



ÅBO AKADEMI UNIVERSITY

CLOUD COMPUTING

## Assignment 4



FILIBE FELÍCIO (2004623)

MAY 5, 2021

## Elastic web services: resources auto-scaling

### 1) A short description of the steps you took to implement your elastic web service

I choose to create an Auto Scaling group by creating a launch template from an existing EC2 instance. This were the steps I took to implement it:

1. Launching the first EC2 instance
2. Connecting to the instance using ssh
3. Install all the required packages to run the task such as PHP and Apache (like in a previous assignment)
4. Modify the `/etc/rc.local` file by adding `sudo /usr/sbin/httpd start` command (like in a previous assignment)
5. Install the vi text editor
6. Create an index.php file and edit it with the provided code using vi (like in a previous assignment)
7. Create a AMI image from this instance
8. Launch more instances from the AMI
9. Launch a load balancer
10. Create a Auto Scaling Template
11. Create Auto Scaling group from the Template.

This was the code used in the index.php file:

```
1 <?php
2 $time_start = microtime(true);
3 echo "Hello! My name is write-your-name-here and my IP address is: ".
   $_SERVER['SERVER_ADDR'];
4 // Generate a random integer from 200 to 10000
5
6 $random=rand(200,1000);
7
8 $a = Array( Array(1,2,3,4),Array(4,5,6,7));
9 $b = Array( Array(7,5,7,2), Array(3,2,8,2));
10
11 //variable number of loop
12 for ($x=0; $x <= $random; $x++)
13 {
14 //number crushing
15 $c = array();
16 for($i=0;$i<4;$i++)
17 {
18     for($j=0;$j<4;$j++)
19     {
20         $c[$i][$j]=0;
21         for($k=0;$k<4;$k++)
22         {
23
24             $c[$i][$j]=$c[$i][$j]+($a[$i][$k]*$b[$k][$j]);
25         }
```

```

26     }
27 }
28 }
29 echo "<br> random value was:". $random;
30 $time_end = microtime(true);
31 $execution_time = ($time_end - $time_start);
32 echo "<br> Execution time:". $execution_time. " second";
33 echo "<br> Done !";
34 ?>

```

## 2) The used httpperf command(s) to create load on the instances

```

1 httpperf server loadbalancer-1385128410.us-east-1.elb.amazonaws.com
  uri =/index.php wsess =300,5,2 rate 1 timeout 5}

```

3) Screenshots and link(s) to online video(s)\* showing the CPU utilization across the instances over adequate time windows. You should annotate the graph(s) and/or videos with the corresponding events on their timelines (like start of httpperf execution(s), instance creations, instance terminations, etc.).

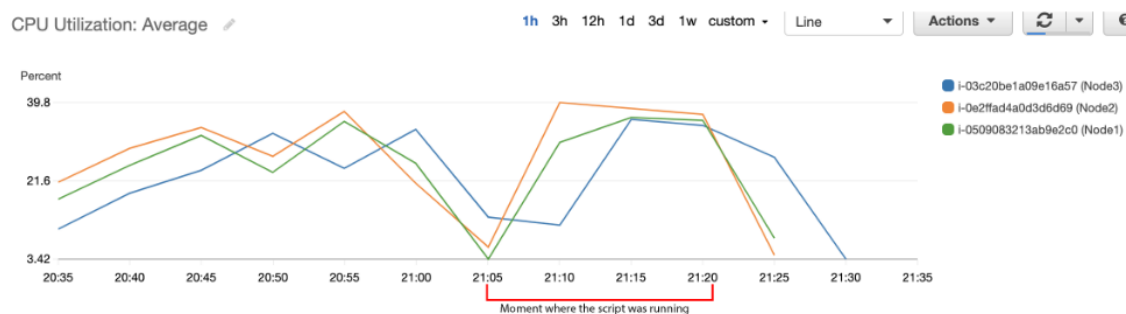


Figure 1

4) A detailed analysis of the obtained behavior and reactivity of your web service to load variations (increases and decreases of load). With the obtained behavior and reaction time, provide possible (concrete) web applications that could benefit from your auto-scaling mechanism for cloud resources.

During the execution of the httpperf there was an increase in the use of the second node to help the first. After the load balancer added the third node and the load stabilized. A lot of web applications would benefit from auto-scaling mechanism for cloud resources, such as, the fault tolerance that comes from the fact that a faulty instance can be terminated and launched in another availability zone and also the capability of adapting the number of instances to the incoming traffic allowing to optimize the performance/cost relation. In particular, some examples of such applications are:

- The Application Load Balancer works with many AWS services, including:
  1. eAWS Auto Scaling
  2. Amazon Elastic Container Service
  3. AWS CloudFormation

4. AWS CodeDeploy
  5. AWS Certificate Manager
- [Safira](#) - a Platform developed to managed everything related to the annual event: Software Engineering Week at Universidade do Minho, organized by student center, that needs hold hundreds of users with an inconstant rate of requests' traffic. For the organization that has limited resources auto-scaling mechanism are necessary.
  - [lichess.org](#) - A Chess Web Application that holds live games, tournaments, ELO system. and everything a chess platform ecosystem needs to provide.

## Conclusion

This chapter gave me a useful insight of another big advantage of using Cloud Computing. It becomes clear to me as an Software Engineering Student that using cloud services is a best option that having our own servers in most cases. In particular, this feature of launching and terminating instances with a load balancer is a huge plus, specially in cases where a application/product/platform needs for periods of time to support an unusual traffic of incoming request, the alternative would be buying more hardware which would mean investment that would only be needed for some of the time which it is not feasible. Load Balancing in Cloud Computing comes as the best solution to optimize cost and performance without the need to abdicate of either.