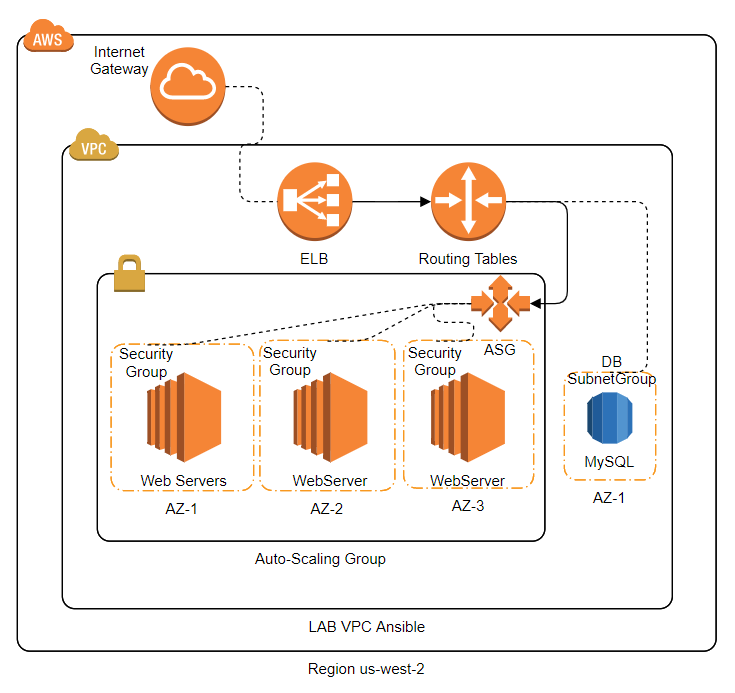
Project Presentation

Greetings

The next diagram shows the architectural deployment design which has been built to support the Web application service.



Deployment:

* This application was developed using python
* Supported by a MySQL databased deployed on RDS from AWS.

Scalability:

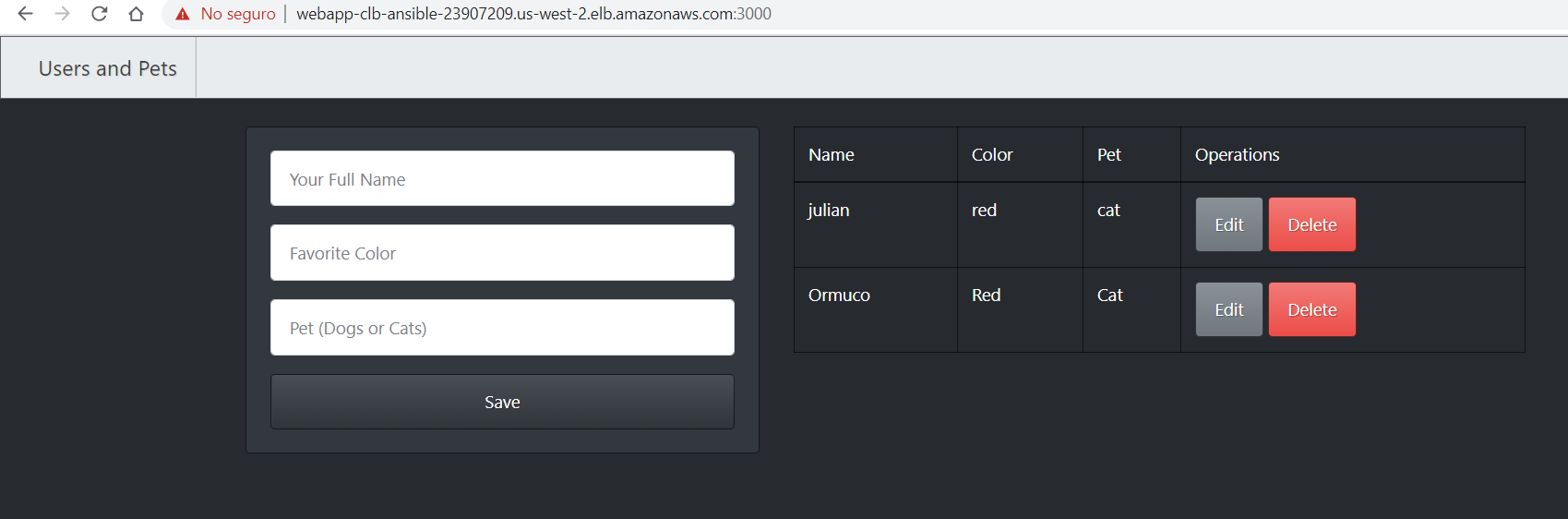
* The instances are part of an autoscaling group and a load balancer, which allow fast failure detection, auto-healing features and horizontal scalability as well as support for high traffic demands, eventually.
* Since the database has been deployed on a managed server is almost infinitely scalable

Security

* The route tables, Network Access Control Lists (NACL) and security groups have been implemented to avoid intrusion or security breaches.
* The database is not publicly accessible, and by firewall policies only the instances within the same VPC can establish communication, adding an additional security factor.
* The web server certificates implementation is in place for the web service.

Testing Data

* URL for you to test:

<https://webapp-clb-ansible-23907209.us-west-2.elb.amazonaws.com:3000/>

* The application has been shared on github in the next link

<https://github.com/jjulianprin/FlaskProject>

For your ease an entries view has been implemented in the same web page. Also the options to edit and delete registers.

* The key shared by you and requested to be included in the server is now in place on authorized keys file.
* SSH is restricted through the load balancer, the public ip address from the actual instance at this moment is 52.13.216.19 (please let me know in any case, so I can double check for the moment when you are doing the connection)
* All the architecture was deployed using ansible playbook, moreover the instances configurations were done using ec2 instances’ user data, specified in the ASG. This is in order to guarantee that all the servers are ready to work in case that the servers goes unavailable by any reason and an auto-scaling event is triggered. This script is also provided as UserDataScript.sh.
* The ansible playbook to perform the cloud deployment is provided for you to review. The name of the file is environmentdeploy.yml
* An ansible playbook was also developed to deploy this application on cloud but ad-hoc. This corresponds to the file WebServerPlaybook.yml; also attached.
* The boto file is also attached for you to be able to connect to AWS and see the architectural design.

It has been a very challenging and fun experience, I did enjoy it a lot. Please don’t hesitate to let me know in case you may require further details on the architectural design or have any other doubts or questions about this exercise.