COS20019 : Assignment 3

Team Member	Student ID
Arsh Khanna	104100124
Aaryan Bhati	104189110
Filippos Batiastatos	102450186

Introduction:

In order to lessen the need for internal administration work, this report presents a comprehensive architecture design for the new system that takes into account the issues and requirements discovered while using managed cloud services.

Manage cloud services using AWS S3 in the following ways: The recommended architecture will leverage AWS's managed cloud services to lower internal systems administration costs and boost scalability. All of the media, including images and videos, will be stored on the AWS Simple Storage Service (S3). This methodology ensures affordability, high availability, and data longevity while facilitating effortless scaling in response to demand fluctuations.

For the next two to three years, given the exponential growth in application usage in terms of scalability for future growth, we anticipate demand to double every six months. To accommodate this expansion, a highly scalable architecture will be developed. Thanks to AWS services like Elastic Load Balancing, Auto Scaling, and Containerization (using, for example, Docker with Amazon ECS or Amazon EKS), the system will be able to handle an increase in user load without compromising performance.

We will address the performance limitation of the current t2.micro EC2 instances while implementing a well-optimized scaling approach to increase the EC2 Instance Compute Capacity. The application will use AWS Auto Scaling to dynamically adjust the number of EC2 instances based on the current load in order to maintain a computing capacity between 50% and 60%. Peak performance will be maintained and overloading will be prevented by doing this.

In order to implement serverless/event-driven solutions, we will boost operational effectiveness and cost-effectiveness, which will lead to the architecture adopting a serverless/event-driven model. AWS Lambda functions will be used for a range of application tasks, ensuring pay-as-you-go pricing, minimal operational overhead, and automatic scaling. Various media processing tasks will be handled by event-driven triggers, making the system incredibly responsive and adaptable.

We advise moving to a more affordable database solution in order to get around the drawbacks of the existing slow and pricey relational database. Considering the simple table structure, AWS offers appropriate options like Amazon DynamoDB, a fully managed NoSQL database that offers high performance, scalability, and a pay-per-request billing model.

In order to address delayed response times in foreign locations, the architecture will make use of Amazon CloudFront or AWS Global Accelerator. These services greatly reduce latency for users worldwide and enhance response times by enabling content caching and delivery through global edge locations.

Future video media handling is expected for the system, so support for video media can be added, and flexibility will be taken into consideration when designing the suggested architecture. We can automatically convert uploaded videos into multiple formats appropriate for different devices using AWS Elemental MediaConvert or Elastic Transcoder, guaranteeing a flawless user experience.

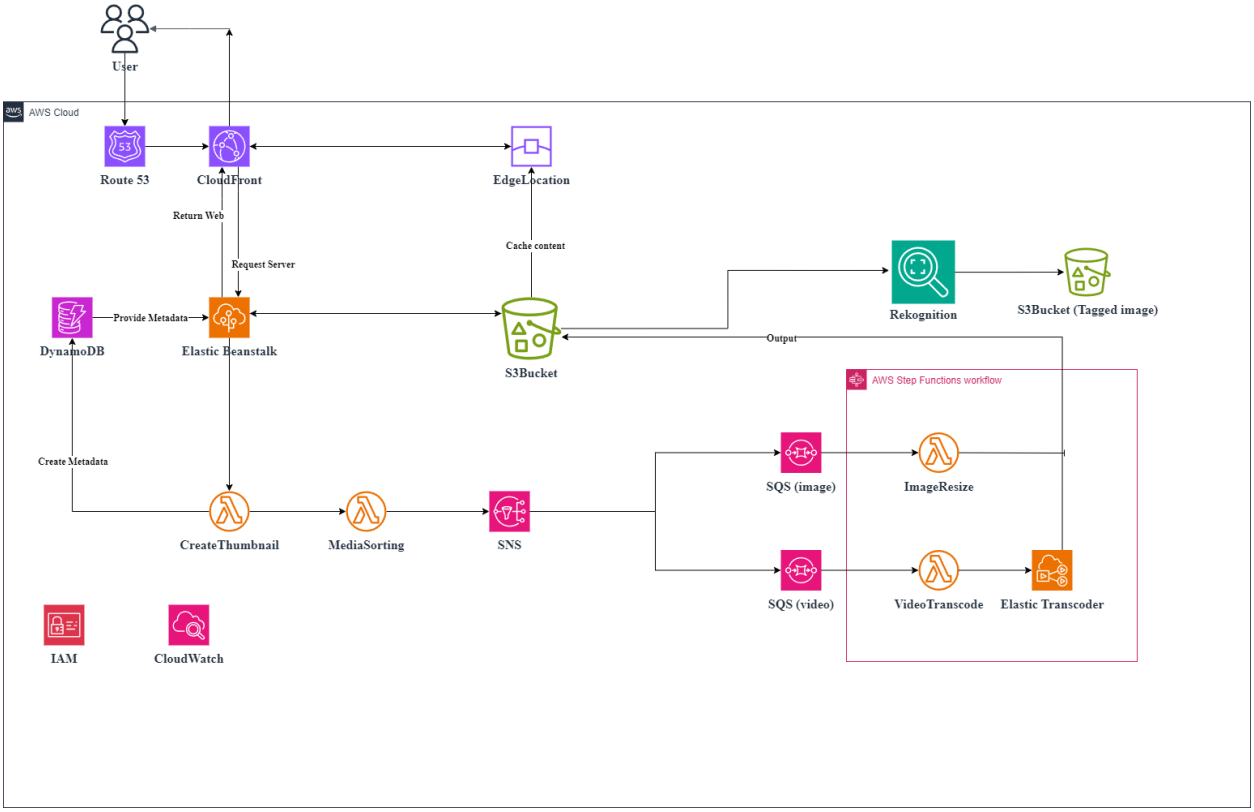
A strong and adaptable design for media reformatting and reprocessing will be incorporated into the architecture. We suggest orchestrating the media processing workflow with AWS Step Functions. The procedure will entail inducing

the automatic generation of substitute media items upon uploading them to S3. The architecture will be built to support AI-based tasks in the future, such as automatically identifying tags in photos. The best platform will be chosen to handle the media processing tasks based on the budget and performance demands.

To guarantee speedy and affordable processing, AWS services like EC2 instances and AWS Lambda will be used. The architecture will additionally introduce a decoupled approach by putting in place a queue-based system. Numerous "worker" nodes will process media transformation jobs concurrently by placing them in a queue. To ensure efficient use of resources and speed up processing, each worker node can specialize in tasks like image reformatting or video transcoding.

Architectural Diagram

Tier Architecture



The three-tier architecture is one of the most well-known architectural design patterns for cloud systems. This pattern served as the basis for the design of photoalbum's new cloud architecture, which consists of three main tiers:

Data tier: houses the database and storage, which house the media files and their corresponding metadata.

Presentation tier: the higher tier that deals with users directly. In this instance, photoalbum refers to the website that both hosts and displays an album of user-uploaded media.

Application tier: the intermediate layer, which handles the main photoalbum operations, such as resizing, uploading media files to storage, and logging their metadata in a database.

Data Tier

File metadata has been stored by the photoalbum web application using AWS Relational Database Service. RDS, however, is not cost-effective given the web application's simplistic table structure. Rather, the database solution will be replaced with AWS DynamoDB, a NoSQL database service. NoSQL databases can scale horizontally, or increase in the number of computing machines, unlike relational databases. This makes them ideal for systems with a basic database structure that are growing quickly. DynamoDB ought to be the ideal option because it is anticipated that web applications will grow by 50% every six months in the upcoming years.

The new architectural design keeps using Amazon S3 for media file storage because it's still a dependable storage option.

Presentation Tier

A lot of overhead and management were needed to host the photoalbum web app on a manually configured network in the past. It is decided to use Elastic Beanstalk, an AWS service that can deploy web applications with less structure management and provisioning, to streamline web deployment while maintaining reliability and availability. Developers can focus more on writing code because the two most important microservices—Auto-Scaling Group and Elastic Load Balancing—can be automated and customized as needed to maintain the system's high availability and scalability.

The business has also discovered that there is a diverse user base for the photoalbum application worldwide. Nevertheless, poor response times have also affected other countries besides Australia. The system's integration with AWS CloudFront is the best way to address this problem. AWS CloudFront is a CDN (Content Delivery Network) service that offers users low-latency delivery of both static and dynamic web content. This can be accomplished by the way that more than 400 edge locations that are installed in different parts of the world appear. Specifically, web content can be cached at a designated edge location so that CloudFront can instantly deliver the cached content to the user upon receiving a request for the web. When the content is sent to the user, CloudFront stores it in the edge location for later use. If the content has not yet arrived at the closest edge location, CloudFront routes the request to the root server.

Application Tier

The web application's ability to upload media files and their metadata is preserved in the new architectural design. It uploads media files to an S3 bucket and then uses an AWS SDK/API call to forward metadata, including the media file's URL, to the CreateThumbnail lambda function, which inserts them into a database. The media-transforming function has undergone the most significant change.
The program must now be able to reprocess videos in addition to images, per the company's requirements. To satisfy this requirement, the architecture needs to add a number of new AWS services.

Decoupled Architecture

The new design has many workstreams, so in order to avoid single points of failure—where problems in one workstream also affect other workstreams—it must be appropriately decoupled. Simple Queue Service (SQS) and Simple Notification Service (SNS) are the best options for decoupling in order to accomplish this.

Based on the fanout architectural design pattern, SNS and SQS are implemented in this architectural design. Assigning incoming events to various downstream processes is the fundamental concept behind this design pattern. More specifically, this is how the system functions:

- The file's S3 URL and a key indicating its media type are added to the message by the MediaSorting lambda function before it is sent to SNS.

- In SNS, a topic is created with two subscriptions, each of which is connected to a single SQS endpoint and has a filtering policy that excludes all messages other than images or videos. A distinct SQS queue is created for each kind of message, and it is there that it awaits polling by the appropriate lambda function.

Extensibility is the main advantage of Amazon Step Functions. Steps Functions can make this process simpler by lowering the amount of code a developer needs to write, assuming the organization wishes to improve the workflows with more features in the future.

Rekognition

It would be ideal to have the ability to identify tags in photos using artificial intelligence in the future. The ideal service for this task should be AWS Recognition. However, as this proposal has not yet been approved, a separate workflow for AWS Rekognition is being tested to avoid any unintended additional storage costs and disruptions to the current image-processing workflow caused by the automation of the stream.

Design Rationale

Scalable architecture

The system can scale up and down in response to traffic and storage needs with the help of implemented services like AWS Lambda, DynamoDB, AWS CloudFront, AWS Route 53, AWS Elastic Beanstalk, AWS SNS, and SQS. Because the architecture's components are successfully decoupled, this design uses SNS, SQS, and Lambda to implement an event-driven solution that promotes scalability and reliability.
Lambda service, Elastic Beanstalk, and Route 53 can scale up to handle heavy traffic loads on this design while maintaining low latency and optimal performance.

Multi-regions performance

Despite being widely used worldwide, the photoalbum application's performance is declining. Because this new design is compatible with CloudFront, dynamic content is cached in edge locations to guarantee low latency while maintaining smooth performance. A better user experience with quicker loading web applications can result from routing traffic to the closest server with Route 53 support.

Infrastructure

The new architecture can reduce the requirement for internal system management by utilizing AWS S3 and newly applied services like AWS DynamoDB, AWS Lambda, and AWS CloudFront. Furthermore, control access (authentication and authorization) through AWS IAM has contributed to improved security. Furthermore, AWS CloudWatch can monitor system performance and improve security. By implementing extra services, this architecture offers the business the convenience of cloud management along with significant enhancements in security, scalability, and reliability. In order to help the company estimate the monthly cost of infrastructure, AWS also offers a service calculator.

Multimedia processing and optimization

The company wants to add features that optimize media files for input and automatically create versions of the application for various devices in response to user demand. Every time a media file is uploaded to the online application, the Lambda function helps to trigger and process the file. This procedure is based on the AWS Steps Function and is separated from SNS and SQS to facilitate the easy integration of any future features.

Database optimization and performance

The storage service has been moved from RDS to DynamoDB in the new design as opposed to the previous one. Because of the strength of NoSQL database services, scalable storage can be provided in response to demand, accommodating growth over the next two to three years. DynamoDB charges based on the amount of storage used by the company and the monthly request volume. A quick-start service that can improve the user experience from offshore locations is offered by DynamoDB.

Design criteria fulfillment

Category	Criteria	Solution
Reliability	▪ Can adapt to changes ▪ Highly available	▪ Replicated, durable data can be automatically

	▪ Backing up data frequently	scaled up or down with Amazon S3. ▪ With multiple copies of your content stored in different locations, AWS CloudFront is highly resilient to failures and can scale up and down based on traffic to reduce latency and improve performance in different regions. ▪ AWS Elastic Beanstalk has the ability to scale up or down in response to actual demand. ▪ AWS DynamoDB is a robust, scalable NoSQL service that replicates data throughout different regions. ▪ Failures can be observed by AWS CloudWatch.
Performance	▪ Quick processing time for uploading media files. ▪ Low latency in fatigue circumstance	▪ Globally distributed edge locations are used by AWS CloudFront to cache dynamic content. ▪ Users can be routed by AWS Route 53 to the fastest and closest server. ▪ If metrics exceed safe bounds, AWS CloudWatch can raise an alarm and initiate automated troubleshooting. ▪ To make media files more accessible to users with varying devices and internet speeds, AWS Elastic Transcoder can convert them into multiple formats and resolutions.
Scalability	▪ Can handle large amounts of traffic. ▪ Scalable to adapt to demand	▪ With the help of caching and load balancing techniques, Amazon CloudFront is highly scalable and can handle millions of requests per second. ▪ AWS Route 53 is capable of processing millions of requests per second and forwards user requests to the closest name server.
Security	▪ Control authorized access and authenticated access	▪ AWS IAM for granting access to the application and verifying identity. ▪ Security threats and malware infections can be

		monitored and detected with the aid of the AWS CloudWatch service. ▪ Data, access control, and thorough logging can all be encrypted using AWS S3.

Alternative Design

An alternative approach to hosting the photoalbum web application instead of using Elastic Beanstalk is to manually set up a network with numerous public and private subnets that cross Availability Zones, NAT gateways for internet access to private subnets, Elastic Load Balancer for load balancing, and Auto Scaling Group for adaptable system expansion using AWS EC2 platform.

This design, which calls for manual configuration from the start, gives developers more control over the system and is appropriate for intricate web applications that demand careful control over every component. The system is notably error prone, though, because the configuring stages are done manually. The effort for developers increases when they have to carefully configure and test the system iteratively before deployment in order to prevent as many errors as possible. Elastic Beanstalk is therefore strongly advised for a less complex web application like photoalbum in order to guarantee performance excellence and reliability while still maintaining a reasonable workload.

Budget

Services	Cost
Dynamo DB	RCU $0.000085 per hr WCU $0.000165 per hr Storage $ 0.25 per GB Appx $1,913
CloudFront	0.085 per GB transferred Appx $ 85 PM
S3	
Elastic Beanstalk	Highly modifiable to suit the requirements of the business but the Appx should be 50-200 per month for us
SQS	For 1TB Appx $90
SNS	Free of charge
Elastic transcoder	Highly modifiable to suit the requirements of the business but the Appx should be $12.5-$100 per month
Rekognition	Appx $200 PM
Route 53	Appx $5 PM
Cloud watch	Appx $ 10 PM
Lambda	Highly modifiable to suit the requirements of the business but the Appx should be $ 10-20 per month

Total	Appx $2326