Swinburne University of Technology – HCM Campus

COS20019 - Cloud Computing Architecture

Assignment 3

Nguyen Ngo Thanh Long (103803053)	Tran Thanh Minh (103809048)
Faculty of Computer Science	Faculty of Computer Science
Swinburne University of Technology	Swinburne University of Technology

Abstract

In the previous phases, the cloud system for photoalbum web application has been completed for current operation. Nevertheless, it is found that the demand for this web app will increase in the future and the current system design is not capable of coping with it. Therefore, this paper proposes a more advanced cloud architecture design for photoalbum application that incorporates several new AWS services. The design ensures extended operation for the website but still meets basic design criteria, including performance excellence, scalability, reliability, security, and cost.

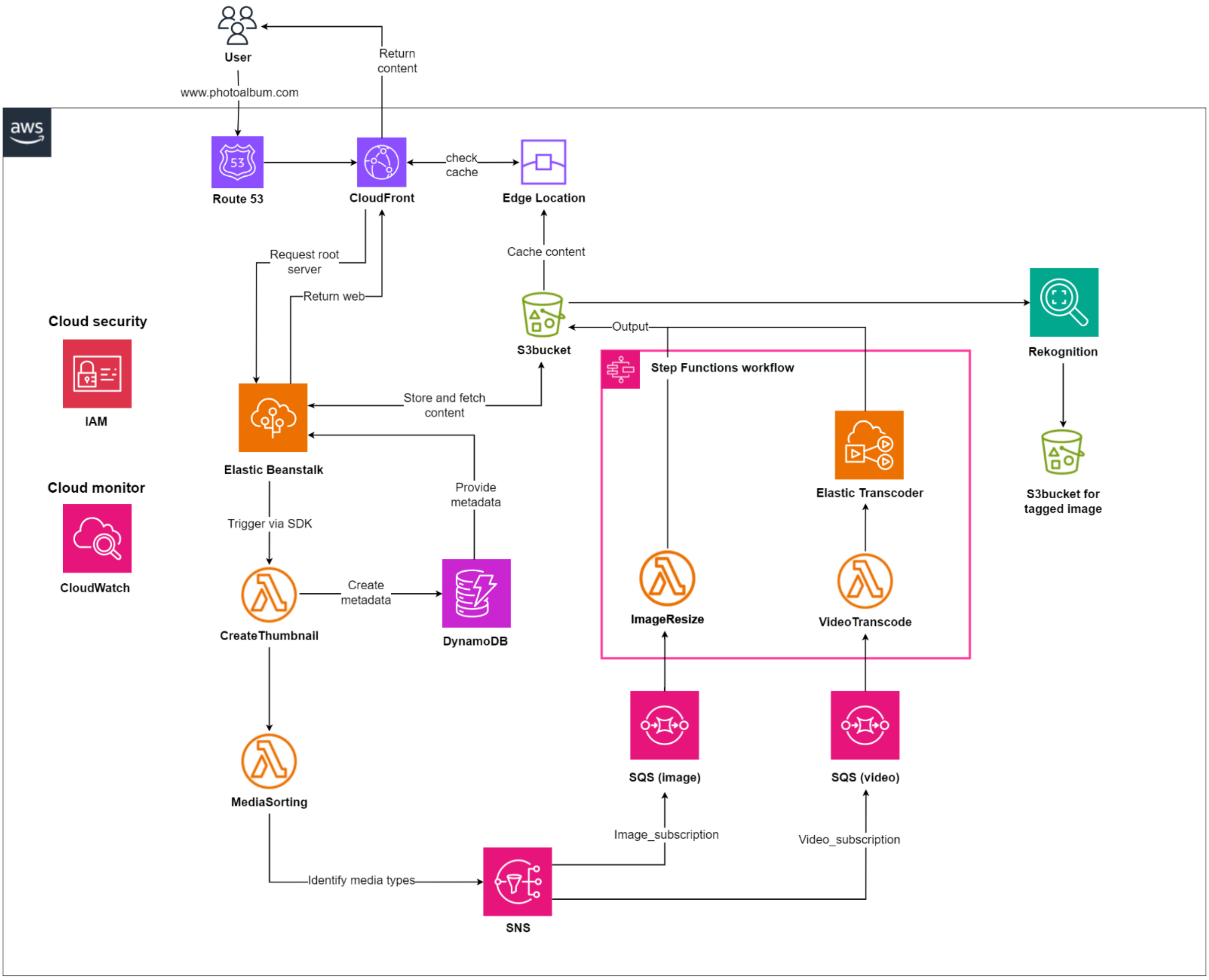
Introduction

Photoalbum web application is a service that allows users to upload and store their photos. As the demand for the application grows, the current system cannot meet the increased load. To ensure the continued availability and performance, a new cloud architecture design is proposed. The new architecture is designed to be more scalable, and higher availability than the current system. It is also designed to be more cost-effective. This paper proposes architecture design associated with UML diagram and table present the total cost. This new architecture will be designed as close as possible to the business scenario to handle the demand in the long term.

Business Scenario

1. Using managed cloud services to minimize the need for in-house systems administration.
2. Cope with the growth that will double every 6 months continuously for the next 2 or 3 years at least.
3. Using cost-effective and fast to run database.
4. Adopt a serverless/event-driven solution.
5. Global response times need to be improved.
6. Expected the system will be extended to handle video media in the future.
7. Automatically produced various versions of uploaded media.
8. Architecture for process media should be extensible.
9. Architecture should be designed to not become overloaded and effectively decoupled.

Architectural Design



1. Three-tier architecture

- One of the most well-known architectural design patterns for a cloud system is the 3-tier architecture. The new cloud architecture for photoalbum is designed based on this pattern which comprise of 3 main tiers:

+ **Presentation tier**: the upper tier that directly interacts with the users. In this case of photoalbum, it is the web page displaying album of photos and receive users' media uploads.

+ **Application tier**: the middle tier that processes photoalbum major tasks, which are uploading media files to storage, recording their metadata into database and produce a resized version of them.

+ **Data tier**: contains the storage and database which store media files and their metadata, respectively.

2. Presentation Tier

- Previously, the photoalbum web app was hosted on a manually configured network which required a lot of overhead and management. To simplify web deployment but still maintain reliability and availability, Elastic Beanstalk is chosen, which is an AWS service capable of deploying web applications with reduced structure management and provision. Especially, the 2 most crucial microservice that main high availability and scalability of the system which are Elastic Load Balancing and Auto-Scaling Group can be automated and customized if needed, allowing developers to put more effort into writing code.

- In addition, the company has learnt that the photoalbum application has a wide range of users around the world. However, other regions apart from Australia have suffered from low response time. The best solution for this issue is to integrate AWS CloudFront into the system. AWS CloudFront is a CDN (Content Delivery Network) service that delivers static and dynamic web content to users at low latency. It can achieve this by the appearance of more than 400 edge locations installed in various regions around the world. Particularly, web content can be cache in a specific edge location so that when user make a request for the web, CloudFront checks the nearest edge location for the cached content and deliver it to the user immediately. If the content has not been at the nearest edge location yet, CloudFront directs request to the root server; as the web is sent to user, CloudFront will store it in the edge location for the next time.

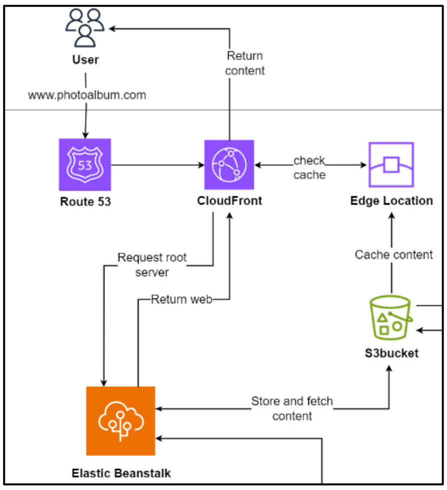


Figure 1: Presentation tier flow

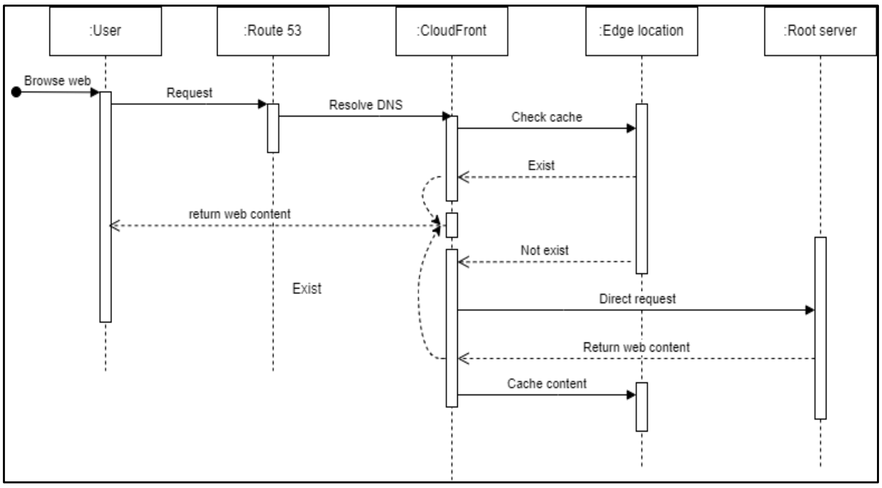


Figure 2: UML sequence diagram illustrates presentation tier workflow.

3. Application tier

- In the new architectural design, the web app functions of uploading media files and their metadata remain the same as the old design: the application uploads media file to S3 bucket and then forward metadata, including URL of the media file, to CreateThumbnail lambda function via AWS SDK/API call, which then insert them into database. The most notable change resides in the media-transforming function.

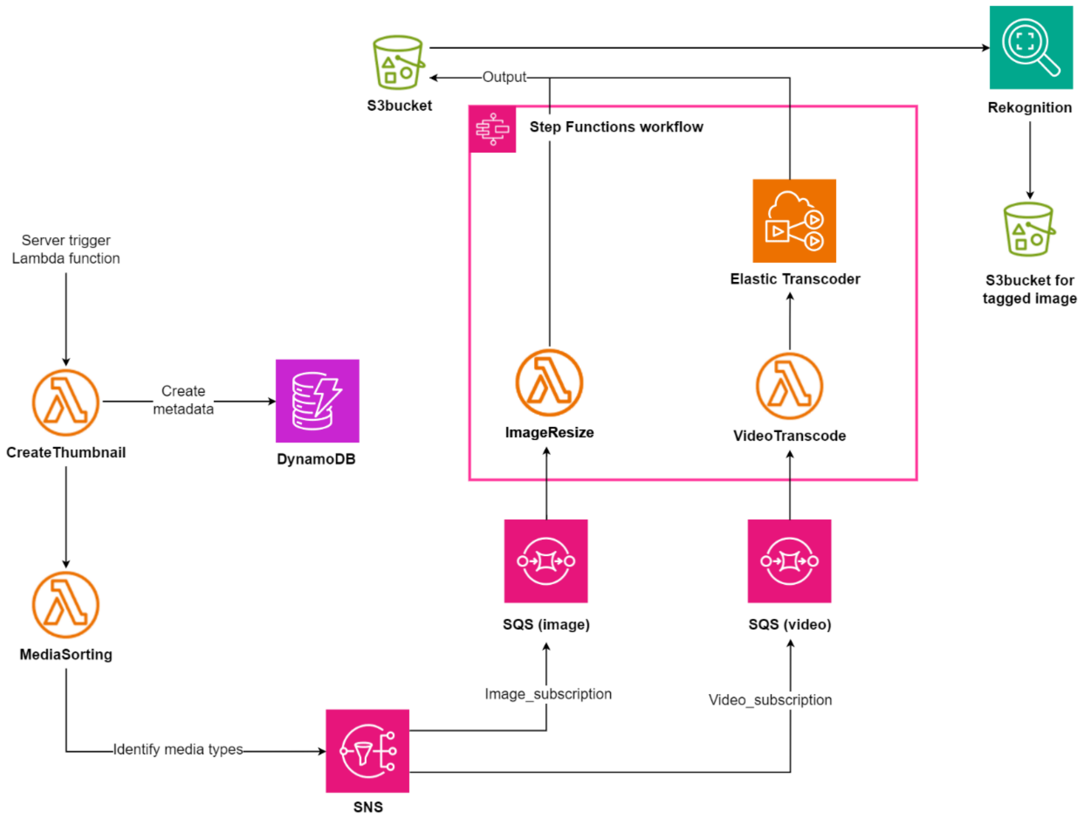


Figure 3: Media-transforming workflow

- The company now requires the application to not only reprocess images, but also videos. Therefore, several new AWS services must be added to the architecture to meet this requirement.

3.1. Decoupled architecture

- Since there are multiple workstreams in the new design, it must be properly **decoupled** to prevent single-point of failure, where issue in one workstream also interrupt other workstreams. To achieve this, Simple Notification Service (SNS) and Simple Queue Service (SQS) are the 2 most appropriate services for decoupling.

- In this architectural design, SNS and SQS are deployed based on fanout architectural design pattern. The main idea of this design pattern is to allocate incoming events to different downstream processes. Specifically, here is how the system work:

+ MediaSorting lambda function adds S3 URL of the file and a key that identifies its media type to the message and sends it to SNS.

+ A topic is created in SNS which contain 2 subscriptions, each of which is linked to one SQS endpoint and has filtering policy to only filter out image messages or video messages. Each type of message is sent to a separate SQS queue and waits to be polled by the corresponding lambda function.

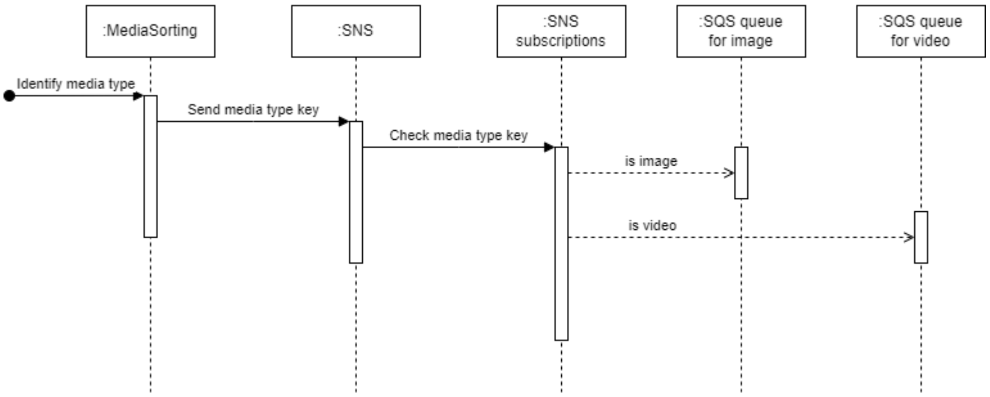


Figure 4: UML diagram for fanout architecture

3.2. Media-transforming procedure

- Each media-processing stream will poll the corresponding SQS queue for message and conduct media file reprocessing. To build up workflow for each process, AWS Steps Function Service is used, which supports automating and visualizing workflow.

+ The workflow of processing image only comprises of one Lambda function that resizes image. Nevertheless, it is still implemented via Steps Function to ensure extensibility, where the workflow might integrate AWS Rekognition to identify tags in the photos in the future.

+ For video-processing workflow, the task of reformatting video is given to AWS Elastic Transcoder. There are 2 main steps: VideoTranscoder lambda function downloads the video from S3 bucket, the video is then transferred to an Elastic Transcoder job for transcoding.

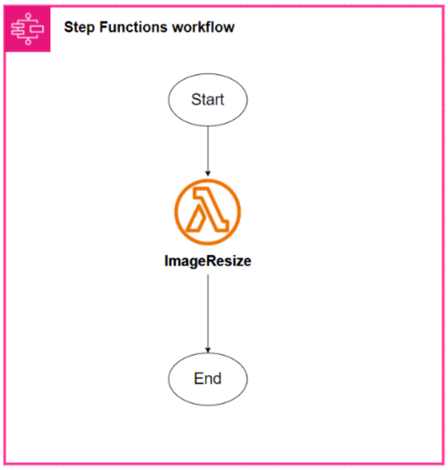


Figure 4: Image-processing workflow

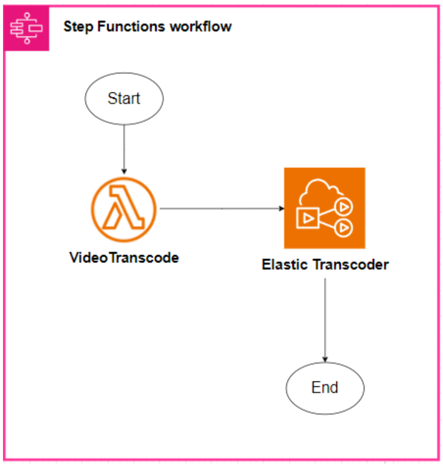


Figure 5: Video-processing workflow

- The biggest benefit of AWS Step Functions is extensibility. Providing that the company wants to enhance the workflows with more features in the future, Steps Functions can simplify this process by reducing the amount of code a developer has to write.

3.3. Tag identifying in photos.

- It is desirable that the function of recognizing tags with AI in the photos might be added in the future. AWS Rekognition should be the perfect service for this task. Nevertheless, since this proposal has not been official yet, a separate workflow for AWS Rekognition is built for testing to prevent any undesirable extra cost on storage and interruption with the current image-processing workflow due to the automation of the stream.

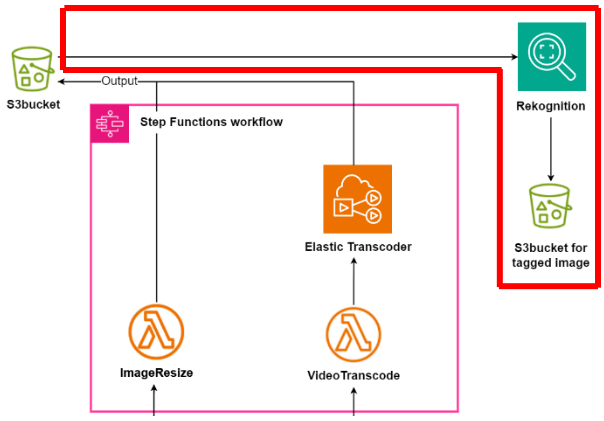


Figure 5: Separate workflow for photo tag identification. AWS Rekognition retrieves processed image from S3 bucket instead of being integrated into the automated photo-reprocessing

4. Data tier

- The photoalbum web application has been using AWS Relational Database Service to store file metadata. However, given such a simple table structure of the web app, RDS is cost-ineffective. Instead, database solution will be changed into AWS DynamoDB, which is a NoSQL database service. Different from relational database, NoSQL database is capable of scaling horizontally (increment in computing machine quantity), hence suitable for a system having a simple database structure but expanding rapidly. As a web application predicted to double every 6 months in the coming years, DynamoDB should be the best solution.

- For media file storage, AWS S3 is still a reliable storage solution and thus continues to be utilized in the new architectural design.

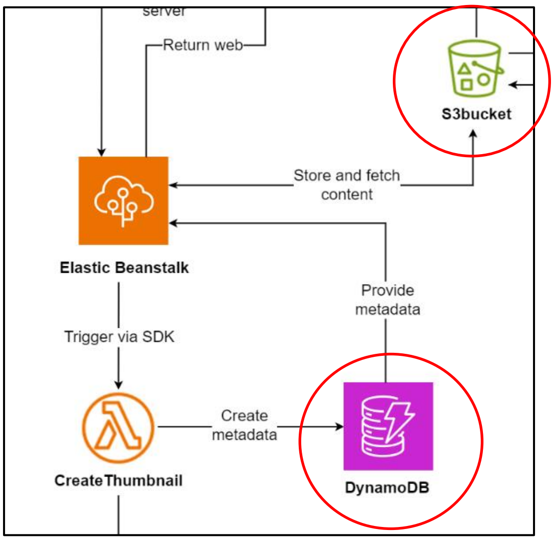


Figure 6: DynamoDB and S3 are the 2 major components of data tier.

II. Design Rationale

1. Business scenario fulfillment

❖ **Infrastructure**

- With the use of AWS S3 and new applied services like AWS DynamoDB, AWS Lambda, AWS CloudFront, the new architecture can minimize the need for in-house system management. Moreover, AWS IAM has helped to enhance the security with control access (authentication and authorization). In addition, AWS CloudWatch can also enhance security and monitor the system's performance. With additional services applied to the business, this architecture brings the ease of cloud management with huge improvements on security, scalability, reliability. AWS also provides a service calculator so that the business can estimate the infrastructure cost monthly.

❖ **Scalable architecture**

- With the applied services like AWS Lambda, DynamoDB, AWS CloudFront, AWS Route 53, AWS Elastic Beanstalk, AWS SNS and SQS, the system can scale up and scale down based on the traffic, storage and adapt to the demand. This design has adopted the event-driven solution with the use of SNS, SQS and Lambda so that the components in the architecture are effectively decoupled, which results in scalability and reliability. When this design receives a large amount of traffic, Lambda service, Elastic Beanstalk, Route 53 can scale up to handle the traffic with low latency and unchanged performance.

❖ **Multi-regions performance**

- Photoalbum application has had wide uptake around the world, but the performance is decreasing. This new design is supported with CloudFront, so the dynamic content is cached in the edge locations to ensure the low latency but keep the performance unchanged. Furthermore, with the support of Route 53, the traffic can be routed to the nearest server which can lead to better user experience with faster loading web applications.

❖ **Multimedia processing and optimization**

- As the demand for use the application, the business wants to implement features for optimizing input media files and automatically convert them into different versions for different devices. With the support of Lambda function, every time a media file is uploaded to the web application, the Lambda function will trigger and process the file. This process is built on AWS Steps Function and decoupled with SNS - SQS so that any additional features can be easily integrated in the future.

❖ **Database optimization and performance**

- Compared with the previous design, the new one has switched the storage service from RDS to DynamoDB. With the power of NoSQL database service, it provides scalable storage based on the demand which can cope with the growth in the next 2 or 3 years. DynamoDB charges for storage that the business uses and the number of requests per month. DynamoDB provides a fast to go service which can enhance the user experience from offshore places.

2. Design criteria fulfillment

Category	Criteria	Solution
Security	- Control authorized access and authenticated access	- AWS IAM for authenticating and authorizing the access to the application. - AWS CloudWatch service can help to monitor and detect security threats, malware infections. - AWS S3 can encrypt data, access control, and detailed logging.
Reliability	- Highly available - Can adapt to changes in demand. - Recover from failures. - Backing up data frequently	- AWS S3 can automatically scale up and scale down based on the demand, durable and replicated data. - AWS CloudFront can scale up and scale down based on the traffic to reduce latency and improve performance in different regions and highly resilient to failures, with multiple copies of your content stored in distinct locations. - AWS Elastic Beanstalk can scale up and scale down based on real-time demand. - AWS DynamoDB is a scalable NoSQL service which is durable and replicates data across regions. - AWS CloudWatch can monitor the failure.
Performance	- Quick processing time for uploading media files. - Low latency in fatigue circumstance	- AWS CloudFront caches dynamic content in edge locations around the world. - AWS Route 53 can route the users to the closest and most performant server. - AWS CloudWatch can alarm if the metrics are out of safe bounds and cause automated actions to address issues. - AWS Elastic Transcoder can convert media files into different formats and resolutions for more accessible to uses with different devices and connection speeds.
Scalability	- Can handle large amounts of traffic. - Scalable to adapt to demand	- AWS CloudFront is highly scalable and handles millions of requests per second with techniques of caching and load balancing. - AWS Route 53 can handle millions of requests per second and route the requests from user to the nearest name server.

		- AWS SNS and SQS are scalable messaging services that can handle millions of messages per second using techniques such as load balancing and failover. - AWS Lambda can also handle millions of requests for the application and process it by using techniques of batching and caching. - DynamoDB is highly scalable pay as you go and has ability of scalability by using techniques of caching. - AWS Elastic Beanstalk can scale up and down based on the received traffic.

Table 1: Design criteria fulfillment

❖ **Costs**

- The new architecture is designed based on the information provided and business scenario to adapt to the growth that the demand for the application has been doubling for every 6 months and continue to grow with this trend in the next 2 or 3 years. There are assumptions for calculating the costs for 1 TB storage. The total cost of operation for the new AWS architecture for 1TB is about $2013.964 (refer to budget at **page 11** for more info, where a table describes in detail each service and its cost).

3. Alternative design

3.1. Web application deployment

- Another method to host the photoalbum web app instead of Elastic Beanstalk is to manually configure a network consisting of multiple public and private subnets spanning across Availability Zones, NAT gateways for private subnets' internet access, Elastic Load Balancer for load balancing and Auto Scaling Group for flexible system expansion with AWS EC2 service.

- This design allows developers to have deeper management in the system since all configurations must be done manually from the beginning, which is suitable for a complex web application requiring insightful control in each component. However, since the configuring phases are manual, the system is noticeably error prone. It is important to carefully configure and test the system iteratively before deployment to avoid as many errors as possible, which increases workload for developers. Therefore, to ensure performance excellence and reliability but still maintain decent workload for a less-complex web application like photoalbum, Elastic Beanstalk is highly recommended.

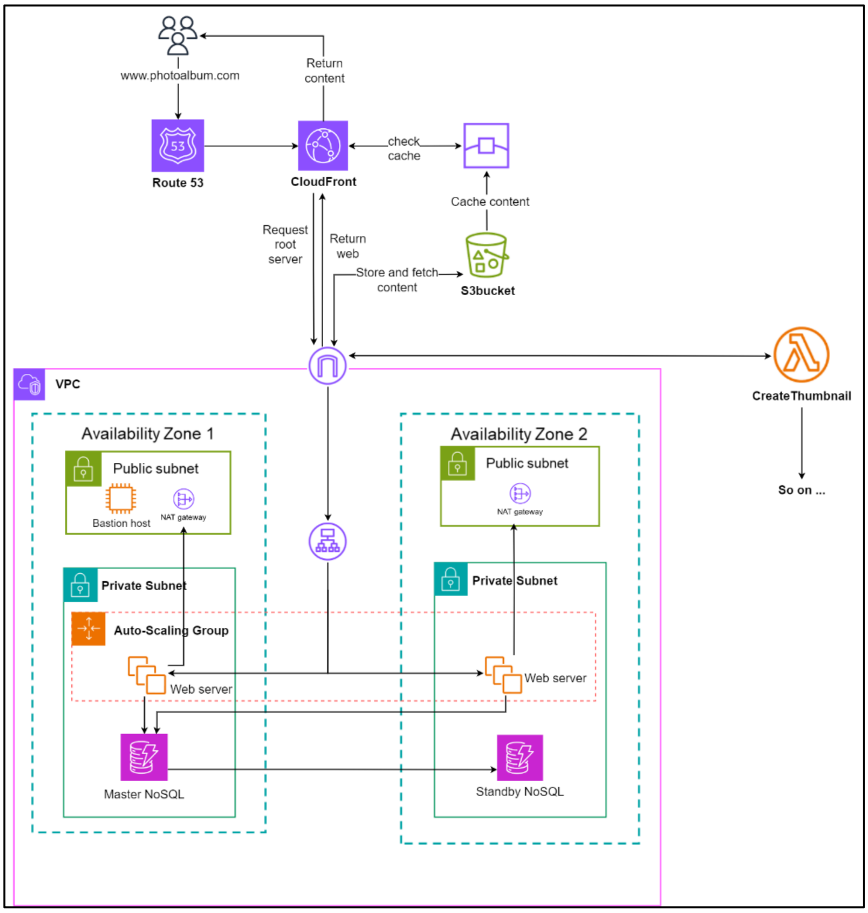


Figure 7: Manually configured network for web app deployment. After the CreateThumbnail function, the workflow remains the same as the proposed design.

3.2. Photo tag identifying workflow.

- The company stated that there might be the feature of identifying tags in photos using AI in the future. Providing that tag identification will be an integral part of the system, AWS Rekognition can be integrated as part of the image-processing workflow. After photos are resized, they are automatically transferred to Rekognition to identify tags and finally, stored in S3 bucket.

- However, being an unofficial feature, tag identification is built in an isolated workflow for initial experiment in the proposed design to eliminate any interference with the mainstream, maintaining reliability of the system and avoid any unwanted charge on Rekognition and storage.

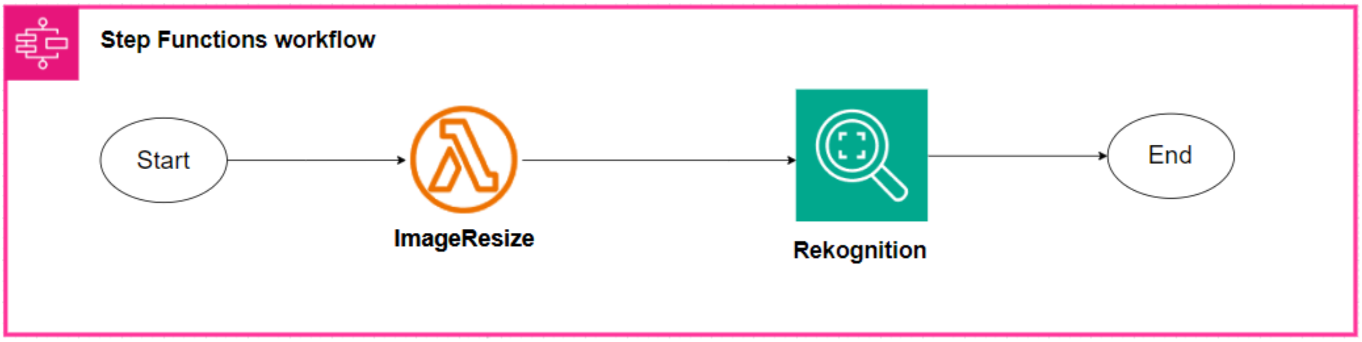


Figure 8: Alternative workflow of image-processing with Rekognition

COS20019 - Cloud Computing Architecture

Budget

Assumption costs for 1 TB:

- Each Lambda function is associated with 4 custom metrics (Total 4 lambda functions).
- Each media file size is about 200KB.
- 1,000,000 queries each month.
- 2 HTTPS request.
- Videos are in 720p quality, and the length is equal to modern MV (5 mins)
- Assume 1,000,000 images are processed for AI.
- AI is trained for 2.5 hours

Some calculations for using in the table below:

Storage = 1TB = 1,000 x 1024^2

Request = 1,000,000

Service	**Cost**	**Total Cost**
Route 53	- Host zones: $0.50/month - DNS query: $0.40/million query/month => Storage x $0.40 / 1,000,000 = $2.597152 => ~ $2.6	$2.6
CloudFront	- Data transfer: $0.17/GB - Request: $0.0125/request => Storage / 200 / 10,000 * 2 * $0.0125 = $26.2144 => ~ $26.2	$26.2
Elastic Beanstalk	- Instance cost: $0.0115 * 6 * 24 * 30 = $50.4 - Traffic cost: $0.000016 * 1,000,000 = $16 => $50.4 + $16 = 66.4	$66.4
S3	- Storage: $0.023/GB/month => Storage * $0.023 = $23.55 - Data Transfer (Get): $0.000005/GB/month - Data Transfer (Get): $0.0000004/GB/month => $0.0000054 * 1,000,000 = $5.4 => $23.55 + $5.4 = 28.95	$28.95
Simple Queue Service (SQS)	- Data transfer (Out): $0.09/GB/month => $0.09 * 1,000 = $90	$90

SNS	- The end point of this design is the SQS, so there is no cost for this service but if it is notification is delivered to an email or mobile, then there will be charged fee.	$0
Elastic Transcoder	- First 50TB: $0.0125/GB => $0.0125 * 1,000 = $12.5	$12.5
CloudWatch	- Metric: $0.3/metric => 4 * 4 * 0.03 = $4.8	$4.8
Lambda	- Memory: $0.00002 per 1ms - Request: $0.0000002/100ms => storage / 200 / 1,000,000 x 4 * $0.2 = $8.96 => $0.0000002 * 128 * 1000 * 4 = $1.024 => 8.96 + 0.256 = $9.984	$9.984
Rekognition	- Data transfer: $0.17/GB => 2.5 (training hours) * 60 (mins) * $0.01666667/min = $2.5 => $0.06666667/min * (10 (transactions) * 60 (mins)) = $0.000111 => 1,000,000 (images) * $0.000111 = $111 => $111 + $2.5 = $113.5	$113.5
DynamoDB	- Storage: $0.25/GB/month => 25 + 999 * 0.25 = $249.75 - Read: $0.25/million request => Storage / 1,000,000 / 4 * 0.25 = $67.108864 - Write: $1.25/million request => Storage / 1,000,000 * 1.25 = $1342.17728 => $249.75 + $67.108864 + $1342.17728 = $1659.035144 => ~ $1659.04	$1659.04
Total Cost		$2013.964

Table 2: Total cost of operation for 1TB media upload