COS20019 Assignment 3

|  |  |
| --- | --- |
| Phan Vu 104222099  *Tutor class: 1.00 pm Saturday*  *Swinburne University of Technology Ho Chi Minh, Vietnam*  104222099@student.swin.edu.au | Tran Kim Thu 104061810  *Tutor class: 1.00 am Saturday*  *Swinburne University of Technology Ho Chi Minh, Vietnam*  104061810@student.swin.edu.au |

***Abstrac*t - This report outlines the architectural plan for a flexible and easily navigable web application designed for a Multimedia Sharing platform. The presentation highlights the essential elements of the design, emphasizing incorporating various cloud services to guarantee strength and scalability. The goal is to delineate the data flow for fundamental operations such as uploading material, tracking user interactions, providing content suggestions, and enabling real-time search functionalities. The application's objective is to expand substantially, which requires ongoing development to meet increasing needs and provide customized user experiences.**

***Index terms* - Cloud infrastructure, web application, scalability, multimedia processing**

***I. Introduction***

Cloud computing has become essential for modern digital applications, providing flexible and practical solutions to address growing needs. This study proposes an architectural plan to improve and expand the Photo Album application, which has gained considerable success and is ready to meet the growing needs of users. Our design utilizes the functionalities of AWS and implements a serverless/event-driven architecture to align with modern cloud computing concepts.

Several problems and opportunities require the improvement of this application's features. This architectural solution utilizes AWS services and modern cloud computing principles to address the issues of escalating user traffic, global accessibility, optimization of media storage, and efficient processing of diverse media formats.

Our goal is to address the business scenario's objectives thoroughly, considering scalability, dependability, security, and cost-effectiveness. Our main goal as the architects of this proposed solution is to clarify the architectural elements and their reasoning to help you understand how each component smoothly fits into the system.

This study examines the use of cloud services to provide a substantial, scalable, and practical architecture for the Photo Album application while embracing technological progress. We evaluate complex architectural details and technology frameworks to determine a robust and future-oriented design and compare it to well-established industry benchmarks.

The following sections provide a complete analysis of the architectural elements, including cloud services, roles, interactions, and design justifications. UML collaboration diagrams and critical design analysis will offer a valuable understanding of the solution's efficacy.

This study aims to propose a comprehensive solution and offer direction for the long-lasting, expandable, and effective digital infrastructure of the Photo Album application, serving as a protectors of an innovative cloud architecture.

***II. Architecture Overview***

**A diagram of a company

Description automatically generated with medium confidence**

*Figure 1.*

1. ***Five Tier Architecture***

***1. User Interface Tier***

**CloudFront:** Provides low-latency, global access to the application, enhancing user experience.

**Elastic Beanstalk:** Simplifies deployment and scaling of web applications, ensuring the application layer can handle varying workloads.

***2. Application Tier***

**Lambda (for business logic):** Enables serverless execution of business logic, allowing for efficient scaling and cost optimization.

***3. Data Storage Tier***

**S3 (for media storage):** Provides scalable and durable object storage for storing large media files.

**DynamoDB (for data storage)**: Offers single-digit millisecond performance at scale, suitable for quick data retrieval and storage.

***4. Media Transformation Tier***

**Lambda (for media transformation)**: Enables on-demand media transformation, promoting efficiency and scalability.

**Elastic Transcoder:** Provides scalable media transcoding, ensuring compatibility with various devices.

***5. Decoupling Tier***

**SQS (for decoupling):** Enhances decoupling between components, allowing asynchronous communication and improving fault tolerance.

1. ***Detail each tier***

**User Interface Layer:** The UI Layer serves as the primary interface for users, acting as the entry point to the underlying media processing engine. This layer is specifically developed to prioritize and deliver a smooth and captivating user experience. Amazon CloudFront is a globally distributed content delivery network (CDN) at the forefront of the UI Layer. CloudFront guarantees the secure, fast, and efficient distribution of media content, improving user experience. It efficiently speeds up static and dynamic material delivery, acting as a dependable link between consumers and the media processing infrastructure. The AWS Elastic Beanstalk service, designed for effortless deployment and scalability of web applications, is effortlessly integrated into the UI Layer. It coordinates the implementation and adjustment of the programs that users interact with, improving the overall ability to control and handle the system.

Furthermore, Amazon S3 enables the practical storage and retrieval of media assets, guaranteeing a robust and expandable basis for media processing. Amazon Cognito is utilized to optimize verifying user identity, registering new users, and managing user permissions. This guarantees that the user's engagement within the media processing system is smooth and compliant with rigorous security protocols. Cognito's capabilities provide an additional level of identity management, boosting the UI Layer's overall security. Amazon Route 53 is incorporated into the UI Layer to handle domain routing efficiently. This service facilitates the connection between user requests and the underlying infrastructure, which includes CloudFront. It ensures that traffic is routed optimally to AWS resources. The integration is essential for ensuring a solid and adaptable user interface. IAM policies are incorporated into the UI Layer to administer access securely. This guarantees that only authorized users can perform particular operations within the media processing system, protecting sensitive user data and media material.

A diagram of a computer

Description automatically generated

*Figure 1.*

**Application Layer:** The Application Layer functions as the central processing unit, coordinating the logic and features that power the media processing system. This layer is crucial in handling user requests, implementing business logic, and communicating with underlying data storage and processing components. AWS Elastic Beanstalk is a significant Application Layer component that offers a platform for deploying, maintaining, and scaling web applications. The system's user-friendly nature simplifies the implementation process, enabling developers to concentrate on developing applications without worrying about the intricacies of managing infrastructure. Elastic Beanstalk effortlessly supports several application architectures and frameworks, making it an optimal basis for the media processing system. Flexible Beanstalk functions as the central point of connection for other essential components in the Application Layer. It connects with AWS Lambda to execute business logic, allowing serverless computing capabilities. The integration with DynamoDB, a highly scalable and performant NoSQL database, enables efficient storing and retrieval of data. In addition, the integration with Amazon Simple Queue Service (SQS) facilitates the separation of components, guaranteeing the ability to handle increased workload and system failures by managing asynchronous communication between various system elements.

A diagram of a company

Description automatically generated

*Figure 1.*

A diagram of a software application

Description automatically generated

*Figure 1.*

**Data Storage Layer:** The Data Storage Layer is the storage location for a wide range of media assets and their corresponding metadata, which is crucial in providing effective and dependable data management. This layer utilizes Amazon Simple Storage Service (S3) to store media and Amazon DynamoDB to store structured data. Amazon S3 is a highly scalable service for storing objects, primarily for storing video assets in the Data Storage Layer. S3 provides unparalleled scalability, data availability, security, and performance. It can hold an almost limitless amount of data, making it an optimal solution for managing media assets’ varied and extensive characteristics in different formats. The smooth incorporation of S3 with AWS Lambda enables real-time media conversion. Lambda functions are automatically activated whenever new media objects are uploaded to S3, allowing for the dynamic processing and alteration of media files. This integration guarantees that the media assets are efficiently prepared for consumption on various devices and platforms. The Data Storage Layer incorporates Amazon DynamoDB, a fully managed NoSQL database service, to meet the needs of structured data storage. DynamoDB has high-speed performance, with response times in the range of single-digit milliseconds, regardless of the amount of data being processed. This makes it an ideal choice for applications requiring high read-and-write operations. DynamoDB effortlessly connects with AWS Lambda to execute business logic. This integration guarantees that when data is fetched or modified in DynamoDB, it will automatically activate serverless functions. This allows for immediate and adaptable reactions to changes to the underlying data structure.

A diagram of a software company

Description automatically generated

*Figure 1.*

A diagram of a company

Description automatically generated

*Figure 1.*

**A diagram of a data flow

Description automatically generated**

*Figure 1.*

**Media Transformation Layer:** The Media Transformation Layer focuses on the dynamic processing and optimizing media assets, ensuring they are appropriately structured for smooth consumption on various devices and platforms. This layer significantly improves the adaptability and availability of media material by utilizing AWS Lambda for media transformation and AWS Elastic Transcoder for transcoding. AWS Lambda is the central component of the Media Transformation Layer, providing serverless computing capabilities for real-time media processing. Lambda functions are activated when new objects are uploaded to Amazon S3, causing dynamic and personalized modifications to media assets. The lambda functions in this layer can be customized to perform various media transformation operations. The workflows may encompass resizing, cropping, watermarking, or applying filters to photos and videos. The Media Transformation Layer effortlessly connects with AWS Elastic Transcoder, a specialized service for converting media files. Elastic Transcoder is precisely engineered to be scalable, user-friendly, and cost-efficient. This software facilitates the transformation of multimedia files from one format to another. The inherent scalability of Lambda and the parallel processing capabilities of Elastic Transcoder guarantee efficient execution of media transformations, even when dealing with large volumes.

A diagram of a diagram

Description automatically generated

*Figure 1.*

A diagram of a function

Description automatically generated

*Figure 1.*

**Decoupling Layer:** The Decoupling Layer is a crucial component in the media processing architecture. It enables asynchronous communication and loose coupling between various parts.   Amazon Simple Queue Service (SQS) is the central component of this layer, functioning as a fully controlled message queuing service. The Decoupling Layer enables scalability and resilience in processing by effectively managing message queues and promoting communication among different components. Amazon SQS facilitates decoupling by separating the members of the media processing system, enabling them to function autonomously. This is accomplished by utilizing message queues, which offer dependable and expandable communication between various system components. SQS connects with AWS Lambda effortlessly as a stimulus for Lambda functions that perform precise business logic. The Decoupling Layer provides asynchronous communication and initiates processing activities across distinct components by utilizing SQS as an event source for Lambda, eliminating direct dependencies. SQS serves as a centralized communication center for coordinating worker services. Every individual worker node can autonomously retrieve and handle messages from the queue, enabling concurrent and dispersed execution of tasks. SQS allows the postponement of message delivery, whereby notices can be retained in the line for a designated duration before becoming accessible for processing.

A diagram of a work flow

Description automatically generated

*Figure 1.*

|  |  |
| --- | --- |
| A diagram of a data storage system  Description automatically generated | A diagram of a workflow  Description automatically generated |

*Figure 1.*

***III. Design Rationale***

***A. Business Scenario Fulfillment***

***1. Scalability for Global Demand***

The business scenario entails managing fluctuating workloads of media processing jobs, which may encounter sudden increases in demand from users worldwide. By utilizing AWS CloudFront for worldwide content distribution and AWS Elastic Beanstalk for expandable web applications, the architecture guarantees that the system automatically adjusts resource allocation according to demand. By implementing responsive scaling, a uniform user experience is ensured across different regions, thereby tackling the issue of handling varying workloads.

***2. Efficient and Dynamic Media Processing***

The business necessitates streamlined and adaptable media processing operations, encompassing photo reformatting, video converting, and prospective forthcoming assignments.   The architecture effortlessly incorporates AWS Lambda for serverless computation and AWS Elastic Transcoder for media transcoding.   Specialized worker nodes are dedicated to tasks such as reformatting photos and transcoding videos, enabling parallelized and efficient material processing. The modular design ensures adaptability to expanding media processing requirements, which reserves Worker Node 3 for future work.

***3. Cost-Effective Resource Utilization***

Maximizing cost efficiency and resource utilization for media processing jobs is a critical corporate concern. The architecture guarantees efficient resource utilization by employing auto-scaling strategies, pay-as-you-go models of AWS Lambda, and tiered pricing structures of Elastic Transcoder. This strategic cost management method enables the business to adjust resources flexibly according to demand, optimizing operational costs in both high-demand and low-demand periods.

***4. Secure and Resilient Data Management***

Securing and safeguarding media assets and user data is a critical business priority. The Data Storage Layer employs Amazon S3 for storing media and DynamoDB for organizing structured data, offering strong security capabilities. Encryption protocols, Identity and Access Management (IAM) regulations, and version control methods guarantee the authenticity and security of media assets and user data. This fulfills the business's robust and reliable data management plan requirement.

***5. Adaptability to Future Technologies***

The company foresees the incorporation of upcoming technologies, such as AI-powered tasks, to improve media processing capabilities. The architecture's progressive approach, demonstrated by the reservation of Worker Node 3 for potential future duties, guarantees flexibility in adapting to developing technology. The modular architecture enables the firm to include new functionalities effortlessly and remain at the forefront of technological innovation.

***6. Asynchronous and Decoupled Processing***

It is crucial to guarantee the effective and independent handling of media jobs to preserve system responsiveness and fault tolerance. Utilizing Amazon SQS, the Decoupling Layer facilitates asynchronous communication and promotes loose coupling among components. This architecture improves the system's ability to withstand and recover from failures, handle many users or tasks, and execute multiple tasks simultaneously. The architecture guarantees the capacity to process charges autonomously, preventing potential obstacles and enabling flexible scalability through modularity.

***7. Real-time Monitoring and Operational Insights***

The business necessitates continuous monitoring and practical insights into the operational well-being and efficiency of the media processing system. The integration with Amazon CloudWatch enables instantaneous tracking and measurement of metrics. This guarantees that the business can actively oversee the design, react to changes in performance, and dynamically optimize the utilization of resources. The architecture gives the company the essential capabilities to make real-time decisions and optimize the system.

***B. Alternative Solutions Considered:***

***1. Virtual machines vs. Containers vs. Serverless computing.***

The organization's level of preparedness will determine the outcome of assuming responsibility and embracing vendor lock-in. Virtual machines and containers provide a higher level of precision in controlling server resources, allowing for a level of control that cannot be achieved in serverless configurations [1]. Serverless technology facilitates expedited time to market for new services through its more straightforward provisioning method. Choosing serverless architecture, specifically Lambda, minimizes operational burden, provides automatic scaling, and fits the requirement for efficient scalability by user expectations.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Aspect*** | ***Virtual Machines (VMs)*** | ***Containers*** | ***Serverless*** |
| **Resource Allocation** | Divided into multiple | Shares host OS resources | Utilizes cloud-based |
|  | VMs | lightweight, no Guest OS | servers, abstracts |
|  |  |  | underlying infrastructure |
| **Isolation** | Hard isolation between | Shares the host kernel, | Multi-tenant environment, |
|  | different VMs | but separate user spaces | limited isolation between |
|  |  | for applications | functions |
| **Portability** | Less portable | Highly portable, can move | Limited portability between |
|  |  | between cloud environments | different cloud providers |
| **Resource Efficiency** | More resource intensive | Efficient resource usage, | Efficient, cost-effective |
|  |  | shares OS resources | usage |
| **Scalability** | Scalable | Highly scalable | Highly scalable |
| **Ease of Deployment** | Dependent on VM setup | Quick deployment, | Rapid deployment, |
|  |  | facilitated by container | abstraction from underlying |
|  |  | images | infrastructure |
| **Cost Efficiency** | May require more | Cost-efficient due to | Cost-effective, billed on |
|  | resources | efficient resource usage | usage, not idle time |
| **Vendor Lock-In** | Potential dependence | Potential dependence | Potential vendor lock-in, |
|  | on specific VM tech | on specific container tech | dependent on provider APIs |
| **Security** | High isolation between | Security concerns due to | Security concerns due to |
|  | VMs, dedicated OS | shared host kernel, | multi-tenancy, shared |
|  |  | additional measures needed | server environment |

*Figure 1.*

***2. SQL vs. NoSQL database***

The preference for DynamoDB stemmed from its ability to function at scale with sub-millisecond latency, making it well-suited for rapid data retrieval and storage. This is particularly important for media-related applications that require high throughput.

|  |  |  |
| --- | --- | --- |
| ***Aspect*** | ***SQL*** | ***NoSQL*** |
| **Definition** | Structured Query Language | Not Only SQL |
| **Database Architecture** | Relational (RDBMS) | Non-relational (Various models) |
| **Suitable for** | Structured data with predefined schema | Unstructured and semi-structured data |
| **Data Storage** | Tables with columns and rows | Collections or documents |
| **Query Language** | SQL | Dynamic schemas, diverse query syntax |
| **Scaling** | Vertically scalable | Horizontally scalable |
| **Transaction Management** | ACID properties for transactions | May or may not follow ACID properties |
| **Complex Queries** | Supports JOIN and complex queries | Limited or no support for JOINs |
| **Data Structure** | Normalized data structure | Denormalized data structure |
| **Examples** | MySQL, PostgreSQL, Oracle | MongoDB, Cassandra, Redis |

*Figure 1.*

***3. Caching options***

Caching at the Edge (CloudFront):

* Rationale: Exploited due to the inherent characteristics of multimedia content.   CloudFront's edge caching strategically delivers frequently visited material from edge locations, lowering latency and optimizing performance by serving information closer to users.

Alternative Considerations:

* AWS ElastiCache: While ElastiCache offers in-memory caching solutions, using CloudFront for edge caching aligns better with the distributed nature of multimedia content delivery, reducing the load on backend systems and enhancing user experience.

***4. Push vs. Pull message handling***

Pull Mechanism (SQS):

* Rationale: Selected for its advantages in decoupling, fault tolerance, and scalability.   The pull mechanism of SQS enables components to obtain messages autonomously, fostering loose connectivity among various system components.

Alternative Considerations:

* Push Mechanism: While push mechanisms might offer real-time delivery, they can create tight dependencies between components, potentially leading to increased coupling and reduced fault tolerance.

***5. The number of tiers in architecture***

Five-Tier Architecture:

* Rationale: Chosen for its capacity to separate different aspects, support expansion, and encourage independent functioning of other components. This hierarchical strategy maximizes system efficiency and facilitates the wide range of capabilities necessary for multimedia processing and content distribution.

Alternative Considerations:

* Reduced Tiers (e.g., 3-tier): A simpler architecture might streamline operations but could sacrifice granularity and segregation of functionalities, potentially limiting scalability and manageability as the platform grows.

***C. Design Criteria***

|  |  |  |
| --- | --- | --- |
|  | **Criteria** | **Solution** |
| **Performance** | 1. Performance Boost | 1. Amazon CloudFront enhances the performance of the media processing system by reducing latency and accelerating the delivery of media assets. |
| 2.Quick Response Times | 2.AWS Lambda contributes to performance by providing serverless compute capabilities. |
| 3.Low-Latency Data Access | 3.DynamoDB plays a key role in performance by offering low-latency data access. |
| 4.Cost-Effective Transcoding | 4. Elastic Transcoder primarily addresses scalability, it indirectly contributes to performance. By offering scalable media transcoding capabilities, the service ensures that the system can efficiently process and deliver media files of varying sizes and formats. |
| **Scalability** | 1.Global Reach | 1.CloudFront's extensive network of edge locations contributes to scalability by providing a distributed infrastructure that can efficiently scale to handle increasing demands. |
| 2.On-Demand Scalability | 2.AWS Lambda is a cornerstone for scalability due to its serverless nature. |
| 3.Scalable and Reliable | 3.DynamoDB directly addresses scalability by offering a highly scalable NoSQL database. |
| 4.Highly Scalable | 4.Elastic Transcoder is explicitly designed for scalable media transcoding. |
| **Reliability** | 1.Fault Tolerance | 1.SQS is employed for asynchronous communication, enhancing fault tolerance by decoupling components. |
| 2.Cascading Failure Prevention | 2.Critical data, including media assets, is stored redundantly using services like Amazon S3 and DynamoDB. |
| 3.Data Integrity and Retention | 3. Decoupling and Worker are equipped with robust retry mechanisms and error handling procedures. |
| 4.Scalability and Load Balancing | 4.Lambda and Elastic Transcoder enables the architecture to dynamically scale based on demand. |
| **Security** | 1.Secure Access Control | 1.AWS IAM is employed to manage access to AWS services and resources securely. |
| 2.Authentication and Authorization | 2.Amazon Cognito is integrated to add secure user sign-up, sign-in, and access control functionalities to web and mobile applications. |
| 3.Access Management | 3. IAM policies are meticulously crafted to manage access control for each component in the architecture. |

*Figure 1.*

***D. Cost Estimation.***

**Assumption Cost for 1TB:**

|  |  |  |
| --- | --- | --- |
| **Service** | **Quantity** | **Total Monthly cost** |
| **S3 storage** | 1 TB per month x 1024 GB in a TB = 1024 GB per month  **1024 GB x 0.09 USD per GB = 92.16 USD**  Tiered price for: 1,024 GB  **1,024 GB x 0.023 USD = 23.55 USD**  Total tier cost = 23.552 USD (S3 Standard storage cost)  S3 Standard cost (monthly): 23.55 USD | **115.71 USD** |
| **CloudFront** | Tiered price for: 5,120 GB  5,120 GB x 0.085 USD = 435.20 USD  Total tier cost = 435.20 USD (Data transfer out)  Data transfer out to internet cost: 435.20 USD  5,120 GB x 0.02 USD = 102.40 USD (Data transfer out to origin)  Data transfer out to origin cost: 102.40 USD  10,000,000 requests x 0.000001 USD = 10.00 USD (HTTPS requests)  Requests cost: 10.00 USD  435.20 USD + 102.40 USD + 10.00 USD = 547.60 USD (Total cost United States)  CloudFront price (monthly): **547.60 USD** | **547.60 USD** |
| **Lambda** | 10,000,000 requests x 10 ms x 0.001 ms to sec conversion factor = 100,000.00 total compute (seconds)  10 GB x 100,000.00 seconds = 1,000,000.00 total compute (GB-s)  1,000,000.00 GB-s x 0.0000166667 USD = 16.67 USD (monthly compute charges)  10,000,000 requests x 0.0000002 USD = 2.00 USD (monthly request charges)  0.50 GB - 0.5 GB (no additional charge) = 0.00 GB billable ephemeral storage per function  16.67 USD + 2.00 USD = 18.67 USD  **Lambda costs - Without Free Tier (monthly): 18.67 USD** | **18.67 USD** |
| **Rout 53** | Tiered price for: 3  3 x 0.50 USD = 1.50 USD  Total tier cost = 1.50 USD (Hosted Zone cost)  10,000 records x 0.0015 USD per record = 15.00 USD (RRset records cost)  1.50 USD + 15.00 USD = 16.50 Total Hosted Zones and RRset records cost  Total Hosted Zones & RRset records cost: 16.50 USD  2 policy record per month x 50.00 USD = 100.00 USD (Traffic Flow cost)  10 million queries x 1000000 multiplier for million = 10,000,000.00 Standard queries in million  Tiered price for: 10,000,000.00 Standard queries  10,000,000 Standard queries x 0.0000004 USD = 4.00 USD  Total tier cost = 4.00 USD (Standard queries cost)  5 million queries x 1000000 multiplier for million = 5,000,000.00 billable Latency based routing queries  Tiered price for: 5,000,000.00 Latency based routing queries  5,000,000 Latency based routing queries x 0.0000006 USD = 3.00 USD  Total tier cost = 3.00 USD (Latency based routing queries cost)  1 million queries x 1000000 multiplier for million = 1,000,000.00 billable Geo DNS queries  Tiered price for: 1,000,000.00 Geo DNS queries  1,000,000 Geo DNS queries x 0.0000007 USD = 0.70 USD  Total tier cost = 0.70 USD (Geo DNS queries cost)  1 million queries x 1000000 multiplier for million = 1,000,000.00 billable IP-based routing queries  Tiered price for: 1,000,000.00 IP-based routing queries  1,000,000 IP-based routing queries x 0.0000008 USD = 0.80 USD  Total tier cost = 0.80 USD (IP-based routing queries cost)  Tiered price for: 1,000 IP (CIDR) blocks  1,000 IP (CIDR) blocks x 0.00 USD = 0.00 USD  Total tier cost = 0.00 USD (IP (CIDR) blocks cost)  16.50 USD + 100.00 USD + 4 USD + 3 USD + 0.70 USD + 0.80 USD = 125.00 USD  **Route53 Hosted Zone cost (monthly): 125.00 USD** | **125.00 USD** |
| **Simple Queue Service** | 10 requests per month x 1000000 multiplier for million = 10,000,000.00 total standard queue requests  Tiered price for: 10,000,000.00 requests  1,000,000 requests x 0.00 USD = 0.00 USD  9,000,000 requests x 0.0000004 USD = 3.60 USD  Total tier cost: 0.00 USD + 3.60 USD = 3.60 USD (Standard queue requests cost)  10 requests per month x 1000000 multiplier for million = 10,000,000.00 total FIFO queue requests  Tiered price for: 10,000,000.00 requests  1,000,000 requests x 0.00 USD = 0.00 USD  9,000,000 requests x 0.0000005 USD = 4.50 USD  Total tier cost: 0.00 USD + 4.50 USD = 4.50 USD (FIFO queue requests cost)  3.60 USD + 4.50 USD = 8.10 USD (Total SQS cost)  **Total SQS cost (monthly): 8.10 USD** | **8.10 USD** |
| **Elastic Transcoder** | DynamoDB data storage cost (Monthly): 256.00 USD  Video assets transcoding cost (Monthly): 180.00 USD  Audio (only) assets transcoding cost (Monthly): 9.00 USD  Total Upfront cost: 0.00 USD  **Total Monthly cost: 189.00 USD** | **189.00 USD** |
| **CloudWatch** | Tiered price for: 5 Dashboards  3 Dashboards x 0.00 USD = 0.00 USD  2 Dashboards x 3.00 USD = 6.00 USD  Total tier cost: 0.00 USD + 6.00 USD = 6.00 USD (Dashboards cost)  CloudWatch Dashboards and Alarms cost (monthly): 6.00 USD  Tiered price for: 10 metrics  10 metrics x 0.30 USD = 3.00 USD  Total tier cost = 3.00 USD (Metrics cost (includes custom metrics))  CloudWatch Metrics cost (monthly): 3.00 USD  **Total Monthly cost: 9.00 USD** | **9.00 USD** |
| **DynamoDB** | Monthly read cost (Monthly): 2.76 USD  Monthly write cost (Monthly): 23.38 USD  DynamoDB Data export to Amazon S3 cost (Monthly): 100.00 USD  DynamoDB Data import from Amazon S3 cost (Monthly): 75.00 USD  DynamoDB data storage cost (Monthly): 256.00 USD  Upfront read cost (Upfront): 30.00 USD  Upfront write cost (Upfront): 150.00 USD  Total Upfront cost: 180.00 USD  **Total Monthly cost: 457.14 USD** | **457.14 USD** |
| **Cognito** | 5,000 MAUs x 0.10 SAML or OIDC federation requests = 500.00 SAML or OIDC federation MAU requests  500.00 SAML or OIDC federation MAUs - 50 free SAML or OIDC federation MAU requests per month = 450.00 billable SAML or OIDC federation MAU requests  Max (450.00 billable SAML or OIDC federation MAU requests, 0 minimum billable SAML or OIDC federation MAU requests) = 450 total billable SAML or OIDC federation MAU requests  450 MAUs x 0.015 USD = 6.75 USD (SAML or OIDC federation MAU requests)  SAML or OIDC federation cost (monthly): 6.75 USD  5,000 MAUs - 50000 free MAU requests per month = -45,000.00 billable MAU requests  Max (-45000.000000 billable MAU requests, 0 Constant Unit) = 0.00 total billable MAU requests  Tiered price for: 0.00 MAUs  Total tier cost = 0.00 USD (User Pool MAUs)  User Pool MAU cost (monthly): 0.00 USD  Advanced security feature cost (monthly): 0 USD  **Cognito MAU cost (monthly): 6.75 USD** | **6.75 USD** |
| **AWS SWF** | 3,000 executions per month x 0.0001 USD = 0.30 USD (Workflow Executions cost)  20 Tasks, Markers, Timers and Signals x 3,000 executions per month x 0.000025 USD = 1.50 USD (Tasks, Markers, Timers and Signals cost)  30 days + 60 days = 90.00 Workflow days  90.00 days x 3,000 executions x 0.000005 USD = 1.35 USD (Open and retained workflow cost)  0.30 USD + 1.50 USD + 1.35 USD = 3.15 USD (Simple Workflow Cost)  **Simple Workflow Cost (monthly): 3.15 USD** | **3.15 USD** |
| **Total:** | | **1.480,12 USD** |

*Figure 1.*