Swinburne University of Technology

COS20019

Cloud Computing Architecture

Assignment 3

Team 191

Name	*ID*
Dac Tung Duong Nguyen	104357292
Ansh Sehgal	104172848

Abstract

In previous phases, the cloud infrastructure supporting the photo album web application was successfully implemented and is presently in use. Nonetheless, the foreseeing increase in demand for this web application in the future poses a challenge for the current system, which lacks the capacity to handle this surge in user activity. To address this issue, this paper proposes an advanced cloud architecture design for the photo album application. This updated design integrates multiple new AWS services, ensuring continuous website functionality while upholding crucial design criteria, including superior performance, scalability, reliability, security, and cost-effectiveness.

I. Introduction

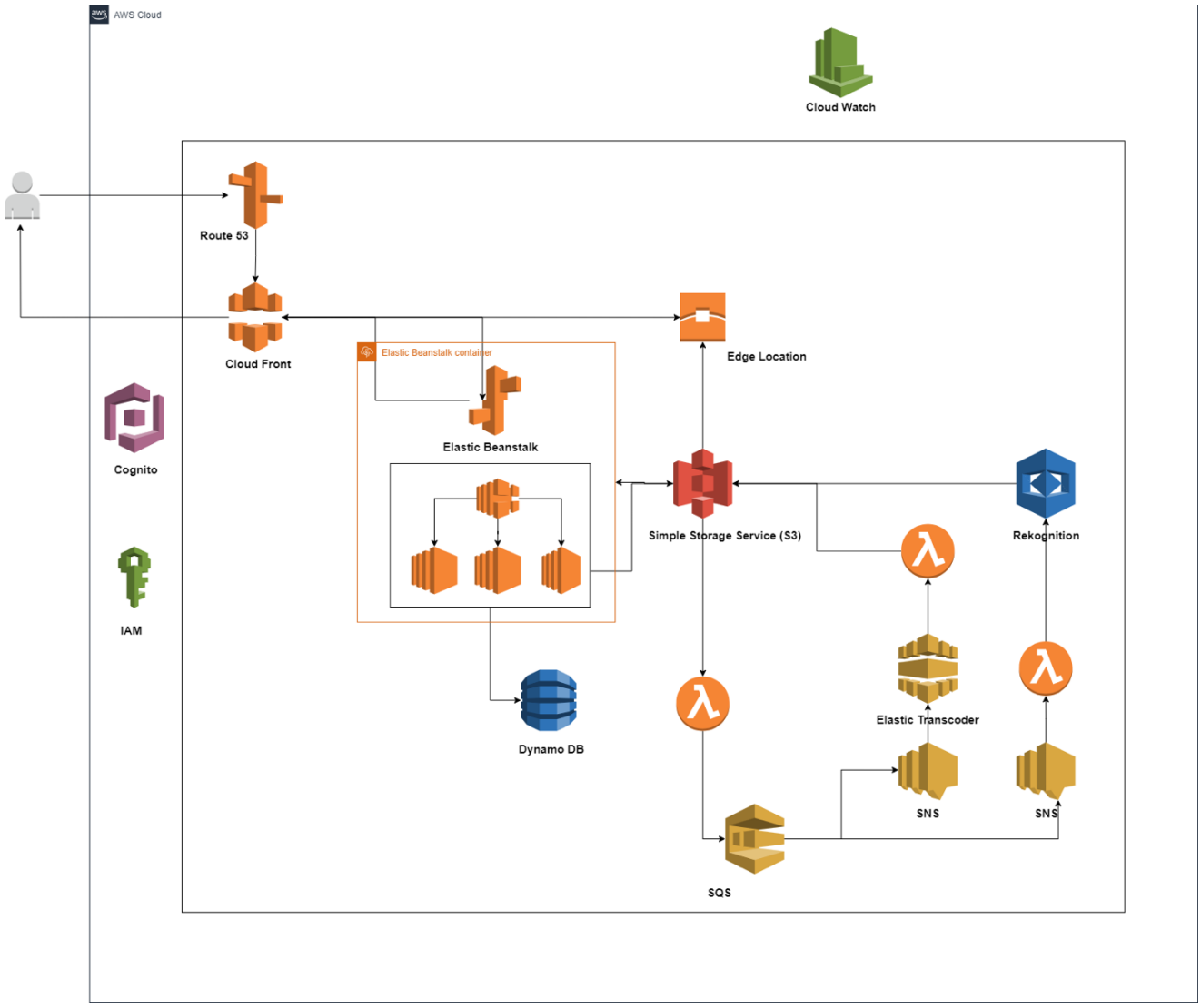
The photo album web application provides users with a platform to upload and store their photos. With the rising demand, the existing system is unable to handle the increased workload effectively. To guarantee consistent availability and performance, a new cloud architecture design is suggested. This fresh approach focuses on enhanced scalability and higher availability compared to the current system. Moreover, it is optimized for cost-effectiveness. The paper outlines the proposed architecture using UML diagrams and cost tables, aligning it closely with real business needs to ensure long-term demand management.

II. Business Scenario

1. Employing managed cloud services to reduce the necessity for internal system administration.

1. Addressing the anticipated growth, expected to double every 6 months for the next 2 to 3 years, continuously.
2. Utilizing a cost-effective and efficient database system for swift operations.
3. Embracing a serverless/event-driven approach for streamlined processes.
4. Enhancing global response times significantly.
5. Preparing the system for future scalability by accommodating video media handling.
6. Automatically generating different versions of the uploaded media files.
7. Ensuring that the architecture for processing media is adaptable and expandable.
8. Designing the architecture to prevent overloading and ensuring effective decoupling of components.

III. Architectural Diagram



Three-Tier Architecture Overview

The three-tier architecture is a well-established design pattern in cloud computing, offering a structured approach to building scalable and maintainable applications. This architecture divides applications into three interconnected layers, each optimized for specific tasks:

1. Presentation Tier (Tier 1):

This layer serves as the application's front-end, directly interacting with end-users.

User Interaction: It encompasses various user interfaces, from web browsers to mobile apps, ensuring a consistent user experience across devices.

AWS Elastic Beanstalk: Our system leverages Elastic Beanstalk, a fully managed service by AWS, to deploy and manage web applications. It abstracts the infrastructure complexities, allowing developers to focus on code, while AWS handles deployment, capacity provisioning, load balancing, and auto-scaling.

AWS CloudFront: To cater to a global audience and ensure low-latency access, we integrate CloudFront. This CDN service caches web content at strategically located edge locations, ensuring users get data from the nearest point, reducing latency and improving load speeds.

2. Application Logic Tier (Tier 2):

Acting as the application's backbone, this layer processes data, enforces business rules, and coordinates tasks.

Media Processing: Upon media file uploads, our system categorizes and processes them based on type:

Videos: Videos are channeled to the AWS Elastic Transcoder through an SNS topic, which then distributes the tasks to specific SQS queues. Elastic Transcoder handles the video transformation, ensuring compatibility across devices and formats.

Images: Images are routed to a dedicated Lambda function via a separate SNS topic and SQS queue. Once processed, AWS Rekognition further analyzes these images, utilizing AI to identify tags, objects, and even detect sentiments.

3. Data Storage Tier (Tier 3):

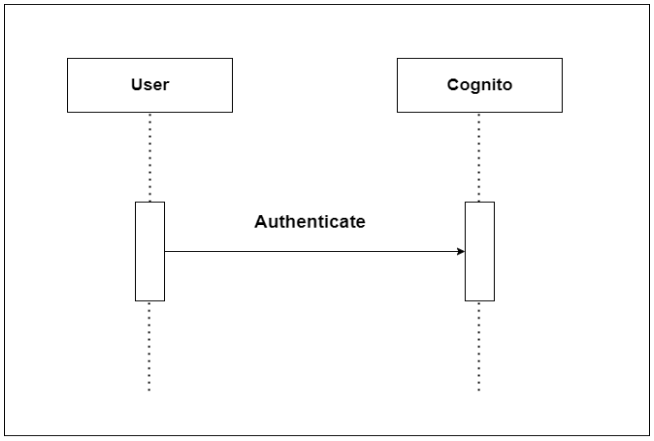
This layer is the data repository, ensuring secure storage, quick retrieval, and data integrity.

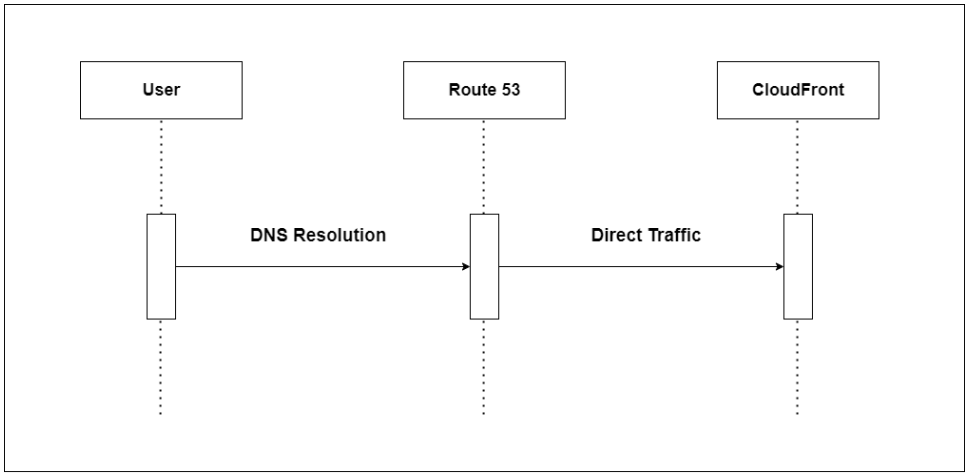
AWS S3: S3 (Simple Storage Service) is our primary choice for storing media files. Known for its durability and scalability, S3 ensures that our media files are stored securely and can be retrieved quickly when needed.

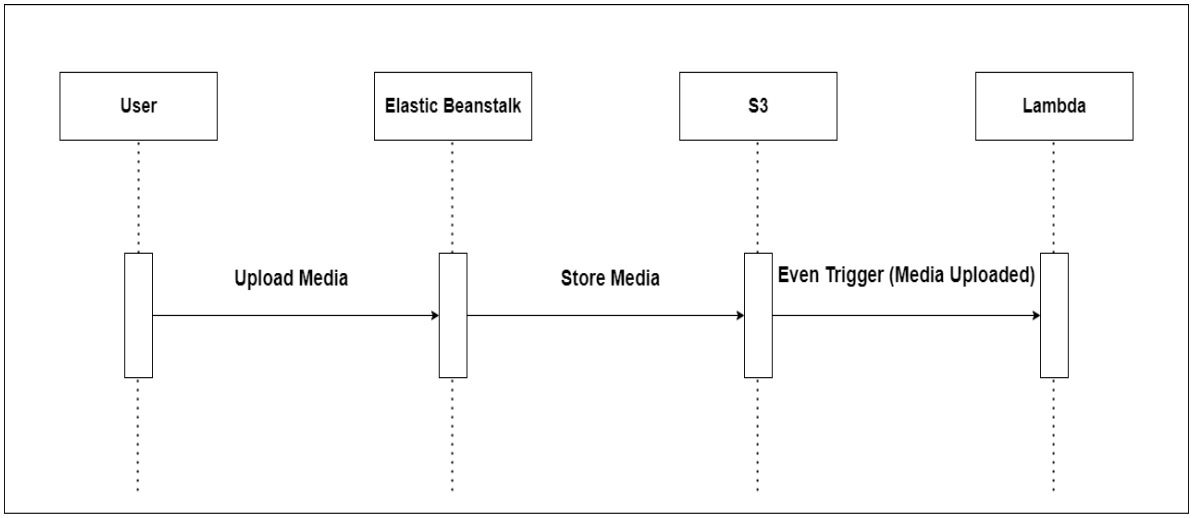
AWS DynamoDB: For metadata storage, we transitioned from the traditional RDS (Relational Database Service) to DynamoDB. As a NoSQL database service, DynamoDB provides the flexibility to store structured and semi-structured data. Its ability to scale horizontally makes it an ideal choice for applications expecting rapid growth, especially when the database structure is relatively straightforward.

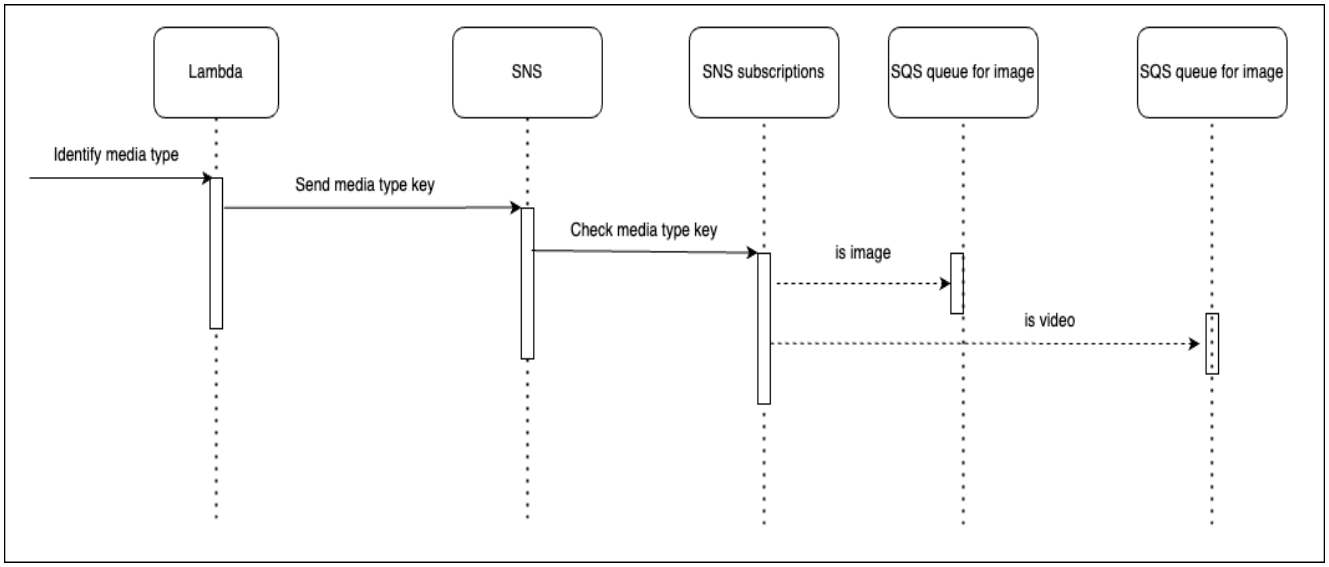
IV. UML Diagram

Media Upload :


User
Cognito
Authenticate


User
Route 53
CloudFront
DNS Resolution
Direct Traffic


User
Elastic Beanstalk
S3
Lambda
Upload Media
Store Media
Even Trigger (Media Uploaded)



V. Description of used AWS services

- CloudFront: Amazon CloudFront is a fast and efficient content delivery network (CDN) service offered by Amazon Web Services. It delivers web content, videos, and images quickly by caching them at various global locations, improving user experience and website performance.

- Lambda: AWS Lambda is a serverless computing service by Amazon Web Services. It executes code without managing servers, automatically scaling based on demand. Developers upload code, and Lambda runs it in response to events, like HTTP requests or data changes.

- Route53: Amazon Route 53 is Amazon Web Services' scalable Domain Name System (DNS) service. It translates user-friendly domain names into IP addresses, ensuring reliable and efficient internet traffic management.

- Elastic Beanstalk: AWS Elastic Beanstalk is a managed service for deploying applications. It handles deployment details like capacity provisioning and scaling, allowing developers to focus solely on writing code.

- DynamoDB: Amazon DynamoDB is a managed NoSQL database service by AWS, providing high-performance, scalable storage for applications with low-latency access to data.

- S3: Amazon S3, or Amazon Simple Storage Service, is a scalable cloud storage service provided by Amazon Web Services. It allows users to store and retrieve data, such as files, images, and videos, over the internet. S3 is widely used for backup, data archiving, and serving static assets for websites and applications.

- Edge location: Edge locations are global data centers used by Amazon Web Services to cache content closer to users. They reduce latency and improve the speed of delivering web content and applications.

- SNS: Amazon SNS is a managed messaging service by AWS. It sends notifications to various endpoints, simplifying communication between distributed systems and applications.
- SQS: Amazon SQS is a managed message queuing service by AWS, facilitating asynchronous communication between components of cloud applications for reliability and simplicity.

- Elastic Transcoder: AWS Elastic Transcoder is a scalable service that converts multimedia files into different formats, simplifying the process of streaming videos across devices.

- ReCognition: AWS Rekognition is an Amazon Web Services service that uses deep learning to analyse images and videos, identifying objects, people, text, scenes, and activities.

- IAM: IAM (Identity and Access Management) is an AWS service that controls access to resources by managing user identities and permissions securely.

- Cognito: Amazon Cognito is an AWS service that simplifies user authentication and management for web and mobile apps, ensuring secure and seamless user experiences.
- Could watch: Amazon CloudWatch is AWS's monitoring service for tracking metrics, logs, and setting alarms in real-time.

VI. Requirements and Justification

Justification:

1. Managed Cloud Services & AWS S3 for Media Storage:

Requirement: The company prefers managed cloud services to minimize in-house systems administration. Media should be stored in AWS S3.

Justification: AWS Elastic Beanstalk is used to abstract infrastructure complexities, allowing developers to focus on code. AWS S3 is utilized for storing photos and other media, providing a durable and scalable storage solution for the Photo Album application.

2. Scalability for Growing Demand:

Requirement: The company expects its demand to double every six months for the next 2-3 years and needs an architecture that can handle this growth.

Justification: AWS Elastic Beanstalk and CloudFront are employed to ensure scalability and rapid content delivery. Elastic Beanstalk, combined with Auto Scaling, ensures the application scales automatically based on demand. CloudFront, with its global edge locations, caters to the growing global user base.

3. Compute Capacity & EC2 Instances:

Requirement: The current system's compute capacity on t2.micro EC2 instances is regularly exceeding the 80% performance limit.

Justification: AWS Lambda is implemented as a serverless solution to address the performance issues, ensuring efficient processing of events without the need for manual server management.

4. Serverless/Event-Driven Solution:

Requirement: The company prefers a serverless/event-driven solution.

Justification: AWS Lambda, combined with SNS and SQS, provides the event-driven architecture for the system, processing media uploads and other tasks efficiently.

5. Database Solution:

Requirement: The relational database is slow and costly. The company seeks a more cost-effective option given the simple table structure.

Justification: AWS DynamoDB is employed as the primary database solution, offering fast performance and scalability suitable for the application's database structure.

6. Global Response Times:

Requirement: Improve response times globally.

Justification: AWS CloudFront is utilized to cache content at multiple edge locations worldwide, ensuring reduced latency and improved load speeds for users globally.

7. Handling Video Media:

Requirement: The system should be able to handle video media in the future.

Justification: AWS Elastic Transcoder is used to handle video transcoding, ensuring videos are optimized for various devices and formats.

8. Media Processing:

Requirement: The company wants automatic production of various media versions (thumbnails, low-res versions, video transcoding) upon upload to S3. The architecture

should be extensible, decoupled, and able to run processing services on suitable platforms.

Justification: AWS Lambda is triggered by S3 events to process images, creating thumbnails or low-res versions.

AWS Elastic Transcoder handles video transcoding tasks.

AWS Rekognition is integrated for AI-based tag identification in photos.

SNS and SQS are used to ensure decoupling, allowing efficient media processing without overloading the application. Multiple 'worker' nodes, like Lambda functions or EC2 instances, process transformation jobs from the queue, ensuring scalability and specialization.

VII. Alternative Solution

1. Virtual machines vs. Containers vs. Serverless computing

Used: Serverless computing (Lambda)

Pros: With AWS Lambda, you only pay for the execution time of the function, eliminating the need to provision or manage servers. It scales automatically, making it suitable for unpredictable workloads.

Alternative: Virtual machines (e.g., EC2) or Containers (e.g., ECS)

Why not the alternative? While EC2 and ECS are powerful, they require more infrastructure management and overhead. For workloads that require sporadic execution based on events, Lambda offers a cost-effective and scalable solution.

2. SQL vs. NoSQL database:

Used: NoSQL database (DynamoDB)

Pros: DynamoDB offers high performance, scalability, and is suitable for applications with varying loads and unpredictable traffic patterns.

Alternative: SQL database (e.g., RDS)

Why not the alternative? RDS requires more maintenance, such as backups, patching, and scaling. For workloads needing flexible schema design and rapid scaling, DynamoDB is a better fit.

3. Caching options:

Used: Content Delivery Network (CloudFront)

Pros: CloudFront caches content in edge locations, reducing latency and improving content delivery speed.

Alternative: In-memory cache (e.g., Elasticache)

Why not the alternative? While Elasticache can speed up database queries, for delivering static content to a global audience, CloudFront provides a more scalable and cost-effective solution.

4. Push vs. Pull message handling options to promote decoupling:

Used: Push (SNS) and Pull (SQS)

Pros: SNS allows for the distribution of messages to multiple subscribers, while SQS decouples services and ensures message delivery.

Alternative: Direct communication without decoupling.

Why not the alternative? Direct communication can lead to tight coupling of services, making the system less scalable and harder to maintain.

5. Number of tiers in the architecture, e.g., 3-tier vs 2-tier:

Used: Multi-tier architecture (as indicated by the usage of services like Elastic Beanstalk for application logic, DynamoDB for data storage, and CloudFront for content delivery).

Pros: Separation of concerns, scalability, and security.

Alternative: 2-tier architecture.

Why not the alternative? A 2-tier architecture could merge application logic and data storage, potentially leading to scalability and maintenance challenges.

6. Media Transformation:

Used: Elastic Transcoder for media conversion and Rekognition for image and video analysis.

Pros: Scalable, managed services that offload the complexity of media processing.

Alternative: Custom-built solutions on EC2.

Why not the alternative? Building custom solutions requires more development and maintenance efforts. AWS managed services simplify the process and are generally more cost-effective for such tasks.

VIII. Design Criteria

- **Performance, Scalability (extensibility, decoupling, etc.):**

  Serverless Architecture (Lambda): By utilizing AWS Lambda, the application ensures it only uses resources when needed. It scales automatically by running code in response to triggers, allowing the system to handle varying loads without

manual intervention. This choice promotes extensibility as new features can be added without major architectural changes.

Decoupling using Messaging Services (SNS & SQS): Amazon SNS and SQS promote decoupling of application components. This ensures that high loads on one component (like media processing) don't impact others, maintaining consistent performance. Decoupled systems are also easier to scale and extend.

Global Content Delivery (CloudFront & Edge Locations): With the use of CloudFront and its edge locations, content is delivered faster to global users, enhancing performance by reducing latency.

- **Reliability:**

  Distributed Storage (S3): Amazon S3 provides 99.999999999% (11 9's) of durability over a given year, ensuring data reliability and integrity.

  Managed Services (Elastic Beanstalk, DynamoDB): Using managed services like Elastic Beanstalk and DynamoDB means AWS handles the underlying infrastructure, ensuring high availability and fault tolerance.

  DNS Management (Route53): Route53 provides reliable and cost-effective domain name registration, ensuring that user requests are appropriately routed to your application.

- **Security:**

  Authentication and Authorization (IAM & Cognito): IAM allows fine-grained access control to resources in AWS, ensuring only authorized entities can access them. Cognito provides secure user sign-up, sign-in, and access control to web and mobile apps.

  Encrypted Storage (S3): Amazon S3 supports data encryption both at rest and in transit, ensuring data confidentiality and security.

Content Protection (CloudFront): CloudFront integrates with AWS Shield, AWS Web Application Firewall, and Route 53 to defend against multiple types of attacks including network and application layer DDoS attacks.

- **Cost:**

Fixed Expenses:

1. Route53:

Domain registration: $12/year.

Hosted zones: $0.50 per zone per month = $6/year.

2. IAM: No additional charge.

3. Cognito:

Free tier includes 50,000 monthly active users. Afterward, it's $0.0055 per MAU (Monthly Active User).

Variable Expenses:

1. AWS Lambda:

First 1 million requests per month are free.

$0.20 per 1 million requests thereafter.

2. S3 (Simple Storage Service):

Standard storage pricing: $0.023 per GB/month for the first 50 TB/month.

3. CloudFront:

Data transfer out to the internet (North America, Europe): First 10 TB/month = $0.085/GB.

4. DynamoDB:

On-demand pricing: $1.25 per million write request units and $0.25 per million read request units.

5. Elastic Transcoder:

Pricing is per minute of video transcoded and the video resolution chosen. For instance, SD video = $0.015/minute and HD video = $0.03/minute.

6. SNS:

First 1 million requests per month are free. Beyond that, $0.50 per 1 million SNS requests.

7. SQS:

First 1 million monthly requests are free.

$0.40 per 1 million requests beyond the free tier.

8. Rekognition:

Image analysis: First 1M images per month = $1 per 1,000 images.

9. Elastic Beanstalk:

Pricing is dependent on the underlying resources (e.g., EC2, RDS, etc.) used to run and store the app.

10. Edge Locations (via CloudFront):

The costs are integrated within CloudFront's data transfer rates.

11. CloudWatch:

Pricing varies depending on metrics, log events, and custom events. Basic monitoring metrics (default every 5 minutes) = $0.30 per metric/month.

Budget Summary (Estimated Monthly Costs):

- Route53: $1
- IAM: $0
- Cognito (if exceeding free tier): Depends on MAUs beyond 50,000
- Lambda (assuming significant usage beyond free tier): $0.20+
- S3: Varies by GB stored
- CloudFront: Varies by GB served
- DynamoDB: Based on read and write request units
- Elastic Transcoder: Depends on video minutes and quality
- SNS: $0.50+
- SQS: $0.40+
- Rekognition: Based on image analysis quantity
- Elastic Beanstalk: Based on underlying resources
- CloudWatch: Varies, estimated $0.30+ per metric

IX. Justification for selection of best solution based on the criteria defined:

According to us, the solution that we have proposed is the best design idea as compared to the alternative solutions.

Performance and Scalability:

AWS Lambda (Serverless Framework): This choice presents an architecture that's inherently designed to scale with demand. Unlike traditional systems where resources might be under-utilized or strained during peak times, Lambda adjusts in real-time. Its event-driven nature ensures that we have optimal performance without unnecessary costs.

Decoupling via SNS & SQS: These services are vital for ensuring our system's components operate independently. By segregating tasks (like image processing or notification sending) from the main application flow, we ensure that a surge in one function doesn't bottleneck others. Moreover, it simplifies scalability as individual components can be scaled based on their unique demands.

CloudFront & Edge Locations: To cater to our global user base, these services ensure that content reaches users via the shortest possible route. By caching content closer to end-users, latency is minimized, making the application responsive no matter where our users are.

Reliability:

Amazon S3: Beyond its scalability, S3 provides an assurance of durability. Its distributed nature ensures that our user data, especially the media files, are secure and always accessible.

Elastic Beanstalk & DynamoDB: With these managed services, the operational overhead is significantly reduced. They come with in-built redundancy, ensuring our application is always up, even if individual components face issues.

Route53: Reliable DNS resolution is crucial for online services. Route53 not only provides this but also has health checks and failovers to ensure users always reach a healthy endpoint.

Security:

IAM & Cognito: Security is multi-faceted. While IAM ensures that within our infrastructure, only authorized actions take place, Cognito provides a robust user management system. This two-pronged approach ensures both internal and external security.

Encryption with S3: Data security is paramount, especially with user media. With S3, we have assurances of both in-transit and at-rest encryption.

CloudFront Protection: In today's cyber landscape, threats can be external too. With CloudFront's integration with AWS's defensive services, we have a shield against a wide range of cyber threats.

In summary, our solution leverages the strengths of AWS services to create an architecture that's not only robust and scalable but also secure and cost-effective. This aligns with the company's growth projections and global user base while ensuring a seamless user experience