

OpenStack Tutorial



Jacopo De Benedetto

SDN Course 2017

Computer Networks Group

University of Goettingen

Agenda



- OpenStack and TryStack intro
- Dashboard Overview
- Configure Network
- Create/Load an Image
- Launch an Instance
- Login into an Instance
- Manage Security Rules
- Create a Volume
- Attach a Volume to an Instance
- Using REST API with OpenStack CLI
- Real use-case scenarios:
 - Deploy a web server
 - Install WordPress

Intro



OpenStack is a free and open-source software to build cloud computing infrastructures

- It hides the complexity of physical resource management

So it means that you don't need to take care of anything?

...not so true

Let's see what we need to have a working virtual environment and validate it with some real-world use-case examples!

Where to start



Fortunately we don't need to create an OpenStack installation on our own (it can take hours.. if you are lucky)

For this Tutorial we will use **TryStack**, a free service to test a real OpenStack deployment:

- trystack.org

To have an account you have to join their Facebook team

- only for this Tutorial we have created some test-accounts that you can use

Dashboard Overview



- Project
 - A project can be seen as a container of some cloud resources
 - As a user you can be part of one or more project
- Compute
 - Let you manage the computational resources of your project, so practically your virtual machines
- Network
 - Here you can configure are your virtual machines are connected together and to the outside world
- Object Store
 - We'll not use it...
- Identity
 - Section to manage identities for the project (we are not allow to use it ☹)

Network



- To launch an instance we need at least one network to attach it on
- The first time you login, you'll find only a «public» network but you're not allowed to use it, so...
 - ..let's create a network!

Create a network



Project > Network > Networks > Create Network

Create Network ✕

Network Subnet Subnet Details

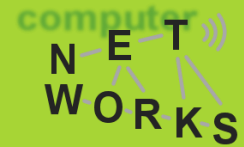
Network Name

Admin State ⓘ

Create Subnet

Create a new network. In addition, a subnet associated with the network can be created in the next panel.

Create a network



Project > Network > Networks > Create Network

Create Network

Network > Subnet > Subnet Details

Subnet Name

Network Address ⓘ

IP Version

Gateway IP ⓘ

Disable Gateway

Cancel « Back Next »

Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Details" tab.

Create a network



Project > Network > Networks

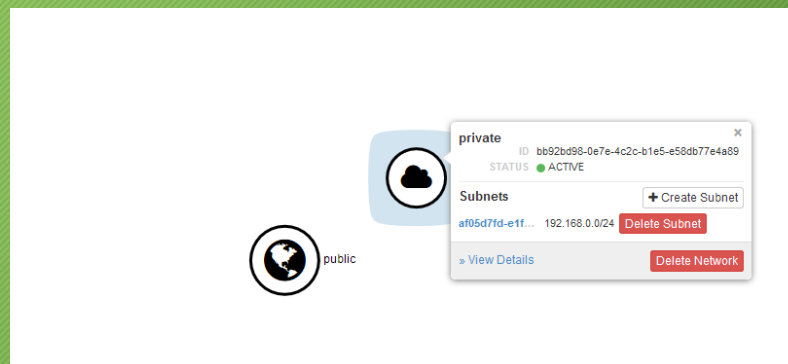
Networks

Filter

<input type="checkbox"/>	Name	Subnets Associated	Shared	Status	Admin State	Actions
<input type="checkbox"/>	private	private-subnet 192.168.0.0/24	No	Active	UP	<input type="button" value="Edit Network"/> <input type="button" value="⌵"/>

Displaying 1 item

Project > Network > Network Topology



Create/Load an Image



To launch an instance we need also an image to start it from

- OpenStack allows you to load different type of images
 - For the supported format please consult the documentation
- TryStack has some preloaded images

Project > Compute > Images

<input type="checkbox"/>	Image Name	Type	Status	Public	Protected	Format	Size	Actions
<input type="checkbox"/>	Fedora25 Atomic	Image	Active	Yes	No	QCOW2	533.1 MB	Launch Instance ▾
<input type="checkbox"/>	Fedora24	Image	Active	Yes	Yes	QCOW2	195.1 MB	Launch Instance ▾
<input type="checkbox"/>	CoreOS	Image	Active	Yes	Yes	QCOW2	711.7 MB	Launch Instance ▾
<input type="checkbox"/>	CentOS7-Atomic	Image	Active	Yes	Yes	QCOW2	1005.1 MB	Launch Instance ▾
<input type="checkbox"/>	CentOS6	Image	Active	Yes	Yes	QCOW2	715.6 MB	Launch Instance ▾
<input type="checkbox"/>	Ubuntu14.04	Image	Active	Yes	Yes	QCOW2	247.4 MB	Launch Instance ▾
<input type="checkbox"/>	Ubuntu16.04	Image	Active	Yes	Yes	QCOW2	289.3 MB	Launch Instance ▾
<input type="checkbox"/>	openSUSE13.2	Image	Active	Yes	Yes	QCOW2	395.8 MB	Launch Instance ▾
<input type="checkbox"/>	Fedora23	Image	Active	Yes	Yes	QCOW2	223.5 MB	Launch Instance ▾
<input type="checkbox"/>	CentOS7	Image	Active	Yes	Yes	QCOW2	872.3 MB	Launch Instance ▾
<input type="checkbox"/>	Cirros-0.3.4	Image	Active	Yes	Yes	QCOW2	12.7 MB	Launch Instance ▾

Displaying 11 items

Launch an Instance



Click on the **Launch Instance** button in the row related to the type of image that you prefer

- For this tutorial I'll use **Ubuntu16.04**

Launch an Instance



Launch Instance

Details * Access & Security Networking * Post-Creation Advanced Options

Availability Zone
nova

Instance Name *
test

Flavor * ?
m1.tiny

Instance Count * ?
1

Instance Boot Source * ?
Boot from image

Image Name *
Ubuntu16.04 (289.3 MB)

Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

Number of Instances 0 of 3 Used

Number of VCPUs 0 of 6 Used

Total RAM 0 of 8,192 MB Used

Launch

Choose the flavor **m1.small** for Ubuntu16.04 because the others are too small

Launch an Instance



- A key pair to remotely access the instance
- No keys available, we need to generate it

Launch Instance

[Details *](#) [Access & Security](#) [Networking *](#) [Post-Creation](#) [Advanced Options](#)

Key Pair ?

No key pairs available

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

Security Groups ?

default

Key-pair



Project > Compute > Access & Security > Key Pairs

You have two possibility:

Import an existing Public Key

Create a new Key Pair

Import Key Pair

Key Pair Name *

Public Key *

Description:
Key Pairs are how you login to your instance after it is launched.
Choose a key pair name you will recognise and paste your SSH public key into the space provided.
SSH key pairs can be generated with the ssh-keygen command:
`ssh-keygen -t rsa -f ccloud.key`
This generates a pair of keys: a key you keep private (cloud.key) and a public key (cloud.key.pub). Paste the contents of the public key file here.
After launching an instance, you login using the private key (the username might be different depending on the image you launched):
`ssh -i ccloud.key <username>@<instance_ip>`

Cancel Import Key Pair

Create Key Pair

Key Pair Name *

Description:
Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).
Protect and use the key as you would any normal ssh private key.

Cancel Create Key Pair

Launch an Instance



The instance will be attached on the «private» network that we had previously created

A screenshot of the 'Launch Instance' dialog box in a cloud management console. The dialog has a title bar 'Launch Instance' and five tabs: 'Details *', 'Access & Security', 'Networking *', 'Post-Creation', and 'Advanced Options'. The 'Networking *' tab is active. Under the heading 'Networks *' with a help icon, there is a checked checkbox next to the label 'private'. Below this, the text 'Select networks for your instance.' is displayed. A blue 'Launch' button is located in the bottom right corner of the dialog.

Launch Instance

Details * Access & Security Networking * Post-Creation Advanced Options

Networks * ⓘ

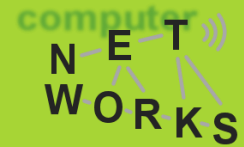
private

Select networks for your instance.

Launch

Skip the other options... and click on **Launch**

Launch an Instance



If everything goes well, you will see the **Power State** of the instance report **Running** (in case of problems, the status will be **Error**)

Project > Compute > Instances

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	test	Ubuntu16.04	192.168.0.102	m1.small	-	Active	nova	None	Running	0 minutes	Create Snapshot ▾

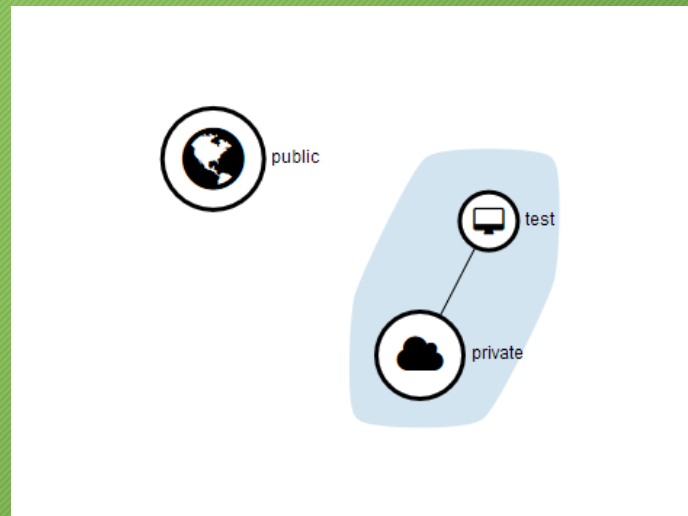
Displaying 1 item

Launch an Instance



And in the network topology you can verify that your instance is attached to the private network

Project > Network > Network Topology



Login into an Instance



How to login into the created instance?

One way is to use the console integrated in OpenStack

- Limited usability
- The login to the Ubuntu image is configured to allow only key-based access

The solution is to access to the Instance via ssh using the key-pair, but... how to reach the Instance via SSH?

Assign a public IP to an Instance



In order to do that we need to let the **private** network be connected to the **public** network

Steps to do:

- Create a Router
- Add to the router ports for both **private** and **public** networks
- Associate an IP from the public network to the instance

Assign a public IP to an Instance



Create a Router:

Project > Network > Routers > Create Router

Create Router ✕

Router Name *

Admin State

External Network

Description:
Creates a router with specified parameters.

Assign a public IP to an Instance



Configure Gateway to the public network:

Project > Network > Routers > Set Gateway

Set Gateway ✕

External Network *

Router Name *

Router ID *

Description:
You can connect a specified external network to the router. The external network is regarded as a default route of the router and the router acts as a gateway for external connectivity.

Assign a public IP to an Instance



Add and interface to the **private** network

Project > Network > Routers > Router Details > Add Interface

Add Interface ✕

Subnet *

private: 192.168.0.0/24 (private-subnet) ▼

IP Address (optional) ⓘ

192.168.0.254 🔒

Router Name *

router

Router ID *

e46ce672-995a-4446-9aa1-15f9ca43385d

Description:

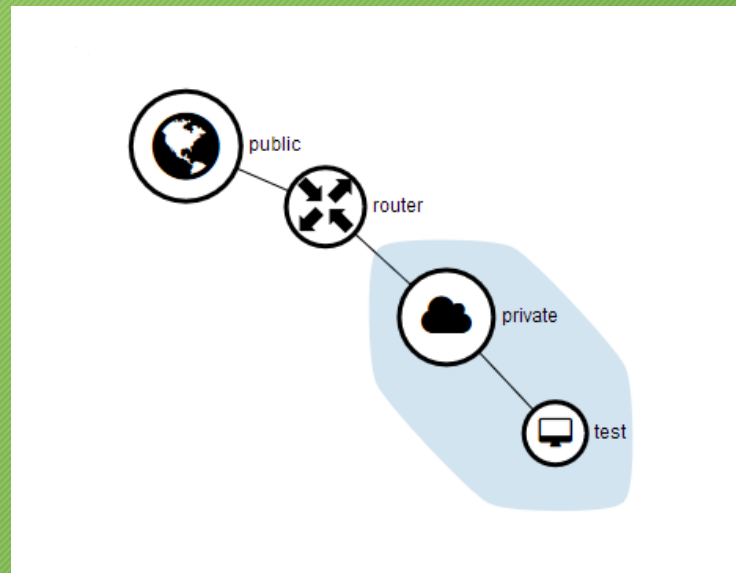
You can connect a specified subnet to the router.

The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.

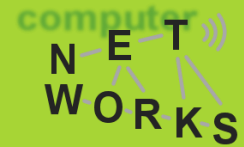
Cancel Add interface

Assign a public IP to an Instance

Network Topology after those operations:



Assign a public IP to an Instance



Finally, assign a floating IP from the public network to the instance

- Floating IPs are IPs from a specific pool configured by the OpenStack administrator. They are not allocated to instances by default, but attached to them only when needed

To attach a floating IP from the public network to the instance:

Project > Compute > Instances > Actions > Associate Floating IP

A screenshot of the 'Manage Floating IP Associations' dialog box. The dialog has a title bar with a close button. It contains three main sections: 1. 'IP Address *' with an empty text input field. 2. 'IP Address *' with a dropdown menu showing 'No floating IP addresses allocated' and a '+' button to the right. To the right of this dropdown is the text 'Select the IP address you wish to associate with the selected instance or port.' 3. 'Port to be associated *' with a dropdown menu showing 'test: 192.168.0.104'. At the bottom right, there are two buttons: 'Cancel' and 'Associate'.

Assign a public IP to an Instance



At first, no floating IPs are associated to the project:
→ Allocate IP

A screenshot of a web-based dialog box titled "Allocate Floating IP". The dialog has a close button (an 'x' in a square) in the top right corner. On the left, there is a dropdown menu labeled "Pool *" with the word "public" selected. To the right of the dropdown, there is a "Description:" section with the text "Allocate a floating IP from a given floating IP pool." Below the description is a "Project Quotas" section. It shows "Floating IP (0)" on the left and "1 Available" on the right. A green progress bar is positioned below the text "Floating IP (0)". At the bottom right of the dialog, there are two buttons: a "Cancel" button and an "Allocate IP" button.

TryStack allows only for one floating IP for the public network

Assign a public IP to an Instance



After a floating IP is allocated, we can assign to the instance

- It is done by associating a port that connects the Floating IP to the Fixed IP
- As we will see, the instance is not aware of the new IP address, but simply all the traffic to the Floating IP will be redirected to the selected fixed IP

The screenshot shows a dialog box titled "Manage Floating IP Associations" with a close button (x) in the top right corner. The dialog contains the following fields and controls:


- An empty text input field labeled "IP Address *".
- A dropdown menu labeled "IP Address *" with the value "8.43.86.94" selected. To the right of the dropdown is a plus sign (+) button.
- A text input field labeled "Port to be associated *" with the value "test: 192.168.0.104" entered.
- A text label to the right of the dropdown: "Select the IP address you wish to associate with the selected instance or port."
- At the bottom right, there are two buttons: "Cancel" and "Associate".

Assign a public IP to an Instance



Final result:

Instances

Instance Name Filter  Launch Inst

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State
<input type="checkbox"/>	test	Ubuntu16.04	192.168.0.104 Floating IPs: 8.43.86.94	m1.small	test	Active	nova	None	Running

Displaying 1 item

Security Group



Last step before login is to make sure that the security rules allow the SSH traffic

- The instance, if not expressed differently during creation, is associated to the **default** security group

Project > Instances

then click on the Instance Name

Instance Details: test

Overview [Log](#) [Console](#) [Action Log](#)

Instance Overview

Information

Name	test
ID	e02c9b40-7656-45f7-bf0a-9a53d07dfc58
Status	Active
Availability Zone	nova
Created	Feb. 23, 2017, 5:35 p.m.
Time Since Created	20 hours, 44 minutes

Specs

Flavor	m1.small
Flavor ID	2
RAM	2GB
VCPUs	1 VCPU
Disk	20GB

IP Addresses

Private	192.168.0.109, 8.43.86.94
---------	---------------------------

Security Groups

default	ALLOW IPv4 22/tcp from 0.0.0.0/0 ALLOW IPv4 1-65535/udp to 0.0.0.0/0 ALLOW IPv4 1-65535/tcp to 0.0.0.0/0 ALLOW IPv4 icmp from 0.0.0.0/0 ALLOW IPv4 icmp to 0.0.0.0/0
---------	--

Manage Security Rules



Project > Access & Security > Security Groups > Manage Rules

Manage Security Group Rules: default (4b8848d5-2cf4-4f69-a2ae-146dad7437b1)

[+ Add Rule](#) [x Delete Rules](#)

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	UDP	1 - 65535	0.0.0.0/0	-	Delete Rule

Displaying 5 items

Add Security Rule



Project > Access & Security > Security Groups >
Manage Rules > Add Rule

Add Rule

Rule *
Custom TCP Rule

Direction
Ingress

Open Port *
Port

Port ?

Remote * ?
CIDR

CIDR ?
0.0.0.0/0

Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Login into an Instance



Finally, let's try to login!

Open a terminal and type:

```
ssh -i <PRIVATE_KEY> ubuntu@<PUBLIC_IP>
```

Attach a Volume to an Instance



Volumes are persistent block storage and can be used for different purposes:

- Boot device
- Portable storage

Attach a Volume to an Instance



Project > Compute > Volumes > Create Volume

Create Volume

Volume Name
test-volume

Description

Volume Source
No source, empty volume

Type
nfs

Size (GB) *
1

Availability Zone
nova

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

Volume Type Description:
nfs
No description available.

Volume Limits
Total Gigabytes (0 GB) 60 GB Available
Number of Volumes (0) 6 Available

Cancel Create Volume

Attach a Volume to an Instance



Project > Compute > Volumes > Actions > Manage Attachments

Manage Volume Attachments

Instance	Device	Actions
No items to display.		
Displaying 0 items		

Attach To Instance

Attach to Instance * ⓘ

Attach a Volume to an Instance



<input type="checkbox"/>	Name	Description	Size	Status	Type	Attached To	Av
<input type="checkbox"/>	test-volume	-	1GB	In-use	nfs	Attached to test on /dev/vdb	no

Displaying 1 item

If you login into the Instance you will see the volume listed ad a disk using the command:

```
> fdisk -l
```

...but you can't really use it until you format and mount it!

instructions on how to do it can be found here: <http://www.darwinbiler.com/openstack-creating-and-attaching-a-volume-into-an-instance/>

Using OpenStack API



Other than the Dashboard, OpenStack resources can be managed through REST API.

- Develop application that interacts with an OpenStack installation
- Control di OpenStack installation with the command line tool (OpenStack CLI)

Using OpenStack API



To use OpenStack API, we first need the credential for authentication:

Project > Compute >
Access & Security >
Download
OpenStack
RC File

Access & Security

Security Groups Key Pairs Floating IPs API Access

[Download OpenStack RC File](#) [Download EC2 Credentials](#) [View Credentials](#)

Service	Service Endpoint
Compute	http://8.43.86.2:8774/v2/e4c2b7f003b34f4aab4f48bed4a30eea
Network	http://8.43.86.2:9696
Volumev2	http://8.43.86.2:8776/v2/e4c2b7f003b34f4aab4f48bed4a30eea
Computev3	http://8.43.86.2:8774/v3
S3	http://8.43.86.2:8080
Image	http://8.43.86.2:9292
Metering	http://8.43.86.2:8777
Volume	http://8.43.86.2:8776/v1/e4c2b7f003b34f4aab4f48bed4a30eea
EC2	http://8.43.86.2:8773/services/Cloud
Object Store	http://8.43.86.2:8080/v1/AUTH_e4c2b7f003b34f4aab4f48bed4a30eea
Identity	http://8.43.86.2:5000/v2.0

Displaying 11 items

Using OpenStack API



The RC file contains the information to login through the keystone service and obtain a token to interact with OpenStack services

Let's try to use the OpenStack Command-Line client

- instructions on how to install and use it can be found here: <https://docs.openstack.org/user-guide/common/cli-install-openstack-command-line-clients.html>

Using OpenStack API



How to use the OpenStack Command-Line client:

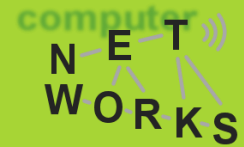
- Execute the content of the RC file
- Call an OpenStack client command

```
jacopo@JUG0E:~$ source openrc.sh
Please enter your OpenStack Password:
jacopo@JUG0E:~$ openstack server list
```

ID	Name	Status	Networks	Image Name
e02c9b40-7656-45f7-bf0a-9a53d07dfc58	test	ACTIVE	private=192.168.0.109, 8.43.86.94	Ubuntu16.04
b6f2cd08-a789-4f64-b607-b7fa1ef8f8c2	test2	ACTIVE	private=192.168.0.108	Cirros-0.3.4

```
jacopo@JUG0E:~$ █
```

Real Use-case scenario:



- Deploy a Web-Server
<https://help.ubuntu.com/lts/serverguide/httpd.html>
- Create a distributed Wordpress platform
<https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-16-04>
 - first do a single-server instance
 - then deploy the Web-Server and the Database int two different instances

Let's try!

Sources and Useful Links



- <http://trystack.org/>
- <https://docs.openstack.org/admin-guide/>
- <https://docs.openstack.org/user-guide/>
- <https://docs.openstack.org/developer/devstack/>
- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/3/html/Getting_Started_Guide/part-Using_OpenStack.html
- <https://www.youtube.com/watch?v=D4mmZUZIM-k&list=PLz0c5JNg5UJYc9ISCZBviBUC1BB7dB5nk>