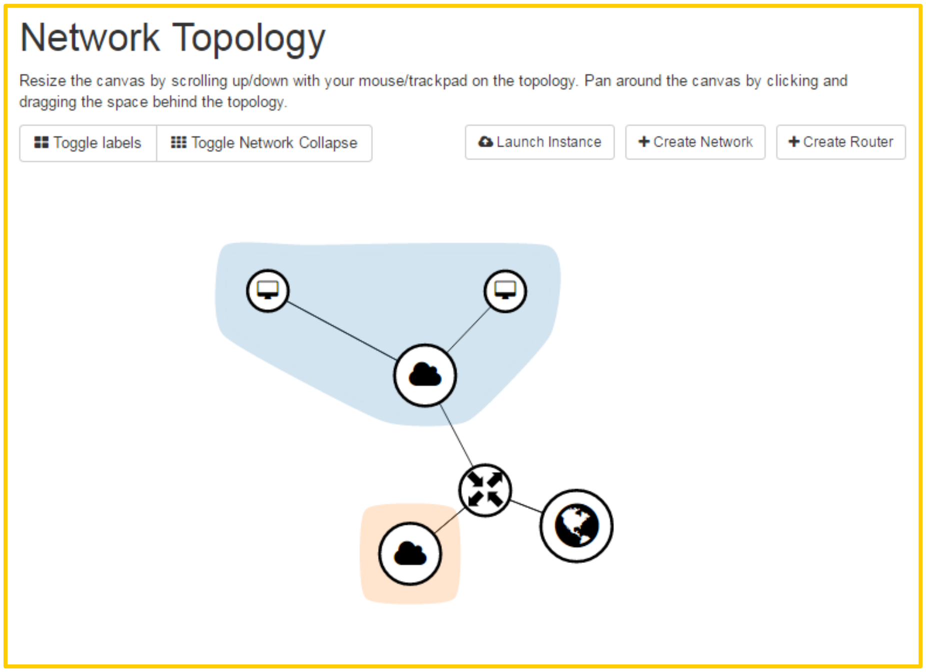
5. Networking

bluvalt provides networking services such as *L3, IP Address Management (IPAM), routing between IP subnets and to the outside world, and more*.

When creating a new instance, **Virtual Network Interfaces** (VIFs) are created for it. For network communication between VMs to function properly, VIFs of different VMs need to be wired together using virtual switches. For communications between different virtual switches, they need to be wired to a virtual router.

5.1. nReview Existing Networks Configuration

Navigate to this path; ***Project → Network → Network Topology***, in order to check the network topology.



Network Topology

Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

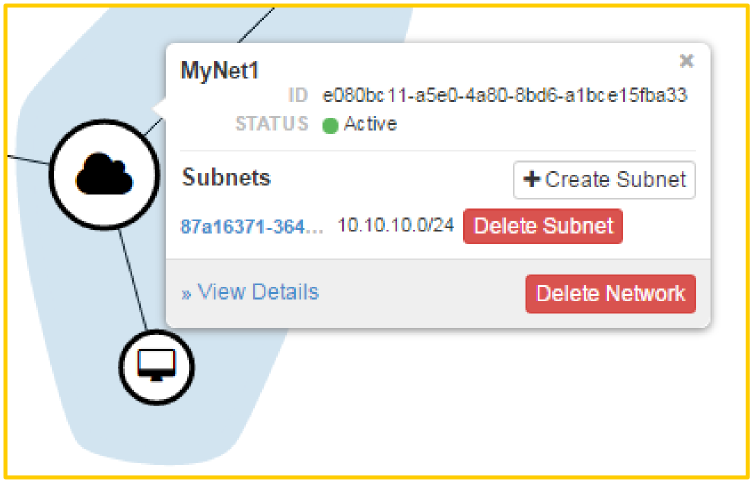
In the above image, you can see the communications between the virtual devices in the project. You can notice that each network is highlighted in a different color. The Icons represents the VMs and network elements in the project, you can also choose to view the devices labels by pressing the **Toggle Labels** button:

Icon	Description
🖥 **Virtual Machine**	This is the instance that is attached to a virtual network.
☁	Internal network for VM-to-VM communication. A private network is selected every time we launch VMs.

Virtual Network	
Virtual Router	Networks subnets are connected with a router that routes the traffic from one subnet to another. It also provides a gateway service for the subnet of private network.
Public Internet	External network created by default. Through this network, VMs can connect to the internet and can be reached via internet.

The **Network Topology** diagram is interactive, allowing you to quickly check the details of elements, perform basic actions, and navigate to most common screens for elements.

For example, you can click on the **Network** icon and check the quick overview.


MyNet1

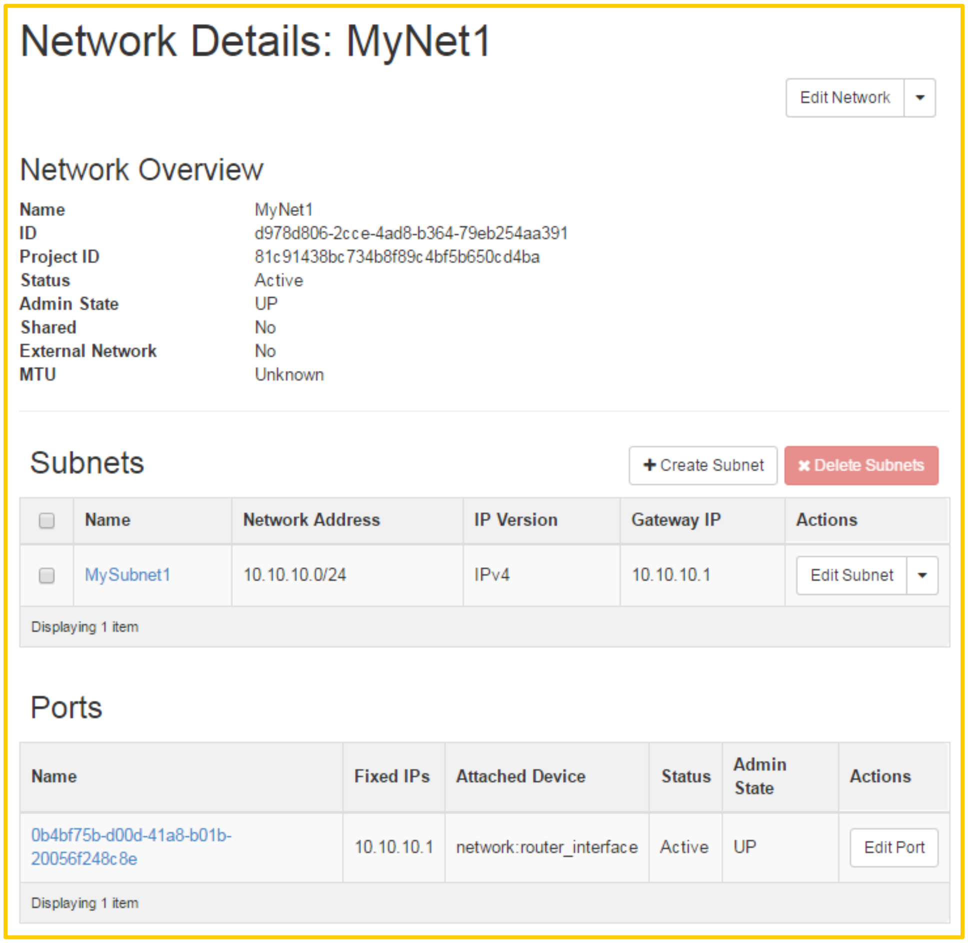
ID e080bc11-a5e0-4a80-8bd6-a1bce15fba33

STATUS ● Active

Subnets

87a16371-364... 10.10.10.0/24

You can click on the **View Details** link to go to the **Network Details** page.



Network Details: MyNet1

Network Overview

Name	MyNet1
ID	d978d806-2cce-4ad8-b364-79eb254aa391
Project ID	81c91438bc734b8f89c4bf5b650cd4ba
Status	Active
Admin State	UP
Shared	No
External Network	No
MTU	Unknown

Subnets

□	Name	Network Address	IP Version	Gateway IP	Actions
□	MySubnet1	10.10.10.0/24	IPv4	10.10.10.1	Edit Subnet

Displaying 1 item

Ports

Name	Fixed IPs	Attached Device	Status	Admin State	Actions
0b4bf75b-d00d-41a8-b01b-20056f248c8e	10.10.10.1	network:router_interface	Active	UP	Edit Port

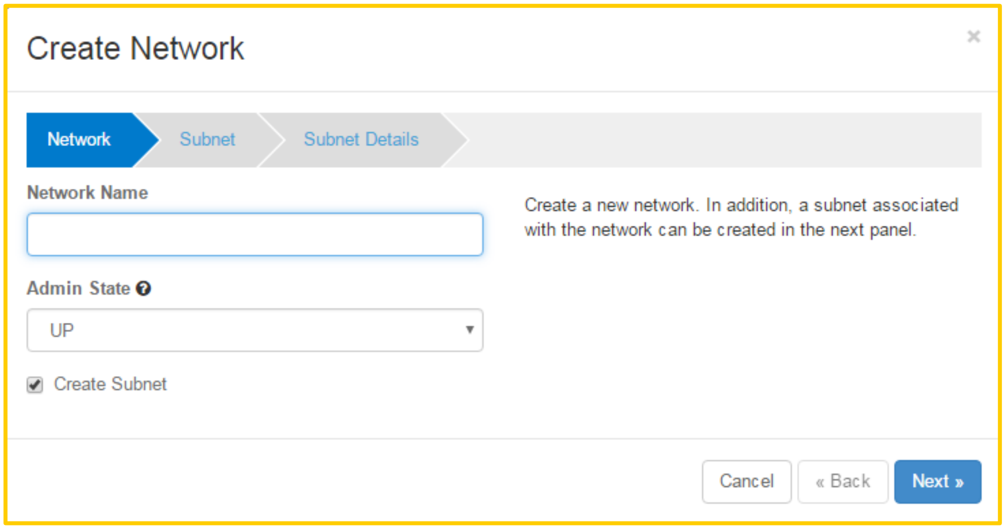
Displaying 1 item

5.2. Create a Network

Apply the following steps to create a Network:

1. Navigate to this path; ***Project → Network → Networks***, in order to check the current networks at the list.
2. Click the **Create Network** button in the top right corner.
3. In the **Create Network** wizard, on the **'Network'** tab specify the Network Name and Admin State (If set to down, the network will be set as down and does not forward any packets), and then press **Next**. (select Create Subnet to create a subnet now or

you can do it later)



Create Network

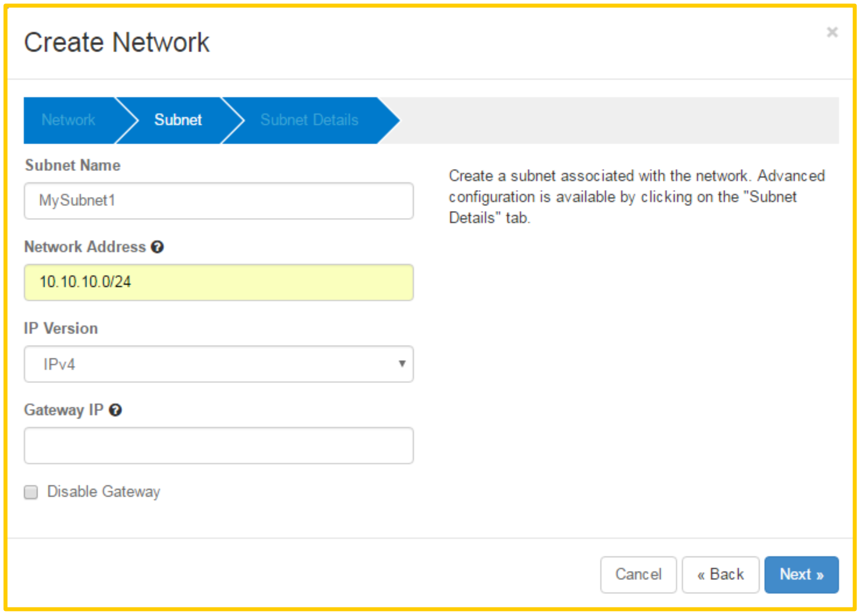
Network Name

Create a new network. In addition, a subnet associated with the network can be created in the next panel.

Admin State: UP

☑ Create Subnet

4. On the **‘Subnet’** tab, choose a **Subnet Name** and a **Network Address** (CIDR) for your subnet. You can provide a gateway IP address value or leave it empty, and it will be set automatically to the first IP of the network.



Create Network

Subnet Name: MySubnet1

Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Details" tab.

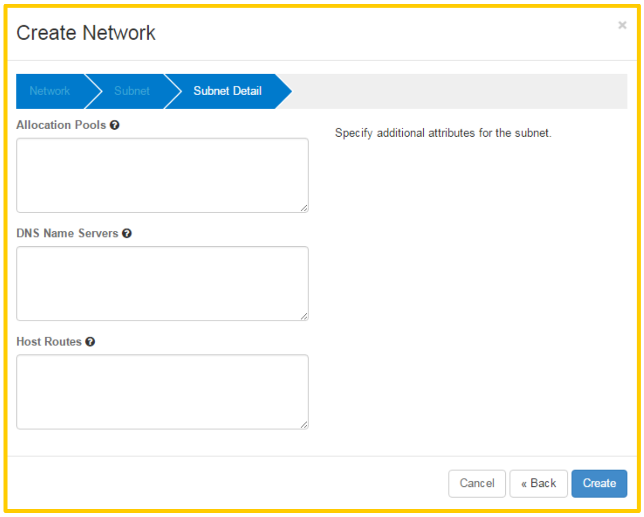
Network Address: 10.10.10.0/24

IP Version: IPv4

Gateway IP

□ Disable Gateway

5. On the **'Subnet Detail'** tab, you can optionally enter the starting and ending IP addresses you want for your DHCP allocation pool in the **Allocation Pools** field. Also you can optionally change the **DNS Name Servers** field and enter the DNS IP addresses of your choice to be assigned automatically to the VMs in the subnet. You can also optionally enter the Destination CIDR and Next Hop for your subnet in the **Host Routes** field to create host routes.


Create Network

Allocation Pools

DNS Name Servers

Host Routes

Specify additional attributes for the subnet.

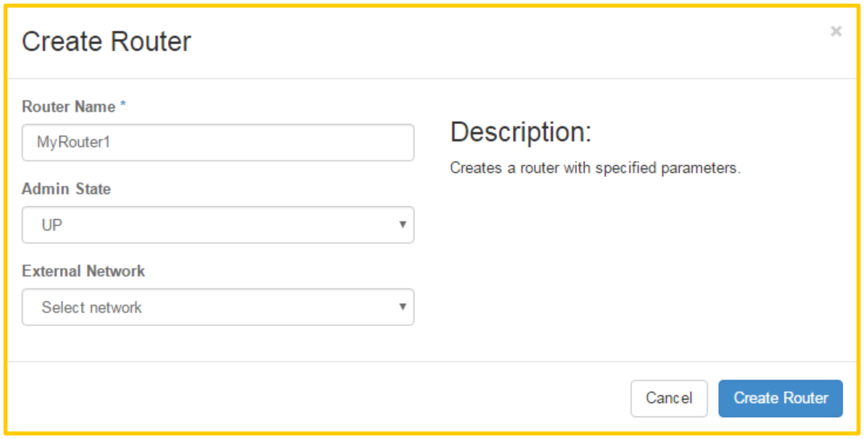
6. Click on **Create** button to keep the record. Hence, the created network will appear in the networks grid. Also you can test the new network by launching two VMs, and verify if those VMs have network connectivity among each other.

5.3. Create a Router

Apply the following steps to create a router:

1. Navigate to this path; ***Project → Network → Routers***, in order to check the current Routers at the list.
2. Click **Create Router** button in the top right corner.

3. In the **Create Router** dialog box, specify a name for the router and the *Admin State (If set to down, the router will be set as down and does not forward any packets)*, if you need to connect the router to the public internet select the *External Network* from the dropdown list and click **Create Router**.



Create Router

Router Name *
MyRouter1

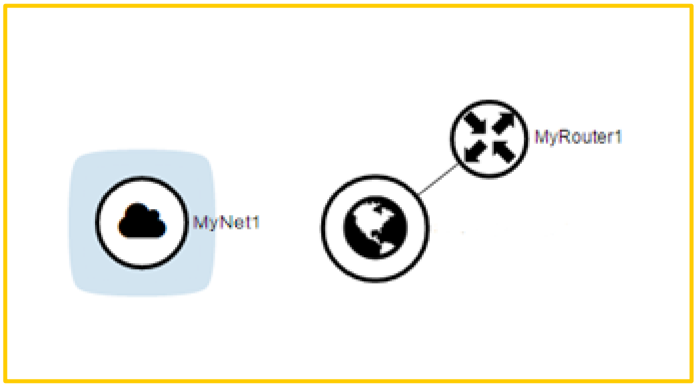
Admin State
UP

External Network
Select network

Description:
Creates a router with specified parameters.

The new router is now displayed in the **'Routers'** tab. If you did not choose to connect the router to the public Internet, you can apply the following steps:

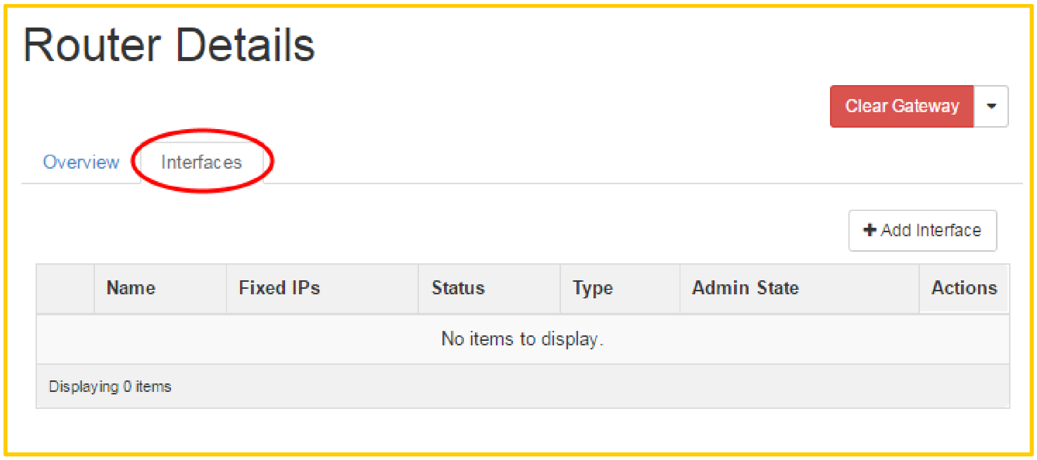
1. On the **Routers** List, click the new router's **Set Gateway** button.
2. In the **External Network** field, specify the network to which the router will connect, and then click **Set Gateway**.
3. To confirm that your router is setup properly, you can click on the **Network Topology** option and ensure that the router is connected to the external network.



Connect a Network to a Router

To connect a network to the newly created router, apply the following steps:

1. On the **'Routers'** tab, click the name of the router.
2. On the **Router Details** page, click the **'Interfaces'** tab, and then click **Add Interface**.

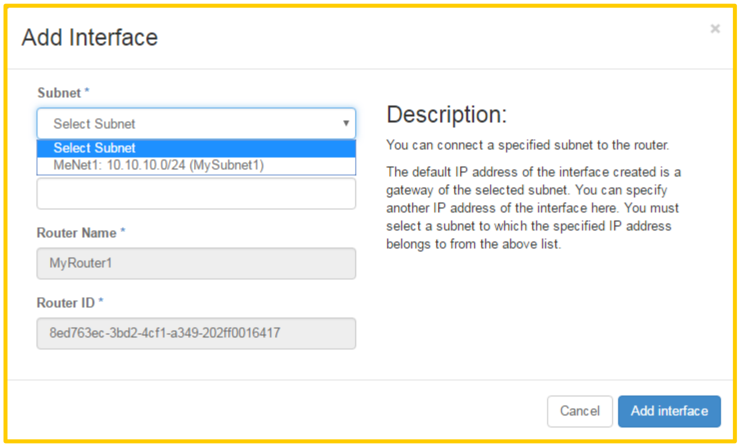


Router Details

	Name	Fixed IPs	Status	Type	Admin State	Actions
No items to display.						

Displaying 0 items

3. In the **Add Interface** dialog box, select a **Subnet**.
   - *Optional Step*: In the ***Add Interface*** dialog box, set an **IP Address** for the router interface for the selected subnet. However, to avoid any conflict, it is recommended to leave it empty.
   - If you choose not to set the **IP Address** value, then by default the first host IP address in the subnet will be used.
   - The **Router Name** and **Router ID** fields are automatically updated.
4. Click **Add Interface**.



Add Interface

Subnet *

Select Subnet

Select Subnet
MeNet1: 10.10.10.0/24 (MySubnet1)

Router Name *

MyRouter1

Router ID *

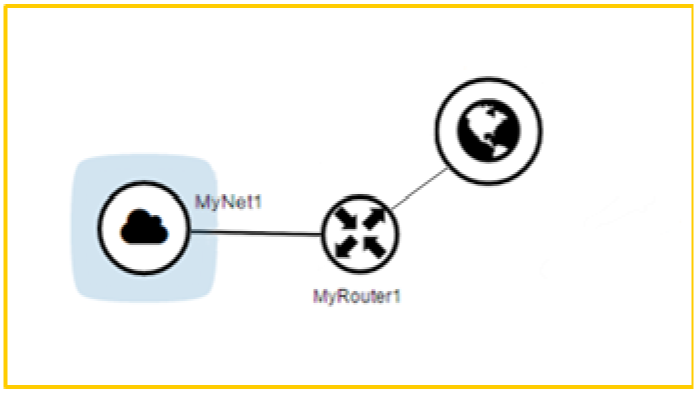
8ed763ec-3bd2-4cf1-a349-202ff0016417

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.

5. To confirm that your network is setup properly, you can click on the **Network Topology** and ensure that the network is connected to the router.



5.4. External Network Connectivity to VM (Floating IP)

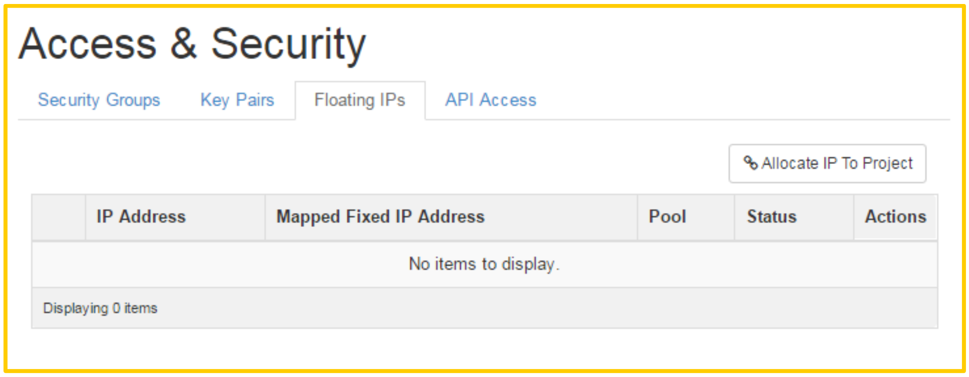
bluvalt allows you to optionally add a public IP addresses to running instances. This public IP address is called '*Floating IP*'. ***bluvalt*** Networking uses **Network Address Translation** (NAT) to assign floating IPs to virtual instances.

Important Note:

- **The VM** should be connected to a network linked to a router that is connected to the external network. Otherwise, you will not be able to associate a floating IP with it.

To assign a floating IP address to a VM, first we will need to allocate a *floating IP* to the project:

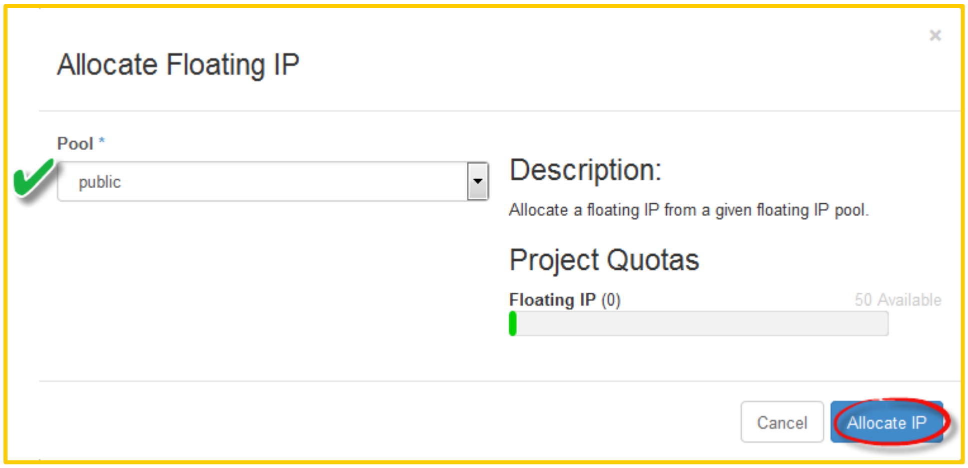
1. Navigate to this path; ***Project → Compute → Access & Security***, in order to check the **'Floating IPs'** tab.
2. Click on the **Allocate IP to Project** button at the top, as shown below.



Access & Security

	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
No items to display.					
Displaying 0 items					

3. In the **Allocate Floating IP** dialog box, click the **Allocate IP** button.



Allocate Floating IP

Pool *: public

Description:

Allocate a floating IP from a given floating IP pool.

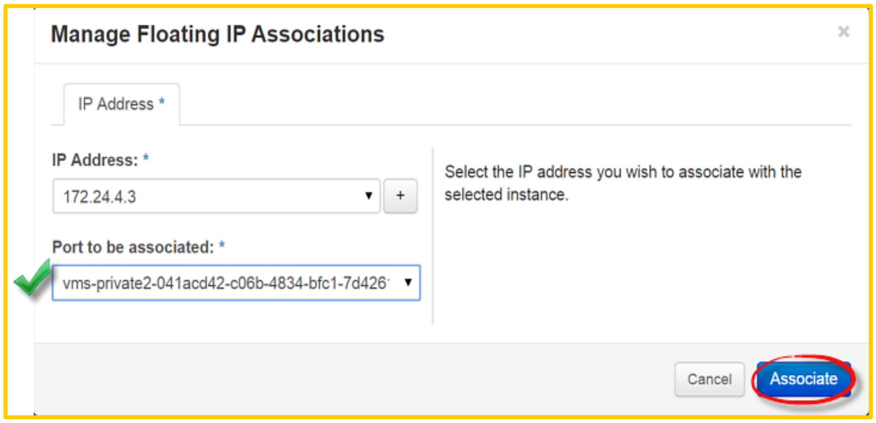
Project Quotas

Floating IP (0) 50 Available

Now that we have allocated a floating IP, we can associate it with a VM:

1. Click the **Associate** button next to the floating IP address.
2. In the **Manage Floating /IP Association** dialog box, the floating IP will be selected by default. Select the VM with which the floating IP will be associated in the *Port to be associated* list.

Click the **Associate** button.



Manage Floating IP Associations

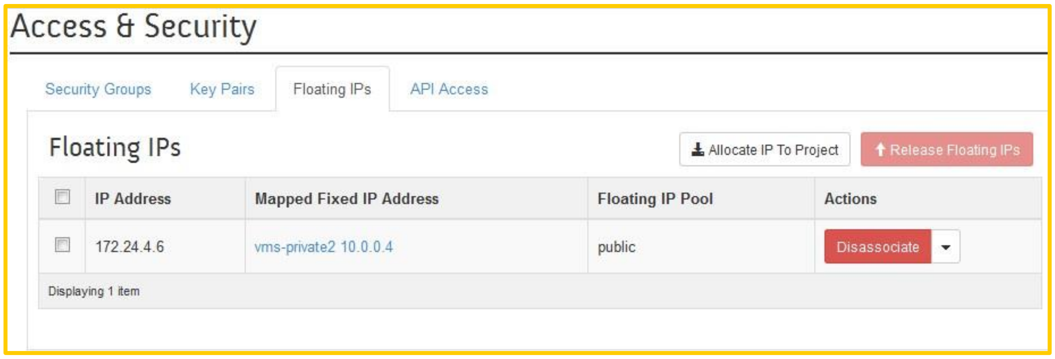
IP Address: * 172.24.4.3

Port to be associated: * vms-private2-041acd42-c06b-4834-bfc1-7d426

Select the IP address you wish to associate with the selected instance.

Notes:

- You can also associate a floating IP with a VM by choosing the **Associate Floating IP** from the instance actions menu.
- You can check the **Floating IPs** page to verify that it is updated with the floating IP associated with the VM.



Access & Security

Floating IPs

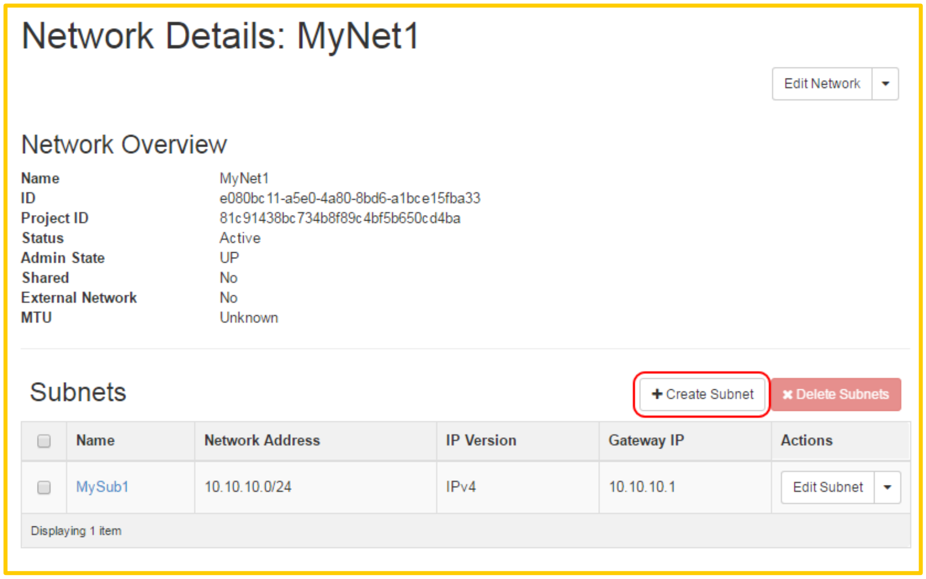
☐	IP Address	Mapped Fixed IP Address	Floating IP Pool	Actions
☐	172.24.4.6	vms-private2 10.0.0.4	public	Disassociate ▾

Displaying 1 item

5.5. Adding a Subnet to a Network

When a network is created, a subnet can be assigned upon creation. However, you can add a new subnet to a network at any time.

1. Navigate to; ***Project → Network → Networks***, and select the network to which you want to add a subnet.
2. Under the *Subnets* section, select **+ Create Subnet.**



Network Details: MyNet1

Network Overview

Name	MyNet1
ID	e080bc11-a5e0-4a80-8bd6-a1bce15fba33
Project ID	81c91438bc734b8f89c4bf5b650cd4ba
Status	Active
Admin State	UP
Shared	No
External Network	No
MTU	Unknown

Subnets

□	Name	Network Address	IP Version	Gateway IP	Actions
□	MySub1	10.10.10.0/24	IPv4	10.10.10.1	Edit Subnet ▾

Displaying 1 item

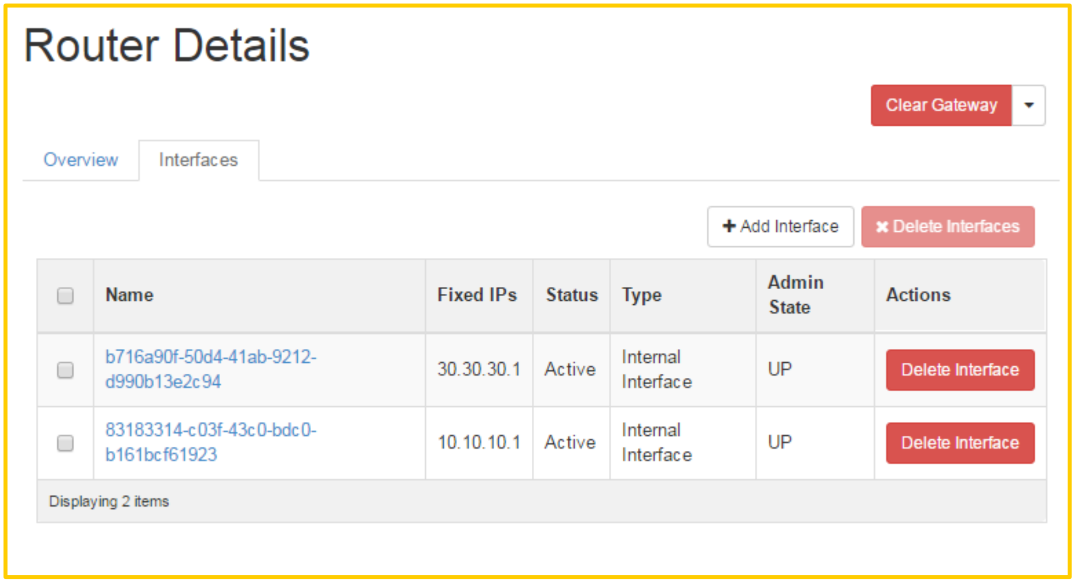
3. On the **'Subnet'** tab, choose a **Subnet Name** and a **Network Address** (CIDR) for your subnet. You can provide a gateway IP address value or leave it empty, and it will be set automatically on the first IP of the network.
4. On the **'Subnet Detail'** tab, you can optionally enter the starting and ending IP addresses you want for your DHCP allocation pool in the **Allocation Pools** field. In addition, you can optionally change the **DNS Name Servers** field and enter the DNS IP addresses of your choice to be assigned automatically to the VMs in the subnet. Moreover, you can optionally enter the Destination CIDR and Next Hop for your subnet in the **Host Routes** field to create host routes.
5. Click on **Create** button, the new subnet will be available with the Subnets section of the network

5.6. Deleting a Network

In order to delete a network, it should be free of all ports and interface connections. You should also *ensure that no instances have an attached interface on the network.*

First, you will have to delete any interface between the network and any router:

1. Navigate to ***Network → Network Topology*.**
2. Examine the network that will be deleted, and check the routers connected to it.
3. Navigate to ***Network → Routers***.
4. In the routers list, click the name of the router that is connected to the switch.
5. Navigate to the **Router Interfaces** page, and delete the interface that is connected to the switch.



Router Details

□	Name	Fixed IPs	Status	Type	Admin State	Actions
□	b716a90f-50d4-41ab-9212-d990b13e2c94	30.30.30.1	Active	Internal Interface	UP	Delete Interface
□	83183314-c03f-43c0-bdc0-b161bcf61923	10.10.10.1	Active	Internal Interface	UP	Delete Interface

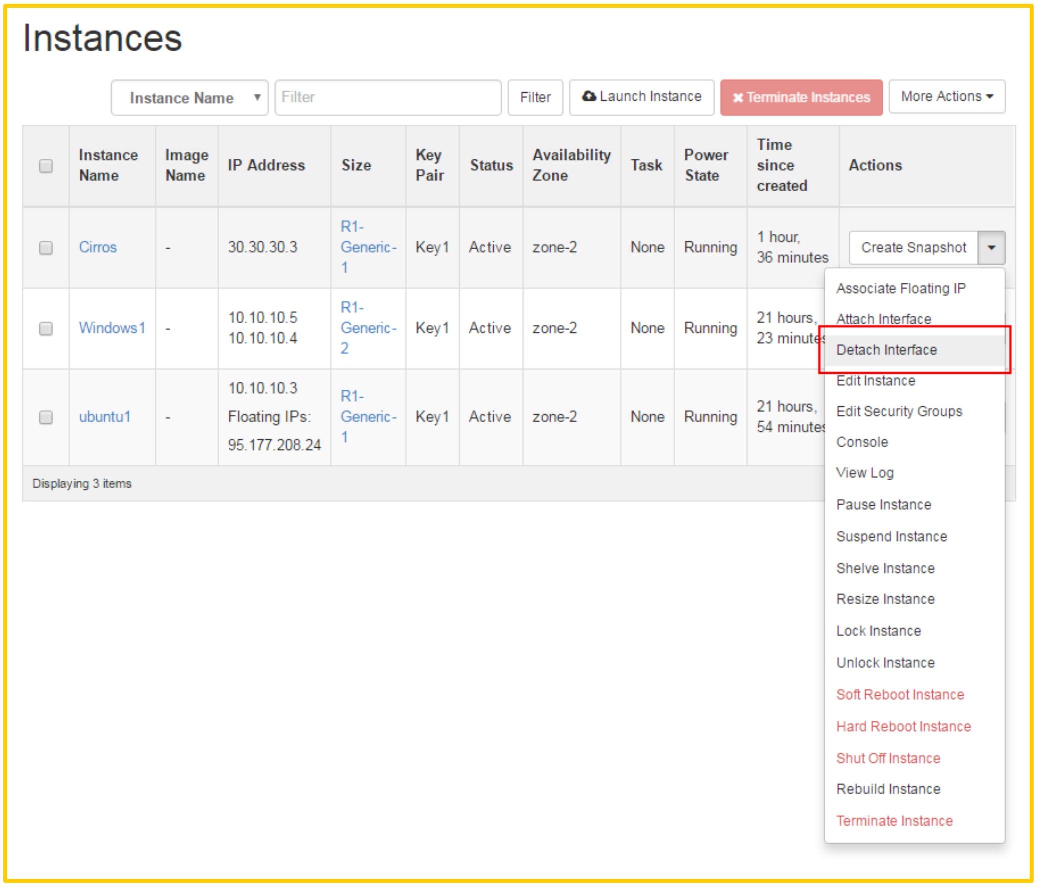
Displaying 2 items

6. Apply the same steps for the rest of the routers that are connected to the switch, if any.

   Now you will need to detach all network interfaces between the instances and the switch

7. Navigate to ***Compute → Instances*.**

In the action menu for the instance that is connected to the network select **Detach Interface** and select the port that is connected to the network.



□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	Cirros	-	30.30.30.3	R1-Generic-1	Key1	Active	zone-2	None	Running	1 hour, 36 minutes	Create Snapshot ▾
□	Windows1	-	10.10.10.5 10.10.10.4	R1-Generic-2	Key1	Active	zone-2	None	Running	21 hours, 23 minute	
□	ubuntu1	-	10.10.10.3 Floating IPs: 95.177.208.24	R1-Generic-1	Key1	Active	zone-2	None	Running	21 hours, 54 minutes	

Displaying 3 items

8. Navigate to ***Network → Networks***, to delete the network.

9. Select the network that will be deleted, and click the **Delete Network** button.

5.7. Deleting a Router

To delete a router, it should be free of all ports to networks. Also you should have no associated floating IP that is assigned to an instance that rely on the router for outside connectivity.

1. Navigate to ***Network → Network Topology***.
2. Examine the router that will be deleted, and check the networks and their connected instances.
3. Check if the instances have any associated floating IPs and *disassociate them*.
4. Click on **Routers** under **Network.**
5. In the *Routers List*, click the name of the router.
6. On the **Router Details** page, delete the interfaces that are connected to the switches.
7. Return back to *Routers List*.
8. Select the router that you want to delete, and click **Delete Router** button.

6. Object Storage

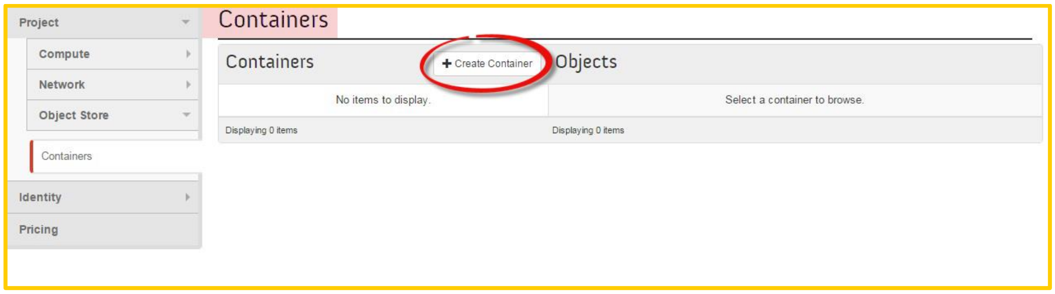
Object Storage (also known as *object-based storage*) is a storage architecture that manages data as objects. As opposed to other storage architectures like file systems which manage data as a file hierarchy and block storage which manages data as blocks within sectors and tracks, *object storage systems store files in a flat organization of containers and use unique IDs to retrieve them*. Object Storage is not directly accessed by the operating system; it is not seen as a local or remote filesystem. Instead, interaction occurs at the application level via an API

6.1. Create a Container

A **Container** is a storage compartment for your data which provides a way for you to organize your data. You can think of a container as a folder in Windows ® or a directory in UNIX ®. The primary difference between a container and these other file system concepts is that *containers cannot be nested*. You can, however, create an unlimited number of containers within your account. Data should be stored in a container, so you should have at least one container defined in your account prior to uploading data.

To create a container, apply the following steps:

1. Open the **Object Store** tab, and click **Containers** category.
2. Click **Create Container**.



No items to display.

Select a container to browse.

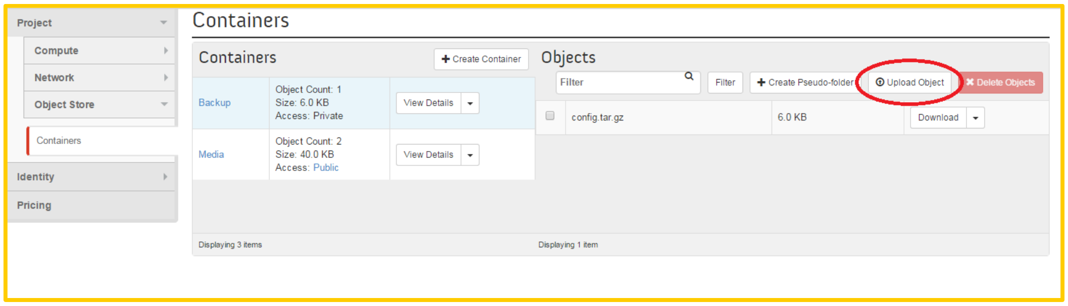
Displaying 0 items

Displaying 0 items

3. In the **Create Container** dialog box, enter a significant **Name** for the container. Choose to make the container either *Private* or *Public*. (A Public Container will allow anyone with the Public URL to gain access to your objects in the container).
4. Click **Create Container** *to keep the record*.

6.2. Upload an Object

1. Open the **'Object Store'** tab, and click **Containers** category.
2. Select the container in which you want to store your object.
3. Click **Upload Object**.



Backup	Object Count: 1 Size: 6.0 KB Access: Private	View Details
Media	Object Count: 2 Size: 40.0 KB Access: Public	View Details

Displaying 3 items

□	config.tar.gz	6.0 KB	Download

Displaying 1 item

The *Upload Object to Container: <name>* dialog box appears. ``*<name>*`` is the name of the container to which you are uploading the object.

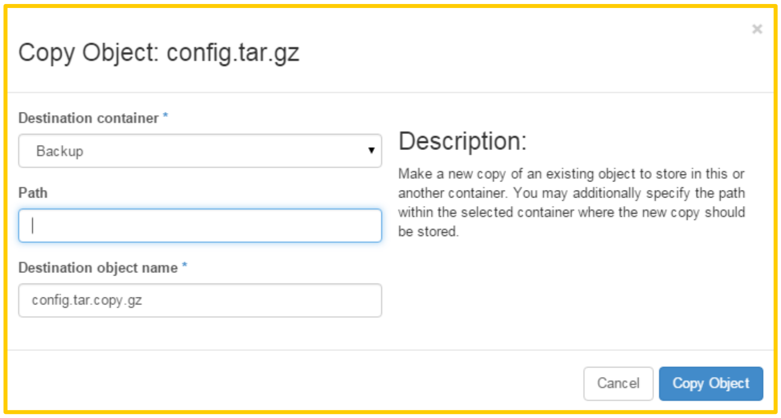
4. Enter a significant **Name** for the object.
5. **Browse** to select the file that you want to upload.
6. Click **Upload Object** *to keep the record*.

6.3. Manage an Object

To copy an object from one container to another, apply the following steps:

1. Navigate to this path; ***Project*** **→** ***Object Store*** *tab* **→** ***Containers*** *Category*, in order to select the container in which you want to store your object.

2. Click **More** and choose **Copy** from the dropdown list.
3. In the **Copy Object** launch dialog box, enter the following values:
    - *Destination Container*: Choose the destination container from the list.
    - *Path*: Specify a path in which the new copy should be stored inside the selected container.
    - *Destination Object Name*: Enter a name for the object in the new container.
4. Click **Copy** *Object*.



Copy Object: config.tar.gz

Destination container *
Backup

Path

Destination object name *
config.tar.copy.gz

Description:

Make a new copy of an existing object to store in this or another container. You may additionally specify the path within the selected container where the new copy should be stored.

You can create a new object in container without an available file and can upload the file later when it is ready. This temporary object acts as a place-holder for a new object, and enables the user to share object metadata and URL info in advance.

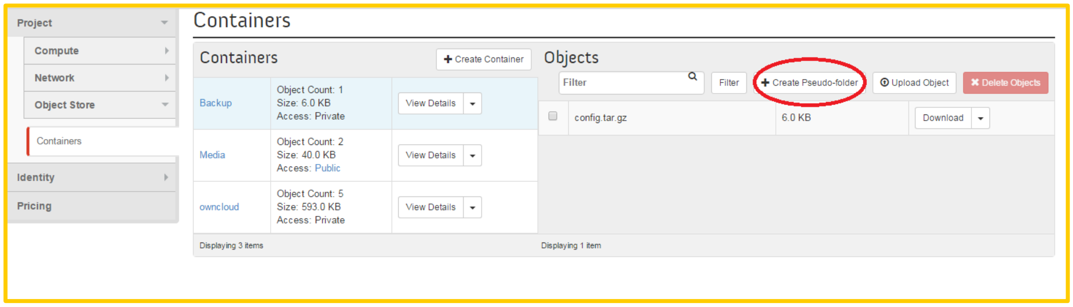
To create a metadata-only object without a file, apply the following steps:

1. Navigate to this path; ***Project*** **→** ***Object Store*** *tab* **→** ***Containers*** *Category*, in order to select the container in which you want to store your object.
2. Click **Upload Object**.
3. Enter a significant **Name** for the object.
4. Click **Update Object**.

6.4. Create a Pseudo-Folder

Pseudo-folders are similar to folders in your desktop operating system. They are virtual collections defined by a common prefix on the object's name.

1. Navigate to this path; ***Project*** **→** ***Object Store*** *tab* **→** ***Containers*** *Category*, in order to select the container in which you want to store your object.
2. Click **Create Pseudo-folder**.



Containers		
Backup	Object Count: 1 Size: 6.0 KB Access: Private	View Details
Media	Object Count: 2 Size: 40.0 KB Access: Public	View Details
owncloud	Object Count: 5 Size: 593.0 KB Access: Private	View Details

Displaying 3 items

Objects		
□ config.tar.gz	6.0 KB	Download

Displaying 1 item

3. Enter a significant name for the pseudo-folder.

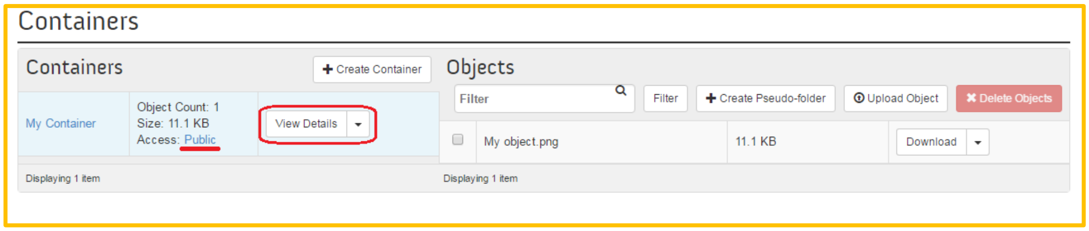
   A slash (/) character is used as the delimiter for pseudo-folders in Object Storage.

4. Click **Create**.

6.5. Sharing an Object link

Although the object storage is designed to store files on the cloud and be mainly interacted with through web APIs, you can still directly share a link of an uploaded object to be downloaded from the object storage system.

1. Make sure the container is set to *Public*, you can do that by selecting the **Make Public** option from the container action menu



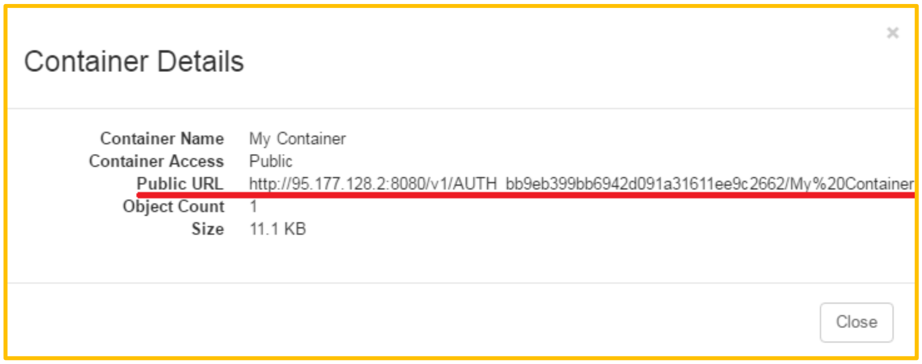
Containers		
My Container	Object Count: 1 Size: 11.1 KB Access: Public	View Details

Displaying 1 item

Objects		
□ My object.png	11.1 KB	Download

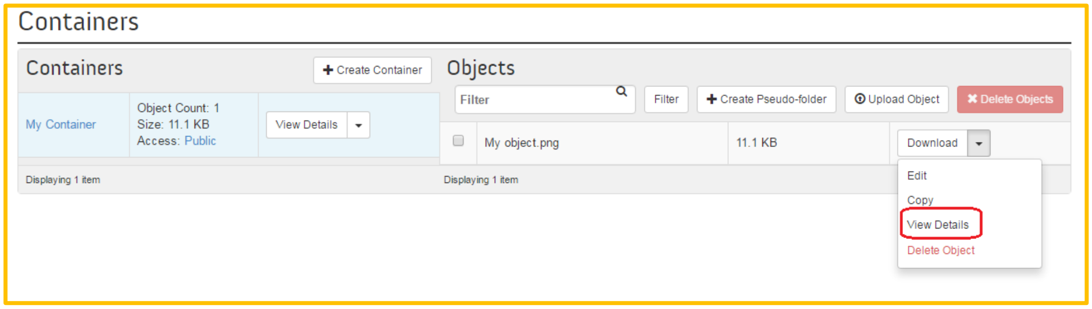
Displaying 1 item

2. From the containers action menu, click on **View Details** and then copy the containers *Public URL*



Container Name	My Container
Container Access	Public
Public URL	http://95.177.128.2:8080/v1/AUTH bb9eb399bb6942d091a31611ee9c2662/My%20Container
Object Count	1
Size	11.1 KB

3. In the action menu of the *Object* select **View Details** and then copy the name of the object



My Container
Object Count: 1
Size: 11.1 KB
Access: Public

Displaying 1 item

□	My object.png	11.1 KB

Displaying 1 item

4. Append the copied object name to the public URL of the container, and then paste the complete string in the browser URL field.

5. If the object is viewable by a browser it will open the object in the browser page, otherwise the browser will download the object as a file