Assignment 3

Dang Khoa Le (individual project)

Student ID: 103844421 Tutorial Wednesday, 06.30PM.

***Abstract*: This document presenting the demonstration for Assignment 3. The assignment task is based on AWS academy infrastructure (without actual interaction).**

This assignment will create a design that facilitate an appropriate business infrastructure using AWS environment. The design will outline all compulsory demand and how it may satisfy the business scenario circumstance, under the most effective way, in term of cost consideration and operation efficiency.

I. Business Scenario:

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

1. Where possible the company would like to use managed cloud services to minimise the need for in-house systems administration. Photo and other media will be stored in AWS S3.

2. The company is not sure how demand for its application will grow in the future but over recently it has been doubling every 6 months. It expects this trend will continue for the next 2 or 3 years at least and it wants the architecture to be able to cope with this growth.

3. The current system EC2 instances are running on t2.micro. The compute capacity is regularly exceeding the 80% performance limit with 6 instances running. The desired load needs to decrease to between 50 and 60%. Ignore this requirement if your solution does not involve EC2.

4. The company would like to adopt a serverless/event-driven solution.

5. The relational database is relatively slow and costly to run. Given the simple table structure, the company would like to explore more cost-effective options.

6. The application has had wide uptake around the world but response time in countries other than Australia has been relatively slow. Global response times need to be improved.

7. It is expected the system will be extended to handle video media in the future.

8. The media can be uploaded by users in all sorts of formats. The company would like various versions of media to be automatically produced (e.g. thumbnails, low resolution versions suitable for mobile phones, or video transcoding). The process for reformatting/transcoding/reprocessing media should meet the following criteria:

a. When a media item is uploaded to the S3 bucket, the creation of these alternative versions should be triggered automatically. Transformed media will also be stored in S3.

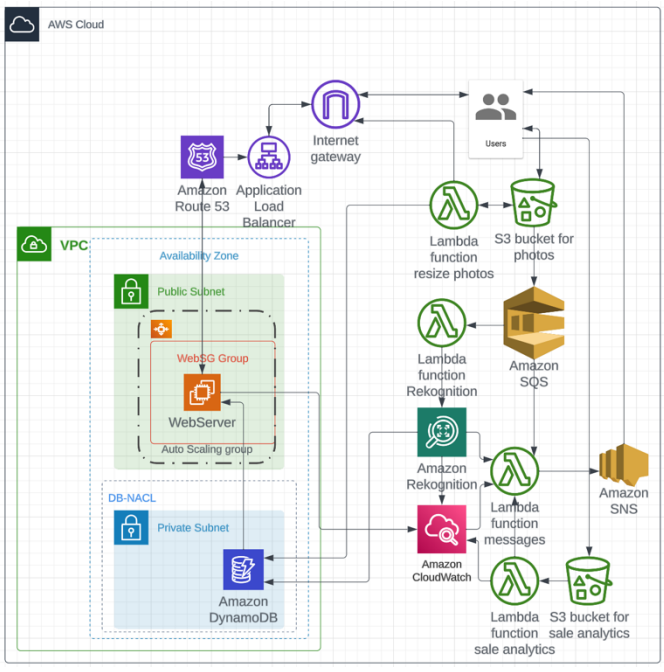
b. The architecture for processing media should be extensible. For example, in the future it may be desirable to add the ability to automatically identify tags in photos using AI.
c. Different processing services should be able to be run on the most suitable platform – e.g. EC2 instance, Lambda, or other AWS managed services. Given cost and performance constraints it is assumed that all services will be provided in the AWS ecosystem.

d. The reprocessing/reformatting of media is often a time consuming task. The architecture should be designed so that the application does not become overloaded and is effectively decoupled.

For example, multiple 'worker' nodes can process transformation jobs that have been placed on a queue. The worker nodes may specialise in particular tasks. For example, one node may specialise in video transcoding which is much more processor and memory intensive than reformatting a photograph.

You need to create a report to the client outline a design for the new system and justifying why that design is best.

II. Architectural Diagram:



III. Description and Justification of Services:

1. **Amazon S3***:* Store media, including photos and videos. Utilize S3's scalability and durability for efficient storage.

2. **Amazon CloudFront:** Implement a content delivery network (CDN) to accelerate content delivery worldwide, improving global response times.

3. **Amazon Route 53:** Route traffic globally for low-latency access. Leverage Route 53's DNS service for domain management.

4. **Amazon Elastic Load Balancer (ELB):** Distribute incoming application traffic across multiple Amazon EC2 instances. Achieve high availability and fault tolerance.

5. **Amazon Auto Scaling:** Automatically adjust the number of Amazon EC2 instances to maintain a desired CPU utilization level. This ensures cost optimization and performance improvement.

6. **EC2 Instance:** WebServer instance serves as the web server and application host. This instance interacts with other services to manage user interactions, including media uploads and sales analysis requests.

7. **Security Group and NACL:** WebSG security group allows SSH, HTTP and HTTPS traffic to the WebServer, allowing the EC2 instance work appropriately. DB_NACLs also allow and ensure all appropriate traffic flow to the DynamoDB in the private subnet.

8. **Amazon CloudWatch:** Monitor resources and application performance. Use custom metrics for EC2 instances to trigger Auto Scaling events. CloudWatch alarms can also be triggered for Rekognition and Sale Analytic events (additional function).

9. **AWS Lambda:** Implement serverless functions for media processing (image and video transcoding). Events such as new media uploads can be resized, Rekognition AI detection, sales analytic and SNS send notification functions can be executed using Lambda.

10. **IAM Role:** An IAM role was created and assigned to the Lambda, EC2 instance, and S3 to allows the WebServer instance interacting with other AWS services in this design. Another IAM role was also specified to the DynamoDB, which enable the system to specify which resources (EC2 instances or Lambda functions) can perform actions on your DynamoDB tables.

11. **Amazon DynamoDB:** Utilize a NoSQL database for efficient storage and retrieval of metadata associated with media files.

12. **Amazon Rekognition:** Leverage AWS's computer vision service for media analysis, such as identifying tags in photos, stock condition, date of expiry, which minimise human labour and optimise workload.

13. **Amazon Simple Queue Service (SQS):** Decouple components by using SQS for queuing media processing tasks, allowing worker nodes to pick up and process tasks independently.

14. **Amazon Simple Notification Service (SNS):** Use SNS to send notifications when media processing tasks are completed. This SNS service is operated by using a designated Lambda function, which can also categorise and execute message (to the user/subscriber).

IV. Sale Analysis Integration (Additional Design):

1. **Data Collection:** First, a mechanism to collect sales data regularly is implemented. This could involve a database or other data storage that keeps track of monthly sales, expenses, net loss, and net revenue.

2. **Scheduled Lambda Function:** Create a Lambda function that is scheduled to run monthly. This scheduled function will generate the sales analysis report for the current month. Lambda supports scheduled events using CloudWatch Events.

3. **Sales Analysis Logic:** Inside the Lambda function, implement the logic to calculate the sales analysis. This includes fetching data for the current month, processing it to calculate metrics like net revenue, and potentially comparing it to the previous month.

4. **Report Generation:** Use Lambda to generate the report. Reports can be generated in various formats, including HTML or PDF, depending on the business requirement.

5. **S3 Storage:** The report can be stored it in an S3 bucket. This is not only for backup but also for sharing the report via a download link.

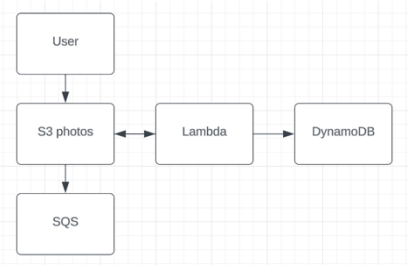
6. **SNS Topic:** Create an SNS topic to which Lambda function that can publish the message. The message will include the download link to the sales analysis report.

7. **SNS Subscriptions:** Set up email subscriptions for the SNS topic. Subscribers (in this case, the business owner's email address) will receive email notifications when a new message is published to the SNS topic.

8. **Email Message Format:** Ensure the SNS message contains the necessary information, such as the report download link, they will be structured the email message in a readable format with tables and visualizations.

V. UML COLLABORATION DIAGRAMS:

Media Upload Collaboration:



◊ User uploads photos and reports to Amazon S3.

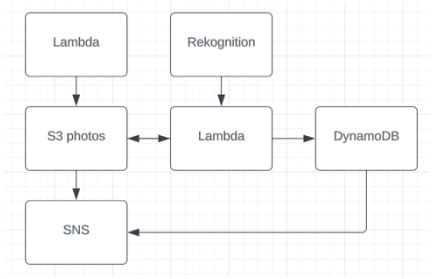
◊ Amazon S3 triggers an event, notifying Lambda functions.

◊ Lambda function processes the media (e.g. creates thumbnails the same in the previous Assignment 2).

◊ Metadata is stored in DynamoDB.

◊ Using SQS decouple tasks. The system also sends an SNS notification for task completion.

Media Processing Collaboration:



◊ Lambda functions are triggered by events (e.g., media upload, processing request).

◊ Lambda processes the media, using Rekognition for tagging, as needed.

◊ Processed media and metadata are stored in S3 and DynamoDB.

◊ All notifications are sent using SNS.

VI. DESIGN RATIONALE:

1. Business Scenario Fulfillment:

◊ Using AWS's managed services reduces in-house system administration, meeting the requirement.

◊ Scalability ensures the architecture can handle the projected growth.

◊ Serverless solutions are adopted for cost-efficiency and event-driven processing.

◊ AWS services improve global response times.

◊ The design is extensible, aligning with future requirements like AI for tagging.

◊ Decoupling via SQS and Lambda supports efficient processing and avoids overloading.

2. Additional Solutions:

◊ Utilized AWS services are chosen based on specific use cases and needs, optimizing performance and cost.

◊ Leveraging serverless computing reduces operational overhead.

◊ NoSQL DynamoDB is selected for its simple table structure and scalability.

◊ Amazon Rekognition is chosen for AI-based tasks.

◊ Multi-tier architecture aligns with scalable and fault-tolerant requirements.

3. Design Criteria:

Performance and scalability are achieved through scalable services and global reach.

Reliability is ensured through AWS's managed services.

Security is addressed through AWS IAM and best practices.

Cost-efficiency is achieved through serverless computing and resource optimization.

4. Justification for Selection:

◊ The selected services effectively fulfill the business scenario.

◊ AWS services are chosen for their scalability, reliability, and manageability.

◊ Serverless computing improves cost-effectiveness and scalability.

◊ Decoupling components enhances system efficiency.

◊ This is a high-level overview of a potential AWS architecture design. The document is able to provide more in-depth analysis, including budget considerations and a detailed description of the design's components.

VII. DESIGN BUDGET ESTIMATION:

Service	AWS Charging Rate (USD per month)	Estimation (USD per month)	Justification (month = 30 day)
Amazon S3	$0.023 per GB for the first 50 TB.	0.05	Estimation of 100 (20kB) images per day.
AWS Lambda	$0.20 per 1 million requests.	0.20	3000 S3 request and 30 Rekognition request.
Amazon Route 53	$0.20 per 1 billion queries per month (Australia).	0.20	Less than 1 billion queries per month.
EC2	$0.0464 per hour for t2.medium (on-demand price, Australia).	22.27	Instance work continously for 16 hours a day.
Security Group and NACL	Security Group and NACL are free.	0.00	Security Group and NACL are free.
Amazon Elastic Load Balancer	$18.44 per Load Balancer per month.	18.44	1 ELB was used.
Amazon Auto Scaling	Auto Scaling itself is free.	0.00	Auto Scaling itself is free without any EC2 instance.
Amazon CloudFront	$0.114 per GB for the first 10 TB of data transfer (Australia).	34.20	Estimation of 5 gB transferred per day.
Amazon CloudWatch	Each custom metric is $0.30 and each alarm is $0.10 per alarm, per month.	3.60	10 metrics and 7 alarm was used.
Amazon DynamoDB	per million read request units (on-demand).	0.45	10000 write and read request units a day.
Amazon Rekognition	$0,001 per processed image (plus extra with addition functionality features).	3.00 to 5.00	100 images are processed everyday.
Amazon SQS	$0.40 per 1 million requests.	1.20	3000 images per month, approximately 3m requests.
Amazon SNS	$0.50 per 1 million requests.	1.50	3000 images per month, approximately 3m requests.
TOTAL		82.11	

The table above illustrates the estimation of budget (per monthly) in this design with the relevant AWS services (the cost is calculated as of 2023, based on Australia/NZ region).

It is also noticeable that the charging fee is USD, and the overall cost may be changed as this table only shown the on-demand pricing from AWS website. Estimation and Justification of the budget may also not explicitly demonstrate the real-life cost of this project design concept.