



Tools & Concepts for Cloud Deployments

Exercise 1: OpenStack

Christopher B. Hauser

Institute of Information Resource Management, Ulm
University

Summer Semester 2019

Exercise 1: OpenStack

Overview

This exercise focuses on the basics of Infrastructure as a Service Cloud Computing. The following lessons first introduce Virtual Machines / Hypervisors, then make you familiar with OpenStack. With this foundation, we will use the Mediawiki Software (used for Wikipedia) as an exemplary cloud application throughout this course.

Lessons

1. OpenStack Basics
2. First Steps with OpenStack
3. Mediawiki Installation

Exercise as PDF

Exercise Solution

- Solution as Markdown
- Solution as PDF

Lesson 1: OpenStack Basics

Introduction

In this lesson, we will introduce OpenStack - the most popular software for cloud data centres. But before we start with OpenStack, some basics are necessary.

Research: Hypervisors and Virtual Machines

While we will discuss the concept of virtualisation and virtual machines in-depth in the lectures, we require some information prior to the hands-on session. For that reason it is necessary that you get some basic understanding of virtual machines.

Please research the web for some definitions. Reading the wikipedia page is sufficient for the start. You will get more knowledge in the course of the lecture.

Question: Hypervisors and Virtual Machines

- What are the basic tasks of a hypervisor?
- What are benefits / drawbacks of using virtual machines compared to physical servers?

Research: Remote Server Management

In the next lesson we will manage a Linux server from remote. If you are not already familiar, it is now the time to learn the basics about:

- **Secure Shell (SSH):** how to use it, how to use key-based authentication
- **Firewalls:** required protocols/ports for ssh, web access, and PINGs

Lesson 2: First Steps with Omistack

Introduction

In this lesson, we will register an account for the OpenStack installation at hand. **Registration may take a while** - so please register in time, in order to proceed with all remaining tasks, lesson 3 and all the following exercises.

Task: Register at the OpenStack

Please resume to the **next task only, when you have access to OpenStack**. When you have access, make yourself familiar with the OpenStack dashboard. Navigate through the links at the left to remember the dashboard's sections later on.

Task: Create an SSH key

In OpenStack, Instances (Virtual Machines, usually Linux servers) are managed via a SSH connection from your pc. For security reasons, authentication is not with a password but with keys (cf. lesson 1).

You can create an SSH key within the OpenStack dashboard. Select "Key Pairs" in the menu. Then press "Create Key Pair". Choose "cloud_key" as "Key Pair Name", and press the "Create Key Pair" button.

OpenStack will start downloading the private key as a file after it has created the Key Pair. It is *important that you save and keep this file*, as you will need it later on and there is no actual way make OpenStack deliver it again. Backup this private key! If you lose your private key, you won't be able to access Instances (Virtual Machines) any more!

Alternatively, you can upload an SSH key, which you generated already on your pc, e.g. using the ssh-keygen tool.

Task: Adapt Security Rules

Before we can work with OpenStack, we need to check the firewall rules to allow remote access. These firewall rules are organised within "security groups". In the OpenStack dashboard, go to "Network" and "Security Groups". You should see only the "default" security group. Click on "manage rules". The rules should be similar to the following ones:

Task: Validate your Network

Before starting your first Instance, a basic virtual network is required, if you want to access and hence use your new instance via OpenStack's virtual network infrastructure.

In the OpenStack dashboard, click the "Network" item in the menu, then select "Networks". You should see at least one entry in the networks table. The networks listed depend on the OpenStack deployment and may differ.

+ Add Rule
✕ Delete Rules

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Ingress	IPv4	Any	Any	-	default	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv6	Any	Any	-	default	Delete Rule
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv6	ICMP	Any	::/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv6	TCP	22 (SSH)	::/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule

Displaying 9 items

Figure 1: rules for security group default

Virtual Networks will be studied in more detail in exercise 3. If you miss the network, please inform your instructor.

If you see only an external network (*e.g. working with Omistack*), then you need to create a private network first: - Press the "Create Network" button, provide the name "private", and in the "Subnet" tab state as "Network Address" e.g. 192.168.1.0/24, then press next and finish. - Go to "Routers", press "Create Router" button, provide the name "private-router", select a external network and press the button to finish. The router needs to be linked to your private network: select the router by clicking on its name, then select "interfaces", press the button "Add Interface", select your own network named "private" and press submit.

Task: Launch your first Instance

Virtual machines in OpenStack are called "Instances". Let's start your first one, by launching an Ubuntu Server. To go "Compute", then "Instances". Then click the "Launch Instance" button on the top right.

In the "Launch Instance" popup, in the "Details" step:

- Name your instance e.g. "main_server"
- Select "nova" as availability zone (or "blade" if no nova exists)

Continue to the "Source" step:

- Select Boot Source: Image
- Create New Volume: No
- Select in the list "Ubuntu Server 16.04" with the arrow button

In the Flavor tab:

- Select "small" as flavor

In the Networks tab:

- The default network should be automatically selected. If not, select it.

In the Security Groups tab:

- Select the "default" security group

In the Key Pair tab:

- select the key you previously created (cloud_key)

Finally, launch the virtual machine, by clicking the "Launch Instance" button. This may take from some seconds to a few minutes. The new instance gets an IP address assigned automatically.

Depending on your OpenStack deployment, this IP address may be private or public. If you have a private IP, to access the instance from remote, add a so called "Floating IP". If you have a public IP, skip the following steps.

If you have a private IP, add a Floating IP to your instance:

- while the instance is spawning, press the “Associate Floating IP” button, after it was spawned, find this option in the drop down menu next to the “create snapshot” button.
- If you see “No floating IP addresses allocated”, then press the “plus” button to allocate a floating IP in the available pool.
- Select the new floating IP and press the “Associate” button.
- Your instance has now two IP addresses: the private one from before, and a public one. You need this public one to access your instance via SSH.

Task: Access your Instance via SSH

It is time to access your new instance via SSH with your favourite SSH client. Make sure you have your SSH key file at hand. To access your instance, you need the public IP address from your instance. This IP address is listed in the OpenStack dashboard, where you launched it.

The **SSH username** for the ubuntu image is **ubuntu**, a password is not set since we use key based authentication.

Some more hints to access via SSH:

Windows For Windows users, we suggest using putty in combination with puttygen. An instruction on how to make use of the OpenStack key with putty is available here in the sections “Converting a Key” and “Using the key with putty”.

Linux / Mac In Linux or Mac, an SSH client is shipped with almost all distributions. Using a key can be enforced by applying the `-i` command line parameter (for details type “man ssh” in a terminal).

If you’re asked “Are you sure you want to continue connecting (yes/no)?” you can accept with typing “yes” and pressing enter.

Lesson 3: Install Mediawiki Application

Task: Install Mediawiki Application

As we are now able to access the shell of the virtual machine, we will install the mediawiki application in the next steps. The basic strategy and a good overview is given on this page. While there are multiple ways to install mediawiki, this and this guide give the best overview. We follow both guides and execute the described steps.

Before we can use the package system (apt-get) of Ubuntu, we should make sure everything is up-to-date.

1. `sudo apt-get -y update && sudo apt-get -y dist-upgrade`
2. `sudo reboot`

Afterwards the server will restart and you need to reconnect to it using ssh.

First, we need to install multiple dependencies required for running the mediawiki installation. As mediawiki is a php application we need a webserver serving the pages, the php5 runtime engine for executing the code and a database as storage backend. We execute the following steps:

1. Install the Apache Webserver - `sudo apt-get install apache2`
2. Install the MariaDB database. - `sudo apt-get install mariadb-server`
3. Install php5 - `sudo apt-get install php php-mysql php-mbstring php-xml libapache2-mod-php`
4. Restart the apache server - `sudo systemctl restart apache2`

Now we can start to install the mediawiki application:

1. Make sure you are in your home directory: `cd ~`
2. Download latest version of mediawiki (currently 1.30.0):

```
wget https://releases.wikimedia.org/mediawiki/1.30/mediawiki-1.30.0.tar.gz
```
3. Extract the tarball: `tar -xvzf mediawiki-*.tar.gz`
4. Switch to the http document root directory of apache2: `cd /var/www/html`
5. Create a symbolic link to the downloaded mediawiki software: `sudo ln -s ~/mediawiki-1.30.0 wiki`

You should now be able to see the initial page of mediawiki at `http://{ip-of-your-vm}/wiki`. If the Browser keeps loading and ends with a timeout, check again the Security Groups (is there a Port 80 as suggested in lesson 1?)

Please note:

- don't worry if the sudo command will print an "unable to resolve host warning"
- you should not need to worry about the version number of mediawiki, choose the latest version found on the mediawiki page

- you cannot download the files from your browser as you want to install the application in a VM. Instead you have to download it from within the VM for instance by using the `wget` or `curl` tools. (Or you have to later copy them to the VM using `scp`)
- MariaDB is a fork of MySQL. It offers the same client and SQL Syntax. So don't be confused if some of the later points mention `mysql`.
- don't worry if the restart command of the `apache2` server outputs a warning (server name not found). Everything is fine and we will address this problem later on.

Task: Setup Database

When accessing your wiki in the current state, you will be notified that you need to configure it first. For this purpose we first of all need to set up the database.

To secure your database, first run `sudo mysql_secure_installation`, accept all default suggestions, and define a root password (use it later in "PASSWORD_FROM_BEFORE").

1. Connect to the database using the `mysql` command line client: `sudo mysql`
2. Before we continue, we have to fix the root password: `UPDATE mysql.user SET password=password('PASSWORD_FROM_BEFORE') WHERE user='root'; UPDATE mysql.user SET plugin='' WHERE user='root';`
3. In the following prompt enter the password you created for the database in the installation step.
4. Create a new database for the wiki: `CREATE DATABASE wikidb;`
5. Create a new database user and grant him all privileges for the database, replace password by a password of your choice and remember it: `GRANT ALL PRIVILEGES ON wikidb.* TO 'wikiuser'@'localhost' IDENTIFIED BY 'password';`
6. Updated the privileges: `flush privileges;`
7. Exit the `mysql` client: `exit`

Notes:

every command within the `mysql` command line needs to end with a semicolon (;) in the current state your database is only accessible from the virtual machine, thus you need to open the `mysql` command line while being connected to the VM

Task: Configure the mediawiki installation

The mediawiki installation can be configured using a web-based installation wizard, that is available when clicking the setup the wiki link on your wiki page.

When the installation wizard asks you for the database setup change the database name to wikidb (remember, you created it earlier), the database user to wikiuser (you created it earlier) and the database password to the one you chose when creating the wikiuser.

When the installation wizard asks you for a name of the wiki and a username and password, enter your name for the wiki and chose a username and password you can remember. Select "Ask me more questions." before pressing the "continue" button.

In the options overview, uncheck the box "Enable outbound mail".

All other options can safely be confirmed by pressing the continue button.

Finally, the wizard will install the wiki and you will automatically download a LocalSettings.php that needs to be copied into the wiki folder of the server.

For this task there are multiple solutions, depending on the system where you downloaded the LocalSettings.php:

(Linux/Mac) use scp to copy the file to your server:

```
scp LocalSettings.php ubuntu@134.60.64.95:~/mediawiki-1.30.0/
```

(Windows) use the graphical <https://winscp.net/eng/index.php> to copy the file inside the folder mediawiki-1.30.0/ in the ubuntu home directory.

(Both) use an editor like vim or nano to open and create a LocalSettings.php on your vm and then paste the content of the downloaded file.

Finally, you can access your new mediawiki installation via http://134.60.64.Y/wiki/index.php/Main_Page

Question: One Instance for Database and Application

So far, we have installed the mariaDB database server on the same server as the mediawiki application and linked them using the localhost address.

Can you explain why this is good for performance in the current case, but may hinder the performance in the long run?

What alternative approach do you suggest?

In the last exercises, we have deployed and configured a Web-based application (mediawiki) in the cloud.

However, your mediawiki installation is currently running on only one virtual host, which is hindering scalability and elasticity. This is definitely not good enough for a large installation as the Wikipedia Foundation uses it for the well-known Wikipedia.

Can you find out what we need to do to cater for a larger, multi-node installation of mediawiki?

The Mediawiki architecture overview https://www.mediawiki.org/wiki/Manual:MediaWiki_architecture may prove helpful.

Task: Create a snapshot

To backup the work you have done in the previous steps, and to be able to restore your virtual machine quickly, please create a snapshot of your virtual machine.

In the instances overview of the OpenStack dashboard, click the “Create Snapshot” Button next to your instance.

This process may take some time.