COS20019 Assignment 3

Serverless/Event-driven Architectural Design Report

Group Number: 179
Tutorial Time: *Thursday, 4:30 pm – 6:30 pm*
Date of Submission: 23 October, 2023

Max Harrison - 104586300
Swinburne University of Technology
Melbourne, Australia
104586300@student.swin.edu.au

Tasks Allocated: Report introduction, services used, and referencing.

Pulkit Pannu - 104093910
Swinburne University of Technology
Melbourne, Australia
104093910@student.swin.edu.au

Tasks Allocated: Design of Architecture and UML Diagrams.

Rohit Raj Saha - 103795228
Swinburne University of Technology
Melbourne, Australia
103795228@student.swin.edu.au

Tasks Allocated: Design rationale and cost estimations

I. Introduction

This report represents a serverless cloud architecture designed for the Photo Album application, addressing several key requirements. These include the transition to managed cloud services, the capability to handle surging demand, the shift to a serverless paradigm, the replacement of the existing slow and costly relational database, enhancement of global response times, and the efficient processing of video and photo uploads.

The report begins by introducing the proposed architecture and its associated use cases, followed by a detailed exploration of the AWS services utilized. The paper then discusses the rationale behind the chosen design and how it meets the business requirements and design criteria. The report concludes by outlining the monthly cost estimation associated with each service.

II. Architectural Diagram

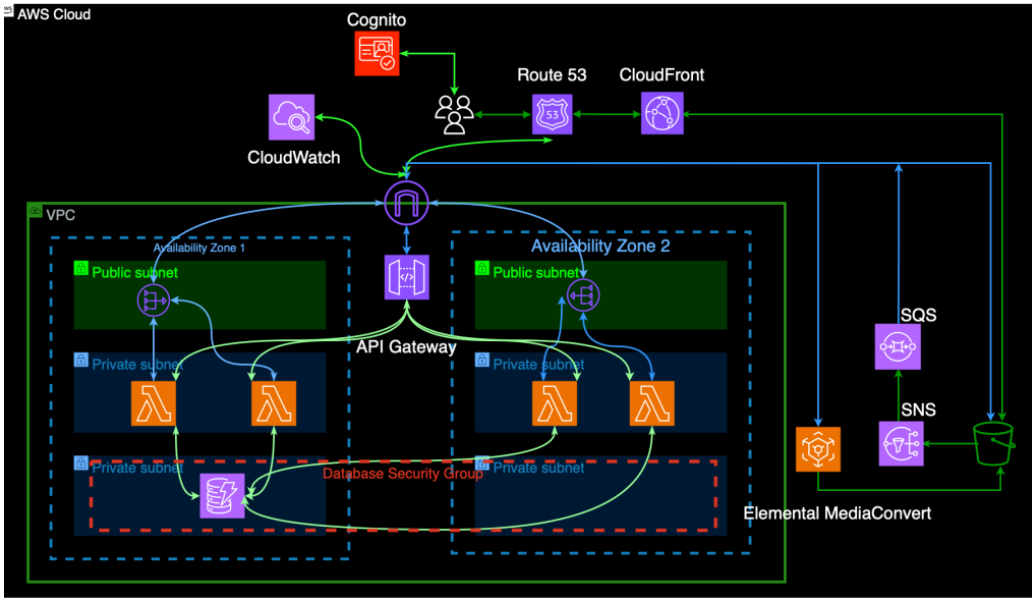

Figure 1: Proposed Infrastructure Diagram

III. UML Collaboration Diagrams

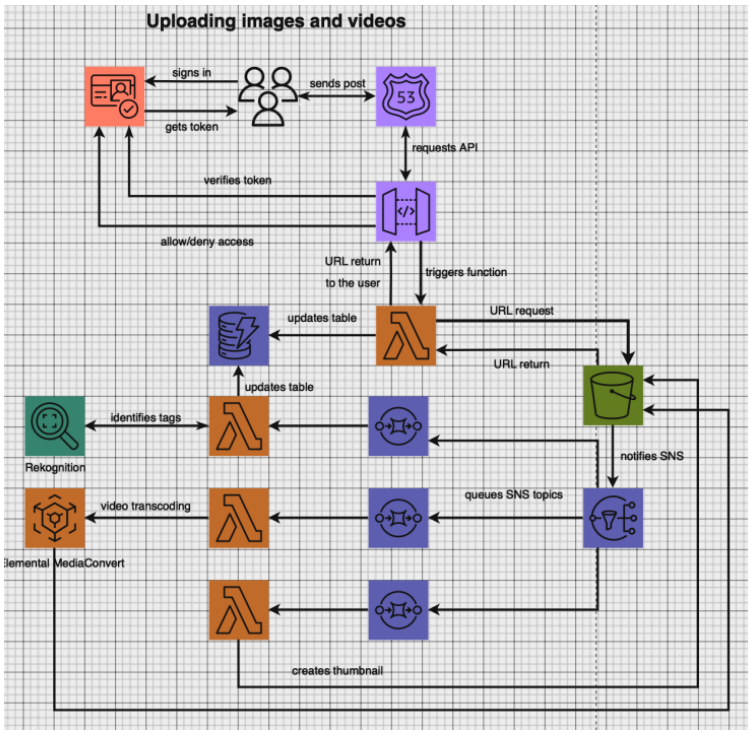

Figure 2: Proposed UML Diagram

IV. AWS Services Used

A. Route 53

Service Description: Amazon Route 53 is a robust and scalable DNS web service. It handles tasks like domain registration, DNS direction, and health assessments. User requests are channeled using route 53 to other AWS resources, including Amazon EC2 instances, S3 storage, and ELB load balancers.

Justification: The photo application uses Route 53 to provide users access via a unique domain name. The widespread network of AWS DNS servers guarantees stability of the application by managing potential network challenges and user routing.

B. CloudFront

Service Description: Amazon CloudFront is a global content delivery system designed to distribute static and dynamic assets such as websites, scripts, images, and videos to end users.

Justification: CloudFront is utilized to boost user response times. This is done by storing content in multiple edge locations close to the user. Additionally, the HTTPS delivery by CloudFront strengthens the application's security. The CloudFront Lambda@Edge feature is used for controlling access to image and video within the S3 storage.

C. API Gateway

Service Description: AWS API Gateway is a platform that simplifies the creation process of APIs. It acts as a consolidated entry point for APIs. It is also equipped with features like rate limiting, cross-origin resource sharing (CORS), and user verification to strengthen API security.

Justification: This gateway seamlessly bridges the gap between the frontend and the Lambda functions in the backend of the application. The API gateway is a very reliable solution as the Photo Album requires high traffic management, scalability, and security.

D. Lambda

Service Description: AWS Lambda offers a serverless platform, allowing developers to execute code without the complexities of configuring or overseeing servers. Lambda functions, which support languages such as Python, Java, and Node.js are activated by particular events, like web interactions or modifications in Amazon S3.

Justification: Lambda is the perfect fit for the company's desire to adopt a serverless approach. It is very affordable with the initial million monthly requests being free. Lambda powers the application's core operations without the constraints of server management and other EC2 trade-offs. Lambda's ability to scale also meets the app's growth demands. Defense is increased using the secure function execution environment.

E. Simple Storage Service (S3)

Service Description: Amazon S3 is a cloud storage platform which provides a suite of management tools. Its object storage design (S3) emphasizes scalability, prompt access, and high availability. Data is categorized into buckets with each item being assigned a distinct key by the user. Individual items have the potential to be up to five terabytes in size.

Justification: S3 was chosen for housing user-uploaded images and videos. This is due to its expansive storage capabilities and its adaptability to store various differing file formats. This aligns with the application's goal to

accommodate multiple media types in the future. Lambda can also be integrated easily with S3 to allow for automated processing of uploaded content which enhances the versatility of this service.

F. DynamoDB

Service Description: Designed for high-efficiency applications of any size, Amazon DynamoDB is a prominent serverless NoSQL database. It boasts features such as built-in security, continuous backups, region-wide auto-replication, in-memory caching, and data transfer tools. With its design, DynamoDB establishes itself as a reliable and adaptable solution for a variety of data requirements and operational demands.

Justification: DynamoDB has been selected to store metadata from the photos and videos uploaded by users. The database's key-value model is in sync with the company's aim of preserving metadata. Not only does DynamoDB provide swift data retrieval, but it also presents a more cost-friendly alternative to traditional relational databases. Additionally, its capability to scale out aligns with the anticipated growth, ensuring efficient cost management as demand surges.

G. Cognito

Service Overview: Amazon Cognito is a scalable solution assisting developers in integrating authentication mechanisms into web and mobile apps. It facilitates the amalgamation of third-party identity providers (like Facebook or Twitter) with proprietary identity systems. Cognito streamlines user management, ensuring secure access to resources.

Justification: The application is envisioned to mandate user authentication prior to service access. Cognito addresses this by introducing an authentication layer to the app. Upon successful authentication, users are issued a JSON Web Token encapsulating their identity, which subsequently verifies backend requests. As a fully managed solution, Cognito can accommodate a vast user base, aligning with the business's preference for managed solutions and the anticipation of growing user numbers.

H. Simple Notification Service (SNS)

Service Description: Amazon SNS is a managed notification service that provides a scalable, flexible, and cost-effective capability to publish messages from an application and deliver them to subscribers or other applications. It supports various subscription types, allowing messages to be sent to a range of endpoints, such as SMS, email, and even Lambda functions and SQS queues.

Justification: When used in conjunction with SQS, this service is extremely effective in optimizing the photo and video processing pipelines of the application. This is done by enabling the parallel processing of multiple media file versions upon their upload to S3. The SNS topic can capture notifications from S3 uploads and subsequently dispatch them to subscribed SQS queues. This design enhances the performance and increases scalability.

I. Simple Queue Service (SQS)

Service Description: Amazon SQS is a managed message queue service which facilitates the decoupling of microservices, distributed systems, and serverless applications. It provides a web service that gives access to a message queue that can be used to store messages while waiting for a computer to process them.

Justification: In collaboration with SNS, SQS plays a pivotal role in streamlining the photo and video processing workflows within the application. Given the varying processing times associated with different file sizes, SQS uses the concept of decoupling to efficiently manage job queues, allowing processing nodes (like Lambda functions or

Elemental MediaConvert) to handle tasks based on their availability. This approach prevents system bottlenecks and ensures no tasks are missed, which enhances the overall system reliability.

J. Elemental MediaConvert

Service Description: AWS Elemental MediaConvert is a comprehensive video processing service that allows users to format and compress video content for delivery to virtually any playback device. It offers a suite of advanced video and audio capabilities, including the ability to create high-quality video transcoding with a wide range of professional-grade features.

Justification: Elemental MediaConvert was selected due to its comprehensive video processing capabilities. It is capable of handling diverse formats to ensure content optimization for various devices. This enhances the overall user experience. The seamless integration with other AWS services and scalability makes it the optimal choice for efficient video processing.

K. Rekognition

Service Description: Amazon Rekognition is a deep learning-powered image and video analysis service. It can identify objects, people, text, scenes, and activities. Moreover, Rekognition provides precise facial analysis and recognition features, making it suitable for tasks like user verification, people counting, and public safety applications.

Justification: Rekognition was integrated for its advanced capabilities in image and video analysis. It helps the application to identify tags from the uploaded photos. It is also very simple to use, eliminating the potential costs of training and deploying a new or custom machine learning model.

L. Virtual Private Cloud (VPC)

Service Description: Amazon VPC allows users to deploy a logically isolated section of the AWS Cloud where resources can be launched. VPC users can further subnet their network, configure route tables, and set up network gateways. This allows for total control over their virtual networking environment.

Justification: The inclusion of VPC in this architecture is crucial, primarily for safeguarding and segregating the resources associated with the photo album. VPC allows specific rules for both incoming and outgoing traffic to be configured. This makes it an essential service for maintaining a cohesive and secure environment.

V. Design Rationale

The proposed architecture is designed to address the challenges and requirements outlined of the Photo Album application. The entire procedure takes place within a Virtual Private Cloud, which ensures a private, separated area of the AWS Cloud, improving security, and allowing for precise network design. The first time a user interacts with the system, they sign in using Cognito, which provides seamless user authentication and authorization. Cognito with its sophisticated security features and its capability to integrate with other AWS services ensures users have unhindered access. Route 53 and CloudFront are included to ensure that worldwide users have fast access and improved response times. CloudFront will cache content closer to global users to ensure low latency when viewing media. Once inside the application, users can upload media, triggering an API Gateway. The gateway handles throttling, traffic control and other aspects of the API lifecycle by managing user requests to upload or fetch media. For the media, once uploaded to the AWS S3 bucket, AWS Lambda is alerted. This serverless function also manages

various operations such as updating the DynamoDB metadata. The NoSQL database service provides rapid and consistent performance. DynamoDB will store the media metadata and scale alongside the application as it grows, maintaining consistent, single-digit millisecond latency.

The media is processed using AWS Rekognition which recognises tags inside photos and Elemental MediaConvert transcodes media and thumbnails are generated for rapid previews. Rekognition is used to tag content, enhancing the media's metadata and searchability and Elemental MediaConvert changes media files from their source into the device playback version.In the background the SQS and SNS duo plays a crucial role. SQS will handle and queue tasks, particularly in high-volume media processing to ensure balance and efficiency. SNS, on the other hand, will keep stakeholders up to date by sending out messages at the end of each processing phase. Finally, CloudWatch is used to ensure the seamless operation and health of the overall architecture. It enables real-time monitoring,allowing for a rapid look at resource performance and health and assuring timely interventions if abnormalities develop.

Alternative Solutions

- ***Virtual Machines and Containers:*** Virtual machines provide a full virtual operating system with its own cpu, memory, and storage. This makes them quite resource intensive, which can be wasteful depending on the applications needs. Due to the company's desire for a serverless approach, AWS Lambda has been utilized instead.

- ***Elastic Load Balancing:*** Elastic Load Balancing automatically distributes incoming application traffic across multiple chosen targets. This ensures the application runs smoothly and prevents a single instance from becoming a bottleneck. ELB was not used in this application because of the serverless approach adopted. Instead AWS Lambda and the API Gateway handle necessary load balancing and routing management. This was determined to be a more cost effective solution.

- ***Elastic Transcoder:*** Elastic Transcoder is a cloud based service designed to convert media files into formats suitable for playback on various devices. Features such as watermarking and thumbnail creation are also included. It also integrates seamlessly with other AWS services allowing for a smooth workflow.

- ***ElastiCache:*** Caching can significantly improve application performance by storing frequently used data in memory temporarily. After reviewing services available such as ElastiCache, we decided that CloudFront would be the best delivery method for users. It has caching, faster delivery, and reduced latency.

- ***2 Tier Architecture:*** A 2 tier architecture combines the presentation and logic layers which can make the design simpler but less scalable. Due to the required focus on high scalability and availability we adopted a 3 tier architecture. This approach ensures components can scale and evolve independent of each other using a presentation, logic, and data layer.

Design Criteria

A. *Performance / Scalability*

   Our architecture prioritizes performance. CloudFront ensures that users experience rapid, low-latency access to media content. Lambda ensures seamless operations, eliminating the need for traditional server management. The combination of SQS and SNS enhances scalability, allowing the system to handle growing demands without overloading. DynamoDB further bolsters this scalability, adeptly managing the application's data as user interactions increase.

B. Reliability

Reliability is achieved through various services. Route 53 ensures easy user access, whereas VPC ensures stability by isolating the cloud environment. CloudWatch continuously monitors system health and detects potential faults in real time. Additionally, Elemental MediaConvert ensures consistent media playback across several platforms guaranteeing the end user's satisfaction.

C. Security

The VPC offers a safeguarded environment for our services to function within, and Amazon Cognito, which controls user authentication, secures user credentials via a token based system. The API Gateway controls and secures all data coming in and going out further enhancing Lambda functions security. Rekognition also has security measures which can flag improper or suspicious details.

VI. Cost Estimation

The projected monthly expenses are $4067. Considering the merging of services and anticipated high user engagement and data handling, this investment guarantees superior user interactions. It also allows for potential future expansion.

AWS Service	**Estimated Monthly Cost (USD)**
Route 53	$30.00
CloudFront	$330.00
API Gateway	$800.00
Lambda	$1100.00
Simple Storage Service (S3)	$52.50
DynamoDB	$470.00
Cognito	$275.00
Simple Notification Service (SNS)	$0.00
Simple Queue Service (SQS)	$0.00
Elemental MediaConvert	$740.00
Rekognition	$120
CloudWatch	$ 150
Total	$4067

Conclusion

Our primary objectives were to transition to managed cloud services, ensure scalability for rising demands, enhance response times, and optimize media processing. The initial decision to adopt a serverless approach allowed us to focus on core application features, resulting in both cost and operational benefits. The seamless integration of services like Lambda, S3, DynamoDB, and Rekognition ensured that our application was not only efficient but also future-proof, and capable of adapting to future user and media processing requirements.

References

Drawing from the content of our weekly labs and the knowledge gained from prior assignments, we designed the architecture for this iteration of the photo application. The official AWS documentation, encompassing topics like S3, DynamoDB, API Gateways, and Lambda, deepened our grasp of these pivotal services. Further, strategic guidance from the Machine Learning and Containers Decision Guides played a crucial role in our service selection, ensuring we met the performance and scalability needs required.

[1] https://docs.aws.amazon.com/s3/?icmpid=docs_homepage_featuredsvcs
[2] https://docs.aws.amazon.com/dynamodb/?icmpid=docs_homepage_featuredsvcs
[3] https://docs.aws.amazon.com/vpc/?icmpid=docs_homepage_featuredsvcs
[4] https://docs.aws.amazon.com/lambda/?icmpid=docs_homepage_featuredsvcs
[5] https://aws.amazon.com/getting-started/decision-guides/machine-learning-on-aws-how-to-choose/
[6] https://aws.amazon.com/getting-started/decision-guides/containers-on-aws-how-to-choose/
[7] https://aws.amazon.com/rekognition/resources/
[8] https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/s3-example-photo-album.html
[9] https://aws.amazon.com/sns/
[10] https://aws.amazon.com/architecture/reference-architecture-diagrams/?solutions-all.sort-by=item.additionalFields.sortDate&solutions-all.sort-order=desc&whitepapers-main.sort-by=item.additionalFields.sortDate&whitepapers-main.sort-order=desc&awsf.whitepapers-tech-category=*all&awsf.whitepapers-industries=*all
[11] https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-rest-api.html
[12] https://aws.amazon.com/elasticbeanstalk/
[13] https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-basic-concept.html
[14] https://aws.amazon.com/route53/