# 2019 - CS**352** – **A**ssignment **4**:

# **S**elfie**L**ess**A**cts on **AWS** – **Amazon**’s **A**pplication **L**oad **B**alancer

* Team : CC\_308\_313\_331\_352

|  |  |  |  |
| --- | --- | --- | --- |
| **Member Name** | **USN** | **Section** | **Effort(in # hrs.)** |
| **Rohan** R | 01FB16ECS**308** | F | 2 |
| **Rohit** U Bogulla | 01FB16ECS**313** | F | 3 |
| **Sailesh** Gaddalay | 01FB16ECS**331** | F | 2 |
| **Sharath** N | 01FB16ECS**352** | F | 3 |

Date of Evaluation : Wed, **04th Apr** 2019 Evaluator : **Mr.Akash Nagaraj**

1. AWS username on which demo was shown : **vocstartsoft/user216090=rohitub222@gmail.com**
2. Account ID : **5276-2484-3223**
3. Summarize your learning as part of this assignment :

As the part of assignment, we extended the prescribed functionality by adding the new APIs.

A typical application that is hosted on the web, should provide concurrency to the users and reliability to requests obtained. This is achieved through Amazon’s Application Load Balancer. Load balancing distributes traffic across the servers (acts & users) and it improves the responsiveness of the application.

ALB consists of *Target Groups* and *Listener Rule*. *Target Groups* routes the requests to the registered target(instance) under it. And the incoming requests are filtered using *Listener Rule* and the target group attached to it will be fired and requests will be forwarded to it. Specifically, our ALB uses *path-based routing* i.e., load-division is achieved using the route path mentioned in the request.

Apart from this, dockers and containers were similar to previous assignment but in different instances.

So, all of the above knowledge was a blueprint using which we developed this task.

­­­­­­

Any other observations/challenges/comments :

Our major challenge was to test our setup in the CCBD Testing website, where most of the times we got ‘Invalid Credentials’ even when they were correct (This happened not only to us, but to most of the other teams as well.)