TÉCNICO LISBOA

AGISIT 20/21	LAB ASSIGNMENT	Guide:	05
Cloud-based Infrastructures		Issue Date:	26 Oct 2020
IaaS (Infrastructure as a Service) - VMCloud - Part 1		Report Due:	02 Nov 2020
Author: Prof. Rui Cruz		Due Date:	03 Nov 2020

1 Introduction

This lab experiment is the first of a series exploring deployment of a Cloud-based Infrastructure. The goal of the experiments in this Lab is to learn about the interactive deployment of a Private Cloud using manual configuration.

A Private cloud is a Cloud Platform operated inside a private network. For that purpose we will use **VMCloud**, a Cloud Platform of Instituto Superior Técnico, based on the **OpenStack** framework.

OpenStack is an open-source framework that began as a joint project between NASA and Rackspace, originally intended to be an alternative for, but compatible with, the Amazon Elastic Compute Cloud (EC2). **OpenStack** offers virtualisation of *compute*, *storage*, *networking*, and many other resources. Each component in **OpenStack** manages a different resource that can be virtualised for the end user. By separating each of the resources that can be virtualised into separate components, makes **OpenStack** a highly modular architecture (see Figure 1).

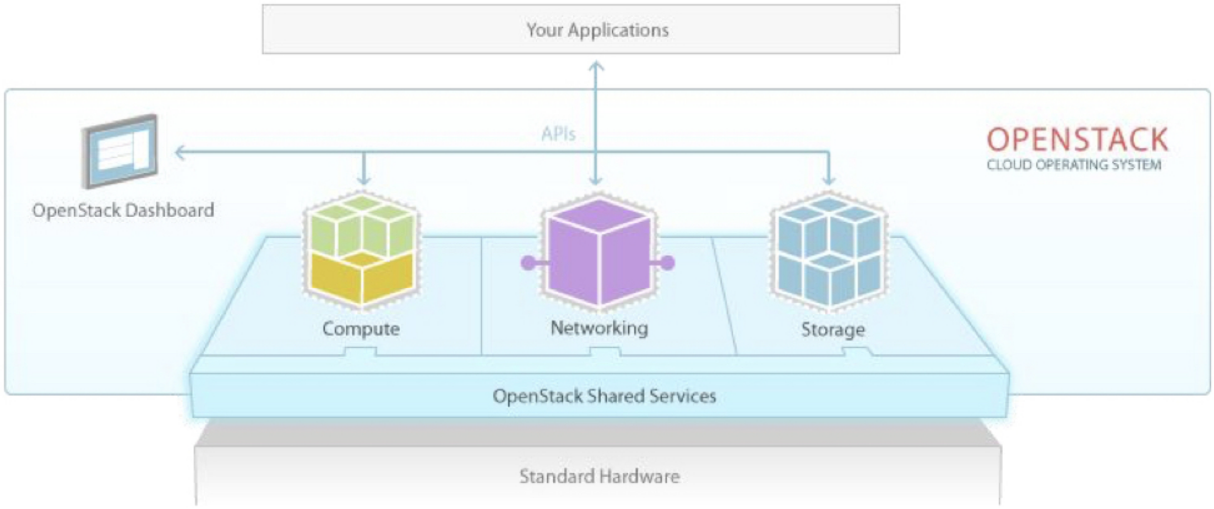


Figure 1: OpenStack component groups

Logically, the components of **OpenStack** can be divided into four main groups: **Control**, **Networking**, **Compute** and **Storage**. The **Control** tier runs the Application Programming Interfaces (API) services, web interface, database, and message bus. The **Networking** tier runs network service agents for networking. The **Compute** tier is the virtualization hypervisor, with services and agents to handle virtual machines. The **Storage** tier manages *block* (Volumes; partitions) and *object* (containers; files) storage for the **Compute** instances.

The base components of **OpenStack**, illustrated in Figure 2, are the following:

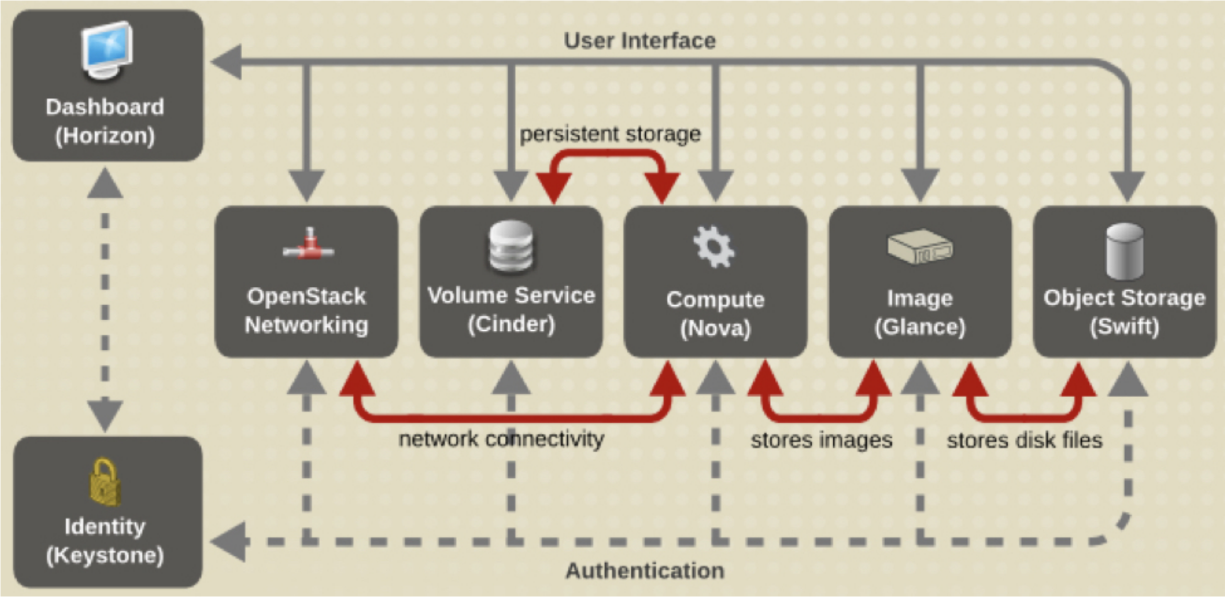


Figure 2: Base components of OpenStack

Horizon (dashboard): is the dashboard web interface component. The dashboard cannot do anything that the API cannot do. All the actions that are taken through the dashboard result in calls to the API to complete the task requested.

Keystone (identity): is the identity management component. Besides authentication, *Keystone* manages *tenants*, users, and roles and a catalog of services and endpoints for all the components. Everything in **OpenStack** must exist in a *tenant*. A *tenant* is a grouping of objects (such as users, instances, and networks) and objects cannot exist outside of a *tenant*. Another name for a *tenant* is *project* (on the command line, the term *tenant* is used, but in the dashboard web interface, the term used is *project*). Users must be granted a role in a *tenant*. Even the administrator of an **OpenStack** Cloud Platform has a *tenant*. The “users” that the **OpenStack** components use to communicate with each other have to be members of a *tenant* to be able to authenticate. Keystone uses username/password authentication to request a token and Public Key Infrastructure (PKI) tokens for subsequent requests.

Glance (image): is the image management component – a registry of pre-installed disk images to boot from. In preparation for a compute instance to launch, a copy of a selected image is first cached to the compute node where the instance is being launched, and then, a copy is made to the ephemeral disk location of the new instance. The disk images have the operating system installed but with some things removed, such as Secure Shell (SSH) host key, and network device MAC addresses.

Neutron (networking): is the network management component. Neutron is an API frontend (and a set of agents) that manages the Software Defined Networking (SDN) infrastructure. Using Neutron, it means that each of the tenants can create virtual isolated networks that can be connected to virtual routers to create routes between the virtual networks. A virtual router can have an external gateway connected to it, and external access can be given to each instance by associating a *floating IP* on an external network. Neutron then puts all configuration in place to route the traffic sent to the *floating IP* address through these virtual network resources into a launched instance. This is also called Networking as a Service (NaaS). NaaS is the capability to provide networks and network resources on demand via software.

Nova (compute instances): is the instance management component. An instance can be launched once there's an image, a network, a key pair, and a security group available. The resource's identifiers are provided to Nova, which looks at what resources being used on the hypervisors, and schedules the instance to spawn on a compute node. The compute node gets the image, creates the virtual network devices, and boots the instance.

Cinder (block storage): is the block storage management component. Volumes can be created and attached to instances. On the instance, the block device can be partitioned and a file system can be created and mounted. Cinder also handles snapshots of the instance.

Swift (object storage): is the object storage management component. Object storage is a simple content-only storage system. Files are stored without the metadata that a block filesystem has. The objects are simply containers and files.

Some other components of interest are:

Ceilometer (telemetry): is the telemetry component, collecting resource measurements and monitor the **OpenStack** Platform. Ceilometer was originally designed as a metering system for billing the users. When Ceilometer reads a meter, it's called a sample. The samples can also be used for alarms. Alarms are monitors that watch for a certain criterion to be met.

Heat (orchestration): is the orchestration component. Orchestration is the process of launching multiple instances that are intended to work together. Orchestration uses templates (files), to define what will be launched.

Preliminary Notes

One nice feature of the software stack we are going to use is that it is portable to many platforms including **YOUR OWN** personal computers, running the following Operating Systems:

- Microsoft Windows from version 10 up
- Apple macOS from versions 10.13 ’High Sierra’ up
- Debian-based Linux, such as Ubuntu (recommended) from versions 12.04 ’Precise’ up.

It is **not recommended** to apply this setup to a virtual machine (**nested virtualization**), although possible, as the configuration requires access to the Hypervisor environment (mandatory Virtualbox) in the host system.

Before proceeding you should verify if you have a “clean” environment, i.e., no Virtual Machine “instances” running (using precious resources in your system), or inconsistent instances in Vagrant and Virtualbox.

For that purpose run the `vagrant global-status` command and observe the results (as in the following example):

```
:~$ vagrant global-status
id       name      provider   state    directory
---------------------------------------------------------------
28fb48a  mininet   virtualbox poweroff /Users/x/Projects/mininet
f0ccec2  web1      virtualbox running  /Users/x/Projects/multinode
f09c279  web2      virtualbox running  /Users/x/Projects/multinode
```

In the above example, you can observe that there are three Virtual Machines, being the first “mininet”, which is **powered off**, but two “web” servers **still running**. It is **advisable to halt those VMs** if running, and then eventually **clean and destroy the VMs** if not needed anymore.

Note: Avoid copying text strings from the examples or configurations illustrated in this document, as pasting them into your system or files may introduce/modify some characters, leading to errors or inadequate results.

AGISIT 20/21	**LAB ASSIGNMENT**	**Guide:**	05
Cloud-based Infrastructures		**Issue Date:**	26 Oct 2020
IaaS (Infrastructure as a Service) - VMCloud - Part 1		**Report Due:**	02 Nov 2020
Author: Prof. Rui Cruz		**Due Date:**	03 Nov 2020

2 Submitting the Results of the Experiments

The experiments that you will execute in this LAB will either produce results that you need to report or from which you will be asked questions about the execution of some procedures. In order to report the results you will achieve, proceed as follows:

2.1 General Procedure

On your system, it is advisable to **create a working folder for this lab** (not the experiment folder!), that will contain the necessary files for reporting. These files may be screenshots, code snippets, text documents, system logs, etc.

It is advisable to give specific and good identifying names to those files, following what will be asked to report. For example, you may be asked to take several screenshots during the procedures, and so, rename those screenshots to the specific items being requested in the assignment. It is also advisable to collect results in a text file or document file while doing the experiment, so that you will have those results when you will need to submit them in the online assignment form.

The procedure for submission is quite simple:

1. This Lab assignment MUST be reported up to the **Report Due** date;
2. In the Moodle (`https://moodle.dei.tecnico.ulisboa.pt/login/index.php`) for the AGISIT course, you will find an **Assignment** for reporting the results from this specific Lab experiment.
3. The Assignment Link opens a **TSUGI CLOUD Web Form** containing the Questions to be answered;
4. The Assignment **Due Date** corresponds to the date it MUST be considered completed.
5. When you are prepared with the requested materials (screen-shots, command line outputs, developed code, etc.) you will submit the items into the respective Exercise Form question boxes of the Assignment;
6. At the end of the Exercise Form you may comment your submission, or respond to feedbacks received (quoting them);
7. When finished answering the Exercise Form, click the button **Done** in the top left of the Form; You will be returned to the Moodle Assignment Link;
8. Please note that this process involves a **Peer Review** mechanism, in which you will be asked to provide **feedback** (anonymously) to the Reports of your classmates. So, after submitting your own assignment, get back to it to see the Reviews that the system have distributed automatically to you;

2.2 Specific Procedure

For this Lab Experiment you will need to submit the following items:

1. Capture a screenshot of the VMCloud Dashboard showing the **Overview of your Project**, and submit in the TSUGI Form where asked;
2. Capture a screenshot of the VMCloud Dashboard showing the **Network Topology** of your Project **before creating any instances**, and submit in the TSUGI Form where asked;
3. Capture a screenshot of the VMCloud Dashboard showing the **Network Topology** of your Project **after creating the instance**, and submit in the TSUGI Form where asked;
4. Post the information about **Security Group**s applied to your VM instance in the TSUGI Form where asked;
5. Post the content of the Log collected from your **Instance Console Log** in the TSUGI Form where asked;
6. Capture a Screenshot of the **Console** of the instance you created, **after log in**, and submit in the TSUGI Form where asked;
7. Capture a Screenshot of the **Console** of the instance you created, after executing the commands “ifconfig” and “uname -a”, and submit in the TSUGI Form where asked;
8. Capture a Screenshot of the the **Overview of your Project** showing the **resources used** in the Project, and submit in the TSUGI Form where asked;
9. Report the **Images and their sizes**, available to create Instances in VMCloud, to submit in the TSUGI Form where asked;
10. Capture a Screenshot of the VMCloud Dashboard showing the **Overview of your Project**, **after having deleted your Instances**, and submit in the TSUGI Form where asked;

WARNING. Submissions MUST BE MADE in the TSUGI CLOUD Web Form you can find in Moodle for this course. No other type of submission will be considered.

AGISIT 20/21	LAB ASSIGNMENT	Guide:	05
Cloud-based Infrastructures		Issue Date:	26 Oct 2020
IaaS (Infrastructure as a Service) - VMCloud - Part 1		Report Due:	02 Nov 2020
Author: Prof. Rui Cruz		Due Date:	03 Nov 2020

3 Setting Up a Tenant Infrastructure

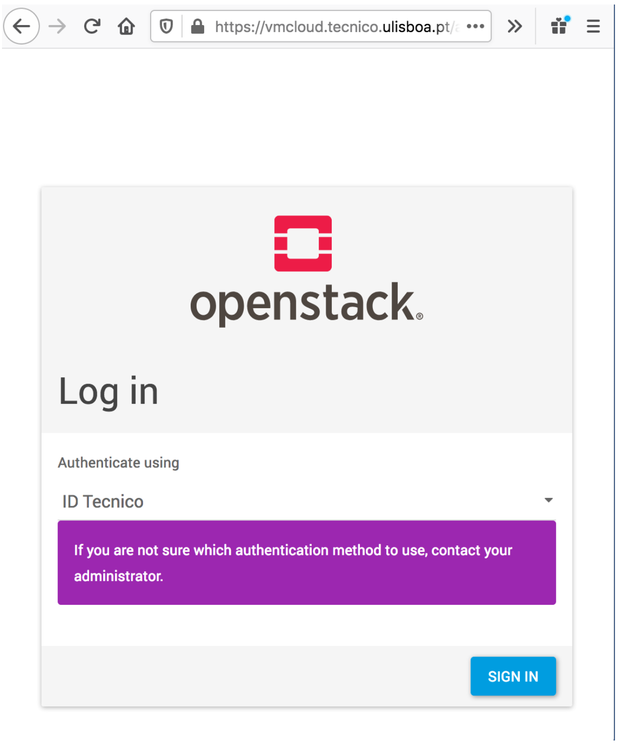
The **VMCloud** of Instituto Superior Técnico, is a Private Cloud Platform, available for internal users of the University.

To familiarize yourself with the process, we will start by manually provisioning the resources, within the limits set by the Platform administrator, using the **VM Cloud** dashboard.

Each Working Group of students of the course will access their Tenant (Project) specifically created for the Group at the VMCloud Platform. The assigned Tenants and the corresponding user names are as follows:

Group Name	Tenant Name
Group 1	gp-AGISIT7-AGISIT-Teams-1
...	...
Group 43	gp-AGISIT7-AGISIT-Teams-43

Using a browser, enter the URL for the VM Cloud dashboard (the *Horizon*), and login with your **IST ID** credentials (Figure 3): `https://vmcloud.tecnico.ulisboa.pt`



https://vmcloud.tecnico.ulisboa.pt/

openstack

Log in

Authenticate using

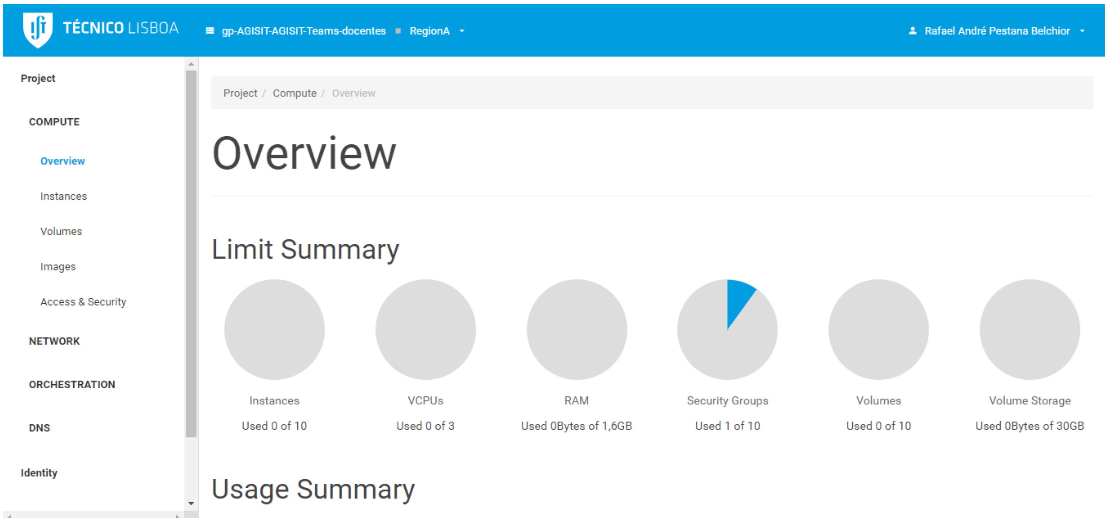
ID Tecnico

If you are not sure which authentication method to use, contact your administrator.

SIGN IN

Figure 3: VM Cloud Login page

After a successfully login the dashboard will present a menu on the left with the available **management Tabs** for your **Project**, the **Compute** instances, the **Network**, the **Volumes** (Object Store) and the **Identity**, as illustrated in Figure 4.



Overview

Limit Summary

Instances	VCPUs	RAM	Security Groups	Volumes	Volume Storage
Used 0 of 10	Used 0 of 3	Used 0Bytes of 1,6GB	Used 1 of 10	Used 0 of 10	Used 0Bytes of 30GB

Usage Summary

Figure 4: VMCloud Dashboard

You can now navigate to the **Network** tab to see the already pre-assigned **front-end** network (coloured Orange) which is "routed" to the **provide**r network, as illustrated in Figure 5.

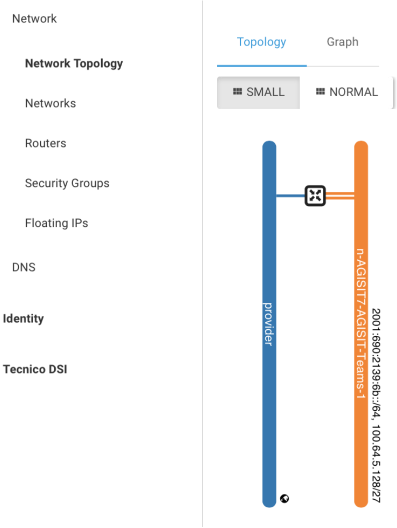


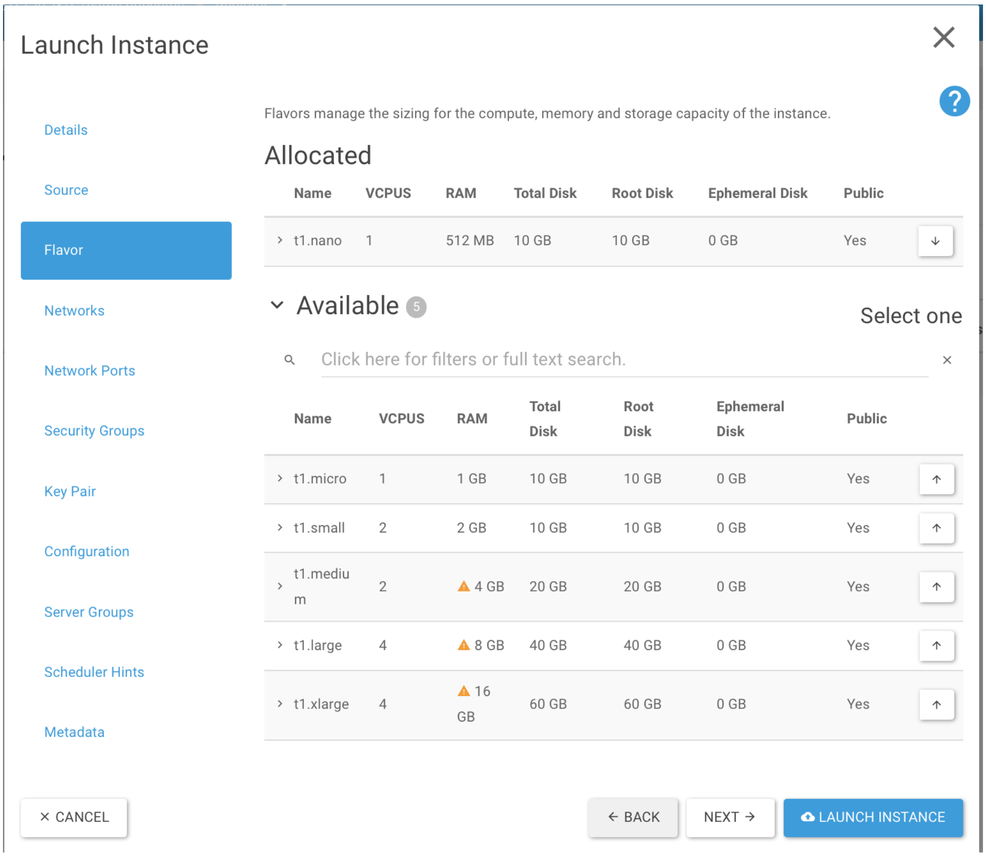
Figure 5: VMCloud Network pre-assigned

3.1 Create Compute Instances

As you already have pre-assigned networks, you can now create some compute instances (servers) and connect them to the front-end network.

For that purpose select the tab **Instances**, of the submenu **Compute**, of **Project**, and in the right Panel select "*Launch Instance*". In the form that pops-up fill the following Figure 6:

1. In *Details* a VM name composed with your IST ID → Instance Name: **vm-ist123456**
2. In *Source* → Select Image as boot source. Click on the **+** button next to **CirrOS**
3. In *Flavour* → Click on the **+** button next to **t1.nano**
4. In *Networks* → verify that your instance is connected to the default network.
5. Finalize by clicking the button "Launch Instance" in the right bottom of the form.



Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
t1.nano	1	512 MB	10 GB	10 GB	0 GB	Yes

Available 5

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
t1.micro	1	1 GB	10 GB	10 GB	0 GB	Yes
t1.small	2	2 GB	10 GB	10 GB	0 GB	Yes
t1.medium	2	⚠ 4 GB	20 GB	20 GB	0 GB	Yes
t1.large	4	⚠ 8 GB	40 GB	40 GB	0 GB	Yes
t1.xlarge	4	⚠ 16 GB	60 GB	60 GB	0 GB	Yes

Figure 6: VMCloud Launch Instance

You can now view the new topology you just created from the Network Topology tab, as you may observe in Figure 7.

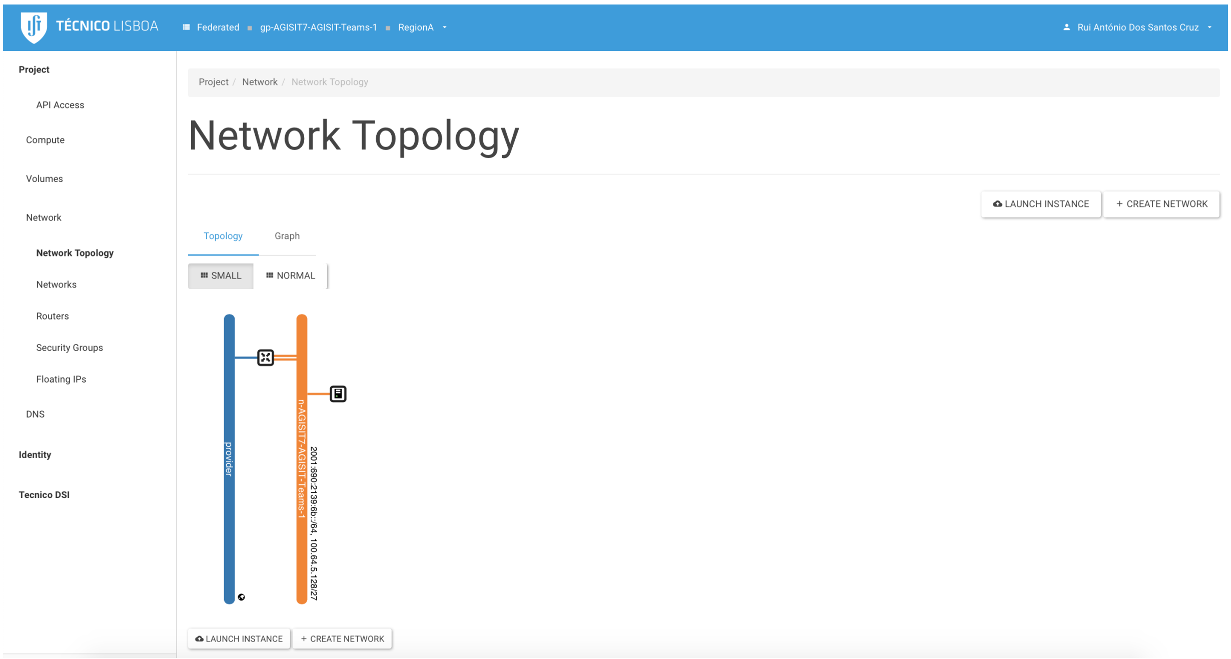


Figure 7: VMCloud Network Topology

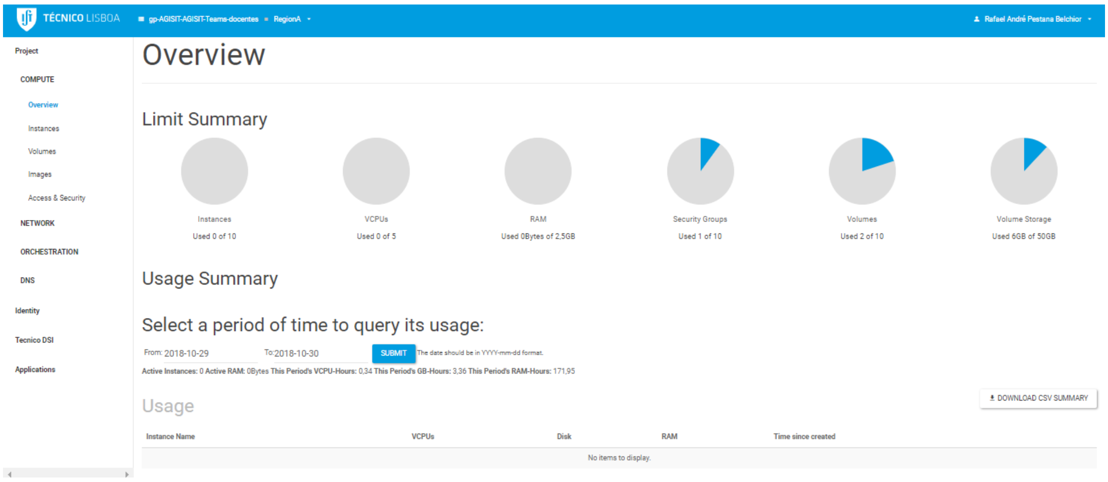
Experiment: Exploring the Horizon interface and the created infrastructure.

1. Select the VM you created in *Instances*. In the **Overview** section, collect the information about **Security Groups**.
2. Now, go to section **Log** and collect all the information in the **Instance Console Log**.
3. Now go to the **Console** section. Try to Login to the machine. Take a screenshot of the console after having logged in.
4. In the console of the VM, execute the commands `ifconfig` and `uname -a` and take a screenshot of the result. Leave the console with Exit.
5. Go back to the Dashboard and select Compute Images. Collect the names of All the Images available, including their sizes.

3.2 Finalize: Reset the Private Cloud Environment of the Tenant

Having finalized the experiment, delete all manual configurations you created in this Lab. Start with the compute instances.

Do not forget also to delete any other resources (for example Volumes) previously used by your instances, and not anymore needed. In the end you should verify that no resources (except the “default” Security Group) are used, as illustrated in Figure 8.



Overview

Limit Summary

Instances	VCPUs	RAM	Security Groups	Volumes	Volume Storage
Used 0 of 10	Used 0 of 5	Used 0Bytes of 2,5GB	Used 1 of 10	Used 2 of 10	Used 6GB of 50GB

Usage Summary

Select a period of time to query its usage:

From: 2018-10-29 To: 2018-10-30 The date should be in YYYY-mm-dd format.

Active Instances: 0 Active RAM: 0Bytes This Period's VCPU-Hours: 0,34 This Period's GB-Hours: 3,36 This Period's RAM-Hours: 171,95

Usage

Instance Name	VCPUs	Disk	RAM	Time since created
No items to display.				

Figure 8: VM Cloud Overview Dashboard