COS20019: Assignment 3

Team 21

Team member	Student ID
Nguyen Dinh Long	102877730
Tran Hoang Hai Anh	104177513
Nguyen Duc Minh	103432499

1. Introduction

This report presents a thorough architecture design for the new system that takes into account the problems and requirements found while utilizing managed cloud services to reduce the need for internal administration work.

With AWS S3, manage cloud services as follows: The suggested architecture will make use of managed cloud services from Amazon Web Services (AWS) to reduce internal systems administration and increase scalability. The AWS Simple Storage Service (S3) will store all the media, including pictures and videos. This approach permits easy scaling as demand increases, guaranteeing affordability, high availability, and data longevity.

We expect demand to double every six months for the next two to three years, considering the exponential growth in application usage in terms of scalability for future growth. The architecture will be created to be highly scalable in order to support this growth. The system will be able to handle an increase in user load without compromising performance thanks to AWS services like Elastic Load Balancing, Auto Scaling, and Containerization (using, for example, Docker with Amazon ECS or Amazon EKS).

We will put into practice a well-optimized scaling approach to increase the EC2 Instance Compute Capacity while addressing the performance restriction of the current t2.micro EC2 instances. In order to maintain a computing capacity between 50% and 60%, the application will use AWS Auto Scaling to dynamically adjust the number of EC2 instances based on the current load. By doing this, overloading will be avoided, and peak performance will be maintained.

The architecture will adopt a serverless/event-driven model as we increase operational effectiveness and cost-effectiveness to achieve the adoption of serverless/event-driven solutions. For a variety of application tasks, AWS Lambda functions will be used, guaranteeing automatic scaling, minimal operational overhead, and pay-as-you-go pricing. Event-driven triggers will take care of various media processing tasks, resulting in an extremely responsive and flexible system.

The limitations of the current slow and expensive relational database can be overcome by migrating to a more cost-effective database solution, which is what we recommend. AWS provides suitable options, such as Amazon DynamoDB, a fully managed NoSQL database that offers high performance, scalability, and a pay-per-request billing model, taking into account the straightforward table structure [4].

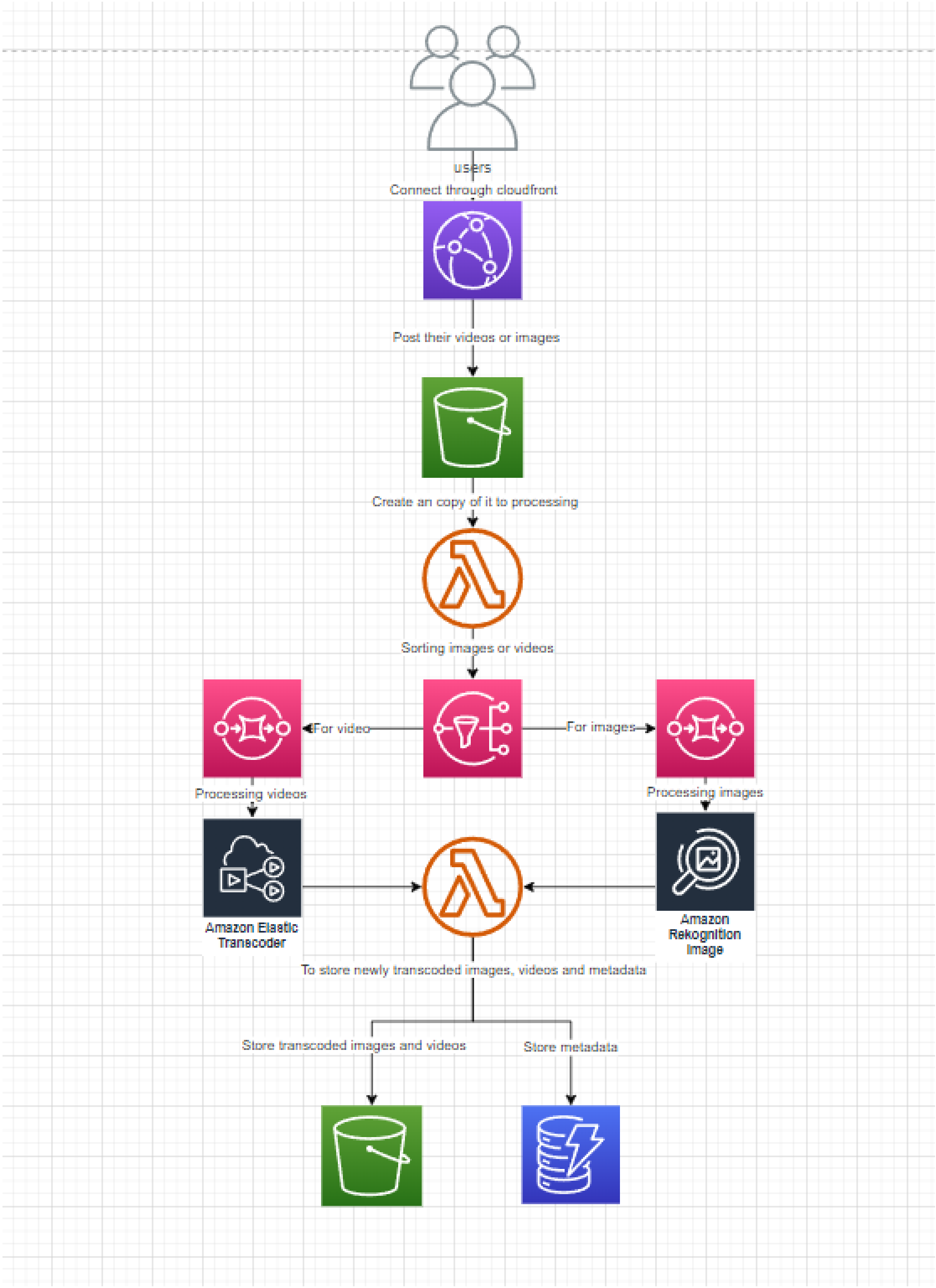
Global Response Time Optimization is used to address slow response times in international locations, the architecture will leverage AWS Global Accelerator or Amazon CloudFront. By enabling content caching and delivery through global edge locations, these services significantly lower latency for users all over the world and improve response times.

As the system is anticipated to handle video media in the future, support for video media can be implemented, and the proposed architecture will be created with flexibility in mind. Using AWS Elemental MediaConvert or Elastic Transcoder, we can automatically convert uploaded videos into various formats suitable for different devices, ensuring a seamless user experience.

The architecture will incorporate a robust and extensible design for media reprocessing and reformatting. We propose utilizing AWS Step Functions to orchestrate the media processing workflow. The process will involve triggering the creation of alternative versions of media items automatically upon upload to S3. The architecture will be designed to accommodate future AI-based tasks, like automated tag identification for photos. Depending on the cost and performance requirements, the most appropriate platform will be used to carry out the media processing tasks.

AWS services like EC2 instances and AWS Lambda will be used to ensure quick and economical processing. By implementing a queue-based system, the architecture will also introduce a decoupled approach. Jobs for media transformation will be put in a queue and processed by a number of "worker" nodes simultaneously. Each worker node can specialize in tasks such as video transcoding or image reformatting to ensure optimal resource usage and reduce processing times.

2. Architectural Diagram


users
Connect through cloudfront
Post their videos or images
Create an copy of it to processing
Sorting images or videos
For video
For images
Processing videos
Processing images
Amazon Elastic Transcoder
Amazon Rekognition Image
To store newly transcoded images, videos and metadata
Store transcoded images and videos
Store metadata

3. UML Collaboration Diagram

Uploading media

User ⇒ CloudFront: Upload Media

CloudFront ⇒ AWS S3: To store Media

AWS S3 ⇒ Lambda: Trigger Sorting images or videos

AWS Lambda ⇒ Amazon SQS: Add Task to Queue

Processing Images

Amazon SQS ⇒ Amazon SNS: For images

Amazon SNS ⇒ Amazon Rekognition image: Adding tagging for images

Amazon Rekognition image ⇒ Lamda: To store newly transcoded images into s3 and their metadata to RDS database

Processing Videos

Amazon SQS ⇒ Amazon SNS: For videos

Amazon SNS ⇒ Amazon elastic transcoder: Processing videos

Amazon elastic transcoder ⇒ Lambda: To store newly transcoded videos into s3 and their metadata in the RDS database

4. Description of all AWS services used

CloudFront: Content delivery network (CDN) service Amazon CloudFront is offered by Amazon Web Services (AWS). It is intended to provide users with content that has low latency and high transfer speeds, such as web pages, videos, images, and other static or dynamic resources.

S3: Amazon Web Services (AWS) offers the popular object storage service, Amazon Simple Storage Service (Amazon S3). It provides highly available, secure, scalable storage for a variety of data types, including backups, log files, backup images, and more.

Lambda: AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It enables you to run code without provisioning or managing servers, making it easier to build and deploy real-time applications and services that respond to events.

SNS: AWS SNS (Amazon Simple Notification Service) is a fully managed messaging service provided by Amazon Web Services. It enables you to send notifications and messages to a large number of subscribers or endpoints, including individuals or systems, via various communication protocols like email, SMS, HTTP, and more. SNS simplifies the process of delivering messages to multiple recipients and allows you to decouple the sender from the receiver, making it a fundamental component for building scalable and decoupled distributed applications.

SQS: AWS SQS (Amazon Simple Queue Service) is a fully managed message queuing service provided by Amazon Web Services. It enables you to decouple the components of your distributed applications by allowing them to communicate asynchronously through messages. SQS queues act as temporary repositories for messages, ensuring that messages are reliably stored and delivered to consumers in a scalable and fault-tolerant manner.

Amazon elastic transcoder: Amazon Elastic Transcoder was a service provided by Amazon Web Services (AWS) that allowed you to convert media files from one format to another. It provided a scalable and cost-effective solution for transcoding videos, audio files, and images, making them compatible with various devices and platforms.

Amazon Rekognition image: Amazon Rekognition Image is a service provided by Amazon Web Services (AWS) that offers advanced image analysis and recognition capabilities using deep learning and computer vision. It allows you to extract valuable information from images and analyze them for various use cases, such as facial recognition, object detection, image moderation, and content-based searching.

DynamoDB: Amazon DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS). It is designed to deliver fast and predictable performance at any scale, making it an excellent choice for applications that require low-latency, high-throughput data storage and retrieval. DynamoDB is built to be highly available, durable, and scalable, making it well-suited for a wide range of use cases.

5. Requirements and Justification

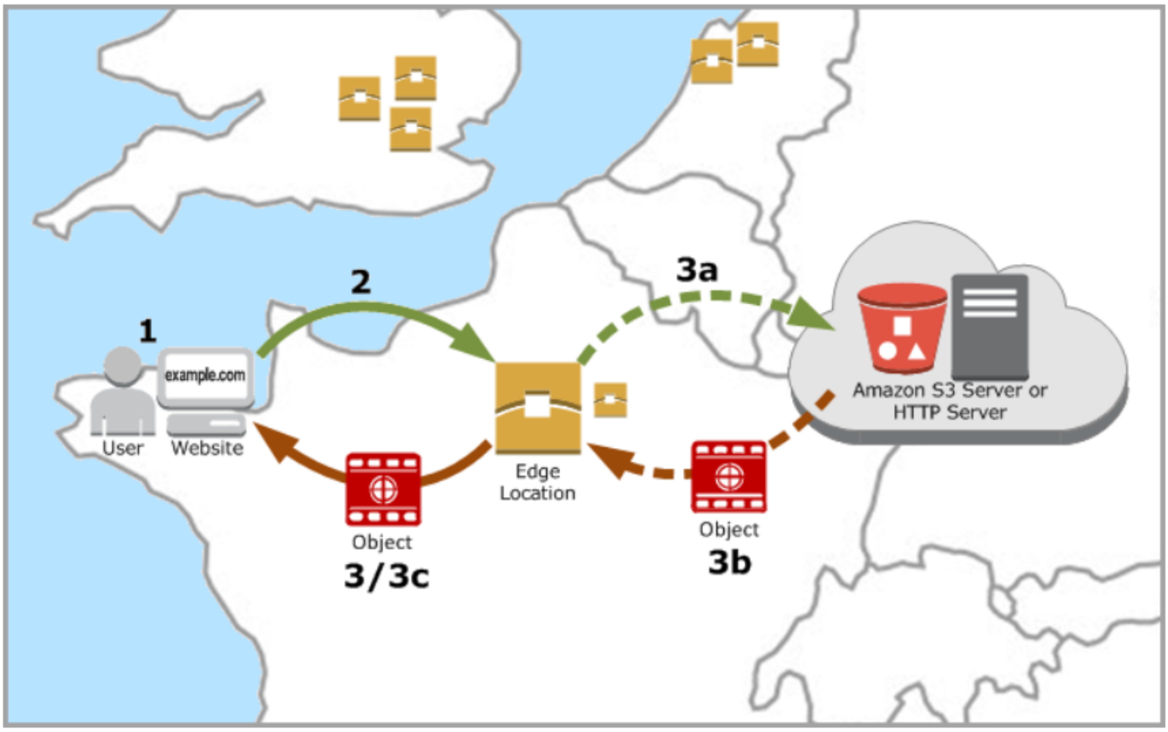
AWS S3

The Photo Album application had a wonderful success, and the company reported that its demand growth has been doubling every six months. Using AWS S3 is a suitable solution to cope with this increasing need for the Photo Album application. According to team Blazeclan [1], Amazon S3 is a low-latency and highly-scalable cloud service for data storage, so users can store as much data as they want without worrying about future data usage. Moreover, the company also stated that they want their users to be able to upload media in all sorts of formats, and Amazon S3 can satisfy it easily. Other benefits of using Amazon S3 are strong security, high availability, low cost, easy management, and simple operation. Compared to EFS, Amazon S3 is better at storing data for long term and backing-up data. Therefore, it can guarantee high availability and data longevity.

AWS CloudFront

The company addressed that the number of users was increasing significantly (including users around the world), and the response time in areas outside Australia was slow. In order to solve this problem, AWS CloudFront should be used because this online service can increase users' accessing speed to the Photo Album application. CloudFront has multiple edge locations, they are linked together to form a worldwide network or data centers. It can reduce the global response time by finding the edge location which has the lowest latency, routing the user request to it, and delivering the content back to the user. The resource of the content can be retrieved from an HTTP server, a bucket of Amazon S3, or a MediaPackage channel.

According to Amazon [2], this is the diagram illustrating how CloudFront works:



The company will reduce response times and latency by using Amazon CloudFront since this service will deliver content from the nearest edge location when the user requests it.

AWS Lambda

As the company requires a serverless/event-driven solution, AWS Lambda should be used for the Photo Album application. According to AWS [3], Lambda has the ability to execute your code in response to events and maintains the underlying compute resources for you automatically. Therefore, when the user uploads a media item to the S3 bucket, it automatically triggers media processing. Lambda can scale automatically and run functions in response to events, so the company can ensure cost efficiency and real-time processing. Moreover, when compared to EC2, AWS Lambda can save more money since EC2 charges users for the time the instance is running. Therefore, AWS Lambda is a suitable service for the company to solve its problem.

ElasticSearch

Elastic represents a robust and dynamic open-source search and analytics engine often synonymous with the broader concept of the Elastic Stack or Elasticsearch. With its unparalleled capabilities, it stands as a stalwart solution for managing vast volumes of data, offering a seamless blend of real-time search and advanced analytical functionalities.

Elastic can be used to improve the search functionality. Features such as filtering photos and media items based on various attributes such as tags, descriptions, and metadata. When photos and media items are indexed in Elastic, the search efficiency will improve drastically. It was mentioned in the business scenario that the database is slow and costly to run. By utilizing Elastic, we can definitely reduce the burden that normal search operations would add to the database system.

DynamoDB

Integrating Amazon DynamoDB can be an efficient choice considering the outlined business scenario and requirements. It is a managed NoSQL database service that offers high scalability, low latency, and automatic scaling. Regarding database design, DynamoDB tables can be used to store various data entities such as users, media metadata, and other relevant information. We can efficiently utilize the appropriate primary keys and secondary indexes to facilitate efficient querying and retrieval of data.

Secondly, this database service of AWS has the capability to automatically scale the load of the database seamlessly. The database of the business in the above-mentioned scenario has its demand double every six months, and migrating to DynamoDB is a great choice so that we have a better solution for the increasing load on the database.

We can leverage AWS Lambda and DynamoDB streams to implement an event-driven architecture. Whenever a new media item is uploaded, or any data change occurs, we can use Lambda functions, which can be used with any other AWS services, to perform operations such as converting media or updating metadata.

DynamoDB’s schema-less nature proves its extensibility by allowing for the adding of new attributes or data types easily. And since AWS has a pay-as-you-go pricing model for DynamoDB, it aligns with the company’s requirement of exploring more cost-effective solutions.

SQS

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that offers reliable, scalable, and decoupled communication between various components of a system. By incorporating SQS, the Photo Album application can achieve improved scalability, reduced load on resources, and efficient media processing.

SQS enables seamless scaling as the application's demand grows. It can handle varying levels of incoming messages and automatically scale based on demand. This aligns perfectly with the company's expectation of doubling growth every six months for the next few years. Another advantage is that SQS fits seamlessly into the serverless/event-driven model desired by the company. It allows for asynchronous communication between services, reducing the need for maintaining and managing the state.

Using SQS, the architecture enables different processing services to be hosted on the most suitable platform within the AWS ecosystem. For instance, compute-intensive tasks can be offloaded to EC2 instances, while less resource-intensive tasks can be managed by AWS Lambda.

6. Design factors

a. Performance, scalability

- AWS CloudFront is used to improve the response time in countries outside Australia
- Elastic Transcoder is used to optimize data for web viewing
- S3 is used because it is a highly-scalable cloud service for data storing
- DynamoDB is used because it has scalable metadata storage

b. Reliability

- Both S3 and DynamoDB are used to guarantee the durability and high availability of the system
- CloudFront is used because it can handle failovers, transfer content from the nearest edge location

c. Security

- S3 has a bucket policy and ACL, which can be used to secure objects
- IAM is used to manage permissions and control accesses

7. Cost budget for each component service and total monthly/quarterly/annual budget

Service name	Cost per month
CloudFront	$0.060 - $0.120 per month
S3	$0.021 - $0.023 per month
Lambda	Free
SNS	$2.00 per month
SQS	$0.24 - $0.40 per month
Amazon Elastic Transcoder	$0.03 per month
Amazon Rekognition Image	$62.5 per month
DynamoDB	$5.33 per month
TOTAL	$70.181 - $70.403 per month

8. Reference

[1]

Published by Team Blazeclan, “5 key benefits of Amazon S3 - blazeclan,” *blazeclan*, Dec. 13, 2018. https://www.blazeclan.com/blog/5-key-benefits-of-amazon-s3/

[2]

“How CloudFront delivers content - Amazon CloudFront,” *docs.aws.amazon.com*. https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/HowCloudFrontWorks.html

[3]

“AWS Lambda – Product Features,” *Amazon Web Services, Inc.*, 2019. https://aws.amazon.com/lambda/features/

[4]

S. Kalid, A. Syed, A. Mohammad, and M. N. Halgamuge, “Big-data NoSQL databases: A comparison and analysis of ‘Big-Table’, ‘DynamoDB’, and ‘Cassandra,’” *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, Mar. 2017, doi: https://doi.org/10.1109/icbda.2017.8078782.

[5]

A. Anand, "Managing Infrastructure in Amazon using EC2, CloudWatch, EBS, IAM and CloudFront," Int. J. Eng. Res., vol. 6, no. 03, pp. 373-378, 2017.