CSE 546 - Cloud Computing
Project-1 : Individual Contribution Portfolio Report

Kedar Sai Nadh Reddy Kanchi
School of Computing and Augmented Intelligence
Arizona State University, Tempe
kkanchi@asu.edu

Individual Contribution

My individual contribution to the project started of with configurations of the IAM role, group and user. IAM roles are used to manage the permissions and access of AWS resources like SQS and S3 from the Instance itself. To configure the IAM roles, I set up the policies required for the IAM instance profile to run the program. This involved creating an all-access policy that would allow the instance to access all the necessary AWS resources, including SQS and S3.

But mainly my contribution to the project was focused on the implementation of the controller and app files. Specifically, I designed the logic for the auto-scaling algorithm in the controller.py file based on the demand (depth) of the request queue. I implemented the scale-out function when the current instances running are less than the maximum count and the total messages in the queue are greater than the current instances.

I also implemented the logic for each user data to run in Ubuntu by the app.py file for each instance. This involved getting the requests from the request queue and storing the images in input buckets. After processing the retrieved images using the deep learning model, I pushed the results in the response queue and stored them in output buckets.

In addition to the implementation of the controller and app files, I also tested the auto-scaling algorithms for different requests concurrently by monitoring the queues and buckets. This testing was an essential part of ensuring that the application could handle numerous concurrent requests and dynamically adjust its scaling based on the depth of the request queue.

Furthermore, I collaborated with other team members to ensure that the application was well-documented and easy to use. I helped to write the report using the provided template and created the README file in the git repository, which detailed the installation requirements and the steps to run the application.

Lessons Learned

During our project, we embarked on the journey of building an elastic cloud application that could automatically scale in response to user demand, all while keeping a watchful eye on cost-effectiveness.

Our cloud app had a clear mission: to deliver an image recognition service using deep learning to analyze user-provided images. While the deep learning model itself was handed to us, the responsibility lay in crafting an application that could make practical use of this model and meet the typical requirements for a cloud-based service.

Our application reinforced the idea that cloud services should cater to the needs and expectations of users. This underscores the importance of understanding the end-users and tailoring the service to deliver value and meet their requirements effectively.

In terms of the technical aspects of building a cloud application, one key lesson we learned was the necessity for efficient scaling. Our app was designed to handle multiple requests concurrently, which meant it needed to automatically scale out when demand increased and scale in when demand decreased. This dynamic scaling helped us make the most of our limited resources, staying within the 20-instance limit dictated by the free tier. We learned that implementing an automated scaling mechanism, based on the depth of the Request SQS Queue, is crucial for optimizing resource utilization in a cloud environment.

Furthermore, the project highlighted the importance of data persistence. Storing all inputs and outputs in Amazon S3 allowed us to ensure data integrity and recoverability. We discovered the significance of proper data organization within S3, with each object stored as a key-value pair.

The project also emphasized the need for speed and accuracy. Cloud applications should be optimized to handle requests swiftly and accurately. Our image recognition service had to meet these criteria, as users expected quick responses, and the recognition results had to be correct. This reinforced the importance of rigorous testing and optimization in delivering a dependable and efficient cloud service.

In conclusion, our journey to build an elastic image recognition application on AWS has provided us with a wealth of lessons that are foundational for cloud application development. These lessons, ranging from the choice of the right IaaS provider to effective scaling, data persistence, and user-centered service design, have equipped us with essential knowledge and skills that will undoubtedly benefit us in building future cloud applications. As we move forward, we'll continue to apply and refine these lessons to create even more robust and efficient cloud solutions.

References

- AWS Documentation