### 1 Overview of CloudVeneto
- 1.1 Projects ................................................. 3
- 1.2 Network access .......................................... 4
- 1.3 Getting help ............................................ 4
  - 1.3.1 Getting help for INFN Padova users .............. 5
- 1.4 Acknowledge CloudVeneto / Scientific citations ...... 5

### 2 Registration ........................................ 7
- 2.1 Apply for an account ................................... 7
  - 2.1.1 Enrollment procedure through INFN AAI or UniPD SSO 10
  - 2.1.2 Apply for an account using Username and Password 13
- 2.2 Apply for other projects ............................... 16
- 2.3 Manage project membership requests (only for project managers) 17
- 2.4 Administer project members (only for project managers) 18
- 2.5 Manage account renewals (only for project managers) 19
- 2.6 Expired users ......................................... 20

### 3 Getting Started .................................. 21
- 3.1 Access the Cloud through the Dashboard .......... 21
- 3.2 Creating a keypair .................................... 22
- 3.3 Importing your keypair ............................... 23
- 3.4 Setting security group(s) ............................ 24
- 3.5 Password management ............................... 25
  - 3.5.1 Foreword ......................................... 25
  - 3.5.2 Setting/changing password ...................... 26
- 3.6 Switching between projects ......................... 26
- 3.7 Accessing the Cloud with command line tools ........ 27
- 3.8 Accessing the Cloud through the euca2ools EC2 command line tools 30

### 4 Managing Virtual Machines ................... 35
- 4.1 Creating Virtual Machines .......................... 35
  - 4.1.1 Improve reliability: creating Virtual Machines from Volumes 40
- 4.2 Accessing Virtual Machines ........................ 42
- 4.3 Logging to a VM ...................................... 42
  - 4.3.1 Common access problems ...................... 42
- 4.4 Access a service running on the VM ............... 43
- 4.5 Copying files to a VM .............................. 45
4.6 Giving a VM public access (getting a floating IP) ................................................. 45
4.7 Creating accounts on your Virtual Machine .......................................................... 49
4.8 Accessing other hosts/services from Virtual Machines .......................................... 50
4.9 Flavors .................................................................................................................... 50
4.10 Stopping and Starting VMs ................................................................................... 51
4.11 Contextualisation .................................................................................................. 52
4.12 Resizing Virtual Machines .................................................................................... 53
4.13 Snapshotting Virtual Machines ............................................................................ 54
4.14 Deleting Virtual Machines ................................................................................... 55
4.15 Migrating an instance to another project/cloud .................................................... 55

5 Managing Containers ............................................................................................... 57
5.1 Running docker containers in the Cloud ................................................................. 57
5.2 Orchestrating containers ....................................................................................... 57

6 Managing GPUs ........................................................................................................ 59
6.1 Creating a GPU instance ......................................................................................... 59
6.1.1 Images for GPU instances ................................................................................ 60
6.2 Monitoring ............................................................................................................. 61
6.3 Policies .................................................................................................................. 61

7 Managing Storage ..................................................................................................... 63
7.1 Ephemeral storage .................................................................................................. 63
7.2 Volumes ................................................................................................................ 64
7.2.1 Create a Volume ............................................................................................... 64
7.2.2 Using (attaching) a Volume ............................................................................ 68
7.2.3 Detaching a Volume ........................................................................................ 70
7.2.4 Deleting a Volume ........................................................................................... 71
7.2.5 Sharing a volume between multiple (virtual) machines ....................................... 71
7.2.6 Transferring the ownership of a volume to another user ................................... 74
7.3 Object Storage (experimental) .............................................................................. 76
7.3.1 Accessing the object storage using the dashboard ............................................. 77
7.3.2 Accessing the object storage using the S3 interface .......................................... 78
7.4 Accessing storage external to the Cloud .................................................................. 79

8 Orchestration .............................................................................................................. 81
8.1 Creating a template ............................................................................................... 81
8.2 Creating a stack ..................................................................................................... 83

9 Managing Images ...................................................................................................... 85
9.1 Public Images ......................................................................................................... 85
9.1.1 Public Images for INFN Padova users ............................................................... 87
9.2 User Provided Images ............................................................................................ 87
9.3 Sharing Images ...................................................................................................... 88
9.4 Building Images .................................................................................................... 91
9.4.1 virt-builder ..................................................................................................... 91
9.5 Deleting Images .................................................................................................... 92
9.6 Migrating an image to another cloud .................................................................... 93

10 Creating Batch Clusters on the Cloud ................................................................... 95
10.1 Intro: the idea ...................................................................................................... 95
10.2 Creating Batch Clusters on the Cloud with HTCondor ........................................ 95
10.2.1 Prerequisites ................................................................................................. 96
10.2.2 The cluster configuration .............................................................................. 97
10.2.3 Start the elastic cluster ................................................................................ 97
Note: For any issues, questions, problems concerning this document please contact the CloudVeneto admins at support@cloudveneto.it

Contents:
CHAPTER 1

Overview of CloudVeneto

CloudVeneto is an OpenStack-based cloud.

It allows the instantiation of Virtual Machines (VMs) of the desired environments (in terms of operating system, installed software, etc.) and of the desired flavors (in terms of processors, memory size, etc.). It also provides storage volumes that can be attached to such virtual instances.

CloudVeneto also offers higher level services, such as the orchestration of multiple resources, the instantiation of elastic batch systems, the provision of Kubernetes clusters, etc.

Even if it is a single, logical Cloud service, its resources are spread in two different locations: Padova (INFN Padova - University of Padova’s “Dipartimento di Fisica e Astronomia”), and INFN Laboratori Nazionali di Legnaro (LNL).

The CloudVeneto is currently based on the Rocky version of the OpenStack middleware.

1.1 Projects

Projects (also known as tenants) are organizational units in the cloud. A project is used to map or isolate users and their resources.

Projects are given quotas on resource usage, in terms of virtual machines, cores, memory, storage, etc.

A project can have a single user as member (personal project) but the typical use case are shared projects, with multiple users as members, which can map to experiments, organizations, research groups, etc.

A user can be on multiple projects at the same time and switch between them.

In the CloudVeneto, projects usually map to experiments or other research groups. Among the project’s users there is a project manager (usually the team leader) who is responsible to manage (accepting or refusing) membership requests for the project.

Warning: Personal private projects are discouraged and are created only for convincing reasons.
1.2 Network access

Cloud instances are by default “visible” from the Local Area Networks (LANs) of both INFN Padova/Unipd Physics Dept. and INFN-LNL. This means that e.g. users can access via ssh VMs of the Cloud directly from their desktops located in INFN Padova/Physics Dept. or INFN Legnaro. It is then possible to control which services/ports can be accessed using the security groups (discussed later) and firewalls on the relevant VMs.

If it is necessary to log on a VM of the Cloud from a location different than INFN Padova/Physics Dept. and INFN Legnaro, it is necessary to go through a gate machine. A Cloud specific gate host (gate.cloudveneto.it) can be used.

If needed, e.g. if a VM should host a service accessible from the Internet, such VM on the Cloud can be given a public IP. If this is the case please contact support@cloudveneto.it.

From a VM of the Cloud, it is possible to access the Internet, while by default it is not possible to access a host/service hosted in the INFN Padova or Legnaro LANs. If, for some reasons, you need to access some services hosted in INFN Padova or Legnaro from the Cloud, please contact support@cloudveneto.it.

1.3 Getting help

In case of problems with the CloudVeneto infrastructure, such as:

- Problems during the registration process
- Problems creating a virtual machine
- Problems creating a volume, or attaching it to an instance

you can contact the admins at support@cloudveneto.it.

Please provide all the information needed to debug the problem, e.g.:

- Your cloud username
- The name of the project
- The IP number of the instance (in case of problems with a virtual machine)
- The image name and the flavor name (in case of problems creating an instance)

The CloudVeneto admins instead don’t provide support to the virtual machines instantiated on the Cloud: once you created a server on the Cloud, such server is managed by you.

This implies that the CloudVeneto support crew doesn’t provide support on topics like:

- How to install/compile/configure your software;
- ssh / scp basic usage;
- Basic linux usage (some documentation is available on Some basics on Linux administration);
- Accessing your VM ‘the graphical way’.

You might ask your Department / Institution technicians in case of problems with your virtual machine that you are not able to solve on your own.

Experiences, problems, best practices, etc. can be shared with the other users of the CloudVeneto using the discuss@cloudveneto.it mailing list. By default all CloudVeneto users are member of this mailing list. If you want to be removed from this mailing list please send an e-mail to majordomo@pd.infn.it, writing in the body of the mail:

unsubscribe discuss <your-email-address>
Changes and planned interventions to the service will be posted on the announce@cloudveneto.it. All registered users to the Cloud are member of this mailing list.

### 1.3.1 Getting help for INFN Padova users

INFN-Padova computing and Network service can provide support to INFN-Padova users only for instances created using the INFN-Padova “blessed” images, described in *Public Images for INFN Padova users.*

When contacting the INFN-Padova computing and Network service to have support with a virtual machine, please provide all the information needed to debug the problem, in particular:

- The IP number of the instance
- The image name
- The flavor name

### 1.4 Acknowledge CloudVeneto / Scientific citations

We kindly ask you to acknowledge the usage of the CloudVeneto infrastructure in any scientific publication or elsewhere. The following quote can be used:

> CloudVeneto is acknowledged for the use of computing and storage facilities.

References:

- Cloudveneto web site: http://cloudveneto.it
To be able to use the CloudVeneto service, first of all you need to apply for an account. The procedure to be followed is described in this chapter.

2.1 Apply for an account

The registration procedure in the cloud is managed through the Horizon Openstack web service interface.

Go to https://cloud-areapd.pd.infn.it/dashboard or https://cloudveneto.ict.unipd.it/dashboard in a browser. The following page should appear:
Click on the **Register** button.

The following page should appear:
• If you have an account on the INFN Authentication and Authorization Infrastructure (INFN AAI) and therefore you have access to the INFN portal, click on the INFN AAI logo and proceed with Enrollment procedure through INFN AAI or UniPD SSO.

• If you instead have an account on the UniPD Single Sign-On (SSO) Infrastructure (e.g. you have a username @unipd.it or @studenti.unipd.it), click on the UniPD logo and proceed with Enrollment procedure through INFN AAI or UniPD SSO.

• If and only if you don’t have neither an account on the INFN Authentication and Authorization Infrastructure (INFN AAI) nor on the UniPD SSO, click on the last icon (the green one, with “Username” and “Password”) and proceed with Apply for an account using User and Password.
2.1.1 Enrollment procedure through INFN AAI or UniPD SSO

Once authenticated on your Identity Provider system, you will be redirected to a form like this:
Fill the form with the required information. In particular please specify the relevant unit/department of your home

2.1. Apply for an account
institution.

For what concerns the **Project Action** (projects have been discussed in *Projects*) you have two options:

- Select Existing Projects
- Create new project

Choose **Select Existing Projects** if you want to apply membership for one or more existing projects (choose them in the relevant box).

Select **Create new project** if instead you want to ask the creation of a new project (and you are the leader of the experiment/research group associated to this project). In this case you will have to specify also a Project name and a Project Description.

You will also have to specify if this project must be private (a personal project where you will be the only member) or not.

Newly created projects will get a default quota of:

- 20 VCPU
- 20 GB RAM
- 200 GB for Volume (e.g. non ephemeral) storage

Allocation of more resources for UniPD users must be approved by the Cloud UniPD “governance”. Mail your request to cloud-unipd-gov@lists.pd.infn.it if you want more resources to be allocated to your project.

**Note:** Public (i.e. not private) projects are projects where other users can apply for membership. They are supposed to be used for experiments or other research groups.

**Personal private projects are discouraged** and are created only for convincing reasons.

**Note:** The person who asks for the creation of a new project is automatically defined as the manager of this project, i.e. he/she will have to manage the membership requests for this project. So the request to create a new project should be done by the relevant experiment/group leader.

When you have filled the form, please read the AUP that you need to accept (by clicking the **Accept AUP** button). Finally click on the **Register** button and you are done.
Your request will be managed by the Cloud administrator and by the manager(s) of the project(s) for which you applied membership. You will get an e-mail when your request is approved or if for some reason your request is refused. In the first case you will also receive an e-mail for the credentials needed to access the Cloud gate machine (see Network Access).

### 2.1.2 Apply for an account using Username and Password

If and only if you don’t have an account on the UniPD SSO or INFN AAI, click on the green icon on the right:

![Registration done](image)

You will receive a notification with the result of the registration process.

A form such as the one of the following image will appear.
Chapter 2. Registration

Please fill in the form with your personal data (First Name, Last Name, Email Address). Choose a User name (please
note that it could be changed by the Cloud admins during the registration process) and a Password. Specify your home institution and the relevant unit/department.

For what concerns the **Project Action** (projects have been discussed in *Projects*) you have two options:

- Select Existing Projects
- Create new project

Choose **Select Existing Projects** if you want to apply membership for one or more existing projects (choose them in the relevant box).

Select **Create new project** if instead you want to ask the creation of a new project (and you are the leader of the experiment/research group associated to this project). In this case you will have to specify also a Project name and a Project Description.

You will also have to specify if this project must be private (a personal project where you will be the only member) or not.

Newly created projects will get a default quota of:

- 20 VCPU
- 20 GB RAM
- 200 GB for Volume (e.g. non ephemeral) storage

Allocation of more resources for UniPD users must be approved by the Cloud UniPD “governance”. Mail your request to cloud-unipd-gov@lists.pd.infn.it if you want more resources to be allocated to your project.

---

**Note:** Public (i.e. not private) projects are projects where other users can apply for membership. They are supposed to be used for experiments or other research groups.

**Personal private projects are discouraged** and are created only for convincing reasons.

---

**Note:** The person who asks for the creation of a new project is automatically defined as the manager of this project, i.e. he/she will have to manage the membership requests for this project. So the request to create a new project should be done by the relevant experiment/group leader.

---

When you have filled the form, please read the AUP that you need to accept (by clicking the **Accept AUP** button).

Finally click on the **Register** button and you are done.
Your request will be managed by the Cloud administrator and by the manager(s) of the project(s) for which you applied membership. You will get an e-mail when your request is approved (and therefore you can start using the CloudVeneto) or if for some reason your request is refused.

### 2.2 Apply for other projects

After you have been given an account on the CloudVeneto, at any time you can ask the creation of a new project or the membership to an already existing project.

Both operations are performed by accessing the **Identity → Projects** tab of the OpenStack dashboard and clicking on **Subscribe to project** as depicted on the following image:
Fulfill your request selecting the relevant choice under the **Project action** dropdown list.

### 2.3 Manage project membership requests (only for project managers)

If you are the manager of a project, you will receive membership requests for this project that you will have to manage (approving or refusing them).

When a user applies to be member of a project that you manage, you will receive an e-mail such as this one:

To manage such requests, open the OpenStack web dashboard, i.e. go to https://cloud-areapd.pd.infn.it or https://cloudveneto.ict.unipd.it in a browser. Log in, and then access **Identity → Subscriptions**. An image such as the following one, with the list of the pending requests, will appear.
To approve a membership request, click on the Approve button (in Actions). A window such as the following one will appear:

Set the expiration date of the account, and click on the Ok button to approve the request.

If, instead, you want to reject the request, select Reject in Actions.

**Note:** It is therefore up to the project manager to set the expiration date of the members of his/her group. A user belonging to multiple projects can have different expiration dates for the different projects he/she belongs to.

### 2.4 Administer project members (only for project managers)

If you are the manager of a project, you can list the members of your project and, if needed, change their role.

Open the OpenStack web dashboard, i.e. go to https://cloudveneto.ict.unipd.it/dashboard in a browser. Log in using the relevant method and access the Identity → Project Members panel. The list of users affiliated to your project
will appear:

![CloudVeneto User Guide, Release 3.11](image)

From here you can also change the role of a specific user (by clicking on **Toggle Role**) from ‘Project User’ to ‘Project manager’ or viceversa.

**Note:** If a user is promoted to Project manager, she will then be allowed to manage affiliation requests to the project, as described in *Manage project membership requests (only for project managers)*.

From this window you can also remove a specific user from the project you manage.

### 2.5 Manage account renewals (only for project managers)

When the affiliation of a user for a project is expiring, as manager of that project you will receive an e-mail such as this one:

![User rubbia requires renewal](image)

To manage such requests, open the OpenStack web dashboard, i.e. go to [https://cloud-areapd.pd.infn.it](https://cloud-areapd.pd.infn.it) or [https://cloudveneto.ict.unipd.it](https://cloudveneto.ict.unipd.it) in a browser. Log in, and then access **Identity → Subscriptions**. An image such as the
following one will appear:

Click on the Renew button (in Actions). A window, such as the one represented in the following image will appear:

Set the new expiration date and then click the OK button.

### 2.6 Expired users

Please note the resources (instances and volumes) owned by expired users (i.e. people who don’t have anymore an account on CloudVeneto) will be removed.

Please also note that, while the ownership of a volume can be transferred from a user to another one (see *Transferring the ownership of a volume to another user*), this can’t be technically possible for virtual machines.
3.1 Access the Cloud through the Dashboard

Once you have been given an account, you can access the functionality provided by the Cloud. There are several ways of interacting with the Cloud. The simplest one is the dashboard, a web based GUI.

To access the production service of CloudVeneto via the dashboard, you must simply go to https://cloudveneto.ict.unipd.it/dashboard/ or https://cloud-areapd.pd.infn.it/dashboard/ in a browser.
You can now log either using the INFN-AAI credentials, the University of Padova Single Sign On (SSO) system, or using the username and password.

### 3.2 Creating a keypair

You can now proceed creating a key-pair. This is a secret key which will allow you to interact with your virtual machine once it is created. This key should be handled with similar security to a password or an ssh key so it should only be stored in a secure directory such as a private area in your home folder.

The steps are as follows:

- Open the **Compute** tab on the left side
- Select **Key Pairs**
• In the Key Pairs section, select **Create Key Pair**.

You will need to give the keypair a name, such as `my_key`.

On completion of the operation, a file `my_key.pem` will be downloaded to your computer.

**Warning:** Be careful not to lose the file you just downloaded since there is no easy way to download it again.

This file should be stored in a safe location. To keep it private, run:

```bash
chmod 600 my_key.pem
```

### 3.3 Importing your keypair

You might already have an ssh key you use to remotely access machines. This means you already have under the `.ssh` directory in your home folder a couple of files named `id_rsa` (or `id_dsa`) and `id_rsa.pub` (or `id_dsa.pub`). If, on the machine you want to log on, your `id_rsa.pub` has been authorized, you can access the machine without providing a password.

Importing your public key allows you to ‘inject’ it on any newly created VM in your project.

The steps are as follows:

• Open the **Compute** tab on the left side

• Select **Key Pairs**

• In the **Key Pairs** section, select **Import Public Key**.

You will need to give the keypair a name (your full username is a good choice), e.g. `paolomazzon`.

On the “Public Key” field paste the content of your `id_rsa.pub` file.
Finally click on the **Import Public Key** button

**Warning:** Be careful not to paste the content of your private key, the one without the `.pub` extension.

You can now use your key pair when instantiating a VM as an access method.

### 3.4 Setting security group(s)

Security groups are sets of IP rules (firewall) that define networking access and are applied to all instances within a project using that group. As described in *Creating Virtual Machines*, when you create an instance you have to specify the security group to be used.

To set such IP rules, users can either add them to the *default* security group or can create a new security group with the desired rules.

For example the following procedure enables SSH and ICMP (ping) access to the *default* security group. The rules apply to all instances within a given project using this security group, and should be set (just once) for every project, unless there is a reason to inhibit SSH or ICMP access to the instances.
This procedure can be adjusted as necessary to add additional security group rules to a project, if needed.

- Log in to the dashboard, choose a project, and click **Network → Security Groups**. The security groups that are available for this project are shown.
- Select the *default* security group and click **Manage Rules**.
- To allow SSH access, click **+ Add Rule**.
- In the *Add Rule* dialog box, enter the following values:

<table>
<thead>
<tr>
<th>Rule</th>
<th>SSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote</td>
<td>CIDR</td>
</tr>
<tr>
<td>CIDR</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>

**Note:** To accept requests from a particular range of IP addresses, specify the IP address block in the **CIDR** box.

- Click **Add**.
- To add an ICMP rule, click **Add Rule**.
- In the *Add Rule* dialog box, enter the following values:

<table>
<thead>
<tr>
<th>Rule</th>
<th>All ICMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction</td>
<td>Ingress</td>
</tr>
<tr>
<td>Remote</td>
<td>CIDR</td>
</tr>
<tr>
<td>CIDR</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>

- Click **Add**.

**Warning:** If you need to enable some services on a Virtual Machine, besides setting the specific IP rules through security groups, be sure that the relevant ports are also enabled (e.g. via firewall) on the Virtual Machine.

### 3.5 Password management

#### 3.5.1 Foreword

**Warning:** This procedure is NOT the one to be used to change the UniPD SSO or INFN AAI password and is NOT the one to be used to change the gate password!

You need to use the procedure described in the following subsection **only if**:

- You access the cloud through username and password and you want to set/change that password;
- You want to use the cloud infrastructure through the command line (see *Accessing the Cloud with command line tools*).
3.5.2 Setting/changing password

From the OpenStack dashboard click on your user’s name (on the top), select Settings from the dropdown menu and then Manage Password.

**Important:** Once again: this is the password to authenticate with the Cloud “internal” authentication mechanism. It is uncorrelated from the UniPD SSO or INFN-AAI one and with the access to the gate host.

![Activate Password](image)

3.6 Switching between projects

As introduced in Projects, a user can be on multiple projects at the same time. The current project is indicated by the top left dropdown menu near the logo.

To switch between projects just open the dropdown menu (as shown in the following figure) and select one of your available projects.
3.7 Accessing the Cloud with command line tools

It is possible to manage the Cloud using command line tools, even if most of the functionality provided by the Cloud can be accessed through the dashboard web interface. The documentation on the OpenStack site contains extended information on the syntax and installation procedure of the command line tools.

**Note:** INFN Padova users can find the OpenStack client installed on *lx.pd.infn.it*.

**Important:** Command line tools can only be used with the Cloud “internal” authentication mechanism. Even if you normally access the cloud dashboard using the UniPD SSO or INFN-AAI it’s now time to set a password.

The OpenStack tools require a set of shell environment variables in order to run. These variables can be obtained from the dashboard and then stored in an ‘rc’ file that you can source (much like your `.profile` when logging into a linux server).

The environment variables are different for projects you work on.

If you log into the dashboard, you will find **API Access** under the **Project** menu on the left hand side.
Select **Download OpenStack RC file** and then **OpenStack RC file (Identity API v3)**, to download the rc file for your current project. Please note that the v3 openrc file requires a quite recent version of the Openstack client.
This file is different for each of the projects you are working on.

The downloaded rc file should be saved onto the machine you want to run the commands from. If you use csh rather than bash/zsh for your shell, you would need to create a new version using `setenv` rather than `export`.

Since the CloudVeneto services are secured using SSL, you will need the `Digicert.pem` “certification authority” file. This file can be downloaded from here.

Once you get the file you need to edit the RC file to set the `OS_CACERT` variable like this:

```
export OS_CACERT=/etc/grid-security/certificates/Digicert.pem
```

**Note:** The certificate can be put anywhere on the client as long as the path you specify is consistent.

To test it works, source the rc script file and enter your password to authenticate. The OpenStack command line tools can then be used, e.g.:

```
$ . SgaraPrj1-openrc.sh
Please enter your OpenStack Password for project SgaraPrj1 as user sgaravat@infn.it:

+--------------------------------------+--------------+--------+----------------------
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Status</th>
<th>Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>89088351-90d8-4346-8ecf-ad08750b9d9a</td>
<td>tinies-uno-5</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.</td>
</tr>
</tbody>
</table>
```

(continues on next page)
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>44d12ad6-cc7e-47c3-a6d5-5e2b7c32d542</td>
<td>tinies-uno-4</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.17</td>
</tr>
<tr>
<td>02a40340-d238-4405-b5f3-3d38d9f9b485</td>
<td>tinies-uno-3</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.12</td>
</tr>
<tr>
<td>bc6098c7-6ec6-4ac1-8aee-9e1edeb33836a</td>
<td>tinies-uno-2</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.6</td>
</tr>
<tr>
<td>1a1ab1b8-a3dd-401d-a8c2-cbd30b02e066</td>
<td>tinies-uno-1</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.10</td>
</tr>
<tr>
<td>4e2c6cc7-bc1c-951e-1eb567247588</td>
<td>dasgara1-2</td>
<td>ACTIVE</td>
<td>SgaraPrj1-lan=10.1.1.11</td>
</tr>
</tbody>
</table>

Note: When you source the rc script you are asked for a password. If the password is wrong, you will be told (with a generic authentication error) only when you issue some OpenStack commands.

### 3.8 Accessing the Cloud through the euca2ools EC2 command line tools

The CloudVeneto also exposes a AWS EC2 compatible interface, which is one of the de-facto standard for computational clouds.

The *euca2ools* are command line tools that can be used to interact with an EC2 based cloud.

You can install the *euca2ools* package on your desktop as follows:

**CentOS / Fedora**

```
# yum install euca2ools
```

**Ubuntu / Debian**

```
# apt-get install euca2ools
```

Note: INFN-Padova users can find the euca2ools installed on `lx.pd.infn.it`.

The euca2ools require a set of shell environment variables in order to run. These environment variables are different per project that you work on.

If you log into the dashboard, you will find **API Access** under the **Project** menu on the left hand side.
Select **Download OpenStack RC file** and then **EC2 Credentials** to download the zip file for your current project. This zip file will be downloaded from the browser.

### 3.8. Accessing the Cloud through the euca2ools EC2 command line tools
This file should be saved onto the machine where you want to run the commands from, and unzipped into a private directory, e.g:

```
$ unzip SgaraPrj1-x509.zip
Archive: SgaraPrj1-x509.zip
  extracting: ec2rc.sh
```

`ec2rc.sh` gives the variables for accessing the Cloud with EC2 APIs. If you use a C shell based shell, you would need to adapt this using setenv.

To test it, you can e.g. try the following:

```
$ . ec2rc.sh
$ euca-describe-instances -I ${EC2_ACCESS_KEY} -S ${EC2_SECRET_KEY} -U ${EC2_URL}
```

---

This content is from the CloudVeneto User Guide, Release 3.11.
Warning: For some euca2ools distributions sourcing the ec2rc.sh script is not enough. You need to explicitly specify access and secret keys and the endpoint with the relevant command line options, e.g.:

```
$ euca-describe-instances -I ${EC2_ACCESS_KEY} -S ${EC2_SECRET_KEY} -U ${EC2_URL}
```
Managing Virtual Machines

**Important:** Virtual machines, even if idle or in shutdown state, allocate resources which therefore aren’t available to other users. Therefore please delete the virtual instances that you don’t plan to use anymore in the near future. Cloud administrators perform regular campaigns to identify and possibly delete unused instances.

### 4.1 Creating Virtual Machines

To create a Virtual Machine (VM) using the dashboard, you need to have already logged into the dashboard, created your private key (as explained in *Creating a keypair*) and set the security group (as discussed in *Setting security group(s)*) to be used for this VM.

To create a VM proceed as follows:

- Be sure you have selected the right Project from the dropdown menu on the top.

- Go to **Compute → Instances** on the left hand menu. This will display a list of VMs currently running in your project.
• Select the **Launch Instance** button. A new window appears.
Here you can enter:

- **Instance name** is the name of the machine you want to create.

- **Flavor** is the size of the machine you want to create. These are specified using VCPUs (number of virtual CPUs), disk space for the system disk, size for the RAM memory. You are advised to start small (the flavor of a virtual machine can be changed later if required). Flavors are discussed in *Flavors*.

- **Instance Count** is the number of virtual machines to be started.

- As **Instance Boot Source** select **Boot from Image** or **Boot from Snapshot** and then specify its name.

- Switch to the **Access & Security tab**.
- As **Keypair** select the keypair you created. This will allow you to log to the VM (usually as root or as an account where you can get admin privileges via sudo) using this SSH key.

- You can also specify the admin (usually root) password of the instance.

  **Warning:** Please note that setting the admin password is not guaranteed to always work (the image can’t support the “injection” of this password). It is therefore strongly suggested to rely on the ssh-key mechanisms.

- Specify the Security group to be used for this VM (security groups are discussed in *Setting security group(s)*).

  - Now switch to the **Networking** tab.
You should see one network called `<ProjectName>-lan`

**Note:** INFN users could see, besides the `<ProjectName>-lan` network, also a network called `<ProjectName>-wan`, if the possibility to use public IP numbers was requested. The former one must be selected if the VM doesn’t need to be visible on the Internet. The `<ProjectName>-wan` network must be selected if instead the VM must have a public IP. It will then be necessary to allocate a public (floating) IP address to this instance, as explained in *Giving a VM public access (getting a floating IP)*.

- Select **Launch** to start the virtual machine being created. You will be returned to the Overview screen, where there will be a line with the instance name, ip adress and status. The status should be ‘Active’ once the install is complete.

Once the status of the machine is ‘Active’, you can watch the console to see it installing and booting. You can click on the VM name and go to a dedicated window or from this same table you can access a pull down menu on the right hand side under **Actions**. There you will see various options and among them **View Log** and **Console**.
For a Linux machine, select **Console** to access to the console of the VM.

**Note:** Virtual Machines instantiated on the Cloud by default aren’t registered in the DNS. This means that you’ll have to refer to them using their IP numbers.

For Virtual Machines supposed to have a long life, INFN Padova users may ask (contacting supporto@pd.infn.it) to have them registered in the DNS. If possible (i.e. if the chosen names are sensible enough and if there are no ambiguities) the registered names in the DNS will be the same as the ones chosen as Instance names.

### 4.1.1 Improve reliability: creating Virtual Machines from Volumes

By default Virtual Machines are instantiated using the local disk of the Cloud compute node. This means that, in case of failure of the compute node, it may happen that the virtual machine content is lost.

For production servers which are not fully redundant and load balanced, to improve the availability it is advisable to use an external storage volume for the system disk of the virtual machine. The advantage is also that, if the compute node hosting the virtual machine has to be switched off e.g. for maintenance, the Cloud administrator before doing this operation can live-migrate the instance to another Cloud compute node basically without any service interruption.

On the other hand, I/O performance is usually better when the instance is created using the local disk of the compute
node with respect to a virtual machine created from volume.

To create a VM from volume, in the **Launch Instance** tab select **Boot from image (create a new volume)** for the **Instance Boot Source** field.

Select the image to be used (**Image Name** field). Then specify the desired size (**Device size (GB)**).

Then proceed as discussed above.

**Note:** Unipd Since volumes for VMs are created on the ceph backend, users will have to first contact support@cloudeneto.it to require some disk space on the ceph storage system. In fact by default they are given no disk space on such volume backend.
4.2 Accessing Virtual Machines

**Warning:** Please note that by default cloud VMs are not registered in the DNS, and therefore you must use their IP address.

### 4.3 Logging to a VM

Virtual machines created on the cloud have their IP assigned on a private network associated with the project they belong to. Therefore they cannot be accessed directly from the internet.

If you need to log on your VMs from the Internet you must go through a gate machine: `gate.cloudveneto.it`.

When your account on the cloud is created you also got access to the gate. Contact support@cloudveneto.it in case of problems with this account.

**Note:** Projects that get their private network on the subnet 10.64.0.0/16 are peculiar: the VMs can be accessed from the INFN-Padova or INFN-Legnaro Local Area Networks (LANs). Those projects are created mostly for INFN users.

Assuming that

- You created a Linux virtual machine using the `<ProjectName>-lan` network;
- Your VM got the 10.67.15.3 IP address;
- You know the default user on the VM. If, as an example, the VM is Ubuntu based there is a default user called 'ubuntu';
- You stored your `my_key` keypair in `~/private` on the gate machine;

you can access your VM from the gate machine issuing

```
ssh -i ~/private/my_key ubuntu@10.67.15.3
```

Conversely, there is no limitation on the ‘outer’ services you can reach from your VM (modulo the services hosted in the INFN Padova/Legnaro LANs, as described in Accessing other hosts/services from Virtual Machines.

#### 4.3.1 Common access problems

You might receive an error upon logging to a VM. Two of the most common are:

**PROBLEM SYMPTOM:**

```
@ WARNING: UNPROTECTED PRIVATE KEY FILE! @
Permissions 0644 for '/home/user/my_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/user/my_key.pem": bad permissions
ubuntu@10.64.14.12: Permission denied (publickey).
```

**SOLUTION:** you forgot to change the permission on your private key (see Creating a keypair). Please run:
and try again.

- **PROBLEM SYMPTOM:**

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The green part is your side of the connection (e.g. the port opened on your PC);
user@gate... is the transport part of the connection;
10.X.Y.Z:80 in red is the remote end of the connection.

The blu arrows depict the data flow.

Once you have opened the tunnel you have, on your machine, a direct entry point to your VM (port 2080 in this case).

From a browser on your local machine, you can now access the web service running on the VM at the url

```
http://localhost:2080
```

• USE CASE 2: you want to access both the web service on port 80 and the ssh service (port 22) on your VM. Proceed as follows:
  – Choose two free TCP port on your machine in the range 1025-65535 (e.g. 2080 and 22);
  – Set up the tunnel using your access to one of the gate machines of the cloud (e.g. gate.cloudveneto.it) and providing your TCP ports of choice and the IP address of your remote VM:

```
ssh -L 2080:10.X.Y.Z:80 -L 2022:10.X.Y.Z:22 user@gate.cloudveneto.it
```

The tunnel needs to stay open as long as you need to access the VM.

From another terminal of your local machine, you can now:
  – access the VM using ssh with the -p (lowercase ‘p’) parameter:

```
ssh -p 2022 -i ~/.private/my_key remoteuser@localhost
```

  (e.g.: ssh -p 2022 -i ~/paolo.pem ubuntu@localhost)

  – copy a file on the VM using scp with the -P (capital ‘p’) parameter

```
scp -P 2022 -i ~/.private/my_key my_local_file.txt remoteuser@localhost:/remote/path/
```

  (e.g. scp -P 2022 -i ~/paolo.pem my_local_file.txt ubuntu@localhost:/tmp/)

  – from your web browser you can access the web service on the VM by accessing the url
If `sshfs` is installed, you can use it with the tunnel with the following command:

```bash
sshfs -p 2022 remoteuser@localhost:/remote/path /local/path -o IdentityFile=~/.private/my_key
```

It is a file system client that mounts the remote file system locally.

**Note:** Tip: if the last part of your VM IP is Z, choosing 2000+Z as the local TCP port is a good way to memorize the (local port → remote VM) association of your tunnel.

### 4.5 Copying files to a VM

Copying files to your VM might be a little more complex since VMs don’t have an access facing the internet. Your options are:

- Access the VM directly if you are using the INFN network;
- Use one of the gate machines and make it in a two step fashion:
  - Copy your file from your machine to the gate;
  - Copy file from the gate machine to your VM;
- Exploit the port forwarding mechanism explained above to access port 22 of your VM from your PC.

### 4.6 Giving a VM public access (getting a floating IP)

If needed, e.g. if a VM should host a service accessible from the Internet, such VM on the Cloud can be given a public IP. For this purpose you will need:

- to instantiate the VM, as explained in *Creating Virtual Machines*;
- to allocate a floating (public) IP;
- to associate the allocated floating IP to the relevant VM.

Floating IP addresses can have their associations modified at any time, regardless of the state of the instances involved. The following procedure details the reservation of a floating IP address from an existing pool of public addresses and the association of that address to a specific instance.

- From the dashboard click **Floating IPs** on the **Network** menu.
• Click Allocate IP To Project.

Choose the pool from which to pick the IP address. You must choose:

- **INFN-WAN**, if you are a INFN user (i.e. if the IP address of your instance is 10.63.x.y);
- **Unipd-WAN**, if you are referring to a University of Padova project (i.e. if the IP address of your instances is 10.67.x.y);
- **CloudVeneto-WAN**, in all other cases (i.e. if the IP address of your instances is 10.68.x.y).

Then click Allocate IP.

• Click on Associate for the just allocated floating IP.
In the Manage Floating IP Associations dialog box, choose the following options:

- The IP Address field is filled automatically, but you can add a new IP address by clicking the + button.
- In the Ports to be associated field, select a port from the list (the list shows all the instances with their fixed IP addresses).
Finally click **Associate**.

To disassociate an IP address from an instance, click the **Disassociate** button.

To release the floating IP address back into the pool of addresses, click the **Actions** button and select the **Release Floating IP** option.
Note: By default the possibility to use public IP numbers is disabled and therefore by default it is not possible to allocate a floating IP to an instance. If public IPs are needed for your project, please contact support@cloudveneto.it specifying what is/are the relevant service(s) and the port(s) that need to be open.

Warning: Instances with public floating IPs are regularly scanned to be sure they don’t expose vulnerable services. In case you are contacted by the CloudVeneto support team about a security problem, you need to promptly fix it. If this is not done, the use of the floating IP will be revoked.

Important: INFN users will have to create the VM on the <ProjectName>-wan network, if the VM must be given a public IP.

To control which services/ports of your virtual machine can be accessed, be sure you are using the right security group (as discussed in Setting security group(s)) and you have correctly configured firewall (iptables, firewalld etc.) on the relevant VM.

4.7 Creating accounts on your Virtual Machine

Important: Please remember, that, as stated in the AUP that you accepted when you applied for an account on CloudVeneto, you can allow access to the Virtual Machines (VMs) only and exclusively to people who have already an account on this CloudVeneto service, or have an account at INFN Padova, or have an account at INFN Laboratori Nazionali di Legnaro.

If you need to create accounts on the virtual machine, please see Adding a user to your VM.

If you are an INFN user, and you are using a SL6x-INFNPadova-x86-64-<date> or CentOS7x-INFNPadova-x86-64-<date> image, please see Public Images for INFN Padova users.
Important: Please also note that, as stated in the AUP, you will be held responsible for all the activities performed on the VMs created by yourself.

### 4.8 Accessing other hosts/services from Virtual Machines

There is no limitation on the ‘outer’ services you can reach from your VM. However by default it is not possible to access a host/service hosted in Padova or Legnaro.

If, for some reasons, you need to access some services hosted in Padova or Legnaro from the Cloud, please contact support@cloudveneto.it.

### 4.9 Flavors

As shown in *Creating Virtual Machines*, when an instance has to be created it is necessary to specify the flavor to be used for this VM.

Flavors define the virtual machine size such as:

- Number of virtual CPU cores (VCPUs)
- Amount of memory
- Disk space

Information about the flavors can be seen in the **Flavor Details** box that appears in the Dashboard when you launch a new instance.

<table>
<thead>
<tr>
<th>Flavor Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>VCPUs</td>
</tr>
<tr>
<td>Root Disk</td>
</tr>
<tr>
<td>Ephemeral Disk</td>
</tr>
<tr>
<td>Total Disk</td>
</tr>
<tr>
<td>RAM</td>
</tr>
</tbody>
</table>

‘Root Disk’ is the size of the root disk. This is ephemeral storage, i.e. such storage is deleted when the relevant instance is deleted (see Ephemeral Storage).

‘Ephemeral Disk’ is the size of the supplementary ephemeral disk.
Warning: For what concerns VCPUs, please note that the CloudVeneto is configured to allow some “overbooking” so that usually a physical core is mapped to 4 VCPUs.

Note: If you find that a specific flavor you require is not available, please contact support@cloudveneto.it.

4.10 Stopping and Starting VMs

VMs can be stopped and started in different ways available from the Actions menu of every instance found on the (Compute → Instances) table.

Warning: The cleanest way to shutdown (or reboot) an instance is however to log on the VM and issue from the shell the shutdown or reboot command. In fact if the Soft Reboot Instance or Hard Reboot Instance or Shut Off Instance actions are chosen, there could be problems with networking when the VM is later restarted.
Pause Instance/Unpause Instance allows for temporary suspension of the VM. The VM is kept in memory but it is not allocated any CPU time.

Suspend Instance/Resume Instance stores the VM onto disk and recovers it later. This is faster than stop/start and the VM returns to the status is was when the suspend was performed as opposed to a new boot cycle.

4.11 Contextualisation

Contextualisation is the process to configure a virtual machine after it has been installed. Typical examples would be to create additional users, install software packages or call a configuration management system. These steps can be used to take a reference image and customize it further. Contextualisation is only run once when the VM is created.

Most of the available public images include a contextualisation feature using the open source cloud-init package.

With cloud-init, data to be used for contextualisation are called user data.

Using the Openstack command line tool, the --user-data option of the nova boot command must be used, e.g.:

```
nova boot my_vm --image "SL65-Padova-x86_64-20141023-QCOW2" \
   --flavor m1.xsmall --user-data my_data.txt --key_name my_key
```

For example to run a command during contextualisation, the #cloud-config directive can be used:

```
cat > cern-config-users.txt << EOF
#cloud-config
runcmd:
  - [ /usr/bin/yum, "install", -y, "cern-config-users" ]
  - [ /usr/sbin/cern-config-users, --setup-all ]
EOF
```

User data can be provided as a gzip file if needed where the user data is larger than 16384 bytes, e.g.:

```
cat > userdata4zip.txt <<EOF
#!/bin/sh
wget -O /tmp/geolist.txt http://frontier.cern.ch/geolist.txt
EOF
gzip -c userdata4zip.txt > userdata4zip.gz
```

```
nova boot my_server --image "SL65-Padova-x86_64-20141023-QCOW2" \
   --flavor m1.xsmall --user-data userdata4zip.gz --key_name my_key
```

With the #include or Content-Type: text/x-include-url directives, it is possible to specify a list of URLs, one url per line. The userdata passed by the urls can be plain txt, gzip file or mime-multi-part script. E.g.:

```
cat userdata.txt <<EOF
#!/bin/bash
wget -O /tmp/robots.txt http://www.ubuntu.com/robots.txt
EOF
cat > userdata4include.txt <<EOF
#include
# entries are one url per line. comment lines beginning with '
' are allowed
# urls are passed to urllib.urlopen, so the format must be supported there
http://frontier.cern.ch/userdata.txt
EOF
```
CloudVeneto User Guide, Release 3.11

cloud-init supply also a method called “multiple part” to supply user data in multiple ways, which means you can use userdata script and cloud-config (or other methods recognized by cloud-init) at the same time. cloud-init provides a script (write-mime-multipart) to generate a final userdata file. Here is an example:

```bash
#!/bin/bash
mkdir -p /tmp/rdu
echo "Hello World!" > helloworld.txt
EOF

write-mime-multipart -o userdata4multi.txt userdata4script userdata4config

Then use write-mime-multipart to generate userdata4multi.txt and use it to launch an instance:

write-mime-multipart -o userdata4multi.txt userdata4script userdata4config
→ userdata4inc

nova boot my_server --image "SL65-Padova-x86_64-20141023-QCOW2" \
   --flavor m1.xsmall --user-data userdata4multi.txt --key_name my_key

On Internet a lot of documentation (along with examples) is available on cloud-init, such as in the Ubuntu Documentation.

4.12 Resizing Virtual Machines

If the size of a virtual machine needs to be changed, such as adding more memory or cores, this can be done using the resize operation. Using resize, you can select a new flavor for your virtual machine. The operation will reboot the virtual machine and might take several minutes of downtime, so this operation should be planned as it will lead to application downtime.

To resize a VM using the graphical Interface:

- Detach any attached volume as described in Detaching a Volume

  Warning: Failure in doing so might lead to VM and/or Volume corruption!

- Select the Compute → Instances menu and then Resize Instance option on the Actions.

- In the Resize Instance box select the desired flavor.
• After the new flavor has been selected, the status will become ‘resize’ or ‘migrating’.

• The status will change after several minutes to ‘Confirm’ or ‘Revert Resize/Migrate’. You may need to refresh the web browser page to reflect the new status.

• Select **Confirm Resize/Migrate** if you wish to change the instance to the new configuration.

The status will then change to ‘Active’ once completed.

### 4.13 Snapshotting Virtual Machines

A snapshot of a Virtual Machine is an image which preserves the disk state of that running instance. Snapshots can be used e.g. to backup your instances, or to migrate your instances to another cloud.
Snapshots can also be used to replicate an installation from one instance to new instances: new virtual machines can in fact be created from a previously saved snapshot of an instance. However, as described in User provided images, we suggest instead to create new fresh images, or to customize (using contextualization) existing images. If this is not possible, and therefore you need to create a snapshot of a VM (and then use such snapshot to create new instances), we suggest to use the smallest (in terms of disk size) flavor for the VM that will be snapshotted.

To save a snapshot of your VM:

- **Shutdown your VM**: Log in as root to your VM. Please DO NOT shutdown the VM from the Openstack dashboard to make sure all data are correctly flushed on disk. Issue the `shutdown -h now` or `poweroff` command;
- From the Compute → Instances table select the desired VM and click Create Snapshot on the Actions menu.

**Warning**: Only the content of the ‘root disk’ is saved when you do a snapshot. So if the instance was created using a flavor that has a supplementary ephemeral disk, the content of such disk is NOT saved when snapshotting.

**Note**: In CloudVeneto snapshot size is limited to 25 GB.

### 4.14 Deleting Virtual Machines

VMs can be deleted using the **Terminate Instance** option in the OpenStack dashboard.

**Warning**: This command will immediately terminate the instance, delete all content of the virtual machine and erase the ephemeral disk. This operation is not recoverable.

**Important**: Virtual machines, even if idle or in shutdown state, allocate resources which therefore aren’t available to other users. Therefore please delete the virtual instances that you don’t plan to use anymore in the near future. Before deleting your virtual machines, you may want to save a snapshot of their disk image, as explained above.

### 4.15 Migrating an instance to another project/cloud

The migration of an instance from one project to another one, or to a different OpenStack cloud, can be done using snapshots.

In short the procedure to migrate an instance is the following:

- Create a snapshot of the instance in the source project, if possible (please refer to the instructions provided at Snapshotting Virtual Machines which also explain what are the restrictions of this procedure).
- Transfer the snapshot from the source project to the destination one (this was discussed in the previous section: Migrating an image to another cloud).
- In the target environment boot a new instance from the snapshot.
A container is a standard unit of software that packages up everything needed to run an application: code, runtime, system tools, system libraries and settings. This allows applications to run reliably when moved from one computing environment to another.

Container engines, such as Docker, operate at the Operating System level, so they can be run inside a VM.

### 5.1 Running docker containers in the Cloud

For information about Docker and how to install it, please look at the [Docker Documentation](#). We report here only about a couple of possible issues that need to be properly addressed.

The first one concerns networking. Docker by default sets the default MTU to 1500, but the default MTU for Cloud virtual machines is reduced by the overhead added by some networking components. Thus you need to reduce the Docker MTU to say 1420.

[Here](#) are the detailed steps on how to change the MTU for containers in docker.

The second issue concerns the file system: there could be problems running docker on old CentOS 7 releases where xfs is used as file system, as reported [here](#). Newest CentOS 7 releases are not affected by this issue.

If, for some reasons, you can’t use a newer version of CentOS 7 not affected by this problem, the instructions reported [here](#) might help.

### 5.2 Orchestrating containers

If you need to orchestrate multiple containers, we suggest to use Kubernetes (K8S), which is the most popular container orchestrator. Chapter *Creating Kubernetes Clusters on the Cloud* explains how to deploy a Kubernetes cluster on the Cloud.
CloudVeneto provides Unipd Physics Dept. and INFN Padova users with some GPUs (Graphics Processing Units). These are:

- 4 GPU Nvidia V100
- 4 GPU Nvidia Tesla T4
- 1 GPU Nvidia Quadro RTX 6000
- 2 GPU Nvidia TITAN Xp
- 1 GPU Nvidia GeForce GTX TITAN

Using a CloudVeneto GPU means accessing a virtual machine which has full access and direct control of such GPU device.

### 6.1 Creating a GPU instance

GPU instances, i.e. virtual machines which have access to one or more GPUs can be created only from the HPC-Physics project. The only exception is for the T4 GPUs that are usable also from the PhysicsOfData-students project.

So, first of all, you need to request the affiliation to such project (see Apply for other projects for the relevant instructions).

The instructions to then create a GPU instance are the very same for the creation of a 'standard’ virtual machine (see Creating Virtual Machines). You will only have to pay attention to use one of these special flavors:

- **cloudveneto.18cores56GB20+75GB1V100**
  
  Flavor for an instance with 1 GPU Nvidia V100, 18 VCPUs, 56 GB of RAM, 20 GB of ephemeral root disk space, 75 GB of extra ephemeral disk space.

- **cloudveneto.36cores112GB20+170GB2V100**
Flavor for an instance with 2 GPU Nvidia V100, 36 VCPUs, 112 GB of RAM, 20 GB of ephemeral root disk space, 170 GB of extra ephemeral disk space.

- **cloudveneto.72cores224GB20+360GB4V100**
  Flavor for an instance with 4 GPU Nvidia V100, 72 VCPUs, 224 GB of RAM, 20 GB of ephemeral root disk space, 360 GB of extra ephemeral disk space.

- **cloudveneto.15cores90GB20+700GB1T4**
  Flavor for an instance with 1 GPU Nvidia T4, 15 VCPUs, 90 GB of RAM, 20 GB of ephemeral root disk space, 700 GB of extra ephemeral disk space.

- **cloudveneto.30cores180GB20+1400GB2T4**
  Flavor for an instance with 2 GPUs Nvidia T4, 30 VCPUs, 180 GB of RAM, 20 GB of ephemeral root disk space, 1400 GB of extra ephemeral disk space.

- **cloudveneto.60cores360GB20+2800GB4T4**
  Flavor for an instance with 4 GPUs Nvidia T4, 60 VCPUs, 360 GB of RAM, 20 GB of ephemeral root disk space, 2800 GB of extra ephemeral disk space.

- **cloudveneto.8cores40GB20+500GB1Quadro**
  Flavor for an instance with 1 GPU Nvidia Quadro RTX 6000, 8 VCPUs, 40 GB of RAM, 20 GB of ephemeral root disk space, 500 GB of extra ephemeral disk space.

- **cloudveneto.8cores40GB20+500GB1TitanXP**
  Flavor for an instance with 1 GPU Nvidia Titan Xp, 8 VCPUs, 40 GB of RAM, 20 GB of ephemeral root disk space, 500 GB of extra ephemeral disk space.

- **cloudveneto.16cores80GB20+1000GB2TitanXP**
  Flavor for an instance with 2 GPUs Nvidia Titan Xp, 16 VCPUs, 80 GB of RAM, 20 GB of ephemeral root disk space, 1000 GB of extra ephemeral disk space.

- **cloudveneto.4cores20GB20+200GB1GeforceGtx**
  Flavor for an instance with 1 GPU Nvidia GeForce GTX TITAN, 4 VCPUs, 20 GB of RAM, 20 GB of ephemeral root disk space, 200 GB of extra ephemeral disk space.

**Warning:** When you snapshot an instance created using one of such flavors, please consider that only the root disk is saved. The content of the extra ephemeral disk is not saved!

**Warning:** For historical reasons there are still some instances created using flavors with a large root disk: since the size limit for images and snapshots is 25 GB, such instances cannot be snapshotted.

**Note:** Before allocation one or more GPUs, please register such allocation in this document. Please be sure to fill also the ‘estimated end date’ field. If the GPU(s) you want to use are not available, you may use that document to register a reservation.

### 6.1.1 Images for GPU instances

You are responsible to create the image to be used (see *User Provided Images* and *Building Images*).
These instructions explain how to install CUDA toolkit and the relevant drivers.

**Note:** For better performance, we suggest to create images:
- in raw format
- setting the properties `hw_disk_bus=scsi` and `hw_scsi_model=virtio-scsi`, i.e., using the command line tool:

```bash
# glance image-update --property hw_disk_bus=scsi <image-id>
# glance image-update --property hw_scsi_model=virtio-scsi <image-id>
```

Just for reference, for the HPC-Physics project we provide a CentOS7.x image (GPU-CentOS7-INFNPadova-x86_64-<date>) which has the same content of the CentOS7x-INFNPadova-x86-64-<date> public images, and in addition provides the CUDA toolkit and the needed drivers. This image was tested with Nvidia T4 GPUs and with Nvidia V100 GPUs.

**Warning:** On a VM instantiated using this image, cuda is installed at the first boot (and its installation can take several minutes). You may understand if the installation has been done if the following command:

```bash
# rpm -q cuda-drivers
```

returns something like.

```
cuda-drivers-440.33.01-1.x86_64
```

A further reboot might then be needed.

### 6.2 Monitoring

Unfortunately it is not straightforward to see which GPUs are being used and which ones are available using the CloudVeneto Openstack dashboard.

You can refer to [this page](#) for such information (please note that this page is updated every 30 minutes).

### 6.3 Policies

Please consider the following policies when using GPU instances:

- Since there is a high request to use GPUs, please **delete** your instance as soon as you don’t need it anymore. This is because virtual machines, even if idle or in shutdown state, allocate resources (GPUs in particular) which therefore aren’t available to other users.
- Once activated, your virtual instance is **managed by you**.
- Before allocation one or more GPUs, please register such allocation in [this document](#). Please be sure to fill also the ‘estimated end date’ field. Instances for which there isn’t an entry in this document can be deleted by the Cloud administrators.
- Please don’t reserve the GPU(s) for long (i.e > 1 week) periods.
There are several ways of handling disk storage in the CloudVeneto:

- **Ephemeral storage** exists only for the life of a virtual machine instance. It will persist across reboots of the guest operating system but when the instance is deleted so is the associated storage. The size of the ephemeral storage is defined in the virtual machine flavor.

- **Volumes** are persistent virtualized block devices independent of any particular instance. Volumes may be attached to a single instance at a time, but may be detached or re-attached to a different instance while retaining all data, much like a USB drive. The size of the volume can be selected when it is created within the quota limits for the particular project.

- **Object Storage** (experimental) allows to store and retrieve potentially lots of data with a simple API. It’s built for scale and optimized for durability, availability, and concurrency across the entire data set. Object storage is ideal for storing unstructured data that can grow without bound.

### 7.1 Ephemeral storage

Ephemeral storage exists only for the life of a virtual machine instance. It will persist across reboots of the guest operating system but when the instance is deleted so is the associated storage. The size of the ephemeral storage is defined in the virtual machine flavor.

Among the flavor details (that are listed in the Dashboard when a VM has to be launched or can be seen using the `openstack flavor list` command), there is an attribute called ‘Ephemeral’. When you use a flavor with an ephemeral disk size different from zero, the instance is booted with an extra virtual disk (besides the root disk) whose size is indicated by the ephemeral value. Please note that, when snapshotting a virtual machine instantiated using a flavor with the supplementary ephemeral disk, the content of such extra disk won’t be saved: only the content of the root disk will be snapshotted.

**Warning:** Please note that backups are not performed on ephemeral storage systems.
7.2 Volumes

Volumes are persistent virtualized block devices independent of any particular instance. Volumes may be attached to a single instance at a time (i.e. not like a distributed filesystem such as Lustre or Gluster), but they may be detached or re-attached to a different instance while retaining all data, much like a USB drive.

**Warning:** Please note that backups are not performed on volumes.

7.2.1 Create a Volume

The steps to add a Volume are:

Using the Dashboard, click on **Volumes** → **Volumes** and then **Create Volume**.
In the “Create Volume” window specify the name of the volume (*testvol* in the example below) and the desired size (12 GB in the example). As *Volume Source* specify “No source, empty volume”.

7.2. Volumes
Multiple volume types exist, and you need to specify the type to be used for the volume to be created.
Note: If you are a University of Padova user, please select the *equallogic-unipd* volume type.

If you are an INFN user, please select the *ceph* volume type (the default) unless you have been told by the cloud administrator to use another volume type.

In general different quotas for the different volume types are set. Unfortunately the OpenStack dashboard shows only the overall quota. To see the quota per each volume type you need to use the OpenStack CLI (see: *Accessing the Cloud with command line tools*) and run the `cinder quota-usage ${OS_PROJECT_ID}` command.

E.g.:

```
$ cinder quota-usage ${OS_PROJECT_ID}
+----------------------------+--------+----------+-------+
<table>
<thead>
<tr>
<th>Type</th>
<th>In_use</th>
<th>Reserved</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup_gigabytes</td>
<td>0</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>backups</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>gigabytes</td>
<td>72</td>
<td>0</td>
<td>400</td>
</tr>
<tr>
<td>gigabytes_ceph</td>
<td>48</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>gigabytes_equallogic-unipd</td>
<td>24</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>gigabytes_iscsi-infnpd</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>per_volume_gigabytes</td>
<td>0</td>
<td>0</td>
<td>5000</td>
</tr>
<tr>
<td>snapshots</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>
```

(continues on next page)
In this example the project was given 400 GB. For the ceph and equallogic-unipd volume types the quota is 200 GB, while it is not possible to create a volume using the iscsi-infnpd volume type.

**Warning:** If you try to create a volume using a type for which the quota is over, you will see a generic ‘Unable to create volume’ error message.

### 7.2.2 Using (attaching) a Volume

The new defined volume will appear in the **Volumes** tab.

To attach this volume to an existing instance, click on **Actions → Manage Attachments**:
Select the relevant Virtual Machine…
... and click on Attach Volume.

Log in to the instance and check if the disk has been added:

```bash
grep vdb /proc/partitions
253 16 12582912 vdb
```

If needed, create a file system on it (this will scratch the disk!):

```bash
mkfs -t ext4 /dev/vdb
```

Mount the volume:

```bash
mount /dev/vdb /mnt
```

**Note:** Please note that you can attach a volume of yours to an instance created by another user of the same project, but the opposite is not possible: you can’t attach a volume created by another user to an instance of yours.

### 7.2.3 Detaching a Volume

To detach a volume from an instance, first of all log into the virtual machine that has the volume mounted, and unmount it:

```bash
umount /mnt
```
Then, using the Dashboard, click on Volumes, click on Actions → Manage Attachments for the relevant volume and select Detach Volume. The detached volume can then be associated to another VM, as described above (you won’t have to re-create the file system, otherwise you will lose the content of the volume!)

7.2.4 Deleting a Volume

If a volume is not needed any longer, to completely remove it (note that this step cannot be reverted!):
- if needed, detach the volume from the associated instance
- using the Dashboard, click on Volumes → Volumes, select the relevant volume and then select Delete Volumes.

7.2.5 Sharing a volume between multiple (virtual) machines

As discussed in Volumes, a volume may be attached to a single instance. However it can be shared with other virtual machines of the Cloud (and/or with other hosts) using NFS.

1. Configure NFS server
   a. Once a volume has been created, formatted and attached to an instance acting as NFS server, create the mount point and mount the volume on this virtual machine:

   ```
   mkdir /dataNfs
   mount /dev/sdb /dataNfs
   ```

   b. Ensure that on this virtual machine the packages providing the rpc.nfsd daemon are installed:

   ```
   # yum whatprovides "/rpc.nfsd"
   Loaded plugins: dellsysid, fastestmirror
   Loading mirror speeds from cached hostfile
   * base: centos.fastbull.org
   * epel: mirror.23media.de
   * extras: centos.fastbull.org
   * updates: fr2.rpmfind.net
   1:nfs-utils-1.3.0-0.8.el7.x86_64: NFS utilities and supporting clients and,...
   → daemons for the kernel NFS server
   Repo: base
   Matched from:
   Filename: /usr/sbin/rpc.nfsd

   1:nfs-utils-1.3.0-0.8.el7.x86_64: NFS utilities and supporting clients and,...
   → daemons for the kernel NFS server
   Repo: @base
   Matched from:
   Filename: /usr/sbin/rpc.nfsd

   # yum install nfs-utils
   ```

   c. Insert the correct export directive in the /etc/exports file. For example if the volume must be visible in read-only mode to all the virtual machines of the same subnet 10.67.1.* (check the subnet with the ifconfig command) the content of the /etc/exports file will be:

   ```
   /dataNfs 10.67.1.*(ro,no_root_squash)
   ```
Note that there are no spaces between the ‘*’ and the ‘(‘.

If the volume must be visible in read-write to all the virtual machines of the same subnet 10.67.1.*, you might export it using the async or sync option. In short: async is much faster wrt sync, but can lead to data corruption if the server crashes during write operations (async means that the NFS server will acknowledge data before it’s committed to disk, while sync does the opposite: the server will only acknowledge data after it’s written out). Sync can be slow in particular when you have to write many files, since the open()/creat() and close() system calls have to wait for the new data to hit disk.

To export a volume using the async option the /etc/exports will be something like:

```
dataNfs 10.67.1.* (async, rw, no_root_squash)
```

To export a volume using the sync option the /etc/exports will be something like:

```
dataNfs 10.67.1.* (sync, rw, no_root_squash)
```

d. Check the firewall on the virtual machine. Ensure that the other instances have both UDP and TCP access to ports 111, 2049 and 875:

   - For newer distributions using firewalld issue the following commands:
     
     ```bash
     firewall-cmd --add-service=nfs
     firewall-cmd --permanent --add-service=nfs
     ```

   - For older distributions using iptables, the /etc/sysconfig/iptables file should be something like this:

     ```
     *filter
     :INPUT ACCEPT [0:0]
     :FORWARD ACCEPT [0:0]
     :OUTPUT ACCEPT [0:0]
     -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
     -A INPUT -p icmp -j ACCEPT
     -A INPUT -i lo -j ACCEPT
     -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
     -A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT
     -A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT
     -A INPUT -m state --state NEW -m tcp -p tcp --dport 875 -j ACCEPT
     -A INPUT -m state --state NEW -m udp -p udp --dport 875 -j ACCEPT
     -A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT
     -A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT
     -A INPUT -j REJECT --reject-with icmp-host-prohibited
     -A FORWARD -j REJECT --reject-with icmp-host-prohibited
     COMMIT
     ```

     Please remember to restart iptables after any change on that file:

     ```bash
     service iptables restart
     ```

e. Check the security group (see Setting security group(s)): access to ports 111, 875 and 2049 (IPv4 Ingress both TCP and UDP) should be guaranteed:
f. Restart nfs server using:

```bash
systemctl restart nfs
```

or

```bash
service nfs restart
```

2. Configure client machine(s)

a. To mount the volume on the other VMs, check that the package providing the `mount.nfs` command is installed:

```bash
# yum whatprovides "/mount.nfs"
```

Loaded plugins: dellsysid, fastestmirror
Loading mirror speeds from cached hostfile
* base: centos.fastbull.org
* epel: mirror.23media.de
* extras: centos.fastbull.org
* updates: fr2.rpmfind.net
1:nfs-utils-1.3.0-0.8.el7.x86_64 : NFS utilities and supporting clients and → daemons for the kernel NFS server
Repo : base

(continues on next page)
Matched from:
Filename : /sbin/mount.nfs

1:nfs-utils-1.3.0-0.8.el7.x86_64 : NFS utilities and supporting clients and daemons for the kernel NFS server
Repo : @base
Matched from:
Filename : /sbin/mount.nfs

# yum install nfs-utils

b. issue a mount command such as this one (assuming 10.67.1.4 is the NFS server):

```bash
mount -t nfs 10.67.1.4:/dataNfs /mnt
```

**Warning:** It is highly suggested that the VM acting as NFS server is instantiated with enough resources (at least 2 VCPUs and 4 GB of RAM). Moreover it should be used only for hosting the NFS server (i.e. it shouldn’t be used for other activities).

**Note:** Please note that this procedure can be used to mount a volume also on hosts outside the Cloud: it is sufficient to specify the IP address of these hosts in the /etc/exports file on the instance acting as NFS server.

### 7.2.6 Transferring the ownership of a volume to another user

This section explains how to transfer the ownership of a volume to another person, that can possibly be affiliated to another project.

**Warning:** The volume to be transferred must not be attached to an instance.

The owner of the volume to be transferred should follow the following procedure:

Using the Dashboard, click on **Volumes → Volumes**. For the volume that you want to transfer, on the right side click on the black dart and choose **Create Transfer**.
Specify a Transfer Name and click on **Create Volume Transfer**

Create Volume Transfer

**Transfer Name**

```
volume_migration
```

**Description:**
Ownership of a volume can be transferred from one project to another. Once a volume transfer is created in a donor project, it then can be "accepted" by a recipient project. This is equivalent to the cinder transfer-create command.

Save the Transfer ID and the Authorization Key, or download them.

7.2. Volumes
These Transfer ID and the Authorization Key must be communicated to the person who should become the new owner of the volume. Such user must follow the following steps:

Using the Dashboard, click on Volumes → Volumes, and then click on Accept Transfer.

Fill in the Transfer ID and the Authorization Key received from the donor and then click on Accept Volume Transfer.

7.3 Object Storage (experimental)

The CloudVeneto Object Store is built upon Ceph and supports two interfaces:

- S3-compatible: provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API.
• Swift-compatible: provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.

Objects (which are typically files) in the object storage are organized in containers, also called buckets.

**Warning:** Please note that backups are not performed on object storage.

### 7.3.1 Accessing the object storage using the dashboard

To create a container, using the Dashboard, click on **Object Storage → Containers** and then click on **+ Container**.

Fill the **Container Name** field and then click on the **Submit** button.

You can then upload files to a container. Select the relevant container and then click on the up arrow:
Choose the file to upload and click the **Upload File** button:

Using the dashboard, you can then retrieve the objects (files) previously uploaded to a container.

**Warning:** Please note that containers (and their objects) are owned by the project, and not by the individual who created them. This means that, if you create a container and upload some files on this container, the other members of the project can see these files and possibly also delete them.

### 7.3.2 Accessing the object storage using the S3 interface

You can also access the object storage using the S3 interface.

`s3cmd` is a convenient command-line tool that can be used. First of all you need to create a `$HOME/.s3cfg` file, that requires the following configuration:

```bash
host_base = cloud-areapd.pd.infn.it:5210
host_bucket = cloud-areapd.pd.infn.it:5210
use_https = true
access_key = <your EC2 access key>
secret_key = <your EC2 secret key>
```

To find your EC2 credentials, in the Dashboard go to Project → API Access and then click on View Credentials.

To create a bucket in the Object Storage service, you can use `s3cmd` with the ‘`mb`’ command:
Then with the ‘ls’ command you can verify that the bucket has been created:

```
$ s3cmd ls
2020-04-14 16:30  s3://mybucket
```

Once created you can start adding files to the bucket:

```
$ s3cmd put /etc/fstab s3://mybucket
upload: '/etc/fstab' -> 's3://mybucket/fstab'  [1 of 1]
628 of 628  100% in  0s  780.01 B/s done
$

$ s3cmd ls s3://mybucket
2020-04-14 16:34 628 s3://mybucket/fstab
```

You can delete files from a bucket with the ‘del’ s3cmd command:

```
$ s3cmd del s3://mybucket/fstab
delete: 's3://mybucket/fstab'
$

$ s3cmd ls s3://mybucket
```

Once a bucket is empty, you can remove it with the ‘rb’ s3cmd command:

```
$ s3cmd rb s3://mybucket
Bucket 's3://mybucket/' removed
$

$ s3cmd ls
```

**Note:** As said above, containers (buckets) and their objects are owned by the project, and not by the individual who created them. If this is a problem, it is possible to have a personal object storage account, usable through the S3 interface.

### 7.4 Accessing storage external to the Cloud

As explained in *Network Access*, from an instance of the Cloud by default it is not possible to access a host/service hosted in INFN Padova or Legnaro. This also means that by default on a virtual machine of the CloudVeneto it is not possible to mount a file system exported from a storage server hosted in INFN Padova or Legnaro.
CloudVeneto provides an orchestration service, implemented through the OpenStack **Heat** component, that allows you to spin up multiple instances, and other cloud services in an automated fashion.

In Heat parlance, a **stack** is the collection of objects, or **resources**, that will be created by Heat. This might include instances (VMs), volumes, networks, subnets, routers, ports, security groups, security group rules, etc.

Heat uses the idea of a **template** to define a stack. If you want to have a stack that creates two instances connected by a private network, then your template would contain the definitions for two instances, a network, a subnet, and two network ports.

Either native **HOT** templates, and **AWS CloudFormation** (CFN) templates are supported. Templates in HOT (Heat Orchestration Template) format are typically, but not necessarily required, to be expressed as **YAML** while CFN (AWS CloudFormation) formatted templates are instead typically expressed in **JSON**.

### 8.1 Creating a template

This is a working example of a HOT template which:

- Creates a virtual machine connected to a project network;
- Creates a storage volume;
- Attaches this volume to the previously created VM;

```yaml
heat_template_version: 2015-04-30

description: Template which creates a VM and a cinder volume; volume is then attached to this VM

parameters:
  instance_name:
    type: string
    description: VM Name
    constraints:
```
resources:

- allowed_pattern: "[a-zA-Z0-9-]+"

resources:

my_volume:
  type: OS::Cinder::Volume
  properties:
    name: "testVolume"
    size: 3

my_instance:
  type: OS::Nova::Server
  properties:
    name: { get_param: instance_name }
    image: Centos7x86_64
    flavor: cloudveneto.small
    security_groups: [default]
    key_name: paolomazzon
    admin_pass: heattest
    networks: {["network": testing-lan]}

my_volume_attachment:
  type: OS::Cinder::VolumeAttachment
  properties:
    volume_id: { get_resource: my_volume }
    instance_uuid: { get_resource: my_instance }

outputs:

  instance_fixed_ip:
    description: fixed ip assigned to the server
    value: { get_attr: [my_instance, first_address] }

Templates have three sections:

# This is required.
heat_template_version: 2015-04-30

parameters:
  # parameters go here

resources:
  # resources go here (this section is required)

outputs:
  # outputs go here

The resources section specifies what resources Heat should create:

resources:
  my_resource_id:
    type: a_resource_type
    properties:
      property1: ...
      property2: ...

Hardcoded values can be replaced with parameters. The actual value to be used is then specified when the stack is created. In our example a parameter is used for the name of the VM to be created:
parameters:
  instance_name:
    type: string
    description: VM Name
    constraints:
      - allowed_pattern: "[a-zA-Z0-9-]+"

A full description of all the resources that can be used in the stack creation can be found accessing the Orchestrator
→ Resource Types menu.

Sometimes we want to extract information about a stack. In our example the output is the fixed IP of the VM created
by the stack:

outputs:
  server_ip:
    description: fixed ip assigned to the server
    value: { get_attr: [my_instance, first_address]:

Heat templates allow also to insert user data via cloud-init, e.g:

server01:
  type: OS::Nova::Server
  properties:
    image: s166
    flavor: cldareapd.xsmall
    user_data_format: RAW
    user_data:
      str_replace:
        template: |
        #!/bin/sh
        yum install -y httpd
        service httpd start
        iptables -I INPUT 4 -m state --state NEW -p tcp --dport 80 -j ACCEPT
        service iptables save
        service iptables restart

Resource startup order can be managed in Heat, as explained in this page. For example it is possible to create a VM
only when another one has been successfully started.

The Heat Orchestration Template (HOT) specification is available here.

### 8.2 Creating a stack

To create a stack using the browser please select the Orchestrator → Stacks left menu. From here select + Launch
stack.

You will be prompted to select a template.
You will then be asked to fill in the parameters of the template and launch the stack.

Then you can follow the status of your stack on the dashboard.
In a cloud environment, Virtual Machines are instantiated from images. These images are registered in an Image Management Service, in our case provided by the Glance OpenStack component.

### 9.1 Public Images

Some images in the CloudVeneto are provided by the Cloud administrators. These images are public, and visible to all users. They appear with Visibility equal to Public in the Compute → Images menu.
### Chapter 9. Managing Images

#### Images

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Status</th>
<th>Visibility</th>
<th>Protected</th>
<th>Disk Format</th>
<th>Size</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS</td>
<td>Image</td>
<td>Active</td>
<td>Public</td>
<td>No</td>
<td>QCOW2</td>
<td>637.31 MB</td>
<td>Launch -</td>
</tr>
<tr>
<td>osme</td>
<td>Image</td>
<td>Active</td>
<td>Public</td>
<td>No</td>
<td>QCOW2</td>
<td>12.67 MB</td>
<td>Launch -</td>
</tr>
<tr>
<td>Federico-Boss-27</td>
<td>Image</td>
<td>Active</td>
<td>Private</td>
<td>No</td>
<td>QCOW2</td>
<td>221.96 MB</td>
<td>Launch -</td>
</tr>
</tbody>
</table>

Displaying 3 items.
9.1.1 Public Images for INFN Padova users

The SL6x-INFNPadova-x86-64-<date> and CentOS7x-INFNPadova-x86-64-<date> images are basic SL6.x / CentOS 7.x images which also include cloud-init to perform contextualization based on the user data specified when the VM are instantiated. They also configure CVMFS and the relevant squid servers.

Such images also configure the Padova LDAP server for user authentication. This means that it is just necessary to “enable” the relevant accounts on the VM adding in the /etc/passwd file:

```
+name1::::::
+name2::::::
...```

and creating their home directories.

You might also want to set a different home directory wrt the one specified in LDAP, e.g.:

```
+pippo::::::/ehome/pippo:
```

Changes done in /etc/passwd could not be applied immediately by the system. In this case a:

```
nsccd -i passwd
```

should help.

You can access instances created using these images, through ssh using the Cloud keypair, considering the ‘root’ account, e.g.:

```
ssh -i ~/private/my_key root@10.64.17.3
```

**Note:** The SL6x-INFNPadova-x86-64-<date> and CentOS7x-INFNPadova-x86-64-<date> images also allow INFN-Padova system administrators to log (with admin privileges) on the instance.

INFN-Padova computing and Network service (supporto@pd.infn.it) can provide support only for instances created using such images (only to INFN-Padova users).

**Warning:** These images for INFN-Padova users are supposed to be used only by INFN Padova users and only for projects using a INFN (i.e. 10.64.x.0/24 network). Using these images on Unipd projects (i.e. the one using 10.67.x.0/24 networks) can cause problems.

9.2 User Provided Images

Users can provide their own images and upload them to the Cloud Image service: these images are private, meaning that they are only available to the users of the project they are uploaded for.

**Note:** Users are not allowed to publish public (i.e. available to all projects) images.

Many open source projects such as Ubuntu and Fedora produce pre-built images which can be used for certain clouds. If these are available, it is much easier to use them compared to building your own.

- Fedora repository
• Ubuntu repository
• CentOS 7 images

Using an Ubuntu image as an example, after you downloaded the image from the relevant web site, to upload such image using the command line tools (see Accessing the Cloud with command line tools) you need to:

• Authenticate to OpenStack using the openrc script:

```bash
$ . demo-openrc.sh
```

• Issue the following command:

```bash
$ openstack image create --private --disk-format=qcow2 --container-format=bare --file bionic-server-cloudimg-amd64.img ubuntu-bionic
```

Once loaded, the image can then be used to create virtual machines.

**Note:** Images are usually provided in several formats. The most used ones in Cloud environments are ‘qcow2’ and ‘raw’.

When considering the format to choose, you should take into account how the instantiation of a virtual machine on a Cloud compute node works. When a virtual machine is instantiated, first of all the relevant image is staged to the target hypervisor, if needed (i.e. if this is the first instance created on that compute node using this image). Then the image is converted to raw, if it is not already in that format.

A qcow2 image is usually much smaller with respect to the same image provided in raw format. This means that:

• the registration to the OpenStack image service for images in qcow2 format is faster with respect to images in raw format

• the staging of images in qcow2 format to the target compute node is faster with respect to images in raw format

However qcow2 images, as reported above, need to be translated into raw format, and this can take a non negligible time, in particular for images with a large virtual disk size.

Some system software is delivered in ISO image format. For example, these steps show how to create an image from the Freedos ISO available from here

```bash
$ openstack image create --private --disk-format=iso --container-format=bare --file=fd11src.iso freedos11
```

**Note:** In CloudVeneto image size is limited to 25 GB.

### 9.3 Sharing Images

As mentioned before, users are not allowed to publish public images. However images can be shared between different projects. This is currently only possible via the command line tools.

If an image has been uploaded to your currently active project, using the procedure described in User Provided Images, you can then use the `openstack image add project` operation to share that image with another project.

To share an image, first source the project profile for the project containing the image you want to share (e.g. *Fedora-Cloud-Base-27*) and find its id with the `openstack image list` command (`d4b02b71-755e-47ad-bb27-1ea5c23bf7cb` in the example):
You then need to change (to ‘Shared’) the visibility of the image:

```
$ openstack image set --property visibility=shared d4b02b71-755e-47ad-bb27-1ea5c23bf7cb
```

You now need to find the id of the project you wish to share the image with. This will generally be done by looking at the openrc file of that project and finding the `OS_PROJECT_ID` variable (in this example, it is e81df4c0b493439abb8b85bfd4cbe071).

To share the image with id d4b02b71-755e-47ad-bb27-1ea5c23bf7cb to the project whose id is e81df4c0b493439abb8b85bfd4cbe071, use the command:

```
$ openstack image add project d4b02b71-755e-47ad-bb27-1ea5c23bf7cb
  → e81df4c0b493439abb8b85bfd4cbe071
```

```
+------------+--------------------------------------+
| Field      | Value                                |
+------------+--------------------------------------+
| created_at | 2018-03-19T16:09:21Z                 |
| image_id   | d4b02b71-755e-47ad-bb27-1ea5c23bf7cb |
| member_id  | e81df4c0b493439abb8b85bfd4cbe071    |
| schema     | /v2/schemas/member                  |
| status     | pending                              |
| updated_at | 2018-03-19T16:09:21Z                |
+------------+--------------------------------------+
```

**Note:** Because of a bug in OpenStack this command could return an error message such as:

```
403 Forbidden: Not allowed to create members for image d4b02b71-755e-47ad-bb27-1ea5c23bf7cb. (HTTP 403)
```

even if actually the command worked.

Then a member of the target project (with id e81df4c0b493439abb8b85bfd4cbe071 in our example) needs to accept the image:

```
$ openstack image set --accept d4b02b71-755e-47ad-bb27-1ea5c23bf7cb
```

In the target project, the image will appear in the image list:
Chapter 9. Managing Images
9.4 Building Images

Users can also build custom images, that can then been uploaded in the Cloud Image service as described in User Provided Images.

There are several tools providing support for image creation. Some of them are described in the Openstack documentation.

One example is virt-builder, which is briefly described in the next subsection.

Please consider these guidelines when creating an image:

- Data inside images can be accessed by the other members of the project. So please don’t store sensitive information (e.g. password) in the image.
- Please make sure that the cloud-init and cloud-utils packages are installed. This is needed for contextualisation (see Contextualisations), which allows to customize the image after installation.
- If you use Fedora, CentOS, or RHEL, the cloud-utils-growpart and gdisk packages must be installed, since they are needed for extending partitions. If you use Ubuntu or Debian, the cloud-initramfs-growroot package, which supports resizing root partition on the first boot, must be installed.

Note: Another possible way to create an image is:

- instantiating a virtual machine using an already registered image
- applying your own customization on this instance (e.g. installing new packages)
- creating a snapshot of this image
- using the snapshot to instantiate new virtual machines

However this usually means creating very large and quite inefficient images. Therefore we strongly suggest instead to use the provided public images and applying your customizations using contextualization (see Contextualisations) or to create new images using one of the tools mentioned above.

9.4.1 virt-builder

Virt-builder is a command line tool for quickly creating customized images.

It takes cleanly prepared, digitally signed OS templates and customizes them.

Documentation how to use virt-builder is provided in the relevant man pages.

Here we provide some examples.

To see the operating system available to install, please use the following command:

```
virt-builder --list
```

The following command:

```
virt-builder -v -x centos-7.7 \
   --uninstall "chrony" \ 
   --install "ntp,cloud-init,cloud-utils,cloud-utils-growpart,gdisk" \ 
   --timezone Europe/Rome" \ 
   --write "/etc/ntp.conf:server ntp.pd.infn.it" \ 
   --run-command 'systemctl enable ntpd'
```

(continues on next page)
creates a centos 7.7 image, called centos-7.7.qcow2 in qcow2 format, where the packages ntp, cloud-init, cloud-utils, cloud-utils-growpart and gdisk are installed.

The above command removes from the image the package chrony.

The option –timezone Europe/Rome sets the timezone.

The string ‘server ntp.pd.infn.it’ is written into the file /etc/ntp.conf.

The command ‘systemctl enable ntpd’ is also issued (to start the ntpd service at boot).

The option –edit '/etc/resolv.conf:s/nameserver 10.0.2.3//' is used as a workaround to address a problem in the centos7 default image (a wrong nameserver is defined in /etc/resolv.conf).

The following command will instead create an ubuntu 18.04 image, called k8s-ubuntu-18.04.raw, in raw format:

```bash
virt-builder -v -x ubuntu-18.04 \
--install "net-tools,vim,apt-transport-https,ca-certificates,curl,software-properties-common,docker.io,sudo,wget,rsync,cloud-init,cloud-utils" \
--timezone Europe/Rome \
--output k8s-ubuntu-18.04.raw --format raw \
--run install.sh \
--firstboot-command 'apt-get update' \
--firstboot-command 'apt-get upgrade -y' \
--selinux-relabel
```

Besides specifying a set of packages to be installed, the above command specifies that the script install.sh must be run on the disk image. The script can be used e.g. to install packages from an extra repository:

```bash
$ cat install.sh
#!/bin/bash
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
apt-get update
apt-get install -y kubelet kubeadm kubectl
```

The command also specifies that the commands `apt-get update` and `apt-get upgrade -y` must be executed at the first boot.

virt-builder is very easy to install (it is provided by the `libguestfs-tools-c` package) and doesn’t require root privileges. If needed, access to an instance where virt-builder is installed can be provided: in such case please contact support@cloudveneto.it.

### 9.5 Deleting Images

Images that are not used anymore can be deleted. Deletion of images is permanent and cannot be reversed.

To delete an image, log in to the dashboard and select the appropriate project from the drop down menu at the top left. On the **Compute → Images** page select the images you want to delete and click **Delete Images**. Confirm your action pressing the **Delete Images** again on the confirmation dialog box.
Warning: Don’t delete an image if there are virtual machines created using this image, otherwise these VMs won’t be able to start if rebooted.

9.6 Migrating an image to another cloud

Sometimes it becomes necessary to migrate an image (or snapshot) from one OpenStack cloud to another one. In short the procedure is the following:

In the source cloud:
- Download the image/snapshot as a file

In the destination cloud:
- Upload the downloaded image file to the new environment

To download the image from the source cloud, you need to use the OpenStack CLI. After having sourced the environment script of the source cloud (as explained in Accessing the Cloud with command line tools), run the `openstack image list` command to find the id of the image (photo-slave in the following example):

```
$ openstack image list | grep photo-slave
| 753e8fa1-3f1b-407d-9294-d22b89ec3184 | photo-slave | active |
```

Then use the `openstack image save` command, using the relevant id, to download the image file:

```
$ openstack image save --file photo-slave.qcow2 753e8fa1-3f1b-407d-9294-d22b89ec3184
```

To upload the image to the destination environment, first of all source the environment script of the target cloud. Then run the `openstack image create` command:

```
$ openstack image create --private --disk-format=qcow2
  --container-format=bare
  --file photo-slave.qcow2 photo-slave
```
The virtual machines provided by CloudVeneto can also be used to implement batch clusters where users can run their applications (normal jobs or docker containers).

In this chapter we explain how to implement a dynamic batch cluster based on HTCondor and elastiq using ECM.

10.1 Intro: the idea

You create on the cloud a virtual machine that acts as a master for a dynamic batch system (implemented using HTCondor). When you create the master you will need to describe the cluster configuration, as described in The cluster configuration.

The master node will be able to spawn new slave nodes (where jobs are executed) when jobs are submitted to the batch system. The elastic cluster will provide a number of virtual resources that scales up or down depending on your needs. The total number of active virtual nodes is dynamic.

The master node will act also as submitting machine: you can log in on this node and submit jobs to the batch system. These jobs will run on the slave nodes, get done, and eventually the slaves will be released.

Note: The master can use a different flavor with respect to the slave nodes.

10.2 Creating Batch Clusters on the Cloud with HTCondor

When you create the master, using the instructions reported in Creating Virtual Machines, you will need to specify some user-data to describe the cluster configuration, as described below.

Note: The master and the slaves must use the same image (which should have all the needed software).
10.2.1 Prerequisites

- You should be registered in the Cloud as member of a project.

**Warning:** The project where you want to create the cluster should have only one available network (i.e the lan) otherwise, due to a bug with ec2api, the instantiation of slaves fails.

- You need to have created a SSH key-pair, ad explained in *Creating a keypair*. This will allow you to log in the master and slave nodes.

- You need to download the EC2 credentials of the project you want to use, as explained in *Accessing the Cloud through the euca2ools EC2 command line tools*.

- You need to identify the image to be used for the master and for the slaves. Currently supported operating systems are RHEL6.x and derivates (SL6.x, CentOS6.x, etc.), RHEL7.x and derivates and Ubuntu. uCernVM based images are also supported.

For such image you need to know the relevant EC2 (AMI) id (see *How to find the EC2 (AMI) id of an image*).

- You need to set a specific security group to be used for the master node. This security group must include the following rules:

<table>
<thead>
<tr>
<th>Direction</th>
<th>Ether Type</th>
<th>IP Protocol</th>
<th>Port Range</th>
<th>Remote IP Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egress</td>
<td>IPv4</td>
<td>Any</td>
<td>Any</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>Egress</td>
<td>IPv6</td>
<td>Any</td>
<td>Any</td>
<td>::/0</td>
</tr>
<tr>
<td>Ingress</td>
<td>IPv4</td>
<td>ICMP</td>
<td>Any</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>Ingress</td>
<td>IPv4</td>
<td>TCP</td>
<td>22 (SSH)</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>Ingress</td>
<td>IPv4</td>
<td>TCP</td>
<td>9618</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>Ingress</td>
<td>IPv4</td>
<td>TCP</td>
<td>41000 - 42000</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>

**Note:** Instead of modifying the rules of an existing security group, we suggest to create a new security group named e.g. “master_security_group”. Security groups are discussed in *Setting security group(s)*.

The slave nodes will instead use the *default* security group of your project. This group must include the following rule:

<table>
<thead>
<tr>
<th>Direction</th>
<th>Ether Type</th>
<th>IP Protocol</th>
<th>Port Range</th>
<th>Remote Security Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress</td>
<td>IPv4</td>
<td>Any</td>
<td>Any</td>
<td>&lt;master_security_group&gt;</td>
</tr>
</tbody>
</table>

where `<master_security_group>` is the name of the security group that was chosen for the master node.

- You need to download the ECM software. As explained in *The cluster configuration*, this will be used to create the batch cluster configuration:

```bash
$ git clone https://github.com/CloudPadovana/ECM.git
```

- You need to install the euca2ools package to discover the EC2 (AMI) id of images. Euca commands are also used by ECM to show the list of available images for the cluster.

```bash
yum install euca2ools (on RHEL/CentOS)
apt-get install euca2ools (on Ubuntu)
```
### 10.2.2 The cluster configuration

You must properly configure the ecm.conf file stored in the ECM directory (created when you downloaded via git the ECM software)

```bash
$ cat ecm.conf
FLAVOR_VMS=
MAX_VMS=
MIN_VMS=
JOBS_PER_VM=
IDLE_TIME=
KEY_NAME==
```

Where:

- `<FLAVOR_VMS>` is the name of the flavor to be used for the slave nodes. Flavors have been discussed in *Flavors*. Available flavors are listed in the dashboard when you try to launch a VM.
- `<MAX_VMS>` is the maximum number of slave nodes that can be instantiated.
- `<MIN_VMS>` is the minimum number of slave nodes (never terminated, always available).
- `<JOBS_PER_VM>` is the maximum number of jobs that will be run in a single slave node.

**Important:** You have to verify that the number of jobs per VM is compatible with the number of VCPUs of the selected flavor.

- `<IDLE_TIME>` is the time (in seconds) after which inactive VMs will be killed.
- `<KEY_NAME>` is the name (without the .pem extension) of ssh key previously created (see *Creating a keypair*) to be injected in the batch cluster nodes.

**Note:** The batch system will use each CPU as a separate job slot. So if you have a flavor with 4 VCPUs, and you submit 1 job, the master will create 1 slave and use 1 of the 4 available VCPUs. If you submit 4 jobs, again the master will create 1 slave, and will use all the 4 VCPUs. Large flavors means less machines to be created but possibly a sub-optimal usage of resources.

### 10.2.3 Start the elastic cluster

To start the elastic cluster, you only need to instantiate the master node.

When you create such master, you will need to specify some user-data to describe the cluster configuration. The ecm.py script will create such user-data file for you, using as input the ecm.conf file you previously edited. (see *The cluster configuration*).

First of all you have to set the relevant EC2 credentials:

```bash
$ source ec2rc.sh
```

Then you must launch ecm.py file and follow the instructions. We will create a CentOS 6 cluster as an example.

```bash
$ python ecm.py
```

Choose the Operating System (OS) you want to use for your master and worker nodes:

(continues on next page)
1: Fedora
2: Ubuntu
3: uCernVM
4: CentOS7
5: CentOS6

OS type => 5

**Important:** The same OS will be used to instantiate both master node and worker node(s).

Select the image **for** your CentOS6 based master **and** your CentOS6 based WNs:
1: CentOS 6
2: Other image. [WARNING] You have to know the EC2-id of image

Image => 1

**Warning:** If you choose “Other image” you have to manually insert the image id in EC2 format (see *How to find the EC2 (AMI) id of an image*).

The script will now print something like:

Now you can use the master-centos6-2017-03-16-17.45.31 file to instantiate the master node of your elastiq cluster.

Now you have to start the master node. As explained in *Creating Virtual Machines*, go to ‘Instances’ and create a new instance with [Launch Instances].

In the details select:

[details]

- *Instance Name* => whatever you like
- *Flavor* => whatever you like; can be different from the flavor chosen for the slave nodes
- *Image name* => The same image chosen for the slaves.

[Access and Security]

- *Key pair* => The key-pair that will be used to log on the nodes of the batch cluster
- *Security Group* => the security group for the master (choose only this one)

[post creation]

- *Customization Script Source* => Select “File” from the dropdown menu and use “Choose File” to upload the user_data_file created by the ecm.py script

Then press launch.

Once you requested the creation of the master node, after some minutes, you will see that the master virtual machine and some (depending on the “MIN_VMS” attribute you defined in the ecm.conf) slave nodes are created.

Get the IP address of master, and log in on this machine using the key you have imported/created before i.e:
$ ssh -i ~/.ssh/id_rsa root@10.64.YY.XXX

For security reasons, as root you can not submit jobs to the HTCondor cluster. So make sure that a ‘normal’ account exists in the master node. In case you can create it using the command:

```bash
# adduser <username>
```

Create a password for this account:

```bash
# passwd <username>
```

You have to import any external disk, create homes, etc, as you would do in a normal machine.

### 10.2.4 How slave nodes are installed and configured

The instantiation of slaves nodes is managed by the elastiq service running in the master node.

The min and max number of nodes is set by user in the ecm.conf: the total number of active nodes will change dynamically with jobs need.

The installation of condor and its configuration on slaves is automatically provided by ECM. Inside the user_data_file created by the ecm.py script there is the parameter SLAVE_USERDATA whose value is the script for the installation and configuration of the slave, coded in b64. The original uncoded script used for condor installation and configuration on slaves is stored in the ECM/slave_files directory. There is a file for each Operating system, depending on the system selected for the master. These files provide the basic configuration for condor, to support both the Vanilla and Docker universes.

Generally the user doesn’t need to modify these files. If, for some reasons, you need to modify the condor configuration or you need to install additional packages on the slaves, this script can be modified: ECM will take care to code it in b64 and add the new value to the SLAVE_USERDATA parameter in the user_data_file.

### 10.2.5 Use the elastic cluster

Log to the master node using your unprivileged account. Check if condor is running with the command:

```bash
$ condor_q
```

```
-- Schedd: 10-64-20-181.virtual-analysis-facility : <10.64.20.181:41742>
ID OWNER SUBMITTED RUN_TIME ST PRI SIZE CMD
0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
```

Check the status of the cluster:

```bash
$ condor_status
```

```
Name OpSys Arch State Activity LoadAv Mem ActvtyTime
slot1@10-64-22-84. LINUX X86_64 Unclaimed Idle 0.000 1977 2+12:44:58
slot2@10-64-22-84. LINUX X86_64 Unclaimed Idle 0.000 1977 2+12:45:25
Total Owner Claimed Unclaimed Matched Preempting Backfill
 X86_64/LINUX 2 0 0 2 0 0 0 0
Total 2 0 0 2 0 0 0 0
```

Create you HTCondor ‘job file’. A simple example is the following:
$ cat test.classad

Universe = vanilla
Executable = /home/<username>/test.sh
Log = test.log.$(Cluster)$.(Process)
Output = test.out.$(Cluster)$.(Process)
Error = test.err.$(Cluster)$.(Process)
Queue <number_of_jobs_to_submit>

where test.sh is the executable you want to run.

Submit your jobs issuing the command:

$ condor_submit test.classad

and check their status with:

$ condor_q

You can find documentation about HTCondor here.

10.2.6 Use the elastic cluster to run docker containers

The HTCondor elastic cluster can also be used to run docker containers. You don’t need to install docker on your
images: this is done by ECM.

Once the cluster is created, verify that Docker is enabled on the slaves:

# condor_status -l | grep -i Docker
StarterAbilityList = "HasTDP,HasEncryptExecuteDirectory,HasFileTransferPluginMethods, 
HasJobDeferral,HasJICLocalConfig,HasJICLocalStdin,HasPerFileEncryption,HasDocker, 
HasFileTransfer,HasReconnect,HasVM,HasMPI,HasRemoteSyscalls,HasCheckpointing"
DockerVersion = "Docker version 1.7.1, build 786b29d/1.7.1"
HasDocker = true
StarterAbilityList = "HasTDP,HasEncryptExecuteDirectory,HasFileTransferPluginMethods, 
HasJobDeferral,HasJICLocalConfig,HasJICLocalStdin,HasPerFileEncryption,HasDocker, 
HasFileTransfer,HasReconnect,HasVM,HasMPI,HasRemoteSyscalls,HasCheckpointing"
DockerVersion = "Docker version 1.7.1, build 786b29d/1.7.1"
HasDocker = true

The following is a simple example which runs a docker container, which is downloaded from docker-hub:

$ cat test-docker.classad

universe = docker
docker_image = debian
executable = /bin/cat
arguments = /etc/hosts
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
Log = test-docker.log.$(Cluster)$.(Process)
Output = test-docker.out.$(Cluster)$.(Process)
Error = test-docker.err.$(Cluster)$.(Process)
request_memory = 100M
Queue <number_of_jobs_to_submit>
10.2.7 How to find the EC2 (AMI) id of an image

As explained above, to use the Elastic batch cluster you need to know the EC2 (AMI) id of the image you want to use.

First of all you need to install the euca2ools and to download the EC2 credentials for your project, as explained in Accessing the Cloud through the euca2ools EC2 command line tools.

Uncompress (unzip) the EC2 zip file, and source the ec2rc.sh script to set the correct environment:

```bash
$ source ec2rc.sh
```

To discover the EC2 id of your image or snapshot, use the `euca-describe-images` command:

```bash
$ euca-describe-images -I ${EC2_ACCESS_KEY} -S ${EC2_SECRET_KEY} -U ${EC2_URL}
```

In the example above:

- the EC2 image id of the uCernVM 2.3-0 image is ami-0000031b
- there is a snapshot whose EC2 is is ami-00000447.

**Note:** In case you have snapshot on the output of the euca-describe-images you notice that you have no name associated with the ami-id. To obtain a nicely formatted list of (ami-id, name) couples you can use the following command:

```bash
$ euca-describe-images --debug 2>&1 | grep 'imageId|name' | sed 'N;s/
/ /'
```

You can also see all the information of an image, e.g.:

10.2. Creating Batch Clusters on the Cloud with HTCondor 101
### 10.2.8 Restart the elastic cluster

This section describes the steps to be followed to restart the elastic cluster (this might be needed e.g. after a reboot).

- If possible, delete all the slave nodes via dashboard;
- Log on the master node as root;
- Check if both condor and elastiq services are already running

```bash
# service condor status
# service elastiq status
```

In this case new slaves will be created and will join the cluster in some minutes.

- if only condor service is running but elastiq isn’t, please restart elastiq using the following command:

```bash
# service elastiq start
```

or

```bash
# elastiqctl restart
```

and wait for the creation of new slaves that will connect to the cluster in some minutes;

- If condor isn’t running and some elastiq processes are up and running, kill them:

```bash
# ps -ef | grep elastiq
# kill -9 <n_proc>
```

and start the condor service using the following command:

```bash
# service condor start
```

Now the condor_q command should return
and the condor_status command output should be empty (no nodes running)

Then start the elastiq service

```
# service elastiq start
```

In some minutes the minimum number of nodes should connect to the condor cluster and the condor_status command output should show them e.g.:

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot1@10-64-22-215 LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1896</td>
<td>0+00:24:46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot2@10-64-22-215 LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1896</td>
<td>0+00:25:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot1@10-64-22-217 LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1896</td>
<td>0+00:24:44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot2@10-64-22-217 LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1896</td>
<td>0+00:25:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot1@10-64-22-89. LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Benchmark</td>
<td>1.000</td>
<td>1896</td>
<td>0+00:00:04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot2@10-64-22-89. LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.040</td>
<td>1896</td>
<td>0+00:00:05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.2.9 Troubleshooting the elastic cluster

Check the log files

The elastiq log file is /var/log/elastiq/elastiq.log. It is not straightforward to interpret it without knowing its structure. When you start the elastiq service the first part of log file reports the check of cloud user’s credentials and of other parameters configured in the elastiq.conf file (i.e. the userdata file for slave nodes)

```
INFO [__init__.conf] Configuration: ec2.image_id = ami-9f3da3fc (from file)
INFO [__init__.conf] Configuration: ec2.flavour = cldareapd.medium (from file)
2INFO [__init__.conf] Configuration: ec2.aws_secret_access_key = [...] (from file)
INFO [__init__.conf] Configuration: ec2.key_name = my_key (from file)
INFO [__init__.conf] Configuration: ec2.user_data_b64 = [...] (from file)
INFO [__init__.conf] Configuration: ec2.aws_access_key_id = [...] (from file)
INFO [__init__.conf] Configuration: quota.max_vms = 3.0 (from file)
INFO [__init__.conf] Configuration: quota.min_vms = 1.0 (from file)
NFO [__init__.main] Loaded batch plugin "htcondor"
DEBUG [htcondor.init] HTCondor plugin initialized
DEBUG [__init__.main] EC2 image "ami-9f3da3fc" found
```

If your credentials are wrong, you will see in the log file an error as:

```
ERROR [__init__.ec2_running_instances] Can't get list of EC2 instances (maybe wrong credentials?)
```

If you specified a wrong ami_id for the image of slave nodes, the error message in the log will be:

```
ERROR [__init__.main] Cannot find EC2 image "ami-00000000"
```

Elastiq periodically checks all the VMs. If a VM is correctly added to the condor cluster, it logs:

```
10.2. Creating Batch Clusters on the Cloud with HTCondor
103
```
When elastiq instantiates a new VM it logs something like:

```
WARNING [__init__.ec2_scale_up] Quota enabled: requesting 1 (out of desired 1) VMs
INFO [__init__.ec2_scale_up] VM launched OK. Requested: 1/1 | Success: 1 | Failed: 0
| ID: i-f026f340
DEBUG [__init__.save_owned_instances] Saved list of owned instances: i-f026f340
```

and when elastiq deletes an idle VM it logs:

```
INFO [__init__.check_vms] Host 10-64-22-190.INFN-PD is idle for more than 2400s:
→ requesting shutdown
INFO [__init__.ec2_scale_down] Requesting shutdown of 1 VMs...
```

In the master node of condor, logs are located in /var/log/condor directory. You may refer to the HTCondor documentation for more information on these log files.

```
# ls -l /var/log/condor/
total 76
-rw-r--r--. 1 condor condor 24371 Jan 18 08:42 CollectorLog
-rw-r--r--. 1 root root 652 Jan 18 08:35 KernelTuning.log
-rw-r--r--. 1 condor condor 2262 Jan 18 08:35 MasterLog
-rw-r--r--. 1 condor condor 0 Jan 18 08:35 MatchLog
-rw-r--r--. 1 condor condor 19126 Jan 18 08:42 NegotiatorLog
-rw-r--r--. 1 root root 13869 Jan 18 08:42 ProcLog
-rw-r--r--. 1 condor condor 474 Jan 18 08:35 ScheddRestartReport
-rw-r--r--. 1 condor condor 2975 Jan 18 08:40 SchedLog
```

Check the running processes

Processes expected to run are:

- for the condor service:

  ```
  # ps -ef | grep condor
  condor 764 1 0 14:09 ? 00:00:00 /usr/sbin/condor_master -f
  root 960 764 0 14:09 ? 00:00:00 condor_procd -A /var/run/
  →condor/procd_pipe -L /var/log/condor/ProcLog -R 1000000 -S 60 -C 996
  condor 961 764 0 14:09 ? 00:00:00 condor_collector -f
  condor 974 764 0 14:09 ? 00:00:00 condor_negotiator -f
  condor 975 764 0 14:09 ? 00:00:00 condor_schedd -f
  ```

- for the elastiq service:

  ```
  # ps -ef | grep elastiq
  elastiq 899 1 0 14:09 ? 00:00:00 SCREEN -dmS __|elastiq|__ /bin/sh -c python /usr/bin/elastiq-real.py --logdir=/var/log/elastiq --config=/etc/elastiq.conf --statefile=/var/lib/elastiq/state 2> /var/log/elastiq/elastiq.err
  ```
Warning: The `condor_status` information isn’t updated very frequently. So it can happen that `condor_status` shows nodes that have been already removed from the cloud by elastiq.
Creating Kubernetes Clusters on the Cloud

The virtual machines provided by CloudVeneto can also be used to deploy Kubernetes clusters where users familiar with Kubernetes can run their applications. **Kubernetes (K8S)** is the most popular production-grade container orchestrator.

In this chapter we explain how to deploy a Kubernetes cluster using **Ansible**.

### 11.1 Intro: the idea

The provided Ansible playbooks allow you to deploy a Kubernetes cluster both on bare metal and on an OpenStack cloud. The installation is based on the **kubeadm** tool configured with a pre-generated admin token and flannel network. The playbooks can enrich the cluster installation with a set of services such as:

- Dashboards: K8S legacy and Grafana;
- Monitoring: Prometheus operator;
- Big Data Analytics: Apache Spark and Kafka operators.

### 11.2 System requirements

The deployment environment requires:

- Ansible 2.5.0+ (on the user client)
- Ubuntu 18.04 (for master and node images)
- 2 CPUs or more per machine, 2 GB or more of RAM per machine (any less will leave little room for your apps)
11.3 Getting Started

This section represents a quick installation and is not intended to teach you about all the components. The easiest way to get started is to clone the ‘ansible-k8s’ repository:

```
# git clone https://github.com/zangrand/ansible-k8s.git
# cd ansible-k8s
```

The directory structure should be like:

```
ansible-k8s/
  |-- config
  |   |-- config
  |   |   |-- keystone_client.py
  |   |   |-- tls-ca-bundle.pem
  |   |-- examples
  |   |   |-- spark-pi.yaml
  |   |   |-- kcluster.yaml
  |   |   |-- ktopic.yaml
  |   |-- deploy_k8s.yaml
  |   |-- deploy_master_openstack.yaml
  |   |-- deploy_node_openstack.yaml
  |   |-- group_vars
  |   |   |-- all
  |   |-- inventory
  |   |-- k8s
  |   |   |-- alertmanager-service.yaml
  |   |   |-- dashboard-setup.yaml
  |   |   |-- grafana-service.yaml
  |   |   |-- os-k8s-node.yaml
  |   |   |-- prometheus-service.yaml
  |   |-- openstack_config.yaml
  |   |-- README.md
  |-- roles
     |-- auth
     |   |-- keystone
     |   |   |-- files
     |   |   |   |-- infn_ca.pem
     |   |   |   |-- k8s-auth-policy.yaml
     |   |   |   |-- k8s-keystone-auth.yaml
     |   |   |   |-- k8s-keystone-auth.yaml_orig
     |   |   |   |-- keystone_client.py
     |   |   |   |-- tls-ca-bundle.pem
     |   |   |   |-- webhookconfig.yaml
     |   |-- tasks
     |   |   |-- main.yml
     |-- common
     |   |-- tasks
     |   |   |-- main.yml
     |-- docker
     |   |-- tasks
     |   |   |-- main.yml
     |-- haproxy
     |   |-- tasks
     |   |   |-- main.yml
     |   |-- templates
     |   |   |-- haproxy.cfg.j2
```

(continues on next page)
11.4 Deployment on the cloud

The provided Ansible playbook is able to create and configure properly all hosts (i.e. the VMs) on CloudVeneto and deploy Kubernetes on it. To do it:

- edit the file openstack_config.yaml and fill up all required attributes (i.e. OS_USERNAME, OS_PASSWORD, OS_PROJECT_NAME, OS_PROJECT_ID, OS_NETWORK, etc), the same used for accessing OpenStack by its client;
- define the VMs characteristics of the master and nodes, in terms of name, flavor, and image (default values for flavor and image are defined);
- specify the number of nodes (i.e. OS_NODES) of your cluster.

Look at the comments inside the openstack_config.yaml file for more details.

Verify if the ‘shade’ Python module is available on your environment, otherwise install it:

```
# pip install shade
```

Execute:
11.5 How to access your Kubernetes cluster

There are two different ways to access the Kubernetes cluster: the kubectl command line tool or the dashboard.

11.5.1 Kubectl

The kubectl command line tool is available on the master node. If you wish to access the cluster remotely please see the following guide: Install and Set Up kubectl.

You can enable your local kubectl to access the cluster through the Keystone authentication. To do it, copy all files contained into the folder ansible-k8s/config/ to $HOME/.kube/.

The tls-ca-bundle.pem file is CA certificate required by the CloudVeneto OpenStack based cloud. Do not forget to source the openrc.sh with your OpenStack credentials and OS_CACERT variable set.

The only manual configuration required is to edit $HOME/.kube/config and set the IP address of your new K8S master.

To allow other users to access your K8S cluster and operate on a subset of its resources, edit the auth-policy file with:

```
# kubectl -n kube-system edit configmap k8s-auth-policy
```

modify in the first block the line “resources” and replace “type”: “role”, “values”: [“k8s-user”] with e.g. “type”: “user”, “values”: [“username1”, “username2”]

11.5.2 Kubernetes Dashboard

The cluster exposes the following dashboards:

- K8S dashboard: https://master_ip:30900
- Prometheus UI: http://master_ip:30901
- Alertmanager UI: http://master_ip:30902
- Grafana UI: http://master_ip:30903

To login into the K8S dashboard use the token of the kube-system:default service account. To get it, execute the following command from your environment, or from the master node:

```
# kubectl -n kube-system describe secret kubernetes-dashboard
```

(continues on next page)
11.6 Testing your Kubernetes cluster

The cluster comes up by default with two K8S operators implementing the popular Big Data Analytics and Streaming platforms Apache Spark and Kafka (you can avoid this by removing the roles spark and kafka in the file deploy_k8s.yaml).

### 11.6.1 Launching the Spark application spark-pi

You can use the Spark application spark-pi to verify that the cluster works properly. Just take the examples/spark-pi.yaml file and execute the following kubectl commands:

```bash
# kubectl apply -f spark-pi.yaml
# kubectl get sparkapplications spark-pi
NAME   AGE
spark-pi 5m

# kubectl describe sparkapplications spark-pi
Name: spark-pi
Namespace: default
Labels: none
Annotations: kubectl.kubernetes.io/last-applied-configuration:
    "apiVersion": "sparkoperator.k8s.io/v1beta1","kind": "SparkApplication
    ...

API Version: sparkoperator.k8s.io/v1beta1
Kind: SparkApplication

# kubectl logs -f spark-pi-driver | grep "Pi is roughly"
Pi is roughly 3.1458557292786464
```

Note: In case of problems with the sparkoperator API Version, look at the output of

```bash
# kubectl logs -f spark-pi-driver | grep "Pi is roughly"
```

Note: To login into the Grafana dashboard as administrator use the credentials: username=admin and password=admin. The first login requires the changing of the default password for security reasons.
# kubectl api-versions

## 11.6.2 Creating a Kafka cluster with a topic

Declare the cluster structure as in the kcluster.yaml file taken from the examples directory, and execute the following kubectl command:

```shell
# kubectl apply -f kcluster.yaml
```

For further details on configuration see [https://strimzi.io/docs/master/#assembly-deployment-configuration-str](https://strimzi.io/docs/master/#assembly-deployment-configuration-str)

A topic for the Kafka cluster can be declared as in the ktopic.yaml file taken from the examples directory, and created by executing the following kubectl command:

```shell
# kubectl apply -f ktopic.yaml
```

Kubernetes provides a port on the master for accessing the created cluster. The port number is reported by the following kubectl command:

```shell
# kubectl get service kcluster-kafka-external-bootstrap -o=jsonpath='{.spec.ports[0].nodePort}'
```

Other useful commands for monitor the status of the cluster are:

```shell
# kubectl get service
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>kcluster-kafka-0</td>
<td>NodePort</td>
<td>10.97.1.118</td>
<td>&lt;none&gt;</td>
<td>64s</td>
</tr>
<tr>
<td>9094:31945/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-kafka-1</td>
<td>NodePort</td>
<td>10.100.252.199</td>
<td>&lt;none&gt;</td>
<td>64s</td>
</tr>
<tr>
<td>9094:31730/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-kafka-2</td>
<td>NodePort</td>
<td>10.106.128.149</td>
<td>&lt;none&gt;</td>
<td>64s</td>
</tr>
<tr>
<td>9094:31608/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-kafka-bootstrap</td>
<td>ClusterIP</td>
<td>10.109.113.86</td>
<td>&lt;none&gt;</td>
<td>65s</td>
</tr>
<tr>
<td>9091/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-kafka-brokers</td>
<td>ClusterIP</td>
<td>None</td>
<td>&lt;none&gt;</td>
<td>65s</td>
</tr>
<tr>
<td>9091/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-kafka-external-bootstrap</td>
<td>NodePort</td>
<td>10.107.133.0</td>
<td>&lt;none&gt;</td>
<td>64s</td>
</tr>
<tr>
<td>9094:32161/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-zookeeper-client</td>
<td>ClusterIP</td>
<td>10.103.223.73</td>
<td>&lt;none&gt;</td>
<td>93s</td>
</tr>
<tr>
<td>2181/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kcluster-zookeeper-nodes</td>
<td>ClusterIP</td>
<td>None</td>
<td>&lt;none&gt;</td>
<td>93s</td>
</tr>
<tr>
<td>2181/TCP,2888/TCP,3888/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kubernetes</td>
<td>ClusterIP</td>
<td>10.96.0.1</td>
<td>&lt;none&gt;</td>
<td>3d1h</td>
</tr>
<tr>
<td>443/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```shell
# kubectl get pod
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>kcluster-entity-operator-7b8d767b5c-1h6kp</td>
<td>3/3</td>
<td>Running</td>
<td>0</td>
<td>3m55s</td>
</tr>
<tr>
<td>kcluster-kafka-0</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m28s</td>
</tr>
<tr>
<td>kcluster-kafka-1</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m28s</td>
</tr>
<tr>
<td>kcluster-kafka-2</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m28s</td>
</tr>
<tr>
<td>kcluster-zookeeper-client</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m56s</td>
</tr>
<tr>
<td>kcluster-zookeeper-0</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m56s</td>
</tr>
<tr>
<td>kcluster-zookeeper-1</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>4m56s</td>
</tr>
</tbody>
</table>

(continues on next page)
# Testing your Kubernetes cluster

```
# kubectl get kafkatopics
NAME    AGE
ktopic   12s
```

```
kcluster-zookeeper-2  2/2  Running  0  4m56s
strimzi-cluster-operator-6464cfdf94cf-tmbqd  1/1  Running  0  3dlh
```
CHAPTER 12

Some basics on Linux administration

12.1 Introduction

The possession of great power necessarily implies great responsibility.

-- William Lamb, 2nd Viscount Melbourne

CloudVeneto provides an infrastructure where your virtual machines can live. After you have activated your virtual machine(s) you are on your own for the most part of the day to day administration tasks.

Note: We will only focus on Linux VMs, showing differences between the RedHat (CentOS, Fedora, ...) and Debian (Ubuntu, Mint, ...) distributions.

Throughout this chapter we will address the former with RH and the latter with DEB.

Some of these everyday tasks might be:

- Use your VM as root only when needed;
- Installing/deinstalling software;
- Adding another user to your VM;
- Formatting the volume you just attached to your VM;
- Automatically remount the volume on next reboot.

In the following sections we provide some very small introductory instructions on performing such tasks.

12.2 Setting up ‘sudo’

Nobody (not even administrators) use a Unix/Linux system always as root.

If you do you should stop immediately (no jokes!).
Normally you have your user with limited privileges and, when needed, you use `su` (which stands for ‘switch user’) to become root, perform the privileged task and then you go back to the normal user.

A more flexible approach is using `sudo` (Super User DO) to perform the ‘one shot’ task or to allow certain users to perform only some tasks as the superuser. The configuration of sudo is performed by modifying the `/etc/sudoers` file or (better) by adding a file in the `/etc/sudoers.d` directory.

Follow these instructions to allow the user `paolo` (change this to your username) to perform any command as the superuser providing his own password, or to modify your user privileges (in this case there is already a file with your username in the `/etc/sudoers.d` directory):

- Become the root user:
  - RH: using `su` and providing the root password
  - DEB: using `sudo su -` and providing your password
- Create the file `/etc/sudoers.d/paolo` using your preferred editor;
- Add this line inside the file:
  ```plaintext
  paolo ALL = (ALL) ALL
  ```
  If you want the user to do everything without even providing the password put this line instead:
  ```plaintext
  paolo ALL = NOPASSWD: ALL
  ```

- Change the file permissions: `chmod 0440 /etc/sudoers.d/paolo`

Now when paolo logs in to the VM he is allowed to perform superuser tasks by prefixing the command with `sudo`.

If you want to limit the user to perform only certain commands (or classes of commands, e.g. installing/deinstalling software) you can look at the sudo documentation on your VM using

```
man sudoers
```

### 12.3 Managing software through the package manager

RedHat and Debian based linux distributions both have their software management system. On RH each software is packaged in rpm form (RPM stands for RedHat Package Manager) while DEB uses deb packages.

Package contents are not only copied on the VM when installed, but are also listed on a database that can be queried to search for new software, to find out which package installed which files and so on. (Note: there is no such functionality on Windows servers . . .).

Rather than manipulating the packages directly with the rpm (RH) or dpkg (DEB) commands, you will mostly use a command line package manager such as yum or apt.

We will now try to install the wget package on a RH and a DEB system.

#### 12.3.1 Managing software on RH distributions

Let’s try to install the wget software on CentOS or Fedora linux.

We will use `yum` (dnf on Fedora 21) and `rpm` for the task.

Since we will be performing operations as the superuser, if you haven’t already, please set up `sudo` first.

1. Query the system to search for wget (no need to be root for that):
2. Install the wget package (note that we are using sudo here):

```
[paul@maz03 ~]$ sudo yum install wget
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mi.mirror.garr.it
* extras: mi.mirror.garr.it
* updates: mi.mirror.garr.it
Resolving Dependencies
---> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be installed
---> Finished Dependency Resolution

Dependencies Resolved

Package Arch  Version Repository Size
-----------------------------------------------
Installing:
wget x86_64 1.14-10.el7_0.1 base 545 k

Transaction Summary
-----------------------------------------------
Install 1 Package

Total download size: 545 k
Installed size: 2.0 M
Is this ok [y/d/N]: y
```

Downloaded packages:
wget-1.14-10.el7_0.1.x86_64.rpm | 545 kB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : wget-1.14-10.el7_0.1.x86_64 1/1
  Verifying : wget-1.14-10.el7_0.1.x86_64 1/1

Installed:

Name and summary matches only, use "search all" for everything.

---
3. Query the rpm database to see what has been installed:

```
[paolo@maz03 ~]$ rpm -ql wget
/etc/wgetrc
/usr/bin/wget
/usr/share/doc/wget-1.14
/usr/share/doc/wget-1.14/AUTHORS
/usr/share/doc/wget-1.14/COPYING
/usr/share/doc/wget-1.14/MAILING-LIST
/usr/share/doc/wget-1.14/NEWS
/usr/share/doc/wget-1.14/README
/usr/share/doc/wget-1.14/sample.wgetrc
/usr/share/info/wget.info.gz
/usr/share/locale/be/LC_MESSAGES/wget.mo
.....
/usr/share/locale/sv/LC_MESSAGES/wget.mo
/usr/share/locale/tr/LC_MESSAGES/wget.mo
/usr/share/locale/uk/LC_MESSAGES/wget.mo
/usr/share/locale/vi/LC_MESSAGES/wget.mo
/usr/share/locale/zh_CN/LC_MESSAGES/wget.mo
/usr/share/locale/zh_TW/LC_MESSAGES/wget.mo
/usr/share/man/man1/wget.1.gz
```

4. You now decide you don’t need wget anymore. Remove the package (root needed!):

```
[paolo@maz03 ~]$ sudo yum remove wget
```

```
Loaded plugins: fastestmirror
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package Arch  Version Repository Size
================================================================================
Removing:
wget x86_64 1.14-10.el7_0.1 @base 2.0 M

Transaction Summary
================================================================================
Remove 1 Package

Installed size: 2.0 M
Is this ok [y/N]: y
```

(continues on next page)
12.3.2 Managing software on DEB distributions

Let’s try to install the wget software on Debian or Ubuntu Linux.

We will use `apt` and `dpkg` for the task.

Since we will be performing operations as the superuser, if you haven’t already, please set up `sudo` first.

1. Update your local cache of available softwares (superuser privileges needed):

   ```bash
   ubuntu@maz03:~$ sudo apt-get update
   sudo: unable to resolve host maz03
   Ign http://nova.clouds.archive.ubuntu.com trusty InRelease
   Ign http://nova.clouds.archive.ubuntu.com trusty-updates InRelease
   Hit http://nova.clouds.archive.ubuntu.com trusty Release.gpg
   Hit http://nova.clouds.archive.ubuntu.com trusty Release
   Ign http://security.ubuntu.com trusty-security InRelease
   ....
   ....
   Fetched 10.2 MB in 3s (3257 kB/s)
   Reading package lists... Done
   ```

2. Query the cache for wget (no privileges needed).

   Note that DEB systems also query descriptions and ‘related’ softwares.

   ```bash
   ubuntu@maz03:~$ apt-cache search wget
   devscripts - scripts to make the life of a Debian Package maintainer easier
texlive-latex-extra - TeX Live: LaTeX additional packages
wget - retrieves files from the web
abcde - A Better CD Encoder
apt-mirror - APT sources mirroring tool
apt-ftp - Update a non-networked computer using apt and removable media
axel - light download accelerator - console version
axel-db - light download accelerator - debugging symbols
axel-kapt - light download accelerator - graphical front-end
filetea - Web-based file sharing system
getdata - management of external databases
libcupt3-0-downloadmethod-wget - alternative front-end for dpkg -- wget download
puf - Parallel URL fetcher
pwget - downloader utility which resembles wget (implemented in Perl)
snarf - A command-line URL grabber
wput - tiny wget-like ftp-client for uploading files
   ```

3. Install wget as the superuser:

12.3. Managing software through the package manager 119
4. Query the deb database and see what files have been installed by wget:

```
ubuntu@maz03:~$ dpkg -L wget
/. 
/usr 
/usr/bin 
/usr/bin/wget 
/usr/share 
/usr/share/man 
/usr/share/man/man1 
/usr/share/man/man1/wget.1.gz 
/usr/share/info 
/usr/share/info/wget.info.gz 
/usr/share/doc 
/usr/share/doc/wget 
/usr/share/doc/wget/copyright 
/usr/share/doc/wget/AUTHORS 
/usr/share/doc/wget/NEWS.gz 
/usr/share/doc/wget/MAILING-LIST 
/usr/share/doc/wget/README 
/usr/share/doc/wget/changelog.Debian.gz 
/etc 
/etc/wgetrc
```

5. You now decide you don’t need wget anymore. Remove the wget software from the system (keep config files).

Note: you can alternatively ‘purge’ the software completely as described below.

```
ubuntu@maz03:~$ sudo apt-get remove wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  wget
0 upgraded, 0 newly installed, 1 to remove and 25 not upgraded.
After this operation, 651 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```
6. Discover which files have been left behind by the wget software:

```bash
ubuntu@maz03:~$ dpkg -L wget
/etc
/etc/wgetrc
```

7. Completely remove (purge) all the files installed by wget:

```bash
ubuntu@maz03:~$ sudo apt-get purge wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  wget*
0 upgraded, 0 newly installed, 1 to remove and 25 not upgraded.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
(Reading database ... 51119 files and directories currently installed.)
Removing wget (1.15-1ubuntu1.14.04.1) ...
Purging configuration files for wget (1.15-1ubuntu1.14.04.1) ...
```

12.4 Adding a user to your VM

You may need to give access to your VM to another user. Given that there are no graphical tools or fancy icons to do the task you are going to use some command line tools.

We are going to add the user ‘pemazzon’ (Paolo E. Mazzon) to your system.

1. `$ sudo useradd -m -c 'Paolo E. Mazzon' pemazzon`

The meaning of parameters is:

- `-m` = create a ‘home directory’ for the user under /home
- `-c` = set this as a description of the user

**Warning:** It may be necessary to enable password authentications through ssh. Check the file `/etc/ssh/sshd_config` and be sure that you have `ChallengeResponseAuthentication yes` inside. If you modified that file restart the ssh service using

DEB systems: `sudo restart ssh`

or

RH systems: `sudo systemctl restart sshd`

2. Set a password for the user: you will decide a password that will be valid just for the first login. You will force the user to change it immediately.
CloudVeneto User Guide, Release 3.11

$ sudo passwd pemazzon
... enter twice times the password you want to set for the user ...

3. Force the user to change his password on first logon:

$ sudo chage -d 0 pemazzon

4. Mail the user the password you have set.

12.5 Formatting/resizing a volume you just attached

We already showed on Using (attaching) a Volume how to start using a volume you have attached to your VM. We will give you here some more details.

If you just created an empty volume you first need to create a filesystem on it before you can put some data inside. The volume you just attached is merely ‘raw space’ and has no concept about files and directories.

You may also think about partitioning your volume, e.g. to split volume space in ‘slices’, as you may have done installing linux.

Given that in the CloudVeneto you can add as many volumes you want (up to your volume quota, of course) partitioning a volume is simply not recommended.

Suppose now that you have filled the volume space. You have the option to resize it from the cloud dashboard but the result may not be the one you expect until you do some operations from inside your VM.

We are going to resize the volume ‘test’ from 2 to 4 GB and use the newly available space on a VM.

We will create the volume from scratch. Obviously you can skip these 3 steps if you are resizing an existing volume.

1. Create a 2 GB volume named ‘test’ and attach it to one of your VM as described in Volumes
2. Create a filesystem and mount it as described in Using (attaching) a Volume
3. Check the available space is 2 GB and the filesystem is filling up the partition

ubuntu@maz03:~$ sudo fdisk -l /dev/vdb

Disk /dev/vdb: 2154 MB, 2154823680 bytes
15 heads, 30 sectors/track, 9352 cylinders, total 4208640 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/vdb doesn't contain a valid partition table

ubuntu@maz03:~$ df -k /mnt

Filesystem 1K-blocks Used Available Use% Mounted on
/dev/vdb 2005688 3096 1880992 1% /mnt

Let's resize the volume

1. Umount it first from the VM (if mounted):
2. Detach it from the VM using the dashboard: use “Edit Attachments” and confirm your request.

3. When the volume is detached the “Extend Volume” option will be available. Select it...

4. ... and grow the volume to, say, 4GB:
5. Now attach again the volume to the VM and let’s check, from inside the VM, what’s happening:

```
ubuntu@maz03:~$ sudo mount /dev/vdb /mnt
ubuntu@maz03:~$ sudo fdisk -l /dev/vdb

Disk /dev/vdb: 4309 MB, 4309647360 bytes
16 heads, 63 sectors/track, 8350 cylinders, total 8417280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/vdb doesn't contain a valid partition table
```

The disk size is now 4309 MB, so the system recognize the fact that the volume have grown.

Let’s check the available space:

```
ubuntu@maz03:~$ df -k /mnt
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/vdb 2005688 3096 1880992 1% /mnt
```

we see here that it is still 2 GB! This is due to the fact that the filesystem has not been touched by the resize operation: the volume service of the cloud has no knowledge of what’s inside your volume.

To use the new space we need to resize the filesystem, obviously from inside the VM, to let it span all the volume:

```
ubuntu@maz03:~$ sudo umount /mnt
ubuntu@maz03:~$ sudo resize2fs /dev/vdb
resize2fs 1.42.9 (4-Feb-2014)
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vdb is now 1052160 blocks long.

ubuntu@maz03:~$ sudo mount /dev/vdb /mnt
ubuntu@maz03:~$ df -k /mnt
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/vdb 4078888 4120 3873956 1% /mnt
```

You can now see you have all the 4 GB available.
12.6 Automatically remount volumes on reboot

Connecting a volume to your VM using the ‘mount’ command is a one-shot solution. If you need to reboot your VM for some reason you will have to re-issue the command again.

Forget to do so might cause the following:

1. You write data under the /mnt directory (or wherever you mount your volume) thinking you are writing on your volume with, say, 1 TB of space;
2. The volume is not mounted there so you are writing instead on the same space where your operating system lives;
3. You eventually fill up your filesystem and your VM crash/starts malfunctioning;
4. Your VM might not boot anymore and you have to call for help.

We will now create an entry on the /etc/fstab file to remount the volume upon reboot.

**Warning:** A big warning! DO NOT edit the /etc/fstab file by transferring it on a windows machine and then back to your VM. Bad things will happen…

The /mnt directory is normally used as the ‘mount point’ for various devices. Normally you would create a directory under /mnt for each device and attach the device on that directory. Obviously this is not mandatory: you can mount filesystems almost everywhere (e.g. /data, /opt/myprograms and so on.)

All the operations will be performed as the supersuser.

1. Acquire root privileges

   ubuntu@maz03:~$ sudo su -
   root@maz03:~#

2. Create the ‘mount point’

   root@maz03:~# mkdir -p /mnt/volume1

3. Edit the /etc/fstab file: we will use the ‘nano’ editor for that:

   root@maz03:~# nano /etc/fstab

   Your screen should look like this one:
4. Add a line telling you want to mount the device /dev/vdb under /mnt/volume1 (you have already created an ext4 filesystem on it).

   This should be the content of your file:

   ![Nano editor screen showing the `fstab` file with the line `LABEL=cloudimg-rootfs / ext4 defaults 0 0` and additional line `LABEL=cloudimg-rootfs /dev/vdb /mnt/volume1 ext4 defaults 0 0`]

5. Write your file to disk by pressing **CTRL+o** ... ... and confirming with enter.
6. Exit the editor by pressing CTRL+x. Go back to your normal user by issuing the ‘exit’ command or by pressing CTRL+d

Now your volume will appear under the ‘/mnt/volume1’ directory everytime your VM boots up. You can also mount the volume just issuing

```
sudo mount /mnt/volume1
```

The system will lookup in /etc/fstab and mount the correct volume corresponding to the /mnt/volume1 mount point.

### 12.7 Specific instructions relevant for INFN-Padova users

In this section we discuss about some specific topics relevant only for INFN-Padova users.

#### 12.7.1 Enabling INFN Padova LDAP based authentication on the Virtual Machine

When creating a custom image, it might be needed to enable a LDAP server to manage authentication for users. This section explains how to enable the INFN Padova’s LDAP server for user authentication on the VMs of the Cloud. To do that, the following LDAP client configurations, targeted to SL6.x systems, need to be available on the image used to start the VMs.

First of all, the following packages must be installed:

- openssl
- openldap
- openldap-clients
- pam-ldap
- nss-pam-ldapd
- nss-tools
Then the following files (included in this ldap.tar tar file) must be installed on the Virtual Machine:

- /etc/openldap/cacerts/cacert.pem
- /etc/openldap/ldap.conf
- /etc/pam_ldap.conf
- /etc/nsswitch.conf
- /etc/nslcd.conf
- /etc/pam.d/system-auth-ac
- /etc/pam.d/password-auth-ac

To do that, it is enough to log on the VM and:

```
cd /
tar xvf /path/ldap.tar
```

Make sure that the following links exist:

```
/etc/pam.d/password-auth -> password-auth-ac
/etc/pam.d/system-auth -> system-auth-ac
```

Then it is necessary to start the nslcd and nscl services:

```
service nslcd start
service nscl start
chkconfig nslcd on
chkconfig nscl on
```

Then it is just necessary to “enable” the relevant accounts on the VM adding in the /etc/passwd file:

```+name1:::::::+
+name2:::::::+
...```

and creating their home directories.

You might also want to set a different home directory wrt the one specified in LDAP, e.g.:

```+pippo:::::::/ehome/pippo:
```

Changes done in /etc/passwd could not be applied immediately by the system. In this case a:

```
nscl -i passwd
should help.
```

**Note:** Please note that the `SL6x-INFNPadova-x86-64-<date>` and `CentOS7x-INFNPadova-x86-64-<date>` images have already the LDAP client properly configured to use the Padova LDAP server. Using these images it is just necessary to enable the relevant users in /etc/passwd and create their home directories.

### 12.7.2 Install Mathematica (only for INFN Padova users)

INFN-Padova users can follow these instructions to install Mathematica on their Linux box.
First of all mount on your VM the directory where the installation kit is available. As root:

```
mount hellgate.pd.infn.it:/sw/linux /mnt
```

For Ubuntu systems:

```
sudo mount hellgate.pd.infn.it:/sw/linux /mnt
```

Go to the relevant directory and launch the installer. As root:

```
cd /mnt/Mathematica/<version>/Installer
./MathInstaller
```

For Ubuntu systems:

```
cd /mnt/Mathematica/<version>/Installer
sudo ./MathInstaller
```

The installer will ask the target installation directory (the default is /usr/local/Wolfram/Mathematica/<version>) and the directory where the executables will be linked (default /usr/local/bin).

Finally create the file /usr/local/Wolfram/Mathematica/<version>/Configuration/Licensing/mathpass with the following content:

```
!mathlm.pd.infn.it
```

That’s all! You can now umount the /mnt directory:

```
cd ; umount /mnt
```
CHAPTER 13

Indices and tables

- genindex
- search