

Cloud Tutorial

Last updated: DATEPLACEHOLDER

Authors:

Alexander Vapirev (KU Leuven), Thomas Danckaert (UAntwerpen)

Acknowledgement: VSCentrum.be



Audience:

This document is a hands-on guide to the VSC cloud computing platform, which relies on the open-source software [OpenStack](#). It should complement the official documentation at <https://docs.openstack.org>.

Contact Information:

For all questions concerning the VSC cloud computing platform, please contact cloud@vscentrum.be.

We welcome your feedback, comments and suggestions for improving the OpenStack Tutorial

Glossary

NFS Network File System, a protocol for sharing file systems across a network, often used on Unix(-like) operating systems.

NIC Network Interface Controller, a (virtualized) hardware component that connects a computer to a network.

REST REpresentational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services.

YAML A human-readable text-based data serialization format.

Heat Heat is the OpenStack orchestration service, which can manage multiple composite cloud applications using templates, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.

Heat Orchestration Template A Heat Orchestration Template (HOT) is a text file which describes the infrastructure for a cloud application. Because HOT files are text files in a YAML-based format, they are readable and writable by humans, and can be managed using a version control system. HOT is one of the template formats supported by Heat, along with the older CloudFormation-compatible CFN format.

Horizon Horizon is the name of the OpenStack Dashboard.

OpenStack OpenStack (<https://www.openstack.org>) is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS), whereby virtual servers and other resources are made available to customers.

OpenStack Dashboard OpenStack Dashboard (Horizon) provides administrators and users with a graphical interface to access, provision, and automate deployment of cloud-based resources. The design accommodates third party products and services, such as billing, monitoring, and additional management tools. The dashboard is also brand-able for service providers and other commercial vendors who want to make use of it. The dashboard is one of several ways users can interact with OpenStack resources. Developers can automate access or build tools to manage resources using the native OpenStack API or the EC2 compatibility API..

OpenStack Instance OpenStack Instances are virtual machines, which are instances of a system image that is created upon request and which is configured when launched. With traditional virtualization technology, the state of the virtual machine is persistent, whereas OpenStack supports both persistent and ephemeral image creation.

OpenStack Volume OpenStack Volume is a detachable block storage device. Each volume can be attached to only one instance at a time.

share A share is a remote, mountable file system. Users can mount and access a share on several hosts at a time.

stack In the context of OpenStack, a stack is a collection of cloud resources which can be managed using the Heat orchestration engine.

Contents

Glossary	3
1 Access to the VSC Cloud	8
1.1 Dashboard Login	8
1.2 Application Credentials	9
2 The OpenStack Dashboard	11
3 Upload and manage images	14
4 Configure access and security for instances	17
4.1 Security Groups	17
4.2 SSH keypairs	17
4.3 Floating IP addresses	18
5 Launch and manage instances	22
5.1 Launch an instance	22
5.2 Connect to your instance using SSH	25
5.3 Track usage for instances	26
5.4 Create an instance snapshot	26
5.5 Manage an instance	26
6 Create and manage volumes	28
7 Orchestration Using Heat	31
7.1 Heat Orchestration Templates	31
7.2 The Template Generator	32
7.3 Managing stacks	32

Introduction

The VSC cloud platform uses the open-source software [OpenStack](#), version “rocky”. This guide explains the specifics of the VSC environment, and provides a hands-on introduction to OpenStack. For reference, you should consult the OpenStack project’s own documentation at <https://docs.openstack.org>.

Chapter 1

Access to the VSC Cloud

Access to the VSC cloud is linked to the central VSC account system (account.vscentrum.be), so you do not need a separate login or password. In order to use the cloud services,

- you need an active VSC account and
- your account must be a member of one or more OpenStack projects.

New users can obtain an account at www.vscentrum.be/cluster-doc/account-request. Contact cloud@vscentrum.be if you want to start a new OpenStack project, or join an existing one.

You can interact with the VSC Cloud using the OpenStack Dashboard, a web interface, or the OpenStack command line interface, which you can use from any system, and which is installed for you on the UGent login node `login.hpc.ugent.be`. You can log in to the Dashboard using the VSC accountpage, as illustrated in the next section. To get access from the command line interface, you'll need to obtain an application credential, as explained in section [1.2](#).

1.1 Dashboard Login

You can access the OpenStack web interface, or Dashboard, via cloud.vscentrum.be.

To log in, choose the (default) authentication method *VSC Accountpage* and click **Connect**.

From here on, follow the standard procedure to log in to your VSC account, using your home institution's single sign-on system. You can find a detailed description in the HPC tutorial at www.vscentrum.be/support/tut-book/vsc-tutorials. The following chapters explain how to accomplish basic tasks using the Dashboard.

1.2 Application Credentials

If you want to use the OpenStack command line interface — or, for advanced users, use the OpenStack APIs directly — you need to identify yourself using an application credential. An application credential contains a secret piece of information which grants access to an OpenStack project on your behalf.

You can create an application credential using the dashboard:

1. Log in to the dashboard, and, if you are a member of more than one project, select the project for which you want to create an application credential.
2. Open the **Identity** tab, and click **Application Credentials**.
3. You can now see an overview of your application credentials (initially none). Click **Create Application Credential**.
4. Fill out the **Create Application Credential** dialog:

Name, Description Choose a name (mandatory) and description that remind you of the purpose of this credential.

Secret We recommend to leave this empty, in which case OpenStack will generate a random secret for you.

Expiration Date, Expiration Time It is good practice make the token expire. An expiration date limits the impact if the secret is accidentally exposed, and you can always create a new credential when an old one is expired.

Roles A role defines a set of access rights. By selecting a subset of roles for this credential, you can limit the access rights granted by this credential. It is a good idea to select only the minimal set of roles required for the task you want to accomplish.

Click **Create Application Credential**.

5. A summary dialog with the credential's id, name, and secret is displayed. If you close the window, you can't retrieve the secret anymore, so you should save it now. A convenient solution is to download the openrc file, a shell script that sets the appropriate environment variables for the command line interface.

The newly created credential is now shown in the overview. If you accidentally expose a credential somewhere, you should delete it here to prevent unauthorized access to the system.

Chapter 2

The OpenStack Dashboard

After login, you can see the Overview tab of Horizon, the OpenStack Dashboard.



This chapter briefly describes the different components of the dashboard. You can read the official documentation at <https://docs.openstack.org/horizon/rocky/user>.

Note: The VSC cloud uses a customized dashboard. Some features mentioned in the official OpenStack documentation were intentionally removed, please contact cloud@vscentrum.be if you need access to one of these disabled features.

Project tab

Resources (instances, data volumes, networks, ...) in OpenStack are organized into different projects, and every user is a member of one or more projects. Every project member has full access to all of the project's resources.

From the Project tab, you can access the following categories:

API Access View API endpoints.

Compute

- Overview: View reports for the project.
- Instances: View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through VNC.
- Images: View images and instance snapshots created by project users, plus any images that are publicly available. Create, edit, and delete images, and launch instances from images and snapshots.
- Key Pairs: View, create, edit, import, and delete key pairs.
- Server Groups: Server groups provide a mechanism to group servers according to certain policy.

Volumes

- Volumes: View, create, edit, and delete volumes.
- Snapshots: View, create, edit, and delete volume snapshots.

Network

- Networks: Create and manage public and private networks.
- Security Groups: View, create, edit, and delete security groups and security group rules..
- Floating IPs: Allocate an IP address to or release it from a project

Orchestration

- Stacks: Use the REST API to orchestrate multiple composite cloud applications.
- Resource types: Show a list of all the supported resource types for HOT templates.
- Template versions: The version of a Heat template specifies the format of the template and also the corresponding features that will be validated and supported.
- Template generator: A graphical interface to build and edit templates.

Shares

- Shares: Create and manage shares.

Identity tab

From the Identity tab, you can access the following categories:

Projects View, create, assign users to, remove users from, and delete projects.

Users View, create, enable, disable, and delete users.

Application Credentials With application credentials, a user can grant applications limited access to their cloud resources.

Chapter 3

Upload and manage images

A virtual machine image, referred to in this document simply as an image, is a single file that contains a virtual disk that has a bootable operating system installed on it. Images are used to create virtual machine instances within the cloud. The image files themselves are never modified, but you can copy the image into a persistent instance (see chapter 5).

As a user of the VSC cloud, you can upload and manage your own virtual machine images. For information about creating image files, see the [OpenStack Virtual Machine Image Guide](#).

Note: Shared storage in the VSC cloud is connected to a separate network, which is only accessible from within the OpenStack environment. Therefore, if you want to access your VM from outside of OpenStack, and use the shared storage at the same time, you must make sure your VM image is configured use multiple network interface cards (NICs).

You can choose who can access an image you have created. The following access policies for images exist:

public Public images are provided by the VSC, and can be accessed by all users.

private If you create a private image, only members of the same project have access.

shared You can also choose to share your image with a list of other projects.

community Community images are user-created images which are freely accessible to all other users.

Note: You can also use the **openstack** and **glance** command-line clients or the Image service to manage images.

Upload an image

Follow this procedure to upload an image to a project:

1. Open the Compute tab and click Images category.
2. Click Create Image.

The Create An Image dialog box appears.

3. Enter the following values:

Image Name Enter a name for the image.

Image Description Enter a brief description of the image.

Image Source

File Browse for the image file on your file system and add it.

Format Select the image format (for example, QCOW2) for the image.

Image Requirements

Architecture Specify the architecture. For example, **i386** for a 32-bit architecture or **x86_64** for a 64-bit architecture.

Kernel, Ramdisk Can be left empty, as this is determined by the image file.

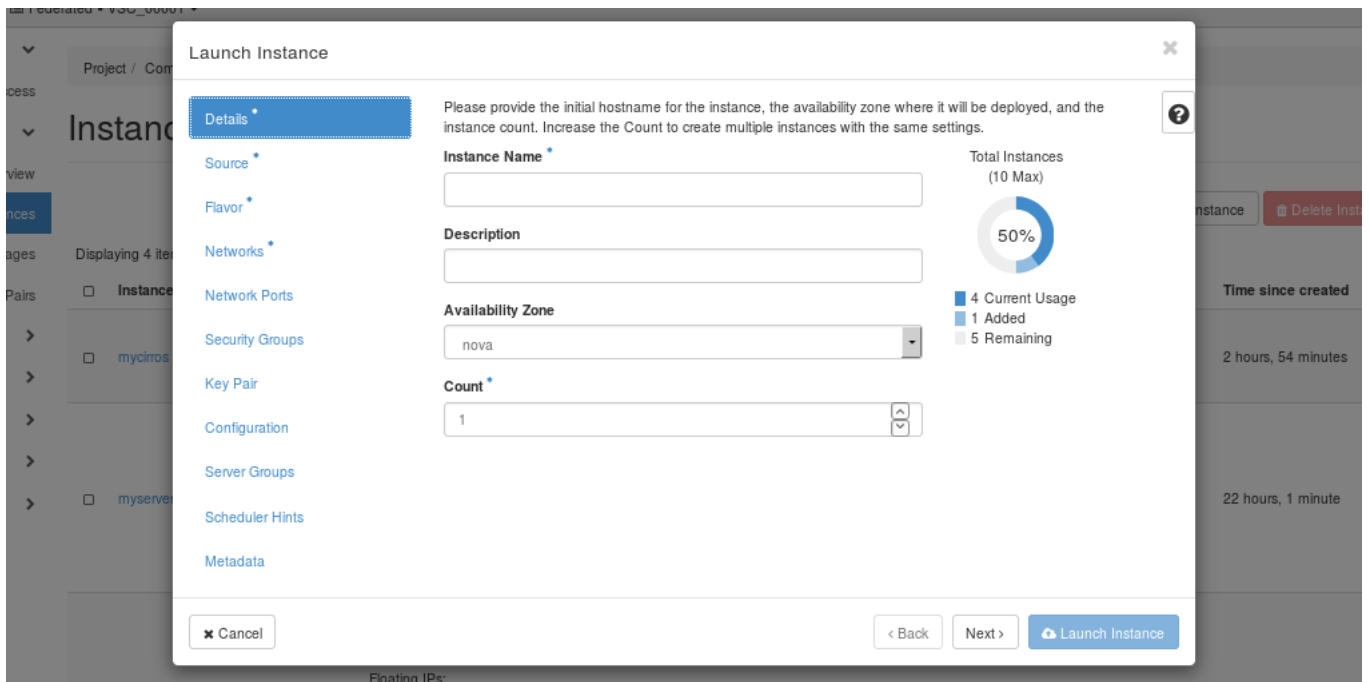
Minimum Disk (GB), Minimum RAM (MB) Choose suitable values, according to the requirements of your image's operating system.

Protected If set to **Yes**, users need to unlock the image before they are able to delete it.

Metadata You can add resource metadata in this tab. The glance Metadata Catalog provides a list of metadata image definitions.

4. Click **Create Image**. The image is queued to be uploaded. It might take some time before the status changes from Queued to Active.

Another way to create an image is to check-mark on the left side one of the available images and then click on Launch on the right of the screen. That way the mandatory fields in 'Image Details' tab in the 'Create Image' pop-up dialog are automatically filled and the user will be directly presented with the 'Launch Instance'.



Update an image

Follow this procedure to update an existing image.

1. Open the project tab and click the Images category.
2. Select the image that you want to edit.
3. In the Actions column, open the drop-down menu and select **Edit Image**.
4. In the Edit Image dialog box, you can perform various actions. For example:
 - Change the name of the image.
 - Change the description of the image.
 - Change the format of the image.
 - Change the minimum disk of the image.
 - Change the minimum RAM of the image.
 - Change the protected status of the image.
 - Change the metadata of the image.
5. Click **Edit Image**.

Chapter 4

Configure access and security for instances

4.1 Security Groups

Openstack security groups are sets of IP filter rules that define networking access and are applied to all instances within a project. In the VSC cloud, each project contains a default security group, which allows you to ping instances and connect using SSH. You can add rules to the default security group or add new security groups with rules.

4.2 SSH keypairs

Key pairs are SSH credentials that can be automatically injected into an instance when it is launched. To use key pair injection, the image that the instance is based on must contain the **cloud-init** package. Each project should have at least one key pair. For more information, see the section *Add a key pair*. For general instructions on SSH keys, we refer to chapter 2 of [the VSC HPC tutorial](#).

Note: In the VSC cloud, public keys you've uploaded to the VSC account portal are automatically available in OpenStack projects, so you don't have to create or import new keys.

If you have generated a key pair with an external tool, you can import it into OpenStack. The key pair can be used for multiple instances that belong to a project. For more information, see the section *Import a key pair*.

Note: A key pair belongs to an individual user, not to a project. To share a key pair across multiple users, each user needs to import that key pair.

Note: This chapter only explains the required configuration in order to make your instance accessible via SSH. For instructions on how to connect to a running instance, once it has been configured correctly, see page 25 of chapter 5.

Add a key pair

1. Open the Compute tab.
2. Click the Key Pairs tab, which shows the key pairs that are available for this project.
3. Click Create Key Pair.
4. In the Create Key Pair dialog box, enter a name for your key pair, and click Create Key Pair.
5. Respond to the prompt to download the key pair.
6. Save the ***.pem** file locally.
7. To change its permissions so that only you can read and write to the file, run the following command:

```
$ chmod 0600 yourPrivateKey.pem
```

Note: If you are using the OpenStack Dashboard from a Windows computer, use PuTTYgen to load the ***.pem** file and convert and save it as ***.ppk**. For more information see the [WinSCP web page for PuTTYgen](#), and chapter 2 of [the VSC HPC tutorial](#).

8. To make the key pair known to SSH, run the **ssh-add** command.

```
$ ssh-add yourPrivateKey.pem
```

Import a key pair

1. Open the Compute tab.
2. Click the Key Pairs tab, which shows the key pairs that are available for this project.
3. Click Import Key Pair.
4. In the Import Key Pair dialog box, enter the name of your key pair, copy the public key into the Public Key box, and then click Import Key Pair.

The Compute database registers the public key of the key pair.

The OpenStack Dashboard lists the key pair on the Key Pairs tab.

4.3 Floating IP addresses

When an instance is created in OpenStack and connected to the `_vm` network, it is automatically assigned a fixed IP address in that network. This IP address is permanently associated with the instance until the instance is terminated. However, the `_vm` network can only be reached from within the OpenStack environment.

If you need to access an instance from the outside, you need to use one of your project's floating IP addresses. Unlike fixed IP addresses, floating IP addresses can have their associations modified at any time, regardless of the state of the instances involved. In the VSC cloud, floating IP's are accessible from the the Ugent login node `login.hpc.ugent.be`.

Floating ip port forwarding

OpenStack’s networking API, called Neutron, makes it possible to forward different ports of the same floating ip to arbitrary ports in one of OpenStack’s virtual networks. This is the recommended way to use floating ip’s in the VSC cloud.

You’ll need to forward a separate port for every service you wish to reach. For example, if you want to access an instance using SSH, you’ll need to create a port forwarding rule from a selected port of the floating IP, to the port in the `_vm` network where your instance’s SSH server is listening (typically port 22).

You can quickly set up such forwarding rules using `neutron_port_forward`, a command line tool available on the UGent login node, `login.hpc.ugent.be`. In order to use it, you must create an application credential for the role “User”, and save it as an openrc file (see section 1.2 on page 9). Transfer the openrc file to your VSC storage space, so `neutron_port_forward` can read it. To set up new port forwarding rules, run the script providing the path to the openrc file as an argument to the `-o` option, and a file describing your port forwarding configuration as argument to the `-m` option:

```
$ neutron_port_forward -o <openrc file.> -m <ini-file>
```

The following is an example configuration file:

```
1 [DEFAULT]
2 floatingip=193.190.85.40
3 network=_vm
4
5 [classa]
6 pattern=classa-(\d+)
7 22=52000:100:22
8 5900=55900
9
10 [classb]
11 pattern=classb-(\d+)
12 80=52080
```

Here we define defaults for the floating ip and target network, and two classes. Instances are assigned to a class if their name matches the regular expression given in `pattern`. The value of `pattern` must be a valid Python regular expression, and the first capturing group (if any) must match an integer.

Port forwarding rules are given in the form `target=source(:multiplier:offset)`. This will set up a forwarding rule from the floating IP port

$$(\text{source} + \text{multiplier} * i + \text{offset}) \rightarrow \text{target} ,$$

where i is the integer matched by the first capturing group, and “target” is a port of the fixed IP for the instance in the chosen network, in this case the `_vm` network. “multiplier” and “offset” are optional and default to 1 and 0 respectively. In our example, this results in the following set of port forwarding rules, all for the floating IP address 193.190.85.40:

```

52122 → classa-1:22
52222 → classa-2:22
...
55901 → classa-1:5900
55902 → classa-2:5900
...
52081 → classb-1:80
52082 → classb-2:80
...

```

You can also see an overview of existing port forwarding rules for the ip addresses in your configuration file using `neutron_port_forward --show`. Each rule has an internal id, which you can see if you combine the options `--show` and `--id` as follows:

```
$ neutron_port_forward -o <openrc file.> -m <ini-file> --show --id
```

To remove port forwarding rules, use the option `--remove=<list of id's>` with a comma-separated list of the id's of the rules you want to remove. Rules are removed automatically if the target instance is deleted.

`neutron_port_forward` provides a few more options and advanced features, run the command with the `--help` option for more information.

Attach a floating ip

A floating IP address can also be attached to an instance, just like the fixed IP addresses. Because this approach uses one of the few available floating ip addresses for every instance you want to connect to, you should only use it for testing purposes.

Note: If you want to use a floating ip for port forwarding as in the previous section, it cannot be attached to an instance at the same time.

This procedure details the reservation of a floating IP address from an existing pool of addresses and the association of that address with a specific instance.

1. Open the Network tab.
2. Click the Floating IPs tab, which shows the floating IP addresses allocated to your project.
3. In the Floating IPs list, click Associate next to the address you want.
4. In the Manage Floating IP Associations dialog box, choose the following options:

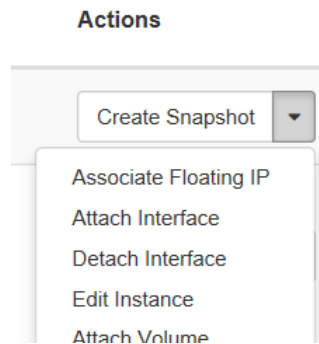
IP Address This field is filled automatically.

Port to be associated Select a port from the list. The list shows all the instances with their fixed IP addresses.

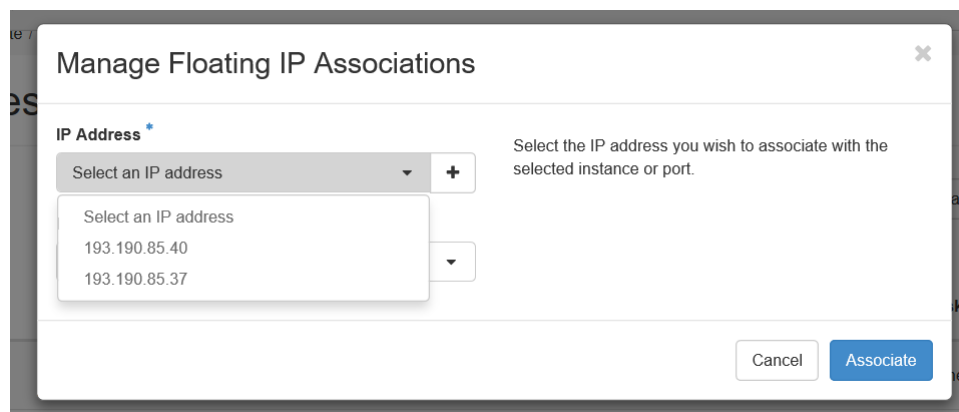
5. Click Associate.

Another way to associate a floating IP is after the user has already launched an instance which appears in the list of running instances in the Project->Compute->Instances tab:

1. Expand the drop-down menu on right next to the instance
2. Select Associate Floating IP



3. A pop-up window will appear and under IP Address select from the drop-down menu an IP address from the available pool.



4. Click Associate

If the IP has been successfully associated in the upper right corner of the browser screen will appear a green confirmation. If not successful a red notification will pop up that something went wrong.

Note: To disassociate an IP address from an instance, click the Disassociate button in the Actions column.

Warning: *Do not* use the Release Floating IP option in the Actions column or on the overview page. This will remove the floating IP from the pool assigned to your project, something which you, as a regular user, cannot undo. If you've accidentally released a floating IP, contact cloud@vscentrum.be to have it restored.

Chapter 5

Launch and manage instances

Instances are virtual machines that run inside the cloud. You can launch an OpenStack Instance from the following sources:

- Images uploaded to the Image service. Note that, because images are read-only, any changes made while the instance is running will be lost when the instance is deleted, unless you choose to “Create New Volume”. If you create a new volume, the VM’s state will persist on the volume, even when the current instance is deleted.
- Image that you have copied to a persistent volume. The instance launches from the volume, which is provided by the **cinder-volume** API through iSCSI.
- Instance snapshot that you took.

5.1 Launch an instance

1. Open the Compute tab and click Instances category.

The dashboard shows the instances with its name, its private and floating IP addresses, size, status, task, power state, and so on.

2. Click Launch Instance.
3. In the Launch Instance dialog box, specify the following values:

Details tab

Instance Name Assign a name to the virtual machine.

Note: The name you assign here becomes the initial host name of the server. If the name is longer than 63 characters, the Compute service truncates it automatically to ensure dnsmasq works correctly.

After the server is built, if you change the server name in the API or change the host name directly, the names are not updated in the dashboard.

Server names are not guaranteed to be unique when created so you could have two instances with the same host name.

Description You can assign a brief description of the virtual machine.

Availability Zone By default, this value is set to the availability zone given by the cloud provider (for example, **us-west** or **apac-south**). For some cases, it could be **nova**.

Count To launch multiple instances, enter a value greater than **1**. The default is **1**.

Launch Instance

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source

Image

Create New Volume

Yes No

Delete Volume on Instance Delete

Yes No

Volume Size (GB)

1

Allocated

Name	Updated	Size	Type	Visibility
Select an item from Available items below				

Available 10

Select one

Q Click here for filters.

Name	Updated	Size	Type	Visibility
------	---------	------	------	------------

Source tab

Select Boot Source Your options are:

Image

Image snapshot

Volume

Volume snapshot

Depending on the type of boot source, the list of available items changes.

Create New Volume If you enable this option when launching from an image or instance snapshot, the image or snapshot will be copied to a volume. This way, the state of your instance persists after shutdown and reboot.

Flavor tab. Specify the size of the instance to launch.

Note: The flavor is selected based on the size of the image selected for launching an instance. For example, while creating an image, if you have entered the value in the Minimum RAM (MB) field as 2048, then on selecting the image, the default flavor is **m1.small**. If a '!' warning sign is displayed next to a resource for one of the flavors, that means that this flavor would exceed the project's quota for that resource, and therefore is not available.

Networks tab. Add one or more networks to the instance.

Network Ports tab. Activate the ports that you want to assign to the instance.

Security Groups tab. 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 is forwarded to instances.

The default security group is assigned to the instance automatically.

Key Pair tab. Specify a key pair.

If the image uses a static root password or a static key set (neither is recommended), you do not need to provide a key pair to launch the instance.

Configuration tab. Specify a customization script that runs after your instance launches.

Server Groups tab.

Scheduler Hints tab.

Metadata tab. Add Metadata items to your instance.

4. Click Launch Instance.

The instance starts on a compute node in the cloud.

Note: If you did not provide a key pair, security groups, or rules, users can access the instance only from inside the cloud through VNC. Even pinging the instance is not possible without an ICMP rule configured.

You can also launch an instance from the Images or Volumes category when you launch an instance from an image or a volume respectively.

When you launch an instance from an image, OpenStack creates a local copy of the image on the compute node where the instance starts.

For details on creating images, see [Creating images manually](#) in the *OpenStack Virtual Machine Image Guide*.

5.2 Connect to your instance using SSH

Before you can connect to your instance using SSH, you must set up a floating IP, as discussed in section 4.3.

You can only reach the floating IP's from the UGent login node `login.hpc.ugent.be`, so you'll need to access the UGent login node first, and hop to your instance from there. If you do not have a suitable private key in your VSC storage space, you need to set up an SSH agent with key forwarding locally, i.e. on the machine where you store the private key of an authorized keypair for the instance. Section 2.1.4 of the HPC tutorial explains how to set this up (<https://www.vscentrum.be/support/tut-book/vsc-tutorials>).

1. Connect to the UGent login node, using the `ssh -A` option to enable agent forwarding:

```
$ ssh -A vsc12345@login.hpc.ugent.be
```

2. Copy the address of the floating IP where your instance can be reached. In our example, the address is 193.190.85.40.
3. From the login node, connect to the instance. Use OpenSSH's `-p` option to specify the port where the instance's SSH server can be reached, e.g. for port 50022:

```
$ ssh -p 50022 ubuntu@193.190.85.40
```

The default images do not allow SSH logins for the root user. There is a default user instead, who can get administrative privileges using `sudo`. In our example, we have used the username `ubuntu` for Ubuntu images. Attempting to log in as root will return an error message with the proper user name.

5.3 Track usage for instances

You can track usage for instances for each project. You can track costs per month by showing meters like number of vCPUs, disks, RAM, and uptime for all your instances.

1. Open the Compute tab and click Overview category.
2. To query the instance usage for a month, select a month and click Submit.
3. To download a summary, click Download CSV Summary.

5.4 Create an instance snapshot

1. Open the Compute tab and click the Instances category.
2. Select the instance from which to create a snapshot.
3. In the actions column, click Create Snapshot.
4. In the Create Snapshot dialog box, enter a name for the snapshot, and click Create Snapshot.

The Images category shows the instance snapshot.

To launch an instance from the snapshot, select the snapshot and click Launch. Proceed with launching an instance.

5.5 Manage an instance

1. Open the Compute tab and click Instances category.
2. Select an instance.
3. In the menu list in the actions column, select the state.

You can resize or rebuild an instance. You can also choose to view the instance console log, edit instance or the security groups. Depending on the current state of the instance, you can pause, resume, suspend, soft or hard reboot, or terminate it.

Difference between *suspend*, *pause*, *shelve*, *shut off*, *delete*

Suspend Stores the state of the VM on the disk, all memory is written to disk, and the server is stopped.

Pause Stores the state of the VM in the (RAM) memory.

Shelve Shelving stops the instance and takes a snapshot of it. Then depending on the value of the *shelved_offload_time* config option, the instance is either deleted from the hypervisor (0), never deleted (-1), or deleted after some period of time (> 0). Shelve preserves all associated data and VM resources but does not retain anything in memory.

Shut off The server is powered down by the user, either through the OpenStack Compute API, or from within the server by issuing a *shutdown -h* command. In this state the user retains all computational resources associated with the VM. The instance can be later restarted.

Delete The VM is deleted and removed from OpenStack together with any associated processes and resources.

For more details see the OpenStack documentation on [Virtual Machine States and Transitions](#) and [Server concepts](#).

Chapter 6

Create and manage volumes

An OpenStack Volume is a block storage device which you attach to instances to enable persistent storage. You can attach a volume to a running instance or detach a volume and attach it to another instance at any time. You can also create a snapshot from or delete a volume. Only administrative users can create volume types.

Create a volume

1. Open the Volumes tab and click Volumes category.
2. Click Create Volume.

In the dialog box that opens, enter or select the following values.

Volume Name Specify a name for the volume.

Description Optionally, provide a brief description for the volume.

Volume Source Select one of the following options:

- No source, empty volume: Creates an empty volume. An empty volume does not contain a file system or a partition table.
- Snapshot: If you choose this option, a new field for Use snapshot as a source displays. You can select the snapshot from the list.
- Image: If you choose this option, a new field for Use image as a source displays. You can select the image from the list.
- Volume: If you choose this option, a new field for Use volume as a source displays. You can select the volume from the list. Options to use a snapshot or a volume as the source for a volume are displayed only if there are existing snapshots or volumes.

Type Leave this field blank.

Size (GB) The size of the volume in gibibytes (GiB).

Availability Zone Select the Availability Zone from the list. By default, this value is set to the availability zone given by the cloud provider (for example, **us-west** or **apac-south**). For some cases, it could be **nova**.

3. Click Create Volume.

The dashboard shows the volume on the Volumes tab.

Attach a volume to an instance

After you create one or more volumes, you can attach them to instances. You can attach a volume to one instance at a time.

1. Open the Volumes tab and click Volumes category.
2. Select the volume to add to an instance and click Manage Attachments.
3. In the Manage Volume Attachments dialog box, select an instance.
4. Enter the name of the device from which the volume is accessible by the instance.

Note: The actual device name might differ from the volume name because of hypervisor settings.

5. Click Attach Volume.

The dashboard shows the instance to which the volume is now attached and the device name.

You can view the status of a volume in the Volumes tab of the dashboard. The volume is either Available or In-Use.

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

Detach a volume from an instance

1. Open the Volumes tab and click the Volumes category.
2. Select the volume and click Manage Attachments.
3. Click Detach Volume and confirm your changes.

A message indicates whether the action was successful.

Create a snapshot from a volume

1. Open the Volumes tab and click Volumes category.
2. Select a volume from which to create a snapshot.
3. In the Actions column, click Create Snapshot.
4. In the dialog box that opens, enter a snapshot name and a brief description.
5. Confirm your changes.

The dashboard shows the new volume snapshot in Volume Snapshots tab.

Edit a volume

1. Open the Volumes tab and click Volumes category.
2. Select the volume that you want to edit.
3. In the Actions column, click Edit Volume.
4. In the Edit Volume dialog box, update the name and description of the volume.
5. Click Edit Volume.

Note: You can extend a volume by using the Extend Volume option available in the More dropdown list and entering the new value for volume size.

Delete a volume

When you delete an instance, the data in its attached volumes is not deleted.

1. Open the Volumes tab and click Volumes category.
2. Select the check boxes for the volumes that you want to delete.
3. Click Delete Volumes and confirm your choice.

A message indicates whether the action was successful.

Chapter 7

Orchestration Using Heat

Heat is the name of the OpenStack orchestration engine, which can manage complete configurations of all servers, volumes, users, networks and routers that make up a cloud application. Instead of managing every component separately, we can create, start, stop or clean up our complete application in a single step. Such a set of collectively managed resources is called a stack.

Heat has its own dashboard interface, which you can find under the **Orchestration** tab of the main OpenStack dashboard. Official documentation for Heat and its dashboard interface can be found at the following locations:

- <https://docs.openstack.org/heat/rocky>
- <https://docs.openstack.org/heat-dashboard/rocky>

7.1 Heat Orchestration Templates

A stack's resources and their mutual dependencies can be specified in a text file, called a Heat Orchestration Template (HOT). The syntax of these templates conforms to the YAML standard, for which many text editors provide specialized editing modes. The following example describes a stack consisting of a single VM:

```
1 heat_template_version: 2018-03-02
2
3 description: Deploy a single compute instance
4
5 parameters:
6   user_network:
7     type: string
8     label: user_network
9     description: Add the required VM network
10    constraints: [ custom_constraint: neutron.network ]
11   user_key:
12     type: string
13     label: ssh_user_key
14     description: Public ssh key for user authentication
15     constraints: [ custom_constraint: nova.keypair ]
16
```

```

17 resources:
18   my_instance:
19     type: OS::Nova::Server
20     properties:
21       security_groups: [ default ]
22       networks: [ network: { get_param: user_network } ]
23       key_name: { get_param: user_key }
24       image: Ubuntu_16.04_2NICs
25       flavor: m1.small

```

Our example contains four main sections:

heat_template_version The HOT specification has evolved since its initial release. The key `heat_template_version` indicates the version of the syntax used in this template. It's value can be a release date or (in recent version) the name of the version.

description A description is optional, but recommended.

parameters An optional section, parameters allow users to configure various properties when instantiating a new stack, without having to edit the template itself. A parameter value can be used elsewhere in the template using the function `get_param`.

resources This section contains all the resources used by the Stack. In this case, there is just a single VM instance.

Optional additional sections are **parameter_groups**, **outputs**, and **conditions**.

The ‘[Template Guide](#)’ in the Heat documentation contains a specification of the HOT format, as well as information on how to describe the various types of resources in a template. VSC also provides some example templates in the repository <https://github.com/hpcugent/openstack-templates>.

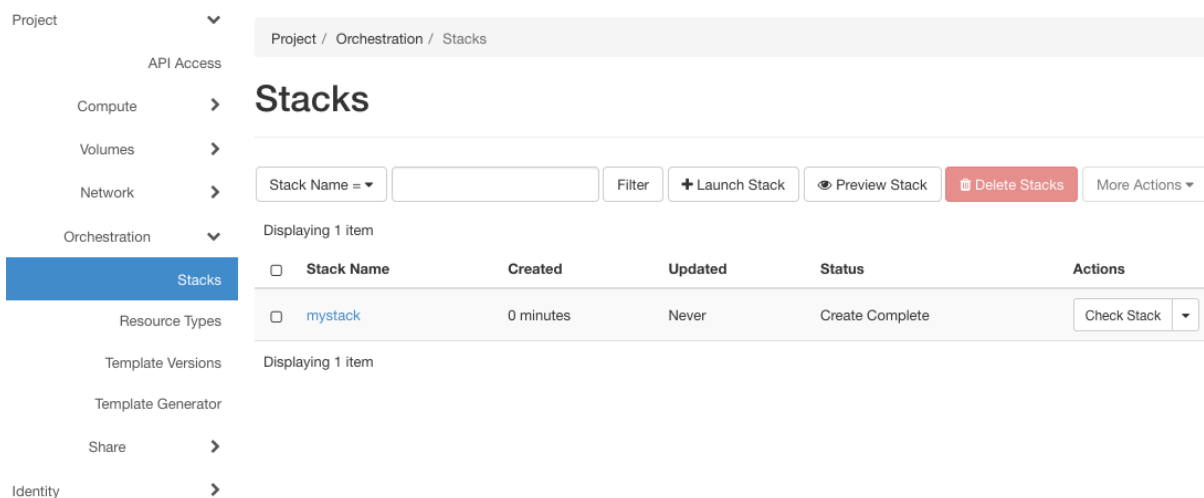
7.2 The Template Generator

The Heat dashboard provides a graphical interface where users can draw templates by dragging resources onto a canvas, and connecting them. Users can then download a template generated from this interface, or immediately instantiate it as a stack.

Note: Currently, there are a number of issues with the template generator, which require manual edits to the generated templates. Therefore, the template generator is currently not very useful. We will update this section as soon as these problems are solved.

7.3 Managing stacks

The **Stacks** button in the **Orchestration** tab takes you to the overview page where you can launch, suspend, resume and delete stacks.



The overview page contains a list of all currently existing stacks (either running or suspended), and buttons to perform the following actions:

Launch Stack starts a wizard to create a stack from a template. Depending on your choice of **Template Source**, you can provide a local file on your system, directly type (or paste) the template text, or enter a URL to download a template from that location. You can also immediately launch a stack from the template generator (see section 7.2) using the button **CREATE STACK**.

Preview Stack starts a similar wizard, but only performs a sanity check of your template, without instantiating the stack. If the check passes, you can inspect the parameters of the stack that would be created. The wizard does not allow you to enter input parameter values, so any mandatory input parameters should be provided in an environment (see ‘[Environments](#)’ in the Heat template guide).

Delete Stacks deletes the marked stacks. Beware that deleting a stack also deletes all of a stack’s physical resources, unless a different policy was set in the **deletion_policy** property for those resources (see the item ‘[Resources section](#)’ in the HOT specification).

More Actions hides the following additional actions:

Check Stacks verifies if the resources for selected stacks are still running.

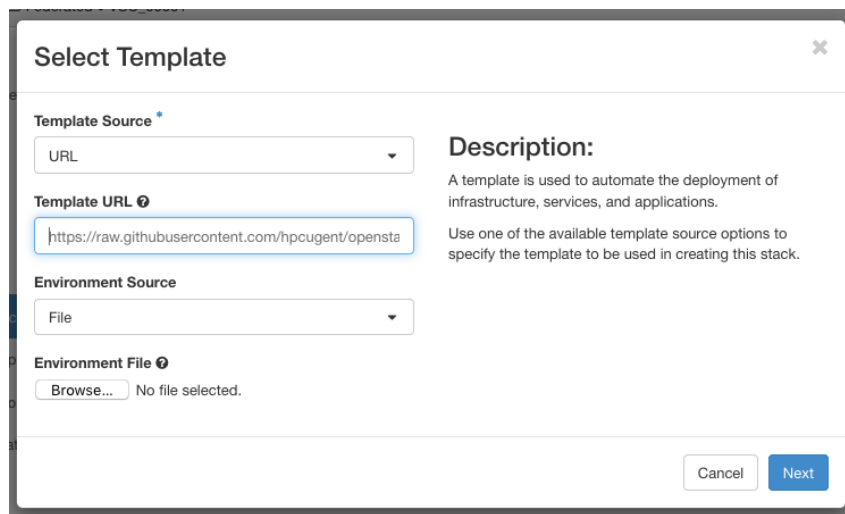
Suspend Stacks suspends all resources of the selected stacks.

Resume Stacks resumes the selected (suspended) stacks.

You can quickly suspend, resume or delete a single stack using the drop-down menu in the **Actions** column of the overview. This menu also contains the option **Change Stack Template**, which allows you to update a Stack by providing a new template.

Example: launching a stack

We can instantiate one of the examples from the VSC repository <https://github.com/hpcugent/openstack-templates> by providing a “raw” Github url:



Select Template

Template Source *

URL

Template URL ?

<https://raw.githubusercontent.com/hpcugent/opensta>

Environment Source

File

Environment File ?

Browse... No file selected.

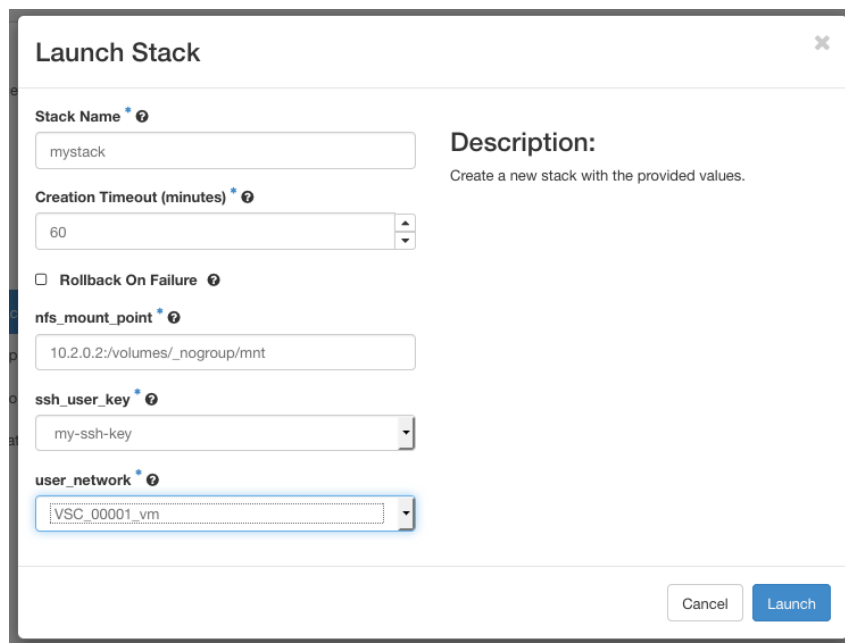
Description:

A template is used to automate the deployment of infrastructure, services, and applications.

Use one of the available template source options to specify the template to be used in creating this stack.

Cancel Next

If the template uses parameters, we can specify their values on the next page of the wizard. In our example, `nfs_mount_point`, `ssh_user_key` and `user_network` are the template parameters.



Launch Stack

Stack Name *

mystack

Creation Timeout (minutes) *

60

☐ Rollback On Failure ?

nfs_mount_point *

10.2.0.2:/volumes/_nogroup/mnt

ssh_user_key *

my-ssh-key

user_network *

VSC_00001_vm

Description:

Create a new stack with the provided values.

Cancel Launch

Finally, press the Launch button to create the stack.

Chapter 8

Shared file systems using Manila

OpenStack's Manila service makes it possible to create and manage shared NFS file systems for virtual machines. This service is not automatically enabled in the VSC cloud, so you should contact cloud@vscentrum.be if you want to use shared file systems in your project.

Creating a Shared File System

Creating a shared file system using the Horizon interface is quite straightforward:

1. Open the Share tab, and click Shares. A list of existing shares (if any) is shown.
2. Click the **Create Share** button to open the following dialog:

Fill out the following fields:

Share Name Choose a name.

Description Optionally, add a description.

Share Protocol Use the default NFS protocol.

Size (GiB) Set the size of the shared file system to be created. The total available storage and the amount currently used are shown on the right.

Share Type Here, you must select “cephfsnfstype” (the only choice).

Metadata You can attach additional metadata to your shared file system, which can be queried later on.

Other fields are not mandatory. By default, the shared file system will only be visible within the current project (Visibility: “private”). Be careful with the option “Make visible for all”: enabling it will set the visibility of your shared file system to “public”, making it visible for any other project in the VSC cloud as well.

3. Click **Create** to complete this step.

At this point, the shared file system exists within OpenStack, but it cannot be used until we define access rules for it.

Defining NFS access rules

You must define rules that define which machines on the network may obtain read or write access to your shared file system. By default, in absence of any rules, a shared file system cannot be accessed by anyone.

1. Open the drop-down menu in the **Actions** column for your share, and click **Manage Rules**.
2. You can now see all Share Rules for this shared file system. For a newly created file system, the list will be empty. Click **Add rule**.
3. Fill out the **Add Rule** dialog:

Access Type Only “ip” is supported.

Access Level Choose if you want to give read and write (“rw”) or read-only (“ro”) permission with this rule.

Access To Here, you can specify an ip address, or an address range, to which the rule applies. The addresses should be specified according to the format expected by an NFS exports configuration file. The following table contains a few examples, where it is assumed that the project’s `_nfs` network has the address range 10.10.x.0/24:

10.10.x.13	Allow this single ip address.
0.0.0.0/0	Allow any ip address.
10.10.x.0/24	Allow any ip address from the project’s <code>_nfs</code> network. For a non-public shared file system this has the same effect as the previous rule, because such a shared file system can only be accessed from within our project’s <code>_nfs</code> network anyway.
10.10.x.0/28	Allow addresses 10.10.x.0 until 10.10.x.15.

Click **Add** to add the rule.

Your rule now appears in the list. You can add as many rules as you wish, to set the access level for different addresses or address ranges.

Accessing a shared file system

When the proper access rules for the shared file system are in place, you can access it from an instance with a matching ip. In order to be able to mount the shared file system, your instance needs

- a NFS client, installed by default on images provided by the VSC cloud, and
- access to the `_nfs` network. Because your instance likely has to connect to the `_vm` network as well, your VM should have two NIC’s. Again, this is taken care of in the default images.

When you are ready to mount the network file system on an instance, look up the network location of your file system using the Dashboard:

1. Open the Share tab and click Shares. The list of all shared file systems in your project is shown.
2. Click the name of the shared file system you wish to access.
3. In the section “Share Overview”, look for the item **Export locations**.
4. Copy the content of the **Path:** field.

Once you know the location of your shared file system, you can mount it on any VM with the appropriate access rights, e.g. for a shared file system with location `10.2.0.2:/volumes/_nogroup/918...a78:`

```
$ sudo mount 10.2.0.2:/volumes/_nogroup/918..a78 /mnt
```