policloud

User Manual

Alessandro Frossi

April 14, 2015

policloud

Contents

policloud

1. PoliCloud Project

1.1 What is PoliCloud

PoliCloud (Cloud and Things Laboratory), or simply 'PoliCloud', is the private cloud designed, managed, implemented and deployed entirely by Politecnico di Milano. Coordinated by Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), PoliCloud aims at allowing researchers and students to benefit from the flexibility and elasticity of today's cloud-computing infrastructure. The services that PoliCloud aims to support range from storage for big data archival to pure computing power for scalable data analysis. PoliCloud would not be a reality without the donation from Yahoo, Inc., consisting of 227 rack-mountable, diskless servers, totalling 454 CPUs (1816 cores) and approximately 6 TB of memory.

1.2 Architecture

PoliCloud environment is geographically distributed over 3 different sites: two at Politecnico di Milano - Leonardo and one at Politecnico di Milano - Polo Territoriale di Como. At the time of writing, Leonardo Campus in Milan hosts the core of the PoliCloud deployment, composed of roughly 20 machines, which act as the main deployment and delivers services to the major part of the interested departments. This section is currently hosted in DEIB Department (*Milano - DEIB* from now on). An additional environment is under active development and will consist of 50 servers, will have more than double computational power, redundancy, high availability support and better performance overall; this new deployment is going to be hosted in Z1 server farm and will be referred to as *Milano - Z1* from this point on.

Finally, PoliMi branch in Como hosts the remote part of the PoliCloud environment, composed of 25 servers. This remote part is locally administered and is currently an independent region; it will be integrated in the overall architecture in the future, thus creating a single installation spanning over multiple geographical regions.

Each site runs a complete OpenStack deployment, currently with a single Controller node hosting all the core compute services: Nova (OpenStack controller), Glance (image manager), Cinder (volume manager), MySQL and RabbitMQ; Neutron agents (network services) are hosted

on a dedicate node thus avoiding performance bottlenecks. At this time no high-availability configuration is in place: service redundancy and segregation is under testing and will be deployed in the new environments, so that critical services won't be single points of failure. Finally, PoliCloud relies on a storage cluster to store and replicate images and virtual volumes; this cluster is built on commodity hardware using Ceph (`http://www.ceph.com`), which provides automatic data replication and high availability across multiple nodes. As a result of this design choice, no VM data (operating system or user data) is saved anywhere on computation nodes, but is always stored in the cluster; this provides a higher fault resilience, since virtual machines can be migrated without data loss event after a sudden host crash.

Table 1.1 shows the current configuration for the three PoliClou sites (Milano - DEIB, Como and Milano - Z1) and their resources in terms of VCPUs[1], RAM and available storage.

	Milano - DEIB	**Como**	**Milano - Z1**
Controller nodes	1	1	-
Network nodes	1	1	-
Storage nodes	4	4	-
Computation nodes	8	16	-
Total VCPUs	352	512	-
Total RAM	251 GB	329 GB	-
Total storage	18 TB	26 TB	-

Table 1.1: PoliCloud configuration for each site

1.3 Services

PoliCloud currently offers cloud virtual machines provisioning service. More services, like web application automatic hosting and deployment, will be added in the future.

1.4 How to request access

In order to gain access to PoliCloud resources, it's sufficient to send an e-mail to the administrator (alessandro.frossi@polimi.it) stating what kind of VMs are needed and what kind of applications are going to run in them. This would allow a requirement analysis to determine if the system is suitable for such needs and estimate the resource constraints of the project.

After the request is approved, the following information are required:

- project name
- project's main contact
- users to be registered / added

For each user, the requestor must provide first name and last name and a valid @polimi e-mail address. Once created, a confirmation email with the associated username and password will be sent.

[1] VCPUs are the virtual cores that can be allocated on a VM. Physical CPUs are usually shared among VMs, so this number is a multiple of the number of physical CPUs in the system.

policloud

2. Project management

2.1 Dashboard

Each environment is accessible through a web site, called *dashboard*, which allows a user to manage the project(s) he belongs to and the machines associated to them. The dashboard is primarily used as a mean to instantiate virtual machines and control their life cycle: create, suspend, resize and terminate them; it also includes a VNC web client to graphically connect to the VMs and interact with them without using a client.

As for now there's a distinct dashboard for each environment, and the corresponding URLs are reported in Table 2.1.

Environment	Address
Milano - DEIB	`http://131.175.56.226`
Como	`http://131.175.141.226`
Milano - Z1	TBA

Table 2.1: Dashboard URLs for PoliCloud environments

The credentials to access the dashboard are unique for every environment, so if a user was created to work in *DEIB* environment, the same user will not work in *Como* and vice versa. The same happens with passwords: if a user exists in two or more dashboards, the passwords are not linked, so that they may differ from one environment to the other. It's the administrator's duty to tell the user which is the correct dashboard to use, as part of the procedure explained in Section 1.4.

Figure 2.1 shows how the dashboard appears after a successful login.

2.2 Instances

The *Instances* tab shown in Figure 2.2 contains the list of all the VMs in the current project, along with some useful information like the IP addresses (both internal and floating), the flavor they are built on, the associated keys and their status.

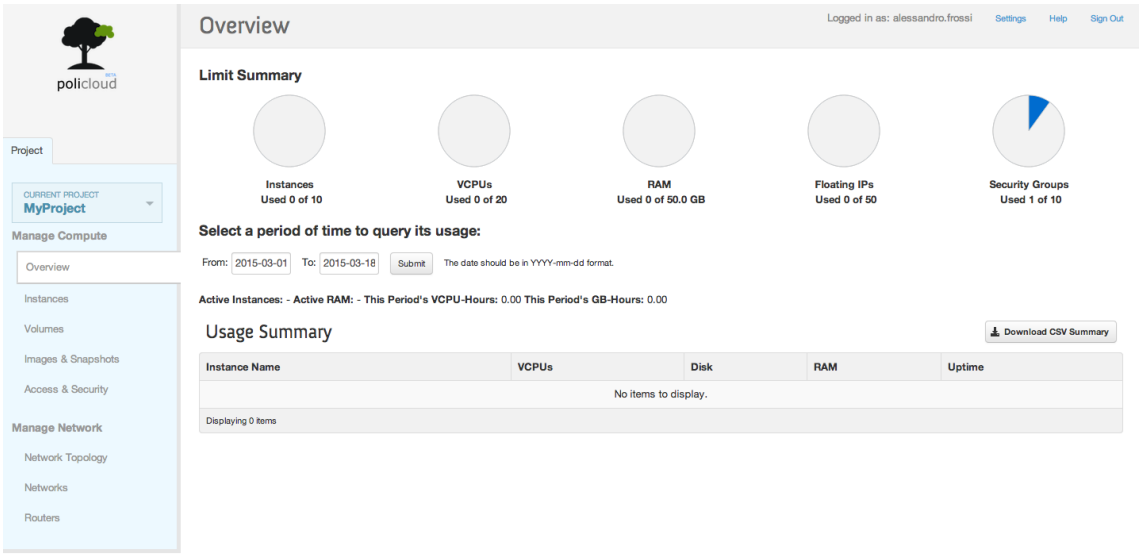

Figure 2.1: PoliCloud Dashboard

Every virtual machine has at least one IP address but has usually two or more. One of them *must* be an internal one, often associated to the *fixed network*: this is a virtual network where all the virtual machines reside and can communicate; there's no project boundary in this network, which means that every machine is able to see every other VM in the system. There's also no physical boundary, so that machines on different host servers can exchange packets. The *fixed* network has its own DHCP server so there's no way for the user to force a static IP while leaving a machine attached to that network. Table 2.2 shows the networks reserved for *fixed* and *floating* IPs. However, it is not mandatory to associate a machine to the *fixed* network: it is possible to create a tenant (project) specific virtual network that won't be accessed by virtual machines outside the project. This is usually done to segregate sensitive hosts.

Environment	Fixed IP network	Floating IP network
Milano - DEIB	10.75.130.0/24	10.75.131.224/27
Como	10.75.130.0/24	10.75.131.224/27
Milano - Z1	TBD	TBD

Table 2.2: Networks for floating and fixed IP addresses.

2.3 Volumes

Volumes are fundamentally the disks that are attached to the virtual machines. There's not limit to their dimension, provided that the project quotas are not exceeded; there's, however, a maximum number of volumes that can be created, defined by quotas as well (see Section 2.6 for details). Every volume can be attached to one machine at a time, but it can unmounted (provided it is not the root disk) and hot-attached to another running machine; it is not possible to attach a volume to a shutoff machine.

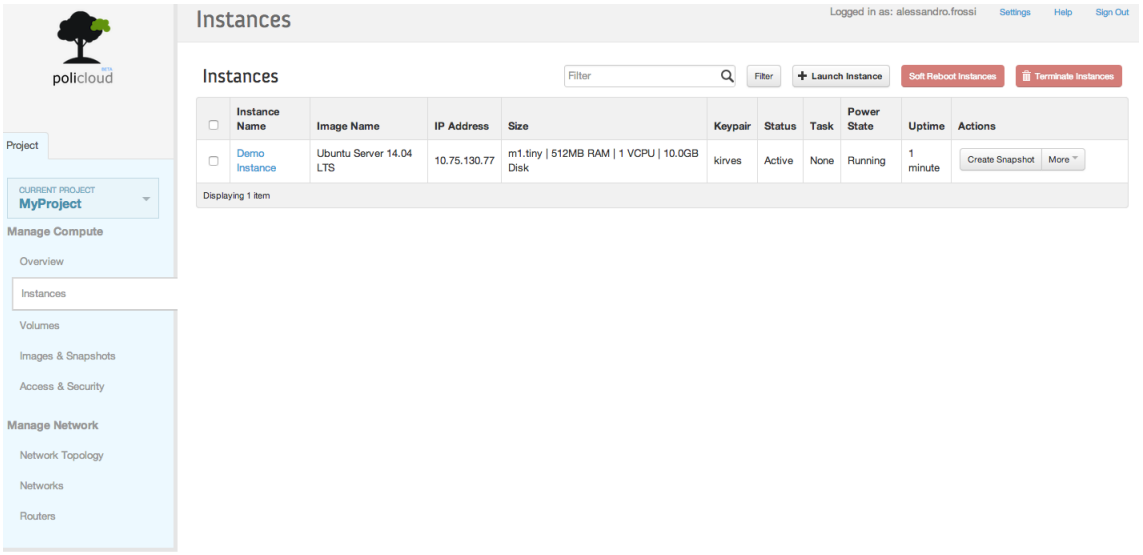

Figure 2.2: Instances tab

This section also contains *snapshots*: an exact image of a volume in a given point in time. Snapshot creation has to be triggered manually when needed; once created, new VMs or even new volumes can be created from that starting point, giving the user a way to create templates for machines.

R Volumes and snapshots are always persistent and automatically replicated by the storage cluster.

2.4 Access & keys

This dashboard section contains all the information regarding external access to the project and the virtual machines it contains. These information can be either tied to the user (SSH keys), to the project (Floating IP, Security Groups) or to the system (API endpoints).

2.4.1 Keypairs

Keypairs are SSH keys that get associated to virtual machines at creation time in order to restrict access to the rightful owner of the VM: no one without the corresponding private key will be able to log into the machine unless some configuration changes are made. A keypair can be created in two ways: imported from a preexisting public key (useful for not having to deal with multiple keys if you already have one) or created from scratch. In the latter case the user downloads a *.pem* file containing the private key.

R *.pem* files are the only thing needed to gain root access to the machines the keypair is associated to; it is important not to give them away, unless to trusted coworkers.

A keypair is associated to a user, not a project. To grant multiple users access to a VM it is possible to:

- create a keypair whose private key is shared among the users and associate VMs with it

- have one user create the VM with his own keypair and manually add other users' public key to *.ssh/authorized_keys* file

2.4.2 Floating IPs

Floating IPs are IP addresses directly connected to the external network that are used to allow external access to a machine. They are not directly connected to an interface within a VM (this means it is not possible to see the floating IP from a ssh session) but rather packets are routed from the cloud infrastructure to the virtual network. Unlike fixed IP addresses, floating IPs are not necessary for a VM to work and can be associated and disassociated while the VM is running; it is quite common and advisable to associate floating IPs only for the time needed to setup the machine or to login into it.

This dashboard section lists all the floating IPS claimed by the current project and the machines, if any, they are associated with.

R Floating IPs are a limited and valuable resource in a cloud environment, so users are asked to release them as soon as they are not needed in order to put them back in the availability pool.

Floating IP addresses are taken from a particular network (see Table 2.2) that is statically translated to a public network. As a consequence, the reported floating IP cannot be used as it is to interact with the associated VM; it has to be translated before-hand. Translation rules are reported in Table 2.3: last octet from IP address is maintained in translation. E.g. if the reported floating IP is 10.75.131.230, the corresponding public IP will be 131.175.56.230 for Milano - DEIB environment.

Environment	Floating IP	Public IP
Milano - DEIB	10.75.131.X	131.175.56.X
Como	10.75.131.X	131.175.141.X
Milano - Z1	TBA	TBA

Table 2.3: Floating IP NAT rule.

2.4.3 Security Groups

Security groups are the set of rules for the virtual firewall of an OpenStack project. By default there's no restriction on *Egress* traffic, while *Ingress* traffic is blocked. It is possible, however, to add custom / predefined rules to a security group to loosen the restrictions; as a consequence, all packets satisfying the said rule are forwarded to the VM. This does not, however, mean that the user is free to open any port that he needs: he can intervene only on the software firewall but cannot bypass checks on department firewalls, managed by ASICT. Tables 2.4, 2.5 show the set of already opened ports that can unblocked via security groups. Additional ports have to be requested to PoliCloud admin alongside with the reason why they are needed.

R Security groups affect only traffic coming from outside the cloud environment. Traffic between VMs is not filtered by security groups or department firewalls.

In addition, *Milano - DEIB* has restrictions on *Egress* traffic originating from the virtual machines, as summarized in Table 2.6.

Protocol	Port	Service
TCP	22	SSH
TCP	80	HTTP
TCP	443	HTTPS
TCP	2222	Git over SSH
TCP	3000 $\sim$ 3500	Custom services
TCP	3389	RDP
TCP	6080	
TCP	8000	Custom
TCP	8080	Custom

Table 2.4: Non blocked ports for ingress traffic in Milano - DEIB environment.

Protocol	Port	Service
TCP	22	SSH
TCP	80	HTTP
TCP	443	HTTPS
TCP	3306	MySQL
TCP	3389	RDP
TCP	8000	Custom
TCP	8080	Custom
TCP	8888	Custom

Table 2.5: Non blocked ports for ingress traffic in Como environment.

2.4.4 API Access

OpenStack allows registered users to use cloud services via native client instead of relying only on the dashboard. To do so, it exposes APIs for every service it hosts: computing, storage, authentication and network; it also provides EC2 compatible APIs to use third party clients. This dashboard section reports the API endpoints (see Figure 2.3). Note that the IP is not public, but has to be translated using the rules explained in Table 2.3.

R APIs are exposed on ports currently blocked by department firewalls, so this feature is currently not usable. It will be enabled for use in the future.

2.5 Network

Network tab shows the virtual network layout of the current project. *Fixed* and *floating* networks are present by default and managed by the system, so they cannot be modified or removed: they are, respectively, the main virtual network the VMs associate to if no particular network requisite has to be satisfied and the network containing the public IPs. Users can, however, create custom virtual networks to segregate their machines and prevent them to exchange packets with other machines in the system.

Custom networks are defined in terms of address pool, netmask, gateway and DHCP server, just like a normal LAN; it is also important to note that VMs on custom networks will be able to access the Internet if a proper router is created to connect the networks.

Protocol	Port	Service
TCP	21	FTP
TCP	22	SSH
TCP	80	HTTP
TCP	110	POP3
TCP	143	IMAP4
TCP	443	HTTPS
TCP	587	SMTP
TCP	993	IMAPS
TCP	995	POP3S
TCP	1723	PPTP
TCP	3389	RDP
TCP	8000	Custom
TCP	8080	Custom
UDP	123	NTP
UDP	1194	OpenVPN
UDP	4500	IPSec
UDP	500	ISAKMP

Table 2.6: Non blocked ports for egress traffic in Milano - DEIB environment.

2.6 Quotas

OpenStack uses *quotas* to enforce resource usage limitations on a per project basis, to avoid excessive resource claims by single users. Table 2.7 shows the default quotas.

Element	Default value
VCPUs	20
RAM (GB)	51.2
Storage (GB)	1000
Volumes	10
Snapshots	10
Key Pairs	100
Instances	10

Table 2.7: Default project quotas.

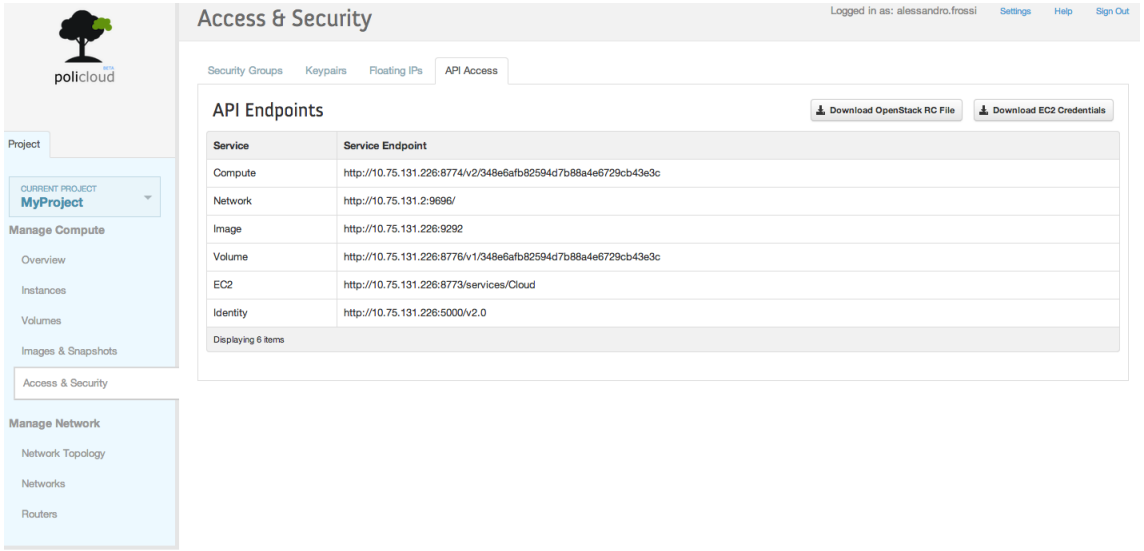

Figure 2.3: API Access Tab

policloud

3. Provisioning

3.1 VM Creation

The instantiation of a virtual machine on PoliCloud infrastructure is merely the decision of 4 key factors of a machine:

- the *flavor*, i.e. its *size* (Section 3.2)
- the content of the root disk (Section 3.3)
- the way to access the machine (Section 3.4)
- the associated network (Section 3.5)

3.2 Flavor

PoliCloud uses the standard OpenStack notation (which is very similar to Amazon's) to define VM sizes, or *flavors*. Flavors describe a VM hardware architecture in terms of number of VCPU, RAM and dimension of the root disk (in case of ephemeral storage, as described in Section 3.3); there are 5 basic flavors responding to different requirements and billed differently, as reported in Table 3.1.

Flavor	VCPUs	RAM	Root disk (GB)	Price
m1.tiny	1	512 MB	10	TBD
m1.small	1	2 GB	20	TBD
m1.medium	2	4 GB	20	TBD
m1.large	4	8 GB	20	TBD
m1.xlarge	8	16 GB	20	TBD

Table 3.1: Base PoliCloud flavors.

In addition, custom flavors can be requested and created purposely for specific projects to better tailor requirements; their price would be subsequently determined by the resources it claims.

R Some flavors are disabled if no nodes have enough free resources to accommodate the request. On some conditions, however, flavors do not get disabled and can be instantiated; this may result in a generic creation error.

3.3 Disk

Once the hardware configuration of a VM has been chosen, the user can decide what will be the content of the root disk of the machine, that is the operating system (Unix, Windows), distribution and version the machine will be based on. Available images include Windows Server R2, Windows 7 64bit, RedHat Enterprise Linux 7.0, CentOS 6.5, Fedora 18, Fedora 19, Ubuntu Server 12.04, Ubuntu Server 14.04. Several boot options are available, but they can all be summarized in two choices: booting from an image or a volume. Booting a VM from an image (or a snapshot, which is treated just like a custom image) results in a root disk cloned from the image itself and resized to the flavor default size (the current PoliCloud version has problems resizing a root disk, see Section 4.1.1). The machine is booted from this newly created volume but it is *not persistent*: the volume and all its content is destroyed as soon as the VM is terminated; this is called *ephemeral storage*.

R An *ephemeral volume* is not destroyed by shutting down a machine, but only terminating it. The only way to convert to a persistent storage is to create a snapshot of the running machine and create a new volume from that snapshot. The VM will have then to be recreated using that volume as root disk.

It is advisable, instead, to always boot a machine from a purposely created volume, to obtain a *persistent storage*. Persistent volumes do not get destroyed even after terminating the virtual machine, but rather their data is maintained and remains available for newly instantiated machines. Volume creation is shown in Figure 3.1: a root disk has necessarily to be created based on an image to allow the VM to boot. Additional disks, instead, are usually left blank.

3.4 Keypair

Every VM has to be associated with an existing Keypair to allow SSH access. Figure 3.2 shows the configuration page that allows to set a keypair for the machine. Refer to Section 2.4 for key creation and limitations.

3.5 Network

The last fundamental component of a virtual machine is the network(s) it must be attached to. Using the menu shown in Figure 3.3 it is possible to select the desired LANs: for every one of them, a virtual interface in the machine is created and connected as if different cables were connected to different ports of a switch. Note that the *floating* network is always missing from the list because it behaves differently from a standard network: it is not possible to attach to that, but rather single IPs have to be attached to virtual machines. Also, no virtual interface is associated to floating networks: *NAT-ing* is performed by the PoliCloud architecture.

3.6 External Access

External access to virtual machines is only possible after associating a *Floating IP* to an interface (or *port*) of the same VM. Figure 3.4 shows the floating IP association menu, where the user can collect a public IP from the pool and statically connect it to a virtual machine.

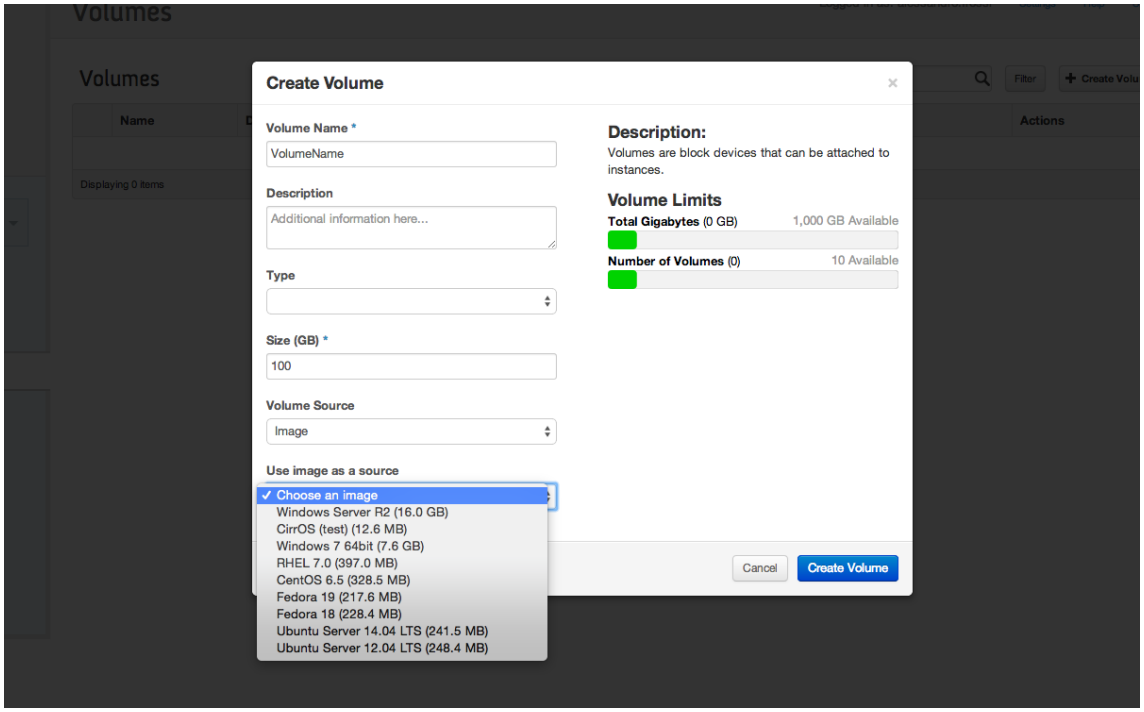

Figure 3.1: Volume creation

Floating IP, however, is just the first requirement to gain access from the Internet. See Section 2.4.2 and Section 2.4.3 for additional information.

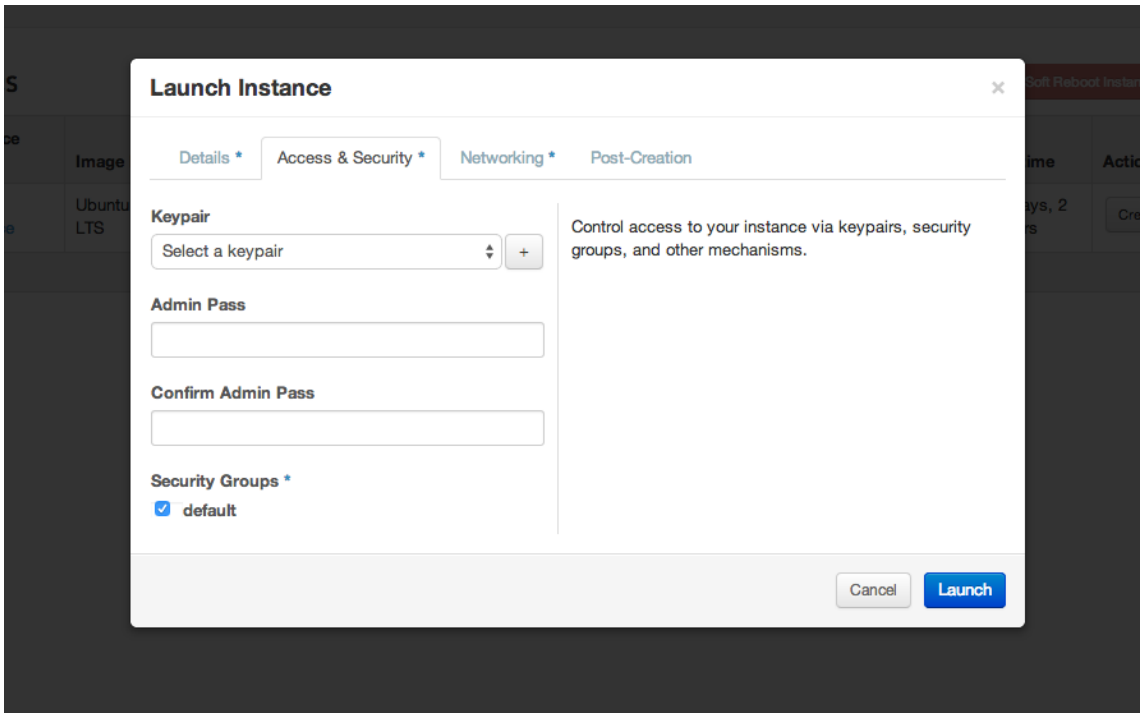

Figure 3.2: Keypair configuration for VM

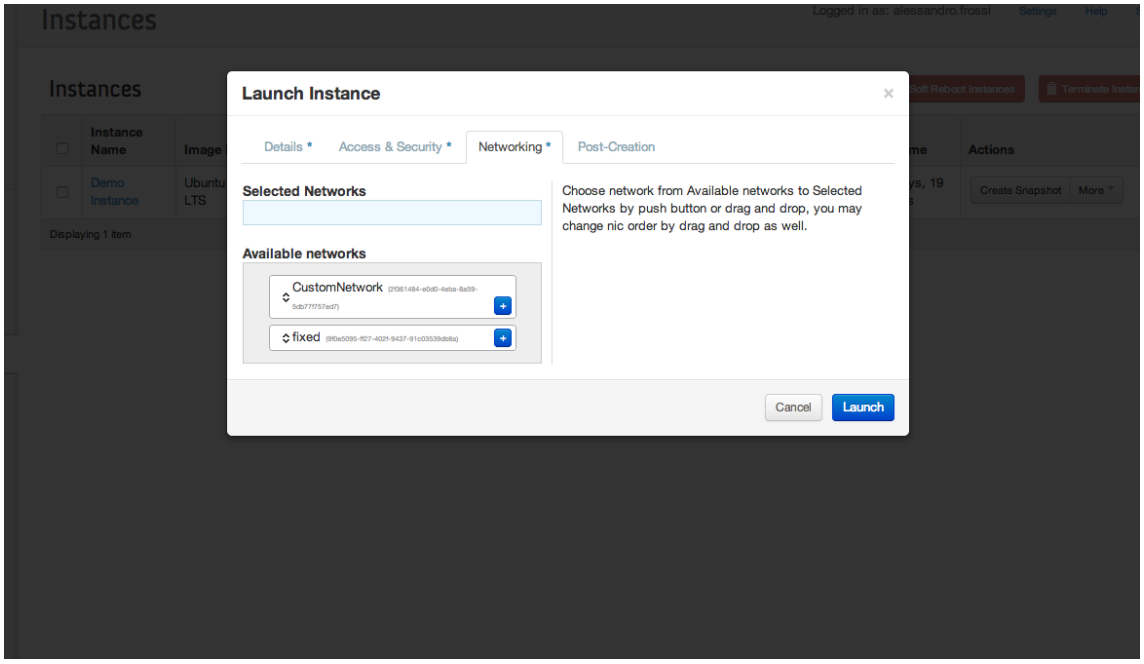

Figure 3.3: Network configuration for a newly created VM.

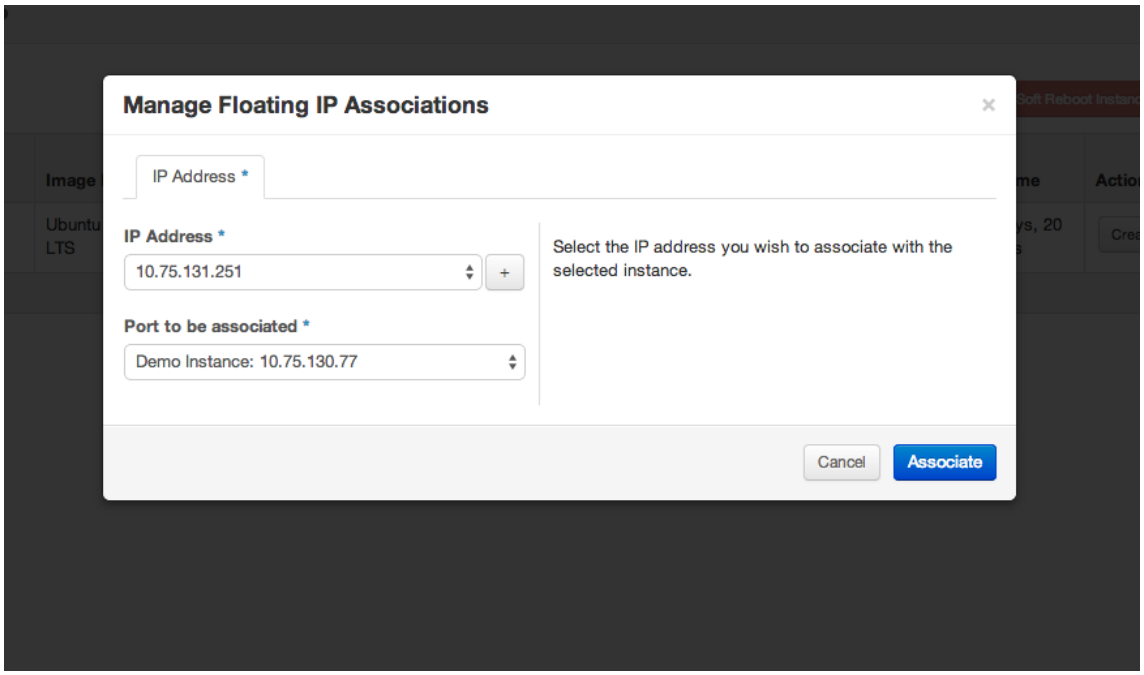

Figure 3.4: Floating IP association menu. If no IP has been reserved to the project yet, it is possible to allocate one by click on the ‘plus’ icon.

policloud

4. Known Issues

This chapter reports all the known issues present in PoliCloud at the time of writing this user manual. For some of them a solution is reported, while for others only mitigation is possible. These sections will be updated when issues get resolved or some new problems are found and investigated.

4.1 Storage

4.1.1 Ephemeral Disk Size

Systems: all
Added on: installation

When booting a VM from an image, the system is not able to resize the root disk using the current storage system. This results in a root disk of the exact size of the image and almost full.

Mitigation: always boot a VM from volume so that its root disk can be of the desired size.

4.1.2 Performance

Systems: all
Added on: installation

Storage cluster is still in development and must undergo several phases of fine tuning and dimension improvement. For now, therefore, might show poor performances under heavy duty or when many write requests are issued.

Mitigation: there's no userside mitigation; the storage cluster will be improved in the next releases.

4.1.3 Quota Exceeded

Systems: all
Added on: installation

When trying to exceed storage quotas, the dashboard reports a generic error rather than providing insightful information.

Mitigation: if an error is triggered when creating a snapshot or a volume try deleting unused volumes before trying again. Or consider asking for a quota upgrade.

4.2 Network

4.2.1 Traffic replication

Systems: all
Added on: installation

Traffic fragmentation and packet multiplication occurs under heavy network traffic or big files transfers. This is due to the fact that the network infrastructure has a little management overhead for every packet and, since server NICs do not have Jumbo frames support, often the 1500 byte *Maximum Transfer Unit* (MTU) is exceeded and packet is fragmented. In addition, current switches do not support *fragmentation offload*, which results in network performance degradation.

Solution: if the machine is expected to be generating or receiving high amounts of data, run the following commands to avoid crashes or network congestions:

```
sudo ethtool -K eth0 tso off gro off gso off
sudo sysctl -w net.ipv4.tcp_sack=0
sudo ifconfig eth0 mtu 1400
```

4.2.2 Floating IP dashboard delay

Systems: all
Added on: installation

After associating a floating IP to a virtual machine, the dashboard takes a couple of minutes before showing it in the instance details; the IP is however already active and can be used nevertheless.

Solution: take note of the IP when associating it (from the confirmation message in the upper right corner of the dashboard) or wait for the dashboard to show it in the details