



DVI566 - Int. to Cloud Computing

Booklet for Laboratory 2

The purpose of Lab 2 is to practice with the fundamental cloud mechanisms and resources, that are

- Creation and use of VM¹, virtual storage, perimeters (Virtual Private Cloud - VPC)
- Monitoring of resource usage
- Virtual networking

Before starting the Lab you are required to create an AWS Educate account. See the documentation on Canvas.

An overview of the services available at AWS is on the white paper

https://d1.awsstatic.com/whitepapers/aws-overview.pdf?did=wp_card&trk=wp_card

Of your interest are the sections on: AWS Management Console; AWS Command Line Interface; Compute Services; Storage; AWS Identity and Access Management (subtopic of Security Identity and ; Amazon VPC, AWS CloudWatch

Laboratory 2 is composed of three tasks

1. Setup and run an amazon EC2 instance
2. Monitoring the performance of an EC2 instance while under CPU and disk pressure.
3. Connect two VMs inside a perimeter (VPC) and monitor the performance while the instance is under web requests pressure.

Before starting the lab, I strongly suggest you: (i) to read all the steps for each experience; (ii) to figure out what are the goals and activities to be performed; (iii) to study the documentation related to the tools you are going to use in the experience.

Assessment (U/UX/G): You will be evaluated on the basis of the submitted report. Specific/punctual instruction are not given to let you master the concepts and strengthening your critical thinking and problem solving capability.

¹ In the Amazon Web Service jargon, a virtual machine is called EC2 instance. In this document, I will use the terms virtual machine (VM) and EC2 instance interchangeably.

The lab is designed for linux instances and, in particular, for Ubuntu AMI. However, you are free to chose other type of Linux AMI or Windows instances,

I Task I - Setup and run an amazon EC2 instance

The goal of this experience is to get familiar with: Amazon EC2 and related services: AWS storage options, security groups and VPC. You will practice that concepts creating and launching a VM on the Amazon cloud. This task consist of

1. Creating a security group,
2. Creating a Virtual Private Cloud (VPC)
3. Creating a virtual machine and run the virtual machine in the VPC
4. Access the virtual machine, from your computer, using SSH or the AWS console

I.1 Resources

Useful resources for this task are

1. The following sections of the “Amazon EC2 User guide (for linux instances)”
https://aws.amazon.com/documentation/ec2/?icmpid=docs_menu:
 - What is amazon EC2
 - Setting UP
 - Getting Started
2. The following sections of the CloudWatch user guide
<http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>
 - What is Amazon CloudWatch?
 - Getting Started
 - Using Dashboards
 - Using Metrics
 - Using Alarms
3. The user guide for Virtual private clouds, available here
http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html

2 Task 2 - Monitoring VMs

The goal of this experience is to learn how to monitor VMs in a cloud environment. This task consists of

1. *To launch an EC2 instance* (e.g. t2.micro). To observe the CloudWatch statistics access the CloudWatch service console.
2. *To create your own dashboard with the metrics you wish to monitor*. For example, you can create a dashboard with the following graphs: CPUUtilization, DiskReadOps, DiskWriteOps, DiskReadBytes, DiskWriteBytes. Be aware that the t2.micro uses ESB storage so you have to select CPU utilization from EC2 metrics and Disk related metrics from EBS metrics

3. *To generate workload for the VM to practice with monitoring.* Two type of workload should be generated:
 - CPU and disk intensive workload generated with tools like sysbench, and stress-ng. Instructions to use sysbench follow in section 2.2. If you have previous experiences with other stress tools/benchmark tools you can use those.
 - Web workload generated with ab (Apache bench) or JMeter. Instruction to use ab follow in section 3.2

Tip. After you have launched the VM I suggest updating the package database, e.g. running apt-get update

2.1 Resources

Useful resources for this task are

- The “Monitoring Amazon EC2” how-to guide. Focus on Automated and Manual monitoring, status check, detailed monitoring (1-minute period), ESB monitoring, CloudWatch and available CloudWatch performance metrics. The guide is available at http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html
- The CloudWatch user guide <http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>
- The sysbench documentation; suggest links are <https://github.com/akopytov/sysbench>
<https://www.howtoforge.com/how-to-benchmark-your-system-cpu-file-io-mysql-with-sysbench>
<https://wiki.mikejung.biz/Sysbench>

2.2 Examples of how to generate load with sysbench

First, you need to install sysbench. Access your EC2 instance and run the commands

```
sudo apt-get update
sudo apt-get install sysbench
```

Then, to check that sysbench works, you run the command

```
sudo sysbench --test=cpu --cpu-max-prime=200000 run
```

If everything work properly you can do a more intense CPU stress test. For example, the following command

```
for each in 1 2 4 8 16 32 64; do sysbench --test=cpu --cpu-max-prime=200000 --num-threads=$each run; done
```

runs sysbench for 7 times and at each run the number of threads is increased starting from 1 up to 64.

The following steps are an example of how to stress the disk. First you need to create a file set and then you can perform read and write operation on the files. To create the file set run the following command

```
sysbench --test=fileio --file-total-size=8G --file-num=3 prepare
```

An example of command to run the read and write test for 5 times, with an increasing number of threads (from 1 to 32) is the following:

```
for each in 1 4 8 16 32; do sysbench --test=fileio --file-total-size=8G --file-test-mode=rndwr --max-time=240 --max-requests=0 --file-block-size=4K --file-num=3 --num-threads=$each run; done;
```

Refer to the sysbench documentation and tutorials to understand the meaning of the command's options.

IMPORTANT: to observe a significant variation of the metric values, the amount of workload to be generated depends on the capacity of the EC2 instance you selected. Change the values of the benchmark's parameters given in the above examples according with the EC2 instance you selected.

3 Task 3 - Communication

The goal of this experience is to learn how to connect two VMs in a VPC. This task consists of

1. To create a new - non default -VPC (or using the one created before if non-default)
2. To launch two VMs, let us call them VM1 and VM2, in the VPC
3. To install apache web server (the apache httpd) on VM1 and apache2-utils on VM2 (you can eventually use an image that already has these tools pre-installed)
4. To stress the apache httpd on VM1 with the Apachebench (ab) running on VM2. The two VMs should communicate using their private IP address. When running the test, monitor the network traffic on the VM1 (e.g. byte per seconds and number of active connections).

3.1 Resources

Useful resources for this task are:

- Amazon web service documentation on networking <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-networking.html>
- Amazon web service documentation on security <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security.html>
- Apache web server documentation <http://httpd.apache.org/>
- ApacheBench documentation <https://httpd.apache.org/docs/2.4/programs/ab.html>

3.2 An example of ApacheBench stress test

ApacheBench (ab) is a benchmark that generates http requests. First you should create a web page on the http server. Then, with ab, you can simulate multiple users requesting a web page. The simplest syntax of the ab command is

```
ab -n <num_of_request> -c <num_of_concurrent_users> <your_web_site>
```

for example

```
ab -n 100000 -c 10 http://ip-10-251-50-12.ec2.internal/index.html
```

send 100000 http requests, generated from 50 concurrent users, to the web server hosted on the VM with private IP address 10.251.50.12 and with internal hostname “ip-10-251-50-12.ec2.internal”

Note that the intensity of the load you generate depends on the value of the ‘n’ and ‘c’ parameters in the ab command. Play with the number of http requests (n) and the number of concurrent users (c) to observe significant values of the performance metrics.

4 Assessment

The group should produce a report, that **could be integrated with screen recording** (a feaster way to show the system at work -- please add audio or text comments) describing:

1. The configuration of your security group and VPC. You should motivate for your configuration choices
2. The results of your performance test (Task 2 and Task 3). You should describe:
 - What are the performance metrics selected and why (motivate your choices)
 - How you generated the load (describe the scripts used and motivate your choices)
 - The meaning of your results (why you get such results and what they mean)

Always remember: to motivate your choices; to discuss the issues encountered and how you solved those issues.

No specific template for the report is provided. You can use the BTH template for reports or your own template. In the first page of the template, please specify the group number and group members

The assessment is pass or fail (U/Ux/G). I hope that all the group members will give a valuable contribution to the work. However, do not hesitate to contact me if there are group's members not contributing.