# 2019 -CS352 - Assignment 4: SelfieShare Load Balancer

| Member Name | USN | Section | #hrs effort |
| --- | --- | --- | --- |
| Maanvi Nunna | 01FB16ECS187 | D | 6 |
| Meghana Ganesh | 01FB16ECS205 | D | 4 |
| Nandagopal NV | 01FB16ECS221 | D | 5 |
| P Abhishikta Sai | 01FB16ECS242 | D | 5 |

Date of Evaluation 13/04/19 Evaluator : Srinivas Shekar

1. AWS username on which demo was shown

[abhishikta.sai21@gmail.com](mailto:abhishikta.sai21@gmail.com)

Users Instance IP : 34.236.27.23

Acts Instance IP : 18.210.123.131

Load balancer DNS : http://load-balancer-570395361.us-east-1.elb.amazonaws.com/

1. Summarize your learning as part of this assignment

We set up Amazon’s load balancer, created two target groups and used it to perform path based routing to our two instances. The paths and actions that were added in the load balancer are:

-Forward to users instance if these two paths are seen : **api/v1/users/\*** and **api/v1/users**

-Otherwise the request is redirected to the acts instance for any other path by default

We also restricted the ports to 22 and 80 on both the instances, as this had to be changed from the previous assignment where we set the port numbers as 8000 and 8080. Next, when the acts microservice has to make a request to the users instance, we made sure that the Origin header, that is editable in the CORS function of flask and python, where we specified the IP address of the acts instance. Also, we added the required count APIs to the two python-flask files on both the instances. Lastly, the number of acts API was also added to the acts instance.

Any other observations/challenges/comments

Upon checking the working of our instances against the script provided, we repeatedly got the message something went wrong, try again. This we thought was a problem of not mentioning the ports properly however after retrospecting our APIs we realized that the count APIs were just returning the number instead of a number enclosed in square brackets. We found the test script’s error messages quite misleading in this respect. Fixing this issue passed all the test cases.

There is a small observation and we are not sure as to why it is happening- when we run our backend on both the instances, there are two IPs(identity unknown to us) which are constantly trying to access our instances and the backend sends back a 404 BAD request. As far as we know, it is not affecting the instances and we speculate that this might be due to editing the CORS origin header.