Swinburne University of Technology

Ho Chi Minh Campus

COS20019

Cloud Computing Architecture

Assignment 3

Serverless/Event-driven Architecture Design Report

1. Nguyen Quoc Thang - **104193360**
2. Lau Ngoc Quyen - **104198996**
3. Nguyen Do Nhat Nam - **104061616**

Lecture Section: Tuesday 7:00 AM

March 2024

Abstract

- This abstract presents the concept of AWS Serverless/Event-driven Architecture Design, a transformative approach to web operations leveraging Amazon Web Services (AWS). The architecture design eliminates and reduces the need for server management, enabling businesses to focus on core functionalities and deliver value efficiently. Leveraging real-time events and triggers, Adapting AWS services such as AWS Lambda, Amazon S3, Amazon DynamoDB, and Amazon API Gateway provide a scalable and flexible framework. This abstract highlights various benefits of AWS Serverless/Event-driven Architecture Design in improving performance and efficiency for businesses. A thorough examination of the specific business scenario is emphasized for effective implementation and optimization of the architecture.

- **Keywords:** Cloud computing, AWS Cloud Computing services, Serverless/Event-driven, Business Scenario, Design Rationale , Three-tier Architecture

1. Introduction

- This article examines the importance of cloud computing, specifically AWS, in altering enterprises' online architectures. It emphasizes the cost-effectiveness and efficiency of cloud computing in infrastructure management, bridging the divide between on-premises and cloud-based websites. The independence of AWS services allows for a serverless/event-driven strategy, increasing web operation efficiency and flexibility by automatically triggering web functionalities. The article provides a customized serverless/event-driven architectural design based on AWS services, with the goal of improving business performance and efficiency for specific scenarios. It underlines the need of properly analyzing the company's business situation before delivering a personalized design

2. Business Scenario

- The Photo Album application you developed has met with amazing success and needs to be further developed to meet increasing demand. In particular, the following problems/requirements have been identified by the company:

2.1 Management of Infrastructure:

- Company requires to use managed AWS cloud services to minimize the need for in-house systems administrations and data storage.

2.2 Scalability:

- Ensuring flexible scalability and reduced operating costs
- Apply serverless solutions
- The designed architecture needs to be scalable to cope with expected growth in the future.

2.3 Database Optimization

- Exploring alternatives options for a relational database that provide perfect balance between cost and effectiveness.

2.4 Cross-region performance

- Improve the system's data response capabilities

2.5 Media processing evolution and optimization

- Develop a scalable architecture for the next generation of autonomous cars;
- Have the capacity to self-scale in order to prevent overload
- Create an architecture that can easily handle video content in the future
- Install a content distribution network that is closer to users worldwide

Our design below will show how we solved the above problems and will also demonstrate the effectiveness of our design.

3. AWS Serverless Architecture Design

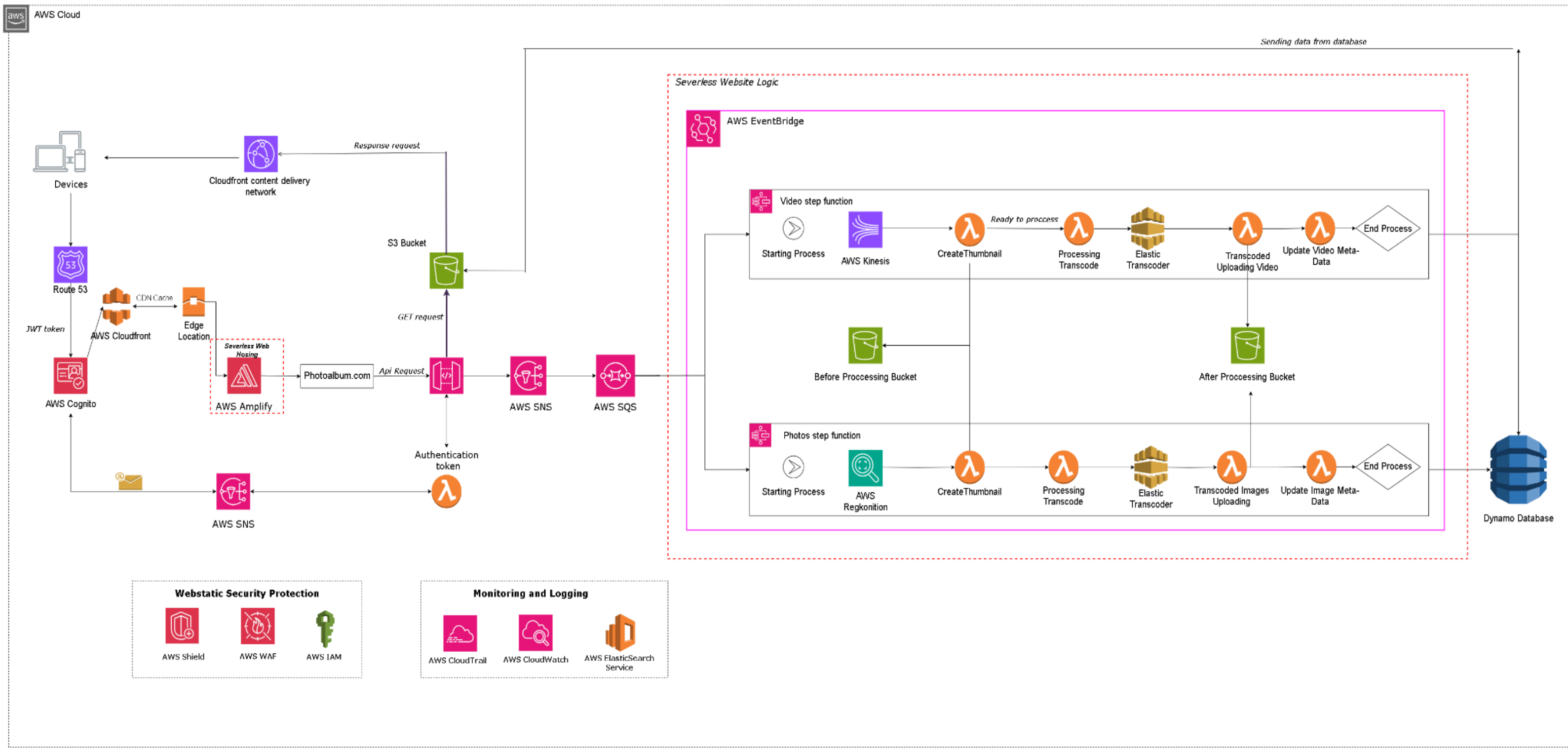

Image 1: Serverless Architecture Design

3.1 AWS Services for Content Delivering

- To meet the demands of a quickly developing website and guarantee a great user experience while maintaining a competitive edge, effective content delivery solutions must be utilized. AWS provides two major services to help do this: AWS Route 53 for DNS resolution and Amazon Web Services CloudFront for content delivery

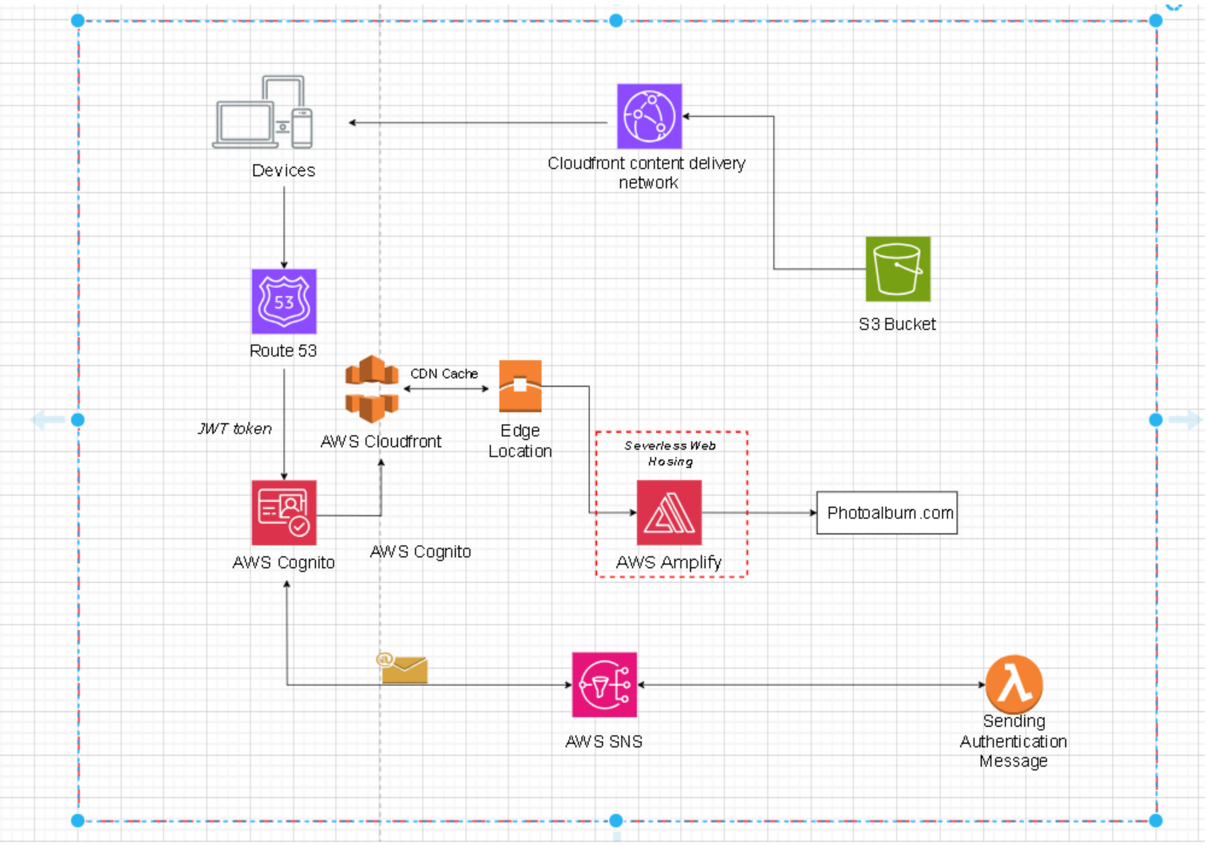

Image 2: Content Delivery System

- AWS Route 53 is shown as the starting point in the diagram. It represents the DNS management service provided by AWS. Route 53 allows users to navigate through website DNS and interact with website static hosted by AWS Amplify . It acts as the entry point for incoming requests to the serverless applications.
- Before entering the website, users must authenticate their identity via AWS Cognito (With the help of AWS WAF) by sending an authentication SMS for highly secured user's information and avoiding DDos and SQL injection attacks to the web static
- Amazon Web Services CloudFront is a content delivery network (CDN) service offered by AWS. CloudFront speeds up the delivery of online material to end users by caching it at edge locations. Edge locations are demonstates as small data centers distributed globally. These locations are part of the CloudFront infrastructure and act as caching user endpoints for content delivery. They are strategically located to reduce latency and improve performance.
- All the Media data will be stored in AWS S3 Bucket and delivered via Cloudfront if only the user makes a request.

3.2 Three-Tier Serverless Architecture

- Nowadays, three-tier serverless architecture is a popular software architectural model because it is strong, adaptable, and affordable. The presentation tier, logic tier, and data tier are its three distinct layers.

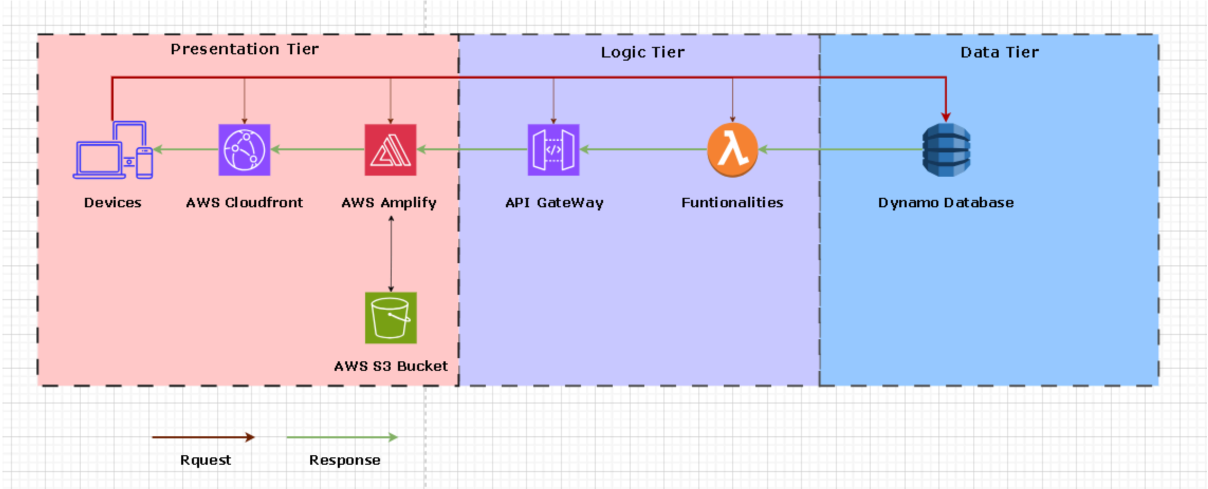

Image 3 : Three-tier Architecture

- The user sends requests through the devices, as shown in the above diagram, and AWS CloudFront serves them with the content. Our presentation layer is then constructed and deployed by AWS Amplify using the material that has been saved in an S3 bucket. I'll go to the logic layer next. The API Gateway and Lambda components of our logic layer are responsible for managing the APIs and delivering dynamic content to our website. The AWS noSQL database service DynamoDB is utilized in my data tier to store and handle the application's data. It is also possible to add the needs of different users with this architecture.

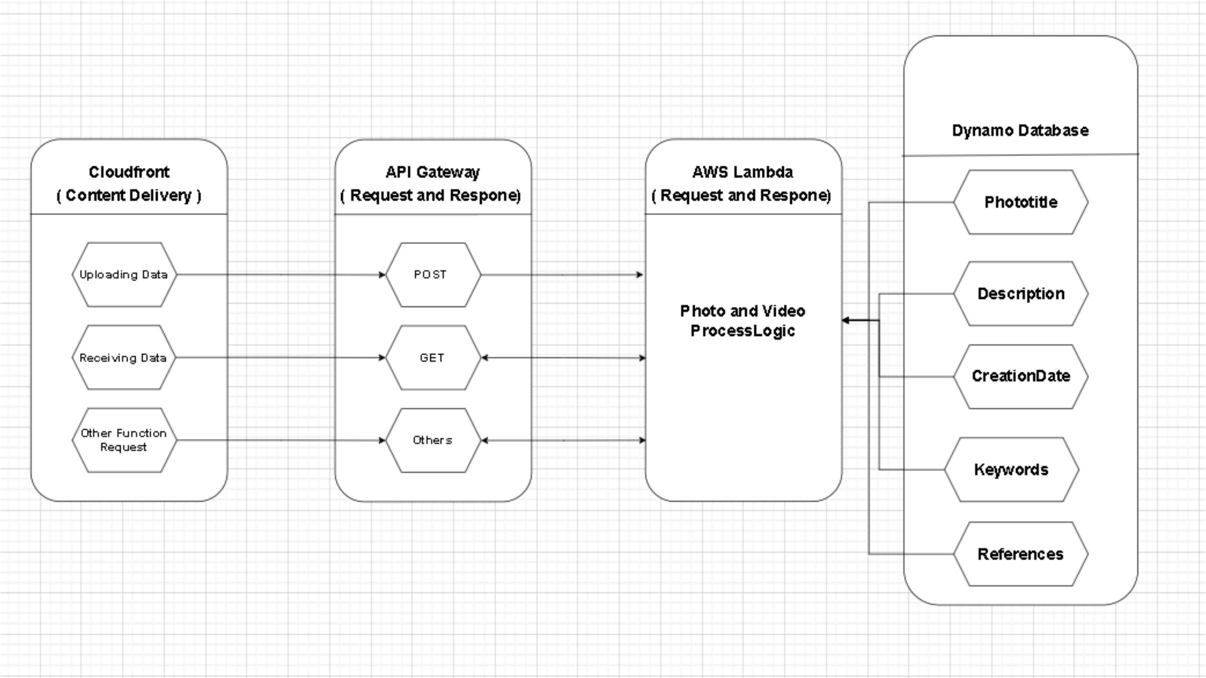

Image 4 : Architecture UML Diagram

- There are several important advantages to using a three-tier design, including the ability to separate functionality, the encouragement of source code and application component reuse, and the segregation of layers contributing to The application is easy to update and maintain, and it helps us safeguard data and manage access to it.

3.3 Dependable Fanout Architecture

- Modern cloud applications frequently employ the fan-out architectural pattern to transmit events or messages from a single source to numerous downstream consumers. AWS SNS, SQS, and Lambda are just a few of the cloud services that can be used to build the design. The message publisher in this architecture is SNS, the message queue is SQS, and the message consumer is Lambda.

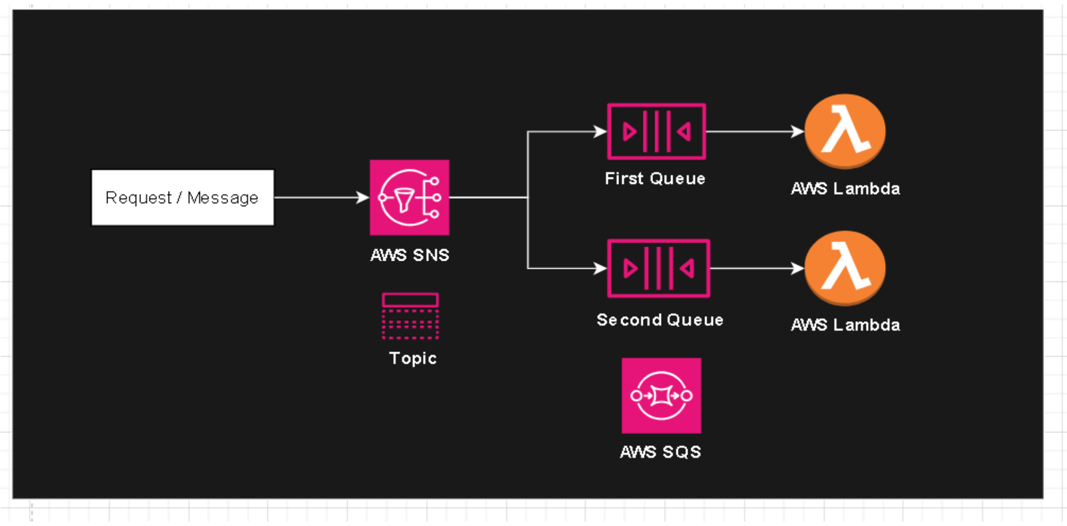

Image 5 : Dependable fan-out pattern

- Using SNS, SQS, and Lambda in the AWS fan-out architecture, messages or request API calls could be distributed from a single source to numerous downstream consumers in a scalable and reliable manner. The way the architecture operates is that messages are published to a topic using SNS, then they are dispersed to various SQS queues that have subscribed to that topic,which creates a flawless decoupled process to reduce latency and 'traffic conjunction' between services. The messages in the queues trigger lambda functions, enabling downstream processing and analysis.

3.4 Decoupled Media Procedure

- Once a message from API gateway is sent, AWS SQS will be announced to trigger a process into the queue and by AWS SNS,which is transferred to the queue for processing, the media will be sent to the media processing step function ,which AWS Event Bridge will implement to process the functionalities of the application.

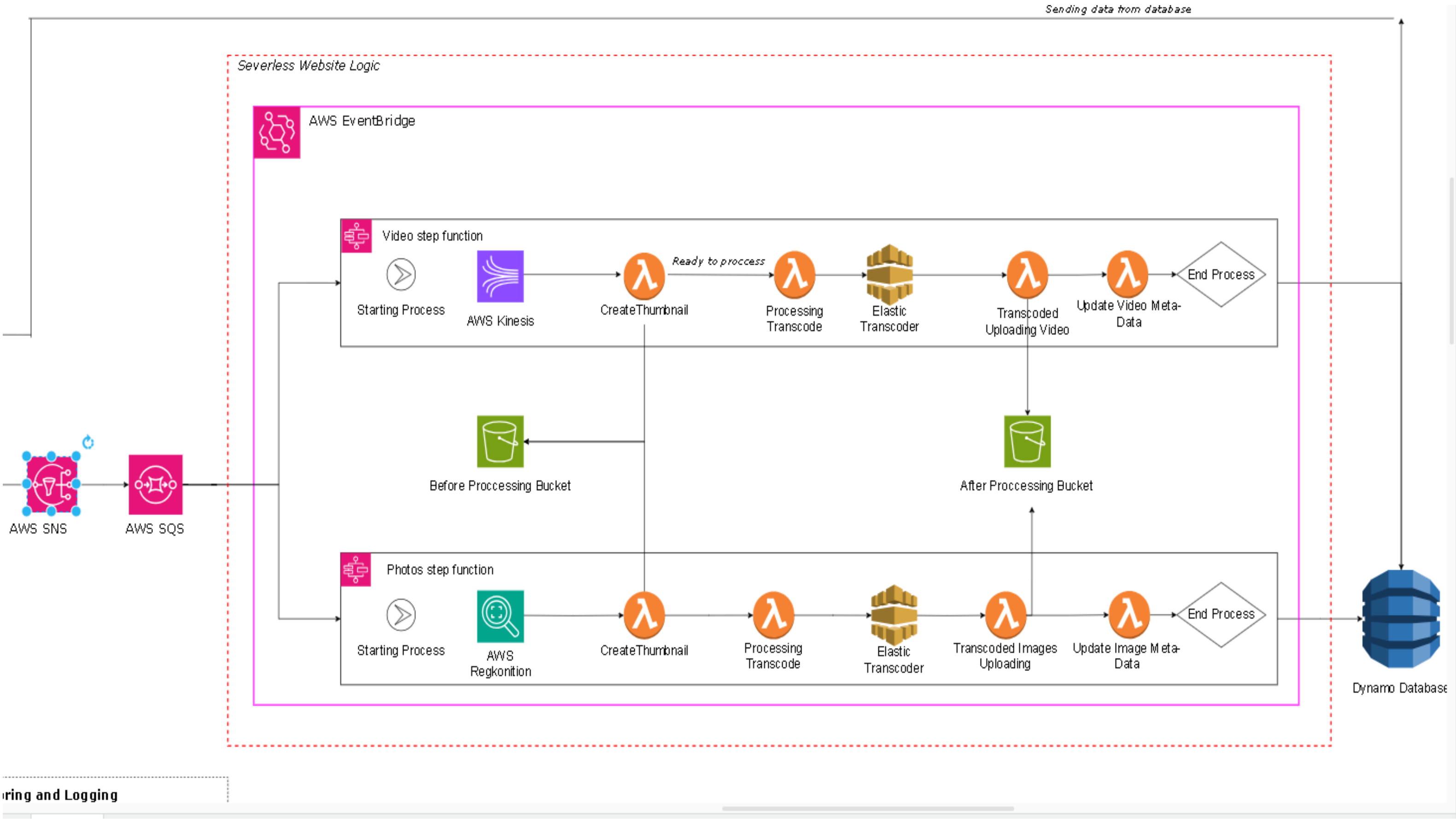

Image 6 : Decoupling Design Process

- The procedure will be divided into two decoupled sections: Photos handling process and Video handling process.

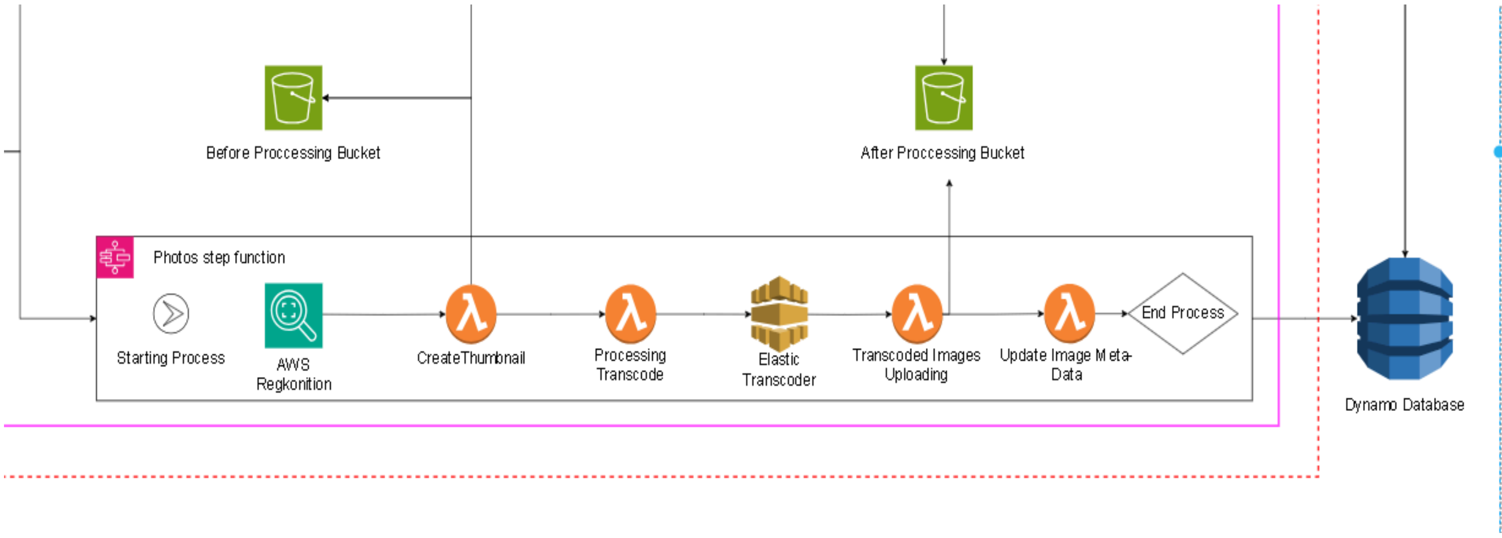

Image 7 : Photos Handling Process

- After being queued by AWS SQS for reducing time-consumption and overloading from clients, AWS Step Functions process will be provoked and trigger the photo processing by AWS Event Bridge,which builds a scalable and effectively decoupling application.
- Initially, AWS Rekognition will perform tasks such as facial recognition, object detection and photos moderation. Its duties are to identify and analyze each photo's particularity for CreateThumbnail lambda in the next step.All the photos will be classified by its specific tag and stored in Before Processing Bucket. Following the reformatting / reprocessing step, Processing Transcode will perform the transcoding of the original image into different format or resolution based on the API call and input data from users. After the photos are transcoded by the Lambda function, the reformatted / reprocessed photos will be uploaded automatically by sending it into Dynamo Database and retrieved directly to users through Cloudfront. The transcoded photos will also be placed on a separate bucket.

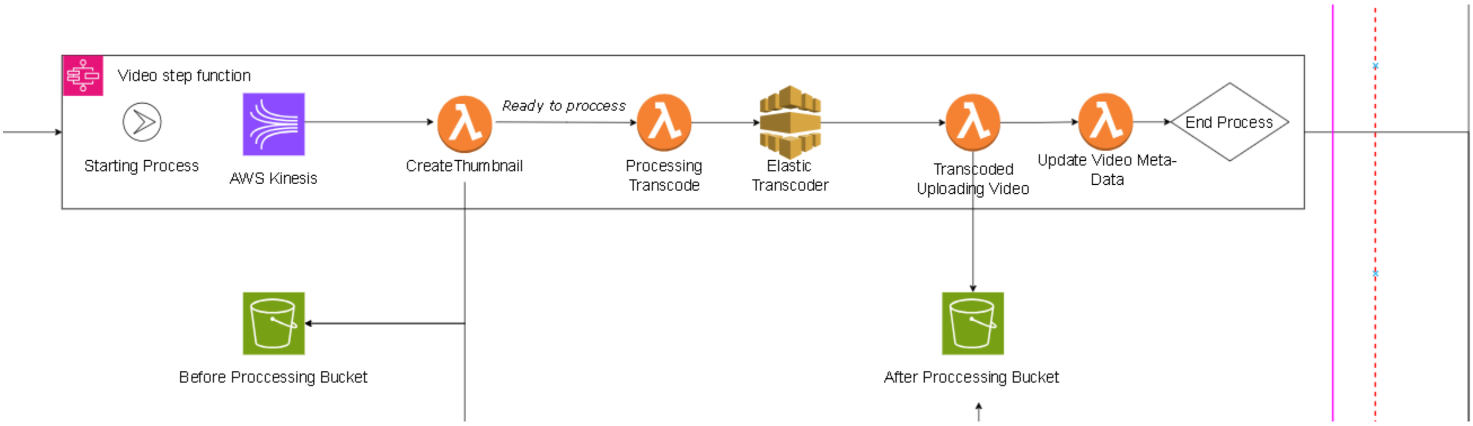

Image 8 : Video Handling Process

- The CreateThumbnail lLambda function is triggered by the AWS Kinesis Lambda. This function generates a thumbnail image from an uploaded video by extracting a representative frame. The thumbnail image is then stored in an Amazon S3 bucket, which provides scalable and durable object storage.Simultaneously, the transcoded video, which may involve converting the original video into different formats or resolutions, is also stored in the same Amazon S3 bucket. This allows for efficient storage and retrieval of the processed videos.Following the storage of the transcoded video, the metadata associated with the video, such as video title, duration, tags, or other relevant information, is updated in a database or any appropriate data storage solution. This ensures that the metadata remains synchronized and accessible for future retrieval or analysis. Upon completion of the video transcoding, storage, and metadata updates, the process concludes. The application has successfully utilized AWS Amplify, API Gateway, SQS, AWS Kinesis Lambda, CreateThumbnail Lambda, Amazon S3, and a database to enable a streamlined workflow for video uploads, transcoding, storage, and metadata.

3.5 Addition Services

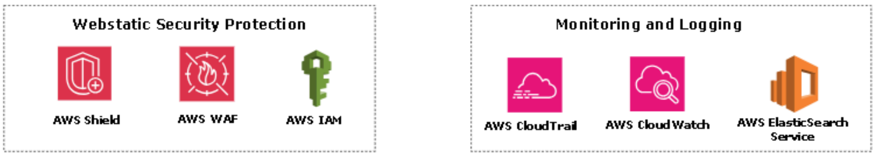

Image 9 : Website Security and Developer Monitoring

- CloudTrail, CloudWatch, Elasticsearch, AWS WAF, AWS Shield, and AWS IAM are important AWS services that offer various benefits for building a highly secure and scalable highly available Web site .
- CloudTrail enables clients to monitor and track activities within their AWS accounts by recording API calls and actions. This provides valuable insights into account usage, helps with security audits, and supports compliance monitoring.
- Amazon CloudWatch is a monitoring and observability service provided by Amazon Web Services (AWS). It is designed to collect and track metrics, log files, and events from various AWS resources and applications. CloudWatch provides developers with valuable insights into

the performance and health of their applications, infrastructure, and services running on AWS.Developers can use CloudWatch to collect and store metric data, which can be visualized using customizable dashboards. These metrics can be related to CPU utilization, network traffic, database performance, and many other aspects of system behavior. By monitoring these metrics, developers can gain a deep understanding of their application's performance and identify areas for optimization and improvement.

- Elasticsearch is a search and analytics engine that allows clients to search and analyze large volumes of data. It supports advanced querying, visualization, and near real-time search experiences, enabling clients to extract meaningful insights from their data.
- AWS WAF (Web Application Firewall) helps protect web applications from common web exploits and attacks. It provides customizable rules to filter and monitor HTTP/HTTPS traffic, safeguarding applications from malicious traffic and ensuring data integrity.
- AWS Shield is a managed Distributed Denial of Service (DDoS) protection service. It safeguards applications against volumetric, state-exhaustion, and other DDoS attacks, ensuring high availability and mitigating potential disruptions.
- AWS IAM (Identity and Access Management) is a service that enables clients to manage user identities and access permissions for AWS resources. It allows for centralized control and fine-grained access management, ensuring secure and auditable access to AWS services.
- Collectively, these services provide clients and developers with comprehensive monitoring, security, analytics, and identity management capabilities. They enhance the overall security posture, operational efficiency, and data-driven decision-making for AWS deployments.

4. Initial Architecture Design

- Now that we have explored the AWS architecture and its potential benefits, let's delve into how this design can address the identified pain points in the business scenario. By leveraging the various components of AWS, such as compute, storage, networking, and security, we can develop a scalable and reliable solution. One pain point highlighted in the

previous discussion is the need for real-time processing of streaming data. AWS provides services like Kinesis Data Streams, which can handle large volumes of streaming data in real-time. By utilizing AWS Lambda functions, such as CreateThumbnail Lambda and ProcessTranscodeLambda, we can efficiently process and transcode videos on-the-fly, enabling seamless delivery and storage in Amazon S3.
Another pain point revolves around metadata management. AWS offers database services like Amazon DynamoDB or Amazon RDS, which can be integrated to store and update video metadata in a structured and efficient manner. This ensures that relevant information such as video title, duration, and tags are readily available for future analysis or retrieval.

In terms of alternative solutions and design criteria, it is crucial to consider factors such as cost-effectiveness and flexibility. AWS offers a pay-as-you-go model, allowing businesses to scale resources based on demand, thereby optimizing costs. Additionally, the use of serverless computing with AWS Lambda eliminates the need for provisioning and managing servers, providing a more flexible and cost-efficient solution.

By carefully analyzing the pain points, utilizing the capabilities of AWS architecture, and considering alternative solutions, we can design a robust and efficient system that addresses the specific needs of the business scenario. This comprehensive approach ensures scalability, reliability, and cost-effectiveness, ultimately leading to a successful implementation and improved operational efficiency for the organization.

4.1 Alternative Solution Design

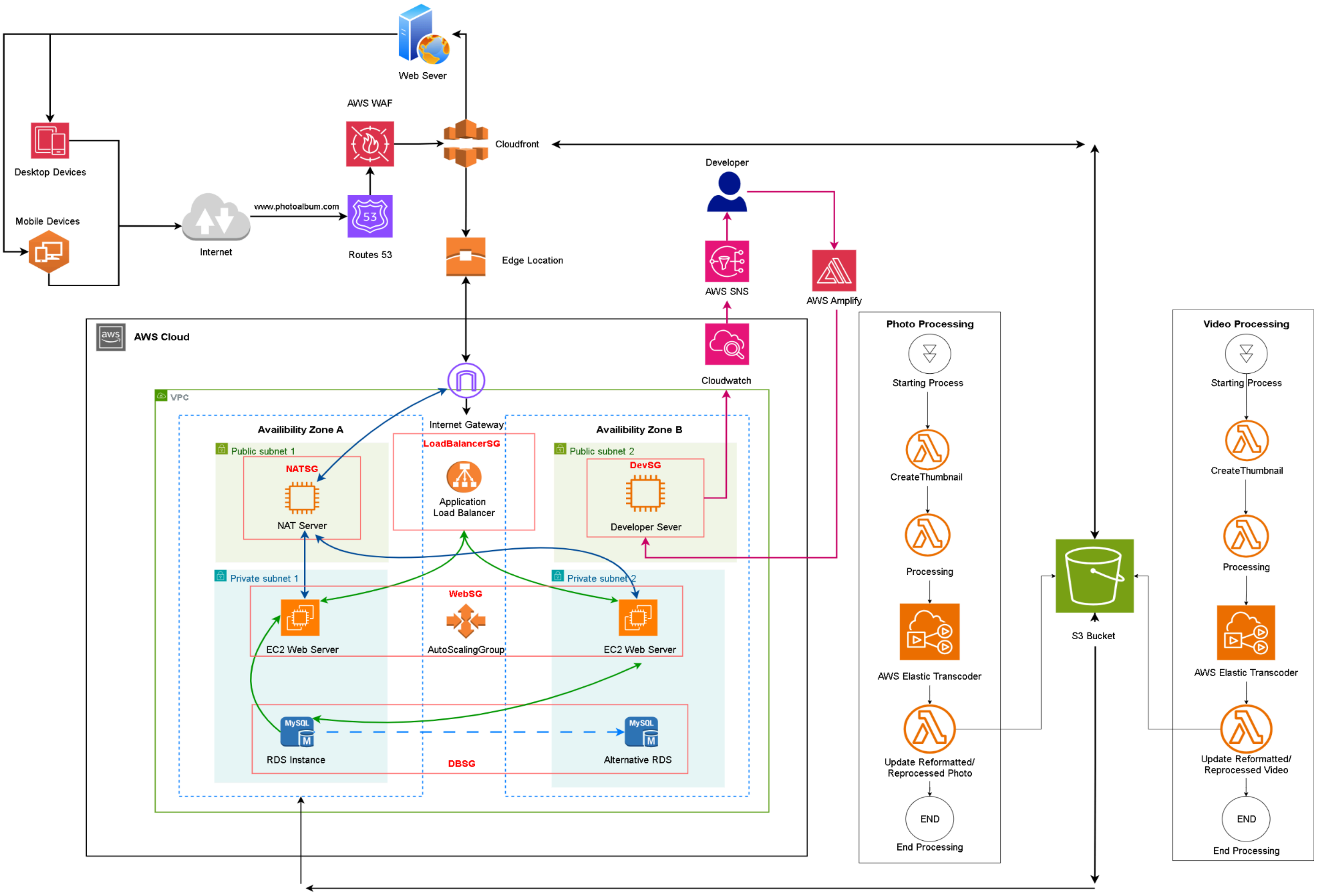

Image 10 : Initial Website Architecture

- During the architecture building phase, two versions of the AWS architecture were developed. The Initial architecture utilized simpler AWS services. However, it is important to note that despite its simplicity in terms of service selection, the original architecture was fundamentally more complex compared to the alternative architecture that is currently in use.

Initial Presentation Tier

- Users that want to access media material will be sent to CloudFront, a content delivery network (CDN) service offered by Amazon Web Services. CloudFront serves as a proxy, directing user requests to an EC2 instance that hosts the website. However, the website is housed in a private subnet, therefore it is not immediately accessible to users. To allow user access, NAT instances are used. The NAT instance functions as a bridge, allowing visitors to see website content by routing their requests from the public internet to the private subnet where the website is hosted. This configuration guarantees that the website stays safe within the private subnet while also allowing users to access and see the media content via The NAT instances.

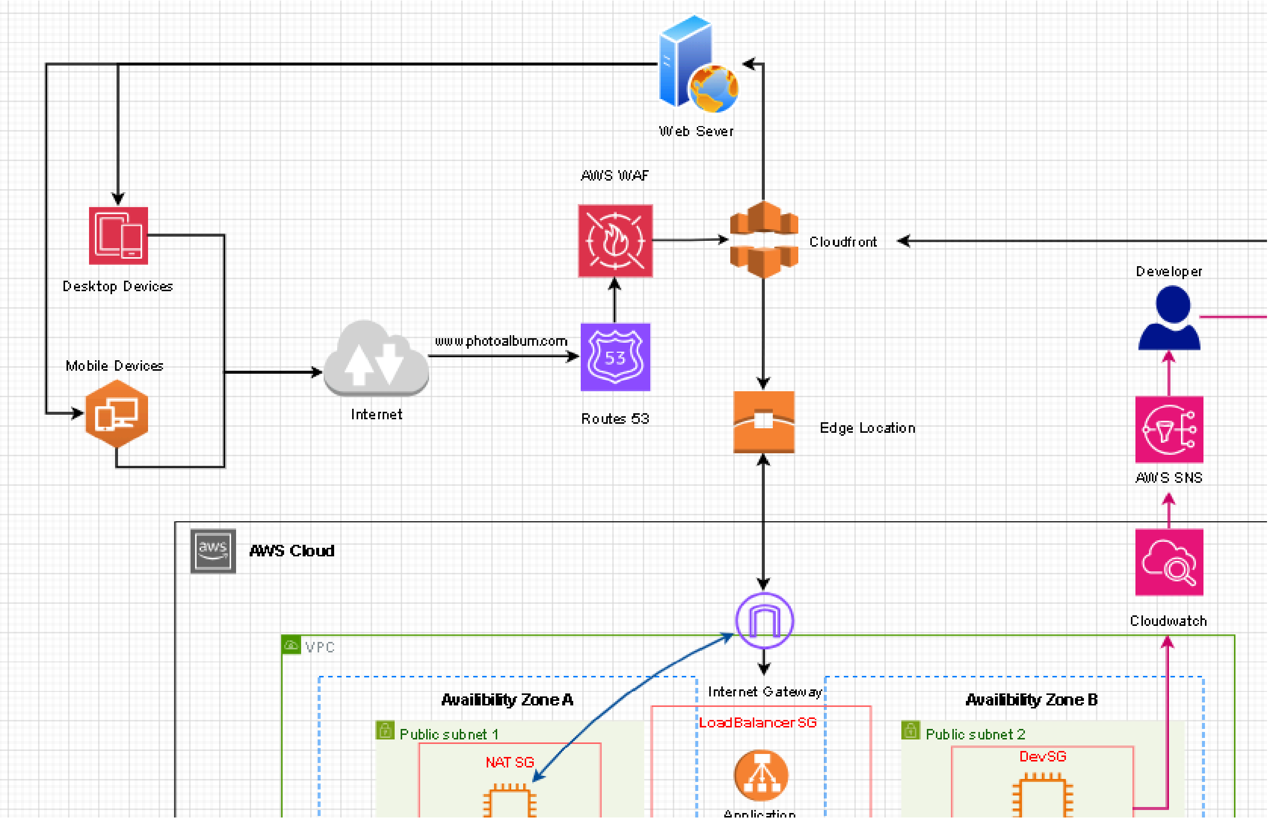

Image 11 : Accessing Website via Internet Gateway

- In the second architecture, users access the website by directly connecting to it through the internet gateway. The website's source code, meta-data and others are stored in RDS instances after requesting processed data being sent by Amazon S3 Bucket.AWS Cloudfront facilitates the deployment and hosting of the website by receiving it via RDS instances.

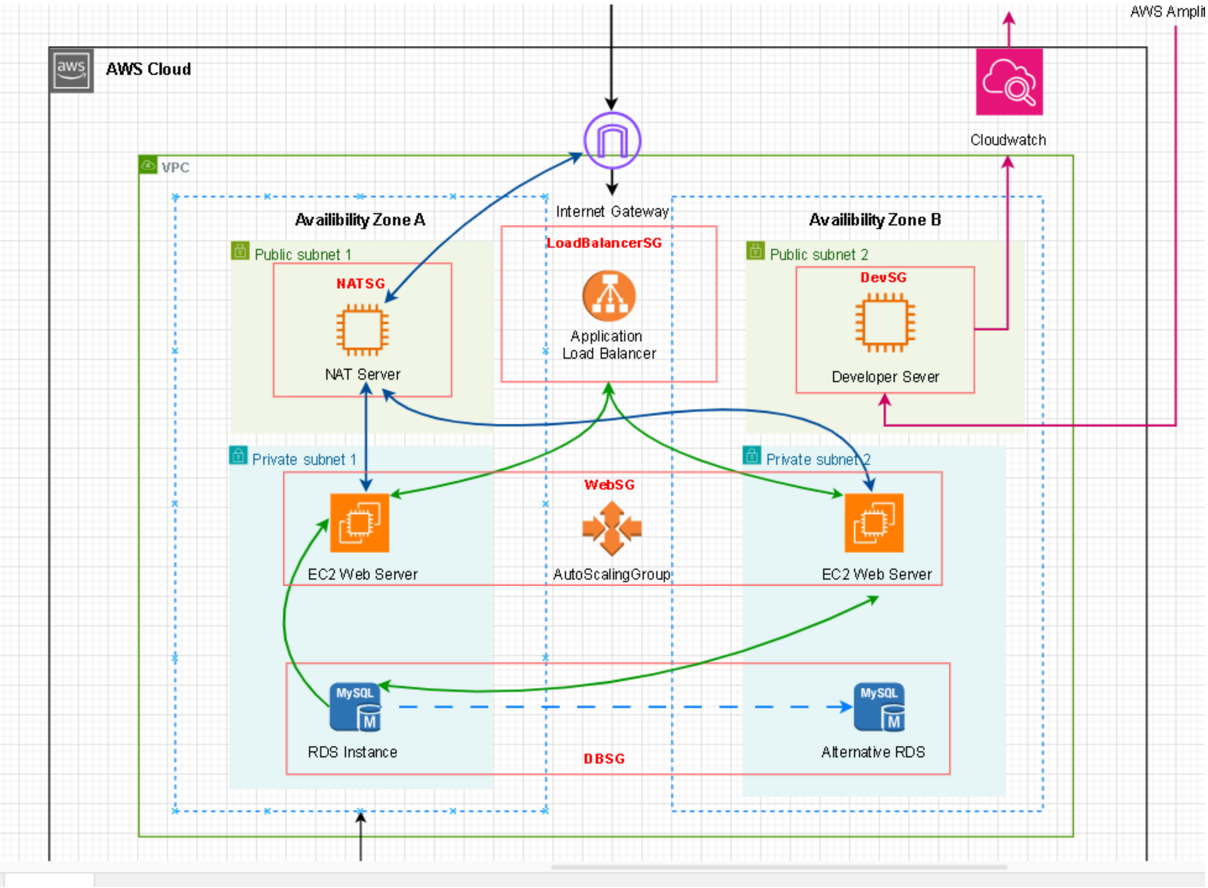

Image 12 : Controlled Access with NAT instances in cloud enviroment

- To enable communication between users and the website, private IP addresses are used. When a user sends a request to the website, the request is routed through the internet gateway, which allows external internet traffic to enter the AWS network. The request then reaches a NAT device, which performs network address translation. The NAT device translates the private IP addresses used within the AWS network to public IP addresses recognizable on the internet.
- By using private IP addresses translated by a NAT device, the architecture ensures secure communication between the users and the website while maintaining accessibility. This setup also reinforces the high availability

advantage of this architecture by eliminating the need for additional components like instances or gateways.

- This cloud infrastructure on Amazon Web Services (AWS) offers a robust and adaptable solution for businesses. The core of this system is a Virtual Private Cloud (VPC), essentially a secure, isolated network within the broader AWS cloud. This VPC ensures control and high security by creating a private environment for your resources. To safeguard against unexpected outages, the VPC is divided across two Availability Zones (AZs). These geographically distinct zones act as backups for each other, so if one zone encounters an issue, the other continues to function seamlessly.
- Within the VPC, resources are strategically placed in subnets, categorized as either public or private. Public subnets, like Public Subnet 1 in the diagram, have a direct route to the internet, allowing internet-facing resources like the EC2 Web Server to operate. Private subnets, on the other hand, prioritize security and do not have a direct internet connection. Resources like the RDS MySQL database in a private subnet benefit from this isolation. To enable these private resources to access the internet when needed, a NAT Gateway acts as a controlled intermediary, facilitating outbound connections.
- Security Groups function like virtual firewalls, meticulously controlling both inbound and outbound traffic throughout the VPC. This ensures only authorized communication flows within the network. Amazon EC2, the core building block, provides on-demand virtual servers that can be scaled up or down based on needs. An Auto Scaling Group automates this process, ensuring optimal resource allocation during traffic fluctuations. Additionally, an Application Load Balancer distributes incoming traffic efficiently across multiple targets, such as the EC2 Web Servers, preventing any single server from becoming overloaded.
- In essence, this intitial cloud infrastructure offers a multitude of advantages. It provides a secure and isolated environment with exceptional fault tolerance through Availability Zones. On-demand scalability through EC2 and Auto Scaling Groups ensures cost-effectiveness and optimal performance. Finally, the load balancer distributes traffic efficiently, guaranteeing a smooth user experience. This comprehensive solution empowers businesses to build and manage robust, secure, and adaptable applications in the cloud.

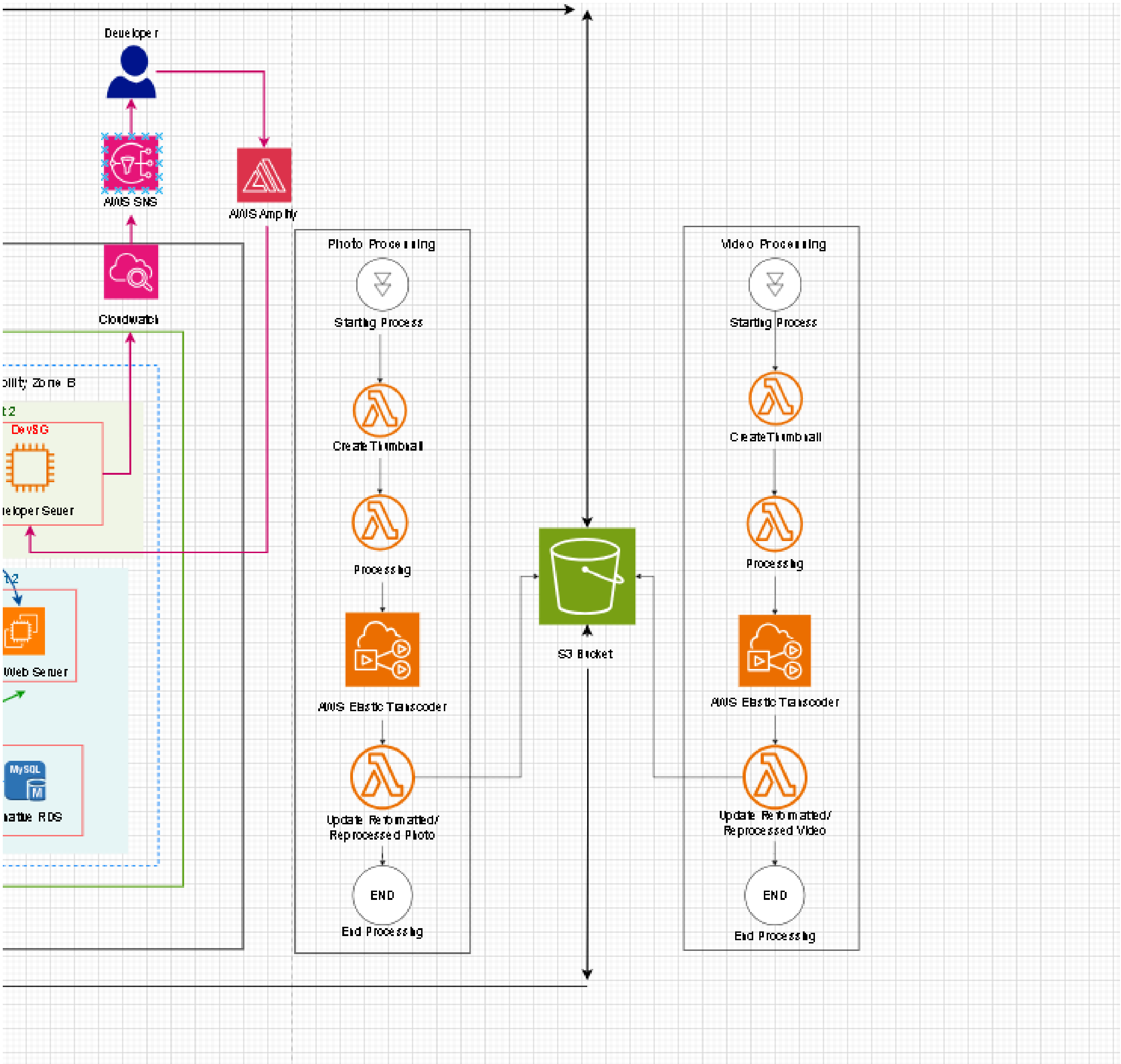

Image 13 : Media Publishing Process

- This pipeline efficiently processes images and videos. Images are uploaded to secure S3 storage and potentially prepped on a developer server. Thumbnails are created for faster web loading, and both images and videos are converted to an optimized format using Elastic Transcoder. Processed data is then stored in a MySQL database, with a managed alternative via RDS available. The video path might include additional processing after format conversion. This cloud-based approach offers scalability, secure storage, format flexibility, and efficient data management. Developments can control and collect metrics and log of the website via DevServer instance collected by Cloudwatch and sent messages directly by AWS SNS.

Traditional Three-tier and Serverless Three-tier Architecture Comparison

- The serverless three-tier architecture is chosen for its ease of management, leveraging the capabilities provided by AWS. With this approach, AWS takes care of certain management tasks such as scaling and security, relieving the administrator of those responsibilities. In the original VPC-based architecture, although an auto-scaling group is utilized, the administrator still needs to monitor instances and the DevServer Instance for potential updates. However, in the serverless architecture, AWS Amplify handles all the scaling automatically, allowing the administrator to focus more on improving performance and ensuring optimal functionality rather than spending significant time monitoring scaling operations.
- In terms of security, the VPC environment can be complex to configure manually for each component, requiring the administrator to set up security groups and network access control lists (NACLs). This process can be frustrating and challenging to navigate. On the other hand, the serverless architecture implemented with AWS Amplify offers built-in security features. It provides authentication, authorization, and data encryption, ensuring a robust security posture. Additionally, security is bolstered through the use of IAM (Identity and Access Management) roles, which grant granular access control and enhance overall security measures.
- Furthermore, AWS offers additional services that can enhance security and monitoring within the serverless architecture. CloudWatch enables administrators to monitor and collect metrics, logs, and events, providing insights into the performance and health of the application. CloudTrail logs API and management events, allowing for auditing and tracking changes made to resources. These services contribute to a comprehensive security and monitoring framework within the serverless architecture, further strengthening its advantages over VPC-based applications.

4.2 Design Criteria

- In this section we will look specifically at the design of an AWS system, we focus on the design, construction, as well as organizational structure of the services used to meet the needs of a complex web application

a. **Performance :**

Criteria:

- Efficiently speed up the process of processing, uploading and downloading images and videos
- Ensure web application page loading with minimal latency to provide users with the best possible experience

Solution:

- To automatically and effectively process and convert images, we make use of technologies like AWS Lambda and Elastic Transcoder. Additionally, I handle and analyze streaming data that includes photos and videos using AWS Kinesis and AWS Rekognition.
- Use CloudFront to speed up website loading by installing content onto globally distributed servers located closest to users.

b. **Scalability**

Criteria:

- The website is capable of automatically scaling itself to accommodate increasing traffic and increasing amounts of stored data.
- Flexible extensibility and reuse to store and modify data effectively

Solution:

- Use AWS Lambda and AWS DynamoDB to dynamically scale up and down based on the amount of work being added or pulled
- Using API Gateway allows us to build, process and manage APIs and provides the ability to scale up to handle increased traffic and scale down when traffic decreases.
- Use AWS SQS and AWS SNS to distribute content and send messages to internal or external systems, reducing the burden on key components and handling large workloads with ease. .

c. **Reliability**

Criteria:

- The ability to operate continuously and can be used at all times even when problems occur

- The capacity to recognize mistakes and take appropriate action automatically
- The ability to recover to normal quickly after an incident occurs
- The capacity to identify issues and swiftly and efficiently fix them

Solution:

- Decoupled architecture to lessen the impact of errors
- Use AWS CloudFront and AWS CloudTrail to monitor and log activities, which help detect errors or resolve them automatically
- AWS S3 has the ability to automatically back up data from servers or systems, data will be protected and can be restored every time the system crashes.

d. Security

Criteria:

- The system is capable of protecting the system from theft and unauthorized access
- The ability to detect unusual activities or actions that threaten the security of the system
- Access control capabilities, so only authorized users can access system resources
- Defend the system from DDoS assaults to guarantee application accessibility.

Solution:

- Use AWS IAM to control access to AWS resources
- Use AWS Shield to protect AWS systems, applications, and resources from DDoS attacks
- Use AWS WAF to prevent invalid requests from malicious systems, or protect applications and systems from security attacks such as SQL Injection, or some other type of attack

e. Estimated Cost

- Based on the business scenario, the application's demand has been increasing every 6 months and is expected to continue for the next 2-3 years, based on available data. To estimate future website costs, cost summaries for media upload sizes of **50GB, 100GB** are provided below.
- The overall cost of operating the **50GB** architecture is about $245.99. The pricing includes the Content Delivery Network, Web architecture, and media processing technique and others.

- Similarly, the future year's projection for **100GB** is $389.38. For further information, check

Detailed Cost Analysis for Uploading 50GB of Media

Service	Cost of each feature	Total
Routes 53	- 3 hosted zone : 3 * 0.5 = 1.5 USD - 10.000.000 standard queries * 0.0000004 USD = 4 USD -2 HTTPS Checks (AWS) * 1 USD = 2 USD 2 latency Measurement checks (AWS) * 1.00 USD = 2 USD	-Total monthly: 1.5 + 4 + 2 + 2 = 9.5 USD
AWS Cognito	50000 monthly active users(MAUs) => free	-Total monthly cost: 0 USD
AWS CloudFront	- Data transfer out to internet cost : 50GB * 0.085USD = 4.25USD -Data transfer out to origin cost : 50GB * 0.02GB = 1 USD -Request cost: 2,500,000 requests * 0.0000001 USD = 2.5USD	-Total monthly cost for United states region: 7.75 USD
AWS Amplify	-3 minutes build monthly * 0.01 = 0.03USD -50GB data stored/month * 0.023 = 1.15 USD -50GB Data Served/month * 0.15 = 7.5USD	-AWS Amplify static web hosting costs(monthly) : 8.68 USD
AWS API Gateway	-HTTP API request cost(monthly) : 1 USD -Tiered price total for REST(Push/Post/Delete...) API Requests : 3.5 USD	- Total monthly cost for API calls: 4.5 USD

AWS SNS	-Email notification(Sign up/Sign in/ Authentication Rechange information) : 1 million/month *1000000 multiplier = 1000000/month -SQS notification : 2 million/month *1000000 multiplier = 2000000 per month - Tiered price for 1.000.000 calls 1000 calls * 0 USD = 0USD 999.000 calls * 0.00002 USD = 19.98 USD	-Total tier cost : 0 + 19.98 = 19.98 USD
AWS SQS	- FIFO Queues/month : 0$ -Data transfer : 0$ -Data transfer OUT/month : 0.09 * 100 = 9 USD	- Total monthly for SQS queues: 9 USD
AWS Kinesis	-Number of days for data retentions : 168 hours -Stream cost : 29.20 USD	-Kinesis data stream cost On-demand mode(monthly) : 29.20$
AWS Lambda		-Lambda cost - with free tier(monthly) : 1.8 USD
AWS S3	-Tiered price for : 50 GB 50GB * 0.023 = 1.15 USD -S3 standard cost(monthly) : 1.19USD -10.000.000 GET requests in a month * 0.0000004/request = 4 USD - S3 Object Lambda charges : 9.5367432 USD - AWS Lambda charges : 2 USD	- Total cost : 16.73 USD
AWS Event Bridge	-Total API destination cost(monthly) : 0.2 USD	- Total monthly : 8.45 USD

	-Total events cost(Monthly) : 4 USD - Total event replay cost(Monthly) : 4.25 USD	
AWS Step Function	- Standard workflows pricing (monthly) : 0 USD	- Total monthly cost: 0 USD
AWS Rekognition	- Image pricing : 1.02 USD	- Total monthly cost: 1.02 USD
AWS Elastic Transcoder	- 50 GB are used for transcoding video and image : 0.03 * 200 = 6 USD	- Total monthly cost: 6 USD
AWS DynamoDB	-Monthly write cost (Monthly): 23.38 -Monthly read cost (Monthly): 2.76 -DynamoDB data storage cost (Monthly): 12.50 -DynamoDB Data export to Amazon S3 cost (Monthly): 5.00 -DynamoDB Data import from Amazon S3 cost (Monthly): 7.50 -DynamoDB Backup and restore cost (Monthly): 5.00 -Upfront write cost (Upfront): 150.00 -Upfront read cost (Upfront): 30.00	-Total Upfront cost: 180.00 USD -Total Monthly cost: 56.14 USD
AWS WAF	-1 Web ACLs per month x 5.00 USD = 5.00 USD (WAF Web ACLs cost) -4 Rule Groups per Web ACL x 4 Rules per Rule Group = 16.00 -Total Rules in Rule Groups	- WAF cost (monthly): 33.60 USD

	16.00 -Total Rules in Rule Groups + 4 Rule Groups per Web ACL + 4 Rules added per Web ACL + 4 Managed Rule Groups = 28.00 -Total billable Rules per Web ACL 1 Web ACLs per month x 28.00 Billable Rules per web ACL per month x 1.00 USD = 28.00 USD (WAF Rules cost) 1 requests per month x 1000000 multiplier for million x 0.0000006 USD = 0.60 USD (WAF Requests cost) 5.00 USD + 28.00 USD + 0.60 USD = 33.60 USD	
AWS IAM	3 average roles per account + 3 average users per account = 6 -10 number of accounts to monitor x 6 total number of average roles and average users x 0.20 USD = 12 USD Total Cost for Unused Access Findings (monthly): 12.00 USD 10 requests to CheckNoNewAccess API + 10 requests to CheckAccessNotGranted API = 20 Total number of Requests 20 requests x 0.002 USD = 0.04 USD Total Cost for Custom policy Checks (monthly): 0.04 USD	- Total Monthly cost: 12.04 USD

AWS Cloudwatch	Tiered price for: 8 metrics 8 metrics x 0.30 USD = 2.40 USD -Total tier cost = 2.40 USD (Metrics cost (includes custom metrics)) -CloudWatch Metrics cost (monthly): 2.40 USD -CloudWatch Metrics cost (Monthly): 2.40 -CloudWatch API requests cost (Monthly): 0.00	-Total : 9.6 USD
AWS CloudTrail	- S3 data events cost (monthly): 3.00 USD - Lambda data events cost (monthly): 8.00 USD - CloudTrail data events cost (monthly): 11.00 USD	-Total Upfront cost: 0.00 USD -Total Monthly cost: 11.00 USD
AWS ElasticSearch Service	-Amazon OpenSearch Serverless cost (monthly): 0.00 USD	-Total : 0 USD
Estimated Total Cost		-245.99 USD

Detailed Cost Analysis for Uploading 100GB of Media

Service	Cost of each feature	Total
Routes 53	3 hosted zone : 3 * 1.5 = 4.5 USD - 10.000.000 standard queries * 0.0000004 USD = 4 USD -2 HTTPS Checks (AWS) * 1 USD = 2 USD 2 latency Measurement checks (AWS) * 1.00 USD = 2 USD	-Total monthly:4.5 + 4 + 2 + 2 = 12.5 USD
AWS Cognito	50000 monthly active users(MAUs) => free	-Total monthly cost: 0 USD
AWS CloudFront	- Data transfer out to internet cost : 100GB * 0.085USD = 8.5 USD -Data transfer out to origin cost : 100GB * 0.02GB = 2 USD -Request cost: 5,000,000 requests * 0.0000001 USD = 0.5USD	-Total monthly cost for United states region: 11 USD
AWS Amplify	-3 minutes build monthly * 0.01 = 0.03USD -100 GBdata stored/month * 0.023 = 2.3 USD -100 Data Served/month * 0.15 = 15USD	-AWS Amplify static web hosting costs(monthly) : 17.33 USD

AWS API Gateway	-HTTP API request cost(monthly) : 1 USD -Tiered price total for REST(Push/Post/Delete...) API Requests : 3.5 USD	- Total monthly cost for API calls: 4.5 USD
AWS SNS	-Email notification(Sign up/Sign in/ Authentication Rechange information) : 1 million/month *1000000 multiplier = 1000000/month -SQS notification : 2 million/month *1000000 multiplier = 2000000 per month - Tiered price for 1.000.000 calls 1000 calls * 0 USD = 0USD 999.000 calls * 0.00002 USD = 19.98 USD	-Total tier cost : 0 + 19.98 = 19.98 USD
AWS SQS	- FIFO Queues/month : 0$ -Data transfer : 0$ -Data transfer OUT/month : 0.15 * 100 = 15 USD	- Total monthly for SQS queues: 15 USD
AWS Kinesis	-Number of days for data retentions : 336 hours -Stream cost : 58.4 USD	-Kinesis data stream cost On-demand mode(monthly) : 58.4$
AWS Lambda		-Lambda cost - with free tier(monthly) : 1.8 USD

AWS S3	-Tiered price for : 100 GB 100GB * 0.023 = 2.3 USD -S3 standard cost(monthly) : 2.56USD -10.000.000 GET requests in a month * 0.0000004/request = 4 USD - S3 Object Lambda charges : 9.5367432 USD - AWS Lambda charges : 2 USD	- Total cost : 33.46 USD
AWS Event Bridge	-Total API destination cost(monthly) : 0.2 USD -Total events cost(Monthly) : 4 USD - Total event replay cost(Monthly) : 4.25 USD	- Total monthly : 8.45 USD
AWS Step Function	- Standard workflows pricing (monthly) : 0 USD	- Total monthly cost: 0 USD
AWS Rekognition	- Image pricing : 2.04 USD	- Total monthly cost: 2.04 USD

AWS Elastic Transcoder	- 100 GB are used for transcoding video and image : 0.06 * 400 = 24 USD	- Total monthly cost: 24 USD
AWS DynamoDB	-Monthly write cost (Monthly): 46.76 USD -Monthly read cost (Monthly): 5.52 USD -DynamoDB data storage cost (Monthly): 25 USD -DynamoDB Data export to Amazon S3 cost (Monthly): 10 USD -DynamoDB Data import from Amazon S3 cost (Monthly): 15 USD -DynamoDB Backup and restore cost (Monthly): 10 USD -Upfront write cost (Upfront): 150.00 -Upfront read cost (Upfront): 30.00	-Total Upfront cost: 180.00 USD -Total Monthly cost: 112.28 USD

AWS WAF	-1 Web ACLs per month x 5.00 USD = 5.00 USD (WAF Web ACLs cost) -4 Rule Groups per Web ACL x 4 Rules per Rule Group = 16.00 -Total Rules in Rule Groups 16.00 -Total Rules = 28.00 USD -Total billable Rules per Web ACL 1 Web ACLs per month x 28.00 Billable Rules per web ACL per month x 1.00 USD = 28.00 USD (WAF Rules cost) 1 requests per month x 1000000 multiplier for million x 0.0000006 USD = 0.60 USD (WAF Requests cost) 5.00 USD + 28.00 USD + 0.60 USD = 33.60 USD	- WAF cost (monthly): 33.60 USD
AWS IAM	3 average roles per account + 3 average users per account = 6 -10 number of accounts to monitor x 6 total number of average roles and average users x 0.20 USD = 12 USD Total Cost for Unused Access Findings (monthly): 12.00 USD 20 requests x 0.002 USD = 0.04 USD Total Cost for Custom policy Checks (monthly): 0.04 USD	- Total Monthly cost: 12.04 USD

AWS Cloudwatch	Tiered price for: 16 metrics 16 metrics x 0.30 USD = 4.8 USD -Total tier cost = 2.40 USD (Metrics cost (includes custom metrics)) -CloudWatch Metrics cost (monthly): 2.40 USD -CloudWatch Metrics cost (Monthly): 2.40 -CloudWatch API requests cost (Monthly): 0.00	-Total : 12 USD
AWS CloudTrail	- S3 data events cost (monthly): 3.00 USD - Lambda data events cost (monthly): 8.00 USD - CloudTrail data events cost (monthly): 11.00 USD	-Total Upfront cost: 0.00 USD -Total Monthly cost: 11.00 USD
AWS ElasticSearch Service	-Amazon OpenSearch Serverless cost (monthly): 0.00 USD	-Total : 0 USD
Estimated Total Cost		-389.38 USD

4.3 Fulfilling Business Scenario Requirements and Providing Justification for the Serverless Architecture

4.3.1 AWS Design Management

- The design relies on AWS managed cloud services, including S3, Amplify, CloudFront, DynamoDB, and Lambda, to reduce the requirement for in-house system management and storage. AWS's security and monitoring services help company owners focus on other areas of their operations. Using managed cloud services improves scalability, dependability, and cost-effectiveness, alleviating infrastructure management challenges.

4.3.2 Scalability

- The proposed architecture outlined above is designed to accommodate the anticipated growth of the website on a semi-annual basis. To address this need, a serverless and event-driven solution has been implemented. This involves leveraging various managed cloud services, such as AWS Lambda functions, which can automatically scale up or down based on incoming traffic. Additionally, AWS S3 is used for storing static web content, while AWS DynamoDB serves as a persistent storage solution, ensuring that the web application can handle a high volume of requests without any performance degradation.

- Furthermore, the event-driven nature of the architecture, facilitated by AWS SNS, AWS SQS, and AWS Lambda, ensures that the components of the application are loosely coupled. This decoupling enhances scalability and reliability within the architecture. As a result, the system can effectively handle both anticipated spikes and declines in traffic without the need for on-premises system administration.

4.3.3 Database Optimization

- In contrast to the previous architecture where AWS RDS was utilized, the proposed architecture for the business now incorporates DynamoDB, a fully managed NoSQL database service. By leveraging DynamoDB, the system can automatically scale its resources based on fluctuations in website visits, allowing for seamless scalability and adaptability. Additionally, DynamoDB's serverless nature offers a cost-effective solution, as it charges based on the amount of data stored and the consumed throughput specific to the application's needs.
- The inclusion of data replication across multiple availability zones in DynamoDB ensures optimal user experience and provides ease of operation for the business owner. In contrast, AWS RDS requires manual scaling and resource maintenance, which introduces additional complexity and overhead. By transitioning to DynamoDB, the architecture gains enhanced scalability, cost efficiency, and simplified resource management.
- Overall, the decision to use DynamoDB over RDS brings numerous advantages, including cost savings, scalability, and robust availability, making it a compelling choice for our database needs within the web application architecture.

4.3.4 Cross-Region Performance

- With the increasing popularity and reliability of our website both within Australia and offshore, it is imperative to cater to the user experience of our offshore customers as well. To address this requirement, the proposed architecture incorporates AWS CloudFront and AWS Amplify, which can significantly enhance cross-region performance and meet our business needs.
- By leveraging AWS CloudFront, a content delivery network (CDN), we can cache and deliver content from the nearest location to the user. This approach minimizes latency and ensures faster content delivery, resulting in an improved user experience. Additionally, AWS Amplify seamlessly integrates with AWS CloudFront, automatically distributing static assets and content through CloudFront's network. This integration streamlines the process and ensures efficient content delivery across regions.
- The utilization of these services guarantees that our web application can handle cross-region traffic without any issues, leading to enhanced user satisfaction, improved business outcomes, and a competitive advantage over our rivals. Ultimately, the combination of AWS CloudFront and AWS Amplify enables us to deliver our content quickly and efficiently to offshore customers, solidifying our position in the market.

4.3.5 Advancements and Enhancements in Media Processing Techniques

- To accommodate the growing size and feature set of our website while maintaining a competitive edge and visitor retention rate, the proposed architecture focuses on optimizing existing features and revolutionizing the media processing workflow.
- The architecture employs a fanout pattern, which is a distributed computing approach designed to efficiently process large volumes of data in parallel. This enables media processing tasks to be distributed across multiple processing nodes, allowing the architecture to scale dynamically based on demand.
- The media processing procedure itself follows a serverless and event-driven approach, leveraging AWS Lambda functions for

automated processing of media files upon upload. This ensures that media files are processed automatically without the need for manual intervention. The serverless nature of this approach enables seamless scalability, as the system can easily handle fluctuations in workload based on spikes and declines in demand.

- Additionally, the procedure is designed to be decoupled, ensuring reusability and accommodating future feature enhancements. This means that new business requirements can be addressed by adding new features and additional services to the procedure without causing interference with existing functions.
- Overall, the proposed architecture optimizes media processing by leveraging a fanout pattern, serverless/event-driven approach, and decoupled design, providing scalability, automation, and flexibility for future feature evolution.

5. Conclusion

- In summary, adopting a serverless/event-driven architecture on AWS offers numerous advantages for businesses that require a scalable and cost-effective solution. Leveraging AWS services such as AWS Lambda, Amazon SNS, and Amazon SQS enables the system to respond to real-time events automatically, eliminating the need for manual management. This allows business owners to allocate more resources towards delivering value to their customers, rather than focusing on infrastructure management. The serverless/event-driven approach on AWS optimizes efficiency, scalability, and cost-effectiveness, ultimately enhancing business productivity and customer satisfaction.

6. References

- *Serverless Land. Retrieved from* https://serverlessland.com/eventbridge *(Accessed on April 2, 2024)*
- *Amazon Elastic Transcoder. Retrieved from* https://aws.amazon.com/elastictranscoder/ *(Accessed on April 2, 2024)*
- *AWS Shield. Retrieved from* https://aws.amazon.com/shield/ *(Accessed on April 2, 2024)*

- *Serverless/Event-Driven Architecture Sample with Best Decoupling Practices. Retrieved from* https://www.linkedin.com/pulse/serverlessevent-driven-architecture-sample-best-decoupling-reddy *(Accessed on April 2, 2024)*
- *"Serverless Three-Tier Architecture: A Comprehensive Guide" by Slobodan Stojanović. Retrieved from* https://medium.com/better-programming/serverless-three-tier-architecture-a-comprehensive-guide-6b1b7d9f3d33 *(Accessed on April 2, 2024)*
- *Mogiligundla, B.K.R. Serverless/Event-Driven Architecture Sample with Best Decoupling Practices. LinkedIn. Retrieved from https://www.linkedin.com/pulse/serverlessevent-driven-architecture-sample-best-decoupling-reddy/ (Accessed on January 4, 2022)*