



AWS cloud solution for AspireClothes E-commerce

Capstone Project 4

Submitted to IN4.0 Group

1. Abiola Abiodun
2. Sameh Mostafa
3. Faisal Ali
4. Abdul Baig

March 2021

Table of Contents

1. Introduction	2
1.1. Adaptive e-commerce solution	2
1.2. Scalable serverless e-commerce website with 0% commitment	2
1.3. Low level architecture	3
1.4. High level Architecture	5
1.4.1. Description of how the architectural network works	6
1.5. Cost estimate	8
1.5.1. Assumptions	8
1.6. Pre-requisite for running this architectural model	9
2. Conclusion	10
3. Reference	10
4. Appendix	11

Table of figures

Figure 1: E-commerce Store Interactions	2
Figure 2: Simple architecture of an e-commerce store	3
Figure 3: Low level architecture showing groups of AWS services for security, monitoring and storage	4
Figure 4: High level architecture for AspireClothes E-commerce store	5

List of tables

Table 1: Selected AWS technologies used	6-7
Table 2: Cost Estimate for the services used for our architecture	8

Word count: 1503 (excluding tables)

1.0. Introduction

1.1. Adaptive E-commerce solution

E-commerce stores has fluctuating web traffic, depending on seasonal and marketing promotions like Black-Friday. To cut cost, modern e-commerce stores are deployed in cloud environments which permit scaling up and down, depending on the expected traffic. In this way organizations only pay for what they use, and always provide their users/customers with the right experience. Typical e-commerce store interactions and transaction is presented in a flowchart illustrated in Figure1 showing the different operation by users and operators .

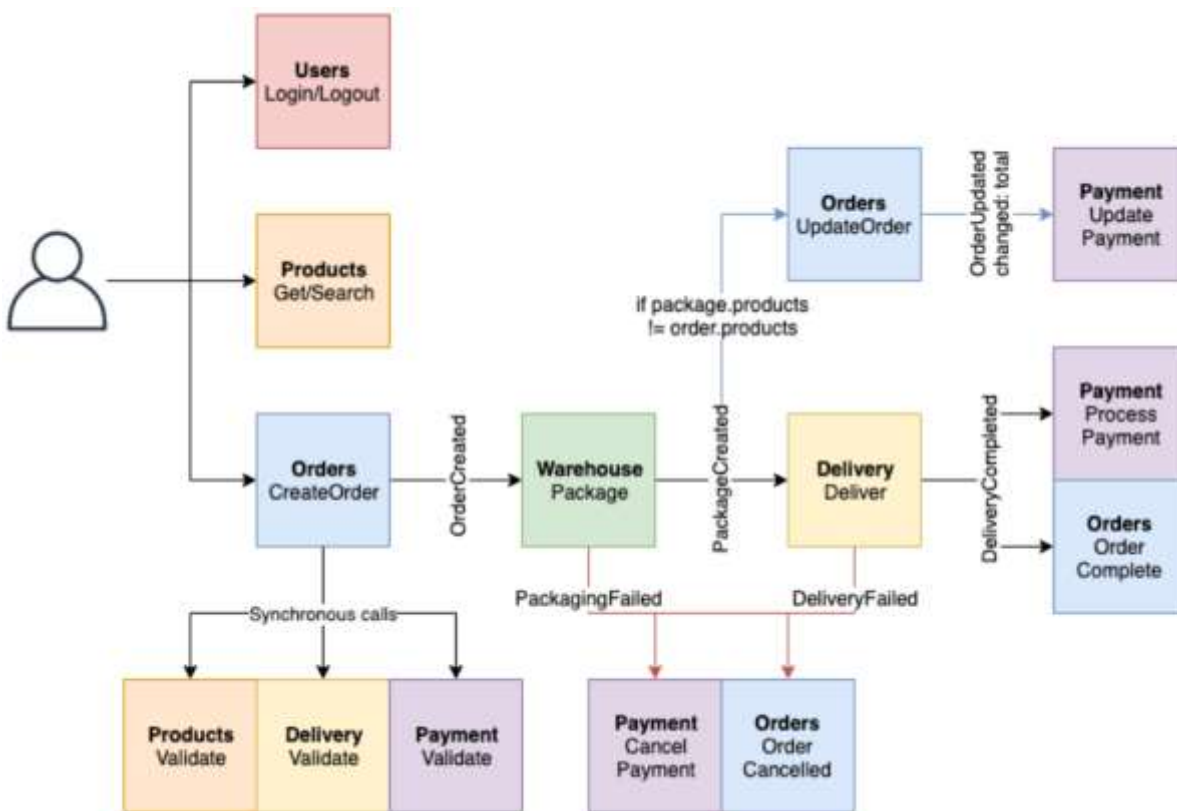


Figure 1: E-commerce Store Interactions

1.2. Scalable serverless e-commerce website with 0% commitment

Amazon Web Services offers e-commerce cloud computing solutions to small and large businesses that want a flexible, secured, highly scalable, and low-cost solution for online sales and retailing^[1]. There are range of services to choose from to develop these solutions, whether server-based or serverless, but we are going to focus on serverless systems illustrated in figure 2 because server-based system incurs continuous expenses as the website remain live. However, serverless system automatically activate to perform an operation and shutdown without incurring charges for the idle time.

We will build a real-world use case of AWS serverless solution for AspireClothes. This solution will meet the following requirements:

- High availability, scalable, low latency and highly secured
- Products listings, details, and carting.
- Customers' ratings and communication
- Supply chain channels
- Payments and delivery channel
- Create account and login pages
- Real-time analytics of traffics.

1.3. Low level architecture

Find below a summarise schematic of a low-level architecture.

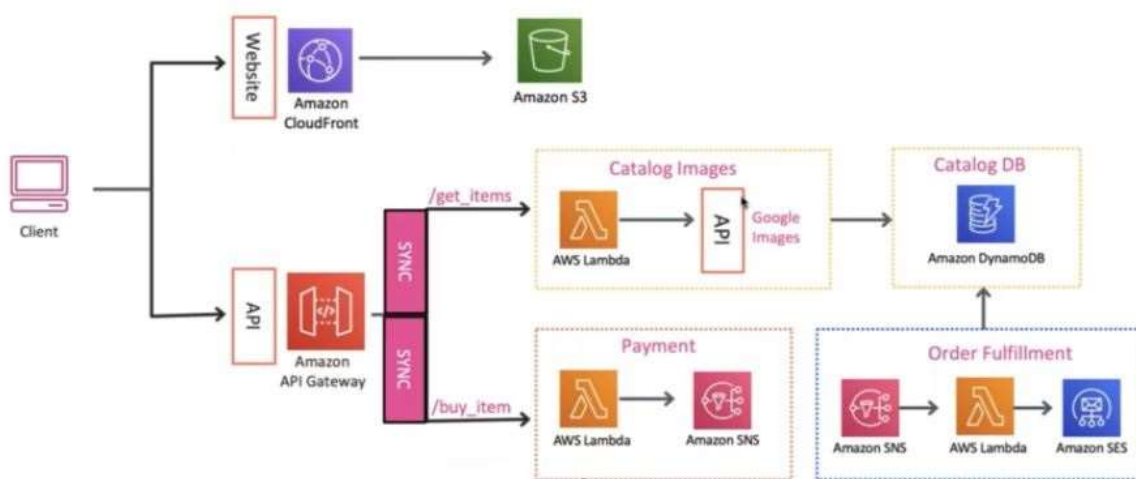


Figure 2: Simple architecture of an e-commerce store

A simple e-commerce store for a start-up such as AspireClothes will require a simple adaptive architecture that is loosely coupled and allowed for easy scalability on expansion as it grows. Example of such simple architecture with basic components interaction closely grouped is illustrated in figure 3.

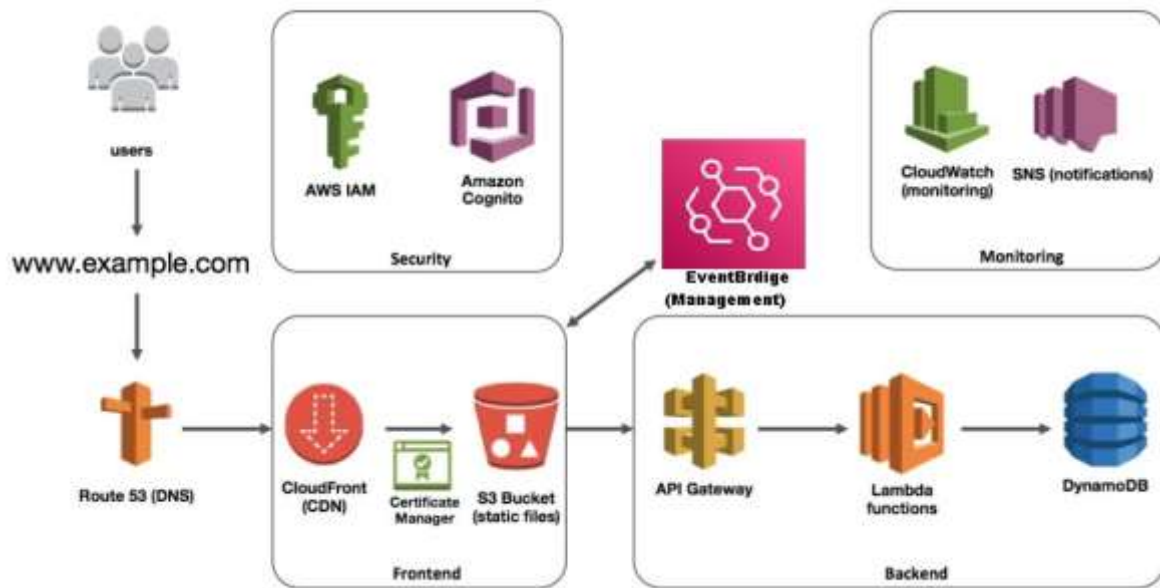


Figure 3: Low level architecture showing groups of AWS services for security, monitoring and storage

Figure 3 describes an adaptive e-commerce store architecture deployed on AWS. Objects such as product listing on the webpage and other files needed for hosting the static sections of the websites are stored on S3 bucket for low latency. Lambda functions along with API gateway and the DynamoDB ensure carting, checkout, and payment. For user registration and login, Amazon Cognito provides users with identities and data synchronisation. For security, the backend section of the website is deployed in a private network. Adopt AWS security best practice along with IAM for internal authentication and API gateway authorisation or adopting AWS shield for database security against DDOS. CloudWatch for monitoring store's traffic and SNS for traffic/sales notifications and SES for marketing and communication with customers. For a thorough and a more detailed architecture, check out figure 4 illustrated below.

1.4. High level architecture

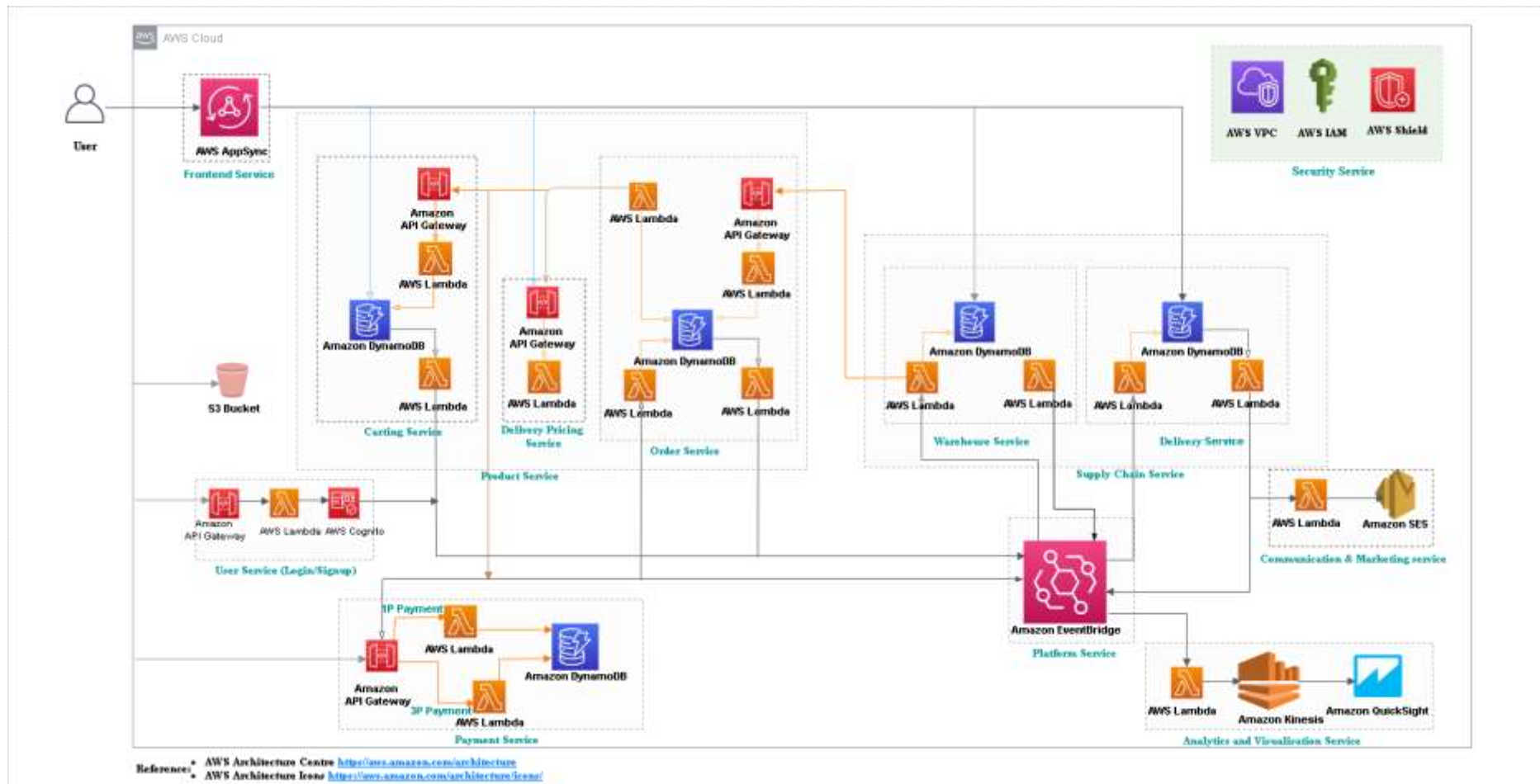


Figure 4: High level architecture for AspireClothes E-commerce store

Figure 4 illustrates a high-level AWS architecture for AspireClothes with 100% serverless based and deployed on AWS Serverless Application Model (AWS SAM). The databases are deployed on DynamoDB for logs and Aurora serverless for analytics to increase flexibility. The web layer is controlled by Amazon EventBridge which makes it possible to seamlessly connect different events or sections of the web including third-party services adopted for either user sign-on and supply chain automation or Amazon Cloud Front and the routing done by Amazon route 53. To add extra layer of security to AWS VPC on the serverless cloud, we use IAM for internal authentication, AWS Shield for DDOS prevention and finally QuickSight for real-time data analytics and visualisation. The architecture above is for only one availability zone, therefore data snapshots and backup should be duplicated in other availability zones. To withstand system failure or high traffic from seasonal sales like Black Friday, elastic load balancer and Route 53 can be used to redirect traffic to other zones. The architecture was created using the official Amazon Web Services Cloud icons.

1.4.1. Description of how the architectural network works

When a client or customer visit AspireClothes website via public internet connection, they are directed by AppSync to get/search for the products stored in the S3 bucket(s). Their activities are recorded in the product service (defined by lambda functions and stored in our DynamoDB table). When the customer choose a product or add to cart, the operation is redirected to the EventBridge which trigger payment service (direct payment or 3rd party payment like PayPal) and warehouse service which either check for the product in shelf or contact suppliers. All these operations communicate with the database via lambda functions. When the order is complete, EventBridge send a notification to the administrator via AWS SNS to confirm sales and a confirmation of purchase to the customer. A delivery service is also setup. The generated structured data from these transactions are stored on Amazon serverless Aurora. The administrator can further analyse and visualise these data in real-time(QuickSight) either monthly or quarterly to identify market trends and key customers as well as build a communication channel with them. This will also be done with a lambda function.

Table 1: Selected AWS technologies used

Services	Technologies used	Reason
Compute	AWS Lambda is serverless compute either behind APIs or to react to asynchronous events.	To save cost on infrastructure (instance)
Storage	<ul style="list-style-type: none"> Amazon DynamoDB is a scalable NoSQL database for persisting information. To stick with serverless storage for a more structured data (SQL), we used Amazon Serverless Aurora. 	DynamoDB was adopted because of low latency and serverless nature.

Monitoring and Analytics	<ul style="list-style-type: none"> QuickSight was used for real-time analysis and monitoring of customer's logs and transactions stored in Serverless Aurora. 	CloudWatch is for EC2, hence QuickSight is best suited
Communication Messaging	<ul style="list-style-type: none"> AWS AppSync for interactions between users and the ecommerce platform. And SES to support. Amazon API Gateway for service-to-service synchronous communication (request/response). Amazon EventBridge for service-to-service asynchronous communication (emitting and reacting to events). AWS SNS and SES for marketing and communication with mailing list. 	Real-time serverless data query and communication
Security (Authentication & Authorization)	<ul style="list-style-type: none"> Amazon Cognito for managing and authenticating users and providing JSON web tokens used by services. AWS Identity and Access Management for service-to-service authorization, either between microservices (e.g. authorize to call an Amazon API Gateway REST endpoint), or within a microservice (e.g. granting a Lambda function the permission to read from a DynamoDB table). AWS Shield is connected to CloudFront and Route53 for DDOS protection. 	For highly secured system
Continuous Integration and Continuous Delivery (CI/CD)	<ul style="list-style-type: none"> AWS Serverless Application Model for defining AWS resources as code in most services. AWS Cloud Development Kit (CDK) for defining AWS resources as code in the payment-3p service. Amazon CodeCommit as a repository to trigger the CI/CD pipeline. Amazon CodeBuild for building artifacts for microservices and running tests. Amazon CodePipeline for orchestrating the CI/CD pipeline to production. 	

1.5. Cost estimated

In this section, we will estimate the costs of AspireClothes web application based on some usage assumptions and Amazon's pricing model. All pricing values used here are from Q1 2021 in EU-West-2 region and can be verified using AWS pricing calculator ^[2].

1.5.1. Assumptions:

For our pricing example, we will assume that AspireClothes online store will receive the following traffic per month: 100,000-page views, 1000 registered user account, 200GB data transferred considering an average page size of 2MB. 5,000,000-code executions (Lambda functions) with an average of 200 milliseconds per request.

Table 2: Cost Estimate for the services used for our architecture

Service	Quantity	Charges (month)	Estimates (per month)
AWS AppSync	1	Free	It's free for the first 12months and subsequent variable charges
AWS EventBridge	1	varies	For 5million events, charges = \$5.00 (event publishing) + \$2.86 (archive processing) + \$0.66(storage) + \$5.00 (replaying) = \$13.52 per month
API Gateway		varies	Charges \$3.50 per million of API calls received and \$0.09 per GB transferred out to the Internet. If we assume 5million requests per month with each response with an average of 213KB, the total cost of this service will be \$ 17.93.
S3 Bucket	1	\$0.023 per GB stored.	Normally \$0.023 per GB/month stored, and \$0.004 per 10,000 requests and \$0.09 per GB transferred. However, with CloudFront usage, transfer costs will be reduced and a total estimate of \$0.82 for our 10,000 traffics.
AWS CloudFront	1	varies	For 10,000-HTTPS request North America 70%-\$0.085, \$0.010; Europe 15%-\$0.085, \$0.02; Asia 10%-\$ 0.140, \$0.012; South America 5%-\$0.250, \$ 0.022. As we have estimated 200GB of files transferred with 2,000,000 requests, the total will be \$21.97. Total: \$ 21.99.
AWS Route53	1	\$0.50	We need to buy a domain name which cost \$0.5/month. Also, we need to pay \$0.40/million DNS queries to our domain. 100,000-page views will cost only \$0.04. Total of \$0.54/month
Amazon Cognito	1	\$0.0055	First 50,000 monthly active users is free, subsequent charges is \$0.0055/month. Further charged for Cognito Syncs of user profiles. It costs \$0.15 for each 10,000 sync

			operations and \$0.15 per GB/month stored. An estimate of 1,000 active and registered users with less than 1MB per profile, with less than 10 visits per month in average, we can estimate a charge of \$0.30. Total: \$ 0.30
AWS Lambda	16 (+/-)	\$0.8/msec	First 1 million requests are free, you pay for 4 million that will cost US\$ 0.80. With compute service, 5 million executions of 200 milliseconds each gives 1million seconds. As we are running with a 512MB capacity, it results in 500,000 GB-seconds, where 400,000 GB-seconds of these are free, resulting in a charge of 100,000 GB-seconds that costs \$1.67. Total:\$ 2.47
AWS DynamoDB	5	varies	\$0.47 per month for write, \$0.09 per month for every read, \$0.25 per GB/month stored (first 25GB free), \$0.09 GB per GB transferred out to the Internet. \$0.1/million for data capture for analytics. Assuming 5M read, 5M write and 5M others, estimate becomes \$10.44
Amazon Aurora (serverless)	1	\$0.06	8hrs x 21days x \$0.06 = \$10.08
AWS IAM	1	free	Charges for only the resources user consume.
AWS QuickSight	1	\$0.6/hr	maximum charge of \$5/reader/month for unlimited use after first two months free.
Amazon SNS	1	free	Free for our case-study
Amazon SES	1	varies	\$0 for the first 62,000 emails you send each month, and \$0.10 for every 1,000 emails sent after that. \$0.12 for each GB of attachments sent
AWS Shield	1	\$0.025/GB	Highly expensive but secure the CloudFront.
AWS VPC	1	\$0.01/hr	\$10 a month
Certificate manager	1	free	Certificate manager provide SSL/TLS certificate

Based on the estimate in table 2, a total of \$125 will be required to kickstart the e-commerce store on AWS. Any additional expenses might take it to about \$150.

1.6. Pre-requisite for running this e-commerce design

- A staff with basic computer skills (administrator)
- AWS command line Interface (CLI) or console
- AWS serverless application model (AWS SAM)
- An AWS Account with permissions to create: AWS Identity and Access Management (AWS IAM) roles, AWS Simple Storage Service (Amazon S3) buckets, Amazon DynamoDB tables, and AWS Lambda functions, AppSync, EventBridge and others.

2.0. Conclusion

AspireClothes required a cost-effective AWS architectural plan to migrate their e-commerce store to the cloud. After quick review of the CEO demands and other basic requirements, a low level-architecture was used to illustrate the services needed and their interactions. After a thorough analysis of cost with respect to AWS best practices, a high-level architecture was generated and illustrated in figure 4. A cost estimate for the first month of deployment was further established and the key of administrators needed to manage the service.

3.0. References

- [1] AWS, Ecommerce Cloud Hosting - Ecommerce Applications. Accessed 16 March 2021[online]. Available: <https://aws.amazon.com//ecommerce-applications/>
- [2] AWS Pricing calculator. Accessed 12 March 2021[online]. Available: <https://calculator.aws/#/>
- [3] AWS architecture icons. Accessed 13 March 2021[online]. Available on <https://aws.amazon.com/architecture/icons/>
- [4] Seongtaek Oh, AWS architecture e-commerce, Slideshare, May 19, 2016. Accessed 12 March 2021[online]. Available: <https://www.slideshare.net/SEONGTAEKOH1/awsarchitectureecommerce>
- [5] Savia Lobo, a serverless online store on AWS could save you money, Packtpub hub, June 14, 2018. Accessed on 14, March 2021[online]. Available on: <https://hub.packtpub.com/a-serverless-online-store-on-aws-could-save-you-money-build-one/>
- [6] AWS serverless e-commerce platform, AWS github repository. Accessed 14 March 2021[online]. Available: <https://github.com/aws-samples/aws-serverless-ecommerce-platform#technologies-used>

4.0. Appendix

Server-based architecture concept to also consider.

