Cloud Computing Architecture – COS20019

Assignment 3 – Report
Serverless/ Event-driven Architecture Design Report

Team members:

Nguyen Gia Bao – 104180317

Nguyen Duc Huy – 104177335

Pham Mai Hanh – 104198828

Tran Duy Minh – 104170994

Lecturer:

Tung Do

Class time:

8:00 am-12:00 pm every Saturday

Date of submission:

26/11/2023

I. Introduction

Owing to the ever-growing technological advancement, cloud computing has found multifaceted applications across digital projects. Among these, services that adopt a serverless architecture, such as AWS stand out, being a cost-effective tool catering to numerous requirements regarding innovation and efficiency. The aforementioned service is predicated on streamlining the development and management of websites on-premises and on the cloud. Consequently, it provides users with approaches that not only enhance the web-making experience but also offer the flexibility to adapt to changes, exemplified by its event-driven solution. This method allows direct response to specific events, which strengthens the system's scalability and responsiveness.

With AWS services, we have planned to build an environment for users to upload and view different media such as artworks, videos, audio files, or just unstructured data. With our services, users can create an environment where they can upload their media of choice, the system then processes the media (e.g., image resizing, video frame processing, etc.) and displays it on the website.

II. Architecture Overview

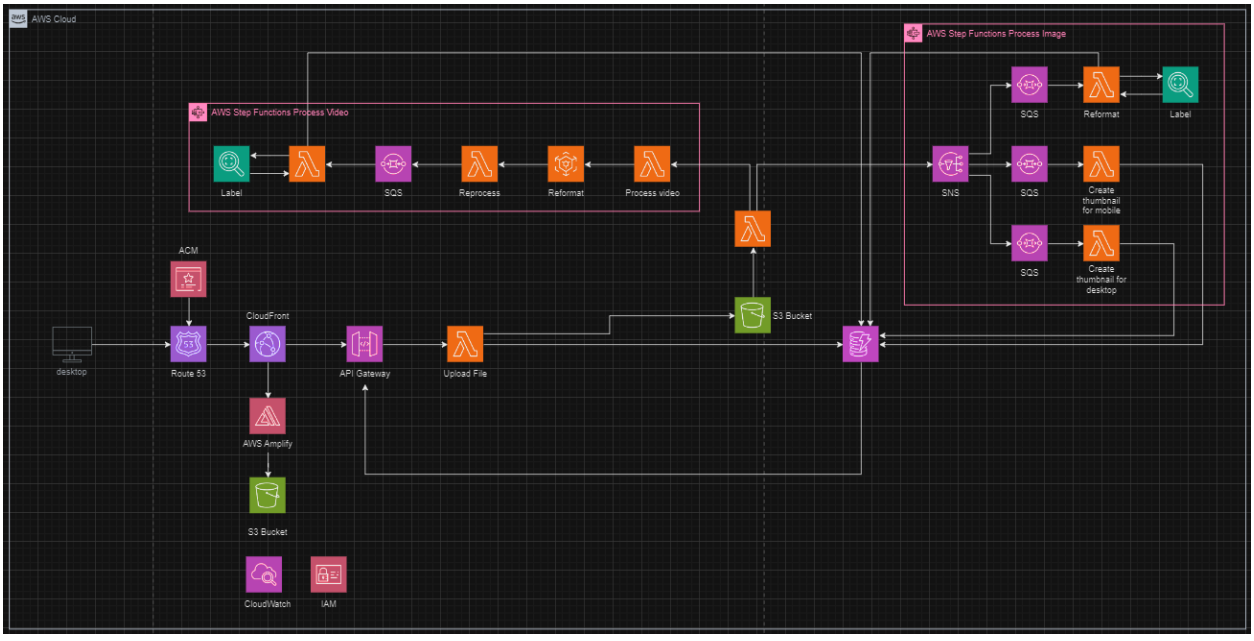

Figure 1: Architecture Overview

The photo album web application (Figure 1) above is a scalable solution that our has come up with that allow user to upload all sort of media formats. The process included reformatting and automatically identify tags in photos using AI. To achieve this our website architecture are required to have the following key component:

- AWS CloudFront: Hosting the front end of the website.

- AWS Amplify: Builds and deploys the web front.
- API Gateway: Gateway for the application to access data, functionality of our website backend.
- S3: Store images and videos that uploaded by users
- AWS SQS: To make the web easy to decouple, scale and handling time-consuming tasks.
- DynamoDB: A fully managed NoSQL database service that can store and retrieve data. It also serves as a primary data storing place for the website.
- AWS SNS: Sent massage to produce thumbnails for nether desktop or mobile
- AWS Rekognition: Analyze uploaded images or videos.

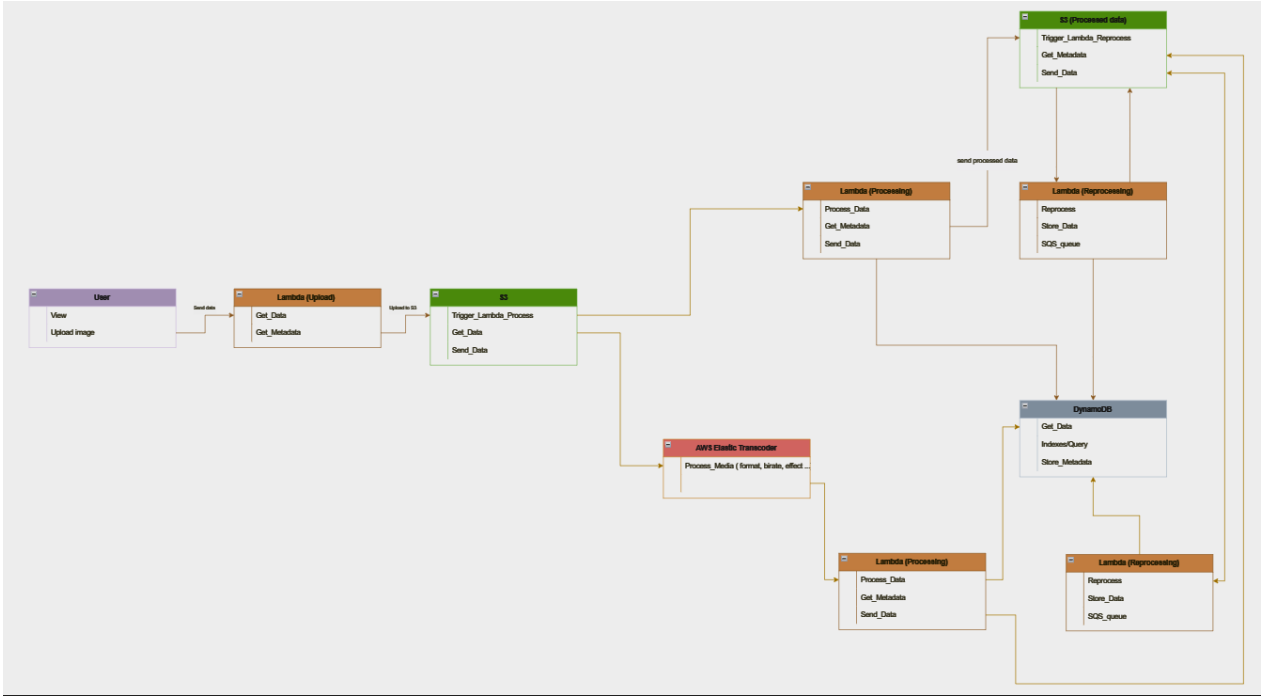

Figure 2: General data flow diagram

In the general workflow above (Figure2), in the website user can upload a file which may be pictures, arts, videos, and more. The data then go through Route 53, which act as the entry point for the user, manage their request to access our services. It then route the traffic to CloudFront, together make it faster for user from any location to access web content, reduce latency and optimize upload process by using the content and the nearest location. The data then capture by a Lambda function and upload it on S3 bucket, provide a serverless architecture design for upload and store files. Depend on the data that being uploaded, S3 will trigger AWS Elastic Transcoder service which will configure the raw data (e.g., resize image, video effect process, …), send it to Lambda to classify media type. The data process for uploading images and videos is much different from each other. For uploading images, the size is much smaller, and the image format include JPG, PNG, GIF, …. In this process we uses Lambda to handle tasks such as resizing, cropping, rendering or changing color scheme. Regarding uploading videos, formats are much more complex with some

like, MP4, MOV, … Videos include hundreds if not thousands of frames and sound along with resolution and bitrate. Therefore, we use AWS Elemental MediaConvert to handle more heavy weight tasks such as adding effects, frames and overlay. Following that, the user's media upload, the system will process the content based on its media type and store the resulting metadata/processing details in DynamoDB for showcasing on the website.

III. Architecture Justification

Business Scenario - Justification:

1. Where possible the company would like to use managed cloud services to minimize the need for in-house systems administration. Photo and other media will be stored in AWS S3.

2. The company is not sure how demand for its application will grow in the future but over recently it has been doubling every 6 months. It expects this trend will continue for the next 2 or 3 years at least and it wants the architecture to be able to cope with this growth.

3. The company would like to adopt a serverless/event-driven solution.

4. The company wants to explore more economical database options.

5. A solution to enhance the web application's global response time.

6. The company desires the capability for users to upload various media formats to the web application.

7. The data processing architecture should be flexible and possess the ability to automatically detect tags in photos using AI.

8. The web application should maintain optimal performance without overload and effective decoupling.

A. Performance and scalability

To enhance the website performance and scalability our team focus on the following AWS services: Lambda, S3, SNS, and SQS.

With Lambda, functions are executed solely in response to specific events, such as file uploads or user interactions. This eliminates the need for constant server provisioning, enhancing cost-effectiveness. Lambda also automatically scales resources based on workload, ensuring optimal performance. This adaptive scaling mechanism eliminates manual intervention, optimizing resource utilization and minimizing costs [1].

S3 excels in providing high-performance storage, ensuring efficient storage and retrieval of photo and media files. Its capabilities accommodate large volumes of data and high request rates, aligning seamlessly with the company's growth trajectory. S3's unparalleled scalability allows the

company to effortlessly adjust storage capacity based on demand. As user traffic and data volume increase, S3 can dynamically provision additional resources, eliminating the need for manual intervention and ensuring optimal performance.

Amazon SNS delivering notifications in real-time highlights its strong performance. This feature is crucial for scenarios where prompt awareness of events is essential for effective business operations. And with flexible notification delivery, SNS support various notification delivery protocols underscores its adaptability, contributing to a high level of performance and scalability.

Amazon SQS is optimized for horizontal scalability, allowing a virtually unlimited number of computers to read and write messages concurrently. While a single client can send or receive messages at a rate of 5 to 50 messages per second, higher performance can be achieved by requesting multiple messages in a single call (up to 10). The system's elasticity enables it to scale dynamically to handle varying workloads without necessitating application changes.

B. Database options

In order to improve website efficiency and it cost, we use Amazon DynamoDB as our database. DynamoDB, an exclusive fully managed NoSQL database provided by Amazon [2], offers superior performance compared to relational databases. Switching to Amazon DynamoDB not only enhances performance but also capitalizes on its non-relational data storage capabilities. Moreover, DynamoDB is also excels in scalability and availability [3].

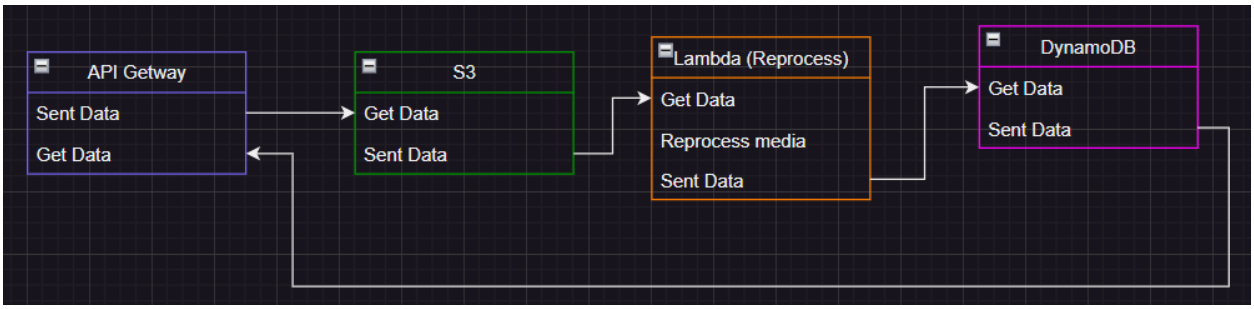

Figure 3: Data structure of the website

Amazone offered two pricing structure for DynamoDB, which are: “On-demand capacity” and “Provisioned capacity”. As the company remains uncertain about the anticipated application growth in the future, our suggestion is to use the "On-demand capacity" structure. With this option, it only charges based on the request units utilized. Therefore, it offers great capabilities for autoscaling and the implementation of a serverless architecture. Whereas, “Provisioned capacity” structure is billed by hourly per the use of capacity used [4].

C. Global response time improvement

The website global response time can be improved by using Amazon CloudFront and Amazon Route 53. CloudFront Amazon Web Service is a content delivery network (CDN), plays an important role in optimizing global response times by leveraging networks of edge locations worldwide. Through its caching mechanisms, CloudFront stores content at these edge locations, significantly reducing latency by delivering data closer to end-users globally. Its distributed architecture efficiently handles requests by offloading traffic to these edge locations, ensuring swift and reliable content delivery even during periods of high demand [5]. Ultimately, CloudFront's combination of caching, distribution, security, and integration capabilities facilitates an enhanced and efficient user experience across different regions worldwide.

Amazon Route 53 is a DNS service where the user can use it to distribute loads across regions, it has redundant backed by 100% SLA, which it is fast and utilizes worldwide anycast. Utilizing geolocation routing, it directs users to the nearest server or CDN based on their geographic location, diminishing latency and enhancing response times [6]. Therefore, the user will have better response time regardless of where the user are in the world when including Amazon CloudFront and Route 53 in the architecture.

D. Handling video and photos

When a user upload images or video on the website, the media then will be stored in S3, then it will be classified by lambda to sort out photos and videos. Then, the media will get reprocess through Amazon Lambda and SQS, which allows the user to send, store and receive messages between components at any volume, without losing message [7]. Implementing Amazon SQS in the architecture also result in preventing the application to not overloaded and effectively decoupled (like we mentioned in part A)

For automatically identify tags in photos using AI, our team decided to use Amazon Rekognition service. By adding Amazon Rekognition to the architecture, it can detect objects, scenes, faces and text in image, … [8]. Therefore, in the future the company can automatically identify tags in photos in the website.

In order to optimizes cost and performance of the services within the ecosystem AWS, we design this architecture to have the most suitable service for the right task. For instance, we use lambda to handle processing image and for heavier task like video processing we leave those to AWS Elemental MediaConvert to handle the job. By doing this, it will improve the overall performance of the architecture and also optimizing the cost.

IV. Alternative Consideration

1. Serverless Computing over Virtual Machines and Containers:

Serverless computing has emerged as the preferred choice over virtual machines and containers due to its inherent advantages in scalability and cost-effectiveness. It eliminates the need to provision or manage servers, offloading this responsibility to the cloud provider. This results in simplified operations and reduced infrastructure costs. While virtual machines and containers offer flexibility and control, they introduce additional management overhead and can lead to higher costs, especially with fluctuating workloads. In contrast, serverless computing scales effortlessly based on demand, ensuring efficient resource utilization and cost optimization.

Performance:

- Low Latency: Serverless computing must ensure low latency to minimize delays in video access or processing, ensuring a smooth and responsive user experience [19].
- High Throughput: Serverless computing should efficiently handle large volumes of video processing tasks and data transfers, supporting a growing user base and increasing video traffic without compromising performance.

Scalability:

- Automatic Scaling: Serverless computing should automatically scale based on demand, optimizing resource utilization and eliminating the need for manual provisioning and server management.
- Elasticity: Serverless computing should handle fluctuating workloads and traffic spikes without impacting performance, ensuring system stability and responsiveness.

2. NoSQL Database (Amazon DynamoDB) over SQL Database:

For storing video metadata, a NoSQL database, particularly Amazon DynamoDB, has proven to be a more suitable choice compared to a traditional SQL database. NoSQL databases excel in handling large volumes of unstructured data, making them well-suited for managing video metadata, including attributes like video titles, descriptions, and tags. SQL databases, while structured and suitable for relational data, may struggle with the flexibility and scalability required for video metadata management. NoSQL databases provide a more efficient and scalable solution for this specific use case.

Flexibility:

- Flexible Data Modeling: Amazon DynamoDB should provide flexible data modeling for unstructured and semi-structured video metadata, allowing efficient storage and retrieval of attributes.
- Schema-less Design: Amazon DynamoDB should follow a schema-less design, accommodating dynamic data storage and adapting to changing video metadata requirements for evolving video formats and content types.

Durability:

- High Data Availability: Amazon DynamoDB should ensure high data availability, protecting video metadata from loss or corruption and ensuring users always have access to needed information [20].
- Data Replication: Amazon DynamoDB should replicate data across multiple nodes to prevent data loss in hardware failures, ensuring the integrity and accessibility of video metadata [20].

3. Amazon ElastiCache for Caching:

Amazon ElastiCache is the preferred caching solution due to its ability to improve performance and reduce the load on the NoSQL database. By caching frequently accessed video metadata, ElastiCache minimizes the need to repeatedly query the database, significantly improving response times and reducing overall system latency. While other caching options, such as in-memory caching, can be effective, ElastiCache offers the benefit of managed services, reducing the burden of server administration and ensuring optimal performance.

Scalability:

- Horizontal scalability: Amazon ElastiCache should be able to scale horizontally by adding additional cache nodes to accommodate increasing caching needs and handle growing video metadata storage requirements. This ensures that the system can handle increasing data volumes without performance degradation [21].
- Elastic scaling: Amazon ElastiCache should support elastic scaling, allowing it to automatically adjust its cache capacity based on demand. This ensures that the system can handle fluctuating workloads and traffic spikes without affecting performance or incurring unnecessary costs [21].

Reliability:

- High availability: Amazon ElastiCache should be highly available to prevent caching disruptions and ensure continuous access to cached video metadata. This is crucial for maintaining system stability and preventing performance bottlenecks.
- Fault tolerance: Amazon ElastiCache should be fault-tolerant, able to withstand hardware failures or node outages without affecting the overall system's performance or availability. This ensures that cached video metadata remains accessible even in the event of partial system failures.

4. Push-Based Messaging (Amazon Simple Notification Service) for Decoupling:

Push-based messaging, particularly Amazon Simple Notification Service (SNS), proves to be a valuable tool for decoupling system components. Using push-based messaging allows components to communicate asynchronously without the need for direct dependencies or real-time communication. This decoupling promotes flexibility and scalability, enabling components to evolve independently without affecting the overall system. SNS simplifies the implementation of

push-based messaging, providing a reliable and scalable mechanism for inter-component communication [22].

5. 3-Tier Architecture over 2-Tier Architecture:

The 3-tier architecture, consisting of presentation, application, and data tiers, has emerged as the preferred choice due to its advantages in scalability, flexibility, and security.

- Scalability: The 3-tier architecture enables independent scaling of each tier, facilitating efficient resource allocation and optimal performance handling.

- Flexibility: The separation of concerns in the 3-tier architecture promotes flexibility, allowing for easier modifications and updates to individual tiers without affecting the entire system.

- Security: The 3-tier architecture enhances security by isolating the data tier from the presentation tier, reducing the attack surface and minimizing potential security vulnerabilities. While the 2-tier architecture is simpler in structure, it may become less manageable and scalable as the system grows in complexity. The 3-tier architecture provides a more robust and maintainable solution for the video processing and delivery system.

Additional Considerations:

We considered factors like service maturity, ease of use, and AWS support when selecting services. Chosen services have a mature track record, are user-friendly, and are well-supported by AWS.

V. Architecture Evaluation

1. Operational excellence pillar

- AWS Lambda:

AWS Lambda is a serverless and event-driven solution, allowing users to execute code without infrastructure provisioning and management. This eliminates operational overhead and enables response to high demand in double-digit milliseconds facilitated by Provisioned Concurrency

- SNS:

SNS Asynchronous Communication can integrate and decouple distributed applications, utilizing push-based and many-to-many messaging. SNS dynamic auto-scaling adeptly handles high-throughput workloads such as message traffic, ensuring consistent performance.

- SQS:

SQS eliminates overhead and decouples system components, enabling asynchronous communication, and alleviating the need to manage software or maintain infrastructure. SQS scales elastically and efficiently accommodating variable workloads and spikes in message traffic. SQS works as a buffer between message producers and consumers, preventing bottlenecks and optimizing performance

- CloudWatch and CloudFront:

CloudWatch provides end-to-end observability of various AWS services, enabling data visualization and analysis. CloudFront effectively minimizes latency by delivering data across over 600 globally dispersed Points of Presence. This is achieved through automated network mapping and intelligent routing. The dynamic scaling feature ensures consistent performance, even in the face of traffic spikes or increased demand.

- DynamoDB:

DynamoDB is a flexible and fully managed NoSQL database that offers adaptive capacity and auto-scaling, automatically accommodating fluctuations in traffic patterns and workload demands. DynamoDB's implementation of best practices such as batch operations, item size reduction, and query optimization.

2. Security pillar

- IAM:

IAM's logging and monitoring features play a vital role in tracking user activities and API calls, offering administrators valuable insights into their AWS environment for security analysis, compliance, and troubleshooting. IAM provides per-account or scaled access across AWS services and accounts, adhering to the principle of least privilege. This approach minimizes risks associated with accidental misconfigurations, deliberate malicious activities, or unauthorized access. IAM enhances security with multi-factor authentication and the provision of temporary credentials for workloads accessing AWS resources, mitigating risks associated with prolonged access keys.

- Route 53:

Route 53 directs end users to resources with globally dispersed Domain Name System servers and automatic scaling. Route 53 assigns and accesses custom domain names in Amazon Virtual Private Cloud, utilizing internal AWS resources and servers without exposing DNS data to the public Internet.

- CloudFront:

CloudFront provides number of security features such as SSL/TLS encryption for secure data transmission, origin protection to restrict access, IP allow/block lists for access control based on

IP addresses, field-level encryption to safeguard sensitive data in HTTP cookies and query strings, and automatic DDoS protection at both network and application layers [17].

- AWS Certificate Manager

AWS Certificate Manager (ACM) is a service that lets user easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) [23], which significantly enhances the web application security .

3. Reliability pillar

For ensuring a reliable architecture, our primary AWS services to achieve this are Lambda, DynamoDB, Elemental MediaConvert, SQS, and CloudWatch.

- Lambda operates within AWS Regions and Availability Zones, offering multiple physically separate and isolated Availability Zones that are interconnected through low-latency, high-throughput, and highly redundant networking [9].
- DynamoDB is fully managed service that replicates data across multiple AZs, which ensured the data durability and availability. Amazon DynamoDB also offers two features to help support data resiliency and backup, which are “On-demand backup and restore” and “Point-in-time recovery” [10].
- Elemental MediaConvert enables parallel processing of video encoding jobs, and AWS manages automatic recovery from hardware failures, which ensuring uninterrupted operation.
- SQS ensures reliable message queuing by replicating messages across multiple AZs. This redundancy ensures message processing continuity even if a message is lost or a queue becomes unavailable in one AZ, elevating the overall reliability of media processing.
- CloudWatch provides actionable insights to optimize application performance, effectively manage resource utilization, and gain a comprehensive understanding of system-wide operational health. This service allows monitoring of the entire stack, including applications, infrastructure, network, and services, while also offering alerting capabilities for containerized applications [11].

4. Performance efficiency pillar

To ensure the performance of the architecture, our key AWS components for achieving this goal includes Lambda, Elemental MediaConvert, SQS, and CloudWatch.

- The Lambda service can handle automatic scaling, minimizing latency, and maximizing throughput. Through dynamic auto-scaling in response to demand, Lambda ensures optimal resource utilization, enhancing overall performance [12].

- As users access Amazon SQS via an HTTP request-response protocol, the latency of the request (the time between initiating the request and receiving a response) imposes constraints on the throughput achievable from a single thread using a single connection [13]. By doing that, SQS ensures that media processing task performed without overloading the application and enhancing the application performance.
- Elemental MediaConvert leverages a distributed architecture to achieve accelerated video encoding by enabling parallel processing of video encoding jobs. This results in a significant reduction in media processing time, ensuring rapid delivery of media content to users and enhancing the overall performance of the application [14].
- The reason that we selected DynamoDB as the database it is because DynamoDB a fully managed NoSQL database designed for high-performance handling of large-scale and concurrent workloads. With the capability to deliver response times in the single-digit millisecond range, it facilitates swift retrieval and storage of media metadata, thereby enhancing the efficiency of data access operations [15].
- Amazon CloudWatch plays a crucial role in real-time monitoring and visibility of the application's performance metrics and resource utilization.

5. Cost optimization pillar

To optimize the cost of the architecture, our key AWS components for achieving this goal are:

- AWS Lambda stands out as a cost-effective solution due to its on-demand execution and autoscaling features. The elimination of continuous server provisioning and the efficient scaling based on workload contribute to significant cost savings for the business [16].
- With AWS SNS the business can apply the “pay-as-you-go model”, coupled with features like message filtering, contributes to cost-effectiveness by allowing organizations to optimize costs based on actual usage [18].
- DynamoDB offers “On-demand Capacity” pricing, which only charged for that the business demanded [4]. This could help the business handling the growth of the website and optimize the of the database.
- Elemental MediaConvert provide “pay-as-you-go” pricing and faster video encoding, this helps reduce the overall cost of the architecture.
- CloudWatch allows the business to monitor their used data and their cost, this feature could help the business adjust their resources usage and implement cost-saving measures.

VI. Cost

In this part, we estimate the web app has 10,000 daily active users, upload average about 6 images and 2 video upload per day.

AWS Services	Monthly cost	Yearly cost	Cost Summary

AWS Lambda	213.97 USD	2,600 USD	Architecture (x86), Number of requests (1000000000/ per month), Duration of each request (10 ms), Amount of ephemeral storage allocated (512 MB), Amount of memory allocated (128 MB), Invoke Mode (Buffered)
AWS SQS	26.1 USD	313.2 USD	Standard queue requests (30 million per month), FIFO queue requests (30 million per month)
AWS S3	242.69 USD	2912.28 USD	S3 Standard storage (10 TB per month), Data returned by S3 (10 TB per month)
AWS Elemental MediaConvert	525 USD	6300 USD	Output usage (70000 minutes per month), Tier (Basic), Video Codec (AVC), Video Quality (Single Pass), Video Resolution (SD), Video Framerate (<= 30 fps)
AWS CloudWatch	23.67 USD	284.04 USD	Number of Metrics (10), Number of Lambda functions (8), Number of requests per function (8 per minutes)
AWS CloudFront	892.08 USD	10704.96 USD	Data transfer out to internet (5 TB per month), Data transfer out to origin (5 TB per month), Number of requests (HTTPS) (1000000 per month), CloudFront Edge Locations (Japan)
AWS API Gateway	450.00 USD	5400 USD	HTTP API requests units (millions), Requests (100 units per month), Average size of each request (34 KB), REST API request units (millions), Requests (100 units per month),
AWS OpenSearch Service	4,997.58 USD	59970.96 USD	Number of instances (1), Storage for each Amazon OpenSearch Service instance (General Purpose SSD (gp3)), UltraWarm storage cost (0), Nodes (1), Instance type (r5.2xlarge.search), Utilization (On-Demand only) (100 %Utilized/Month), Instance Node Type (Memory optimized), Storage Type (EBS Only), Pricing strategy (OnDemand), Nodes (1), Instance type (r5.2xlarge.search), Utilization (On-Demand only) (100 %Utilized/Month), Instance Node Type (Memory optimized), Storage Type (EBS Only), Pricing strategy (OnDemand), Number of nodes (2), Instance type (ultrawarm1.large.search), Utilization (On-

			Demand only) (100 %Utilized/Month), Pricing strategy (OnDemand)
AWS Route 53	115.50 USD	1386 USD	Hosted Zones (1), Additional Records in Hosted Zones (10000), Traffic flow (2)
AWS Amplify	117.30 USD	1407.6 USD	Duration of each request (in ms) (500), Number of build minutes (10000 per month), Data stored per month (100 GB), Data served per month (100 GB)
AWS Rekognition	600 USD	7200 USD	Number of images processed with labels API calls per month (300000 per month), Number of images processed with content moderation API calls per month (300000 per month)
AWS DynamoDB	497.23 USD	5966.76 UDS	Table class (Standard), Average item size (all attributes) (10 KB), Write reserved capacity term (1 year), Read reserved capacity term (1 year), Data storage size (1 TB)
Total Cost	10,844.91 USD	105,973.44 USD	

VII. Conclusion

In conclusion, in this report we managed to build a cloud service like AWS S3 for efficient media storage and transitioning to a serverless/event-driven solution to accommodate the application's rapid growth. Optimizing compute capacity, exploring cost-effective database alternatives, and improving global response times. Proposed an automated and extensible media processing system, utilizing various AWS services, ensures scalability and innovation. Lastly, we calculate the average cost of the web architecture.

References

[1] Amazon (n.d.). *Serverless Computing - AWS Lambda Features - Amazon Web Services*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/lambda/features/#:~:text=AWS%20Lambda%20is%20a%20serverless.

[2] Amazon Web Services. “What Is Amazon DynamoDB? - Amazon DynamoDB.” *Docs.aws.amazon.com*, docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html.

[3] robvet (2022). *Relational vs. NoSQL data*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data.

[4] Onojakpor, O. (2023). *Simple DynamoDB Pricing and Cost Optimization Guide | CloudForecast*. [online] www.cloudforecast.io. Available at: https://www.cloudforecast.io/blog/dynamodb-pricing/ [Accessed 19 Nov. 2023].

[5] Amazon Web Services, Inc. (n.d.). *FAQs | What is Amazon Cloudfront CDN?* [online] Available at: https://aws.amazon.com/cloudfront/faqs/#:~:text=Amazon%20CloudFront%20is%20a%20web.

[6] Swinburne (n.d.). *Cloud Computing Architecture Lecture 10 Routing across Regions -Route53 Architectural Design Patterns Sample Architectures*.

[7] Amazon (n.d.). *Message Queuing Service - Amazon Simple Queue Service - AWS*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/sqs/#:~:text=Amazon%20Simple%20Queue%20Service%20(Amazon [Accessed 20 Nov. 2023].

[8] Amazon (n.d.). *Amazon Rekognition Image*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/rekognition/image-features/.

[9] Amazon (n.d.). *Resilience in AWS Lambda - AWS Lambda*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/lambda/latest/dg/security-resilience.html.

[10] Amazon (n.d.). *Resilience and disaster recovery in Amazon DynamoDB - Amazon DynamoDB*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/disaster-recovery-resiliency.html.

[11] Amazon (n.d.). *Amazon CloudWatch Product Features - Amazon Web Services (AWS)*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/cloudwatch/features/#:~:text=CloudWatch%20enables%20you%20to%20monitor.

[12] Amazon (n.d.). *Performance optimization - AWS Lambda*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/lambda/latest/operatorguide/perf-optimize.html.

[13] Amazon (n.d.). *Increasing throughput using horizontal scaling and action batching - Amazon Simple Queue Service*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-throughput-horizontal-scaling-and-batching.html [Accessed 22 Nov. 2023].

[14] Amazon (n.d.). *What is AWS Elemental MediaConvert? - MediaConvert*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html.

[15] Amazon (n.d.). *Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/dynamodb/#:~:text=Amazon%20DynamoDB%20is%20a%20fully.

[16] Amazon (n.d.). *Cost optimization - AWS Lambda*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/lambda/latest/operatorguide/cost-optimize.html [Accessed 23 Nov. 2023].

[17] Sysdig. (n.d.). *AWS CloudFront Security*. [online] Available at: https://sysdig.com/learn-cloud-native/cloud-security/aws-cloudfront-security/.

[18] Amazon (n.d.). *Amazon Simple Notification Service (SNS) Product Details - AWS*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/sns/details/#:~:text=Flexible [Accessed 23 Nov. 2023].

[19] Zambrano, B. (2018). *Serverless Design Patterns and Best Practices: Build, secure, and deploy enterprise ready serverless applications with AWS to improve developer productivity*. [online] *Amazon*. Packt Publishing. Available at: https://www.amazon.com/Serverless-Design-Patterns-Best-Practices/dp/178862064X [Accessed 23 Nov. 2023].

[20] www.oreilly.com. (n.d.). *Designing Data-Intensive Applications [Book]*. [online] Available at: https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/.

[21] Amazon Web Services, Inc. (2019). *Amazon ElastiCache- In-memory data store and cache*. [online] Available at: https://aws.amazon.com/elasticache/.

[22] Singh, A.D. and Carpenter, J. (2023). *Building and Delivering Microservices on AWS: Master software architecture patterns to develop and deliver microservices to AWS Cloud*. [online] *Amazon*. Packt Publishing. Available at: https://www.amazon.com/Delivering-Microservices-AWS-architecture-microservices/dp/1803238208 [Accessed 23 Nov. 2023].

[23] Amazon (n.d.). *AWS Certificate Manager FAQs | Certificate Management | Amazon Web Services*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/certificate-manager/faqs/#:~:text=Private%20Certificate%20Authority- [Accessed 25 Nov. 2023].