

COS20019: Assignment 3

Serverless/Event-driven Architectural Design Report

Lorraine Becker

Engineering and Technology, Swinburne University of
Technology, PO Box 218, Hawthorn, VIC 3122, Melbourne,
Australia
Email: 1104584773@student.swin.edu.au

Oliver Moloney

Engineering and Technology, Swinburne University of
Technology, PO Box 218, Hawthorn, VIC 3122, Melbourne,
Australia
Email: 104273068@student.swin.edu.au

Danny Paul Osman

Engineering and Technology, Swinburne University of
Technology, PO Box 218, Hawthorn, VIC 3122, Melbourne,
Australia
Email: 104249645@student.swin.edu.au

Tessa Mordue

Engineering and Technology, Swinburne University of
Technology, PO Box 218, Hawthorn, VIC 3122, Melbourne,
Australia
Email: 104540555@student.swin.edu.au

Abstract—This report examines alternative solutions to develop a design for a scalable highly available Web Site and justifies our chosen solution with a design rationale

Keywords—*Serverless/Event-driven Solution, AWS, Architecture, Photo Album Application*

I. INTRODUCTION

This report demonstrates an architecture design that meets the company's requirements in reducing the need for in-house systems administration work through the use of Amazon Web Services (AWS) managed cloud services. In AWS Simple Storage Service (S3), all photos and other media will be stored. This strategy provides better affordability, improved availability, and ensures long-term data durability which assists in facilitating easier scaling as demand increases.

It is expected for the application's demand to double every six months, with the rapid growth projected to continue for at least the next two to three years. The architecture will be created to handle this exponential increase in demand, ensuring high scalability and reliability in performance. Moreover, an increase in user load will be managed effectively through AWS, along with its services such as Containerization like Docker with Amazon ECS or Amazon EKS, Elastic Load Balancing (distributes traffic among multiple EC2 instances and prevents any one instance from becoming overloaded as user demand increases) and Auto Scaling (adjusts EC2 instance numbers in response to demand and ensures reliable performance as user traffic changes).

To achieve a workload reduction of 50% to 60%, the application will implement AWS Auto Scaling. This strategy will effectively enhance the compute capacity of EC2 instances so that the existing t2.micro EC2 instances do not exceed the 80% performance threshold, preventing overloading.

To adopt a serverless/event-driven solution, the company will use AWS Lambda functions, a serverless computing service that allows running code without provisioning or managing servers. By using Lambda functions, the application can respond to user actions in real-time without the need for managing infrastructure. This approach offers pay-as-you-go-pricing, reduces operational overhead, scales automatically based on demand, and aligns well with an event-driven architecture. Additionally, within an event-driven architecture, where actions trigger responses across various services, this will give a highly responsive and adaptable system.

To address the challenges caused by the slow, expensive relational database, moving to another cost-effective solution for its simple table structure would be necessary. This means that transitioning to Amazon DynamoDB would be one viable option as this NoSQL database service offered by AWS is fully managed. DynamoDB, which also happens to be one of the most optimised NoSQL file storage systems for speed as well as being scalable and maintaining low operational costs. This makes it ideal for applications with straightforward data models. Its pricing model is pay-per-request because that way it pays only based on the resources used by the company hence avoiding any additional investments during its inception stage while reducing how much is spent running it. Additionally, DynamoDB can automatically scale up when workloads increase without requiring manual intervention, thus enabling the company to meet higher demands while maintaining optimal performance and cost-effectiveness.

In managing slow response times across international locations, the architecture will utilise either AWS Global Accelerator or Amazon CloudFront for Global Response Time Optimisation. These services leverage content caching and delivery through global edge locations, significantly lowering the latency for global users and augmenting overall response times.

The design will have a durable and versatile structure for processing and transforming media. Our proposal recommends using AWS Step Functions in order to simplify the processing of media by creating various versions when uploading onto S3. Furthermore, the design will be prepared to handle tasks that rely on Artificial Intelligence (AI) such as identifying tags on photos without manual intervention.

II. ARCHITECTURE DESIGN



```

graph LR
    RouteS3[Route S3  
+Route()] -- "Route to AWS Amplify" --> AWSAmplify[AWS Amplify  
-GetData()  
-SendData()]
    AWSAmplify -- "Send Upload Request" --> APISGateway[API Gateway  
-Options()  
-Get()  
-Put()  
-Delete()]
    APISGateway -- "Send Put Request" --> Lambda[Lambda  
-SendMedia()  
-RequestMediaType()  
-RequestMediaTypes()  
-CategorizeData()]
    Lambda -- "Send Media" --> RouteS3
  
```

The diagram illustrates the AWS Amplify Backend Architecture. It shows the flow of data and requests between several components:

- Route S3**: Contains the `+Route()` method. It receives an input from the left and sends a **Route to AWS Amplify** request to the AWS Amplify component.
- AWS Amplify**: Contains the `-GetData()` and `-SendData()` methods. It receives the **Route to AWS Amplify** request and sends a **Send Upload Request** to the API Gateway component.
- API Gateway**: Contains the `-Options()`, `-Get()`, `-Put()`, and `-Delete()` methods. It receives the **Send Upload Request** and sends a **Send Put Request** to the Lambda component.
- Lambda**: Contains the `-SendMedia()`, `-RequestMediaType()`, `-RequestMediaTypes()`, and `-CategorizeData()` methods. It receives the **Send Put Request** and sends a **Send Media** request back to the Route S3 component.

```

graph TD
    SNS[AWS SNS  
-SendMedia()  
-AssignMediaTopic()]
    SQS[AWS SQS  
-QueueRequest()  
-SendMedia()]
    AR[Amazon Rekognition Image  
-ProcessData()  
-SendData()]
    AEI[Amazon Elastic Image  
-ProcessData()  
-SendData()]
    P[Publish Lambda  
-SendData()]
    S3[S3 Bucket  
-GetData()  
-StoreData()]
    DDB[AWS DynamoDB  
-SendData()  
-StoreData()]

    SNS -- "Send media with specific type to queue" --> SQS
    SQS -- "Store photo media to be transcoded" --> AR
    SQS -- "Store video media to be transcoded" --> AEI
    AR -- "Send transcoded data to DB" --> P
    AEI -- "Send transcoded data to DB" --> P
    P -- "Send media to S3" --> S3
    P -- "Send media to S3" --> DDB

```

IV: DESCRIPTION OF ALL AWS SERVICES USED

Amazon CloudFront is a service provided by AWS that delivers various forms of data including videos, applications and API to users on your network. The methods used by CloudFront ensure the network has low latency as well as high transfer speeds. In addition to streamlined content delivery, it can also monitor users on your network and provide data on your network traffic. This service is considered cost effective for a network as you only pay per data request that is used to deliver data to users on the network.

2) DynamoDB

DynamoDB is a database service provided by AWS. It has various performance and scalability features that enable the network to operate at optimal speeds even when resources are under pressure. Firstly, DynamoDB can upscale and downscale your database relative to the flow of traffic through the network at a given point in time to ensure the network functions efficiently without any downtime. Furthermore, due to the improved response times as compared to regular SQL based databases, DynamoDB can provide response times within 1-2 milliseconds compared to conventional database solutions. Lastly, the service is fully managed by AWS meaning the hardware side of the database as well as software and backups are handled by AWS.

3) S3

Amazon S3 (Amazon Simple Storage Service) is a service provided by AWS that is a web based, cloud storage service that creates online backup and archiving of data and applications used on the AWS services. In regards to scalability, the S3 service can scale storage as much as the user requires as there are no limits on how much data can be stored within the S3 Bucket. In addition, the S3 service provides additional security measures to ensure the data being stored is secure; security protocols such as IAM policies can be implemented to limit certain users on the network to access and modify the bucket limiting the risk factors of unauthorised users accessing the data being stored. In regard to data management, the S3 service also implements versioning meaning that all versions of a specific file being stored on the bucket are saved. This means if an incorrect version of the same file is uploaded, the previous version can overwrite the currently stored file.

4) SNS

SNS (Simple Notification Service) is a service provided by AWS that enables messages to be distributed to a variety of users on the network. This service can be utilised to send messages via a variety of mediums for example, email addresses or mobile devices. The service can be modified so that certain users subscribed for correspondence only receive messages, for example, this may be subscribers to your network's service who are of a higher level of membership or those who have opted in for weekly newsletters. With this feature implemented, the network decreases the strain on the network by limiting the amount of resources being used to only those that are necessary. Therefore, the SNS service can be tailored to various mediums, various types of subscribers as well as the capacity of the network. SNS can also be implemented on the backend for users on the network with the service able to utilise the CloudWatch service to send and notify users when aspects of the network are under stress.

5) SQS

SQS is a message queuing system that allows you to downscale and upscale microservices, systems and serverless applications running on the AWS network. The service allows you to communicate as well as store communications between software components. By utilising asynchronous messaging, the service can decouple parts of an applications software which enables them to run independently and therefore make the application more modular which in turn makes it more scalable. Furthermore, the SQS services employ the IAM service to limit access to messaging queues as well as using server side encryption to protect message contents.

6) Amazon Elastic Transcoder

The Amazon Elastic Transcoder is a service provided by AWS that can convert media files into versions of the file that will be able to play on smartphones, smart tablets and computers. It essentially can optimise media files to be viewed on all formats. It provides ease of use features for the user including predefined presets that the media file can be converted to that are commonly used for media output on devices. This makes the process of transcoding media much more streamlined than manually transcoding files and does not require a vast understanding of the subject. In addition, the service can scale to cater for any size of media file as well as any amount of files meaning the service can scale to any size needed by the user. In terms of cost factors, the service uses a pay as you go pricing scheme meaning you only pay for the length of the media as well as the quality of the media you are wanting to output (e.g a 30 minute video being transcoded to a resolution of 2560x1600 will cost higher than a 2 minute video being transcoded to a resolution of 1920x1080).

7) Amazon Rekognition Image

Amazon Rekognition is a deep-learning based image and video analysis service that users can implement into their videos and images without any extensive understanding of deep learning. The service can recognise various things including objects, humans, text, activities and content that may be deemed inappropriate. In addition to these areas being identified by the service, a confidence score is also provided to identify the accuracy of the object being identified.

8) Lambda

Lambda is provided by AWS and is a serverless computer service. The service essentially allows you to run code without managing or provisioning any additional servers. The service uses the code provided by the user to scale your application when certain triggers, set by other AWS services or requests have been met. For example, file uploads to the S3 bucket may trigger an event to be executed by the lambda service on the network.

9) Route 53

Route 53 is a service provided by AWS that provides a DNS system for your network. The service allows users to have their network requests routed to various network resources including EC2 instances running and S3 buckets on the network. In addition, the service allows for users to register domain names directly through the service enabling ease of use for users. Furthermore, route 53 utilises routing policies to route traffic and control the distribution of resources across the network through various methods including weighted routing, geolocation routing and simple routing.

10) API Gateway

API Gateway is a service that streamlines the process for developers to create, publish, maintain, and monitor application programming interfaces. The service allows you to define your API routes, methods, and integrations with the provided configuration interface. In addition, the service also allows for versioning, meaning updates can be rolled out or rolled back to existing APIs as needed.

11) AWS Amplify

AWS Amplify is a service provided by AWS that streamlines the development process for web and mobile applications. This is done by the service offering a variety of components to be deployed for your application that speeds up the development process including for both front-end and back-end development, security, and application data storage. In regards to front-end development, the suite provided has a variety of framework integrations for popular codebases including React. In regards to back-end development, the Lambda service can be integrated into the application in order to trigger back-end processes when a front-end event occurs. In regards to security, authentication measures including AWS's own authentication SDK's can be implemented to ensure safe and secure access to the application.

12) Edge Location

Edge locations are utilised by AWS's CloudFront service. These locations are placed around the world at specific locations in order to minimise network latency and uphold high transfer speeds for the network. Furthermore, these edge locations allow for further scalability and redundancy of the network. In the event of a network failure or outage, there are multiple locations to ensure that the downtime is minimised greatly and high network availability is upheld. In addition to this, edge locations cache content that is accessed frequently by users on the network to decrease overall network stress on web servers and S3 buckets that store frequently accessed files, this not only decreases network stress but increases network speeds as well.

V. REQUIREMENTS AND JUSTIFICATION

1). AWS S3

In order to fit the design criteria for the PhotoAlbum website, the use of AWS S3 is an optimal solution due to the factors of reliability, security, scalability, latency, and the overall integration into the AWS ecosystem. In terms of reliability, AWS S3 satisfies the needs of the business PhotoAlbum website due to its resilience to data corruption and hardware failures as well as regular maintenance provided by AWS in order to ensure the service remains functional and operational. AWS S3 also provides encryption at rest with server-side encryption, as well as allowing for client-side encryption. AWS S3 is also compatible with IAM roles allowing for a unified security system across the architecture. AWS S3 provides high scalability enabling the architecture to become flexible with its consumed resources relative to what it requires to function. AWS S3 also allows for versioning which can prevent the loss of data resulting from corruption, or user error such as accidental modification or deletion. AWS S3 is used for the storage of incoming media before transcoding as well as storage for the output transcoded media created by the PhotoAlbum site. This architecture involves two separate buckets for both of the aforementioned uses.

2) AWS CloudFront

AWS CloudFront provides low-latency transfer speeds through the use of AWS Edges, a global network of data centres where CloudFront caches data inside. These Edges are protected by AWS Shield and AWS WAF which prevent the interception of data or forms of injection which would pose major risks to the PhotoAlbum website and the stored data. By using AWS CloudFront, the PhotoAlbum can provide the user with a quick and efficient user experience. CloudFront is highly scalable due to the payment process it follows, using network requests as a metric rather than a flat rate. While providing fast transfer speeds, having the global network of Edges allows the PhotoAlbum to be able to quickly transfer data regardless of the user's location. Finally, CloudFront provides integration with IAM roles to ensure settings cannot be changed by an unauthorised user which could compromise the PhotoAlbum site and architecture.

3) AWS Lambda

AWS Lambda allows for the execution of code of most types for backend operations or applications. As Lambda is an event-driven architecture, it can automatically respond to changes in an S3 bucket. As a result of this, the nature of this architecture allows for an optimised, performant, and reliable solution as it reduces the number of instructions and actions required for a task to be completed. As a result of this, the overall cost of using Lambda is significantly cheaper due to these optimizations. In the PhotoAlbum architecture, there are two instances of Lambda present: one is responsible for communicating with other services whilst the other collects and stores the data inputted into their respective storage locations.

4) DynamoDB

DynamoDB provides a cost-efficient, flexible, scalable, and low-latency cloud-based solution to data storage. The various features included in DynamoDB responsible for scalability and performance will prove useful in the expansion of the PhotoAlbum Architecture, with the flexibility NoSQL provides being ideal for metadata storage and communicating said data with Lambda. Additionally, Amazon states that DynamoDB has a “25% lower cost of equivalent database environments” providing a financial advantage for the PhotoAlbum architecture which when combined with its scalability drastically improves the functionality of the architecture as a sustainable long-term framework. Concerning the scalability of DynamoDB, the database can be upscaled or downscaled according to the flow of traffic through the network to compensate for high loads and maintain efficiency with lower loads. AWS also provides data and software backups which act as insurance if an error were to occur that causes damage to the data within the database or the database as a whole. DynamoDB is considered one of the most optimised NoSQL file storage systems in regards to speed and also provides a low cost due to the pay-per-request pricing model, only using what is necessary and therefore only costing relative to what is necessary for the PhotoAlbum website's functionality.

5) AWS SQS & SNS:

AWS SQS and SNS can be used hand-in-hand to run operations asynchronously from each other and provide notifications when tasks are completed, allowing the two to function together can allowing for further integration between AWS Services, essentially chaining services. SNS and SQS can be chained together utilising the lambda function. This chaining function can enable further scalability for the network for example, a lambda function can be triggered depending on the SQS and SNS messages sent, these functions can include updating DynamoDB tables with further data or API gateway endpoints being invoked. An API gateway being invoked could be crucial for a network structure as the HTTP endpoint is required for the on-device client to make HTTP requests and interact with the backend of the API. This process can be streamlined by chaining SQS and SNS messages with Lambda to ensure host devices can interact with the endpoint.

6) AWS Amplify

The AWS Amplify suite is a streamlined tool that cuts down both front and back-end development. It has a variety of framework integrations that can be utilised meaning overall compatibility of the photo album would be optimal, especially in cases of scalability where more complex frameworks may be useful, for example, if the photo album were to be developed to be more complex, the React framework may be integrated to generate a more complex photo album with various features. In addition to this, the implementation of user authentication procedures means the website can be more secure and not accessed by volatile sources. For example, to minimise strain on EC2 resources, multi-factor authentication could be employed to ensure all users accessing the site are authorised to access the backend. Overall the utilisation of AWS Amplify would enable the photo album to be more easily scaled in the future as well as more complex features to be implemented more efficiently, lastly, the added security features from AWS would mean access to the backend of the photo album is more limited as there are more fail-safe procedures implemented to restrict unauthorised access.

7) Route 53

Route 53 allows for the routing of network requests to a wide range of network resources as a DNS web service. This can involve the routing of EC2 instance and S3 bucket requests through a network and acts as what routes the web server's requests to the rest of the architecture in the case of the PhotoAlbum site. Route 53 can also provide a domain to the site, giving a human-readable URL as an access point allowing for more traffic to the PhotoAlbum website. Additionally, the use of controlling the distribution of resources via routing policies can prove beneficial to the PhotoAlbum website as it provides a large benefit while the architecture is scaling dynamically to direct traffic in a way that prevents unneeded pressure on the system and maintains fast response times. This specific factor would be important if the Photo Album website is to scale to cater to a larger user base.

8) API Gateway

The use of API gateways in the context of the PhotoAlbum website acts to streamline the process that developers are required to use when developing interfaces. It aims to connect front-end application services to the architecture as a bridge of sorts, interfacing the data being sent from the front-end application to the processes within the architecture. The API Gateway will make a request to AWS Amplify and receive a response to display on the website's front end. API Gateway allows for versioning which will allow for changes to be made to the API Gateway without the risk of losing the previous version and by extension, website functionality.

VI. ALTERNATIVE SOLUTION

Our proposed alternative to the redesign will boost the reliability, performance, scalability, safety, and comfortability of the photo album application by taking a varied approach that is flexible. To make it able to adapt to changing needs our idea is to mix cloud services from different providers such as Amazon Web Services as well as other top vendors. Multi cloud strategy helps application to have more tools available reducing dependency on one provider hence making it more resilient against possible downtime events or disruptions in service delivery. Meanwhile, this technique provides an opportunity for specific use cases to use the most suitable services (for better performance and reliability) despite variation in operating conditions.

Additionally, we suggest taking up microservices when adopting multi-cloud architecture for designing our applications. This way the app is broken down into smaller parts that are independent thus leading to greater flexibility during development and faster ability of launching any change within it. In this case each microservice can independently scale on-demand; therefore leading to effective resource utilisation and lowering the chances of bottlenecks or slow performance issues arising. Microservices also support continuous integration/deployment practices within a system without affecting system stability or reliability which is necessary for innovation.

Our proposal addresses performance concerns by leveraging advanced content delivery networks(CDNs) and managed DNS services. In addition to Amazon Web Services CloudFront, consider exploring other alternatives like Cloudflare or Akamai whose caches and routing capabilities are much more advanced in terms of content delivery acceleration and latency reduction for the end users. Managed DNS services such as Cloudflare's DNS or DNS Made Easy can optimise DNS resolution by directing users towards servers that are nearer and faster leading to better response times as well as user experience quality. By using these technologies in conjunction with AWS services we ensure high performance regardless of location worldwide and across different network situations.

Scalability was also a significant issue considered within our design solution, hence we suggested that auto-scaling and elastic computing resources should be used to adjust to high levels of user demand. On the other hand, services like AWS Auto Scaling as well as Kubernetes container orchestration platforms would ensure dynamic resource scaling based on prevailing real-time traffic patterns meaning that there would be an application capable of handling surges in user activity without compromising on its performance or reliability. Moreover, server-less computing functions facilitated through AWS Lambda plays a critical role in allowing automatic scaling of compute resources whenever there is a request coming in hence enhancing scalability and resource efficiency even more .Incorporating these scalable architectural patterns will make us create a design solution which allows the photo album app to change its focus across various workloads and demands of customers thereby meeting need globally with minimum delay.

VI. DESIGN CRITERIA FULFILMENT

Category	Criteria	Justification
Reliability	<ul style="list-style-type: none"> - Adaptable to changes - High adaptability to changes - Implements frequent Data Backups 	<ul style="list-style-type: none"> - Amazon S3 and DynamoDB are used to guarantee the durability and easy availability of the system. - CloudFront is used because it is able to handle failovers, as well as transfer content from the nearest edge location.
Performance	<ul style="list-style-type: none"> - Efficient Content Delivery - Transcoding Efficiency 	<ul style="list-style-type: none"> - To speed up the content delivery and reduce latency, we are using AWS CloudFront with AWS Route 53 in an effort to optimise it. - AWS Elastic Transcoder is used for effective media files transcoding to several formats, for easy running and better user experience.
Scalability	<ul style="list-style-type: none"> - Scalable Content Delivery - Elastic Resource Allocation 	<ul style="list-style-type: none"> - Using AWS CloudFront and AWS Route 53 to scale delivery resources while user requirements change. This ensures scalability in performance. The system relies on the following to deliver resources elastically while maintaining minimal resource wastage, thus guaranteeing a much more efficient system. - DynamoDB has scalable metadata storage. - Amazon S3 is a highly-scalable cloud service for data storing.
Security	<ul style="list-style-type: none"> - Fine-Grained Access Control - Comprehensive Monitoring and Alerting - Data Encryption 	<ul style="list-style-type: none"> - CloudWatch service of AWS is integrated to ensure comprehensive monitoring and alerting that allows for real-time detection of security threats and proactive response measures thus enhancing robust security. - AWS S3 uses encryption mechanisms to uphold data security alongside fulfilling regulatory compliance requirements to further increase data protection and security measures said means AWS Identity & Access Management (IAM) should be implemented for finer user control and widespread access to resources such as permissions incrementing its overall security.

Fig 4 Design Criteria and Justification

VII. COST BUDGET FOR EACH COMPONENT SERVICE AND TOTAL MONTHLY BUDGET

Service Name	Cost per month	Justification
CloudFront	\$0.060 - \$0.120 per month	CloudFront delivers content using global edge locations with minimal time lags and maximally high speeds of data transmission. So that global users can have an effective user experience that is also speedy is vital for an app like Photo Album app which contains many media files. The benefits justify the cost as there are huge improvements on performance alongside user satisfaction.
DynamoDB	\$5.33 per month	The cost is relatively low for the reliability and scalability it provides, making it a cost-effective choice for database management.
S3	\$0.021 - \$0.023 per month	What makes S3 outstanding among other storage systems available online today is its ability to store media files with redundancy at low prices while at the same time being highly durable. The software charges for only the space used which is critical when managing large amounts of photos and videos. It's cost-effective because of its extensible storage capability combined with high durability index and good security mechanisms that ensure that terabytes or petabytes remain available when needed for computation.
SNS	\$2.00 per month	SNS is a premium message service that ensures reliable messaging among services efficiently. This is very important when there's an event-driven pattern in place that requires timely and dependable communication among various components of the system. Therefore, it comes at a price to cater for its reliability as well as importance to the system.
SQS	\$0.24 - \$0.40 per month	Amazon SQS offers a cost effective way to decouple and coordinate distributed applications components scalability. It helps to govern media tasks process flow in order to ensure smooth efficient processing of those tasks. It justifies cost by having reliable handling message
		queues that are necessary for its performance.
Amazon Elastic Transcoder	\$0.03 per month	Elastic Transcoder offers a cheap way to convert media from one format into another. It's necessary for video viewing on various devices to create an uninterrupted viewing experience. Therefore paid version offers customised products in this field hence being advantageous in comparison with free plan terms
Amazon Rekognition Image	\$62.5 per month	While the price is higher, Rekognition's advanced capabilities significantly decrease the manual work needed to make sense of and deal with huge files of media.
Lambda	Free	Lambda enables running programs without managing any server instances reducing operational costs. It charges only for the amount of computing time it uses with costs scaling from USD.20 for 1 GB-second up to thousands or even millions of requests/day according to the statistics. This significant reduction in infrastructure management and provision of increased flexibility justifies the free tier cost
Route 53	\$0.50 per month	Route 53 provides reliable DNS service required by web sites intending to directly link end users towards any application running on an EC2 instance. It helps maintain high availability levels coupled with optimal performance rates, ensuring minimal downtimes. Thus contributing directly to business continuity plans in case of disaster recovery, although it does not come cheap.
Total	\$70.681 - \$70.903	Total monthly cost is justified by the combination of services that ensure a good mix performance, availability and security which are key factors in web service delivery. To meet the company's growing requirements for scalable cloud solutions, each service performs an important function that contributes to the solid, scalable and cost-effective cloud solution.

Fig 5 Cost Budget and Justification

REFERENCES

- [1] ARCHIVED: Reliability Pillar - AWS Well-Architected Framework. (n.d.). Available at: <https://d1.awsstatic.com/whitepapers/architecture/AWS-Reliability-Pillar.pdf?ref=wellarchitected-ws>
- [2] AWS CloudFront (2018). Content Delivery Network (CDN) | Low Latency, High Transfer Speeds, Video Streaming | Amazon CloudFront. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/cloudfront/>
- [3] AWS (2019). Amazon Route 53 - Amazon Web Services. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/route53/>
- [4] Amazon Web Services, Inc. (2019). AWS | Amazon Elastic Transcoder - Media & Video Transcoding in the Cloud. [online] Available at: <https://aws.amazon.com/elastictranscoder/>
- [5] AWS (2019). Amazon DynamoDB - Overview. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/dynamodb/>
- [6] AWS (2023). Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/s3/>
- [7] AWS (2019). AWS Lambda – Serverless Compute - Amazon Web Services. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/lambda/>
- [8] AWS (n.d.). Amazon S3 Security Features - Amazon Web Services. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/s3/security/?nc=sn&loc=5>.
- [9] AWS (2023). Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/s3/>.
- [10] AWS (2023). What is Amazon S3? - Amazon Simple Storage Service. [online] docs.aws.amazon.com. Available at: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>.
- [11] docs.aws.amazon.com. (n.d.). Using versioning in S3 buckets - Amazon Simple Storage Service. [online] Available at: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html>.
- [12] docs.aws.amazon.com. (n.d.). AWS glossary - AWS Glossary. [online] Available at: <https://docs.aws.amazon.com/glossary/latest/reference/glos-chap.html>.

