OpenStack Tutorial

Matteo Panella - matteo.panella@lngs.infn.it

Laboratori Nazionali del Gran Sasso - INFN

Cloud Computing Tutorial
Summary

1. Introduction to OpenStack

2. Gran Sasso Clouds

3. Sample session
Summary

1. Introduction to OpenStack

2. Gran Sasso Clouds

3. Sample session
OpenStack is an enterprise-grade open source IaaS platform. It provides:

- computing resources (vCPU, RAM, system images...)  
- network resources (L2 networks, virtual routers...)  
- storage resources (persistent virtual disk devices, VM snapshots)
OpenStack development is overseen by the OpenStack Foundation. The Foundation is backed by 5600 individual members and 850 organizations.

Different companies contribute to OpenStack development:

- AT&T
- Canonical
- HP
- IBM
- Rackspace
- Red Hat, Inc.
- SUSE

...and many more!
OpenStack in numbers

Some numbers about OpenStack:

- 3 years of development
- 1278 contributors
- 1289000 lines of code
- more than $10 million in funding

OpenStack is quickly becoming the *de facto* standard for private IaaS clouds.

For more information visit [http://www.openstack.org/](http://www.openstack.org/)
OpenStack is composed by 5 different components:

1. identity and authentication service
2. computing service
3. networking service
4. storage service
5. dashboard (web frontend)
OpenStack components

Your Applications

OpenStack Shared Services

Standard Hardware

Compute

Networking

Storage

OpenStack Dashboard

APIs

M. Panella (LNGS)
Identity and authentication service

What it does

Authenticates users and verifies project membership.

Authentication and authorization based on:

- username/password pair
- projects (also known as “tenants”)
- roles
Computing service

What it does

Creates, runs and manages instances (virtual machines).

Resources managed by this service:

- instances
- vCPU
- RAM
- instance metadata (hostname, SSH keypairs, boot scripts... )
Networking service

What it does

Provides network resources to the computing service.

Resources managed by this service:

- L2 networks
- subnets
- virtual routers
- firewalling rules
- floating IPs
What it does

Provides persistent storage to the computing service.

Resources managed by this service:

- volumes (virtual disk devices)
- snapshots
- OS images
Resources recap

Every user has access to the following resources:

- instances
- vCPU
- RAM
- networks
- virtual routers/firewalls
- virtual disks
Resources recap

Every user has access to the following resources:

- instances
- vCPU
- RAM
- networks
- virtual routers/firewalls
- virtual disks

Nota Bene

Resources belong to projects, only SSH keypairs belong to users!
An instance is a self-contained virtual machine:

- resource allocation based on “flavors”
- boots from a **pre-installed system image** or from a **snapshot**
- has a fixed **private** IP address
- can have one or more floating IP addresses
- can have an ephemeral disk
- can have one or more volumes attached
Instances boot off a standard system image and can be customized upon boot with a **user script**. Most system images come with SSH enabled out-of-the-box and keyed to a user-specific SSH keypair. **Password-based login is disabled for the default account.**
Instances boot off a standard system image and can be customized upon boot with a user script. Most system images come with SSH enabled out-of-the-box and keyed to a user-specific SSH keypair. **Password-based login is disabled for the default account.**

**Warning**

Instances exist as long as they are not terminated. Upon termination the system disk will be irreversibly destroyed. To retain system configuration make a snapshot or simply suspend the instance.
OpenStack provides isolated L2 networks to each project. Each network can have one or more subnets and virtual routers. Private IP addresses are automatically allocated to instances, while floating IP addresses must be manually allocated to projects and assigned to instances.
The networking service also provides *virtual firewalls* for all instances with multiple independent rule sets. Each instance can have more than one rule set attached to it. The rules also apply to traffic *within* the same network.
Volumes are persistent virtual disk devices. Each instance can have one or more volumes attached to it and each volume may be attached to one instance at a time. They are created and destroyed independently from instances, so they are suited for long-term storage of data. When created, volumes are like empty disks. They must be partitioned and formatted in order to use them.
Volumes

Volumes are persistent virtual disk devices. Each instance can have one or more volumes attached to it and each volume may be attached to one instance at a time. They are created and destroyed independently from instances, so they are suited for long-term storage of data.

When created, volumes are like empty disks. They must be partitioned and formatted in order to use them.

Warning

Due to their nature, volumes are not backed up automatically. Users have to arrange backup of volume contents on their own!
1 Introduction to OpenStack

2 Gran Sasso Clouds

3 Sample session
Gran Sasso Clouds is the OpenStack environment of LNGS. It is currently a standalone system. In the future it will be integrated with other computing services like U-LITE. It is available upon request to experimental collaborations, working groups, LNGS services and individual users.
Currently, Gran Sasso Clouds has the following resources available:

- 48 CPU cores
- 80 GB of RAM
- several TB for volume and snapshot storage

Capacity will be expanded as needed as more users start working with it.
Gran Sasso Clouds is extremely flexible and can be used for many scenarios:

- internal and public web sites
- wikis
- blogs
- database services
- software development
- data analysis
- Monte Carlo
- prototyping
- ...
Accessing Gran Sasso Clouds

The web frontend

Gran Sasso Clouds can be accessed from https://stackctl.lngs.infn.it. All operations can be performed via the web frontend.

Who can access it

Everybody with a standard LNGS account can request access to the Computing and Network Service. CNS staff will enable access for your account and assign it to one or more projects (experiment and/or LNGS service).
Direct access to Gran Sasso Clouds is required only to manage resources (instances, networks, volumes...). Depending on the kind of use case, end users do not require any account at all or just a regular UNIX account on the instances themselves.
Direct access to Gran Sasso Clouds is required only to manage resources (instances, networks, volumes...). Depending on the kind of use case, end users do not require any account at all or just a regular UNIX account on the instances themselves.

What about groups?

Groups need to designate a few selected individuals that will manage resources.
Gran Sasso Clouds supports various network configurations. The standard network configuration *just works* for most users and groups. Experimental collaborations and working groups that require a more complex network configuration can contact the CNS for more information.
1. Introduction to OpenStack

2. Gran Sasso Clouds

3. Sample session
Security group rules

Edit Security Group Rules

<table>
<thead>
<tr>
<th>IP Protocol</th>
<th>From Port</th>
<th>To Port</th>
<th>Source</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>-1</td>
<td>-1</td>
<td>0.0.0.0.0 (CIDR)</td>
<td></td>
</tr>
<tr>
<td>Any</td>
<td>-</td>
<td>-</td>
<td>default</td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>22</td>
<td>22</td>
<td>0.0.0.0.0 (CIDR)</td>
<td></td>
</tr>
</tbody>
</table>
Keypair management

Access & Security

Keypairs

- **Keypair Name**: matteo

Actions:
- **Create Keypair**
- **Import Keypair**
- **Delete Keypair**

Logged in as: mpanella

<table>
<thead>
<tr>
<th>Security Groups</th>
<th>Keypairs</th>
<th>Floating IPs</th>
<th>API Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Displaying 1 item
### Running instances

#### Project

**CURRENT PROJECT**

- **calcolo**

**Manage Compute**

- Overview
- Instances

**Volumes**

- Images & Snapshots
- Access & Security

**Manage Network**

- Networks
- Routers
- Network Topology

---

#### Instances

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>IP Address</th>
<th>Size</th>
<th>Keypair</th>
<th>Status</th>
<th>Task</th>
<th>Power State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tarsnap</td>
<td>192.168.42.6</td>
<td>m1.small</td>
<td>2GB RAM</td>
<td>1 vCPU</td>
<td>2GB Disk</td>
<td>matteo</td>
<td>Active</td>
</tr>
<tr>
<td>luna03node003</td>
<td>192.168.42.4</td>
<td>m1.small</td>
<td>2GB RAM</td>
<td>1 vCPU</td>
<td>20GB Disk</td>
<td>steslano</td>
<td>Active</td>
</tr>
<tr>
<td>pinger</td>
<td>192.168.42.2</td>
<td>m1.small</td>
<td>2GB RAM</td>
<td>1 vCPU</td>
<td>2GB Disk</td>
<td>matteo</td>
<td>Active</td>
</tr>
</tbody>
</table>

Displaying 3 items
### Running instances

![Instance Management Interface](https://example.com/instance.png)

#### Table of Instances

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>IP Address</th>
<th>Size</th>
<th>Keypair</th>
<th>Status</th>
<th>Task</th>
<th>Power State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tarsnap</td>
<td>192.168.42.6</td>
<td>m1.small 2GB RAM</td>
<td>matteo</td>
<td>Active</td>
<td>None</td>
<td>Running</td>
<td>Create Snapshot More...</td>
</tr>
<tr>
<td>lunarode003</td>
<td>192.168.42.4</td>
<td>m1.small 2GB RAM</td>
<td>stestalo</td>
<td>Active</td>
<td>None</td>
<td>Running</td>
<td>Create Snapshot More...</td>
</tr>
<tr>
<td>pinger</td>
<td>192.168.42.2</td>
<td>m1.small 2GB RAM</td>
<td>matteo</td>
<td>Active</td>
<td>None</td>
<td>Running</td>
<td>Create Snapshot More...</td>
</tr>
</tbody>
</table>
Launching an instance
Launch Instance

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

Keypair
- matteo

Security Groups
- default
- gluster
- web

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

[Launch] [Cancel]
Instance networking

Launch Instance

Selected Networks

Available networks

calcolo (96e7a38f-de71-4ab5-a369-80a9e744012)

Choose network from Available networks to Selected Networks by push button or drag and drop, you may change nic order by drag and drop as well.
Post-boot customization

Launch Instance

Details  Access & Security  Networking  Volume Options  Post-Creation

Customization Script

You can customize your instance after it's launched using the options available here.

The "Customization Script" field is analogous to "User Data" in other systems.
Boot log

M. Panella (LNGS)
OpenStack Tutorial
Sep 26, 2013
Volumes

Volumes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
</table>

No items to display.

Displaying 0 items
## Volumes

### OpenStack Tutorial

- **Date:** Sep 26, 2013

#### Volumes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No items to display.

**Creating a Volume:**

- Click the `+ Create Volume` button to create a new volume.

**Current Project:**

- **calcio**

**Manage Compute**

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

**Manage Network**

- Networks
- Routers
- Network Topology
Creating a volume

**Description:**
Volumes are block devices that can be attached to instances.

**Volume Quotas**
- Total Gigabytes (0 GB): 1,000 GB Available
- Number of Volumes (0): 10 Available

**Create Volume**
- Volume Name: workshop
- Description: Volume for CC workshop
- Type:
- Size (GB): 10

[Create Volume]
Volumes (cont’d)

A screenshot of a volume management interface. The screen shows a list of volumes with columns for Name, Description, Size, Status, Type, Attached To, and Actions. One volume named 'workshop' is visible, described as 'Volume for CC workshop' with a size of 10GB and in an 'Available' status. The interface is part of the OpenStack project, as indicated by the logo and URL in the browser header.
Volumes (cont’d)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>workshop</td>
<td>Volume for CC workshop</td>
<td>10GB</td>
<td>Available</td>
<td>-</td>
<td></td>
<td>Edit Attachments</td>
</tr>
</tbody>
</table>

![Screenshot of OpenStack Volumes interface](image)

The screenshot shows the Volumes interface in the OpenStack web console, with a focus on managing and exploring volumes. The interface includes options for creating new volumes, viewing volume details, and managing attachments. The highlighted volume, labeled `workshop`, is associated with a description of 'Volume for CC workshop' and has a status of 'Available'.
Attaching a volume

Manage Volume Attachments

Attachments

<table>
<thead>
<tr>
<th>Instance</th>
<th>Device</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No items to display.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displaying 0 items</td>
</tr>
</tbody>
</table>

Attach To Instance

Attach to Instance

Device Name

workshop (14a6eab9-7dcc-4b8b-bb8d-30d6180f) /dev/vdc

Attach Volume
Cancel
Never use /dev/vda or /dev/vdb as device names!
Attaching a volume

- OpenStack Tutorial
  - Sep 26, 2013
### Floating IPs

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Instance</th>
<th>Floating IP Pool</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.3.122</td>
<td>-</td>
<td>Ings</td>
<td>Associate Floating IP</td>
</tr>
<tr>
<td>172.16.3.127</td>
<td>lunanode003</td>
<td>Ings</td>
<td>Disassociate Floating IP</td>
</tr>
<tr>
<td>172.16.3.131</td>
<td>-</td>
<td>Ings</td>
<td>Associate Floating IP</td>
</tr>
<tr>
<td>172.16.3.133</td>
<td>-</td>
<td>Ings</td>
<td>Associate Floating IP</td>
</tr>
</tbody>
</table>

Displaying 4 items

This image shows a screenshot of an OpenStack dashboard with the Floating IPs tab open. The table lists four floating IP addresses, each associated with the Ings floating IP pool. The screenshot includes a button to allocate an IP to a project and another to release floating IPs.
## Floating IPs

![Image of OpenStack dashboard showing Floating IPs](https://stackctl.lngs.infn.it/project/access_and_security/)

### Access & Security

#### Floating IPs

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Instance</th>
<th>Floating IP Pool</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.3.122</td>
<td>-</td>
<td>Ings</td>
<td>Associate Floating IP</td>
</tr>
<tr>
<td>172.16.3.127</td>
<td>lunanode003</td>
<td>Ings</td>
<td>Disassociate Floating IP</td>
</tr>
<tr>
<td>172.16.3.131</td>
<td>-</td>
<td>Ings</td>
<td>Associate Floating IP</td>
</tr>
<tr>
<td>172.16.3.133</td>
<td>-</td>
<td>Ings</td>
<td></td>
</tr>
</tbody>
</table>

Displaying 4 items
Associating a floating IP

IP Address

IP Address

172.16.3.133

Select the IP address you wish to associate with the selected instance.

Port to be associated

workshop: 192.168.42.5

Cancel

Associate
Inside the VM

The authenticity of host '172.16.3.133 (172.16.3.133)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.3.133' (ECDSA) to the list of known hosts.
Linux wheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

debian@workshop:$ sudo -i
debian@workshop:# mkfs.ext4
mkfs.ext4
mkfs.ext4dev
root@workshop:# mkfs.ext4 /dev/vdc
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block:
Maximum filesystem blocks=268435456
80 block groups
32768 blocks per group, 32768 fragments per group
64 nodes per group
Superblock backups stored on blocks:
32768, 98384, 163840, 229376, 294912, 819200, 884736, 1005632
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@workshop:#
Questions?