SUSE Cloud

2.0

www.suse.com

September 24, 2013

End User Guide

End User Guide

List of Authors: Tanja Roth, Frank Sundermeyer

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

About This Guide

SUSE® Cloud is an open source software solution that provides the fundamental capabilities to deploy and manage a cloud infrastructure based on SUSE Linux Enterprise. SUSE Cloud is powered by OpenStack, the leading community-driven, open source cloud infrastructure project. It seamlessly manages and provisions workloads across a heterogeneous cloud environment in a secure compliant, and fully-supported manner. The product tightly integrates with other SUSE technologies and with the SUSE maintenance and support infrastructure.

This guide helps cloud users to launch and manage instances, manage networks, manage volumes, and track usage. Most of these tasks can either be achieved with the Web interface (the SUSE Cloud Dashboard) or the OpenStack command line tools.

Many chapters in this manual contain links to additional documentation resources. These include additional documentation that is available on the system as well as documentation available on the Internet.

For an overview of the documentation available for your product and the latest documentation updates, refer to `http://www.suse.com/documentation/suse_cloud20`.

1 Available Documentation

The following manuals are available for this product:

Deployment Guide (↑*Deployment Guide*)
: Gives an introduction to the SUSE® Cloud architecture and describes how to set up, deploy, and maintain the individual components.

User Guide for Administrators (↑*User Guide for Administrators*)
: Guides you through management of projects and users, images, flavors, quotas, and networks with SUSE Cloud Dashboard or the command line interface.

End User Guide (page i)
: Describes how to manage images, networks, instances, volumes, and track usage.

HTML versions of the product manuals can be found in the installed system under `/usr/share/doc/manual`. Find the latest documentation updates at http://www.suse.com/documentation where you can download the manuals for your product in multiple formats.

2 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests
: For services and support options available for your product, refer to http://www.suse.com/support/.

  To report bugs for a product component, log in to the Novell Customer Center from http://www.suse.com/support/ and select *My Support* > *Service Request*.

User Comments
: We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to http://www.suse.com/documentation/feedback.html and enter your comments there.

Mail
: For feedback on the documentation of this product, you can also send a mail to `doc-team@suse.de`. Make sure to include the document title, the product version, and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

3 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: directory names and filenames

- *`placeholder`*: replace *`placeholder`* with the actual value

- `PATH`: the environment variable PATH

- `ls`, `--help`: commands, options, and parameters

- `user`: users or groups

- Alt, Alt + F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard

- *File*, *File* > *Save As*: menu items, buttons

- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

4 About the Making of This Manual

This book is written in Novdoc, a subset of DocBook (see `http://www.docbook.org`). The XML source files were validated by `xmllint`, processed by `xsltproc`, and converted into XSL-FO using a customized version of Norman Walsh's stylesheets. The final PDF is formatted through XEP from RenderX.

1 Using SUSE Cloud Dashboard

The SUSE® Cloud Dashboard is a Web interface that enables cloud administrators and users to manage various OpenStack services. It is based on OpenStack Dashboard (also known under its codename Horizon).

After a short introduction to the Dashboard, learn how to execute key tasks such as creating images, managing networks, launching and managing instances, and how to use volumes for persistent storage.

1.1 Requirements

The following requirements need to be fulfilled to access the SUSE Cloud Dashboard:

- The cloud operator has set up SUSE Cloud.
- You have a recent Web browser that supports HTML5. It must have cookies and JavaScript enabled. For using the Dashboard's VNC client, which is based on noVNC, your browser needs to support HTML5 Canvas and HTML5 WebSockets. For more details and a list of browsers that support noVNC, refer to https://github.com/kanaka/noVNC/blob/master/README.md, and https://github.com/kanaka/noVNC/wiki/Browser-support, respectively.

1.2 SUSE Cloud Dashboard—Overview

Learn how to log in to SUSE Cloud Dashboard and get a short overview of its Web interface.

1.2.1 Logging in to the SUSE Cloud Dashboard

To access the SUSE Cloud Dashboard, ask the cloud operator for the following information:

- Hostname or (public) IP address of the SUSE Cloud Dashboard. (The Dashboard is available on the node that has the `nova_dashboard-server` role.)
- Username and password of the cloud administrator or cloud user with which you can log in to SUSE Cloud Dashboard.

1 Start a Web browser and make sure that JavaScript and cookies are enabled.

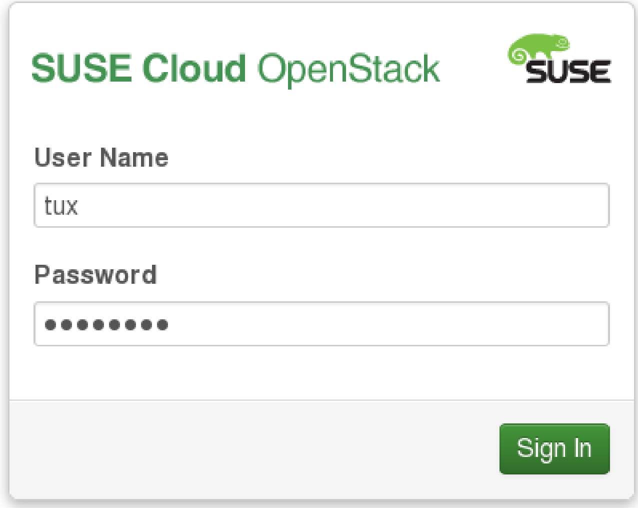
2 As a URL, enter the hostname or IP address that you got from the cloud operator.

```
https://IP_ADDRESS_OR_HOSTNAME/
```

> **NOTE: Certificate Warning**
>
> Depending on your browser and browser options, you may get a certificate warning when trying to access the URL for the first time. (In case no certificate is provided when setting up the Dashboard, SUSE Cloud uses a self-signed certificate that is not considered trustworthy by default).
>
> In this case, verify the certificate.
>
> To proceed anyway, you can add an exception in the browser to bypass the warning.

3 On the SUSE Cloud Dashboard login screen, enter the *User Name* and *Password* and click *Sign In*.

Figure 1.1: *SUSE Cloud Dashboard—Login Screen*



SUSE Cloud OpenStack

User Name

tux

Password

●●●●●●●●●

After logging in, the Dashboard's Main Screen (User's View) appears.

1.2.2 Main Screen (User's View)

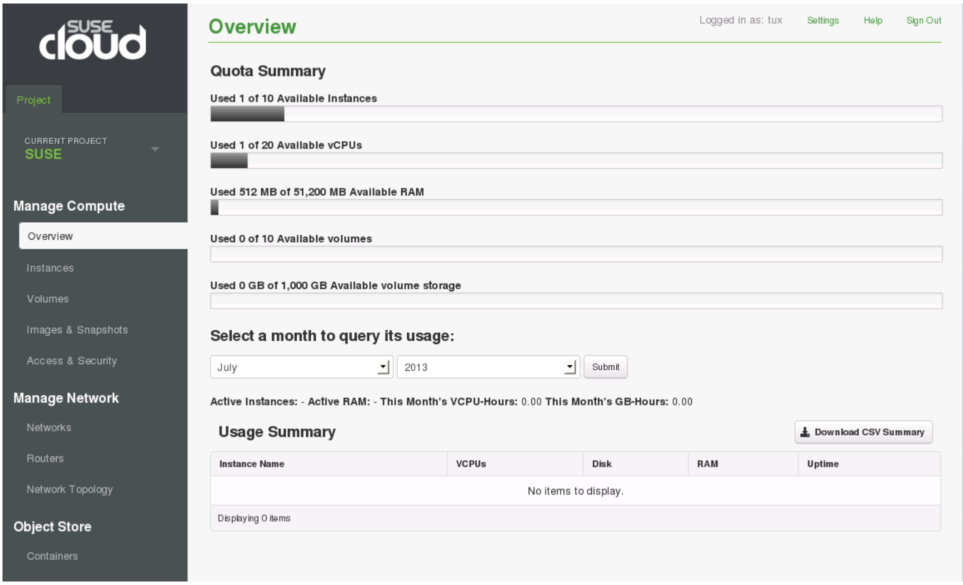
The top-level row of the main screen shows the username with which you are logged in. It also lets you access the *Settings* (regarding language and timezone), the *Help* pages, or lets you *Sign Out* of the Web interface.

NOTE: Available Functions

The visible tabs and functions in the Dashboard depend on the access permissions of the user that is logged in. They are defined by roles.

If you are logged in as a user, the main screen only shows the *Project* tab on the left navigation bar. This shows details for the projects that you are a member of.

Figure 1.2: SUSE Cloud Dashboard—Project Tab



Overview

Quota Summary

Used 1 of 10 Available Instances

Used 1 of 20 Available vCPUs

Used 512 MB of 51,200 MB Available RAM

Used 0 of 10 Available volumes

Used 0 GB of 1,000 GB Available volume storage

Select a month to query its usage:

July | 2013

Active Instances: - Active RAM: - This Month's VCPU-Hours: 0.00 This Month's GB-Hours: 0.00

Usage Summary

Instance Name	VCPUs	Disk	RAM	Uptime
No items to display.				

Displaying 0 items

Select a *Project* from the drop-down list on the left-hand side to access the following categories. They are sorted into the groups: *Manage Compute*, *Manage Network*, and *Object Store*.

Manage Compute

Overview

Shows basic reports on the project and lets you track usage.

Instances

Lists instances launched by users of the project. From here, you can launch, terminate, pause, or reboot any instances, connect to them via VNC or create a snapshot of an instance.

Volumes

Lists volumes created by users of the project. Form here, you can create volumes and attach them to instances.

Images & Snapshots

Lists images, instance snapshots and volume snapshots created by users of the project, plus any images that are publicly available, or that have been shared with the current user. From here, you can launch instances from images or from instance snapshots.

Access & Security

Lets you manage security groups and keypairs, allocate or release floating IP addresses, access the API, and download RC files.

Manage Network

Networks

Shows all networks and subnets that are available to the currently selected project. Lets you create networks and subnets for the current project.

Routers

Lets you create routers and set gateways.

Network Topology

Displays a graphical representation of the network topology. From here, you can also launch instances, and create networks or routers.

Object Store

Containers

Lets you create containers and upload objects to OpenStack Object Storage.

1.3 Managing Images

In the SUSE Cloud context, images are virtual disk images that represent the contents and structure of a storage medium or device, such as a hard drive, in a single file. Images are used as a template from which a virtual machine can be started. For starting a virtual machine, SUSE Cloud always uses a copy of the image.

Permissions to manage images are defined by the cloud operator during setup of SUSE Cloud. Image upload and management may be restricted to cloud administrators or cloud operators only.

After uploading an image to OpenStack Image, it cannot be changed any more ("golden image").

Images have both contents and metadata; the latter are also called properties. The following properties can be attached to an image in SUSE Cloud. Set them from the command line when uploading or modifying images.

Image Properties

Name (`--name`, optional)
: Specifies a name with which the image will be listed in the SUSE Cloud Dashboard and in the command line interface.

Kernel ID (optional)
: The image's kernel ID. This parameter is only needed if an external Kernel is associated with the image. The ID points to the Kernel glance image.

Ramdisk ID (optional)
: The image's ramdisk ID. This parameter is only needed if an external ramdisk is associated with the image. The ID points to the ramdisk glance image.

Container Format (`--container-format`, optional)
: Indicates if the VM image's file format contains metadata about the actual virtual machine. Set it to `bare` as the container format string is not currently used in any OpenStack components anyway. For details, refer to `http://docs.openstack.org/developer/glance/formats.html`.

Disk Format (`--disk-format`, required)
: Specifies the image's disk format. Example formats include `raw`, `qcow2`, and `ami`. For details, refer to `http://docs.openstack.org/developer/glance/formats.html`.

Public (`--is-public`, optional)
: Boolean value, default: `false`. If set to `true`, the image is publicly available.

Hypervisor Type (optional)
: If your cloud consists of both KVM and Xen nodes, specify at least the hypervisor type the image requires, otherwise it might be scheduled on an incompatible node. For example:

```
hypervisor_type=xen
```

Further example are: `kvm`, `qemu`, `xenapi`, and `powervm`.

Architecture (optional)

Specifies the architecture the image requires. For example:

```
architecture=x86_64
```

VM Mode (optional)

Specify the hypervisor ABI (application binary interface) with the `vm_mode` flag. It can take the values `pv`, `hvm`, or `xen`. Use `vm_mode=xen` for XEN PV image import, or `vm_mode=hvm` for XEN HVM image import. For KVM, the correct mode is selected automatically.

Procedure 1.1: *Adding Images*

As a user, you can only upload or modify images for the projects that you are a member of. Public images can only be uploaded or modified by cloud administrators.

Since the OpenStack Grizzly release, it is possible to upload images also via the Dashboard. Images can either be uploaded from external URLs or from a local file system. Upload of compressed image binaries (`.zip` and `.tar.gz`) is supported.

NOTE: Limitations for Image Upload via Dashboard

- For upload from an external URL, the image must be available via a valid HTTP URL that leads directly to the image binary. URLs that redirect or serve error pages will result in unusable images.

- Avoid upload of large images via Dashboard. For image sizes of several GB, it is strongly recommended to use the command-line tools. For details, refer to Section “Adding Images” (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

- Only a limited set of image properties are accessible from the Dashboard. It is therefore recommended to set and modify image properties from the command line. For details, refer to Section “Modifying Image Properties” (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

1 Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3. On the *Project* tab, select the *Images & Snapshots* category.

4. Click *Create Image*.

5. In the window that opens, enter a *Name* for the image.

6. If you have an external URL to upload the image from, enter the URL into *Image Location*. For a local image to upload, enter the path into *Image File* or click *Browse*.

7. Select the image *Format*.

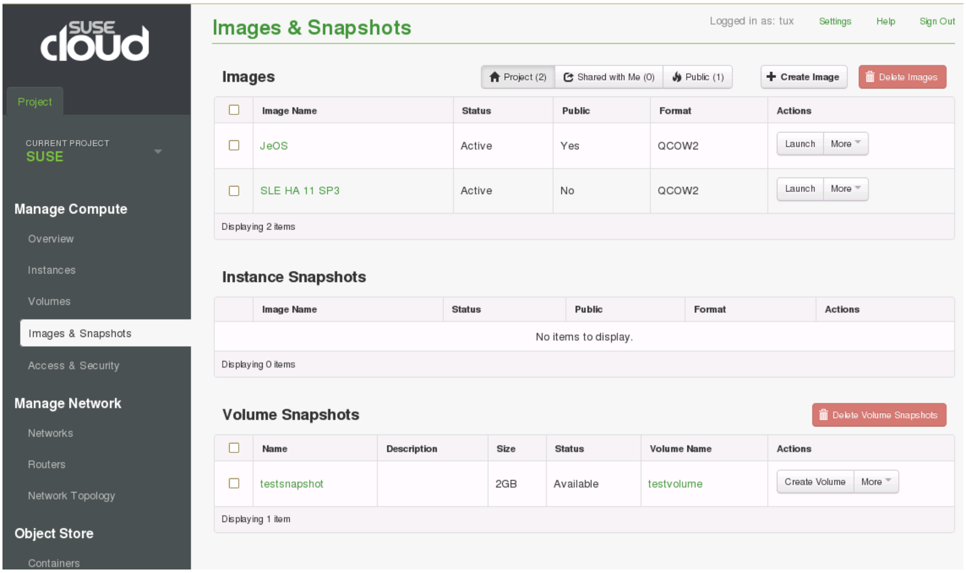
8. If you want to specify a minimum disk size or minimum RAM size that is required to boot the image, enter the values into the respective input fields. Otherwise the minimums are set to `0`.

9. Confirm your changes.

   The image is uploaded to the OpenStack Image service.

After images have been added, they appear in the SUSE Cloud Dashboard on the *Project* tab. View them in the *Images & Snapshots* category.

Figure 1.3: *SUSE Cloud Dashboard—List of Images (User's View)*



Images & Snapshots

Images

□	Image Name	Status	Public	Format	Actions
□	JeOS	Active	Yes	QCOW2	Launch More
□	SLE HA 11 SP3	Active	No	QCOW2	Launch More

Displaying 2 items

Instance Snapshots

Image Name	Status	Public	Format	Actions
No items to display.				

Displaying 0 items

Volume Snapshots

□	Name	Description	Size	Status	Volume Name	Actions
□	testsnapshot		2GB	Available	testvolume	Create Volume More

Displaying 1 item

From there, you can also edit some image properties by selecting *More* > *Edit*. As only a limited set of image properties are accessible from the Dashboard, it is preferable to set and modify image properties from the command line. For example, to make sure that an image is only launched on appropriate hypervisors, you can specify image properties referring o a certain architecture, hypervisor type or application binary interface (ABI) that the image requires. For more details, refer to Section “Adding Images” (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

If you need to delete an image, proceed as follows.

Procedure 1.2: *Deleting Images*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 On the *Project* tab, select the *Images & Snapshots* category.

4 To delete one or multiple images, activate the check boxes in front of the images that you want to delete.

5 Click *Delete Images* and confirm your choice in the pop-up that appears.

A message on the Web page shows if the action has been successful.

1.4 Managing Networks

Use the categories in the *Manage Network* group to configure virtual networks. Certain parts of managing networks are only available to cloud administrators: for example, creating or modifying external and shared networks. The deletion of shared and external networks is also limited to administrators only, as is the creation of ports. However, networks as such, subnets, and routers can be created by any user—of course, only for the respective projects that the user belongs to.

Procedure 1.3: *Creating, Modifying or Deleting Networks*

In the OpenStack Networking API, a network is an isolated L2 network segment (similar to a VLAN). It forms the basis for describing the L2 network topology in a given OpenStack Networking deployment.

As a user, you can only create or modify networks that are not-shared and not external and that are part of the project that you belong to. For any other network-related tasks, contact your cloud administrator.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Networks* category.

4 To create a new network or modify an existing one:

> **4a** Click *Create Network* or select a network and click *Edit Network*.
>
> **4b** In the window that opens, enter a name for your network or change the existing name.
>
> **4c** Leave the *Admin State* checkbox activated—unless you want the network to be marked as `down` and to not forward any packets.

4d To additionally create a subnet to be associated with the new network, refer to Procedure 1.4, “Creating or Modifying Subnets” (page 12).

4e Confirm your changes to create the new network.

5 To delete an existing network:

5a Activate the check boxes in front of the networks that you want to delete.

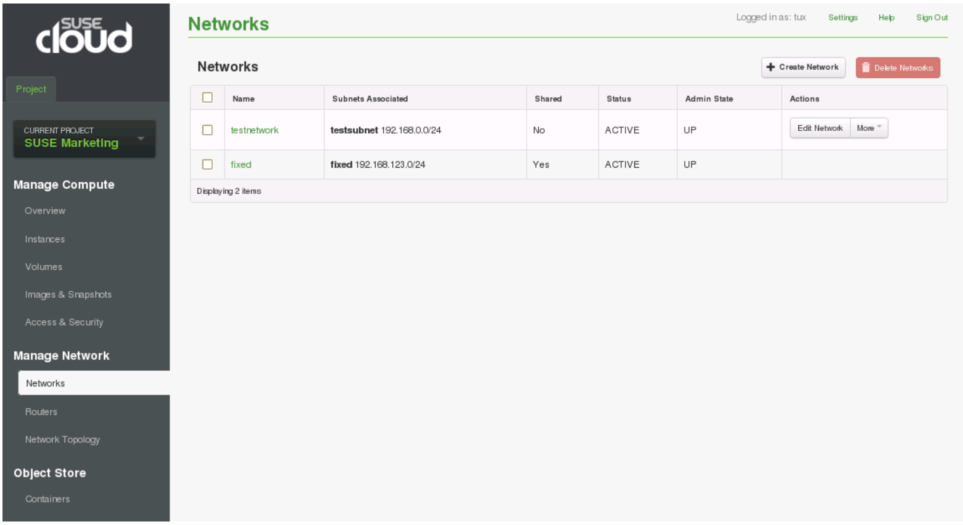
5b Click *Delete Network* and confirm your choice in the pop-up that appears.

A message on the Web page shows if the action has been successful.

NOTE: Deleting Networks

If a network cannot be deleted (although you have the respective permissions to delete it), it is because it still has ports assigned: either from a router or from a running VM instance.

Figure 1.4: *SUSE Cloud Dashboard—List of Networks (User's View)*



Networks

□	Name	Subnets Associated	Shared	Status	Admin State	Actions
□	testnetwork	**testsubnet** 192.168.0.0/24	No	ACTIVE	UP	Edit Network More
□	fixed	**fixed** 192.168.123.0/24	Yes	ACTIVE	UP	

Displaying 2 items

Procedure 1.4: *Creating or Modifying Subnets*

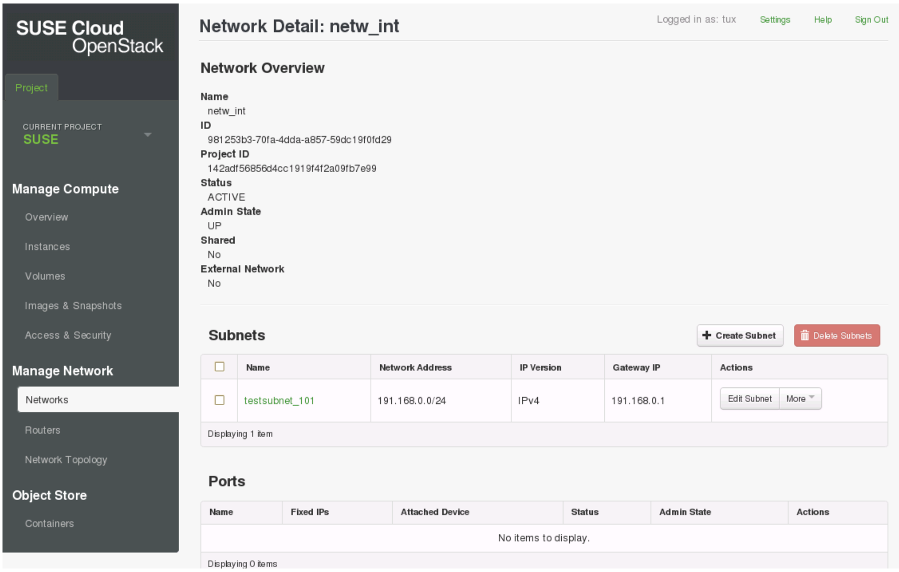
A subnet is a block of IP addresses and other network configuration (for example, a default gateway, DNS servers) that can be associated with an OpenStack Networking network. Each subnet represents an IPv4 or IPv6 address block. Multiple subnets can be associated with a network, if necessary.

By default, the first IP address of the specified network address is used as Gateway (for example, if you specified `192.168.0.0/24` as network address, `192168.0.1` is used as gateway).

1. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

2. Click the *Networks* category.

3. Select a network and click its *Network Name* to view the details for the network.



Network Detail: netw_int

Network Overview

Name
netw_int
ID
981253b3-70fa-4dda-a857-59dc19f0fd29
Project ID
142adf56856d4cc1919f4f2a09fb7e99
Status
ACTIVE
Admin State
UP
Shared
No
External Network
No

Subnets

□	Name	Network Address	IP Version	Gateway IP	Actions
□	testsubnet_101	191.168.0.0/24	IPv4	191.168.0.1	Edit Subnet More

Displaying 1 item

Ports

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

Displaying 0 items

4. Click *Create Subnet* or select a subnet and click *Edit Subnet*.

5. In the window that opens, enter a name for the subnet or change the existing name.

6 Enter or change the *Network Address*. It must be specified in CIDR format, for example:

```
192.168.0.0/24
```

7 Choose the *IP Version* to use (`IPv4` or `IPv6`).

8 If you want to use another *Gateway IP* like the default one, enter a *Gateway IP*. Otherwise, leave the input field blank.

9 If you do not want to use a gateway, activate *Disable Gateway*.

10 To specify additional attributes for the subnet, click the *Subnet Detail* tab and proceed as follows:

   10a Specify if to *Enable DHCP*.

   10b To define multiple *Allocation Pools*, enter the start and the end IP address for each pool separated by a comma and in one entry per line for each pool. For example:

   ```
   192.168.1.100,192.168.1.120
   192.158.1.100,192.158.1.120
   ```

   10c To specify *DNS Name Servers*, enter an IP address list of servers for the current subnet (one entry per line).

   10d For additional routes announced to the hosts, enter them into the *Host Routes* field as follows: One entry per line, each specifying:

   ```
   DESTINATION_CIDR,NEXTHOP
   ```

11 Confirm your changes to close the dialog.

Procedure 1.5: *Modifying Ports*

Usually, you do not need to create or modify ports manually as described below. In most cases, OpenStack will take care of that automatically (for example, when an instance is launched, a floating IP is associated, or a router interface is created).

A port is an attachment port to a L2 OpenStack Networking network. When a port is created on the network, it will be associated with a security group. If no security group is specified, it will be associated with a default security group. By default, the port will be allocated an available fixed IP address out of one of the designated subnets for each IP version. Users of the OpenStack Networking API can either choose a specific IP address from the block, or let OpenStack Networking choose the first available IP address. When the port is destroyed, the allocated addresses return to the pool of available IPs on the subnet(s).

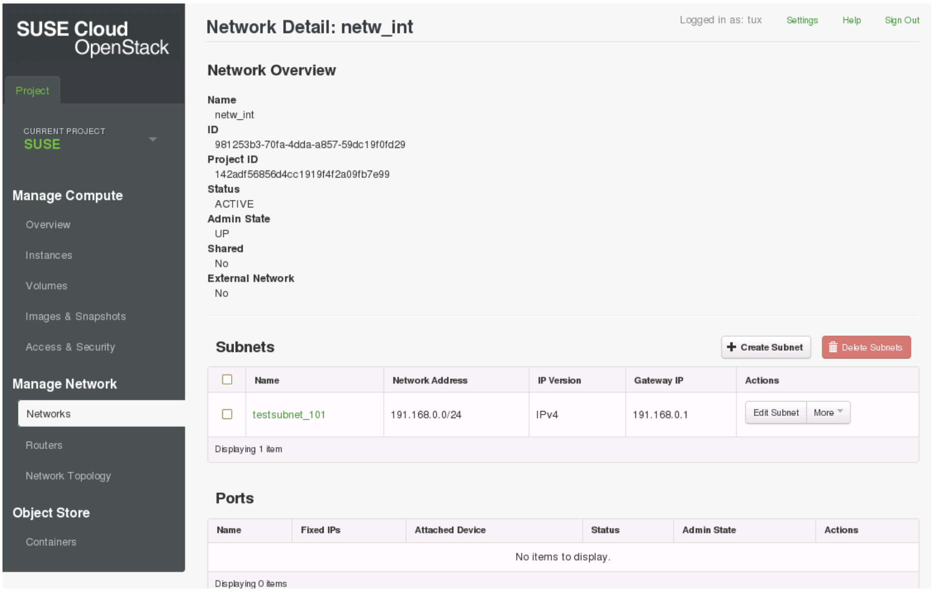
As a user, you can only modify certain port attributes. For any other changes, contact your cloud administrator.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Networks* category.

4 Select a network and click its *Network Name* to view the details for the network.



Network Detail: netw_int

Network Overview

Name
netw_int
ID
981253b3-70fa-4dda-a857-59dc19f0fd29
Project ID
142adf56856d4cc1919f4f2a09fb7e99
Status
ACTIVE
Admin State
UP
Shared
No
External Network
No

Subnets

□	Name	Network Address	IP Version	Gateway IP	Actions
□	testsubnet_101	191.168.0.0/24	IPv4	191.168.0.1	Edit Subnet More

Displaying 1 item

Ports

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

Displaying 0 items

5 Select a port to change and click *Edit Port*.

6 In the window that opens, change the *Name* for the port. Its *ID* is also displayed, but cannot be changed.

7 Leave the *Admin State* checkbox activated—unless you want the port to be marked as `down` and to not forward any traffic.

8 Confirm your changes to close the dialog.

Procedure 1.6: *Creating or Modifying Routers*

A router is used to interconnect subnets and forward traffic among them. Another feature of the router is to NAT internal traffic to external networks. A router has an interface for each subnet it is associated with. By default the IP address of such interface is the subnet's gateway IP. Also, whenever a router is associated with a subnet, a port for that router interface will be added to the subnet's network.

To add a router and to connect an external network to an internal one:

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 On the *Project* tab, select the *Routers* category.

4 Click *Create Router*.

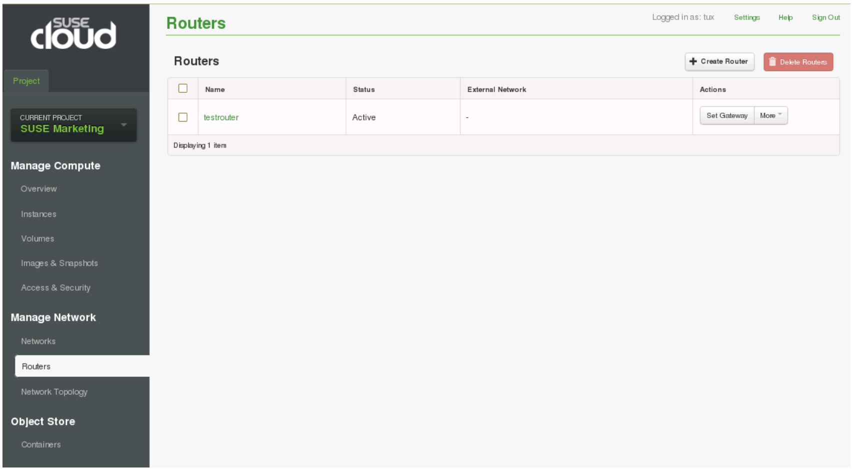
The window that opens, enter a *Router Name* and confirm your choice.

The *Routers* category shows the newly created router.

5 To connect an external network to the router:

5a Click *Set Gateway* for the newly created router.

5b In the windows that opens, select the *External Network* which to connect to the router and confirm your choice.



Routers

□	Name	Status	External Network	Actions
□	testrouter	Active	-	Set Gateway More

Displaying 1 item

6 To connect a subnet to the router:

6a Click the router's name to view the *Router Details*.

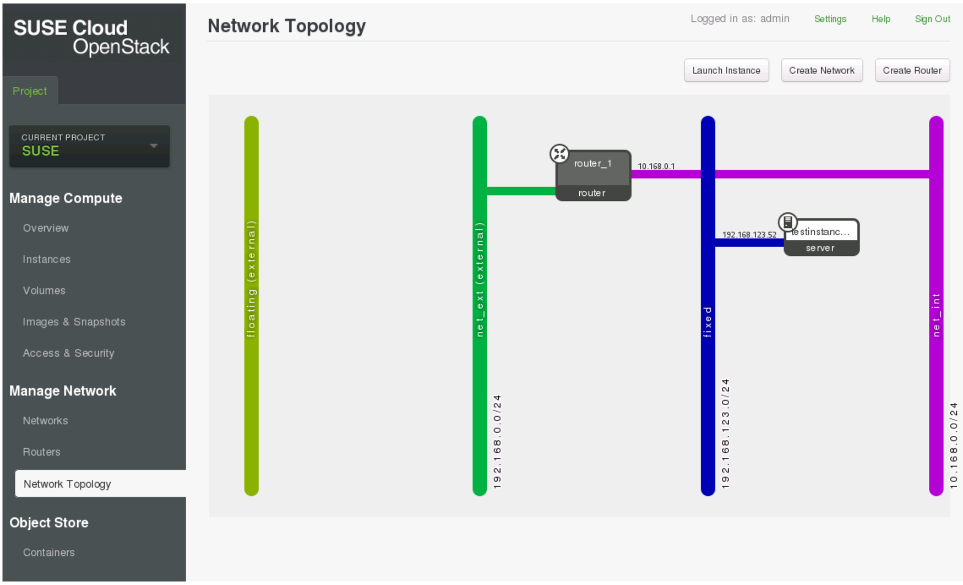
6b Click *Add Interface*.

6c In the window that opens, select the *Subnet* which to connect to the router and confirm your choice.

A message on the Web page shows if the action has been successful.

7 After having created a new router and having connected it to an external network and a gateway, you can view a graphical representation of the network topology by switching to the *Project* tab and clicking *Network Topology*.

Figure 1.5: *SUSE Cloud Dashboard—Network Topology (User's View)*



For more information and example network setups, refer to the OpenStack *Networking Administration Guide* at http://docs.openstack.org/grizzly/openstack-network/admin/content/.

1.5 Launching Instances

Instances are virtual machines that run inside the cloud. To start an instance, a virtual machine image must exist that contains the following information: which operating system to use, a username and password with which to log in to the instance, file storage etc. The cloud contains a pool of such images that have been uploaded to Image and are accessible to members of different projects.

1.5.1 Key Parameters

When starting an instance, you need to specify the following key parameters:

Flavor

In OpenStack, flavors define the compute, memory, and storage capacity of `nova` computing instances. To put it simply, a flavor is an available hardware configuration for a server. It defines the “size” of a virtual server that can be launched.

For more details and a list of default flavors available, refer to Section “Managing Flavors” (Chapter 1, *Using SUSE Cloud Dashboard*, ↑*User Guide for Administrators*).

Keypair

Keypairs are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package.

Create at least one keypair per project. If you already have generated a keypair with an external tool, you can import it into OpenStack. The keypair can be used for multiple instances belonging to that project.

For details, refer to Section 1.6.2, “Creating or Importing Keys” (page 31).

Security Group

In SUSE Cloud, security groups are used to define which incoming network traffic should be forwarded to instances. Security groups hold a set of firewall policies (security group rules).

For details, refer to Section 1.6.1, “Configuring Security Groups and Rules” (page 25).

Network

Instances can belong to one or multiple networks. By default, each instance is given a fixed IP address, belonging to the internal network.

If needed, you can assign a floating (public) IP address to a running instance and attach a block storage device (`volume`) for persistent storage. For details, refer to Section 1.6.3, “Managing IP Addresses” (page 33) and Section 1.8, “Managing Volumes” (page 42).

1.5.2 Booting From Volumes

You can start an instance directly from one of the images available in OpenStack Image or from an image that you have copied to a persistent volume before. For the preparation of the volume, refer to Procedure 1.7 (page 19). When booting an image from a volume, the procedure is basically the same as when launching an instance from an image in Image, except for some additional steps.

Procedure 1.7: *Creating and Preparing the Volume*

To be able to boot an instance from a volume, create the volume and copy an image to it:

1. Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3. Create a volume as described in Procedure 1.17, “Creating or Deleting Volumes” (page 42). Its size must be big enough to store an unzipped image.

4. Create an image with SUSE Studio or SUSE Studio Onsite. For details, refer to Section “Building Images with SUSE Studio” (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

5. Launch an instance as described in Section 1.5.3, “Launching an Instance” (page 20). Make sure to set the respective *Volume Options*, as described in Procedure 1.8: Launching an Instance, from Step 14 (page 23) to Step 14d (page 23).

6. Use `scp` to copy the image created in SUSE Studio or SUSE Studio Onsite to the instance to which you have attached the volume.

7. Log in to the instance by using SSH or the VNC console.

8. Assuming that the attached volume is mounted as `/dev/vdb`, use one of the following commands to copy the image to the attached volume:

   - For a raw image:

     ```
     cat IMAGE >/dev/vdb
     ```

     (alternatively, use `dd`)

   - For a non-raw image:

     ```
     qemu-img convert -O raw IMAGE /dev/vdb
     ```

- For a `*.tar.bz2` image:

  ```
  tar xfjO IMAGE >/dev/vdb
  ```

9. As the image comes with a predefined disk size (that might be smaller than the size of the volume it has been copied to), the image will not use the full size of the volume. To change this, adjust the partition table within the image to match the size of the volume.

10. As only *detached* volumes are available for booting, detach the volume. For details on how to do so, refer to Procedure 1.18, "Attaching Volumes to Instances" (page 44), Step 9.

11. For booting an instance from the volume, continue with Procedure 1.8, "Launching an Instance" (page 21).

1.5.3 Launching an Instance

You can start an instance directly from one of the images available in OpenStack Image. In that case, SUSE Cloud will create a local copy of the image on the respective Compute Node where the instance will be started.

NOTE: Launching Instances from a Volume

Alternatively, you can start an instance from an image that has been copied to a persistent volume. In that case, the instance will be booted from the volume (provided by nova-volume) via iSCSI.

For preparation details, refer to Procedure 1.7, "Creating and Preparing the Volume" (page 19).

To boot an instance from the volume, follow Procedure 1.8, "Launching an Instance" (page 21). Especially note the following:

- Step 4: To be able to select from which volume to boot, launch an instance from an arbitrary image. The image you select there will *not* be booted. It will be replaced by the image on the volume that you choose during the next steps.

- In case you want to boot a *Xen* image from a volume, note the following requirement: The image you launch in Step 4 needs to be of the same type (`fully virtualized` or `paravirtualized`) as the one on the volume.

Procedure 1.8: *Launching an Instance*

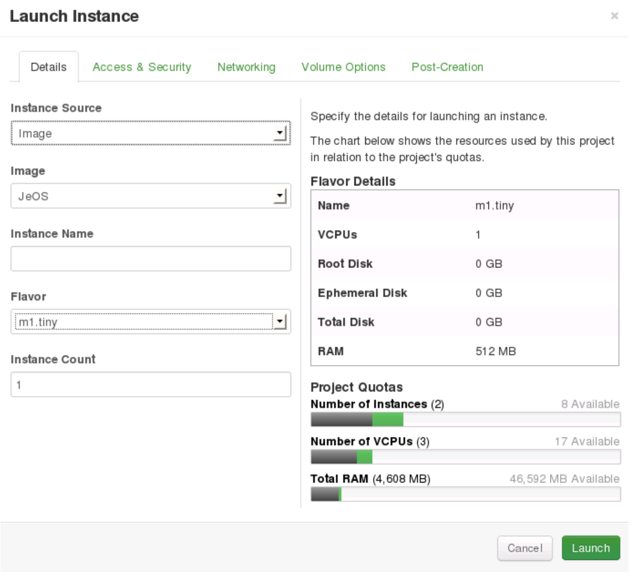
1. Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3. Click the *Images & Snapshot* category. The Dashboard shows the *Images* that have been uploaded to OpenStack Image and are available for this project.

4. Select an image and click *Launch*.

5. The *Launch Instance* dialog opens to the *Details* tab.



Instance Source: Image

Image: JeOS

Instance Name:

Flavor: m1.tiny

Instance Count: 1

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	m1.tiny
VCPUs	1
Root Disk	0 GB
Ephemeral Disk	0 GB
Total Disk	0 GB
RAM	512 MB

Project Quotas

Number of Instances (2) 8 Available

Number of VCPUs (3) 17 Available

Total RAM (4,608 MB) 46,592 MB Available

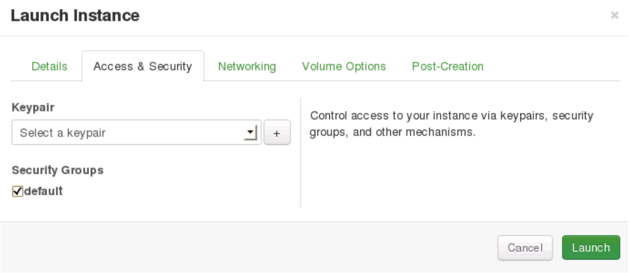
Depending on where you opened the dialog from, the values for the following drop-down lists on the *Details* tab are already preselected: *Instance Source* and *Image*. Specify the key parameters as follows.

6 Enter an *Instance Name* that will be assigned to the virtual machine.

7 From the *Flavor* drop-down list, select the “size” of the virtual machine to launch.

8 In *Instance Count*, enter the number of virtual machines to launch from this image.

9 Switch to the *Access & Security* tab.



Launch Instance ×

Details | Access & Security | Networking | Volume Options | Post-Creation

Keypair

Select a keypair +

Security Groups

☑ default

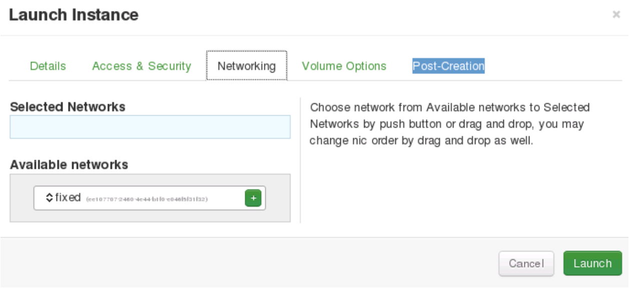
Control access to your instance via keypairs, security groups, and other mechanisms.

Cancel | Launch

10 Select a *Keypair*. For details, refer to Section 1.6.2, “Creating or Importing Keys” (page 31). In case an image uses a static `root` password or a static key set (neither is recommended), you do not need to provide a keypair on starting the instance.

11 Activate the *Security Groups* that you want to assign to the instance. Security groups are a kind of cloud firewall that define which incoming network traffic should be forwarded to instances. For details, refer to Section 1.6.1, “Configuring Security Groups and Rules” (page 25). If you have not created any specific security groups, you can only assign the instance to the default security group.

12 Switch to the *Networking* tab.



Launch Instance ×

Details | Access & Security | Networking | Volume Options | Post-Creation

Selected Networks

Available networks

fixed (ee107707-2460-4e44-b1f0-e046f5f31f32) +

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.

Cancel | Launch

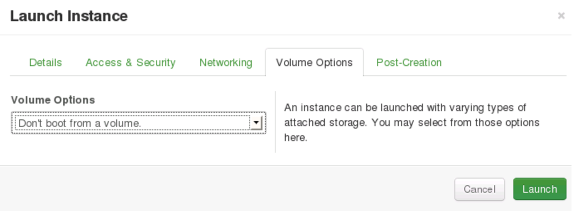
As long as you do not have configured a pool of floating IP addresses yet, the *Available Networks* list only shows the internal network with fixed IPs. For details about configuring floating IPs, refer to Procedure 1.12, “Allocating Floating (Public) IPs to a Project” (page 33).

13 To move one of the *Available Network* to the *Selected Networks* field, click the plus icon next to the network. If you have specified multiple networks for the instance, change the NIC order in the *Selected Networks* field by drag and drop.

With this step, you have set all key parameters for launching an instance.

14 If you want to launch an instance from a volume, additionally specify the following parameters. For more information on launching instances from volumes, refer to Section 1.5.2, “Booting From Volumes” (page 18).

14a Switch to the *Volume Options* tab.



Volume Options: Don't boot from a volume.

An instance can be launched with varying types of attached storage. You may select from those options here.

Its *Volume Options* drop-down list offers varying types of attached storage with which an instance can be launched.

14b To launch the instance from a volume, select `Boot from volume` or `Boot from volume snapshot (creates a new volume)`.

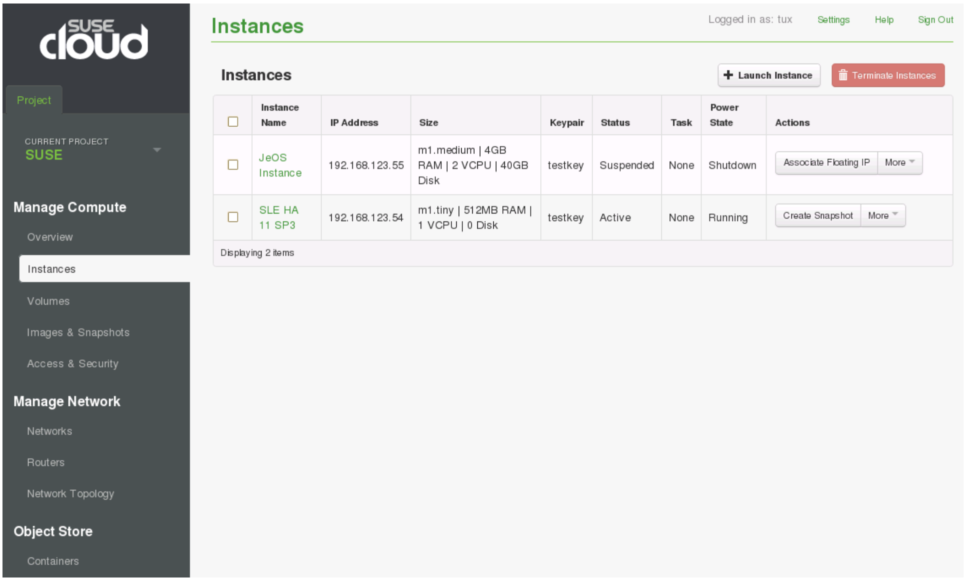
14c Select the *Volume* or *Volume Snapshot* to boot from.

14d Enter a *Device Name* (`vda` for KVM images, `xvda` for Xen images).

15 Click *Launch Instance*. The instance will be started on any of the Compute Nodes in the cloud.

After you have launched an instance, switch to the *Instances* category to view the *Instance Name*, its (private or public) *IP address*, its *Size*, the *Keypair* associated with it, the image's *Status*, its *Task*, and *Power State*.

Figure 1.6: *SUSE Cloud Dashboard—List of Launched Instances*



☐	Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
☐	JeOS Instance	192.168.123.55	m1.medium | 4GB RAM | 2 VCPU | 40GB Disk	testkey	Suspended	None	Shutdown	Associate Floating IP More
☐	SLE HA 11 SP3	192.168.123.54	m1.tiny | 512MB RAM | 1 VCPU | 0 Disk	testkey	Active	None	Running	Create Snapshot More

Displaying 2 items

If you did not provide a keypair on starting and have not touched security groups or rules so far, by default the instance can only be accessed from inside the cloud via VNC at this point. Even pinging the instance is not possible. To change this, proceed with Section 1.6, “Configuring Access to the Instances” (page 24).

1.6 Configuring Access to the Instances

Access to an instance is mainly influenced by the following parameters:

- security groups and rules
- keypairs

- IP addresses

For SSH access to an instance, you usually need to provide a keypair at launch time. The security rules need adjustment, too, since the default rules block access to SSH ports and prevent pinging an instance. To make the instance also accessible from outside the cloud, assign a floating (public) IP address.

1.6.1 Configuring Security Groups and Rules

In SUSE Cloud, security groups are used to define which incoming network traffic should be forwarded to instances. Security groups hold a set of firewall policies (security group rules).

1.6.1.1 Security Groups

When launching an instance, you need to define which security groups it should belong to. A default security group is available for each project. It allows all network traffic from other members of this group and discards traffic from other IP addresses and groups.

Multiple security groups for a project can be defined, with each group holding a different set of firewall policies. This is useful if you have groups of instances that should differ in firewall configuration (for example, front-end and back-end servers). An instance can be assigned to multiple security groups.

Procedure 1.9: *Creating or Deleting Security Groups*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 To create a new security group:

> **4a** Select the *Security Groups* tab and click *Create Security Group*.

4b In the window that opens, enter a *Name* and *Description* for the group and confirm your changes.

5 To delete one or multiple security groups:

5a Activate the check boxes in front of the groups that you want to delete.

5b Click *Delete Security Groups* and confirm your choice in the pop-up that appears.

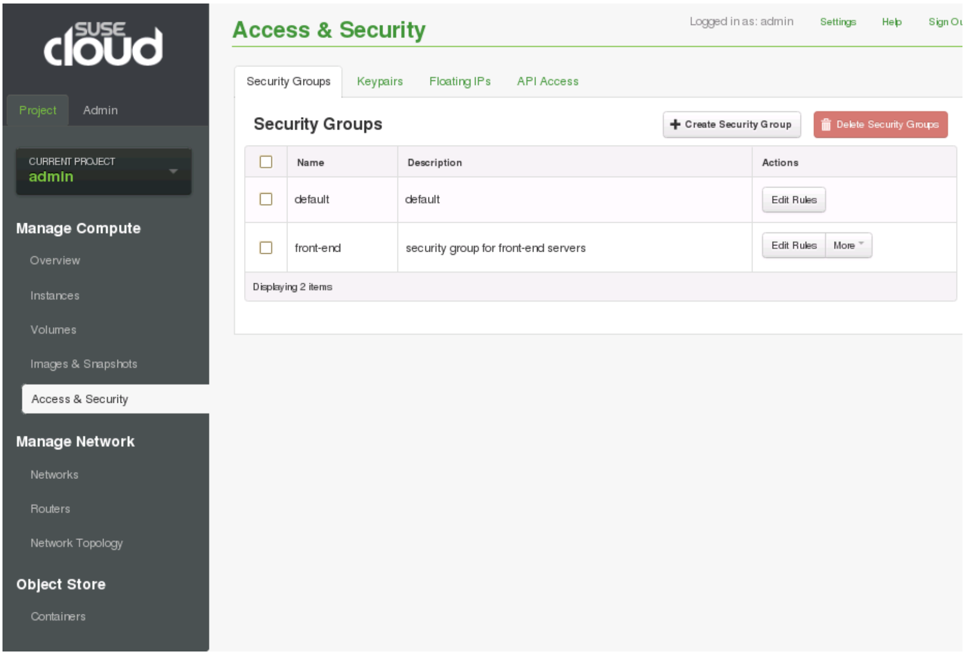
A message on the Web page shows if the action has been successful.

NOTE: Deleting Security Groups

The default security group for a project cannot be deleted.

If another group cannot be deleted, it is because it is still assigned to a running instance.

Figure 1.7: SUSE Cloud Dashboard—List of Security Groups



□	Name	Description	Actions
□	default	default	Edit Rules
□	front-end	security group for front-end servers	Edit Rules More

Displaying 2 items

1.6.1.2 Security Group Rules

You can adjust rules of the default security group as well as rules of any other security group that has been created. As soon as the rules for a group are modified, the new rules are automatically applied to all running instances belonging to that security group.

Adjust the rules in a security group to allow access to instances via different ports and protocols. This is necessary to be able to access instances via SSH, to ping them, or to allow UDP traffic (for example, for a DNS server running on an instance).

Rules in security groups are specified by the following parameters:

IP Protocol

> Protocol to which the rule will apply. Choose between TCP (for SSH), ICMP (for pings), and UDP.

Port/Port Range

For TCP or UDP, define a single port or a port range to open on the virtual machine. ICMP does not support ports. In that case, enter values that define the codes and types of ICMP traffic to be allowed.

Source of traffic

Decide whether to allow traffic to instances only from IP addresses inside the cloud (from other group members) or from *all* IP addresses. Specify either an IP address block (in CIDR notation) or a security group as source. Using a security group as source will allow any instance in that security group to access any other instance.

If no further security groups have been created, any instances are automatically assigned to the default security group (if not specified otherwise). Unless you change the rules for the default group, those instances cannot be accessed from any IP addresses outside the cloud.

Procedure 1.10: *Configuring Security Group Rules*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 On the *Security Group* tab, click *Edit Rules* for the security group you want to modify. This opens the *Edit Security Group Rules* screen that shows the existing rules for the group and lets you add or delete rules.

5 Click *Add Rule* to open a new dialog.

6 To enable SSH access to the instances:

- **6a** Set *IP Protocol* to `TCP`.

- **6b** Select to *Open* a `Port`.

- **6c** In the *Port* input field, enter the value *22*.

6d To enable access from *all* IP addresses (specified as IP subnet in CIDR notation as `0.0.0.0/0`), leave the *Source* and *CIDR* fields unchanged.

Alternatively, allow only IP addresses from other security groups to access the specified port. In that case, set *Source* to `Security Group` and select the desired *Security Group* from the drop-down list.

6e Confirm your changes to add the rule.

7 To enable pinging the instances:

7a Set *IP Protocol* to `ICMP`.

7b Set both *Type* and *Code* to the value `-1`. This enables access to all codes and all types of ICMP traffic, respectively.

A list of types and codes can be found at `http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml`, for example.

7c To enable access from *all* IP addresses (`0.0.0.0/0`), leave the *Source* and *CIDR* fields unchanged.

Alternatively, allow only IP addresses from other security groups to access the specified port. In that case, set *Source* to `Security Group` and select the desired *Security Group* from the drop-down list.

7d Confirm your changes to add the rule.

8 To allow access via UDP port (for example, for a DNS server running on a VM):

8a Set *IP Protocol* to `UDP`.

8b Select to *Open* a `Port`.

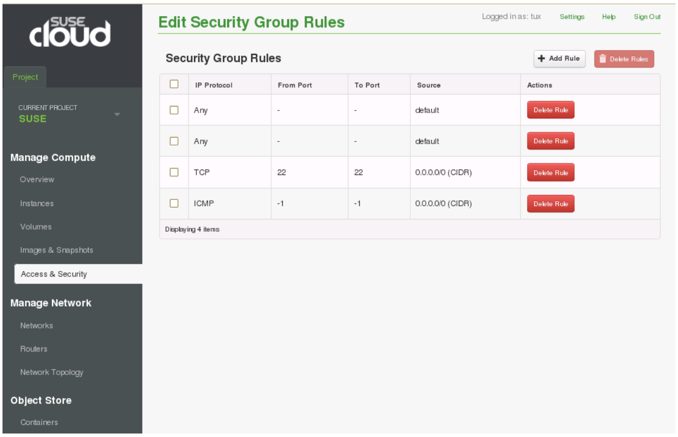
8c In the *Port* input field, enter the value *53*.

8d To allow access from *all* IP addresses (`0.0.0.0/0`), leave the *Source* and *CIDR* fields unchanged.

Alternatively, allow only IP addresses from other security groups to access the specified port. In that case, set *Source* to `Security Group` and select the desired *Security Group* from the drop-down list.

8e Confirm your changes to add the rule.

Figure 1.8: *SUSE Cloud Dashboard—List of Security Group Rules*



Edit Security Group Rules

Security Group Rules

□	IP Protocol	From Port	To Port	Source	Actions
□	Any	-	-	default	Delete Rule
□	Any	-	-	default	Delete Rule
□	TCP	22	22	0.0.0.0/0 (CIDR)	Delete Rule
□	ICMP	-1	-1	0.0.0.0/0 (CIDR)	Delete Rule

Displaying 4 items

Procedure 1.11: *Deleting Security Group Rules*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 On the *Security Group* tab, select the security group to modify and click *Edit Rules*. This opens the *Edit Security Group Rules* screen that shows the existing rules for the group and lets you add or delete rules.

5 Select the rule or rules to remove.

6 Click *Delete Rules* and confirm your choice.

1.6.2 Creating or Importing Keys

Keypairs are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package.

Create at least one keypair per project. If you already have generated a keypair with an external tool, you can import it into OpenStack. The keypair can be used for multiple instances belonging to that project.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 Switch to the *Keypairs* tab.

5 To import a keypair that you have generated with an external tool:

> **5a** Click *Import Keypair*.
>
> **5b** In the window that opens, enter a name for the keypair and copy the public key into the respective input field.
>
> **5c** Confirm your choice.

6 To create a new keypair:

> **6a** Click *Create Keypair*.
>
> **6b** In the window that opens, enter a name for the keypair and confirm your choice.
>
> OpenStack generates a keypair and provides the private key for download as a `*.pem` file.

6c Save the `*.pem` file locally and change its permissions so that only you can read and write to the file:

```
chmod 600 MY_PRIV_KEY.pem
```

The public key of the keypair is registered at the Compute database. The Dashboard lists the keypair in the *Access & Security* category as shown in Figure 1.9, "SUSE Cloud Dashboard—Keypairs" (page 32).

NOTE: Access from a Windows Machine

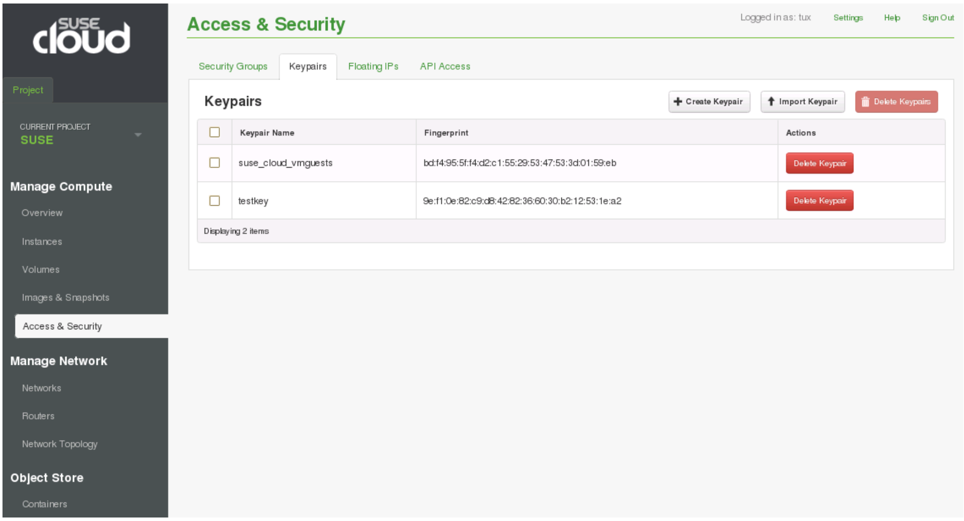
If you want to access an instance via SSH from a Windows* machine, you need to convert the key from `*.pem` format to `*.ppk` format after downloading it. To do so on a Windows workstation:

1. Open the `puttykeygen.exe` file.

2. Click *Load* to open the `*.pem` file.

3. *Save as* public-private key (`*.ppk`).

Figure 1.9: *SUSE Cloud Dashboard—Keypairs*



Access & Security

Keypairs

□	Keypair Name	Fingerprint	Actions
□	suse_cloud_vmguests	bd:f4:95:5f:f4:d2:c1:55:29:53:47:53:3d:01:59:eb	Delete Keypair
□	testkey	9e:f1:0e:82:c9:d8:42:82:36:60:30:b2:12:53:1e:a2	Delete Keypair

Displaying 2 items

1.6.3 Managing IP Addresses

Each instance can have two types of IP addresses: private (fixed) IP addresses and public (floating) ones. Private IP addresses are used for communication between instances, and public ones are used for communication with the outside world. When an instance is launched, it is automatically assigned private IP addresses in the networks to which it is assigned. The private IP stays the same until the instance is explicitly terminated. (Rebooting the instance does not have an effect on the private IP addresses.)

A floating IP is an IP address that can be dynamically added to a virtual instance. In OpenStack Networking, cloud operators can configure pools of floating IP addresses. These pools are represented as external networks. Floating IP are allocated from a subnet that is associated with the external network. You can allocate a certain number of floating IPs to a project—the maximum number of floating IP addresses per project is defined by the quota. From this set, you can then add a floating IP address to an instance of the project.

Procedure 1.12: *Allocating Floating (Public) IPs to a Project*

Before you can assign a floating IP address to an instance, you first need to allocate floating IPs to a project.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 To allocate a floating IP address to the current project:

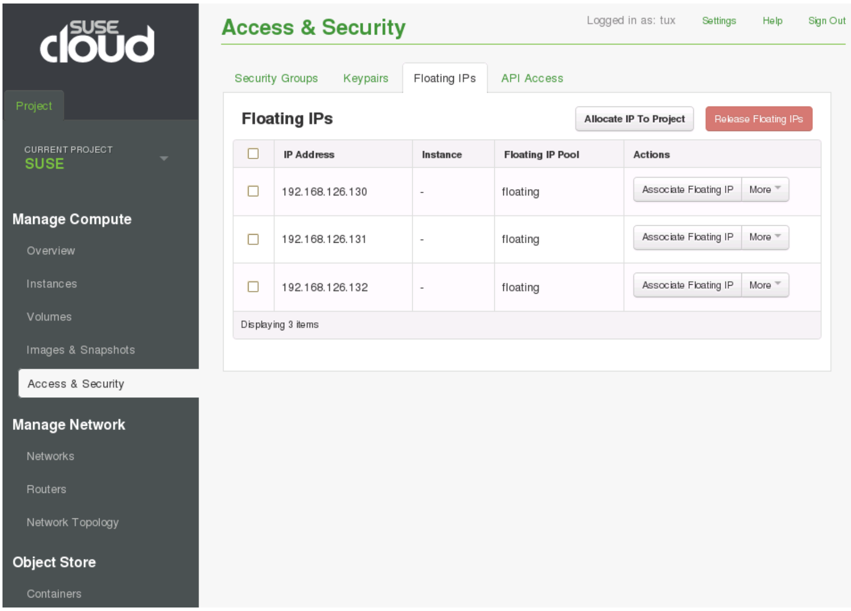
- **4a** Switch to the *Floating IPs* tab.

- **4b** Click *Allocate IP to Project.*

- **4c** In the window that opens, select a *Pool* out of which to take the IP address.

- **4d** Click *Allocate IP*.

The Dashboard shows the allocated IP addresses for the project on the *Floating IPs* tab.



□	IP Address	Instance	Floating IP Pool	Actions
□	192.168.126.130	-	floating	Associate Floating IP More
□	192.168.126.131	-	floating	Associate Floating IP More
□	192.168.126.132	-	floating	Associate Floating IP More

Displaying 3 items

5 To release one or multiple floating IP addresses from a project:

5a Activate the check boxes in front of the IP addresses that you want to release.

5b Click *Release Floating IPs*. The IP addresses are put back into the pool of IP addresses that are available for all projects. If an IP address is currently assigned to a running instance, it will automatically be disassociated from the instance.

Procedure 1.13: *Assigning Floating (Public) IP Addresses to Instances*

After floating IP addresses have been allocated to the current project, you can assign them to running instances. One floating IP address can be assigned to only one instance at a time.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Access & Security* category. It shows the following tabs: *Security Groups*, *Keypairs*, *Floating IPs*, and *API Access*.

4 Switch to the *Floating IPs* tab.

5 To assign an IP to an instance:

5a Click *Associate Floating IP* button for the IP that you want to use for the instance.

In the window that opens, the current IP address is preselected in the *IP Address* drop-down list. The drop-down list *Port to be associated* shows a list of running instances with the respective fixed IP address.

5b Keep or change the floating IP address that is selected in *IP Address*.

5c From *Port to be associated* select the instance to associate with the floating IP address.

5d Confirm your choices.

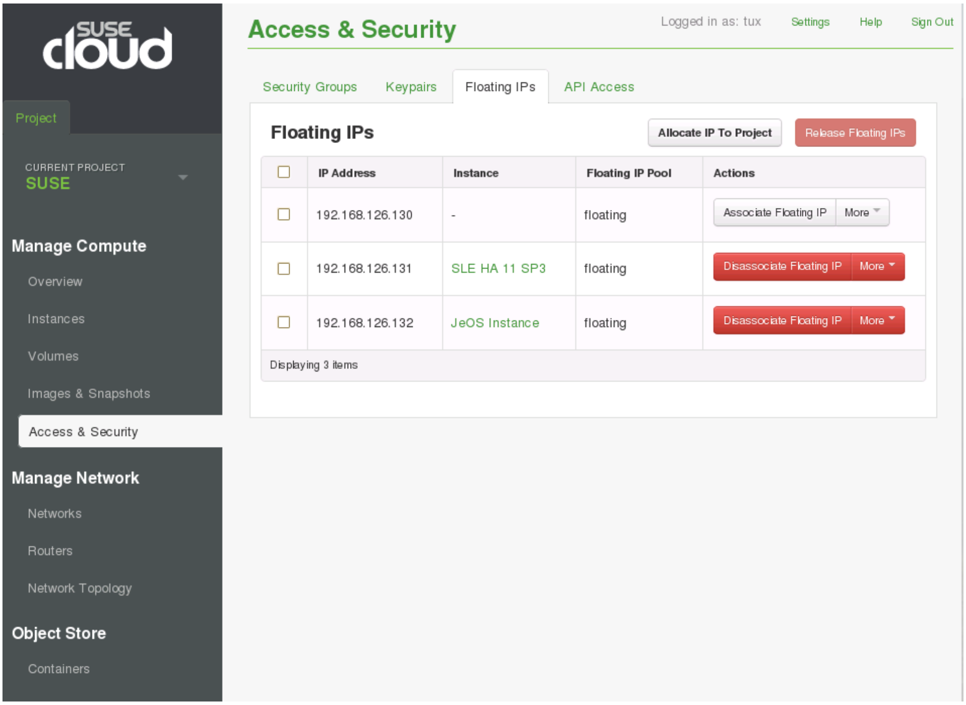
In the *Access & Security* category, the *Floating IPs* tab shows the name of the instance with which the IP has been associated. The instance is now publicly available under the respective floating IPs address (provided you have also configured the security group rules for the instance accordingly). For details, refer to Section 1.6.1, “Configuring Security Groups and Rules” (page 25).

6 To remove a floating IP address from an instance:

6a On the *Floating IPs* tab, select the IP address to remove.

6b Click *Disassociate IP* and confirm your change.

Figure 1.10: *SUSE Cloud Dashboard—Floating IPs Assigned to Instances*



Access & Security

Floating IPs

□	IP Address	Instance	Floating IP Pool	Actions
□	192.168.126.130	-	floating	Associate Floating IP More
□	192.168.126.131	SLE HA 11 SP3	floating	Disassociate Floating IP More
□	192.168.126.132	JeOS Instance	floating	Disassociate Floating IP More

Displaying 3 items

1.7 Managing Instances

The following are typical tasks for managing instances:

- Viewing logs
- Accessing instances from remote
- Creating instance snapshots to preserve a certain disk state of an instance
- Rebooting or terminating instances
- Pausing or suspending instances
- Tracking instance usage

1.7.1 Viewing Instance Logs

1. Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3. Click the *Instances* category.

4. Select the instance and from the *More* drop-down list, select *View Log*.

   Alternatively, click the instance's name and switch to the *Log* tab that opens.

   The Dashboard shows the output of the instance's serial console.

To make use of this feature, the respective image must have set the serial console correctly in GRUB. To do so, append the following to the Kernel line in `/boot/grub/grub.conf`:

```
console=tty0 console=ttyS0,115200
```

1.7.2 Accessing Instances from Remote

The Dashboard's built-in VNC client lets you access instances at any time.

Procedure 1.14: *Accessing an Instance via VNC*

1. Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

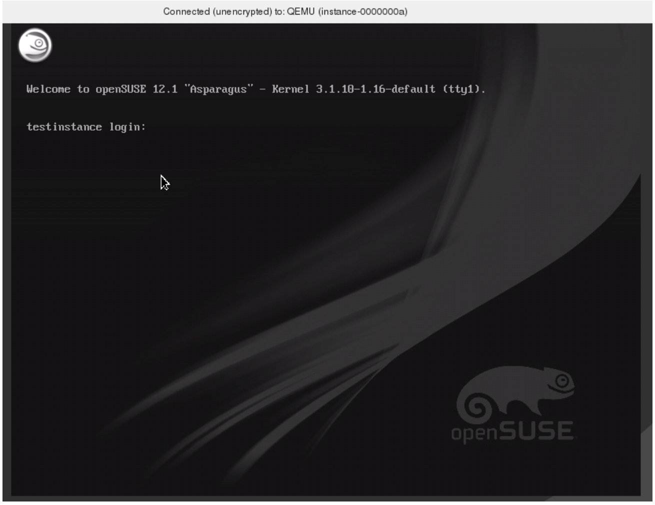
3. Click the *Instances* category.

4. Select the instance to access and from the *More* drop-down list, select *Console*.

   Alternatively, click the instance's name and switch to the *Console* tab that opens.

5. When establishing the first connection, you might be prompted by your browser to trust a certificate before you can see the VNC screen.

6 To display a larger VNC screen, use the link *Click here to show only console*.



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

```
Welcome to openSUSE 12.1 "Asparagus" - Kernel 3.1.10-1.16-default (tty1).

testinstance login:
```

7 To leave the large VNC screen, use the back button of the browser.

To access an instance via SSH, the following requirements need to be fulfilled:

- `sshd` must be running inside the virtual machine.
- Port `22` must be open in the virtual machine's firewall.
- The security group which the instance is assigned to, must be configured to allow SSH access.
- To enable SSH access from outside the cloud, a floating IP address must be assigned to the instance.
- You must know the private or public IP address of the instance.

1.7.3 Using Instance Snapshots

SUSE Cloud's snapshot mechanism allows you to create new images from running instances. This is convenient for upgrading base images or taking a published image and customizing for local use. Instance snapshots preserve the disk state of a running instance. Ephemeral disks are not included in any snapshots. You can launch a new instance from a snapshot.

As snapshots capture the state of the file system, but not the state of the memory, consider the following guidelines before taking snapshots: http://docs.openstack.org/grizzly/openstack-ops/content/snapsnots.html#consistent_snapshots.

NOTE: Removing Former Keypairs

After an instance has once been launched with a certain keypair (that is injected into the respective image by `cloud-init` during start), the keypair will stick to that instance. Unless you remove the former keypair from the instance it is impossible to restart the instance with a different keypair than the original one. This is also true for snapshots taken from an instance.

Run the following command within the instance to remove the former data (for example, to prepare an instance for a snapshot):

```
rm -rf /var/lib/cloud/instances/*
```

After the clean-up, a new keypair can be injected into the instance.

Procedure 1.15: *Creating Instance Snapshots*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

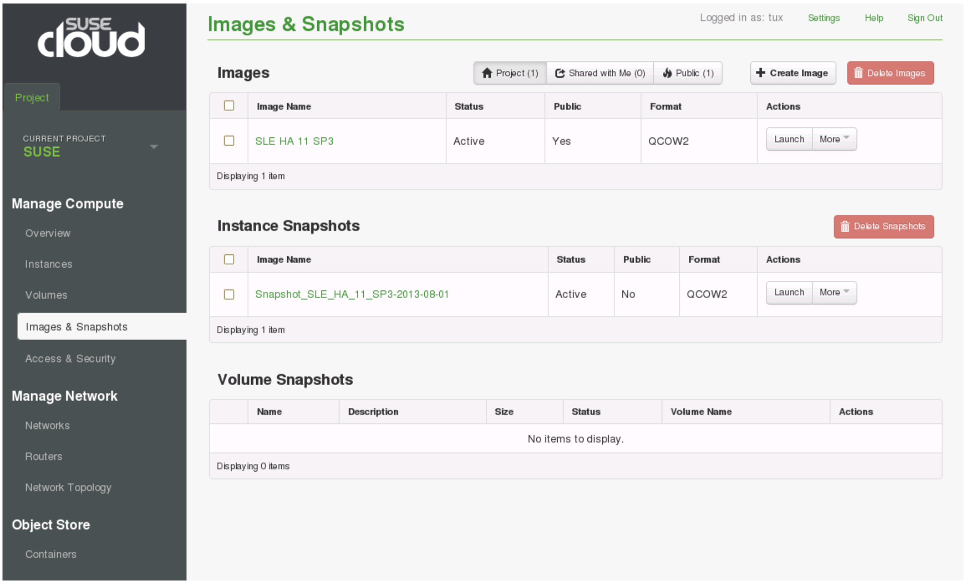
3 Click the *Instances* category.

4 Click *Create Snapshot* next to the instance that you want to snapshot.

5 In the window that opens, enter a name for the snapshot and confirm your changes. The Dashboard shows the new snapshot in the *Images & Snapshot* category below the heading *Instance Snapshot*. Technically, an instance snapshot is an image, too and is automatically uploaded to Glance. The only difference compared to an image that has been directly uploaded to OpenStack Image is the following: images created by snapshots have additional properties in the Glance database.

6 To launch a new instance from the snapshot, select the snapshot and click *Launch*. Proceed with launching an instance as described in Procedure 1.8, "Launching an Instance" (page 21).

Figure 1.11: *SUSE Cloud Dashboard—List of Images & Snapshots*



Images & Snapshots

Images

□	Image Name	Status	Public	Format	Actions
□	SLE HA 11 SP3	Active	Yes	QCOW2	Launch More

Displaying 1 item

Instance Snapshots

□	Image Name	Status	Public	Format	Actions
□	Snapshot_SLE_HA_11_SP3-2013-08-01	Active	No	QCOW2	Launch More

Displaying 1 item

Volume Snapshots

	Name	Description	Size	Status	Volume Name	Actions
			No items to display.			

Displaying 0 items

1.7.4 Controlling Instance State (Pause, Suspend, Reboot, Terminate)

For maintenance reasons, you can pause or suspend images—provided they are running on KVM or Xen. Pausing or suspending avoids the consequences that come with terminating an instance. Instances running on VMware or Hyper-V can only be suspended, but not paused.

If you pause an instance, the content of the virtual machine is stored to memory (RAM) and the image is kept running in a "frozen" state. When suspending an instance, the content of the virtual machine is stored to disk, and memory and VCPUs are freed.

WARNING: Terminating Instances: Risk of Data Loss

Terminating an instance has the following consequences:

- All data on the image's root disk and ephemeral disks are destroyed. To prevent that, use volumes and attach them to an instance for persistent storage.
- If a floating IP address was assigned to that instance, the IP address is disassociated from that image. However, it is still available in the pool of allocated IP addresses for the current project.

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Instances* category.

4 Select the instance that you want to put out of the running state. From the *More* drop-down list, select the respective action.

1.7.5 Tracking Usage

Use the Dashboard's *Overview* category to track usage of instances per project. This lets you track costs per month by showing metrics like number of VCPUs, disks, RAM, and uptime of all your instances. The *Overview* category also shows a *Quota Summary*.

Procedure 1.16: *Querying Instance Usage*

1 Log in to SUSE Cloud Dashboard.

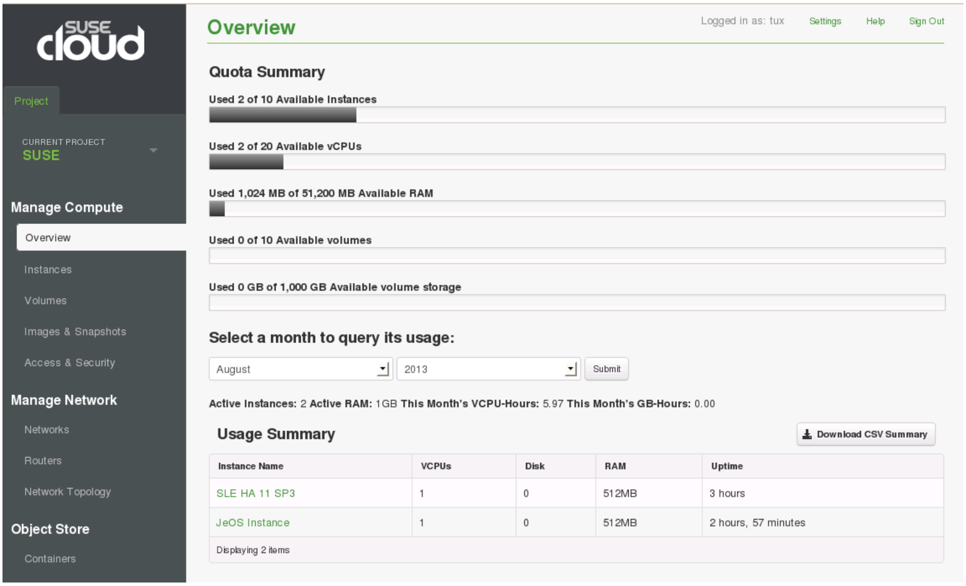
2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Overview* category. It lets you track usage of instances per project and costs per month. The upper part of the page also shows a *Quota Summary*.

4 In *Select a month to query its usage*, select a month and a year to query and click *Submit*.

The *Usage Summary* table shows the results of your query and displays metrics like the number of VCPUs, disks, RAM, and uptime of all your instances. It is also possible to download a CVS summary.

Figure 1.12: *SUSE Cloud Dashboard—Usage Overview*



Overview

Quota Summary

Used 2 of 10 Available Instances

Used 2 of 20 Available vCPUs

Used 1,024 MB of 51,200 MB Available RAM

Used 0 of 10 Available volumes

Used 0 GB of 1,000 GB Available volume storage

Select a month to query its usage:

Active Instances: 2 **Active RAM:** 1GB **This Month's VCPU-Hours:** 5.97 **This Month's GB-Hours:** 0.00

Usage Summary

Instance Name	VCPUs	Disk	RAM	Uptime
SLE HA 11 SP3	1	0	512MB	3 hours
JeOS Instance	1	0	512MB	2 hours, 57 minutes

Displaying 2 items

1.8 Managing Volumes

Volumes are block storage devices that can be attached to instances. They allow for persistent storage as they can be attached to a running instance (or detached and attached to another instance at any time). In contrast to the instance's root disk, the data of volumes is not destroyed when the instance is terminated.

Procedure 1.17: *Creating or Deleting Volumes*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Volumes* category.

4 To create a volume:

 4a Click *Create Volume*.

 4b In the window that opens, enter a *Volume Name* to assign to a volume, a description (optional), and define the size in GB. If a volume type has been defined by a cloud administrator (specifying the capabilities of the storage back-end drivers to be used), select the desired *Type* from the drop-down list.

 4c Confirm your changes.

 The Dashboard shows the volume in the *Volumes* category.

5 To delete one or multiple volumes:

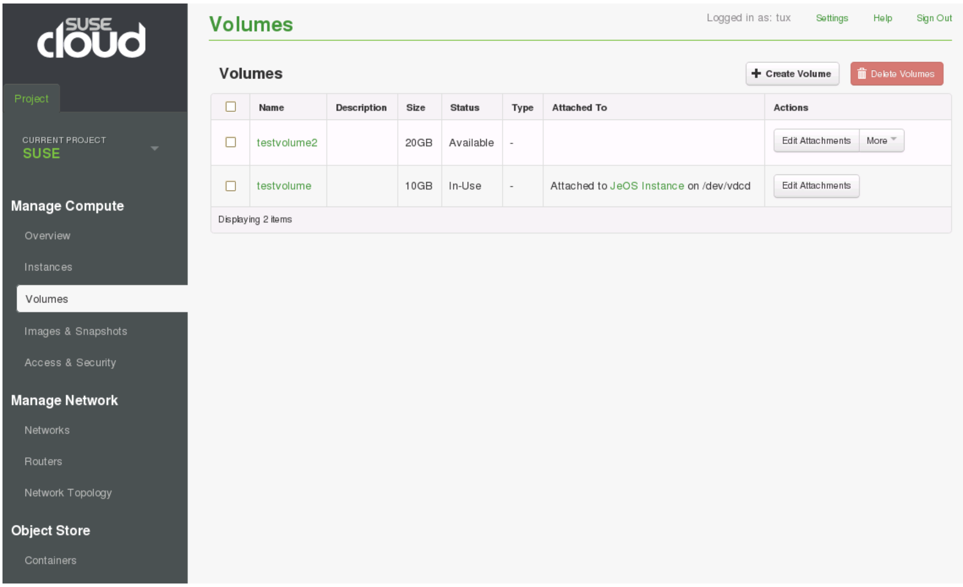
 5a Activate the check boxes in front of the volumes that you want to delete.

 5b Click *Delete Volumes* and confirm your choice in the pop-up that appears. Volumes that are attached to an instance cannot be deleted.

 A message on the Web page shows if the action has been successful.

After having created one or multiple volumes, you can attach them to instances. A volume can only be attached to one instance at a time. View the *Status* of a volume in the *Volumes* category of the Dashboard: the *Attached To* column tells you if the volume is still available or already in use.

Figure 1.13: *SUSE Cloud Dashboard—List of Volumes*



Volumes

☐	Name	Description	Size	Status	Type	Attached To	Actions
☐	testvolume2		20GB	Available	-		Edit Attachments More
☐	testvolume		10GB	In-Use	-	Attached to JeOS Instance on /dev/vdcd	Edit Attachments

Displaying 2 items

Procedure 1.18: *Attaching Volumes to Instances*

1 Log in to SUSE Cloud Dashboard.

2 If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3 Click the *Volumes* category.

4 Select the volume to add to an instance and click *Edit Attachments*.

5 In the window that opens, select an instance to attach the volume to.

6 Enter a *Device Name* under which the volume should be accessible on the virtual machine.

NOTE: KVM and Device Naming

If you are using KVM as hypervisor, the resulting device name in the guest may be different than the one specified in the Dashboard or in the `nova volume-attach` command. The device will be attached to the guest over a virtual PCI bus. When the guest sees a new device on the PCI bus, it picks the next available name, which is `/dev/vdc` in most cases. For `m1.tiny` flavors the volume is typically attached as `/dev/vdb` because it does not have a second local disk, unlike the other flavors.

7 Confirm your changes. The Dashboard shows the instance to which the volume has been attached and the volume's device name.

8 Now you can log in to the instance, mount the disk, format it, and use it.

NOTE: Instances Running Older Kernel Versions

If one of the instances runs a Kernel version that was shipped prior to SUSE Linux Enterprise 11 SP2, you need to reboot the instance to make the device appear.

For instances running the SUSE Linux Enterprise Server SP2 Kernel (or later), it is enough to load the `acpiphp` module manually:

```
modprobe acpiphp
```

9 To detach a volume from an instance:

9a Select the volume and click *Edit Attachments*.

9b In the window that opens, click *Detach Volume* and confirm your changes.

A message on the Web page shows if the action has been successful.

Procedure 1.19: *Creating Volume Snapshots*

Volume snapshots preserve the state of an attached block storage device. They are read-only point in time copies of a volume. You can also create a new volume based upon a volume snapshot.

NOTE: Consistency of Snapshots

To prevent data corruption in snapshots, only take snapshots from volumes that are not currently written to.

While the `--force` option of the `cinder` command line tool allows creation of snapshots while they are attached to an instance, the Dashboard currently does not allow you to do so.

1. Log in to SUSE Cloud Dashboard.

2. If you are a member of multiple projects, select a *Current Project* at the top of the left navigation bar.

3. Click the *Volumes* category.

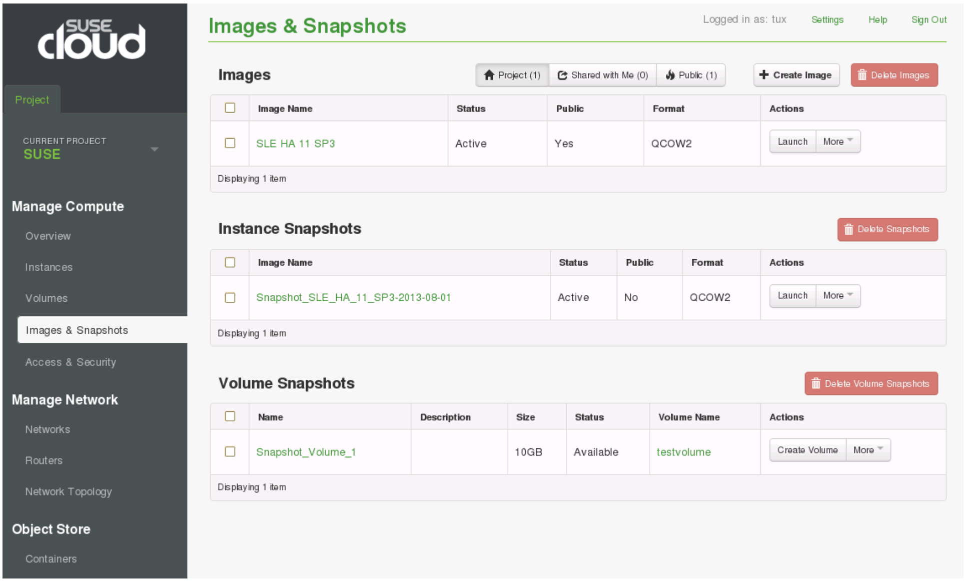
4. Check if the volume of which to create a snapshot is *Attached To* an instance. If yes, click *Edit Attachments*, click *Detach Volume* in the window that opens and confirm your choice.

5. From the *More* drop-down list, select *Create Snapshot*.

6. In the window that opens, enter a *Snapshot Name* and a *Description*.

7. Confirm your changes. The Dashboard shows the new *Volume Snapshot* in the *Images & Snapshots* category.

Figure 1.14: *SUSE Cloud Dashboard—List of Images & Snapshots*



Images & Snapshots

Images

☐	Image Name	Status	Public	Format	Actions
☐	SLE HA 11 SP3	Active	Yes	QCOW2	Launch More

Displaying 1 item

Instance Snapshots

☐	Image Name	Status	Public	Format	Actions
☐	Snapshot_SLE_HA_11_SP3-2013-08-01	Active	No	QCOW2	Launch More

Displaying 1 item

Volume Snapshots

☐	Name	Description	Size	Status	Volume Name	Actions
☐	Snapshot_Volume_1		10GB	Available	testvolume	Create Volume More

Displaying 1 item

Using OpenStack Command Line Clients 2

The OpenStack project provides a variety of command line tools with which you can manage the services within your cloud and automate tasks by using scripts. Each of the core OpenStack components has its own command line tool.

2.1 OpenStack Commands—Overview

Use the OpenStack command-line clients to run simple commands that make API calls. You can use these commands in scripts to automate tasks. Each OpenStack service has its own command-line client, that runs on Linux* or Mac OS* X systems. For some client commands, you can specify a debug parameter to show the underlying API request for the command.

The following command line clients are available for the respective services' APIs.

`ceilometer`
: For managing OpenStack Metering. Provided by the `python-ceilometerclient` package. In SUSE Cloud 2.0 it is included as a technology preview.

`cinder`
: For managing the block storage. Provided by the `python-cinderclient` package.

`glance`
: For managing images. Provided by the `python-glanceclient` package.

`heat`
: For managing OpenStack Orchestration. Provided by the `python-heatclient` package. In SUSE Cloud 2.0 it is included as a technology preview.

`keystone`
: For managing users, projects, roles, endpoints, and credentials. Provided by the `python-keystoneclient` package.

`neutron`
: For configuring networks for guest servers. Provided by the `python-quantumclient` package.

`nova`
: For managing images, instances, and flavors. Provided by the `python-novaclient` package.

`swift`
: For gathering statistics, listing items, updating metadata, and managing files stored by the Object Storage service. Provides access to a swift installation for ad hoc processing. Provided by the `python-swiftclient` package.

Most of them have tab completion.

Help and detailed information about the individual commands and their arguments are available with

```
COMMAND help
```

For help on subcommands, use

```
COMMAND help SUBCOMMAND
```

For example: `glance help` or `glance help image-create`

2.2 OpenStack RC File

To set the necessary environment variables for the OpenStack command line tools, you need to download and source an environment file, the OpenStack RC file. It contains the credentials used by OpenStack services like Compute, Image, and Identity services for a specific project. You can download it from the SUSE Cloud Dashboard, either as user `admin` or as any other user. The filename of the RC file contains the name of the project for which the credentials are valid: *`PROJECTNAME`*`-openstack.rc`.

The RC file requires running Bash or Bourne shell (sh) as shell environment.

Procedure 2.1: *Downloading the OpenStack RC File*

1. Log in to the SUSE Cloud Dashboard.

2. On the *Project* tab, select the project for which you want to download the OpenStack RC file from the *Current Project* drop-down list.

3. Select the *Access & Security* category and switch to the *API Access* tab.

4. Click *Download OpenStack RC File* and save the file.

5. Copy the RC file to the machine on which you want to execute OpenStack commands (for example, uploading an image with the `glance` command).

6. On any shell that you want to execute OpenStack commands from, source the RC file for the respective project, for example:

   ```
   source PROJECTNAME-openstack.rc
   ```

   You will be prompted for an OpenStack password.

7. Enter the OpenStack password of the user who downloaded the *`PROJECTNAME`*`-openstack.rc` file.

   With sourcing the file and entering the password, environment variables are set for that shell. They allow the commands to communicate to the OpenStack services running in the cloud.

2.3 Managing Images

In the SUSE Cloud context, images are virtual disk images that represent the contents and structure of a storage medium or device, such as a hard drive, in a single file. Images are used as a template from which a virtual machine can be started. For starting a virtual machine, SUSE Cloud always uses a copy of the image.

Permissions to manage images are defined by the cloud operator during setup of SUSE Cloud. Image upload and management may be restricted to cloud administrators or cloud operators only.

After uploading an image to OpenStack Image, it cannot be changed any more ("golden image").

Images are owned by projects and can be `private` (accessible to members of the particular project only) or `public` (accessible to members of all projects). Private images can also be explicitly shared with other projects, so that members of those projects can access the images, too. Any image uploaded to OpenStack Image will get an `owner` attribute. By default, ownership is set to the primary project of the user that uploads the image.

Images can either be uploaded to SUSE Cloud with the `glance` command line tool or with the SUSE Cloud Dashboard. As the Dashboard comes with some limitations with regards to image upload and modification of properties, it is recommended to use the `glance` command line tools for comprehensive image management.

For detailed information, refer to Section "Managing Images" (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

2.4 Managing Networks

Networks can now be managed with the `neutron` commands, provided by the `python-quantumclient` package. Alternatively, you can still use the known `nova` commands, provided by the `python-novaclient` package.

2.4.1 Creating or Modifying Networks and Subnets

In the OpenStack Networking API, a network is an isolated L2 network segment (similar to a VLAN). It forms the basis for describing the L2 network topology in a given OpenStack Networking deployment.

A subnet is a block of IP addresses and other network configuration (for example, a default gateway, DNS servers) that can be associated with an OpenStack Networking network. Each subnet represents an IPv4 or IPv6 address block. Multiple subnets can be associated with a network, if necessary.

By default, the first IP address of the specified network address is used as Gateway (for example, if you specified `192.168.0.0/24` as network address, `192168.0.1` is used as gateway).

Certain parts of managing networks are only available to cloud administrators: for example, creating or modifying external and shared networks. The deletion of shared and external networks is also limited to administrators only, as is the creation of ports. However, networks as such, subnets, and routers can be created by any user—of course, only for the respective projects that the user belongs to.

Listing the System Extensions

```
 neutron ext-list -c alias -c name
```

Creating A Network

```
neutron net-create NETW_NAME
```

Creates a network for the current project. If you want to specify a network for another project, specify the project's ID with the `--tenant-id` option. To create a shared network that is accessible to all projects, use the `--shared public-net` option.

Listing Networks Belonging To a Project

```
neutron net-list
```

or

```
neutron net-external-list
```

Lists the networks (or external networks, respectively) that belong to a project.

Creating a Subnet

```
neutron subnet-create NETW_NAME_OR_ID NETW_IN_CIDR_NOTATION
```

For example:

```
neutron subnet-create netw1 10.0.0.0/24
```

Creates a subnet for the current project. In the example above, the specified network address (`10.0.0.0/24`) will be associated with the network named `netw1`.

Creating a Subnet with a Specific Gateway IP

```
neutron subnet-create --gateway 10.0.0.254 netw1 10.0.0.0/24
```

Creating a Subnet Without Gateway

```
neutron subnet-create --no-gateway netw1 10.0.0.0/24
```

Creating a Subnet with DHCP Disabled

```
neutron subnet-create netw1 10.0.0.0/24 --enable_dhcp False
```

Creating a Subnet with a Specific Set of DNS Nameservers

```
neutron subnet-create netw2 40.0.0.0/24 \
  --dns_nameservers list=true 8.8.8.7 8.8.8.8
```

Creating a Subnet with a Specific Set of Host Routes

```
neutron subnet-create netw2 40.0.0.0/24 --host_routes \
  type=dict list=true destination=40.0.1.0/24,nexthop=40.0.0.2
```

Creating a Subnet with Multiple Allocation Pools

```
neutron subnet-create --allocation-pool \
  start=192.168.5.31,end=192.168.5.34 netw2 40.0.0.0/24
```

The `--allocation-pool` option can be specified multiple times.

Deleting a Network

```
neutron net-delete NETW_NAME_OR_ID
```

NOTE: Deleting Networks

If a network cannot be deleted (although you have the respective permissions to delete it), it is because it still has ports assigned: either from a router or from a running VM instance.

Deleting a Subnet

```
neutron subnet-delete SUBNET_NAME_OR_ID
```

2.4.2 Creating or Modifying Ports and Routers

Usually, you do not need to create or modify ports manually as described below. In most cases, OpenStack will take care of that automatically (for example, when an instance is launched, a floating IP is associated, or a router interface is created).

A port is an attachment port to a L2 OpenStack Networking network. When a port is created on the network, it will be associated with a security group. If no security group is specified, it will be associated with a default security group. By default, the port will be allocated an available fixed IP address out of one of the designated subnets for each IP version. Users of the OpenStack Networking API can either choose a specific IP address from the block, or let OpenStack Networking choose the first available IP address. When the port is destroyed, the allocated addresses return to the pool of available IPs on the subnet(s).

As a user, you can only modify certain port attributes. For any other changes, contact your cloud administrator.

Listing Ports for a Project

```
neutron port-list
```

The `device_owner` field in the output describes who owns the port. A port whose `device_owner` begins with `network:` is created by OpenStack Networking. A port whose `device_owner` begins with `compute:` is created by OpenStack Compute.

Listing Details for a Port

```
neutron port-show PORT_ID
```

Updating a Port

```
neutron port-update PORT_NAME_OR_ID
```

A router is used to interconnect subnets and forward traffic among them. Another feature of the router is to NAT internal traffic to external networks. A router has an interface for each subnet it is associated with. By default the IP address of such interface is the subnet's gateway IP. Also, whenever a router is associated with a subnet, a port for that router interface will be added to the subnet's network.

Creating a Router

```
neutron router-create ROUTER_NAME
```

Creates a router with the specified *ROUTER_NAME* for the current project.

Listing Routers

```
neutron router-list
```

Lists all routers for the current project.

Deleting a Router

```
neutron router-delete ROUTER_NAME_OR_ID
```

Setting the External Network Gateway for a Router

```
neutron router-gateway-set ROUTER_ID EXTERNAL_NETW_ID
```

Connects the specified external network to the specified router.

Removing an External Network Gateway for a Router

```
neutron router-gateway-clear ROUTER_ID
```

Adding an Internal Network Interface to a Router

```
neutron router-interface-add ROUTER_ID SUBNET_ID
```

Connects the subnet specified with *SUBNET_ID* to the specified router.

Removing an Internal Network Interface from a Router

```
neutron router-interface-delete ROUTER_ID SUBNET_ID
```

For more information and example network setups, refer to the OpenStack *Networking Administration Guide* at http://docs.openstack.org/grizzly/openstack-network/admin/content/.

2.5 Launching Instances

Instances are virtual machines that run inside the cloud. To start an instance, a virtual machine image must exist that contains the following information: which operating system to use, a username and password with which to log in to the instance, file storage etc. The cloud contains a pool of such images that have been uploaded to Image and are accessible to members of different projects.

When starting an instance, you need to specify the following key parameters:

Flavor

> In OpenStack, flavors define the compute, memory, and storage capacity of `nova` computing instances. To put it simply, a flavor is an available hardware configuration for a server. It defines the “size” of a virtual server that can be launched.
>
> For more details and a list of default flavors available, refer to Section “Managing Flavors” (Chapter 2, *Using OpenStack Command Line Clients*, ↑*User Guide for Administrators*).

Keypair

> Keypairs are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package.
>
> Create at least one keypair per project. If you already have generated a keypair with an external tool, you can import it into OpenStack. The keypair can be used for multiple instances belonging to that project.
>
> For details, refer to Section 2.6.2, “Creating or Importing Keys” (page 63).

Security Group

> In SUSE Cloud, security groups are used to define which incoming network traffic should be forwarded to instances. Security groups hold a set of firewall policies (security group rules).
>
> For details, refer to Section 2.6.1, “Configuring Security Groups and Rules” (page 59).

Network

> Instances can belong to one or multiple networks. By default, each instance is given a fixed IP address, belonging to the internal network.

If needed, you can assign a floating (public) IP address to a running instance and attach a block storage device (`volume`) for persistent storage. For details, refer to Section 2.6.3, “Managing IP Addresses” (page 64).

Before you can launch an instance, you need to look up a few parameters, for example, which images, flavors, and security groups are available. Proceed as follows:

Procedure 2.2: *Launching an Instance*

1 On a shell, source the OpenStack RC file. For details, refer to Section 2.2, “OpenStack RC File” (page 50).

2 Look up the available flavors:

```
nova flavor-list
```

Memorize the ID of the flavor that you want to use for your instance.

3 Look up the available images:

```
nova image-list
```

Memorize the ID of the image that you want to boot your instance from.

4 Look up the available security groups:

```
neutron security-group-list
```

If you have not created any specific security groups, you can only assign the instance to the default security group.

5 Look up your keypair's name (for SSH access) and memorize it:

```
nova keypair-list
```

6 Now you have all the parameters at hand for starting an instance. Do so with the following command:

```
nova boot --flavor FLAVOR_ID --imageIMAGE_ID --key_name KEY_NAME \
--security_group NAME_OF_SEC_GROUP NAME_FOR_INSTANCE
```

The command returns a list of instance properties, including the `status` of the instance. The status `BUILD` indicates that the instance has started, but is not yet online.

7 Check if the instance is online:

```
nova list
```

This command lists all instances of the project you belong to, including their ID, their name, their status, and their private (and if assigned, their public) IP addresses. If your instance's status is `ACTIVE`, the instance is online.

To refine the search, run `nova help list` to view the available options for the command.

If you did not provide a keypair on starting and have not touched security groups or rules so far, by default the instance can only be accessed from inside the cloud via VNC at this point. Even pinging the instance is not possible. To change this, proceed with Section 2.6, “Configuring Access to the Instances” (page 59).

2.6 Configuring Access to the Instances

Access to an instance is mainly influenced by the following parameters:

- security groups and rules
- keypairs
- IP addresses

For SSH access to an instance, you usually need to provide a keypair at launch time. The security rules need adjustment, too, since the default rules block access to SSH ports and prevent pinging an instance. To make the instance also accessible from outside the cloud, assign a floating (public) IP address.

2.6.1 Configuring Security Groups and Rules

In SUSE Cloud, security groups are used to define which incoming network traffic should be forwarded to instances. Security groups hold a set of firewall policies (security group rules).

2.6.1.1 Security Groups

When launching an instance, you need to define which security groups it should belong to. A default security group is available for each project. It allows all network traffic from other members of this group and discards traffic from other IP addresses and groups.

Multiple security groups for a project can be defined, with each group holding a different set of firewall policies. This is useful if you have groups of instances that should differ in firewall configuration (for example, front-end and back-end servers). An instance can be assigned to multiple security groups.

Security groups can now be managed with the `neutron security-group*` commands, provided by the `python-quantumclient` package. Alternatively, you can still use the known `nova secgroup_*-rule` commands, provided by the `python-novaclient` package.

Listing Security Groups

```
neutron security-group-list --tenant_id PROJECT_ID
```

Lists all security groups for the specified project, including the groups' descriptions.

Creating a Security Group

```
neutron security-group-create SEC_GROUP_NAME GROUP_DESCRIPTION
```

Creates a new security group with the specified name and description.

Deleting a Security Group

```
neutron security-group-delete SEC_GROUP_NAME_OR_ID
```

Deletes the specified group.

NOTE: Deleting Security Groups

The default security group for a project cannot be deleted.

If another group cannot be deleted, it is because it is still assigned to a running instance.

2.6.1.2 Security Group Rules

You can adjust rules of the default security group as well as rules of any other security group that has been created. As soon as the rules for a group are modified, the new rules are automatically applied to all running instances belonging to that security group.

Adjust the rules in a security group to allow access to instances via different ports and protocols. This is necessary to be able to access instances via SSH, to ping them, or to allow UDP traffic (for example, for a DNS server running on an instance).

Rules in security groups are specified by the following parameters:

IP Protocol
: Protocol to which the rule will apply. Choose between TCP (for SSH), ICMP (for pings), and UDP.

Port/Port Range
: For TCP or UDP, define a single port or a port range to open on the virtual machine. ICMP does not support ports. In that case, enter values that define the codes and types of ICMP traffic to be allowed.

Source of traffic
: Decide whether to allow traffic to instances only from IP addresses inside the cloud (from other group members) or from *all* IP addresses. Specify either an IP address block (in CIDR notation) or a security group as source. Using a security group as source will allow any instance in that security group to access any other instance.

If no further security groups have been created, any instances are automatically assigned to the default security group (if not specified otherwise). Unless you change the rules for the default group, those instances cannot be accessed from any IP addresses outside the cloud.

Procedure 2.3: *Configuring Security Group Rules*

Security group rules can now be modified with the `neutron security-group-rule*` commands, available from the `python-quantumclient` package. Alternatively, you can still use the known `nova secgroup_*-rule` commands, provided by the `python-novaclient` package. Proceed as follows:

1. On a shell, source the OpenStack RC file. For details, refer to Section 2.2, “OpenStack RC File” (page 50).

2. Look up the existing rules for a security group:

   ```
   neutron security-group-rule-list SEC_GROUP_NAME
   ```

3. To enable SSH access to the instances:

   3a Either from *all* IP addresses (specified as IP subnet in CIDR notation as `0.0.0.0/0`):

   ```
   neutron security-group-rule-create --direction ingress \
     --protocol tcp --port_range_min 22 --port_range_max 22 \
     SEC_GROUP_NAME_OR_ID
   ```

   3b Alternatively, you can allow only IP addresses from other security groups (`source groups`) to access the specified port:

   ```
   neutron security-group-rule-create --direction ingress \
     --protocol tcp --port_range_min 22 --port_range_max 22  \
     --remote-group-id SOURCE_GROUP_NAME_OR_ID SEC_GROUP_NAME_OR_ID
   ```

4. To enable pinging the instances:

   4a Either from *all* IP addresses (specified as IP subnet in CIDR notation as `0.0.0.0/0`):

   ```
   neutron security-group-rule-create --protocol icmp --direction ingress
     SEC_GROUP_NAME_OR_ID
   ```

   This command allows access to all codes and all types of ICMP traffic, respectively.

   4b Alternatively, you can allow only members of other security groups (`source groups`) to ping instances:

   ```
   neutron security-group-rule-create --protocol icmp --direction ingress
    \
     --remote-group-id SOURCE_GROUP_NAME_OR_ID SEC_GROUP_NAME_OR_ID
   ```

5. To allow access via UDP port (for example, for a DNS server running on a VM):

5a Either from *all* IP addresses (specified as IP subnet in CIDR notation as `0.0.0.0/0`):

```
neutron security-group-rule-create --protocol udp \
   --port_range_min 53 --port_range_max 53 --direction ingress  \
   --remote-ip-prefix 0.0.0.0/0 SEC_GROUP_NAME
```

5b Alternatively, you can allow only IP addresses from other security groups (`source groups`) to access the specified port:

```
neutron security-group-rule-create --protocol udp \
  --port_range_min 53 --port_range_max 53 --direction ingress \
  --remote-group-id SOURCE_GROUP_ID_OR_NAME SEC_GROUP_NAME
```

6 To delete a security group rule:

```
neutron security-group-rule-delete SEC_GROUP_ID
```

2.6.2 Creating or Importing Keys

Keypairs are SSH credentials that are injected into images when they are launched. For this to work, the image must contain the `cloud-init` package.

Create at least one keypair per project. If you already have generated a keypair with an external tool, you can import it into OpenStack. The keypair can be used for multiple instances belonging to that project.

In case an image uses a static `root` password or a static key set (neither is recommended), you do not need to provide a keypair on starting of the instance.

Procedure 2.4: *Creating or Importing Keys*

Use the `nova keypair-add` command to generate a new keypair, or to upload an existing public key.

1 To generate a new keypair, execute the following commands:

```
nova keypair-add KEY_NAME > MY_KEY.pem
      chmod 600  MY_KEY.pem
```

The first command generates a new keypair named *`KEY_NAME`*, writing the private key to the file *`MY_KEY`*`.pem` and registering the public key at the Compute database.

The second command changes the permissions of the file *MY_KEY*.pem so that only you can read and write to it.

2. If you already have generated a keypair, with the public key located at ~/.ssh/id_rsa.pub, you can upload the public key with the following command:

   ```
   nova keypair-add --pub_key ~/.ssh/id_rsa.pub KEY_NAME
   ```

   The command registers the public key at the Compute database and names the keypair *KEY_NAME*.

3. Check if the uploaded keypair appears in the list of available keypairs:

   ```
   nova keypair-list
   ```

2.6.3 Managing IP Addresses

Each instance can have two types of IP addresses: private (fixed) IP addresses and public (floating) ones. Private IP addresses are used for communication between instances, and public ones are used for communication with the outside world. When an instance is launched, it is automatically assigned private IP addresses in the networks to which it is assigned. The private IP stays the same until the instance is explicitly terminated. (Rebooting the instance does not have an effect on the private IP addresses.)

A floating IP is an IP address that can be dynamically added to a virtual instance. In OpenStack Networking, cloud operators can configure pools of floating IP addresses. These pools are represented as external networks. Floating IP are allocated from a subnet that is associated with the external network. You can allocate a certain number of floating IPs to a project—the maximum number of floating IP addresses per project is defined by the quota. From this set, you can then add a floating IP address to an instance of the project.

Before you can assign a floating IP address to an instance, you first need to allocate floating IPs to a project.

After floating IP addresses have been allocated to the current project, you can assign them to running instances. One floating IP address can be assigned to only one instance at a time.

Floating IP addresses can be managed with the `neutron` commands, provided by the `python-quantumclient` package. Alternatively, you can still use the known `nova` commands, provided by the `python-novaclient` package.

Procedure 2.5: *Allocating Floating (Public) IPs to a Project*

1. On a shell, source the OpenStack RC file. For details, refer to Section 2.2, “OpenStack RC File” (page 50).

2. To find pools that provide floating IPs, list all external networks:

   ```
   neutron net-list --router:external=True
   ```

   Among these should be at least one network named `floating`.

3. To allocate a floating IP from this pool to the current project:

   ```
   neutron floatingip-create FLOAT_NETW_ID_OR_NAME
   ```

   To allocate several floating IPs, repeat the command.

4. To view the floating IP addresses that have been assigned to the current project, use the following command:

   ```
   neutron floatingip-list
   ```

5. To release a floating IP from the project, use:

   ```
   neutron floatingip-delete ID_OF_FLOATING_IP
   ```

Procedure 2.6: *Assigning Floating (Public) IP Addresses to Instances*

After floating IP addresses have been allocated to the current project, you can assign them to running instances. One floating IP address can be assigned to only one instance at a time.

To assign a floating IP to an instance, you need to know both the ID of the floating IP and of the port that has been allocated to the instance.

1. On a shell, source the OpenStack RC file. For details, refer to Section 2.2, “OpenStack RC File” (page 50).

2. To find out the port that has been allocated to an instance, proceed as follows:

2a List all instances of the project you belong to:

```
nova list
```

The commands shows the running instances, including their ID, their name, their status, and their private (and if assigned, their public) IP addresses.

2b Memorize the instance's ID.

2c Look up the port belonging to the instance ID:

```
neutron port-list -- device_id INSTANCE_ID
```

3 To look up the ID of the floating IP address that you want to assign:

```
neutron floatingip-list
```

It lists all floating IPs that belong to the current project, including their ID.

4 Now you can assign a floating IP to the instance with the following command:

```
neutron floatingip-associate ID_OF_FLOATING_IP PORT_ID
```

5 To check if the floating IP has been assigned, run:

```
neutron floatingip-show ID_OF_FLOATING_IP
```

6 To remove the floating IP address from the instance, use:

```
neutron floatingip-disassociate ID_OF_FLOATING_IP
```

Documentation Updates A

This chapter lists content changes for this document since the initial release of SUSE® Cloud 1.0.

A.1 SUSE Cloud 2.0

The document was updated on the following dates:

- Section A.1.1, “September 19, 2013 (SUSE Cloud 2.0) ” (page 67)

A.1.1 September 19, 2013 (SUSE Cloud 2.0)

OpenStack Service Names

Replaced code names with the OpenStack services' real names, according to `https://wiki.openstack.org/wiki/Documentation/Conventions#General_style_conventions`. For example, talk of *OpenStack Image* instead of *Glance*.

Update to OpenStack Grizzly

The Dashboard and the command line chapter have been updated to the OpenStack Grizzly release on which SUSE Cloud 2.0 is based. This includes the following changes:

- The following section has been added: Section 1.4, “Managing Networks” (page 10).

- Some categories have been added to the Dashboard's main screen, others changed name. Most importantly, the download option for RC files can now be found in the *Access & Security* category. See Section 1.2, “SUSE Cloud Dashboard—Overview” (page 2) and Section 2.2, “OpenStack RC File” (page 50).

- Image upload is now also possible via the Dashboard, though with limited options. For details, refer to Section 1.3, “Managing Images” (page 5)

- The workflow for some tasks has changed, most notably for the following ones: Section 1.5.3, “Launching an Instance” (page 20), Procedure 1.10, “Configuring Security Group Rules” (page 28), Procedure 1.13, “Assigning Floating (Public) IP Addresses to Instances” (page 34), and Procedure 1.18, “Attaching Volumes to Instances” (page 44).

- On the command line, managing security groups, security group rules, and IP addresses can now also be done with the `neutron` commands, provided by the `python-quantumclient` package. For details, see Section 2.6.1, “Configuring Security Groups and Rules” (page 59) and Section 2.6.3, “Managing IP Addresses” (page 64). However, the former way of executing these tasks with the `nova` python client is also still supported.