CSCI5409: Adv Topic in Cloud Computing

Winter 2024

"Term Assignment"

Professor

Prof. Robert Hawkey

Student

B00955671

Sumit Mansukhlal Savaliya

sm572004@dal.ca

Repository Link:

https://git.cs.dal.ca/courses/2024-winter/csci4145-5409/ssavaliya

9th April 2024

Table of Contents

Project Details: PDF-Image Parser

The PDF-Image Parser is an innovative cloud-based service tailored for businesses operating within the customized fashion industry. Its primary function is to facilitate the extraction of images from PDF files, specifically designed to cater to the needs of companies involved in apparel design and customization.

As a critical tool for graphics designers, the PDF format is widely utilized for sharing apparel designs. However, extracting images from these files can be a tough task. The PDF-Image Parser service simplifies this process by offering users the ability to parse images directly from PDFs and obtain public links to seamlessly integrate these images into their websites.

The intended users of this service encompass a broad spectrum within the customized fashion business, including but not limited to apparel brands, fashion designers, print-on-demand services, and e-commerce platforms specializing in custom apparel.

Key functionalities of the service include:

1. Image Extraction from PDF:
    - Can parse images from a given pdf in the form of base64.
2. Storage on S3:
    - Facility to save the extracted images in an S3 bucket for easy access and generation of public links.
3. Link Management in RDS and Email Distribution:
    - Storage of the generated public access links in RDS (Relational Database Service).
    - Sending of the links to users via email, utilizing the email addresses provided in the request.

Performance targets for the PDF-Image Parser service:

1. **Reliability**: Ensure high availability and uptime (EC2).
2. **Scalability**: Handle increasing loads efficiently (Load Balancer).
3. **Response Time**: Provide fast extraction and link generation (Lambda).
4. **Throughput**: Process multiple requests concurrently.
5. **Data Integrity and Security**: Securely store and handle data (API Gateway, S3 and RDS).

Services and Comparison

Compute

AWS EC2:

- **Functionality:** Deployed backend for handling storage and email service. EC2 instances provide scalable compute capacity in the cloud and are suitable for running long-running applications or services that require persistent compute resources.
    - Full control over the virtual server environment.
    - Ideal for services requiring continuous availability.

AWS Lambda:

- **Functionality:** Used to parse images from the given PDF (base64 encoded). Lambda functions are event-driven, serverless compute services that automatically scale to match the incoming request rate. They're well-suited for short-lived, stateless tasks like image parsing from PDF files.
    - Automatic scaling: Lambda automatically scales out (and in) based on the number of incoming requests.
    - Cost-effectiveness: Lambda incurs costs only when functions are executed, making it more cost-effective for sporadic workloads compared to running and maintaining EC2 instances.
    - No server management: With Lambda, there's no need to provision or manage servers, operating systems, or runtime environments. This reduces operational overhead and simplifies deployment and maintenance tasks.
- **Reason for Choosing AWS Lambda over EC2:** The decision to use AWS Lambda over EC2 was primarily driven by the nature of the workload, which consists of short-lived, stateless tasks like image parsing from PDF files. Lambda's automatic scaling, cost-effectiveness for sporadic workloads, and serverless architecture align well with the requirements of the application.

Price Comparison: EC2 Vs Lambda

let's consider a hypothetical scenario where we need to run an application for 24 hours a day for one month. We'll calculate the costs for both services based on this scenario.

EC2: t3.medium (2vCPU, 4GB RAM)

Total cost on Demand = $140.41 Per Month

Lambda: $0.0000166667 per GB-second**.** $0.20 per 1M requests

Assuming 0 requests for the getting the operational cost and the application's runtime requires 1 GB of memory, and each request takes 1 second to process.

Duration cost:

- Total GB-seconds = 1 GB * 2,592,000 seconds = 2,592,000 GB-seconds
- Cost for GB-seconds = $0.0000166667 * 2,592,000 = $43.20

Total Lambda cost = $43.20

As we can see, there is a clear difference of price between EC2 and Lambda. The only way lambda could exceed the estimates is when there is a huge unreal spike in requests.

Storage

AWS RDS:

- **Functionality:** Stores parsed data obtained from Lambda functions and S3, housing crucial information such as email addresses, unique identifiers, and public links for parsed images.
    - Ideal for services necessitating continuous availability, ensuring reliability and accessibility for users.
- **Reason for Choosing AWS RDS over DynamoDB:** RDS was chosen over DynamoDB due to its support for structured data, flexible querying with SQL, transaction support, reliability, and scalability options.

Price Comparison: RDS Vs DynamoDB

DynamoDB is definitely a cheaper option as compared to RDS. But due to the storage structure requirements, using RDS was an optimal choice for the service.

AWS S3:

- **Functionality:** Used for storing parsed images in JPG format, enabling users to access and download them at a later time.
    - S3 incurs charges based on actual usage, aligning well with the sporadic workload nature of image storage.
    - Automatic scaling capabilities facilitate seamless management of storage resources in response to varying demand.

Network

AWS API GATEWAY:

- **Functionality:** Acts as an entry point for client requests, triggering Lambda functions tasked with parsing images from PDF documents and delivering them in base64 encoding.
    - Enables straightforward invocation of Lambda functions via HTTP endpoints, ensuring prompt execution of image parsing tasks.
    - Provides robust security features such as authentication mechanisms and request validation, safeguarding endpoints from unauthorized access and malicious attacks.

General

AWS ELASTIC LOAD BALANCER (ELB):

- **Functionality:** Serves as a traffic distribution mechanism, distributing incoming requests across EC2 instance to ensure optimal workload distribution and high availability.
    - Enhances fault tolerance by automatically rerouting traffic.
    - Seamlessly integrates with other AWS services such as AWS CloudWatch for monitoring.

AWS SNS (Simple Notification Service):

- **Functionality:** Utilized for sending email notifications containing parsed image links to users.
    - Facilitates the rapid delivery of email notifications to users, ensuring timely communication of relevant information.
    - Offers high reliability and durability in message delivery, minimizing the risk of message loss or delivery delays.

Deployment Model

I chose to use AWS Public Cloud because it offers many helpful services and makes it easy to keep things running smoothly. With AWS, I can ensure high availability of resources and work better with infrastructure without worrying about costs. It is easy to scale up and scale down based on demand allowing efficient development and delivery.

Delivery Model

I have picked Software as a Service (SaaS) as delivery model. It lets me access service from any device. The infrastructure can handle lots of users at once and takes care of all the maintenance and updates. This means things get up and running quickly, and it's cost-effective for companies / users using this service.

Architecture

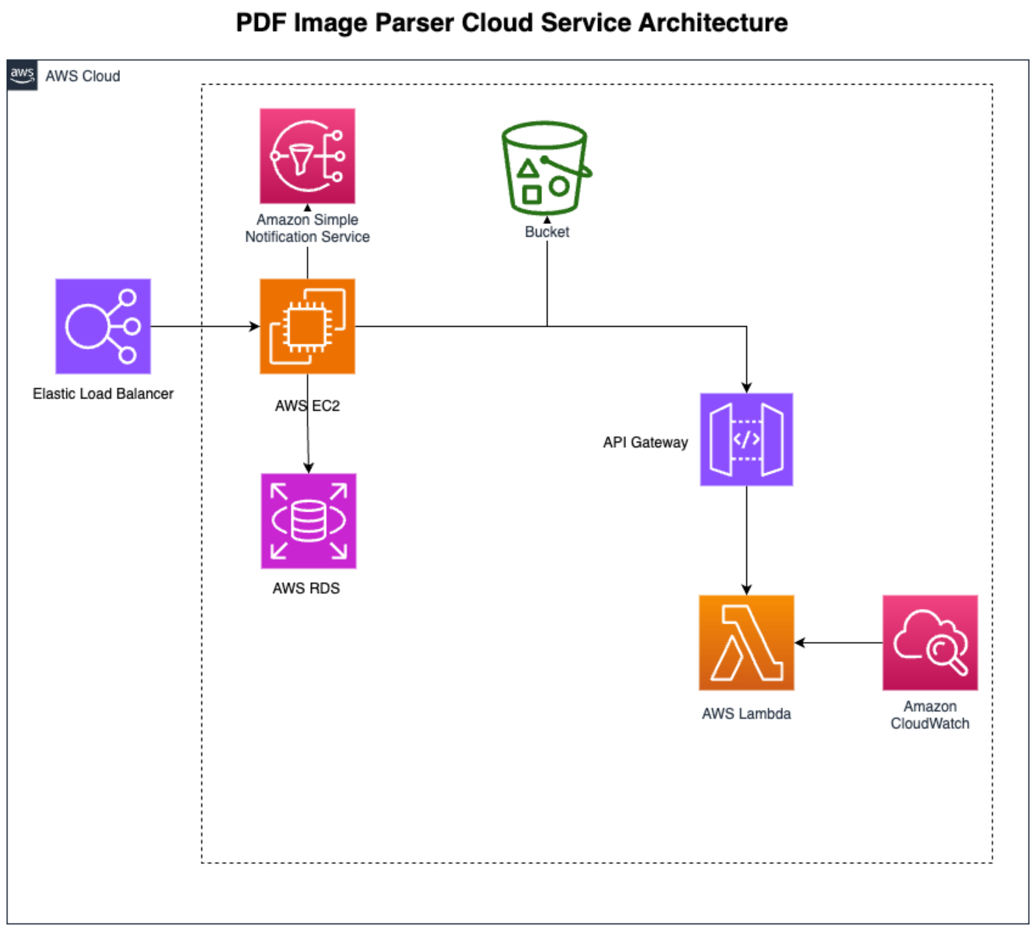

Figure 1: *Architecture Diagram [1]*

How do all the cloud mechanisms fit together to deliver your application?

The user interacts with the service using load balancer's public IP / DNS. This is the only entry point for the traffic over this service.

From here on, request is forwarded to EC2 where data is decoupled and sent to Lambda for parsing images.

API Gateway is used to invoke lambda and pass data securely. Once lambda finishes the parsing (i.e. 5s), it returns all the parsed image to EC2 in base64 format.

Here on, EC2 stores the image in S3 and generates public access links for the stored images.

These links are then stored to RDS for record purpose and email to user provided email in the request using SNS.

Where is data stored?

The data is stored in S3 and RDS. RDS is used for storing relational data to maintain history and parsed data links. S3 is used to store the parsed images.

What programming languages did you use (and why) and what parts of your application required code?

Two key components drive my application:

JavaScript and TypeScript. JavaScript is utilized for the EC2 Backend, powered by Node.js. It's responsible for data cleaning before passing requests to Lambda, as well as handling post-processing tasks like storing to S3, referencing in RDS, and sending Email Notifications to users. On the other hand, TypeScript is employed for coding the Lambda function. This choice is primarily influenced by its extensive library support for achieving image parsing functionality swiftly. TypeScript facilitates rapid development and minimizes type errors, ensuring fault tolerance in the Lambda application.

How is your system deployed to the cloud?

EC2 Backend: The EC2 Backend is deployed using User Data in EC2 via CloudFormation. The process involves fetching the code from GitHub and executing commands to run the application on the required port (80). Initially, Node.js and NPM are installed to enable application execution.

Lambda Image Parsing: the code is deployed on Lambda using Docker and ECR. Docker deployment was chosen due to OS dependencies inherent to the image parsing service, which necessitated containerization. Certain C++ OS level dependencies required for the service could only be installed by containerizing the application.

Architecture choice:

The outlined architecture adheres to cloud-native application design principles, leveraging services such as Lambda, EC2, SNS, RDS, etc. The system is meticulously crafted to prioritize scalability, fault tolerance, and security. Moreover, logging and disaster recovery measures are integrated using CloudWatch log groups to enable robust monitoring.

Security

The provided architecture aims to ensure data security across all layers.

The architecture leverages AWS services like Lambda, EC2, SNS, RDS, and S3, which are designed with robust security features. These services offer encryption, access controls, and compliance certifications to protect data at rest and in transit.

Security groups and subnets are specified for EC2 instances and load balancers to control inbound and outbound traffic, minimizing exposure to potential threats.

The RDS Instance is not given public access. It is only accessible to EC2 which makes the data securely accessible only to authorised users.

Existing Vulnerabilities

The Lambda function is accessible through API gateway which leaves a vulnerability if URL is leaked. The possible solution for this is to enable authentication and authorisation in the project using tokens in request headers.

Cost to Replicate in Private Cloud

To replicate this cloud service, the organisation will have to make multiple hardware purchases ranging from servers to gateways.

1. **Servers:** The company will need to purchase servers with minimum (1GB RAM, 1vCPU).
    - **Estimated Cost:** $606 per month including maintenance costs. (0.35 vCPU/0.7GB RAM/100GB Storage) [2]
2. **Storage Devices:** Storage devices such as HDD, SSD will need to be purchased with minimum storage of 100GB. This requirement will keep fluctuating upon scaling and is harder to maintain/replace when the application scales.
    - **Estimated Cost:** $129. (1TB SSD Storage)
3. **Networking Equipment**: Routers, switches, and firewalls would be necessary to build the network infrastructure and ensure connectivity between different components.

    - **Estimated Cost**: $2500-$4500 with an average price of $3800. (Excludes setup and maintenance cost)

4. **Operating System**: we will need an operating system installed on each server for Lambda and EC2 applications, such as Linux or Windows Server, to run the necessary software components due to their OS level dependencies.
    - **Estimated Cost**: Free (Linux)
5. **Firewalls and Intrusion Detection Systems (IDS)**: To protect the network from unauthorized access and potential cyber threats.
    - **Estimated Cost**:
        1. Licence Fee: $1000 - $2000 / Year
        2. Equipment: $1000 - $10,000 (Excluding Maintenance)
        3. One Time Installation: $1400 - $2000
        4. Network Engineers: $77,782 Per Year in Canada [3].

Estimated Total Cost to setup a Private on Premises Infrastructure:

$8000 Per Month (Includes equipment + Maintenance Cost) + $77,782 per year per employee.

Note: This cost does not include disaster recovery or data backup. In case of such events, we can assume total loss of data and additional infrastructure cost.

Monitoring

AWS Lambda function deployed through API Gateway is likely to have the most potential to cost the most money. AWS Lambda charges are based on the number of requests and the compute time consumed by the function. If there's a sudden increase in the number of incoming requests or if the function starts consuming more compute resources due to inefficient code or unexpected workload spikes, it could result in significant cost escalation. Therefore, adding monitoring to the AWS Lambda function would be crucial to ensure costs do not escalate out of budget unexpectedly. Additionally, setting up alerts based on predefined thresholds can help notify of any abnormal behavior or cost spikes.

Future Development

In the future, this service has the potential to undergo a significant architectural overhaul by leveraging AWS SQS (Simple Queue Service), which operates on a consumer and producer mechanism. The proposed flow can be redesigned to incorporate SQS for triggering Lambda functions. Upon triggering, the Lambda function can initiate the parsing process. Upon successful completion of parsing, another Lambda function can be invoked to store the parsed images to both S3 and RDS. Finally, another Lambda function can be triggered to send email notifications to the specified user via SNS. This revamped architecture offers improved scalability, reliability, and decoupling of components, enhancing the overall efficiency and performance of the service.

This development also has potential flaws which might spike the cost as we will me employing more lambda functions. There might be an increased influx of requests which could spike the cost of lambda functions.

References

[1] "draw.io", *draw.io* [Online]. Available: https://app.diagrams.net/. [Accessed: April 09, 2024]

[2] "How much does a server cost in 2024?", *Intelligent Technical Solutions*. [Online]. Available: https://www.itsasap.com/blog/server-cost. [Accessed: April 09, 2024]

[3] "Computer Network Engineer Salary in Halifax", *ECONOMIC RESEARCH INSTITUTE* [Online]. Available: https://www.erieri.com/salary/job/computer-network-engineer/canada/nova-scotia/halifax#:~:text=Salary%20Recap,for%20a%20Computer%20Network%20Engineer. . [Accessed: April 09, 2024]